

Proceedings of the

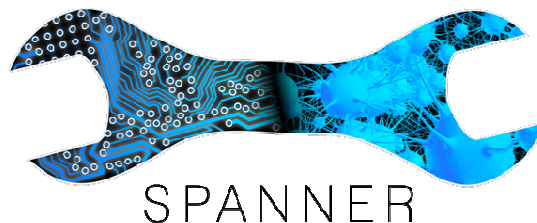
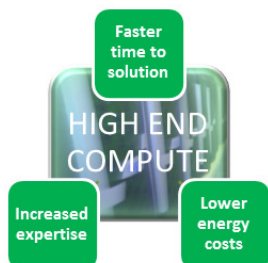
**EMerging Technology
(EMiT) Conference 2019**



9-11 April 2019
University of Huddersfield, U.K.

Edited by M.K. Bane and V. Holmes

Sponsors:



<https://emit.tech>



Welcome to Huddersfield and the fifth Emerging Tech conference.

Building upon events in Manchester and Barcelona during the last 5 years, EMiT continues to explore the challenges of hardware, software, tools and algorithms that are coming over the horizon or that the EMiT community is helping create itself. As this year's conference will explore, opportunities are opening via quantum and neuromorphic computing, with still a large number of challenges posed by machine learning as well as new possibilities for likes of FPGA, GPU and CPU programming. As the keynotes will discuss, we cannot treat hardware in isolation; to maximise our research and R&D (both in academia and industry) we need to take a holistic (co-design) approach and actively consider which emerging techniques may best accelerate our work, and particularly the intersection of new hardware and new techniques.

EMiT is more than a yearly conference. With an active Twitter account (@emit2019) and LinkedIn group, we promote dialogue on all the EMiT themes, as well as provide links and updates on relevant news and research. We aim to expand on this in coming months by building up reference materials on topics suggested by our members. But rather than get ahead of ourselves, we hope you enjoy this year's conference, use both these proceedings and the forthcoming IET publications and share your experiences using the hashtag #EMiT2019

EMiT2019 will be held at The University of Huddersfield in the United Kingdom. The University of Huddersfield has a long history of innovation and supporting emerging technologies. In 2012, it was named the Times Higher Education Entrepreneurial University of the Year and in 2013, Professor Elizabeth Towns-Andrews, the University's 3M Professor of Innovation, won the Queen's Award for Enterprise Promotion. In April 2017, The University of Huddersfield joined the Times Higher Education's world top 200 "young universities".

The Organising Committee has worked hard to realise EMiT2019, so please join us in extending your appreciation to each of them. We would also like to thank each keynote, everybody submitting papers or posters (accepted or not), as well as our sponsors, stall holders and the Yorkshire Sculpture Park, for showing their support to the EMiT series.

We are looking towards EMiT2020 so if you are inspired by this year's conference, either to host or join the Organising Committee, please do speak to us in person or email us at info@emit.tech

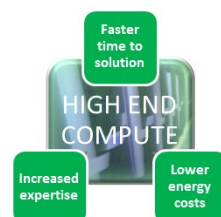
Yours,

EMiT (<https://emit.tech>)

Violeta Holmes, the University of Huddersfield

Michael Bane, High End Compute Ltd
(<https://highendcompute.co.uk>)

University of
HUDDERSFIELD
Inspiring tomorrow's professionals



EMiT2019 Local Organising Committee

Dr. Violeta Holmes	University of Huddersfield
Dr. Anju P. Johnson	University of Huddersfield
Dr. Faheem A. Khan	University of Huddersfield
Dr. Mahmoud Dhimish	University of Huddersfield
Ms. Rebecca Marsden	University of Huddersfield

EMiT2019 Local Organising Committee

Dr. Michael K. Bane	High End Compute Ltd
Dr. Stephen Longshaw	STFC
Prof. Benedict Rogers	University of Manchester
Dr. David Topping	University of Manchester
Dr. Javier Navaridas	University of Manchester

EMiT2019 Advisory Panel

Prof. Jack Dongarra	University of Tennessee
Dr. Kirk E. Jordan	IBM

Printing

The EMiT2019 committees shall not be held responsible for any statement or opinion advanced in papers or otherwise printed in this volume. Authors' papers have been prepared for final reproduction/printing from supplied PDFs without any changes and the authors are fully responsible for information contained in their papers.

Copyright © Michael Bane, EMiT (Emerging Tech) Conference series

Published by
EMiT/University of Huddersfield/High End Compute Ltd/University of Manchester

Proceedings of the 2019 Emerging Technology Conference

9-11 April 2019, University of Huddersfield, Huddersfield, U.K.



ISBN: 978-0-9933426-4-6

Day 1- Tuesday 9th April 2019

The first day of EMIT 2019 is dedicated to two technical workshops that will run in parallel.

1. Deep Learning workshop using both NVIDIA GPUs and a new Neuromorphic platform called SpiNNaker via a software toolchain called SPANNER. A laptop will be required for this workshop.
2. Quantum computing workshop delivered by IBM Research. This will provide a mixture of taught and practical elements, including use of real quantum hardware. Bring your own laptop or a number of fixed workstations will be available on a first come, first served basis.

Day 2 - Wednesday 10th April 2019

08:00 - 09:30	Registration and Refreshments
09:30 - 10:00	Conference Introductions
(15 minutes)	Welcome to the conference and introduction to EMIT 2019 Dr Violeta Holmes, University of Huddersfield
(15 minutes)	Welcome from the Director of Research and Enterprise at Huddersfield, Prof Liz Towns-Andrews
10:00 - 10:45	Keynote Presentation: Prof. Veljko Milutinovic, Indiana University DataFlow SuperComputing for Big Data Analytics
10:45 - 11:10	Break & Posters
11:10 - 12:10	Session 1: Novel Hardware
(20 minutes)	M. Ashworth, G. Riley, A. Attwood & J. Mawer Prospects for Low-power Acceleration of HPC Workloads in EuroExa: FPGA Acceleration of a Numerical Weather Forecast Code
(20 minutes)	J. Lant & J. Navaridas Direct Communication between Distributed FPGA Resources
(20 minutes)	P. A. Bogdan, G. P. Garcia, S. Davidson, M. Hopkins, R. James & S. Furber Event-based computation: Unsupervised elementary motion decomposition
12:10 - 13:10	Lunch & Posters
13:10 - 14:30	Session 2: Deep Learning
(20 minutes)	S. Al-Riyami An efficient way to deal with algorithmically generated data in deep learning
(20 minutes)	M. Seedall, V. Holmes & K. Macfarlane SafeChat System with NLP and DNN
(20 minutes)	A. W. Qurashi & V. Holmes Comparison of Deep Neural Network approach in text and image classification using CPU and GPU systems
(20 minutes)	H. Alattal, F. Khan & Q. Ahmed Denosing an Image Using Deep Learning Techniques
14:30 - 15:00	Break & Posters
15:00 - 16:00	Session 3: GPU Computing
(20 minutes)	J. Grasset, Y. Audouin, S. Longshaw, C. Moulinec and D. Emerson Porting and Optimising TELEMAT-MASCARET for the OpenPOWER Ecosystem
(20 minutes)	M. Turchetto, R. Vacondio and A. Dal Palù Multi-GPU implementation of a 2D Shallow Water Equations Solver over a Multi-Resolution grid
(20 minutes)	V. Stegailov, N. Kondratyuk, G. Smirnov and A. Timofeev The Desmos supercomputer for computational materials science
16:30 - 23:00	Visit to Yorkshire Sculpture Park and evening banquet meal (including travel from and to The University of Huddersfield)



Day 3 - Thursday 11th April 2019

08:00 - 09:00	Registration and Refreshments
09:00 - 09:45	Keynote Presentation: Dr Torsten Hoefler, ETH Zurich High-Performance Communication in Machine Learning
09:50 - 10:30	Session 4: Novel Communication Systems
(20 minutes)	M. Kynigos, J. Navaridas and J. A. Pascual Scalability of a Silicon Photonic Switch for High-Performance Interconnects
(20 minutes)	Z. Elsarraf, F. Khan and Q. Ahmed Performance Analysis of Code-Domain NOMA in 5G Communication Systems
10:30 - 11:00	Break & Posters
11:00 - 12:30	Keynote Presentation & Panel Discussion
(45 minutes)	Dr Peter Hopton, ICEOTOPE Hardware and Infrastructure Technology Challenges to Achieving ExaScale – and EuroExa's Answer
(40 minutes)	Chaired Panel Discussion
12:30 - 13:30	Lunch & Posters
13:30 - 14:30	Session 5: Novel Software
(20 minutes)	L. Ragta, J. Meng, X. Gu and D. Emerson A high level abstraction approach for lattice Boltzmann simulations using future computing systems
(20 minutes)	S. Titarenko, V. Titarenko, G. Aivaliotis and J. Palczewski Constraint-based frequent pattern mining algorithm and its optimisation for multicore systems
(20 minutes)	H. Aagela and V. Holmes Cloud Robotics-Based System for Robot Teleoperation
14:30 - 15:00	Break & Posters
15:00 - 16:00	Session 6: Applications of Emerging Tech
(20 minutes)	M. Vallati and A. Grassi AI to Facilitate Legal Analysis in the PESTLE Context
(20 minutes)	V. Elisseev Holistic Approach to Energy and Power Management in HPC
(20 minutes)	H. Aagela and V. Holmes Collaborative Cloud-based Face recognition approach for Humanoid robots
16:00 - 16:30	Conference Summary & Close
(30 minutes)	Dr Michael Bane, High End Compute EMIT 2019 Summary

EMiT|2019

Emerging Technology Conference

University of
HUDDERSFIELD
Inspiring global professionals

EMiT2019 Emerging Technologies Conference is being within the University of Huddersfield.

Huddersfield is a large market town in West Yorkshire, England. It is the 11th largest town in the UK, with a population of over 162k. Huddersfield itself is a town of Victorian architecture. The railway station is a Grade I listed building described by John Betjeman as "the most splendid station façade in England", second only to St Pancras, London. The station in St George's Square was renovated at a cost of £4million and subsequently won the Europa Nostra award for European architecture.

Huddersfield is within the historic county boundaries of the West Riding of Yorkshire, it is the largest urban area in the metropolitan borough of Kirklees. The town is known for its role in the Industrial Revolution, and for being the birthplaces of rugby league, the Labour Prime Minister Harold Wilson, and the film star James Mason.

Close by is the famous Brontë Country that is less than an hour drive from Huddersfield. The name comes from the Brontë sisters, who wrote such literary classics as Jane Eyre, Wuthering Heights, and The Tenant of Wildfell Hall, while living in the area.

The Yorkshire Sculpture Park who will host the conference banquet dinner, is set in beautiful fields, hills, woodland, lakes and formal gardens combined to create a beautiful landscape and stunning setting.

We sincerely hope you will join us in being part of this exciting event.

EMiT|2019

Emerging Technology Conference
9-11th April, Huddersfield, UK

CONTENTS

KEYNOTE LECTURES	9
DataFlow SuperComputing for Big Data Analytics <i>V. Milutinovic</i>	9
High-Performance Communication in Machine Learning <i>T. Hoeftler</i>	10
PEER REVIEWED PAPERS	11
Session 1: Novel Hardware	12
Prospects for Low-power Acceleration of HPC Workloads in EuroExa: FPGA Acceleration of a Numerical Weather Forecast Code <i>M. Ashworth, G. Riley, A. Attwood, J. Mawer</i>	12
Direct Communication between Distributed FPGA Resources <i>J. Lant, J. Navaridas</i>	16
Event-based computation: Unsupervised elementary motion decomposition <i>P.A. Bogan, G.P. Garcia, S. Davidson, R. Hopkins, R. James, S. Furber</i>	20
Session 2: Deep Learning	25
An efficient way to deal with algorithmically generated data in deep learning <i>S. Al-Riyami, A. Lisitsa, F. Coenen</i>	25
SafeChat System with NLP and DNN <i>M. Seedall, V. Holmes, K. Macfarlane</i>	28
Comparison of Deep Neural Network approach in text and image classification using CPU and GPU systems <i>A.W. Qurashi, V. Holmes</i>	32
Denoising an Image Using Deep Learning Techniques <i>H. Alattal, F. Khan, Q. Ahmed</i>	36
Session 3: GPU Computing	41
Porting and Optimising TELEMAT-MASCARET for the OpenPOWER Ecosystem <i>J. Grasset, Y. Audouin, S. Longshaw, C. Moulinec, D. Emerson</i>	41
Multi-GPU implementation of a 2D Shallow Water Equations Solver over a Multi-Resolution grid <i>M. Turchetto, R. Vacondio, A. Dal Palu</i>	45
The Desmos supercomputer for computational materials science <i>V. Stegailov, N. Kondratyuk, G. Smirnov, A. Timofeev</i>	46
Session 4: Novel Communication Systems	48
Scalability of a Silicon Photonic Switch for High-Performance interconnects <i>M. Kynigos, J. Navaridas, J.A. Pascual</i>	48
Performance Analysis of Code-Domain NOMA in 5G Communication Systems <i>Z. Elsaraf, F. Khan, Q. Ahmed</i>	52
Session 5: Novel Software	57
A high level abstraction approach for Lattice Boltzmann simulations using future computing systems <i>L. Ragta, J. Meng, X. Gu, D. Emerson</i>	57
Constraint-based frequent pattern mining algorithm and its optimisation for multicore systems <i>S. Titarenko, V. Titarenko, G. Aivaliotis, J. Palczewski</i>	58
Cloud Robotics-Based System for Robot Teleoperation <i>H. Aagela, V. Holmes</i>	62

Session 6: Applications of Emerging Tech	66
AI to Facilitate Legal Analysis in the PESTLE Context	
<i>M. Vallati, A. Grassi</i>	66
Holistic Approach to Energy and Power Management in HPC	
<i>V. Elisseev</i>	69
Collaborative Cloud-based Face recognition approach for Humanoid robots	
<i>H. Aagela, V. Holmes</i>	72
 POSTERS	76
Black-box Tracking system for Drones using LoRa	
<i>H. Aagela, V. Holmes</i>	77
Security Orchestration Automation and Response (SOAR) in High Performance Computing Systems	
<i>T. Al-Jofy, V. Holmes</i>	78
Big Data One Millions Songs Dataset	
<i>C. Arinto</i>	79
High End Compute	
<i>M.K. Bane</i>	80
Indoor Two Way Ranging using mm-Wave for Future Wireless Networks	
<i>A. Farooq, Q. Ahmed, F. Khan and T. Alade</i>	81
Tunable Fault Tolerant Spiking Neural Networks on FPGAs	
<i>A.P. Johnson, J. Liu, A.G. Millard, S. Karim, A.M. Tyrrell, J. Harkin, J. Timmis, L. McDaid and D.M. Halliday</i>	82
RISC-V Implementation	
<i>J. Parkinson</i>	83
Energy harvesting promising sign for advanced 5G wireless communication network and Internet of Things (IoT) based storage devices	
<i>J. Yazdani and V. Thayanathan</i>	84



KEYNOTE LECTURES

DataFlow SuperComputing for Big Data Analytics

Prof. Veljko Milutinovic
Indiana University

Abstract—This presentation analyses the essence of DataFlow SuperComputing, defines its advantages and sheds light on the related programming model. DataFlow computers, compared to ControlFlow computers, offer speedups of 20 to 200 (even 2000 for some loops-intensive applications), power reductions of about 20, and size reductions of also about 20. However, the programming paradigm is different, and has to be mastered. The talk explains the paradigm, using Maxeler as an example, and sheds light on the ongoing research, which, in the case of the speaker, was highly influenced by four different Nobel Laureates: (a) from Richard Feynman it was learned that future computing paradigms will be successful only if the amount of data communications is minimized; (b) from Ilya Prigogine it was learned that the entropy of a computing system would be minimized if spatial and temporal data get decoupled; (c) from Daniel Kahneman it was learned that the system software should offer options related to approximate computing; and (d) from Andre Geim it was learned that the system software should be able to trade between latency and precision.



Prof. Veljko Milutinovic received his PhD from the University of Belgrade, spent about a decade on various faculty positions in the USA (mostly at Purdue University), and was a co-designer of the DARPA's first GaAs RISC microprocessor. Later, for almost 3 decades, he taught and conducted research at the University of Belgrade, in EE, MATH, BA, and PHYS/CHEM. Now he serves as the Chairman of the Board for the Maxeler operation in Belgrade, Serbia. His research is mostly in data-mining algorithms and data-flow computing, with the emphasis on mapping of data analytics algorithms onto fast energy efficient architectures. For 7 of his books, forewords were written by 7 different Nobel Laureates with whom he cooperated on his past industry sponsored projects. He has over 40 IEEE journal papers, over 40 papers in other SCI journals (4 in ACM journals), over 400 Thomson-Reuters citations, and about 4000 Google Scholar citations. Short courses on the subject he delivered so far in a number of universities worldwide: MIT, Harvard, Boston, NEU, Columbia, NYU, Princeton, Temple, Purdue, IU, UIUC, Michigan, EPFL, ETH, Karlsruhe, Heidelberg, Napoli, Salerno, Siena, Pisa, etc. Also at the World Bank in Washington DC, Brookhaven National Laboratory, IBM TJ Watson, Yahoo NY, ABB Zurich, Oracle Zurich, etc.

High-Performance Communication in Machine Learning

Prof. Torsten Hoefer
ETH Zurich

Abstract—One of the main drivers behind the rapid recent advances in machine learning has been the availability of efficient system support. Despite existing progress, scaling compute-intensive machine learning workloads to a large number of compute nodes is still a challenging task. In this talk, we provide an overview of communication aspects in deep learning. We address the communication challenge, by proposing SparCML, a general, scalable communication layer for machine learning applications. SparCML is built on the observation that many distributed machine learning algorithms either have naturally sparse communication patterns, or have updates which can be sparsified in a structured way for improved performance, without loss of convergence or accuracy. To exploit this insight, we analyze, design, and implement a set of communication-efficient protocols for sparse input data, in conjunction with efficient machine learning algorithms which can leverage these primitives. Our communication protocols generalize standard collective operations, by allowing processes to contribute sparse input data vectors, of heterogeneous sizes. Our generic communication layer is enriched with additional features, such as support for non-blocking (asynchronous) operations and support for low-precision data representations. We validate our algorithmic results experimentally on a range of large-scale machine learning applications and target architectures, showing that we can leverage sparsity for order-of-magnitude runtime savings, compared to existing methods and frameworks.



Dr Hoefer is an Associate Professor of Computer Science at ETH Zürich, Switzerland. Before joining ETH, he led the performance modelling and simulation efforts of parallel petascale applications for the NSF-funded Blue Waters project at NCSA/UIUC. He is also a key member of the Message Passing Interface (MPI) Forum where he chairs the “Collective Operations and Topologies” working group. Torsten won best paper awards at the ACM/IEEE Supercomputing Conference SC10, SC13, SC14, EuroMPI’13, HPDC’15, HPDC’16, IPDPS’15, and other conferences. He published numerous peer-reviewed scientific conference and journal articles and authored chapters of the MPI-2.2 and MPI-3.0 standards. He received the Latsis prize of ETH Zurich as well as an ERC starting grant in 2015. His research interests revolve around the central topic of “Performance-centric System Design” and include scalable networks, parallel programming techniques, and performance modelling. Further details can be found on his home page at <https://htor.inf.ethz.ch/>



PEER REVIEWED PAPERS

Session 1: Novel Hardware

Prospects for Low-power Acceleration of HPC Workloads in EuroExa: FPGA Acceleration of a Numerical Weather Forecast Code

Mike Ashworth
School of Computer Science
University of Manchester
Manchester, United Kingdom
mike.ashworth.compsci@manchester.ac.uk

Graham Riley
School of Computer Science
University of Manchester
Manchester, United Kingdom
graham.riley@manchester.ac.uk

Andrew Attwood
School of Computer Science
University of Manchester
Manchester, United Kingdom
andrew.attwood@manchester.ac.uk

John Mawer
School of Computer Science
University of Manchester
Manchester, United Kingdom
john.mawer@manchester.ac.uk

Abstract—The EuroExa project proposes a High-Performance Computing (HPC) architecture which is both scalable to Exascale performance levels and delivers world-leading power efficiency. This is achieved through the use of low-power ARM processors accelerated by closely-coupled FPGA programmable components. In order to demonstrate the efficacy of the design, the EuroExa project includes application porting work across a rich set of applications. One such application is the new weather and climate model, LFRic (named in honour of Lewis Fry Richardson), which is being developed by the UK Met Office and its partners for operational deployment in the middle of the next decade. Much of the run-time of the LFRic model consists of compute intensive operations which are suitable for acceleration using FPGAs. We have selected the Xilinx Vivado toolset including High-Level Synthesis (HLS) which generates IP blocks that can be combined with other standard IP blocks in Vivado Design Studio and a bitstream generated for programming the FPGA. A design using twelve matrix-vector IP blocks achieves 5.34 double precision Gflop/s. We shall discuss the implementation, the performance achieved and the prospects for acceleration of the full LFRic weather model.

Keywords—FPGA, High Level Synthesis, numerical weather forecasting

I. INTRODUCTION

Field Programmable Gate Arrays (FPGA) have attracted the attention of both academic and industry research as a mainstream accelerator for large-scale high-performance computing applications [1], for example in atmospheric simulations for weather forecasting [2]. The EuroExa project proposes a high-performance architecture which is both scalable to Exascale performance levels and delivers world-leading power efficiency. This is achieved through the use of low-power ARM processors accelerated by closely-coupled FPGA programmable components. In order to demonstrate the efficacy of the design, the EuroExa project includes application porting work across a rich set of applications.

One such application is the new weather and climate model, LFRic, which is being developed by the UK Met Office and its partners for operational deployment in the early part of the 2020 decade. The LFRic code is named in honour of Lewis Fry Richardson who in 1922 presented not only a detailed description of numerical weather forecasting but also

insights into how such a forecast could be carried out using parallel computing:

"A myriad computers are at work upon the weather of the part of the map where each sits, but each computer attends to only one equation or part of an equation. The work of each region is coordinated by an official of higher rank." [3]

High quality forecasting of the weather on global, regional and local scales is of great importance to a wide range of human activities. The UK Government funding in 2014 for new HPC systems at the Met Office was supported by a business case which anticipated £2 billion of economic impact for a £100 million investment – a Return on Investment of twenty [4][5]. Greater benefits come from increased accuracy, and exploitation of latest developments in HPC has always been of critical importance to the weather forecasting community.

LFRic is a new atmospheric model, being developed at the Met Office in the United Kingdom, which supports simulations for both weather forecasting and climate simulations. The current operational model at the Met Office, the Unified Model, uses a latitude-longitude grid in which lines of longitude converge at the poles leading to problems in performance and scalability, especially on modern highly parallel HPC systems. In a pre-cursor project between the Met Office, the Natural Environment Research Council and the Science and Technology Facilities Council, called GungHo, a new dynamical core was developed using the cube-sphere grid which covers the globe in a uniform way [6].

The GungHo code has also been developed specifically to maintain performance at high and low resolution and for high and low CPU core counts. A key technology to achieve this is Separation of Concerns, in which the science code is separated from the parallel, performance-related code and the PSyclone code generation tool is used to automatically generate code targeting different computer architectures. The LFRic weather and climate model is based on the GungHo dynamical core with its PSyclone software technology [7].

II. THE MATRIX-VECTOR KERNEL

Much of the run-time of the LFRic model consists of compute intensive operations which are suitable for acceleration using FPGAs, and many of those in the dynamics are based on matrix-vector products, for example in the

Helmholtz solver. We have used the Xilinx Vivado toolset including High-Level Synthesis (HLS) to generate, from standard C code, bitstreams for programming the FPGA.

Vivado HLS generates a block of code for the FPGA (an IP block) from a pure C, C++ or OpenCL functions annotated with HLS pragmas, which supply optimization hints and instructions about the data interface. HLS also produces a synthesis report which contains optimization advice and performance metrics, most importantly the Task Latency.

Using this feedback from HLS it is possible to optimise the code without executing it, achieving a substantial reduction in the reported latency. Objectives of the optimization were

- to achieve streaming of data in and out of the IP block with a target of one 64-bit word per clock cycle;
- to achieve pipelining of the arithmetic operations and overlapping of multiplies with additions to achieve one 64-bit multiplication and one 64-bit addition every cycle; and
- to minimize use of resources on the FPGA.

The following optimizations were carried out:

- loops were swapped to make the index over the vertical levels the innermost loop;
- data arrays were transposed where necessary to ensure that data running over the vertical index were sequential in memory; together with the above this ensures a sequential innermost loop of length 40 elements;
- the HLS UNROLL pragma was applied to the innermost loops; unrolling by hand was also tried but shown to result in no additional benefit;
- the HLS PIPELINE pragma was applied to the outermost loop;
- the code just computes the matrix-vector product, without updating the left-hand side array; the update can be performed on the ARM CPU;
- HLS INTERFACE pragmas were added to define the interfaces for the subprogram arguments; in particular the clauses `num_read_outstanding=8`, `max_read_burst_length=64`, `num_write_outstanding=8`, `max_write_burst_length=64`, were used;
- data read from and written to the subprogram arguments were copied into and copied out from local working arrays using `memcpy`; and
- the input vector is constant for iterations of the outer loop, so it is copied in to its local array once at the start; at each iteration of the loop, slices of the matrix are copied in and columns of the output array are copied out.

The last two optimizations in the list above were particularly important. Without them data reads are not streamed, with each word being read independently as though the block is waiting for one read to complete before starting the next. With the optimizations, data is streaming at one word per cycle. We note in particular the benefit of the use of

`memcpy`; HLS recognises `memcpy` and implements it using “burst mode” [8].

Matrix-vector IP blocks generated by HLS were combined with other standard IP blocks in Vivado Design Studio in order to provide functions in the design for data handling, interface with the ARM CPU, BRAM memories, clock control etc. A bitstream may then be generated for programming the FPGA. The design comprises the following:

- a ZynQ UltraScale+ MPSoC IP block, which provides an interface to the ARM processor, through two master AXI4 High Performance Master Full Power Domain ports (HPM0 FPD and HPM1 FPD);
- a number, `nblocks`, of matrix-vector IP blocks;
- the same number, `nblocks`, of Block Memory Generator blocks to provide BRAM block memory, one memory block per matrix-vector block;
- the same number, `nblocks`, of AXI BRAM Controller blocks to provide an AXI protocol interface for each memory block;
- the same number, `nblocks`, of AXI Crossbar switches to allow both the matrix-vector blocks and the ZynQ to access the BRAMs;
- two AXI Crossbar switches to allow the two master ports on the ZynQ to fan out to (1) the slave ports on the BRAM memory controllers and (2) the slave ports on the matrix-vector blocks;
- an AXI Protocol block which provides conversion between AXI4 and AXI4LITE for the slave ports on the matrix-vector blocks;
- a Clocking Wizard IP block, which provides a custom clock and is used to vary the clock speed provided to the other blocks;
- a Processor Reset System block

An example of this design with four matrix-vector blocks is shown in Fig. 1.

The FPGA is then driven by standard C code on the ARM processor. FPGA memory is opened as a device on the ARM and mapped into user space. An area of memory referred to in Vivado as the “Control Registers” is used to control each IP block.

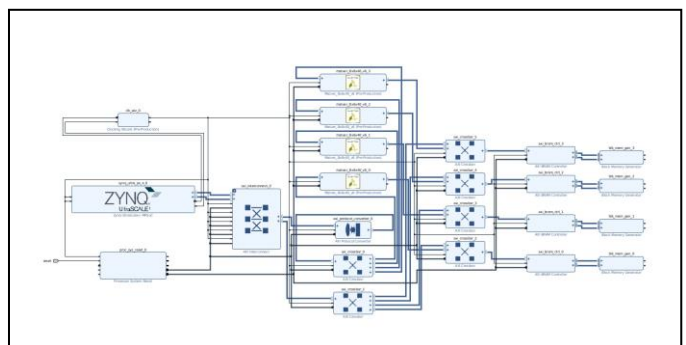


Fig. 1. The full design with four matrix-vector blocks and four BRAM memory blocks implemented in Vivado Design Suite

On the ARM side we load data into BRAM, set the addresses for the three arrays (two input and one output), start the block using the AP_START bit, monitor the AP_IDLE bit to check for completion, and copy output data from BRAM.

A matrix-vector kernel extracted from the LFRic code has been run on a Xilinx UltraScale+ ZCU102 Evaluation Platform [9]. At its heart this board contains a Multi-Processor System-on-Chip (MPSoC) comprising, in addition to other processors, an ARM Cortex A53 quad-core CPU running at 1.2 GHz and a ZynQ UltraScale XCZU9EG-FFVB1156 FPGA. The FPGA contains some 600k logic cells, 2,520 DSP slices and around 3.5 MB of BRAM memory. The ARM CPU is running Ubuntu 14.04.5 and we are using Vivado Design Suite and Vivado HLS, both at version level 2017.4, to generate IP blocks and bitstreams for the FPGA.

Performance of the matrix-vector code was timed, excluding the data transfers between the ARM CPU and the FPGA. The reason for this is that whether data is transferred depends on the context. The major part of this data set, 17MB out of 19MB, consists of the matrices. In any completed port of the LFRic weather model to the FPGA system, the matrices will be generated and used on the FPGA and so will never need to be transferred.

Timings are converted to execution rates in Gflop/s knowing that each 8x6 matrix-vector multiplication requires 2x8x6 flops; two operations, one addition and one multiplication, for each matrix element. The performance for the double precision matrix-vector kernel is shown in Fig. 2.

The speed-up for twelve blocks relative to one block is 10.5x representing a parallel efficiency of 94%. Scaling with clock speed is also good. With twelve matrix-vector blocks, the performance improves from 1.71 double precision GFlop/s at 100 MHz to 5.34 GFlop/s from at 333 MHz, an efficiency of 94%.

We note that matrix-vector multiplication (MVM) is much less computationally efficient than say matrix-matrix multiplication (MXM). For MXM the computational efficiency increases linearly with the matrix size, but for MVM it never increases beyond 0.25 flops/byte.

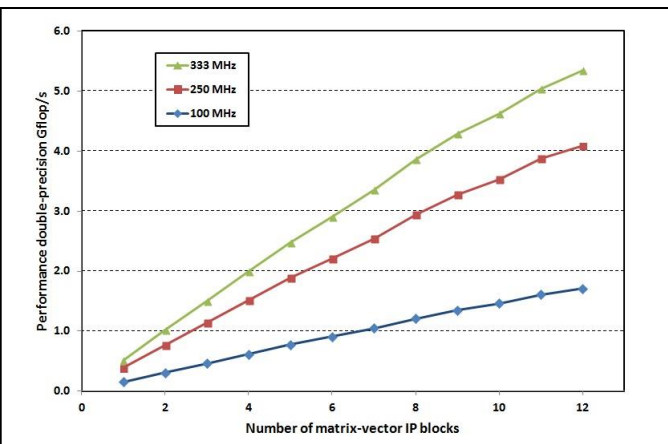


Fig. 2. Performance in GFlop/s of a double precision matrix-vector kernel on the Xilinx ZU9EG FPGA using from one to twelve matrix-vector IP blocks at different clock frequencies

We are limited to twelve IP blocks for this case, not by the resources available on the chip (number of logic gates etc.) but by the complexity of the design. Adding more IP blocks and the required supporting infrastructure causes violation of timing constraints within the design and failure to generate a bitstream.

There are three factors which determine and which limit the achieved performance for our matrix-vector kernel

- the performance in flops/cycle of an individual matrix-vector IP block
- the number of matrix-vector IP blocks in the design
- the clock frequency used to drive the Programmable Logic (PL), principally the matrix-vector blocks but also the associated blocks e.g. the BRAM blocks

The performance of an individual matrix-vector IP block is targeting a peak of 2 flops/cycle but is limited in practice due to overheads associated with data transfers and pipeline start-up costs, to 1.65 flops/cycle (according to the performance estimate of Vivado HLS). The number of IP blocks employed and the clock frequency of the PL are limited by timing constraints. In particular we would like to be able to exploit all the available logic of the FPGA, but find that in practice these timing constraints place a limitation which is more severe than the amount of resources required. In other words, for our application, timing constraints outweigh resource constraints.

An ideal or peak performance figure, P_0 , for this design with twelve blocks running at 333 MHz would be 2 flops/cycle x 12 blocks x 333 MHz = 7.99 GFlop/s. The actual performance, P_a , may be obtained from the ideal performance by two efficiency factors, the single block efficiency, eff_s , and the efficiency with which the blocks are combined in parallel in the design, eff_p , thus:

$$P_a = P_0 \times eff_s \times eff_p = 5.34 \text{ GFlop/s}$$

Assuming the performance estimate from Vivado HLS is realised in practice, gives a value for eff_s of 85%, which implies the parallel efficiency figure for the design, eff_p , is 79%.

There is a relationship, a trade-off, between the number of blocks and the maximum clock speed. For a simple design we can run the code at a higher clock speed, but as the number of matrix-vector blocks and memories increases, the complexity of the design increases and the maximum clock speed decreases. As the clock speed increases and/or the number of matrix-vector blocks increases, the design reaches a point at which timing constraints become important and timing violations cause the implementation to fail.

The maximum clock frequency at which the design operates correctly is shown in Table 1 for different numbers of matrix-vector blocks. The impact on performance is that, although increasing the number of blocks from one to twelve potentially delivers up to a twelve-fold increase in performance, the clock speed is reduced from 450 MHz to 333 MHz, a reduction of 27%, so the 12x potential increase is immediately limited to 8.9x.

TABLE I. MAXIMUM CLOCK FREQUENCY AT WHICH THE DESIGN OPERATES CORRECTLY FOR DIFFERENT NUMBERS OF MATRIX-VECTOR BLOCKS

Number of matrix-vector blocks	Maximum clock frequency (MHz)	Matrix-vector performance (Gflop/s)
1	450	0.688
4	400	2.372
8	333	3.863
12	333	5.339

III. PORTING THE LFRIC WEATHER FORECAST CODE

Having established a methodology using the matrix-vector kernel, we have started to apply these techniques to the LFRic code itself. LFRic can be run in many configurations representing a range of weather and climate scenarios at low-, medium- and high-resolutions. In order to characterise the performance we ran and profiled a baroclinic test case, which has been developed by the Met Office as a part of their performance evaluation procedure [10]. The version of LFRic used for this work implements only parts of the scientific model, namely the dynamics and individual kernels. LFRic dynamics was still under development at the time of this work and important optimisations to its algorithmic performance such as provision of a multigrid preconditioner were not complete. Furthermore, additional science modules such as physics, ocean-coupling and data assimilation will also need to be addressed in the future.

Profiling this test case shows that most of the CPU time is spent in the Helmholtz solver that is used to compute the pressure. Two subroutines account for greater than 50% of the CPU time for this test case. Both of these subroutines spend most of their time performing double-precision matrix-vector multiplication within an outer loop which runs over the vertical levels within the atmosphere.

This Helmholtz kernel, `apply_hx_variable_code`, has been offloaded to the FPGA. It consists of a series of matrix-vector multiplications and ancillary calculations on six input variables. The only difference in our methodology compared with the matrix-vector kernel is that this time we have written the ARM code in standard Fortran rather than C, in order to fit better with the LFRic programming model. This kernel, as with the matrix-vector kernel, has been implemented in a design with twelve IP blocks capable of running independently, thus exploiting spatial parallelism on the FPGA.

A key issue for the LFRic code in exploiting the acceleration potential of the FPGA, as with any accelerator, is reducing the overhead of transferring data between the host CPU and the FPGA. Thus it makes little sense to look at the performance of one small kernel in isolation where that performance will be dominated by data transfer costs. We

need to port a full workflow consisting of a sequence of kernels so that key data structures exist on the FPGA for long periods and ideally are created and used entirely in FPGA memory.

LFRic uses data decomposition across parallel multi-node clusters with halo exchanges between sub-domains carried out using MPI. A part of a workflow may therefore be represented as follows:

```

Kernel 1
Halo exchange for variable x1
Kernel 2
Halo exchange for variable x2
Kernel 3
Halo exchange for variable x3

```

In offloading a whole workflow it is therefore essential to take into account the MPI communications required for halo exchange. Initially the halo exchange will be carried out between host CPUs with data transferred to and from the FPGAs. We note that the amount of data involved for halo exchange is much smaller than the entire data arrays as only boundary data needs to be transferred. As a further optimization step, MPI communications will be available directly from FPGA to FPGA, using communications libraries under development in the EuroExa project.

REFERENCES

- [1] Bacon, D.F., Rabbah, R. and Shukla, S., 2013. FPGA programming for the masses. *Communications of the ACM*, 56(4), pp.56-63
- [2] Gan, L., Fu, H., Yang, C., Luk, W., Xue, W., Mencer, O., Huang, X. and Yang, G., 2014, September. A highly-efficient and green data flow engine for solving euler atmospheric equations. In *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1-6
- [3] L.F. Richardson, *Weather Prediction by Numerical Process*, Cambridge University Press, 1922
- [4] Science and Technology Select Committee (Commons), Session 2010-2012, 13th Report - Science in the Met Office - Volume I, HC 1538, 21 February 2012
- [5] Perrels, A., Th Frei, Francisco Espejo Gil, L. Jamin, and A. Thomalla. "Socio-economic benefits of weather and climate services in Europe." (2013)
- [6] Staniforth, A., Melvin, T. and Wood, N., 2013. Gungho! a new dynamical core for the unified model. In *Proc. ECMWF Workshop on Recent Developments in Numerical Methods for Atmosphere and Ocean Modelling*.
- [7] Adams AV, Ford RW, Hambley M, Hobson JM, Kavcic I, Maynard CM, Melvin T, Mueller EH, Mullerworth S, Porter AR, Rezny M. LFRic: Meeting the challenges of scalability and performance portability in Weather and Climate models. unpublished. arXiv preprint arXiv:1809.07267. 2018 Sep 19
- [8] Xilinx Inc., *Vivado Design Suite User Guide: High-Level Synthesis*, UG902 (v2017.2) June 7, 2017
- [9] Xilinx Inc. 2017. Zynq Ultrascale+ MPSoC. (2017). <https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html>, accessed 7th February 2019
- [10] Christopher Maynard, Met Office, private communication

Direct Communication between Distributed FPGA Resources

Joshua Lant

*APT Group, Department of Computer Science
University of Manchester
Manchester, UK
joshua.lant@manchester.ac.uk*

Javier Navaridas

*APT Group, Department of Computer Science
University of Manchester
Manchester, UK
javier.navaridas@manchester.ac.uk*

Abstract—In recent years the interest in the use of FPGA technology within HPC has been growing rapidly. Burgeoning application domains such as Deep Learning bring computational models which are much more suited to FPGA architectures, being able to take advantage of stream-like processing capabilities over distributed nodes. One of the key obstacles in the uptake of such systems is in the ability of the FPGA resources to act autonomously from the CPU. In this work we examine the progress that has been made in this domain, and compare it with our own work in decoupling the FPGA fabric from CPU resources for reliable network communications.

Index Terms—Interconnects, FPGA, Transport Layer, HPC.

I. INTRODUCTION

FPGAs have shown great promise for the next generation of HPC systems, owing to their strong performance and energy characteristics when faced with data-intensive workloads and stream-like processing. Traditionally shunned due to their difficult programmability and low off-chip memory bandwidth (vs. GPU), maturing ecosystems, larger on-chip memories and integrating advanced memory systems (e.g. HBM [1]) mean architects are now focusing on the potential of FPGAs in HPC/data-centre systems. This is being driven forward by very rapid advances in what is touted to be the FPGA’s “killer app”; Deep Learning [2].

While GPUs remain the go-to accelerator due to their floating point performance and mature ecosystems, increasingly large systems such as those at Microsoft [3] and Amazon’s AWS service are turning towards the FPGA in their data-centers to provide much higher performance-per-watt over GPUs, in a domain where power consumption is becoming an ever growing issue. There are numerous workloads which are shown to outperform the GPU in terms of power consumption. For example, in [4] it is shown that the FPGA can achieve similar performance to GPU on a number of the BLAS¹ routines, but achieving much higher energy efficiency. We believe that as tighter power constraints are placed upon HPC systems, we are likely to see the FPGA feature as a standard component of large heterogeneous HPC systems in the very near future, particularly given their ever maturing programming environments.

This work was funded by the European Union’s Horizon 2020 research and innovation programme under grant agreements No 671553 and 754337.

¹<http://www.netlib.org/blas/>

Another particular advantage that the FPGA can find over GPU is the ability to utilize reduced precision, custom data types and fine grained parallelism to achieve greater performance. As an example, the feed-forward nature of much of the computation involved in Deep Neural Networks (DNNs), and the ability to take advantage of custom data-types and reduced precision [2], for example in binarized neural networks [5], makes the FPGA look like an ideal candidate for accelerating these types of workloads. This is because while GPUs are able to achieve massive parallelism at the thread level, the use of highly customized circuits over distributed FPGA resources offers a clear advantage. It allows for the exploitation of deeply pipelined architectures, naturally providing finer grained parallelism, and by utilizing distributed on-chip memory can offer higher performance for memory-bound operations [6].

II. THE NEED FOR DIRECT COMMUNICATION

A key barrier in the development and exploitation of distributed FPGA resources within the context of the HPC systems is the traditional use of the FPGA as a mere co-processor, loosely coupled to the CPU and network resources attached via PCIe or other equivalent bus-based interconnect (see Figure 1a). Many examples of this sort of acceleration architecture exist; see [7]. Not only does this architectural model exacerbate the limited off-chip memory bandwidth of the FPGA by further distancing the accelerator from the main memory hierarchy of the CPU, but it severely limits the feasibility of data-flow processing between distributed FPGA resources. This is due to the dependence on the CPU for performing network transactions that is a result of the FPGA being seen merely as a peripheral device.

Modern FPGA architectures such as the Xilinx Zynq Ultrascale+, and the Intel Stratix 10, complete with integrated hard-core processors including IOMMUs allow for the configuration seen in Figure 1b, allowing shared memory and cache-coherence between the CPU and FPGA. While this tight coupling allows for lower latency transfers between local accelerator and memory, it does nothing to aid the cumbersome software networking stack in being bypassed. Typical methods such as TCP/IP are required to provide reliable transfer of data between nodes, with costly data copies between network buffers etc.

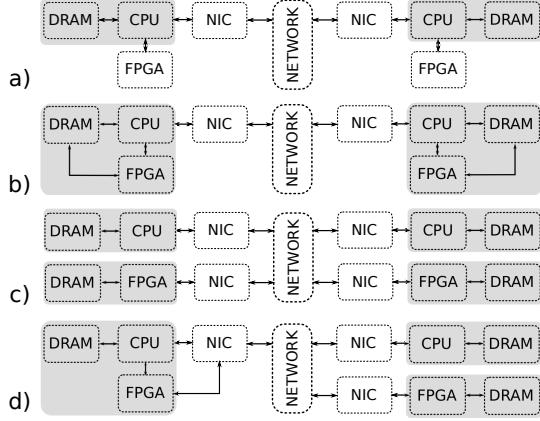


Fig. 1. Possible distributed FPGA System Configurations (top a - bottom d).

Like others have argued [8], we see that the remedy to the issue is to promote the FPGA resources to the status of a full peer within the network, capable of issuing its own reliable transactions into the network, as well as being able to process inbound network traffic directly. In enabling the FPGA to perform RDMA operations directly and offloading traditional networking stacks into hardware, i.e. TCP offloading, this gives rise to the configuration seen in Figure 1c. Here we see that the FPGA is now a peer within the network, and is fully disaggregated from the CPU resources, meaning that FPGA resources can be scaled without increasing the corresponding number of CPUs. However, in this setup the FPGA is unable to exploit a lower latency, shared memory model with other distributed memory spaces; a property which is vital for many workloads and for providing the FPGA better control of the data flow. All this is without mentioning the significant scalability and complexity issues associated with TCP offloading [9].

The solution we have proposed to this issue [10] is to create a NIC which sits in the fabric of the FPGA, and using a custom network protocol is able to utilize a simple geographic addressing scheme [11], where the target node addresses are seen simply as the upper regions of a fully global memory space. The NIC supports hardware primitives for both shared memory and RDMA communications, and the transport layer is fully offloaded into the hardware, bypassing the CPU completely for inter-FPGA data transfers. In doing this we are able to reach the configuration shown in Figure 1d, where the FPGA can act alone, as a fully disaggregated peer on the network, but can also write directly into a shared memory space between the CPU and other resources (local or remote). This opens up the possibility for fine grained acceleration across distributed FPGAs, providing maximum flexibility in the architecture.

III. RELATED WORK

Many attempts have been made by academics and industry alike over the past decade to put FPGA technology to work within a HPC or datacentre context. While much of the

work on datacentre acceleration with FPGAs form the largest such systems, the radically differing requirements in terms of workload, reliability, bounds on jitter etc. mean that many of these solutions are not directly applicable, or must be modified for use in a general purpose HPC system.

Maxwell [12] is a proof of concept for a general purpose FPGA based supercomputer, comprising Intel Xeon CPUs and a total of 64 Xilinx FPGAs configured in a 2d torus, with CPUs and FPGA connected via PCI. This solution obviously suffers the many pitfalls discussed above, with regards to communication being directed through the CPU for network operations beyond a given scale, with only point to point connections between the FPGA for parallel communications. Other systems such as QP [13] use a similar approach, attaching the FPGA as a coprocessor via the PCI bus, forcing the FPGA to communicate through the CPU.

Novo-G# [14] is a system of multiple FPGAs and host CPU within a single server, with many of these servers comprising the entire system. While intra-server FPGAs can communicate directly with one another, inter-server communication must be made through the CPU via PCIe connections, and then via standard Gigabit Ethernet or Infiniband. This limitation means that direct communication between FPGAs is again limited in scale.

Other systems have been designed to use the system bus of the processor in order to couple them far more tightly with the memory system of the CPU. Systems such as the Cray XD-1 [15], which uses AMD's HyperTransport, and the work of Ling et al. at Intel [16] which use the Front-Side-Bus all provide this sort of architectural configuration, allowing for much higher throughput between system memory and accelerator. These architectures however still require the CPU to initiate network transfers, meaning that the use of dataflow style processing over multiple FPGAs is still inhibited by traditional networking techniques.

Recent work at IBM [17] has created a network attached FPGA system, which completely disaggregates the CPU from accelerator resources. This is done in order to allow CPU and FPGA resources to be scaled independently in datacentre environments. They use a hardware offloaded transport layer in order to allow the FPGA to communicate directly to the network. However, this means that all communication must traverse the full networking stack. There is no possibility to perform NUMA type accesses, and therefore tight coupling between the FPGA and CPU memory hierarchies is not possible.

Microsoft have created a system [3] which allows for FPGAs to communicate among themselves at the cloud scale. The FPGAs are situated as a *bump in the wire*, placed between the NIC and a Top-of-Rack switch, and are used for in-network processing. They implement a lightweight transport layer in the fabric of the FPGA which supports TCP/IP, with which the FPGAs can communicate with one another. The main drawback with this architecture in regards to its use in a HPC context is the fact that CPUs still use the traditional software stack for TCP/IP protocols. Only the

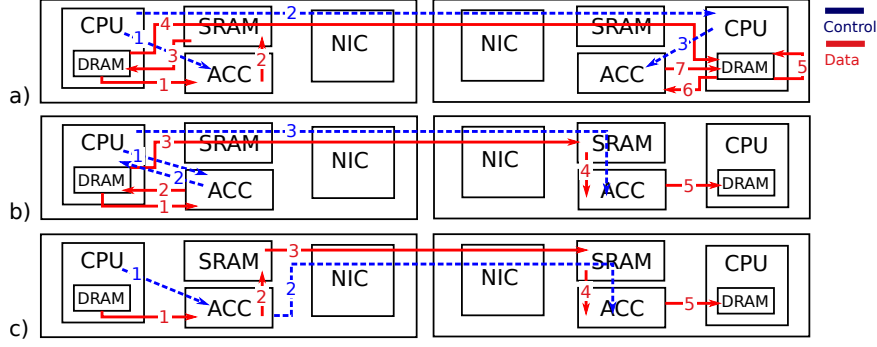


Fig. 2. Control and data paths for a) software based TCP, b) custom software transport, c) our hardware offloaded solution.

FPGAs can communicate using this layer. Another issue is the fact that similarly this mechanism cannot be used for shared memory communications, providing only a method for RDMA transfers.

The poor latency of software implementations of TCP make this an undesirable solution in a HPC environment. While full hardware offloading of the TCP transport layer is possible to alleviate this latency issue, implementations are geared towards financial trading, where scalability is not a primary concern. Attempts have been made to solve this scalability issue, with [9] allowing for ≈ 10000 simultaneous connections. However this implementation is dreadfully wasteful of off-chip RAM resources, requiring around 1.3GB of memory to sustain these connections. Our solution offers a connectionless (data-gram) approach, minimizing the state information required for the transport layer to perform reliable communications.

IV. OUR SOLUTION

Like some of the other works in this field, we regard the hardware offloading of the majority of the networking stack to be imperative for the FPGA to be disaggregated from the CPU resources, to elevate the status of the FPGA from an accelerator to a standalone element within the system. Given the requirements for reliability and the issues caused by packet dropping within a HPC context, this means that the whole transport layer must also be implemented within hardware. The best way we see to do this is to implement the Network Interface within the FPGA fabric. While this means that FPGA resources will be used for networking capabilities when they could be put to use for greater computing power, the burden is not too high within our system. By using a connectionless transport and keeping retransmission data for large transfers within external DRAM (rather than in retransmission buffers in the FPGA) we are able to achieve a modest area overhead in our implementation.

Our NIC contains two segregated data and control paths, one for shared memory transactions, and another for RDMA operations. In this manner the FPGA fabric is capable of writing shared memory operations directly to the NIC, enabling it to submit work directly to remote accelerators. Data can be written to a remote RAM using the DMA engine, and then

shared memory operations can be used to inform the remote accelerator that there is new data to be worked on.

The datapaths are segregated in the NIC because these two communication methods have very different requirements and properties. Due to the fact that the distributed shared-memory operations typically will be formed of small messages, used for control, management, synchronization etc. we store the data in the NIC as it is pushed to the network. In this way retransmission can be performed within the network and the latency of these operations is minimized.

Given that RDMA operations can consist of much larger data transfers, capable of saturating the link bandwidth, it is infeasible to store this data in the NIC for retransmission. Instead a method is sought which enables the NIC to track outstanding DMA operations which are unacknowledged, and enables it to directly rebuild partial transfers in the event of retransmission. The drawback of this method is that double buffering is required in the sender in the event that the user wishes to work on the data while the transfer is in progress, as the user has no knowledge on the status of a partial transfer. It is only informed when a full DMA transfer has completed successfully.

Figure 2 shows the benefits to the communication path when distributed FPGA resources are able to communicate among themselves with no CPU involvement. In Figure 2a we see the data copies and control information which needs to be transferred for a traditional software TCP stack. The TCP protocol requires copies of the data to be made and placed in send buffers for transfer, creating additional latency. In this instance we see that the data needs to be copied from the intermediate SRAM that the accelerator utilizes, back into main memory where the TCP buffers reside. A copy is also required between the receive buffers and user-space memory. Offloading of these buffers to hardware to be closer to the Network Interface would still require an additional copy stage, and would also use excessive memory resources (limiting the scalability of either the window size, or the number of concurrent TCP connections that can be kept track of). As well as this additional data copying, the remote CPU needs to be responsible for notifying the remote accelerator of work, requiring additional control information

to be sent. Therefore additional latency is also seen as the local CPU cannot initialize work directly on the remote FPGA. Our solution alleviates this requirement by writing directly into memory, and using a geographic addressing scheme as a means to locate the destination node.

In Figure 2b we see a solution which utilizes our custom network protocol, but still uses a software based transport layer. In this instance we see that the local CPU is able to submit work directly to the remote accelerator using our direct shared-memory communication mechanism. However, the data to be transferred must still be copied back into DRAM by the accelerator in order for the transport layer to send it. This is because the CPU has no knowledge of the accelerator's work status, and the accelerator has no knowledge of the RDMA transfer status. Depending on how the reliability layer may be implemented, an additional memory copy may also be required to dedicated network send/receive buffers. This solution also results in additional communication requirements between the CPU and accelerator. As shown in the diagram, there is an additional stage of notification between the accelerator and CPU in the critical path, to inform the CPU that it has completed its work. This is not required if the accelerator can initiate network transfers itself.

In Figure 2c we see the data path for our hardware-offloaded solution. In this instance, once the accelerator has completed its work it issues an RDMA operation directly to the NIC, and then writes shared memory operations to the remote accelerator's work buffer, informing it that there is new work to be performed. Once this is completed then the remote accelerator notifies its local CPU that the work has been done and it has new data to process. As is shown, this solution is far more amenable to data-flow type processing, allowing for simpler pipelining of data through the distributed FPGA resources than in a traditional software approach.

While several other solutions [12], [18], allow for this sort of dataflow processing, it is typically performed using only point-to-point links between the FPGAs, limiting the topologies which can be created to tori/mesh topologies with direct nearest neighbour communication only. This in turn limits the scalability of the utilization of distributed FPGA resources to those located within a single rack, and limits the resilience of the system since these topologies mean that a node failure will necessarily affect the neighbouring nodes. When combined with our switch design [11], this solution allows for the creation of modern HPC topologies such as Jellyfish, Dragonfly and Fat-Trees.

V. CONCLUSIONS

In this paper we show that the needs of the network to enable reconfigurable HPC systems to reach their full potential are not being met by today's technologies. We argue that a fully hardware offloaded, connectionless transport layer is the only sensible way to allow for modern HPC communications (MPI and NUMA shared-memory type operations) to be performed directly in inter-FPGA or FPGA-remote CPU configurations. We show how our solution removes several

memory copies from a dataflow scenario, bypassing the CPU for reliable inter-FPGA networking and better facilitating distributed FPGA acceleration.

REFERENCES

- [1] G. Singh, "Xilinx 16nm datacenter device family within-package hbm and ccix interconnect," in *2017 IEEE Hot Chips 29 Symposium (HCS)*, pp. 1–22, IEEE, 2017.
- [2] E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. Ong Gee Hock, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra, *et al.*, "Can fpgas beat gpus in accelerating next-generation deep neural networks?," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 5–14, ACM, 2017.
- [3] A. M. Caulfield, E. S. Chung, A. Putnam, H. Angepat, J. Fowers, M. Haselman, S. Heil, M. Humphrey, P. Kaur, J.-Y. Kim, *et al.*, "A cloud-scale acceleration architecture," in *The 49th Annual IEEE/ACM International Symposium on Microarchitecture*, p. 7, IEEE Press, 2016.
- [4] S. Kestur, J. D. Davis, and O. Williams, "Blas comparison on fpga, cpu and gpu," in *2010 IEEE computer society annual symposium on VLSI*, pp. 288–293, IEEE, 2010.
- [5] E. Nurvitadhi, D. Sheffield, J. Sim, A. Mishra, G. Venkatesh, and D. Marr, "Accelerating binarized neural networks: Comparison of fpga, cpu, gpu, and asic," in *2016 International Conference on Field-Programmable Technology (FPT)*, pp. 77–84, IEEE, 2016.
- [6] G. Lacey, G. W. Taylor, and S. Areibi, "Deep learning on fpgas: Past, present, and future," *arXiv preprint arXiv:1602.04283*, 2016.
- [7] C. Kachris and D. Soudris, "A survey on reconfigurable accelerators for cloud computing," in *2016 26th International conference on field programmable logic and applications (FPL)*, pp. 1–10, IEEE, 2016.
- [8] K. D. Underwood, K. S. Hemmert, and C. D. Ulmer, "From silicon to science: The long road to production reconfigurable supercomputing," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 2, no. 4, p. 26, 2009.
- [9] D. Sidler, G. Alonso, M. Blott, K. Karras, K. Vissers, and R. Carley, "Scalable 10gbps tcp/ip stack architecture for reconfigurable hardware," in *Field-Programmable Custom Computing Machines (FCCM), 2015 IEEE 23rd Annual International Symposium on*, pp. 36–43, IEEE, 2015.
- [10] J. Lant, C. Concatto, A. Attwood, J. A. Pascual, M. Ashworth, J. Navaridas, M. Luján, and J. Goodacre, "Enabling shared memory communication in networks of mpsoes," *Concurrency and Computation: Practice and Experience*, p. e4774.
- [11] C. Concatto, J. A. Pascual, J. Navaridas, J. Lant, A. Attwood, M. Lujan, and J. Goodacre, "A cam-free exascale hpc router for low-energy communications," in *International Conference on Architecture of Computing Systems*, pp. 99–111, Springer, 2018.
- [12] R. Baxter, S. Booth, M. Bull, G. Cawood, J. Perry, M. Parsons, A. Simpson, A. Trew, A. McCormick, G. Smart, *et al.*, "Maxwell-a 64 fpga supercomputer," in *Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, pp. 287–294, IEEE, 2007.
- [13] M. Showerman, J. Enos, A. Pant, V. Kindratenko, C. Steffen, R. Pennington, W.-m. Hwu, *et al.*, "Qp: A heterogeneous multi-accelerator cluster," in *Proc. 10th LCI International Conference on High-Performance Clustered Computing*, 2009.
- [14] A. D. George, M. C. Herbordt, H. Lam, A. G. Lawande, J. Sheng, and C. Yang, "Novo-g#: Large-scale reconfigurable computing with direct and programmable interconnects," in *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–7, IEEE, 2016.
- [15] Cray User Group, "The cray xd1 technical overview," https://cug.org/5-publications/proceedings_attendee_lists/2004CD/S04_Proceedings/pages/Authors/Shan_Amar_Slides.pdf, 2004.
- [16] L. Ling, N. Oliver, C. Bhushan, W. Qigang, A. Chen, S. Wenbo, Y. Zhihong, A. Sheiman, I. McCallum, J. Grecco, *et al.*, "High-performance, energy-efficient platforms using in-socket fpga accelerators," in *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, pp. 261–264, ACM, 2009.
- [17] J. Weerasinghe, R. Polig, F. Abel, and C. Hagleitner, "Network-attached fpgas for data center applications," in *2016 International Conference on Field-Programmable Technology (FPT)*, pp. 36–43, IEEE, 2016.
- [18] A. Putnam, A. M. Caulfield, E. S. Chung, D. Chiou, K. Constantinides, J. Demme, H. Esmailzadeh, J. Fowers, G. P. Gopal, J. Gray, *et al.*, "A reconfigurable fabric for accelerating large-scale datacenter services," *ACM SIGARCH Computer Architecture News*, vol. 42, no. 3, pp. 13–24, 2014.

Event-based computation: Unsupervised elementary motion decomposition

Petrut A. Bogdan

*School of Computer Science
The University of Manchester*

Manchester, UK

petrut.bogdan@manchester.ac.uk

Garibaldi Pineda García

*School of Engineering and Informatics
University of Sussex*

Brighton, UK

g.pineda-garcia@sussex.ac.uk

Simon Davidson

*School of Computer Science
The University of Manchester*

Manchester, UK

simon.davidson@manchester.ac.uk

Michael Hopkins

*School of Computer Science
The University of Manchester*

Manchester, UK

michael.hopkins@manchester.ac.uk

Robert James

*School of Computer Science
The University of Manchester*

Manchester, UK

robert.james@manchester.ac.uk

Steve B. Furber

*School of Computer Science
The University of Manchester*

Manchester, UK

steve.furber@manchester.ac.uk

Abstract—Fast, localised motion detection is crucial for an efficient attention mechanism. We show that modelling a network capable of such motion detection can be performed using spiking neural networks simulated on many-core neuromorphic hardware. Moreover, highly sensitive neurons arise from the presented network architecture through unsupervised self-organisation. We use a synaptic rewiring rule which has been shown to enable the formation and refinement of neural topographic maps. Our extension allows newly formed synapses to be initialised with a delay drawn from a uniform distribution. Repeated exposure to moving bars enables neurons to be sensitised to a preferred direction of movement. Incorporating heterogeneous delays results in more sensitive neural responses. A readout mechanism involving a neuron for each learnt motion is sufficient to establish the input stimulus class.

Index Terms—SpiNNaker, Neuromorphic computing, Spiking Neural Network, structural plasticity, synaptic rewiring, topographic maps

I. INTRODUCTION

Neuromorphic platforms are relatively novel computational systems designed to mimic key aspects of mammalian brain operation: massive parallelism, low energy consumption, fault tolerance and sparsity. These platforms come in multiple flavours ranging from full-custom chip design (mixed analogue-digital [1] and fully digital designs [2]) to using a vast array of off-the-shelf components. SpiNNaker [3] (the full system is pictured in Fig. 1) is a digital many-core neuromorphic platform designed to simulate a vast number of biologically-inspired spiking neurons in real time.

Novel computing architectures require the development of new methodologies and tools to harness their full capabilities.

The design and construction of the SpiNNaker machine was supported by EPSRC (the UK Engineering and Physical Sciences Research Council) under grants EP/D07908X/1 and EP/G015740/1, in collaboration with the universities of Southampton, Cambridge and Sheffield and with industry partners ARM Ltd, Silistix Ltd and Thales. Ongoing development of the software is supported by the EU ICT Flagship Human Brain Project (H2020 785907), in collaboration with many university and industry partners across the EU and beyond.



Fig. 1: The 1 million ARM-core SpiNNaker machine. Capable of simulating on the order of 200 million neurons, with 1,000 synapses each, in real-time.

While using ARM technology for the 18 computational cores present on chip, SpiNNaker has been designed specifically for spiking neural network (SNN) simulations using a purpose-built router and attaching small amounts of fast memory to each individual core.

We build on previous work [4] and present an end-to-end approach to perform elementary motion decomposition using leaky integrate-and-fire neurons and structural and synaptic plasticity [5]. Further, the computational platform which is the basis for these simulations is event-driven [6], including the spiking visual input provided to the network. The biologically inspired sensory processing method presented here is an alternative to traditional frame-based computer vision.

We show that (1) the presented architecture allows for unsupervised learning; that (2) synaptic rewiring enhanced to initialise synapses by drawing from a distribution of delays produces more specialised neurons; and that (3) a pair of readout neurons is sufficient to correctly classify the input based on the target layer's activity using rank-order encoding (first classification neuron to spike wins), rather than spike-rate encoding (classification neuron which fires most in a time period wins).

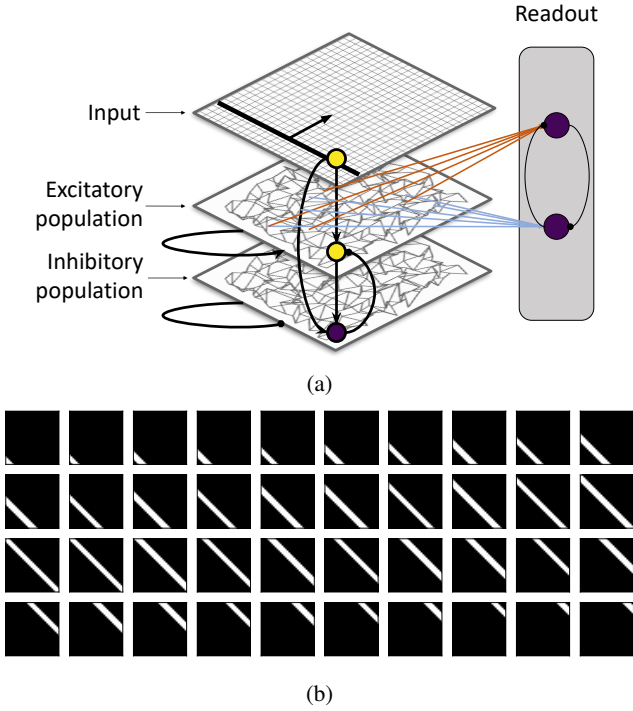


Fig. 2: (a) Network architecture. (b) Example input 45° degree movement represented as its constituent frames (before processing to generate spikes). A new frame is presented every 5 ms and, in total, the presentation of an entire pattern takes 200 ms.

II. METHODS

The SNN architecture (pictured in Fig. 2a) is designed to allow unsupervised learning through self-organisation using synaptic and structural plasticity mechanisms [5]. Neurons in the two target populations are modelled as being positioned at integer locations on a 32×32 grid with periodic boundary conditions. The excitatory population contains neurons which receive sparse excitatory connections from the input layers and from themselves, while projecting to the inhibitory layer and to the readout neurons responsible for the final motion classification decision. The inhibitory population follows a similar structure, but only projects using inhibitory synapses. Very strong inhibition is also present between the readout neurons, implementing a winner-takes-all circuit. The networks are described using the PyNN simulator-independent language for building neuronal network models [7] and the SpiNNaker-specific software package for running PyNN simulations (sPyNNaker [6])¹. The model is simulated in real time on the SpiNNaker many core neuromorphic platform using previously presented neuron and synapse dynamics [5].

The input stimulus consists of bars encoded using spikes representing “ON” and “OFF” pixels (see Fig. 2b for an example before filtering using a previously described technique [8])

¹The data and code used to generate the results presented in this paper are available from doi: 10.17632/wpzvh93vvh.1

as well as a background level of Poisson noise (5 Hz). Each stimulus is presented over a 200 ms time period always moving at a constant speed (200 frames per second). During training the target layers are presented with bars moving in two directions (Eastward or at 0° and Northward or at 90°), but during testing they are presented with moving bars in all directions (randomised over time, in 5 degree increments) – weights and connectivity are fixed during this latter phase. The simulations are initialised with no connections and are trained for around 5 hours, while testing occurs over 20 minutes. As a result of the chosen testing regime, the networks sees over 80 moving bar presentations at each of the 72 angles. This allows us to perform a *pair-wise independent t-test* between the responses at each of the angles in the two cases and establish whether their responses are statistically different. The readout neurons are trained and tested separately from the rest of the network – this process takes on the order of a minute.

Using the structural plasticity mechanism implemented for SpiNNaker, new synapses are formed in two regimes: with heterogeneous, random delays ([1, 15] ms, uniformly drawn) and homogeneous, constant (1 ms) delays; the latter is taken to be the control experiment. Further, according to the structural plasticity mechanism, depressed synapses are more likely to be removed. This optimises the use of the limited synaptic capacity available for each post-synaptic neuron [9]; neurons have a fixed maximum fan-in of 128 synapses with delays which do not change over time.

The direction selectivity index (DSI) will be computed for each neuron after training: $DSI = (R_{pref} - R_{null}) / R_{pref}$, where R_{pref} is the response of a neuron in the preferred direction, and R_{null} is the response in the opposite direction [10]. We compute it for each of the possible directions and establish the preferred direction as that which maximises the DSI after performing a weighted average of neighbouring responses, reducing the influence of noise.

III. RESULTS

The response of the excitatory population in each regime (incorporating heterogeneous delays or not) is plotted for each testing direction (minimum, mean and maximum responses presented in Fig. 3a). The polar plot reveals the firing rate (Hz) of neurons during testing when the input is moving in each of the 72 directions from 0° to 355° in 5° increments in a random order. The network response shows that neurons are responding preferentially to movement, rather than simply to the shape of the input, because the response is asymmetrical – it can differentiate between e.g. a vertical bar moving Eastward and the same vertical bar moving Westward. The *pair-wise independent t-test* is performed to compare the network response in the two regimes (Fig. 3c, red line signifies that $p \geq 0.001$ for that particular angle); the response is higher in one training direction (90°) and less in the other (0°) for the network with heterogeneous delays compared to the control. As such, we proceed by examining individual neurons rather than the average network behaviour. The spatial organisation of neurons and their preferred angle is presented in Fig. 3b, showing that

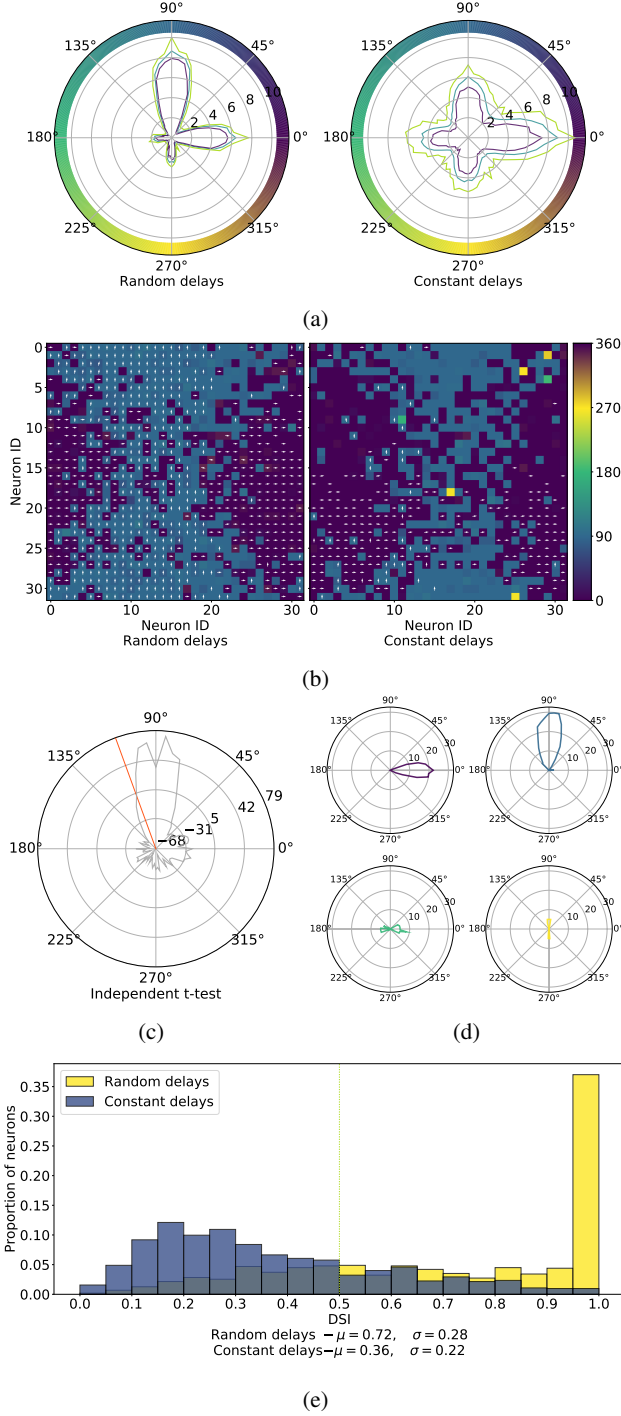


Fig. 3: (a) – minimum, mean and maximum aggregate excitatory population firing response (Hz); (b) – neuron angle preference based on maximum firing rate (the colour) and DSI (the arrow is present if $DSI \geq 0.5$); (c) – *pair-wise independent t-test* comparing the network with heterogeneous delays (on the left in a and b) compared to the control, red lines = insignificant results; (d) – selected individual neuron responses (random delays); (e) – DSI distribution comparison.

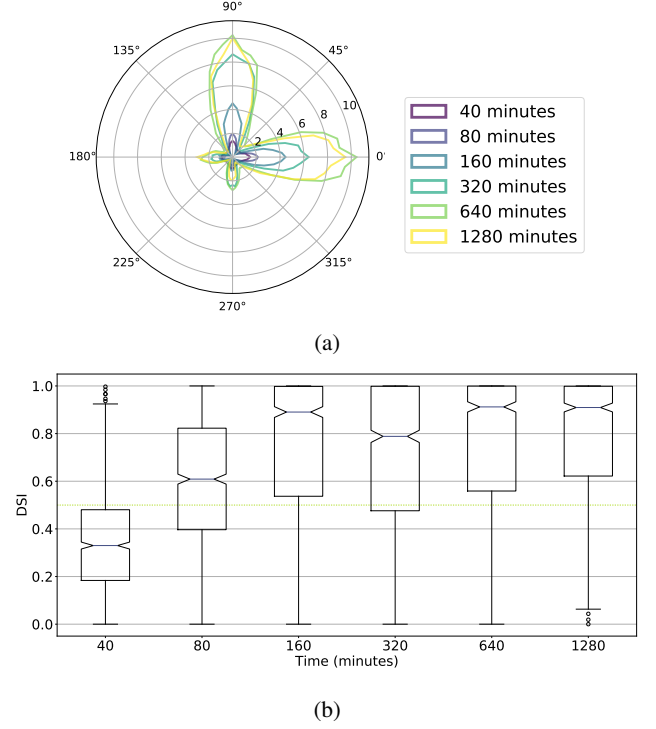


Fig. 4: Network behaviour evolution with longer simulation run times. (a) – average network firing response during inference when trained for ever increasing times; (b) – DSI distribution displayed as a *boxplot* for each simulation in (a). Note: Each data point is a different simulation.

local neural neighbourhoods become sensitised to the same input statistics. There we also look at neurons' maximum responses (encoded by the colour of the cell) in conjunction with the direction which maximises DSI (arrow direction) and $DSI \geq 0.5$ (arrow presence). The direction selectivity index histogram presented in Fig. 3e. compares the two networks; the control network has significantly fewer selective neurons (251 compared to 744) and selectivity is lower on average. Individual responses of our simulated neurons resemble the direction selectivity found in Superior Colliculus [11].

Further, we examine the network behaviour over a wide range of simulations times, ranging from 40 minutes up to 20 hours. Figure 4a shows the evolution of the population-level firing rate and the evolution of the DSI metric (Fig. 4b). The network is thus shown to be stable over long periods of time, rather than showing destructive dynamics.

A readout or classification mechanism relying on 2 mutually inhibitory neurons is sufficient to resolve the two directions presented in the input. Static excitatory connectivity originating from the excitatory layer results in a potential 100% classification accuracy based on rank-order encoding. After 40 seconds, the two neurons have self-organised to respond to one of two input patterns. Figure 5 shows the spiking behaviour of the two neurons in the first 1.8 seconds of training and testing. Spike-timing dependent plasticity (STDP) reduces the

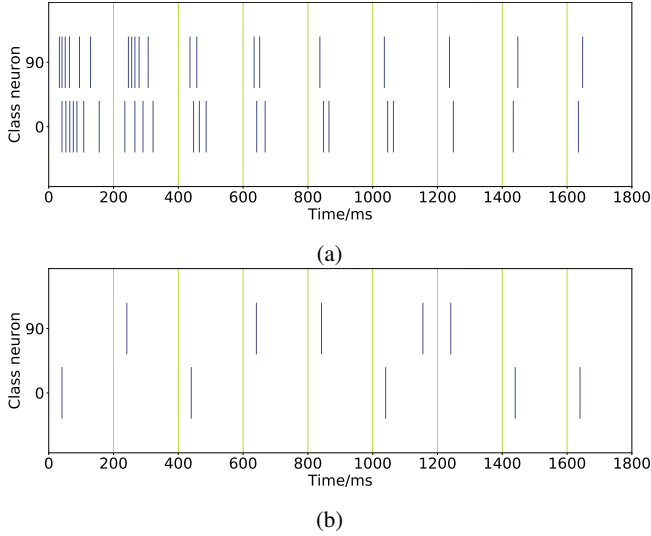


Fig. 5: Initial spiking activity of the two readout neurons during training (a) and testing (b). The full-height vertical bars denote the edges of the pattern presentation time bins (every $t_{stim} = 200$ ms). Neuron class is established post-hoc as the one which maximises classification accuracy.

latency in neural response to the stimuli, making the neurons respond to the stimulus onset, thus making them ideal for classification using rank-order encoding, rather than a winner-takes-all classification based on spike count across a time period [4].

IV. DISCUSSION

We have shown that neurons become sensitised in an unsupervised manner to bars moving in various directions through local learning mechanisms and an interplay between lateral excitation and inhibition. They self-organise their connectivity through synaptic plasticity and rewiring. The rewiring rule with heterogeneous delays selects ideal spatial distributions of synaptic delays driven by STDP. With the current experimental setup, two readout neurons are sufficient for accurate classification of input bar movement direction.

Future work will focus on processing larger and more realistic scenes, as well as handwritten digits. For this, the output layer could be enhanced to produce a correct readout more consistently, e.g through the use of populations of neurons rather than individuals; readout populations would also allow the encoding of the location in the receptive field of the moving bar, rather than its current status as a binary flag of whether a specific movement direction is presented in the input. The readout layer would eventually be replaced entirely using the structures as presented herein to form a more complete visual cortex model. Finally, we will explore further means to control the response of neurons and ensure maximum selectivity for them all.

REFERENCES

- [1] J. Schemmel, L. Kriener, M. Paul, K. Meier, P. Müller, K. Meier, M. Paul, and K. Meier, “An Accelerated Analog Neuromorphic Hardware System Emulating NMDA- and Calcium-Based Non-Linear Dendrites,” *arXiv preprint arXiv*, vol. 1703, no. 07286, pp. 2217–2226, may 2017.
- [2] M. Davies, N. Srinivasa, T.-h. H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-k. K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-h. H. Weng, A. Wild, Y. Yang, H. Wang, I. Labs, and I. Corporation, “Loihi: A Neuromorphic Manycore Processor with On-Chip Learning,” *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.
- [3] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, “Overview of the SpiNNaker system architecture,” *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2454–2467, 2013.
- [4] M. Hopkins, G. Pineda-García, P. A. Bogdan, and S. B. Furber, “Spiking neural networks for computer vision,” *Royal Society Interface Focus*, vol. 8, no. 4, mar 2018.
- [5] P. A. Bogdan, A. G. D. Rowley, O. Rhodes, and S. B. Furber, “Structural plasticity on the SpiNNaker many-core neuromorphic system,” *Frontiers in Neuroscience*, vol. 12, no. 12, pp. 1–20, 2018.
- [6] O. Rhodes, P. A. Bogdan, C. Brenninkmeijer, S. Davidson, D. Fellows, A. Gait, D. R. Lester, M. Mikaitis, L. A. Plana, A. G. D. Rowley, A. B. Stokes, and S. B. Furber, “sPyNNaker : A Software Package for Running PyNN Simulations on SpiNNaker,” *Frontiers in Neuroscience*, vol. 12, no. November, 2018.
- [7] A. P. Davison, P. Andrew, D. Brüderle, J. Eppler, J. Kremkow, E. Müller, and D. Pecevski, “PyNN: a common interface for neuronal network simulators,” *Frontiers in Neuroinformatics*, vol. 2, no. January, pp. 1–10, 2008.
- [8] G. Pineda Garcia, P. Camilleri, Q. Liu, and S. Furber, “pyDVS: An extensible, real-time Dynamic Vision Sensor emulator using off-the-shelf hardware,” in *IEEE Symposium Series on Computational Intelligence, SSICI*, 2016.
- [9] R. George, G. Indiveri, and S. Vassanelli, “Activity Dependent Structural Plasticity in Neuromorphic Systems,” in *Biomedical Circuits and Systems Conference (BioCAS)*. Torino, Italy: IEEE, 2017, pp. 1–4.
- [10] M. Mazurek, M. Kager, and S. D. V. Hooser, “Robust quantification of orientation selectivity and direction selectivity,” *Frontiers in Neural Circuits*, 2014.
- [11] S. Inayat, J. Barchini, H. Chen, L. Feng, X. Liu, J. Cang, X. S. Inayat, J. Barchini, X. H. Chen, L. Feng, X. Liu, and X. J. Cang, “Neurons in the Most Superficial Lamina of the Mouse Superior Colliculus Are Highly Selective for Stimulus Direction,” *Journal of Neuroscience*, vol. 35, no. 20, pp. 7992–8003, 2015.



PEER REVIEWED PAPERS

Session 2: Deep Learning

An efficient way to deal with algorithmically generated data in deep learning

Said Al-Riyami
Department of Computer Science
University of Liverpool
Liverpool, UK
said.alriyami@liverpool.ac.uk

Alexei Lisitsa
Department of Computer Science
University of Liverpool
Liverpool, UK
lisitsa@liverpool.ac.uk

Frans Coenen
Department of Computer Science
University of Liverpool
Liverpool, UK
coenen@liverpool.ac.uk

Abstract—The source of the data used in training deep learning models can vary from application to application. In some cases, the data can be generated on demand algorithmically by a generation process. For example, we can examine mathematical data or data obtained from the simulation of a certain scenario. We can approach the training phase with such data with different strategies. In one approach, the data is generated and stored before it is used in subsequent training. The downside of such an approach is the memory and storage consumption that leads to a limited amount of data that can be efficiently used. In this paper we present an alternative approach in which the data generator is used to generate the data on the fly and send it to the training process. In this paper, we present our implementation of the above approach as a workflow using Keras/TensorFlow framework and Python’s generator and report on the experiments with prime factorisation problem. In such setup, the available resources can be efficiently utilised. CPU is used to generate the data and GPU is used to train the model in parallel.

Index Terms—deep learning, machine learning, datasets

I. INTRODUCTION

Two principle questions on labelled datasets are how to get the data and how to label it. Sometimes the nature of the problem allows us to automate the generation and labelling stages. Conventional workflow for dealing with such datasets adopted in machine learning is to generate and store the labelled data first and then use it for the training. For very large datasets though such an approach can be restrictive due to the limits for available memory and storage.

To alleviate these limitations, we propose an alternative approach, in which the data are generated on the fly by CPU and the training performed by the GPU, in parallel. This approach has several advantages: (1) it consumes less amount of memory and we don’t need to store the data, (2) it can go for unlimited amount of time and generate as much data as we want, and (3) each training epoch have unique data that may help to produce a better model.

To demonstrate the advantages of the approach we use the computational problem of prime factorisation as a case study. In the end, we discuss possible limitations of the proposed workflow.

II. ALGORITHMICALLY GENERATED DATA

The source of the datasets used in supervised machine learning can vary depending on the problem in hand. We consider the following general categories of data based on how they have been generated and labelled:

(1) Non-algorithmically generated data: This is the most common type of data where the data are collected from the environment and labelled with human involvement. For example, images that have been captured and labelled by a human.

(2) Semi-algorithmically generated data: This data where one stage of the process handled by the algorithm. The data might be generated by a human, then labelled by an algorithm, or vice versa. For example, the use of signature-based intrusion detection system to label the computer network traffic.

(3) Algorithmically generated data: any data can be generated and labelled by using an algorithm. Such data can be found in simulation or generated by a mathematical experiment. For example, one can use machine learning to try to factorise the numbers.

Our work is only focused on the third type and we will take the prime factorisation problem as a case study.

III. TRAINING THE MODEL

By the convention, we can approach the problem by just generating the dataset and the labels. Then, use those data to train the model. This might be a suitable approach for the first and second types of data, but it has its limitation for the third type:

- There is a limit of how much data can be generated, because of the amount of memory required.
- It may be unknown in advance how much data will be sufficient for solving the problem at hand. If the amount of data generated is less than optimal, the learned model might be susceptible to overfitting or underfitting.

To deal with these issues we propose an alternative approach. Instead of separating the two phases of generating the model (Getting the data and train the model), we can run both phases in parallel. The part of the application will keep generating the data for as much as needed and another part for training the model. By a typical implementation, CPU will be

used for the data generation and GPU for training the model. This approach has several advantages :

- It requires far less memory than another approach;
- The data can be generated on demand and it can go forever;
- The learning process may result in a better generated model. For each training epoch, the data might be unique, subject to the properties of the data generation algorithm. This may result in a model trained with more data and not prone to overfitting.

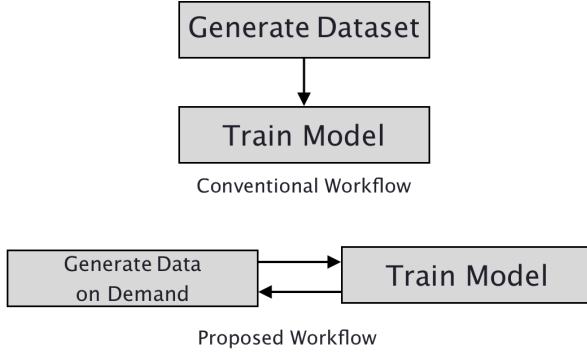


Fig. 1. Difference Between Two Methods

IV. IMPLEMENTATION

To implement the part of generating the data, we will take advantage of Python Generator functions [1]. Python Generator is a function that behaves like an iterator and returns an item only when is called. The generator function uses the keyword ‘yield’ instead of ‘return’. Here an example of a generator to return even numbers:

```

def even_gen():
    n = 0
    while True:
        yield n
        n += 2

```

Code 1. Python Generator Function Example

For our deep learning framework, there are several options. We use Keras Framework for its simplicity [3]. After defining the architecture in Keras, you have 3 option to fit the model (fit(), train_on_batch(), fit_generator()). The common function for training is fit(), which requires the whole data to be available in the memory. For our purpose we will use fit_generator() [4]. It allows us to provide each batch of data from a generator in its turn. The common use of this function is for data augmentation on images. This might includes horizontal and vertical flips, rotation, or adding random noise [2]. We can utilise the same function for our problem. Here is a sample code of the function :

```

model.fit_generator(
    train_generator,
    steps_per_epoch=2000,

```

```

epochs=50,
validation_data=val_generator,
validation_steps=800)

```

Code 2. Keras fit_generator() Example

V. CASE STUDY: PRIME FACTORISATION

One of the most famous computational problems in mathematics is *prime factorisation* [6]. Its complexity underpins the security of well-known public key RSA encryption algorithm [7]. When we have two large prime numbers, it is computationally easy to multiply them, but difficult (e.g. unknown to be possible in PTIME) to factorise the result back. This is a good problem to demonstrate the use of Algorithmically Generated Data.¹ So we consider supervised machine learning problem, in which the input for the model is the product (n) and the label/target are the prime numbers (p and q) such that $n = p \times q$. We provide the input to the model as the binary representation of the value n . The trained model objective is to predict the binary representations of both p and q . The size of each prime number is 512 bits.

The workstation that we used is AMD Ryzen Threadripper 1920X CPU (12 cores and 24 threads) with NVIDIA TITAN V GPU and 32 GB of RAM.

The first workflow to examine, when we separate the generation and training. The dataset size is 5 million instances. For such dataset, we could not generate in our workstation, since it requires larger memory. For that, we used Amazon Cloud Services (AWS) instance that has 192 GB RAM and takes around 2 hours to create. The process here includes the conversion from NumPy array [5] to Panda dataframe [8].

To store the dataframe as CSV file, it takes 1 hour and 57 minutes and uses 56.3 GB. We could not read this file back in our workstation. So, we used a better format called Feather Format [10]. Feather format store the same data with 1.19 GB and the reading/writing takes less than a second.

When we use this data for the training it takes 11.3 GB of RAM (The size of the model is 134,284,800 of total parameters). The model architecture is based on convolutional neural network [11].

By comparison, we need only 1.8 GB of RAM memory when we use our method with the same model size (134,284,800 of total parameters). The usage of the memory is the same for 5 million input or 50 millions.

We can assume that 5 million input require ≈ 10 GB of RAM by using the first method. For 100 million instances, we will need ≈ 200 GB of RAM, which is not available for most typical computer. While the use of the proposed method allows train the model with such big data.

Here is a sample code for the our data generator:

```

def primes_generator(batch_size=256):
    prime_size = 512

```

¹After we started this research we have been positively encouraged to continue by the prediction made for the year 2019 by R. Lipton and K W Regan in the well-known blog in Computer Science: “Deep learning methods will be found able to solve integer factoring. This will place current cryptography is trouble.” Jan 6, 2019, <https://rjlipton.wordpress.com>


```

while True:
    X = []
    y = []
    for i in range(batch_size):
        primes = [number.getPrime(prime_size)
                  for x in range(2)]
        p = min(primes)
        q = max(primes)
        n = format(p*q, 'b').zfill(prime_size*2)
        p_bin = format(p, 'b').zfill(prime_size)
        q_bin = format(q, 'b').zfill(prime_size)
        X.append(list(map(bool,
                           [int(d) for d in str(n)] )))
        y.append(list(map(bool,
                           [int(d) for d in str(p_bin)])) +
                  list(map(bool, [int(d) for d in str(q_bin)])))
    yield ( np.array(X) , np.array(y) )

```

Code 3. Data Generator

And here is a sample for fitting the model:

```

model.fit_generator(
    primes_generator(batch_size=256),
    steps_per_epoch=10000,
    validation_steps=20,
    validation_data=primes_generator(batch_size=256),
    epochs=1000, use_multiprocessing=True, workers=22 )

```

Code 4. Keras fit_generator() Method

When we fit the model, multiple threads in the CPU will run to generate the data. This can be specified by *use_multiprocessing=True* and *workers=number of threads*.

The output of this experiment is not the focus of the paper, but it confirms the difficulty of the problem. So far, no one could factorise a 1024-bit number (each factor 512-bit) with any method [9]. The accuracy that we got with different deep learning models is 50.35%, which far from factoring a 1024-bit number. But the workflow proposed here, makes experimenting with such problem easier.

VI. LIMITATIONS

There might be two limitations for our setup approach :

- The number of CPU threads. If the processor has small number of threads supported, this will effect the performance of the training. For our experiment the processor used support up to 24 threads. This makes the training goes smoother.
- The time needed to execute the generator function. In our case study, when we increase the size of the prime number, the time of the execution increases.

VII. CONCLUSION

While the problems of algorithmically generated data are less common compared with other types, it should be dealt with in a different way. In this work, we compare the conventional way to deal with the problem and compare it with our method. We demonstrate both methods with a case study. We showed how we can train a model with very large data that doesn't fit in a memory.

ACKNOWLEDGEMENT

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research.

REFERENCES

- [1] Python.org, "Python Generator functions documentation". [Online]. Available: <https://wiki.python.org/moin/Generators>. [Accessed: 6- March- 2019].
- [2] Wang, Jason, and Luis Perez. "The effectiveness of data augmentation in image classification using deep learning." Convolutional Neural Networks Vis. Recognit (2017).
- [3] Keras Documentation, "Keras Framework". [Online]. Available: <https://keras.io/>. [Accessed: 6- March- 2019].
- [4] Keras Documentation, "Keras fit_generator()". [Online]. Available: https://keras.io/models/model/#fit_generator. [Accessed: 6- March- 2019].
- [5] NumPy Array, "numpy.array". [Online]. Available: <https://docs.scipy.org/doc/numpy/reference/generated/numpy.array.html>. [Accessed: 6- March- 2019].
- [6] Wikipedia, "Integer factorization". [Online]. Available: https://en.wikipedia.org/wiki/Integer_factorization. [Accessed: 6- March- 2019].
- [7] Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." Communications of the ACM 21.2 (1978): 120-126.
- [8] Pandas Datafram, "pandas.DataFrame". [Online]. Available: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>. [Accessed: 6- March- 2019].
- [9] Wikipedia, "RSA Factoring Challenge". [Online]. Available: https://en.wikipedia.org/wiki/RSA_Factoring_Challenge. [Accessed: 6- March- 2019].
- [10] Github, "Feather Format". [Online]. Available: <https://github.com/wesm/feather>. [Accessed: 6- March- 2019].
- [11] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

SafeChat System with Natural Language Processing and Deep Neural Networks

Michael Seedall
School of Computing and Engineering
University of Huddersfield
Huddersfield, UK
michael.seedall@blackburn.ac.uk

Kate MacFarlane
Faculty of Technology
University of Sunderland
Sunderland, UK
kate.macfarlane@sunderland.ac.uk

Violeta Holmes
School of Computing and Engineering
University of Huddersfield
Huddersfield, UK
v.holmes@hud.ac.uk

Abstract—The internet plays an ever-increasing part in the day-to-day lives of many people. Ubiquitous computing has given rise to sophisticated, streamlined and faster connections across a range of devices. Mobile smart phones are in the hands of children as young as five years old, and whilst this allows them to interact with educational applications and the wealth of information available on-line, it can put them in danger.

There has been a consistent stream of stories involving children and adolescents being at risk because of unsafe on-line behaviour. Predators can prey on the vulnerable, by pretending to be a peer and convincing them, by charm or threats, to compromise their safety. Governments across the globe have initiatives to combat this threat, there are working groups and police task forces in place to respond to both the growing number, and impact of these incidents on children, young people, families and communities. In order to monitor on-line conversation and identify different levels of threats, the SafeChat system was designed and implemented using an ontology-based system and Natural Language Processing (NLP) techniques.

Keywords—artificial intelligence, online safety, natural language processing, deep neural networks, autonomous systems, internet security.

I. INTRODUCTION

Global governmental efforts to address the issue of child safety in an online setting continue. The Internet Taskforce on Child Protection was established in the United Kingdom in March 2001. The task force went on to release a comprehensive set of guidelines for safe practice on the internet aimed at parents and children in 2010. Whilst this was well publicised at the time, it failed to address incidents of children compromising their safety.

To check the engagement with government guidance we have carried out several surveys at the outset of our project, in 2007, research was carried out amongst 437 school children, and 37 of children surveyed said that they had arranged to meet someone they had met online [1, 6]. Subsequently, from December 2015 to March 2016 a focus group of 29 parents were asked to complete an on-line survey into online access, supervision, application usage and privacy for their children.

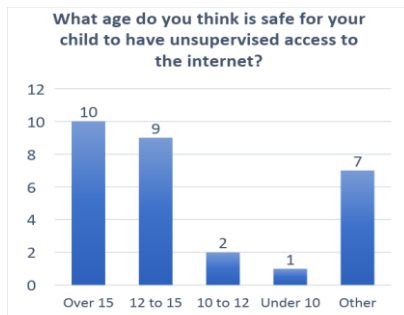


Fig. 1. Results of parent questionnaire (unsupervised access)

Whilst most parents stated that they did worry about their child's safety in an online setting, as seen in figure 1, they went on to confirm that they would let their children access applications and the internet unsupervised once they reached a certain age.

The UK Government Department for Education (DFE) outlines in their 2017 guidance on child sexual exploitation that "Child sexual exploitation is a crime with devastating and long-lasting consequences for its victims and their families". In 2016/17 there had been increases in police recorded child sexual offences and indecent image offences across the UK [3]. Office of Communications (Ofcom) found that one in five 8 to 11-year old's and seven in ten 12 to 15-year old's have a social media profile. The same study also observed that in the age group of 5 to 15-year old's surveyed, 48% of children owned or used a smartphone device [3].

Whilst the number of child grooming and child online grooming cases have increased year on year [4], it could be argued that there is some correlation between the number of crimes against children online versus the continued uptake and ownership of digital devices enabling a growing online child presence.

In response to the growing trend of online child sexual exploitation the UK government introduced new legislation which brought into force section 67 of the Serious Crime Act 2015 in April of 2017. The legislation states that "It is now a criminal offence for anyone aged 18 or over to intentionally communicate with a child under 16, where the person acts for a sexual purpose and the communication is sexual or intended to elicit a sexual response. The offence applies to online and offline communication, including social media, e-mail, texts, letters." [5].

The United Nations published a revised *Convention on the Rights of the Child* [6], Article 16 defines a child's right to privacy and article 17 stipulates that children must have access to information from mass media. Governments are charged with protecting children from sexual exploitation and abduction in articles 34 and 35 respectively.

This paper presents the latest work on the SafeChat system. Recognising the additional overheads of an ontology based multi-agent system [6], coupled with the latest advances in natural language processing and machine learning techniques, current efforts focus on developing a solution using deep neural networks to recognise predator activities and identify risk behaviours to enable real time autonomous intervention in online communication mediums.

The rest of this paper is organised as follows: Section 2 outlines the latest developments in NLP frameworks, section 3 details data gathering and preparation from a variety of sources, which can be used to gather information on behaviours of both victim and perpetrators of online abuse;

Section 4 will present an analysis of initial findings of the data analysis using the latest language processing techniques and tool kits; and finally, Section 5 will present conclusions and discuss future directions for this work.

II. NATURAL LANGUAGE PROCESSING FRAMEWORKS

Natural Language Processing (NLP) is a set of techniques and algorithms that use computers for analyzing natural human language. NLP can be used to solve a variety of problems. Some of the goals of NLP are analysis of (free) text, knowledge and abstract concept extraction from textual data (e.g. text understanding), generative models (e.g. chat bots, virtual assistants, etc.), similarity and classification of words and paragraphs, and sentiment analysis.

Early NLP systems used rules manually designed by domain experts. As the field advanced, the use of machine learning enabled the application of more powerful models that took advantage of ever-growing amounts of data. Today we are taking advantage of Deep Learning and the immense computational power of GPUs and TPUs to tackle ever more complex NLP tasks. Many different deep models have been used since their initial inception in 2000; Deep Learning (DL):

- learns from the data,
- enables more complex reasoning and unsupervised learning,
- learns multiple levels of representation

Word embeddings is using word represented by means of its neighbors.

- Word2Vec is group of efficient predictive models (input, projection and output layers)
- Skip-Gram model and Continuous Bag of Words (BoW) model.

Convolutional Neural Networks (CNN's) can be used for feature extraction of the textual data. In their paper, Shin et al [7] recognize that CNN have given state of the art performance completing sentence classification tasks. They go on to say that this is mainly due to the CNN's ability to extract local features from the data by employing convolution. Recurrent Neural Networks (RNN's) are used for time-series modelling, requiring the 'short memory' of the past, whilst Long Short-Term Memory (LSTM) networks are an extension to RNN that encapsulate long-term memory.

Often the programming language of choice for machine learning is Python. Some popular frameworks being used in NLP solutions are; Torch [8], SpaCy [9], TensorFlow [10], and Caffe2 [11].

- Torch is a scientific computing framework with wide support for machine learning algorithms that puts GPUs first.
- SpaCy is considered to be the fastest NLP (and NLP only) framework. It comes with a lot of pre-trained models to solve many problems straight out of the box.
- TensorFlow is an open-source distributed numerical computational framework released by Google, supporting efficient NLP computations on CPUs and GPUs.

- CAFFE (Convolutional Architecture for Fast Feature Embedding) is a deep learning framework that supports GPU- and CPU-based acceleration computational kernel libraries such as NVIDIA cuDNN and Intel MKL.

One of the most important issues that data scientists encounter is how to represent their data to an algorithm. This is especially relevant in NLP where inputs often differ in lengths, taking the form of sentences or even entire documents. Regardless of input length it is important to develop a representation that can capture similar themes and/or uses of domain-specific terms and vocabulary.

III. DATA PREPARATION

In order to analyse and classify on-line conversations and identify potential predatory attempt, we have acquired over 30,000 lines of predator data which has the potential to extend to over 800,000 lines of discourse once all of the predatory data has been parsed and imported. The typical raw data format is shown in Figure 2. It has to be pre-processed using parsing, to identify component parts including adding the Case Number and inserting a flag in the data to identify the predator/victim. An example of the parsed data is shown in Figure 3.

tblDataIn
RawData
jtwant2play (02/04/07 7:25:28 PM): hi
shelly_belly_93 (02/04/07 7:26:01 PM): hi

Fig. 2. Predatory data

Other digital discourse has been acquired via Twitter (104m lines), Reddit (491m lines) and the Westbury Chat Corpus (180m lines). These other sources of discourse will aid in the identification of general chat behaviour, typical acronyms/types of interaction used in digital discourse and will also aid in testing of predatory behaviour detection when predatory discourse is embedded within a general chat corpus.

From the predator discourse data, we will identify the number of questions being posed by the predator and the victim, then compare this to a comparative sized dataset from the other discourse sources / corpus. We will analyse the data to find the typical linguistics, grammar and phrases that would indicate a question being asked and whether this would aid in the detection and if such questioning prevails throughout all stages of the discourse. In order to grasp the various types of word(s) outside of a standard discourse we will use Apache Spark cluster to perform a word count on the various data sources to aid in the accuracy of question detection and possibly some typical/key indicators to consider when training the intended solution.

tblDataIn						
User Name	ChatText	CaseNumber	Type	Date	Time	CountOfDays
raidersdawg5	u never called me that night	#C2000623	P	24/09/2010	19:50:52	6
danc1njazz	yea i no im sorry	#C2000623	V	24/09/2010	19:51:29	6

Fig. 3. Parsed predatory data

Our initial findings in predator data analysis showed a distinct bias towards the predator interrogation of the victim. A comparison between a predator conversation (1800 lines)

and general chat corpus conversation (1800 lines) displayed the predator conversation had a 6% higher count of question type discourse. However, as further predator data was acquired there was a shift toward the victim interrogation of the predator. Typical lines of questioning can be the victim seeking reassurance from the predator about a type of sexual act or activity.

The predator data analysis has also revealed that some of the predators very quickly suggest a migration to different digital platform / medium to continue their discourse ie: exchange telephone numbers and text message or move from one chat platform to another, with the intention of avoiding detection or looking to increase their interaction with the victim to facilitate an easier exchange of Sexually Explicit Images / Video.

IV. INITIAL ANALYSIS

Once the data is prepared it can then be processed using tools and techniques for feature extraction, classification and analysis. The Natural Language Tool Kit (NLTK) [12] is a fully developed platform that allows the interpretation, analysis and modelling of human language data in a Python programming environment. Given the explicit nature of the collected data, we will use other examples of online discourse to illustrate the way we intend to work with the data.

```
In [2]: from nltk.book import *

*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

Fig. 4. NLTK corpus import

Using NLTK, we can store the data in logical ways and then sort them into a corpus that can be processed as a whole, or in part, depending on the results of initial testing. For example, figure 4 shows imported test corpuses which contains a Chat corpus (text5) and a Personals corpus (text8).

```
In [8]: text5.concordance("meet")

Displaying 5 of 5 matches:
ly back lol JOIN lmao U16 pleased to meet you , hope you guess my name howdy U
lol Lets make babiess !!! ok nice to meet you U64 U107 !!!! PART JOIN (((((((
in bout willis " PART hi U30 nice to meet you lol ... U18 ahhh . U20 ! <<<<, s
ighs happily . U3 did you physically meet neysa ? a bag full o beans girl take
If you 're single , you 're going to meet the person of your dreams . If your

In [9]: text5.similar("meet")

tell me you and what take have let get see shut put be pick check ask
stop help use write
```

Fig. 5. Example of the *concordance* function in NLTK

Once the corpus has been imported there are a number of functions that we can run on the data to quickly establish initial patterns in the discourse. These are:

- **Concordance:** this function lists each instance of a word in the text and displays a list of sentences where it is present, see figure 5.
- **Similar:** is a particularly useful function that lists words used in a similar way to others, which could be key to finding patterns in discourse where someone is trying to avoid detection. It is also useful to see how one user uses language components compared to another user.
- **Collocations:** as seen in figure 4, this function detects the habitual juxtaposition of a particular word with another word (or words) with any regular frequency
- **Lexical Dispersion Plots:** these are a graphical representation of words or lists of words as they appear in the whole corpus, see figure 7.

```
In [23]: text5.collocations()

wanna chat; PART JOIN; MODE #14-19teens; JOIN PART; PART PART;
cute.-ass MP3; MP3 player; JOIN JOIN; times . . .; ACTION watches; guys
wanna; song lasts; last night; ACTION sits; -...)...- S.M.R.; Lime
Player; Player 12%; dont know; lez gurls; long time

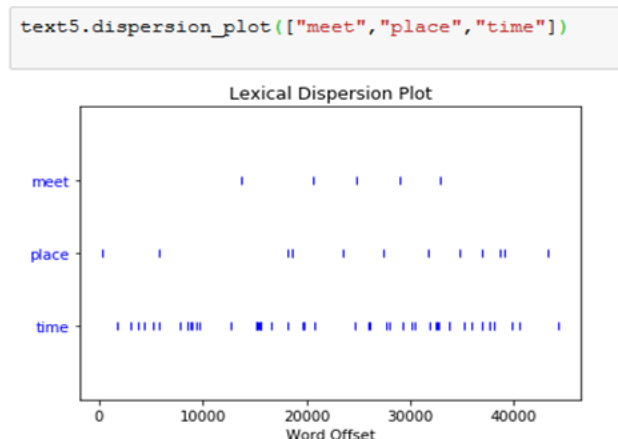
In [25]: text8.collocations()

would like; medium build; social drinker; quiet nights; non smoker;
long term; age open; Would like; easy going; financially secure; fun
times; similar interests; Age open; weekends away; poss rship; well
presented; never married; single mum; permanent relationship; slim
build
```

Fig.6. Example of the *collocations* function in NLTK

While these functions alone do not reveal rich information about the nature of the discourse, they do help to create a picture of the nature and sentiment of some of the data. Used in combination they help to build a clearer picture for possible feature extraction and classification.

Fig.7. Example of the *lexical dispersion* function in NLTK



V. CONCLUSION AND FUTURE WORK

Processing the data effectively is perhaps the most important factor of success in training intelligent systems. Initial findings are promising, and the next steps will be focusing on classifying and testing the data using neural networks.

Convolutional Neural Networks (CNN's) can be used for feature extraction of the textual data. Recurrent Neural Networks (RNN) have been used to good effect when preservation of context is an important factor. In the case of

grooming, context is key, so performance will be measured using both CNN and RNN, and the better system will be adopted to address other online threats, such as, cyber bullying, radicalization and fraud.

Further work will include the development and testing of natural discourse, through laboratory simulations. Multiple chat scenarios can then be tested in real time across bespoke simulated networks to test speed of response, network load and overhead wait times. This will dictate measures needed at a SafeChat system level to secure and maintain transparency of use.

The way humans interact with computers is ever changing and any long-term solutions must take these changes into consideration, potential expansions must include the development of a similar system to work with voice recognition systems. Image and video recognition will also require a similar solution, developing transparent systems to provide protection across these applications areas will present a serious challenge. Combining these systems will facilitate creation of a multi-facet tool for monitoring and detection of potentially predatory behaviour in on-line conversations.

REFERENCES

- [1] MacFarlane, K., and Holmes, V. (2009) Agent-Mediated Information Exchange: Child Safety On-line. In: 2009 International Conference on Management and Service Science. IEEE, pp. 1-5
- [2] Ofcom, (2016) Children and Parents: Media Use and Attitudes Report. United Kingdom
- [3] Bentley, H. et al (2017) How safe are our children? The most comprehensive overview of child protection in the UK 2017. London: NSPCC
- [4] Gov.uk. (2017). New crackdown on child groomers comes into force - GOV.UK. [online] Available at: <https://www.gov.uk/government/news/new-crackdown-on-child-groomers-comes-into-force> [Accessed 15 Dec. 2017].
- [5] UNICEF (2016) What is the UNCRC? | children's rights | UNICEF UK. Available at: <http://www.unicef.org.uk/UNICEFs-Work/UN-Convention/> (Accessed: 12 May 2016)
- [6] MacFarlane, K., and Holmes, V. (2017) Multi-agent System for Safeguarding Children Online, In: Lecture Notes in Networks and Systems, Springer International, ISSN 2367-3370, Volume 16, pp. 228-2
- [7] J. Shin, Y. Kim, S. Yoon and K. Jung, (2018) Contextual-CNN: A Novel Architecture Capturing Unified Meaning for Sentence Classification, *8 IEEE International Conference on Big Data and*
- [8] Torch.ch. (2019). *Torch / Scientific computing for LuaJIT.* [online] Available at: <http://torch.ch/> [Accessed 10 Mar. 2019].
- [9] Anon, (2019). *spaCy · Industrial-strength Natural Language Processing in Python.* [online] Available at: <https://spacy.io/> [Accessed 10 Mar. 2019].
- [10] TensorFlow. (2019). *TensorFlow.* [online] Available at: <https://www.tensorflow.org/> [Accessed 10 Mar. 2019].
- [11] Facebook Research. (2019). *Caffe2 - Facebook Research.* [online] Available at: <https://research.fb.com/downloads/caffe2/> [Accessed 10 Mar. 2019]
- [12] Nltk.org. (2019). *Natural Language Toolkit — NLTK 3.4 documentation.* [online] Available at: <https://www.nltk.org/> [Accessed 18 Mar. 2019].

Comparison of Deep Neural Network approach in text and image classification using CPU and GPU systems

1st Abdul Qurashi

University of Huddersfield

School of Computing and Engineering

United Kingdom

abdul.qurashi@hud.ac.uk

2nd Violeta Holmes

University of Huddersfield

School of Computing and Engineering

United Kingdom

v.holmes@hud.ac.uk

Abstract—Deep Neural Networks (DNNs) and computational industry have the potential to change the way we reason about the environment, as evident from many applications in computer vision, robotics, automotive and medical industry. The advancements made in recent years are due to the developments in hardware technology in High-Performance Computing (HPC), Central Processing Unit (CPU), Graphical Processing Unit (GPU) and Deep Learning frameworks.

DNNs are now able to exceed human accuracy in many domains. The accuracy of DNNs comes at the cost of high computational complexity. When benchmarking DNN systems, the key metrics to consider are accuracy, energy consumption and latency.

In this paper, we present the results of running deep learning algorithms on CPU and GPU systems. The TensorFlow framework was used to build and train deep learning models for image recognition using Fashion MNIST dataset, and text classification using IMDB (Internet Movie Database) dataset.

From our experiments, it is evident that for small datasets the training accuracy is increasing with an increase in the number of epochs and that the processing time on the GPU is slightly faster than when using CPU. However, there is no change in the accuracy of the test dataset as the number of epochs increases.

Keywords: Deep Neural Networks, Computer Vision, High-Performance Computing, CPU, GPU, TensorFlow

I. INTRODUCTION

A large amount of data, generated by ubiquitous computing devices, requires high-performance hardware and software systems in order to facilitate fast processing of this data and achieving high accuracy of data classification and identification [1]. Dynamics of computation and machine learning has also notably changed due to the development of GPUs and deep learning models. A number of frameworks have been developed to support data processing on new computing architectures. Some of the languages and frameworks that are currently being used are CUDA from NVIDIA, TensorFlow from Google, Pytorch, Digits, Jupyter, and many more.

To process large data sets, in applications such as image detection, classification and identification, new AI techniques have been developed, building on the previous successes in the AI. The aim of this research is to investigate a performance of Deep Learning techniques on GPU and CPU systems, using TensorFlow framework. A number of case studies were conducted in order to evaluate these systems in typical data classification and identification tasks, such as image recognition and text classification. The results

obtained in our experiments demonstrate the difference in GPU and CPU based systems performance. The rest of the paper is organized as follows: Section II presents a short overview of AI, ML and DL techniques, Section III focuses on major developments in ANN that have led to the development of building blocks of deep neural networks. Section IV reviews the frameworks used in this research, Section V outlines hardware requirements for our experiments, Section VI shows the building and implementation of deep learning model, whilst Section VII and VIII present the results of the case studies and future work..

II. ARTIFICIAL INTELLIGENCE, MACHINE LEARNING AND DEEP LEARNING

AI is a field that studies the synthesis and analysis of computational agents that act intelligently [2].

A machine learning method originates from artificial intelligence. It has been very successful, in terms of understanding data (because it handles any complicated problem in an adaptable way) and can predict the outcomes of investigation in a specific domain [3]. The goal of the machine learning process is to refine model parameters when used on many examples. Optimal parameter vector enables the correct classification of examples and requires optimization to maximize the performance of an ML model by iteratively adjusting its parameters until the error is minimized. There are three different types of machine learning.

- Supervised
- Unsupervised
- Reinforcement Learning

The relationship between AI, ML and DL is illustrated in Figure 1 considering a timeline from 1950 to 2010, and DL is defined as a subset of machine learning. Recent research [4] focuses on the models and structures inspired by the human brain referred to as deep learning. DL is applied successfully in computer vision and natural language processing, where it outperforms other machine learning algorithms. Using biology inspired concepts, and the foundational units of the human brain - neurons, complex artificial neural network (ANN) layers can be constructed, with several input and output neurons in the input and output layers, and some hidden layers in between. The inputs and outputs are vectorized representation of particular examples. After training

the model, it is possible to validate the system and verify the model with some test data to check the accuracy in image classification and identification.

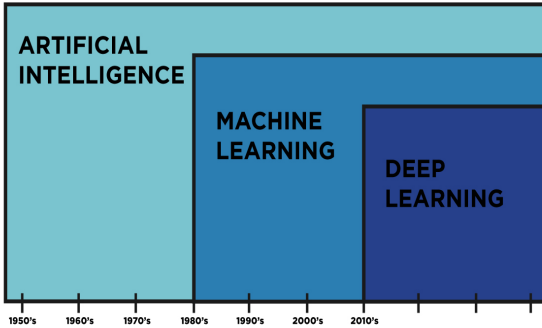


Fig. 1. Difference between AI, ML, DL

The performance of the model can be improved by adjusting parameters in the process of model optimization. DL improves its efficiency with an increase in the amount of data [5]. These are six main types of neural network explained below:

- Feedforward Neural Network
- Radial basis function Neural Network
- Kohonen Self Organizing Neural Network
- Recurrent Neural Network (RNN)
- Deep Convolutional Neural Network
- Modular Neural Network

III. CONVOLUTION NEURAL NETWORK

Convolution Neural Network (CNN) is one of the key types of Artificial Neural Network. Typically, it is used for image recognition. CNNs are made up of several layers that process input data and transform it into an output. CNNs are used for image analysis tests such as scene classification, object detection and segmentation and image processing [6].

In CNNs there are three key concepts: local receptive fields, shared weights and biases, activation and pooling. A small number of neurons are connected by inputs to the hidden layer. These regions are referred to as local receptive regions. The local receptive field is translated across the image to create a feature map from the input layer to hidden layer neurons.

Activation step applies the transformation to each neuron by using activation functions. The activation function (such as ReLU) takes the input of a neuron and maps it to an output value as shown in Figure 2. Convolution layer uses a kernel filter to perform feature extraction. Pooling further condenses the size of convolved features. This reduces the parameters and simplifies the output. Classification of images is then performed using fully connected output layers of the neural network.

IV. FRAMEWORK AND TOOLKIT

Many frameworks for deep learning application are currently being developed. They vary in their architecture,

design and features, and nearly all of them provide great support to developers by giving them a simple and quick execution framework for their applications [7].

A. TensorFlow

It is Python-based open-source framework, which uses dataflow graphs to represent computation [8]. It is originally designed by Google researchers. TensorFlow has the best graph visualization interface compared to other frameworks. It has the capacity to run in diverse environments such as CPU, GPU, mobile devices and a cloud.

B. Jupyter Notebook

It is a web-based interactive development environment which supports multiple languages and is used for data science and for deep learning [9].

Julia + Python + R = Jupyter

Jupyter Notebook enables running of code in a web browser. Firstly, the code is written and runs in small parts called cells. Whenever one cell runs, it will show its output or error which makes it easier to understand and debug.

C. CUDA and CUDA Toolkit

CUDA stands for Compute Unified Device Architecture. It is a software which is built by NVIDIA for parallel computation and programming on GPU. NVIDIA has released CUDA Toolkit, which can perform common computational primitives that are extremely optimized for GPU. The CUDA Toolkit includes GPU accelerated libraries, a compiler, development tools and the CUDA runtime.

D. Anaconda Navigator

It is a graphical user interface that supports the running of different applications, manages conda packages, and is making separate environments (for CPU and GPU) without using a terminal window. It supports many applications such as JupyterLab, Jupyter Notebook, Spyder, and VSCode. Anaconda Navigator is easy to use and supports the installation of different packages in any environment without interfering with other environments.

V. HARDWARE FOR DEEP LEARNING

In the last few decades, technological advances in processing units CPU, GPU, Tensor Processing Units (TPU), dataflow computing, and computer systems such as the High-Performance Computing (HPC) and a cloud, have supported applications of deep learning algorithms in many domains.

A. CPU

CPUs industry had grown along with CPU speed and capabilities to perform general computing tasks, and according to Moore's law the number of transistors and processing power of the processor, until recently, doubled every two years [10]. Modern CPUs have many cores and are capable of running parallel programming tasks, but are often outperformed by GPU accelerated systems.

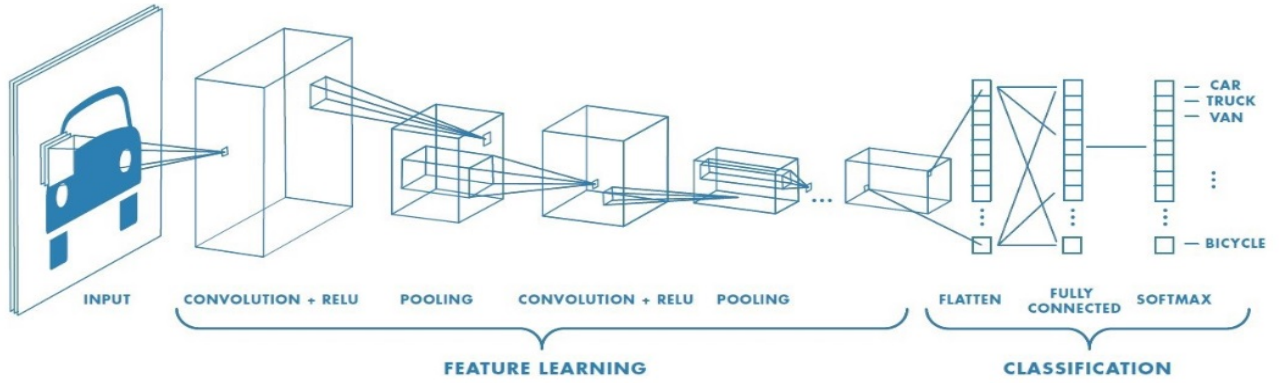


Fig. 2. CNN Architecture [11]

B. GPU

The GPUs exclusively work as parallel architectures, often coupled with the CPUs. Initially, GPUs were used for gaming, rendering or 3D modelling. However, with an improvement in GPU capabilities, they are now being used widely to accelerate computational tasks such as scientific research, deep learning, financial modelling and minerals exploration.

C. TPU

A TPU is an AI accelerator Application Specific Integrated Circuit (ASIC) designed by Google for machine learning application using neural networks and TensorFlow. Just like CPU and GPU, it is a programmable device. It is designed to perform a complex task for many neural networks using matrix architecture instead of vector or scalar data [12]. The TPUs are supporting intensive low-level processing and are using less computing resources than CPUs and GPUs.

VI. IMPLEMENTATION OF DEEP LEARNING

A. Setting environment for DL on a stand-alone machine

To create an environment for DL experiments, it was necessary to prepare a software environment on a stand-alone machine and install CUDA Toolkit 9.0 and cuDNN 7.0, Python, TensorFlow CPU/GPU version, and Anaconda software. An environment was created for TensorFlow and activated with conda commands, and TensorFlow installation was validated. Finally, an environment was created in Anaconda and packages such as TensorFlow CPU/GPU, TensorFlow Board, matplotlib, pyplot, pylint and Jupyter Notebook were installed.

B. Workflow for building and training DL models

A workflow for building and training DL models starts by defining a problem, inputs, potential outputs and vectorized representation of both, and building a neural network to solve it. The input layer should be of an appropriate size to accept the raw data from a text or image, and an output layer could have 2 or 3 outputs. A number of hidden layers, connectivity

and other parameters need to be defined in the internal architecture of the neural network. For supervised learning, a large amount of data is needed to train the network, prepared in uniformly-sized objects that have been labelled. The data needs to be shuffled and divided into separate training, validation and test sets. This is followed by training a model using the training set, an epoch at-a-time. At the end of each epoch, the error between the training set and validation set should be decreasing. When a model is trained it is applied on the test data, and if the performance is not satisfactory, it would be necessary to redesign the network architecture or check that the data has the information required to make the prediction we are interested in.

C. Text classification using IMDB dataset

As an example of DL application, a text sentiment analysis was performed using IMDB dataset of 50,000. The data was divided into two parts each containing an equal number of positive and negative reviews. Each label represents an integer of either 0 or 1, where 0 stands for negative and 1 is for the positive review. A number of layers are required to build a classifier. The first layer is Embedding layer, next is GlobalAveragePooling1D, then Dense layer, and lastly an output layer with a single output node that is using sigmoid activation function, whose value floats between 0 and 1 [13]. The model was able to classify test data to positive and negative reviews with an accuracy of 86%.

D. Image Recognition using Fashion MNIST dataset

Image recognition is one of the applications which was successfully performed using DL. A neural network was trained to classify images of clothes and shoes using Fashion MNIST dataset, which consists of 70,000 grayscale low-resolution images of 28x28. This dataset was divided into two parts, 60,000 images to be used to train our system and 10,000 to test the performance of the system in how accurately images have been classified. The output was a probability distribution, which sums to one. This application was completed successfully, achieving the accuracy of 88% on test data.

VII. RESULTS AND ANALYSIS

It is evident that for small datasets, 50,000 IMDB records and 70,000 fashion images, training accuracy is increasing with an increase in the number of epochs as shown in Figures 3, 4.

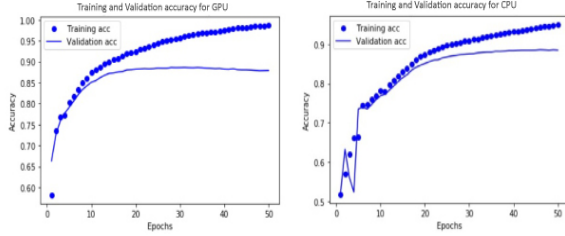


Fig. 3. Accuracy vs Epoch on GPU-CPU training and validation of dataset in text classification



Fig. 4. Accuracy vs Epoch on GPU-CPU training of dataset in Fashion MNIST dataset

There is no change in the accuracy for both cases for the test dataset as the number of epochs is increasing, as shown in Figures 5, 6. Processing time on the GPU is slightly faster than when using CPU.

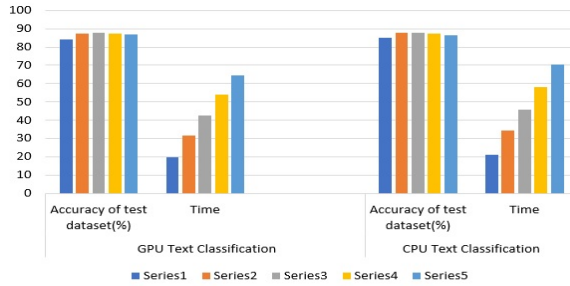


Fig. 5. Time/Accuracy/Epoch graph for GPU-CPU text classification

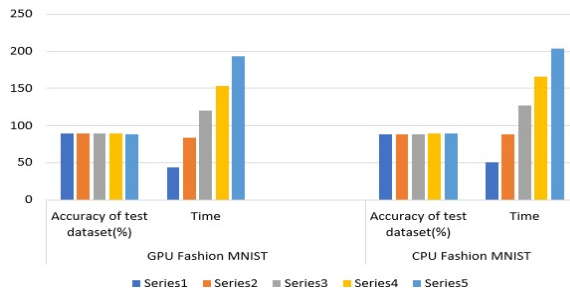


Fig. 6. Time/Accuracy/Epoch graph for GPU-CPU Fashion MNIST dataset classification

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented the results of our investigation of deploying DL techniques in two case studies for text classification and image recognition. The software framework used was TensorFlow and the CPU and GPU versions of DL were implemented on a stand-alone machine. The results of these experiments, with respect to processing time and accuracy, were compared for CPU and GPU implementations. It can be concluded that the processing time for GPU is moderately better than CPU, whilst the accuracy of classification is the same. In future, we will extend our investigation to the systems with more powerful GPU and CPU architectures, and possibly TPU systems, which will lead to a more detailed analysis of the system performance based on the computer architecture used.

ACKNOWLEDGMENT

We would like to acknowledge the contribution of the HPC research group at the University of Huddersfield for providing the resources for this study.

REFERENCES

- [1] Andrzej Kuzelewski and Eugeniusz Zieniuk. Gpu-based acceleration of computations in elasticity problems solving by parametric integral equations system. *Advances in Engineering Software*, 79:27–35, 2015.
- [2] Alan K. Mackworth By David L. Poole. *Artificial Intelligence*. 2017.
- [3] JulianD Olden, JoshuaJ Lawler, and N. LeRoy Poff. Machine learning methods without tears: A primer for ecologists. *The Quarterly Review of Biology*, 83(2):171–193, 2008.
- [4] Difference between artificial intelligence, machine learning and deep learning, 2016.
- [5] V. Sze, Y. Chen, T. Yang, and J. S. Emer. Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329, 2017.
- [6] B. Oh and J. Lee. A case study on scene recognition using an ensemble convolution neural network. In *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pages 351–353.
- [7] Jianing Wei Wei Di, Anurag Bhardwaj. *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. 2018.
- [8] TensorFlow. Tensorflow framework. 2018.
- [9] Jupyternotebook, 2015.
- [10] Gordon Earle Moore. Moores law, 1965.
- [11] Sumit Saha. A comprehensive guide to convolutional neural network-sLthe eli5 way, 2018.
- [12] N. P. Jouppi, C. Young, W. Patil, E. Wilcox, and D. H. Yoon. In-datacenter performance analysis of a tensor processing unit. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pages 1–12.
- [13] TensorFlow. Text classification with movie reviews.

De-noising an Image Using Deep Learning Techniques

Hessah Alattal, Faheem Khan and Qasim Zeeshan Ahmed
Department of Engineering and Technology
University of Huddersfield
Huddersfield, HD1 3DH, UK

Abstract— Image denoising is a traditional task in image processing field and lot of research has been done on this issue. The need to improve denoising performance is a continuous challenge. In this paper, a review of the key ideas related with image denoising is presented and how this issue can be addressed using artificial neural networks as a standard non-parametric statistical tool for pattern recognition, clustering and discriminant analysis. The limitations of traditional fully connected multilayer perceptions on image processing are discussed and it is shown how we can deal with these limitations. This leads to the analysis of the currently used approach in this field known as convolutional neural networks and related Matlab toolboxes on image Processing and deep neural networks. These toolboxes use an available pre-trained denoising convolutional neural network (DnCNN). This existing framework is tested under real conditions and the outputs confirm two of the major claims behind the Matlab DnCNN: the blind denoising capabilities and low time used in the denoising task. Additionally, it was seen that the issue for low noise levels, with signal-to-noise Ratio (SNR) up to 6, the DnCNN will add more error than the noise to be removed. The last leads to suggestion that the use of DnCNN for low noise levels is worth further investigation.

Keywords—Digital image denoising, convolutional neural networks, deep learning.

I. INTRODUCTION

The image noise is an undesirable and inevitable feature present in any image capture process. The noise is usually given by $\mathbf{m} \times \mathbf{n}$ (random matrix) IID (independent and identically distributed) random values (usually from normal distribution) that are added to each of $\mathbf{m} \times \mathbf{n}$ pixels that compose the whole image [1]. In this way, one of the first steps in an image-processing task is to remove the noise, referred to as “image denoising”. The key idea behind this image processing step is to properly identify noise and signal components in each pixel. The last imply that a good noise suppressor must have high discriminant quality to reduce the confusion of signal and noise components. This process has been addressed in many ways for long time, including filtering, smoothing, etc. [2]. From a statistical point of view, the denoising process can be seen as a mean function estimation process over a spatial data.

In recent years with the development of machine learning techniques, the use of neural networks has been used to perform this task as a non-parametric statistical estimator in many research fields, showing quantitative and qualitative improvements in classification and segmentation task, in comparison to traditional statistical approaches that need strong model assumptions [3]. It is important to evaluate the use of this concept on the image denoising issue.

Currently there exist many software that include neural network packages or toolboxes along with image processing. Matlab specifically uses a deep learning noise suppressor, based on a pre-trained convolutional neural network (CNN) [4]. In this paper, the deep learning and image processing toolboxes in Matlab are used to denoising image along with its theoretical framework. This approach shows some important advantages in comparison with other traditional discriminant approaches used for image denoising. The main advantage is that the CNN-based noise suppressor is capable to handle more general Gaussian noise models with unknown noise level known as the “blind Gaussian denoising”. Additionally, the use of the CNN approach helps to boost the denoising performance and has promising improvement in run time by GPU implementation. These main findings put attention on CNN as a promising tool to be applied on existing and upcoming image processing challenges.

II. IMAGE DENOISING BASIC IDEAS

The information on a digital image is usually given by a 2-way matrix of pixel values. Each pixel value comes from a light intensity measurement, by a digital camera directly over the real object or by a digital scanning process from a previous image taken. Due to unavoidable natural noise sources, these measures are taken under noisy conditions. This leads to an output (or measured) matrix with values different to the original image values.

Let us assume that \mathbf{X}_{nxm} is the output image matrix of light intensity values and \mathbf{Y}_{nxm} is the real image matrix. The relationship between these matrices is as follow:

$$\mathbf{X} = \mathbf{Y} + \mathbf{E}, \quad (1)$$

Where $\mathbf{e}_{\mathbf{m} \times \mathbf{n}}$, is a $\mathbf{m} \times \mathbf{n}$ matrix of IID values usually from a normal distribution $N(0, \sigma)$. This is the noise component on the measured image.

The main issue related with (1) is that in real world situations, we usually don't know either \mathbf{Y} or \mathbf{E} matrices, we only have access to the output and noised matrix \mathbf{X} . This leads to the main denoising question: How to obtain a close estimate to the real \mathbf{Y} matrix from the given \mathbf{X} matrix? This inverse problem is the image denoising task. There are many approaches used to perform this task, some of them see this issue from a filtering point of view using a frequency domain representation of the measured matrix \mathbf{X} by fast Fourier transform, and then use a low pass filter under the basic assumption that image signal and noise have enough separation in the \mathbf{X} -spectra, low frequencies components area related to the real image and high frequency components to noise. In other cases, the approach does not imply any domain change and involves

smoothing techniques. Others view this problem from statistical point of view using some probabilistically assumptions closely related to (1). In this paper, we go to develop the denoising task using the deep learning techniques with the help of existing toolboxes in Matlab.

III. DEEP LEARNING FOR IMAGE DENOISING

A. Neural Network as Statistical Learning Framework

The Neural Network approach is based on the use of an artificial neuron, known as perceptron, as a basic building block. In this model, there are J_1 input variables X_i , weighted by the w_i values ($i=1 \dots J_1$) and additionally there is a threshold or bias value θ . They define the neuron *net* function as follow:

$$net = \sum_{i=1}^{J_1} w_i x_i - \theta = w^T x - \theta \quad (2)$$

The neuron output y , is the *net* output transformed by an activation function $\phi(\cdot)$. This function is usually some continuous or discontinuous function, mapping the real numbers into the interval $(-1, 1)$ or $(0, 1)$:

$$y = \phi(net) \quad (3)$$

There are many functional forms for $\phi(\cdot)$, the most used are:

$$\phi(x) = \begin{cases} 1, & x \geq 0 \\ -1 \text{ (or } 0), & x < 0 \end{cases} \quad \text{Hard Limiter (4)}$$

$$\phi(x) = \frac{1}{1 + e^{-\beta x}} \quad \text{Logistic Function (5)}$$

$$\phi(x) = \tanh(\beta x) \quad \text{Hyperbolic Tangent (6)}$$

When more neurons are used and they share its inputs, we have the single layer perceptron. In this network its topology is the feedforward type. The main feature on this topology is that neither neuron has connection with other neuron. The directed arcs go from input nodes (x 's) to neurons, and from neurons to outputs (y 's).

Following this direction are the multiLayer-perceptron (MLP). These type of NN's are usually arranged in the form of layers. As in the single layer type, in such MLP, there is no connection between the neurons in the same layer, and also there is no feedback between layers. In a fully connected layered feedforward network, every node in any layer is connected to every node in its adjacent forward layer.

IV. DEEP LEARNING IN IMAGE DENOISING

The strategy used by DnCNN to address the image denoising task [8] has two major directions: residual learning and batch normalization.

A. DnCNN Main Ideas

Residual Learning: Residual learning of CNN was proposed to solve the performance degradation problem, on which the training accuracy goes down with network depth. The key idea is the assumption that a residual image is much easier to be learned than the original and unreferenced one.

With the residual learning strategy, deep CNN can be easily trained and its performance could be improved [8].

Batch Normalization: There are some problems related to internal covariate shift when CNN are being trained throughout mini-batch stochastic gradient descent (SGD) approach. Due the internal covariate shift, its training efficiency is largely reduced. Batch normalization deals with this issue by incorporating a normalization step and a scale and shift step before the nonlinearity in each layer [8]. For batch normalization, only two parameters per activation are added, and they can be updated with back-propagation. The use of this approach helps to improve the training performance by fast training and low sensitivity to initialization [8].

B. DnCNN Architecture and Features

As in the former discussion, the input of a DnCNN is a noisy image modeled conforming to (1). Instead of other approaches that focus on the problem of learn a function $F(y) = x$ to estimate the true clean image, the DnCNN approach adopt the residual learning strategy to train a residual estimate function $R(y) = e$.

The true clean image estimate is then $x = y - R(y)$. The averaged mean squared error between the true residual images and estimates residual from noisy image. This is the lost function used to learn the DnCNN parameters. This lost function is as follow:

$$l(\theta) = \frac{1}{2} \sum_{i=1}^N \|R(y_i; \theta) - (y_i - x_i)\|^2. \quad (7)$$

In this equation $\{(y_i, x_i)\}$, $i=1 \dots N$ is the set of clear-noised training image pairs. Instead the usually CNN architecture described in section III-B, the DnCNN don't have pooling layers and the size of its convolutional filters are set to be 3×3 [8]. For a DnCNN with depth D , there are three types of layers:

(i) **Conv+ReLU:** For the first layer, 64 filters of size $3 \times 3 \times c$ are used. These filters are used to generate a set of 64 *feature* matrices (maps). Additionally are used rectified linear units (ReLU, $\max(0, \cdot)$), they are used for nonlinearity. In the last formulation c represents the number of image channels, i.e., $c = 1$ for gray image and $c = 3$ for color image.

(ii) **Conv+BN+ReLU:** This is used for layers from 2 to $(D - 1)$. In this case are used 64 filters of size $3 \times 3 \times 64$, and batch normalization is added between convolution and ReLU.

(iii) **Conv:** This is for the last layer (Deep D). In this case c filters of size $3 \times 3 \times 64$ are used to reconstruct the output.

V. DnCNN IMAGE DENOISING WITH MATLAB

The former ideas are implemented in Matlab. A pre-trained DnCNN can be handled by specific functions included in deep learning and image processing Matlab toolboxes [9]. To test the pre-trained DnCNN is perform the denoising task on a real image, and on its outputs are estimating some performance measures. Some basic features of the DnCNN are as follow: 1) Is designed to deal with gray scale images, 2) The total number of layers is 59 [10].

A. Image Testing Procedure

The test procedure is as follow

- 1) **True Image Selection:** In this case an image from the NASA-Mars Curiosity Mission is selected [11].
- 2) **Transform the original image to gray scale.** This could be done with the `rgb2gray` matlab function.
- 3) **Set a noise level:** This level is set according to the true image signal level. The idea is to range a Signal Noise Ratio (SNR) from 10/1 to 1/2 in n -steps. The signal and level is estimated as the observed standard deviation of pixel intensity as follow:

$$\sigma(X) = \sqrt{\frac{1}{N} \sum_{i=1}^N (p_i - \bar{p})^2} \quad (8)$$

The noise level is also the standard deviation of a noise matrix \mathbf{n} , $\sigma(n)$. The SNR is as follow [1]:

$$SNR = \frac{\sigma(X)}{\sigma(n)}, \quad (9)$$

the desired error variance is estimated as follow:

$$\sigma^2(n) = \left(\frac{\sigma(X)}{SNR} \right)^2. \quad (10)$$

- 4) **Develop a sequence of noised images:** This is performed according to (1). The \mathbf{e} matrix is built from $N=n \times m$ IID Gaussian numbers $N(0, \sigma^2_i(n))$ $I = 1..N$. with variance σ^2_i according to (11), for a range of target SNR .

- 5) **Denoising with DnCNN:** This step will be repeated n times with the help of the pre trained DnCNN in Matlab. As a visual output are plotted some denoised images.

- 6) **Error Image Estimation:** For each denoised image is estimated the correspondant residual (error) matrix by $\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}}$

- 7) **Performance Measures:** With N diferent images is then computed some performance measures for each residual image like: a) Mean Square Residual, b) Max Residual.

B. DnCNN Image Denoising Testing

Fig 1 (a) shows the original image in grey scale. This image, as an intensity pixel matrix, is noised by the Matlab command. A Gaussian iid noise with parameters $\mu=0$, $\sigma^2_{\text{noise}} = (\sigma^2_{\text{image}})/16$ is added. This noise variance is set to assure a $SNR=4$, i.e. the signal level is four times the noise level. The noised image is showed in Fig 1 (b). The noised image is denoised with the DnCNN.

The output image (denoised image) is shown in Fig.1 (c). This figure shows a good visual denoising performance when the denoised image is compared with the original image. This perception is visually confirmed with the error image $X_i - \hat{X}_i$. This error matrix with an additional bias value is shown in Fig.1(d). The bias is needed due to close to zero values in error matrix. If we plot the error matrix without bias, we must get a close to black image.

In addition to visual inspection, are estimated statistical measures on the error matrix. These can be appreciated in the context of the following section

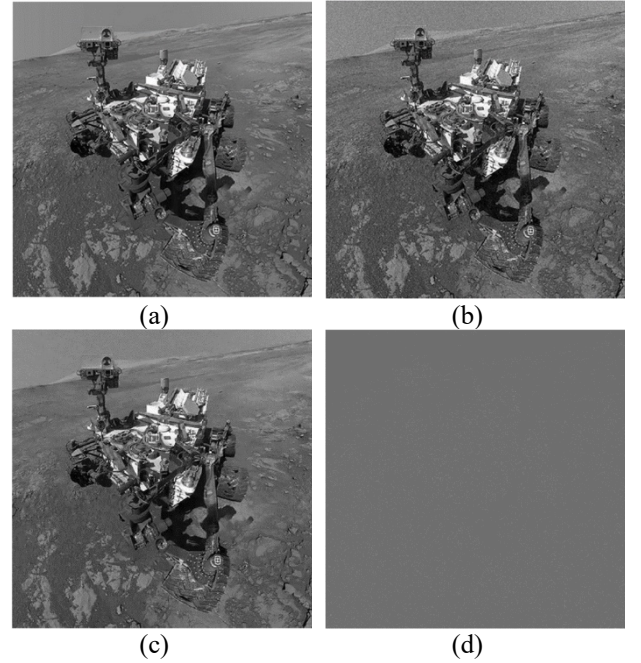


Fig. 1. a) Original Image, b) Noised Image, c) Denoised Image, d) Error matrix

C. Additional Performance Test

Two major claims of the DnCNN are: 1) the robust performance under different and unknown noise levels known as “blind denoising”, 2) the reduced time spent in the denoising process. To test these two major claims, the denoising task is performed under a wide range of SNR levels, and additionally is measured the time spent to perform each image denoising. The Matlab `tic` and `toc` functions are used to get the time used to perform the denoising task by the Matlab `denoiseImage` function. Fig 2 shows these computer performance measures on a Intel Core i3 computer with 6MB of RAM.

In Fig. 2 (a), quick exponential error decay of standard error with SNR is observed. From SNR 6.5 and beyond this decay is slow and close to linear.

Fig 2 (b) shows SER vs SNR . For low SNR values, the SER is greater than SNR . From SNR of 6.5, the SNR is greater than SER . This implies that the denoised image has major deviation from true image than the noised image.

These two measures leads to a good denoising performance for noised images with SNR less than 6. This mean that the DnCNN is not suitable to remove small noise components.

Finally, Fig 2 (c) shows the time spent to perform the image denoising. The maximum value is 10s, additionally it could be seen that beyond a $SNR = 6.5$, the time tend to increase. This is consistent with the results on error and SER . The DnCNN is seen not suitable for very small noise removal with SNR 6.5 and beyond.

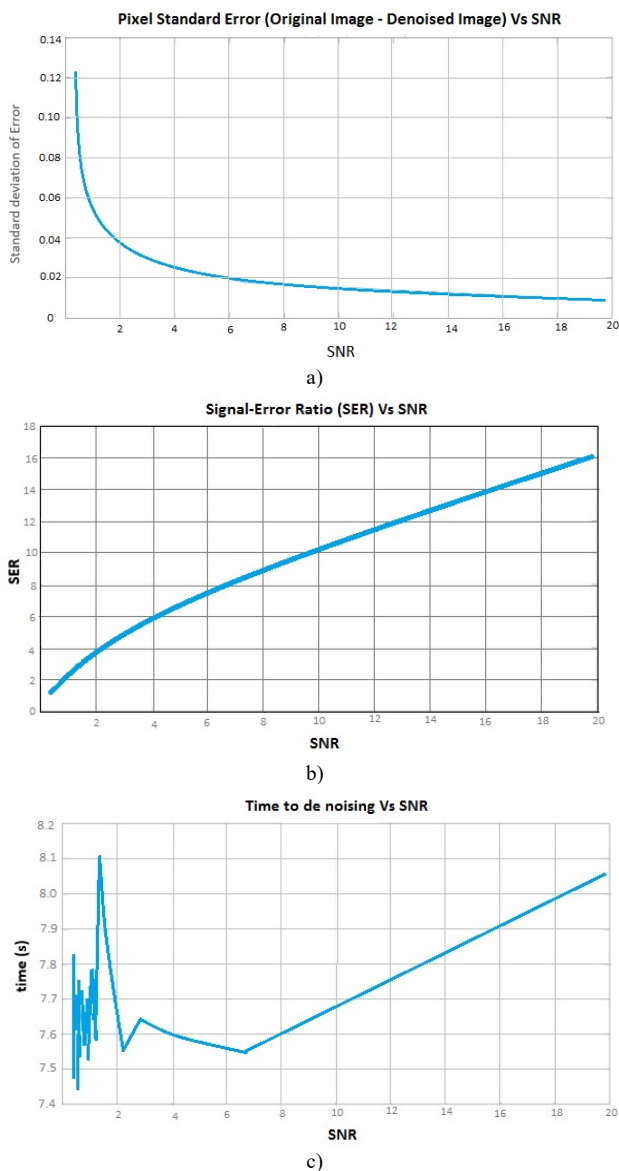


Fig. 2. Performance Measures a) Error signal level, b) Error level, c) Time for image denoising

VI. CONCLUSION

In this paper, we have presented the main ideas behind the theoretical framework of denoising image with convolutional neural networks and its implementation in Matlab. The test results show that the DnCNN has promising performance behaviour under different range of noise levels, blind Gaussian noise, and also use a relatively short time to perform the image denoising task. For very small noise component the DnCNN is not suitable to performs image denoising. If the noise signal is very small the DnCNN must spend more time to perform the denoising task, but this does not lead to improve *SER* when is compared with *SNR*.

REFERENCES

- [1] A. Buades, B. Coll, J. M. Morel, "On image Denoising Methods". *CMLA* (2004) Preprint, 5.
- [2] S. Kaur¹, N. Singh "Image Denoising Techniques: A Review" *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 2, no. 6, June 2014.
- [3] A. Loizos, & M.G. Karlaftis, "Neural Networks and Non-Parametric Statistical models: a Comparative Analysis in Pavement Condition Assessment." *Advances and Applications in Statistics*. 6 (2006).
- [4] K.L. Du · M. N. S. Swamy "Neural Networks and Statistical Learning" Springer-Verlag London (2014).
- [5] A. Deshpande. "A Beginner's Guide To Understanding Convolutional Neural Networks" (2018) <https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/>
- [6] Machine Learning Guru. "Image Filtering". (2018) http://machinelearningguru.com/computer_vision/basics/convolution/image_convolution_1.html
- [7] H. Singhal. "Convolutional Neural Network with TensorFlow implementation". (2017). <https://medium.com/data-science-group-iitr/building-a-convolutional-neural-network-in-python-with-tensorflow-d251c3ca8117>
- [8] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising." *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142-3155, April 2017.
- [9] MathWorks, "Denoise Image Using Deep Neural Network", <https://la.mathworks.com/help/images/ref/denoiseimage.html?lang=en>, (2018).
- [10] MathWorks, "dnCNLayers". (2018) <https://www.mathworks.com/help/images/ref/dnccnnlayers.html>
- [11] NASA. "Mount Sharp 'Photobombs' Curiosity". Image gallery from Mars Curiosity Mission, https://www.nasa.gov/mission_pages/msl/images/index.html, (Page visited January 2019).



PEER REVIEWED PAPERS

Session 3: GPU Computing

Porting and Optimising TELEMAT-MASCARET for the OpenPOWER Ecosystem

Judicaël Grasset

STFC, Daresbury Laboratory
Warrington, United Kingdom
judicael.grasset@stfc.ac.uk

Yoann Audouin

EDF R&D
Chatou, France
yoann.audouin@edf.fr

Stephen Longshaw

STFC, Daresbury Laboratory
Warrington, United Kingdom
stephen.longshaw@stfc.ac.uk

Charles Moulinec

STFC, Daresbury Laboratory
Warrington, United Kingdom
charles.moulinec@stfc.ac.uk

David R. Emerson

STFC, Daresbury Laboratory
Warrington, United Kingdom
david.emerson@stfc.ac.uk

Abstract—TELEMAT-MASCARET is a suite of software for free-surface flow modelling. It is written in Fortran, parallelised with MPI and has been in development since the 1990s. This work aims to parallelise the code on the OpenPOWER architecture without heavily modifying its codebase. To do so the pragma-based programming directives from OpenMP and OpenACC have been tried on IBM POWER8 CPUs and NVIDIA GPUs. The results achieved for the wave propagation module of the suite on GPUs are promising and future works will be carried out on more challenging test cases.

Index Terms—TELEMAT, POWER8, GPU, OPENMP, OPENACC

I. INTRODUCTION

TELEMAT-MASCARET is an open-source suite of hydrodynamic solvers for free-surface flow modelling, originally developed by EDF R&D in the 1990s [1]. Development is now pursued through the TELEMAT-MASCARET Consortium. The software can be used to simulate 2-D or 3-D flows, sediment transport, water quality, wave propagation in coastal areas and rivers. More details on the possible applications and other capabilities can be found on the software’s website [2].

At the moment TELEMAT-MASCARET is only parallelised with MPI, although attempts at hybrid parallelism have been tried in the past [3]. Improving the parallelisation and weak-scaling of TELEMAT-MASCARET would be useful for users who frequently do long time scale simulations.

Current computer trends favour the increase of the number of cores in a single processor and as shown by the current TOP500 list [4], this is alongside the addition of accelerators such as GPUs, combined with memory interconnects designed to reduce latency introduced by transferring data between different memory locations. It is therefore now important that TELEMAT-MASCARET is modified to take advantage of these CPUs and GPUs.

There are two key options when choosing how best to run on GPUs. Either going low-level and programme the kernel directly for GPUs with OpenCL or CUDA, or using pragma-based programming with OpenMP and OpenACC. The first option will give more control and usually more performance but it also means that a specific code has to be written and that

two different versions of the same kernel has to be maintained. However, when using the pragma-based approach, algorithmic changes to code do not infer a re-write of the GPU kernel, with the bulk of the changes needed being the addition of pragmas around the existing code. This approach reduces the burden on those that maintain the original codebase and mean acceptance of changes is more likely. This work therefore concentrates on enabling multi-threaded CPU and GPU acceleration of portions of TELEMAT-MASCARET using a pragma approach.

This paper presents the use of OpenMP in order to reduce the number of MPI processes needed to utilise the CPUs in an OpenPOWER system, combined with the use of OpenACC and OpenMP to offload appropriate computations to available GPUs.

II. RELATED WORK

An attempt to use GPUs with TELEMAT-MASCARET has already been made in [3]. However the method used was completely different from the one described in this article. Belaoura replaced the original matrix-vector product of TELEMAT-MASCARET with the one from the MAGMA library [5], which is able to offload it to GPU. The major problem they encountered was that the MAGMA library was not using the same matrix format. Doing the conversion before and after every matrix-vector product prevented any real-world performance improvement. This work shows how directly accelerating the existing data structures in TELEMAT-MASCARET allows significant gains to be made.

III. PORTING TO THE OPENPOWER ARCHITECTURE

The OpenPOWER foundation [6] is a consortium of entities working to provide an architecture revolving around the IBM POWER processors and accelerators. In this work the architecture used consists of POWER8 processors and NVIDIA GPUs. The processors are interfaced to the GPUs with NVLink instead of PCI-Express. NVLink a high-bandwidth proprietary interface developed by NVIDIA [7], is also used to enable GPU to GPU interconnection.

This work uses the UKRI Science and Technology Facilities Council (STFC) Paragon POWER8 cluster, maintained and run by the Hartree Centre [8] at Daresbury laboratory in Warrington in the UK. Each node of the cluster consists of 2 POWER8 CPUs, each of them with 8 physical cores (up to 8 hardware threads per core) and 4 NVIDIA P100 GPUs with NVLink 1.0 interconnects. Each P100 has 16GB of memory and the 2 POWER8 CPUs share 1TB of memory.

IV. TEST CASE

In order to facilitate the evaluation of the OpenPOWER architecture a test case has been chosen in which most of the computational time is concentrated in a small part of the code and not spread across a lot of different subroutines. Following benchmarks it was decided to use the *fetch_limited/tom_test6* case of the wave propagation module TOMAWAC of the TELEMAT-MASCARET suite. Preliminary benchmarks showed that about 95% of the execution time was spent in a single function called *qnl3*. This function is short and is made of a four-level imbricated loop. As the original test-case mesh was very small, it was refined once in order to increase the computation time. This was achieved with STBTCL, a tool from the TELEMAT-MASCARET suite. The final mesh was made of 18,916 elements and 9,606 points. This test case is part of the official TELEMAT-MASCARET test suite and can be found freely with the source code.

All timings presented in this paper are for the whole duration of the program's execution and not only for the accelerated function. This ensures modifications are generally beneficial for users of the software and not only improvements visible in specific benchmarks.

V. VERSIONS OF SOFTWARE USED

- TELEMAT-MASCARET V8P0R0 (revision 12565)
- Compiler IBM xlf 16.1.1.1
- Compiler GCC gfortran 8.2
- Compiler PGI pgfortran 18.10
- Library CUDA 9.2
- Library IBM Spectrum MPI 10.2.0

VI. TAKING ADVANTAGE OF SMT

Each core of the POWER8 CPU is able to work at different levels of Simultaneous Multi-Threading, (SMT1, SMT2, SMT4, SMT8). This means that each core can execute more than one thread at the same time, e.g. two threads with SMT2. This functionality is comparable with the Hyperthreading technology of Intel processors. While Intel's Hyperthreading can only be used to run a maximum of two threads in parallel, a POWER8 core is able to run up to eight. Benchmarks have shown that TELEMAT-MASCARET does not benefit from the use of SMT8 (maybe because the memory bandwidth is saturated, also SMT8 is not on par with SMT2 or SMT4 as it deactivates the CPU's instruction prefetcher [9]). Standard TELEMAT-MASCARET uses MPI parallelisation and is able to run on thousands of cores [10]. As shown in Table I, the code is able to benefit from using SMT to run MPI processes. It

TABLE I
ORIGINAL MPI VERSION. ONE MPI PROCESS PER ACTIVATED HARDWARE THREAD

PGI pgfortran	SMT1	SMT2	SMT4
1 node	1092s	857s	826s
2 nodes	569s	462s	452s
4 nodes	309s	258s	288s
8 nodes	174s	161s	169s
IBM xlf	SMT1	SMT2	SMT4
1 node	1264s	1018s	1019s
2 nodes	656s	552s	559s
4 nodes	356s	303s	329s
8 nodes	201s	181s	196s
GCC gfortran	SMT1	SMT2	SMT4
1 node	1388s	1034s	974s
2 nodes	639s	546s	526s
4 nodes	344s	295s	309s
8 nodes	193s	176s	182s

is always beneficial to use SMT2 and in some cases SMT4. This work therefore presents results using SMT1, SMT2 and SMT4.

The problem with adding more MPI processes is that it increases the communication time for collective communication. Eventually it is likely that parts of the code will spend more time doing MPI communications than actually performing computation. To decrease this problem this work next looked at using OpenMP to parallelise the *qnl3* subroutine and so reducing the number of MPI processes.

A. OpenMP

The *qnl3* subroutine consists of a four level imbricated loop, with two arrays being updated in the most imbricated loop. OpenMP provides a set of directives to parallelise this kind of problem. In this case the best solution was to add a *parallel for* directive on top of the outermost loop. By doing so the processor is told to distribute the iterations of this to different threads, and each of these threads will execute the whole of the three inner loops. Another point to take into consideration is the fact that several different iterations of the loops can modify the same index of the result arrays, therefore it is necessary to avoid this potential race condition. OpenMP offers two ways of doing this, either by declaring an instruction to be atomic or by using a reduction. Using atomic instructions is usually very costly on CPU, it is therefore preferable to use a reduction. One side-effect of using a reduction is that each thread needs to allocate a temporary array of the size of the original one, which significantly increases the total memory consumption.

Fig. 1 shows the execution time of this implementation with the IBM compiler. Each core executes an MPI process and a number of OpenMP threads, depending on the level of SMT. For instance, on one node with SMT4, 32 MPI processes are executed (16 per processor, 1 per core) and 64 OpenMP threads are executed (4 threads per MPI process). When compared to the original execution time (see Table I) it is clear that the implementation does not perform well.

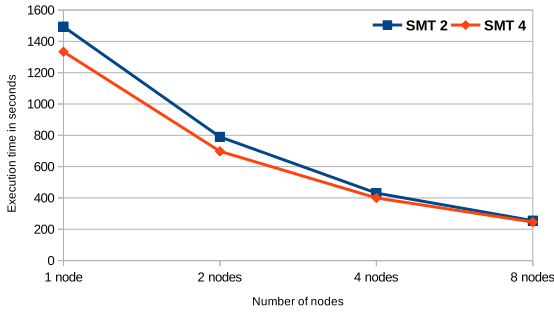


Fig. 1. MPI+OpenMP version. One MPI process per core and one OpenMP thread per activated hardware thread with the IBM compiler

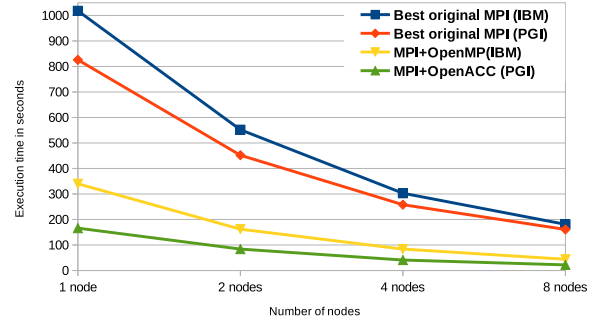


Fig. 2. Comparison of the best original MPI time against the modified MPI+OpenACC on GPU's version and MPI+OpenMP on GPU's version

In fact, in no case was it found to be better to replace MPI processes with OpenMP threads when using the IBM compiler. Executing one MPI process per processor then using all available cores and SMT for OpenMP threads was also tried, but the results were similar to the previous solution, showing no improvements against the pure MPI version with the IBM compiler. Some small tests have shown that there are some performance benefits when using the GCC compiler (see Table II) but the speedup is small (about 1.15x).

TABLE II
COMPARISON OF THE ORIGINAL MPI VERSION AND MODIFIED MPI+OPENMP VERSION ON 8 NODES WITH THE GCC GFORTTRAN COMPILER

	SMT2	SMT4
Original MPI	176s	182s
New MPI+OpenMP	154s	160s

B. Conclusion

Testing and benchmarks have shown that, at least in this specific case with TELEMAT-MASCARET, the use of pure MPI achieves better performance on SMT enabled POWER8 processors than a hybrid MPI+OpenMP approach.

VII. TAKING ADVANTAGE OF GPUS

Following the current trend of adding or increasing the number of GPUs in HPC clusters, Paragon provides four NVIDIA P100 GPUs on each node. We have therefore investigated the possibility of using these to increase the performance of TELEMAT-MASCARET.

A. OpenACC

OpenACC is an open standard set of directives to offload computations on GPUs, the standard is mainly developed by Cray and NVIDIA. While the test cluster used provides three compiler choices (GCC, IBM and PGI), only PGI appears to provide an efficient implementation of the OpenACC standard. The GCC compiler has an OpenACC implementation but it is still a work in progress, and IBM does not implement the OpenACC standard.

The OpenACC implementation for GPUs is quite similar to the OpenMP implementation for CPUs. OpenACC pragmas are used to collapse the four loops in the *qnl3* subroutine used in the *fetch_limited/tom_test6* test-case and distribute the iterations on the available GPUs. The main difference of note for this work between OpenACC and OpenMP is that the version of OpenACC used (version 2.6) does not allow reduction on arrays (this functionality is available in OpenACC 2.7). To replace the reduction, atomic operations are used, in a pure CPU implementation this would be considered a bad approach as atomic instructions are typically slow but here GPU performance implications appear minimal. Using atomic operations also frees the code from creating and merging temporary arrays, leading to no notable increase in memory consumption. This is a welcome result as GPUs often have less memory available than CPUs.

In Fig. 2, results are shown for the OpenACC implementation compared to the original MPI version compiled with the PGI compiler. A notable improvement in execution time can be observed. On one node, the version running on GPUs is five times quicker than the original MPI version, on eight nodes it is seven times faster than the original. The OpenACC version was run on 4 MPI processes and 4 GPUs on each node, with each GPU being linked to an MPI process at the beginning of the program, which is the only process it then communicates with for the duration of its execution.

B. OpenMP

Since version 4.0, OpenMP has offered its own GPU offloading capabilities similar to those provided by OpenACC, again these are pragma-based. Even though the pragmas are completely different from OpenACC those used for offloading are almost functionally equivalent. As the PGI compiler used only supports OpenMP pragmas for CPU, the IBM compiler has been used to evaluate OpenMP GPU offloading performance.

Fig. 2 shows the results for the OpenMP offloading compared to the original MPI version, the two being compiled with the IBM compiler. It can be seen that there is still a notable acceleration when using the GPUs. On one node the version running on GPUs is three times faster than the

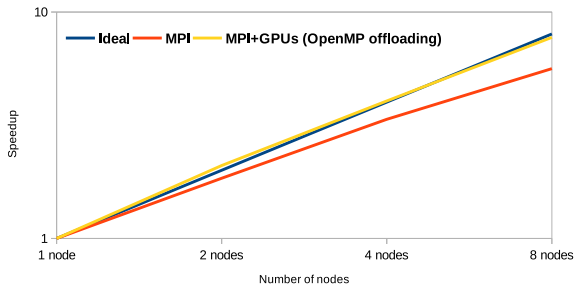


Fig. 3. Comparison speedup of the MPI+OpenMP version on GPUs against the original MPI version

original MPI version and it is four times faster on eight nodes. However the speedup achieved is smaller than the one achieved with OpenACC. In fact, the OpenMP version is about two times slower than the OpenACC version. This difference in performance could be attributed to having to use the IBM compiler rather than the PGI compiler used for the OpenACC tests as the IBM compiler produces slower, standard MPI code as it can be seen in Fig. 2.

C. Conclusion

It has been shown that it is possible, and beneficial, to use accelerators such as GPUs to accelerate some parts of TELEMAC-MASCARET, either by using OpenACC or by using OpenMP. The PGI compiler has been used for the OpenACC implementation and the IBM compiler for OpenMP. It would have been interesting in both case to try the GCC compiler (which should support offloading with either OpenACC or OpenMP) but significant results are yet to be generated, either because the implementation was very slow compared to the other compilers or because it was not working at all.

VIII. GENERAL CONCLUSION

This article first explored the use of a hybrid MPI+OpenMP implementation of the TOMAWAC portion of the TELEMAC-MASCARET suite of solvers for use on an OpenPOWER platform. However, results showed that the classical MPI-only implementation provided better utilisation, even on SMT-enabled POWER8 CPUs. In order to fully utilise the POWER8 platform, an evaluation of the use of GPU acceleration (by way of OpenACC and OpenMP pragmas) was also presented. It was found that it was possible for the test case used in this study to have a five to seven times speedup with PGI and OpenACC and a three to four times speedup with the IBM compiler and OpenMP in comparison to the original MPI version. Finally, as seen in Fig. 3 the scalability is also better than the original MPI version. This increase in performance will benefit users who are using similar cases, they will be able to either run their case quicker or to run it with the same execution time but use the acceleration to increase the accuracy of the simulation.

Future work will look to offload more modules of the suite to GPUs and use test cases provided by users who have computation time distributed across several subroutines. This work will be more complicated and may lead to smaller speedup figures because this will likely involve a larger number of discrete memory transfers between host and GPU. We will also evaluate how OpenMP and OpenACC offloading performs on the GCC 9 gfortran compiler.

IX. ACKNOWLEDGMENTS

This work is supported by the Hartree Centre through the Innovation Return on Research (IROR) programme.

REFERENCES

- [1] GALLAND, Jean-Charles, GOUTAL, Nicole, HERVOUET, Jean-Michel. TELEMAC: A new numerical model for solving shallow water equations. *Advances in Water Resources*, 1991, vol. 14, no 3, p. 138-148.
- [2] <http://www.opentelemac.org/>
- [3] BELAOURA Hamza, Intégration de la bibliothèque MAGMA dans le système TELEMAC-MASCARET, Université de Versailles, Saint Quentin En Yvelines, Internship report
- [4] <https://www.top500.org/statistics/overtime/>
- [5] <https://icl.utk.edu/magma/index.html>
- [6] <https://openpowerfoundation.org/>
- [7] <https://www.nvidia.com/en-gb/data-center/nvlink/>
- [8] <https://www.hartree.stfc.ac.uk/Pages/home.aspx>
- [9] SINHARROY, Balaram, VAN NORSTRAND, J. A., EICKEMEYER, Richard J., et al. IBM POWER8 processor core microarchitecture. *IBM Journal of Research and Development*, 2015
- [10] MOULINEC, Charles, DENIS, Christophe, PHAM, C.-T., et al. TELEMAC: An efficient hydrodynamics suite for massively parallel architectures. *Computers & Fluids*, 2011, vol. 51, no 1, p. 30-34.

Multi-GPU implementation of a 2D Shallow Water Equations Solver over a Multi-Resolution grid.

Massimiliano Turchetto^a, Renato Vacondio^a, and Alessandro Dal Palù^b

^aDepartment of Engineering and Architecture, University of Parma, Parco Area delle Scienze 181/A, 43124, Parma, Italy

^bDepartment of Mathematical Physical and Computer Sciences, University of Parma, Parco Area delle Scienze 53/A, 43124, Parma, Italy

The aim of this work is to present an implementation of a 2D Shallow Water Equations (SWE) solver exploiting the computational capabilities of multiple GPUs spread among a network. This solver is an extension of single-GPU PARFLOOD, which has been proven to be robust and accurate in [2]. The main feature of PARFLOOD is the possibility to run simulations over a multi resolution grid called Block Uniform Quad Tree Grid (BUQG) [1]. While multiresolution grids are largely used by CPU implementations of a vast range of finite volume solvers, today an efficient GPU implementation of such grids is still a challenge due to the difficulties arising when the spatial locality between memory cells cannot be exploited. In this sense the BUQG proved to be a good compromise to exploit multi resolution on GPUs without losing too much efficiency. In the last decade, the need of running fast simulations over increasingly larger grids, has led to a tremendous advancement of HPC systems, most recently also equipped with GPUs. As a natural extension of single-GPU PARFLOOD, the challenge of the multi-GPU version is to run simulations over arbitrary large grids, thus making the code scalable. To this end, the BUQG is partitioned into different parts using an algorithm based on Hilbert Space Filling Curves (HSFC), implemented in the Zoltan Library. Each resulting partition is allocated on a dedicated GPU and managed by a single MPI process; adjacent partitions exchange their borders through MPI messages. The experimental validation of this solver has been carried out performing both the strong and weak scalability tests on the Piz Daint supercomputer. The former, highlighted in Figure 1, shows a dropdown of efficiency slightly better than linear in the number of GPUs, while the latter proved to be constant for a number of GPUs higher than two. Further extensions are currently being implemented, such as the dynamic load balancing and the porting of the code onto IBM Power architecture.

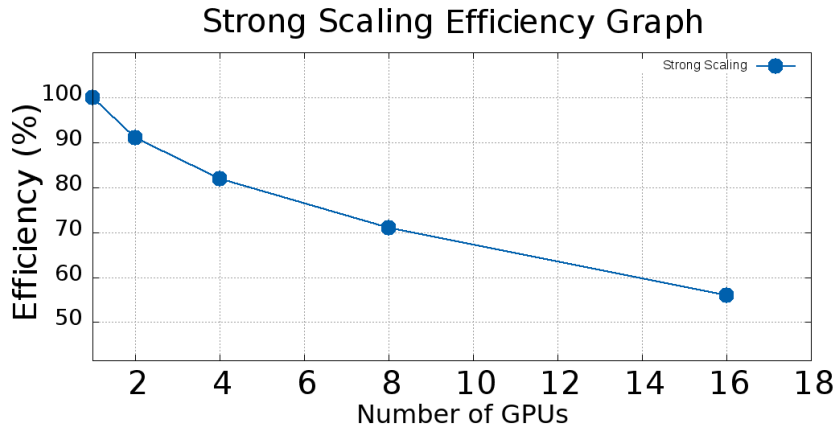


Figure 1: Strong Scalability test performed on the Piz Daint Supercomputer.

References

- [1] R. Vacondio, A. Dal Palù, A. Ferrari, P. Mignosa, F. Aureli, S. Dazzi (2017). A non-uniform efficient grid type for GPU-parallel Shallow Water Equations models. *Adv. Water. Resour.* 88, 119-137
- [2] R. Vacondio, A. Dal Palù, P. Mignosa, (2014). GPU-Enhanced Finite Volume Shallow Water solver for fast flood simulations. *Environ. Model. Softw.* 57, 60-75.

The Desmos supercomputer for computational materials science

V. Stegailov^{a,b,c}, N. Kondratyuk^{a,b,c}, G. Smirnov^{a,b,c}, and A. Timofeev^{a,b,c}

^aDepartment of computer thermophysics, Joint Institute for High Temperatures of Russian Academy of Sciences, Russian Federation

^bInternational Laboratory for Supercomputer Atomistic Modelling and Multi-scale Analysis, National Research University Higher School of Economics, Russian Federation

^cLaboratory of Supercomputer Methods in Condensed Matter Physics, Moscow Institute of Physics and Technology, Russian Federation

Modern MPP systems can unite up to 10^5 nodes for solving one computational problem. The architecture of the individual nodes can differ significantly and is usually selected (co-designed) for the main type of MPP system deployment. The most important component of MPP systems is the interconnect that properties stand behind the scalability of any MPI-based parallel algorithm. In this work, we describe performance results related to the Desmos supercomputer based on 32 1CPU+1GPU nodes connected by the Angara interconnect. Desmos is a supercomputer targeted to MD calculations that has been installed in JIHT RAS in December 2016. Desmos is the first application of the Angara interconnect for a GPU-based MPP system [1, 2, 3].

The Angara interconnect is a Russian-designed communication network with torus topology. The interconnect ASIC was developed by JSC NICEVT and manufactured by TSMC with the 65 nm process. The Angara architecture uses some principles of IBM Blue Gene L/P and Cray Seastar2/Seastar2+ torus interconnects. The torus interconnect developed by EXTOLL is a similar project. The Angara chip supports deadlock-free adaptive routing based on bubble flow control, direction ordered routing and initial and final hops for fault tolerance. The results of the benchmarks confirmed the high efficiency of the Desmos supercomputer for classical MD simulations. The scaling tests for the electronic structure calculations also showed the high efficiency of the MPI-exchanges over the Angara network.

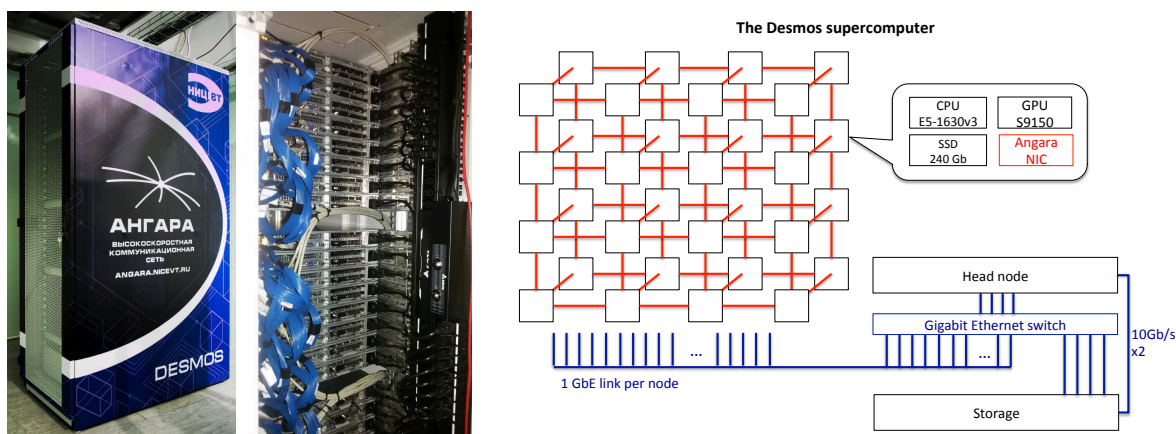


Figure 1: The photos and the scheme of the Desmos supercomputer.

References

- [1] Stegailov V., Agarkov A., Biryukov S., Ismagilov T., Kondratyuk N., Khalilov M., Kushtanov E., Makagon D., Mukosey A., Semenov A., Simonov A., Timofeev A. and Vecher V., *Early evaluation of the hybrid cluster with torus interconnect aimed at cost-effective molecular-dynamics simulations*, LNCS, 10778 pp. 81-90, 2018.
- [2] Kondratyuk N., Smirnov G., Dlinnova E., Biryukov S. and Stegailov V., *Hybrid Supercomputer Desmos with Torus Angara Interconnect: Efficiency Analysis and Optimization*, CCIS, 910 pp. 77-91, 2018.
- [3] Kondratyuk N., Smirnov G. and Stegailov V., *Hybrid Codes for Atomistic Simulations on the Desmos Supercomputer: GPU-acceleration, Scalability and Parallel I/O*, CCIS, 965 pp. 218-229, 2019.



PEER REVIEWED PAPERS

Session 4: Novel Communication Systems

Scalability Analysis of optical Beneš networks based on Thermally/Electrically Tuned Mach-Zehnder Interferometers

1st Markos Kynigos
School of Computer Science
The University of Manchester
Manchester, United Kingdom
markos.kynigos@manchester.ac.uk

2nd Jose A. Pascual
Faculty of Informatics
University of the Basque Country
San Sebastian, Spain

3rd Javier Navaridas
School of Computer Science
The University of Manchester
Manchester, United Kingdom
javier.navaridas@manchester.ac.uk

Abstract—Silicon Photonics is considered a key enabling technology for scaling High-Performance Computing systems into the exa-scale domain. Large-scale optical switches are key components for delivering scalable optical interconnects for High-Performance Computing. However, scaling using Silicon Photonics is inhibited by significant challenges in terms of optical losses and complexity. In this work, we examine the scalability potential of an optical network based on thermally/electrically tuned Mach-Zehnder Interferometers. We describe the system based on this technology and discuss its scalability implications and challenges in terms of optical loss and bit-switching energy consumption.

Index Terms—Silicon Photonics, Optical Benes Networks, Scalability Analysis

I. INTRODUCTION

As high-performance computing (HPC) begins to move into the exa-scale domain, numerous challenges present themselves in terms of system scalability. HPC commonly supports massively parallel workloads, which in turn require a substantial level of communication between the system's compute elements. It is widely acknowledged that interconnection networks constitute a scalability bottleneck for future HPC systems [1]. Furthermore, recent evidence suggests that conventional electrical interconnects will not be able to keep up with system scalability trends in terms of performance, while satisfying the ever-more stringent constraints in power consumption and area [2].

Optical interconnects based on Silicon Photonics (SiPh) have emerged as a promising candidate technology to augment, if not substitute, traditional interconnects. The technology exhibits many benefits that make it a promising solution for future systems. Large scale optical switches are key devices in delivering optical solutions for interconnects in HPC. However, the current state-of-the-art devices as specified in [3] suffer from intrinsic limitations that can lead to increases in optical losses, complexity and package cost.

In this paper, we investigate the scalability potential of an optical thermally/electrically tuned switch based on a Beneš

Network formed with Mach-Zehnder Interferometers (MZIs) [4]. We aim to describe the design of the network based on this technology, as well as to assess the implications of scaling the network in terms of optical losses and bit-switching energy consumption.

II. BACKGROUND

A. Silicon Photonics & MZIs

Silicon is a thoroughly researched material, whose properties have enabled a transformation of the microelectronics domain. Applying the decades of microelectronics research and fabrication experience to photonics, thereby "siliconising" it could enable the continuation of current progress trends. SiPh as a technology is compatible with existing CMOS processes. Due to the nature of the underlying physical principles, the technology exhibits relatively distance-independent energy consumption for communication. Additionally, the technology can offer ultra-high bandwidth capabilities when combined with dense-wavelength-division multiplexing (DWDM). All these factors make silicon photonics a promising candidate for exa-scale interconnects [5].

The basic building blocks of the network we examine here are Mach-Zehnder Interferometers (MZIs). These are devices that leverage the principles of Mach-Zehnder interferometry in order to produce modulation and switching behaviour. A full description of these devices and the underlying principles can be found in [6]. Briefly, these devices are designed to operate relatively uniformly over a large wavelength range [7], and are commonly used to create 2×2 switches.

B. SiPh Beneš Network Switch

In this work, our interest is to examine the scalability of Beneš networks [8] for their use with electro-optic MZIs. Our analysis is based on [4], where a 16×16 Beneš network is constructed out of seven stages of 2×2 MZIs. They contribute an experimental demonstrator and extract a full characterisation of the underlying components. They describe the underlying process used to design and fabricate the basic components, as well as several optimisation processes undertaken to reduce

This work was funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 671553.

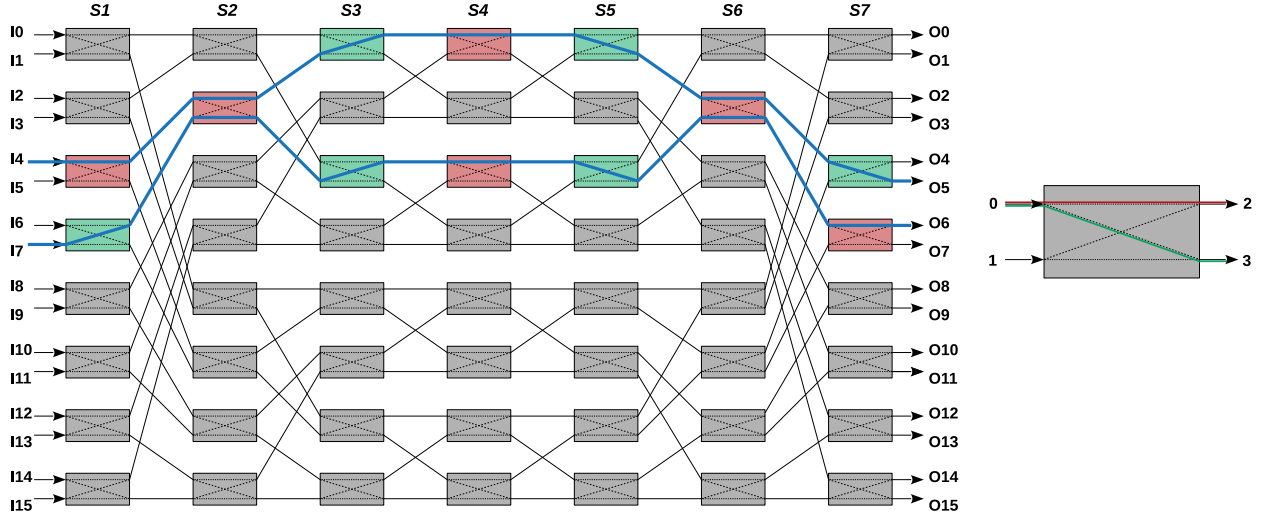


Fig. 1: Left: Topology diagram of the 16x16 Beneš switch. Blue shows paths from I4 to O4 and I7 to O6, green shows MZIs in “cross” state and red in “bar” state. Right: An MZI with port numbers.

the optical loss exhibited by the components. In addition, the thermal and electrical tuning power is reported for each individual MZI which we used as a basis for our analysis, see Table I.

A diagram of the 16×16 Beneš topology can be seen in Fig. 1; the rectangular components represent 2×2 MZI switches. Note we consider binary MZIs where an MZI is either at a “cross” or a “bar” state. Not to be confused with tri-state MZIs [9], with a third, “blocking” state where the phase tuning of interior components forces the switch to be completely blocked. Using tri-state MZIs in an optical network could yield interesting possibilities for future designs; idle elements within a $N \times N$ switch fabric can be tuned to the “blocking” state to dramatically reduce crosstalk, one of the main limitations to scalability.

The Beneš network is a Clos-network variant constructed from 2×2 switches. It requires the minimum number of cross-points to connect 2^i ports in a rearrangeably non-blocking fashion [8]. As such, this paradigm lends itself well to the case of using MZIs as base switching components. Additionally, due to the inherently buffer-less nature of optical communications, packet-switching in optical networks requires electro-optic conversions, which generate huge energy and latency overheads which are undesirable in terms of scalability [10]. However, these overheads can be ameliorated by using circuit switching techniques [11]. The Beneš network’s non-blocking nature can therefore be taken advantage of in terms of path diversity.

C. SiPh Interconnects and Technology

SiPh is acknowledged by the interconnects community as a key enabler for scaling interconnect systems [12], [13]. The community has already proposed many architectures such as

Data Vortex [14], Osmosis [15] or Flexfly [16]. SiPh-enabled systems have also emerged for use in data-centre networks (e.g. [17] or [18]). With recent advances allowing photonic integrated circuits (PICs) using CMOS-compatible processes, a lot of interest has been generated for Optical Networks-on-Chip (ONoCs). A comprehensive study of these can be found here [10]. Notable examples of these are Corona [19], Amon [2] and more recently Venus [20].

The underlying components that make SiPh interconnects possible (e.g. waveguides, microring resonators, MZIs, multi-mode interferometers, transceivers, lasers etc.) are the subject of wide research with novel components being proposed very frequently [21]. For instance, in this work we consider a waveguide optical loss penalty of 1.18 dB/cm (see table I); Thraskias et al. on the other hand mention waveguide-incurred optical losses of as low as 0.2 dB/cm [21]. We note that propagation loss due to waveguides is highly dependent on device technology; nevertheless, this survey illustrates the rate of progress on the technology front. One other key set of components necessary for interconnects is switches; a comprehensive review of the state of the art of SiPh switches can be found in [3], and on MEMS switches for more general optical communications here [22]. As with the model we investigate, the SiPh switches examined in [3] are commonly based on the Beneš topology as well as MZIs with thermal/electrical tuning.

III. SiPH BENEŠ SWITCH AT SCALE

A. Scalability Challenges and Experiment Motivation

As discussed, the MZIs we consider in this paper are *thermally* tuned to reach a “cross” state and *electrically* tuned to reach a “bar” state from that “cross” state. As such, more power is required for an MZI to hold the “bar” state.

TABLE I: Optical Loss and Power Consumption, as reported in [4].

Component	Insertion Loss	Tuning Type	Power Cons.
Waveguide	1.18 dB/cm	Thermal	0-26 mW
Beneš Stage	0.4386 dB	Mean, STD	15.725, 6.608
Waveguide Crossing	0.05 dB		
“Cross” MZI	0.4 dB	Electrical	3.28-5.88 mW
“Bar” MZI	1.4 dB	Mean, STD	5.166, 0.428

Furthermore, an MZI in the “bar” state exhibits substantially more Insertion Loss (hereafter, I_{Loss}) than an MZI in a “cross” state. Thus, using MZIs in the “bar” state generates significant overheads and is therefore considered unfavourable. In addition, note that an I_{Loss} penalty is incurred for each waveguide crossing and that each connection between MZIs entails a different number of crossings. Aggregating the I_{Loss} penalty encountered by a flow can lead to excessive demands on the lasers as the system scales up. As such, it is important to consider ways of reducing these metrics to achieve scale.

These effects, combined with the need to evaluate the system under realistic workloads, outline our experimental motivation.

B. Routing in a SiPh Beneš network

To correctly utilise the model, each element within the MZI array must be electrically/thermally tuned in order to facilitate route allocation and choice.

Each time a flow is to be injected from a source endpoint, the control process calculates the possible routes the flow may take through the network. For N endpoints, each flow can use a maximum of $N/2$ different paths. The route calculation process generates potential paths by varying the interconnected MZIs to be traversed per stage in the left half of the potential path to produce path diversity. The right half of the path is kept stable to ensure the destination endpoint is correctly addressed.

Once all options have been calculated, a path is selected randomly for reservation for the flow to be injected. After selection, the control process iterates through the potential flow paths and, for each encountered MZI, assesses its ability to preserve or switch to the required state. Note that for an MZI in a “bar” state where a previous flow has reserved ports 0 and 2 for example, ports 1 and 3 may be used by another flow. The corresponding scenario applies for the “cross” state as well. If the path assessment completes successfully, the path is reserved by tuning the corresponding MZIs if needed. Otherwise, the process continues for the remainder of the potential paths.

IV. EXPERIMENTS

A. Experiment Setup

The focus of our experimental work is to assess the impact of scaling this network with respect to optical loss and bit-switching energy consumption. To make a more realistic evaluation, we consider the following traffic models:

- **Randomapp** Selects the source and destination uniformly at random.

- **Bisection** Nodes are split into pairs at random and nodes in a pair communicate with each other. This is a best case with no contention.
- **Hotregion** Generates the load from the upper 12.5% of the network, with 25% being directed to the upper 12.5% of the network. The rest is allocated a destination randomly.

We use **phINRFlow** (photonic Interconnection Network for Research Flow-level Simulation Framework), an in-house developed flow-level simulator dedicated to photonic interconnects. This simulator affords a light footprint, is highly scalable and includes the main technological aspects necessary for photonic interconnects. Additionally, the simulator includes a variety of workloads which emulate the behaviour of real applications. Our study assesses maximum Insertion Loss, i.e. the worst-case optical loss exhibited by a flow and bit-switching energy consumption, which is derived from elapsed time and the power metrics found in table I.

B. Maximum Insertion Loss

As mentioned, the aggregated maximum I_{Loss} exhibited by a flow can lead to excessive demands on the lasers within the network. As such, it is imperative to understand how maximum I_{Loss} increases as the system is scaled up, as well as which factors contribute more to the metric. To portray this, fig. 2 depicts a breakdown of maximum I_{Loss} exhibited by flows for the workloads that we use.

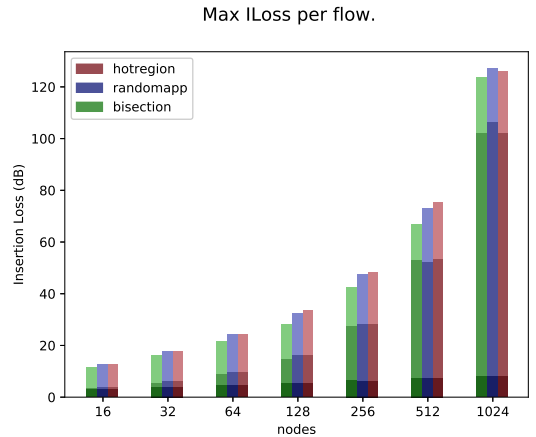


Fig. 2: Breakdown of maximum Insertion Loss per workload. Darker: Waveguide I_{Loss} , mid-gradient: I_{Loss} due to crossings, lighter: MZI-incurred I_{Loss} .

Firstly, it is clear that max. ILoss increases proportionately to network size for all workloads. The least ILoss is consistently exhibited under the bisection workload, something which is expected due to the nature of the traffic distribution. The most ILoss is exhibited under hotregion for all network sizes except the largest, where randomapp exhibits approx. 1.5 dB more ILoss. In all cases, the largest variation in max. ILoss due to workloads is no more than approx. 16% (256 endpoints). Interestingly, for larger network sizes, the chief contributors to max. ILoss are waveguide crossings; this is because the number of crossings scales proportionately to the number of endpoints rather than to the number of stages. However, due to the way MZIs are interconnected, different paths incur a different amount of waveguide crossings; paths that minimise the amount of crossings can be chosen to minimise optical loss. This is an interesting direction for our future work.

For smaller sizes, the chief contributor is MZI-incurred max. ILoss. However, as an MZI in “bar” state incurs more ILoss than one in “cross” state, selecting paths that maximise the MZIs in “cross” can reduce optical losses. Again, we plan to research this in the future.

C. Bit-switching Energy Consumption

The average energy consumption per bit is portrayed in fig. 3. Here, it is clear that energy consumption scales in proportion to the number of stages for all cases. The randomapp workload consistently exhibits the least energy consumption, ranging from 42% less than hotregion for 16 endpoints to 18% less than bisection for the largest size. This is an interesting result which we plan to investigate further in the future. Again, each of the MZI states has different power requirements depending on whether electrical tuning must be applied or not; strategies which prefer paths with the most MZIs in “cross” state should reduce the energy consumption substantially for all workloads.

V. CONCLUSIONS & FUTURE WORK

In this work, we have evaluated the benefits of scaling out a thermally/electrically tuned MZI-based optical Beneš network. We have presented an outline of the system, as well as discussed the implication of scaling to multiple endpoints. In the future, we plan to investigate ways of reducing maximum ILoss and switch energy consumption by leveraging the underlying asymmetries inherent to the switching components. We also aim to explore nested network topologies using variable sizes of this model.

REFERENCES

- [1] D. Thomson and et al., “Roadmap on silicon photonics,” *Journal of Optics*, vol. 18, no. 7, p. 073003, 2016.
- [2] S. Werner, J. Navaridas, and M. Luján, “Amon: An advanced mesh-like optical noc,” in *High-Performance Interconnects (HOTI), 2015 IEEE 23rd Annual Symposium on*. IEEE, 2015, pp. 52–59.
- [3] Q. Cheng, M. Bahadori, M. Glick, S. Rumley, and K. Bergman, “Recent advances in optical technologies for data centers: a review,” *Optica*, vol. 5, no. 11, pp. 1354–1370, Nov 2018.
- [4] L. Lu and et al., “16x16 non-blocking silicon optical switch based on electro-optic mach-zehnder interferometers,” *Opt. Express*, vol. 24, no. 9, pp. 9295–9307, May 2016.

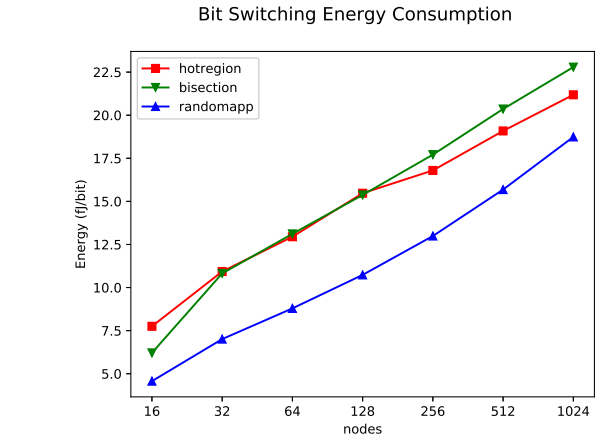


Fig. 3: Average bit-switching energy consumption.

- [5] R. Soref, “The past, present, and future of silicon photonics,” *IEEE Journal of selected topics in quantum electronics*, vol. 12, no. 6, pp. 1678–1687, 2006.
- [6] M. J. Deen, *Silicon photonics : fundamentals and devices*, Chichester, West Sussex, UK, 2012.
- [7] K. Bergman, L. P. Carloni, A. Biberman, J. Chan, and G. Hendry, *Photonic network-on-chip design*. Springer, 2014.
- [8] W. J. Dally and B. P. Towles, *Principles and practices of interconnection networks*. Elsevier, 2004.
- [9] Z. Lu, D. Celo, H. Mehrvar, E. Bernier, and L. Chrostowski, “High-performance silicon photonic tri-state switch based on balanced nested mach-zehnder interferometer,” *Scientific reports*, vol. 7, no. 1, p. 12244, 2017.
- [10] S. Werner, J. Navaridas, and M. Luján, “A survey on optical network-on-chip architectures,” *ACM Comput. Surv.*, vol. 50, no. 6, pp. 89:1–89:37, Dec. 2017.
- [11] J. Bashir, E. Peter, and S. R. Sarangi, “A survey of on-chip optical interconnects,” *ACM Comput. Surv.*, vol. 51, no. 6, pp. 115:1–115:34, Jan. 2019.
- [12] M. A. Taubenblatt, “Optical interconnects for high-performance computing,” *J. Lightwave Technol.*, vol. 30, no. 4, pp. 448–457, Feb 2012.
- [13] S. Rumley and et al., “Optical interconnects for extreme scale computing systems,” *Parallel Computing*, vol. 64, pp. 65–80, 2017.
- [14] O. Liboiron-Ladouceur and et al., “The data vortex optical packet switched interconnection network,” *J. Lightwave Technol.*, vol. 26, no. 13, pp. 1777–1789, Jul 2008.
- [15] R. Luijten and R. Grzybowski, “The osmosis optical packet switch for supercomputers,” in *Opt. Fiber Comm. Conf. and Nat. Fiber Optic Eng. Conf.* Optical Society of America, 2009, p. OTuF3.
- [16] K. Wen and et al., “Flexfly: Enabling a reconfigurable dragonfly through silicon photonics,” 11 2016, pp. 166–177.
- [17] C. Minkenberg and et al., “Reimagining datacenter topologies with integrated silicon photonics,” *J. Opt. Commun. Netw.*, vol. 10, no. 7, pp. B126–B139, Jul 2018.
- [18] N. Calabretta, R. P. Centelles, S. D. Lucente, and H. J. S. Dorren, “On the performance of a large-scale optical packet switch under realistic data center traffic,” *J. Opt. Commun. Netw.*, vol. 5, no. 6, pp. 565–573, Jun 2013.
- [19] D. Vantrease, , and et al., “Corona: System implications of emerging nanophotonic technology,” in *ACM SIGARCH Computer Architecture News*, vol. 36, no. 3. IEEE Computer Society, 2008, pp. 153–164.
- [20] W. Tan, H. Gu, Y. Yang, K. Wang, and X. Wang, “Venus: A low-latency, low-loss 3-d hybrid network-on-chip for kilocore systems,” *J. Lightwave Technol.*, vol. 35, no. 24, pp. 5448–5455, Dec 2017.
- [21] Thraskias and et al., “Survey of photonic and plasmonic interconnect technologies for intra-datacenter and high-performance computing communications,” *IEEE Communications Surveys & Tutorials*, 2018.
- [22] M. C. Wu and T. J. Seok, “Large-scale silicon photonic switches,” in *2018 Asia Communications and Photonics Conference (ACP)*. Ieee, 2018, pp. 1–3.

Performance Analysis of Code-Domain NOMA in 5G Communication Systems

Zeyad Elsaraf, Faheem Khan, and Qasim Ahmed
Department of Engineering and Technology
University of Huddersfield
Huddersfield, HD1 3DH, UK
zeyad.elsaraf@hud.ac.uk

Abstract—Today’s wireless communication networks transmit their signals based on the Orthogonal Multiple Access (OMA) principle. As the number of users increases, OMA based approaches may fail to meet the stringent requirements emerging in the 5th Generation of wireless communications for very high spectral efficiency and massive connectivity. Non-Orthogonal Multiple Access (NOMA) emerges as a solution to improve upon spectral efficiency and user capacity without sacrificing system performance. This paper aims to demonstrate the validity of NOMA as an optimal choice for 5G by comparing it with OMA. Three Code-Domain NOMA (CD-NOMA) schemes are examined and compared with an established OMA technique, Orthogonal Frequency Division Multiplexing (OFDM). The chosen schemes for CD-NOMA are: Low Density Spreading CDMA (LDS-CDMA), Low Density Spreading OFDM (LDS-OFDM), and Sparse Coding Multiple Access (SCMA). The performance of each scheme is evaluated by computing its Bit error rate (BER) and Outage Probability (OP) and simulating them against different values of Signal-to-Noise-Ratio (SNR) over an AWGN channel. It is observed in this paper that, while having varying performance levels, every NOMA scheme outperforms OFDM, thereby proving NOMA to be a prime candidate for implementation in future 5G communication technologies.

Index Terms—5G, orthogonal multiple access, non-orthogonal multiple access, code domain NOMA, spectral efficiency

I. INTRODUCTION

As wireless connectivity spreads across the globe, a challenge for communication systems to accommodate the incoming wave of new users with limited available resources presents itself. To address this issue, novel communication techniques that allow for multiple users to access the same bandwidth have been developed for the fifth generation of mobile communications (5G). The currently utilized Multiple Access (MA) technique, Orthogonal Frequency Division Multiplexing (OFDM), may no longer satisfy the requirements of 5G since it relies heavily on allotting different frequency bands to users and stacking them orthogonally, which limits a system’s user capacity by its bandwidth. Non-orthogonal multiple access (NOMA), however, explores a different approach in regards to increasing the user capacity and spectral efficiency of a system by allowing a number of users to occupy the same frequency band with little inter-user interference (ISI) [1]. NOMA techniques can be roughly categorized into two

main classes: Power Domain (PD-NOMA), and Code Domain (CD-NOMA).

PD-NOMA explores a new dimension to be exploited for increasing user capacity where the available transmitting power at the base station is divided up between the users. A user’s power allocation factor is determined via Channel State Information (CSI). That is, users with low CSI are assigned more power relative to users with higher CSI. Segmenting the available power levels in this manner allows a system to promptly serve users with poor channel conditions or users located at the cell edge as opposed to OMA, where users with higher CSI are favoured and users with poor channel conditions have to wait for access according to the time slot assigned to them. At the transmitter side, users are allocated different power levels and their signals are then superposed and sent through to each user in the system. The difference in power between each user is used to perform Successive Interference Cancellation (SIC) at the receiver side [2]. Signals with higher power levels are subtracted from the received signal, leaving only that user’s low power signal. It is entirely possible to achieve Multiple User Detection (MUD) for an increasing number of users occupying the same frequency subcarrier with more sophisticated SIC methods, although this requires additional processing power at the receiver(s). [3]-[4]

In conventional CDMA, users can share a common channel simultaneously. User separation is done by assigning codes, or spreading signatures, to each user uniquely. However, as a result of this channel sharing, ISI in CDMA-based systems is unavoidable. CD-NOMA mitigates this limitation by utilizing spreading codes with low density signatures (LDS) and interleave sequences. In CD-NOMA, signals are spread using LDS (LDS-CDMA) which are comprised of sparse spreading codes each containing a small number of non-zero elements. The sparsity of the codes allows for the generation of more unique codewords for signal transmission which, in turn, allows for more users to be non-orthogonally superimposed on a chip. Unlike PD-NOMA, by utilizing Message Passing Algorithms (MPA), discussed in [5], user separation at the receiver can be carried out even when the received users’ power levels are comparable. Another advantage of LDS-CDMA is its ability to achieve overloading, that is when the number of users in a system exceeds the processing gain. While the number of

users in an overloaded system requires reduced sparsity of spreading codewords, it was proven in [6] that the number of spreading codes can be increased by up to 300% in a noiseless environment. Overloaded spreading codes are generated in accordance to the Welch Bound Equality [7] in order to reduce ISI.

The LDS-OFDM system can be understood as a system which utilises LDS for multiple access and OFDM for multicarrier modulation mapping. Due to its orthogonal mapping and sparse spreading, LDS-OFDM benefits from frequency diversity as well being able to achieve overloading. This allows a system's user capacity to rise as well as reduce the ISI that would usually accompany it. However, this convenience comes at the price of high, sometimes unaffordable, receiver complexity. [8]

SCMA further optimizes the sparse spreading in LDS-CDMA by combining the LDS spreader with QAM mapping to directly map a set of bits to a complex sparse vector to generate codewords [9]. SCMA codewords are sparse and allow for overloading much like LDS. Codebooks containing multidimensionally mapped codewords replace modulation mapping and spreading, allowing SCMA to benefit from multidimensional and shaping gains as opposed to code repetition in LDS. These gains are offset by a complex design procedure for SCMA codebooks as each multidimensional layer is designed using Euclidean geometry. However, SCMA enjoys a moderate receiver complexity since the codebooks are transparent between the transmitter and receiver. This paper is organised as follows: Section II presents the NOMA techniques' system models. Section III uses the bit error rate and outage probability to evaluate each scheme's performance when transmitting over an AWGN channel (OFDM is used as a base for comparison). Finally, Section IV concludes the paper's findings and suggests further future directions for NOMA research.

II. SYSTEM MODEL

A. LDS-CDMA

Consider a CDMA system with K users. Let $y, \mathbf{H}, x, \mathbf{v}$ denote the superposed transmitted signal, effective received signature, transmitted symbols, and noise vectors respectively. As shown in Fig.1, the LDS spreading in this paper is divided into three stages: Spreading, Zero-padding, and Interleaving. Spreading is done with a randomly generated Hadamard matrix, where the k^{th} user is spread with the codewords in the n^{th} row. Zero-padding and Interleaving are designed to further increase the sparsity of the spread codeword(s) while maintaining the processing gain. The transmitted signal can be represented as

$$\mathbf{y} = \sum_{k=1}^K \mathbf{h}_k x_k + \mathbf{v} \quad (1)$$

which can be further generalised as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (2)$$

The effective received signature can be denoted as

$$\mathbf{H} = \mathbf{A}\mathbf{G}\mathbf{S} \quad (3)$$

where \mathbf{A} , \mathbf{G} , and \mathbf{S} represent the users transmit gain, the corresponding channel gain, and the spreading signature of each user. From (1) and (2), an expression for the received signal of each user can be written as

$$y_k = \sum_{k=1}^K h_k x_k + v_k \quad (4)$$

B. LDS-OFDM

Much like in LDS-CDMA, signal spreading is carried out by spreading, zero-padding, and interleaving. Each users' generated chip is transmitted over a subcarrier belonging to the OFDM mapper where the superposed signal is modulated (Fig.1). Users that are using the same subcarrier are superimposed. Let the set of OFDM data symbols for the k^{th} user, sharing the subcarrier $n = [1, \dots, N]$, be represented as

$$D_{n|k} = \{(k, i) : s_{i,n}^k \neq 0\} \quad (5)$$

where $s_{i,n}^k$ denotes the i^{th} row of the spreading signature matrix \mathbf{s} at the n^{th} subcarrier for the k^{th} user. Let $b_k = [b_1, b_2, \dots, b_K]$ be the set of user data; a transmitted symbol can be presented as

$$x_n^k = \sum_{(i,k) \in D_{n|k}}^{n=N, k=K} b_k s_{i,n}^k \quad (6)$$

As established in (4), the received signal can be denoted by

$$y_n = \sum_{(i,k) \in D_{n|k}}^{n=N, k=K} h_n^k x_n^k + v_n \quad (7)$$

C. SCMA

The SCMA encoding process takes the complex layered codeword i^{th} column of the j^{th} predefined codebook, design method discussed in [10], and uses it to spread the signal of the k^{th} user. Let the set of predefined codewords in each code book be $\mathbf{C}_{(i,j)}$ and the user data set be $b_k = [b_1, \dots, b_K]$, the spread data set can be defined as

$$x_k = \sum_{i,j,k=1}^K \mathbf{C}_{(i,j)} b_k \quad (8)$$

Received signal can then be denoted as

$$y_k = \sum_{i,j,k=1}^K x_k h_k + v_k \quad (9)$$

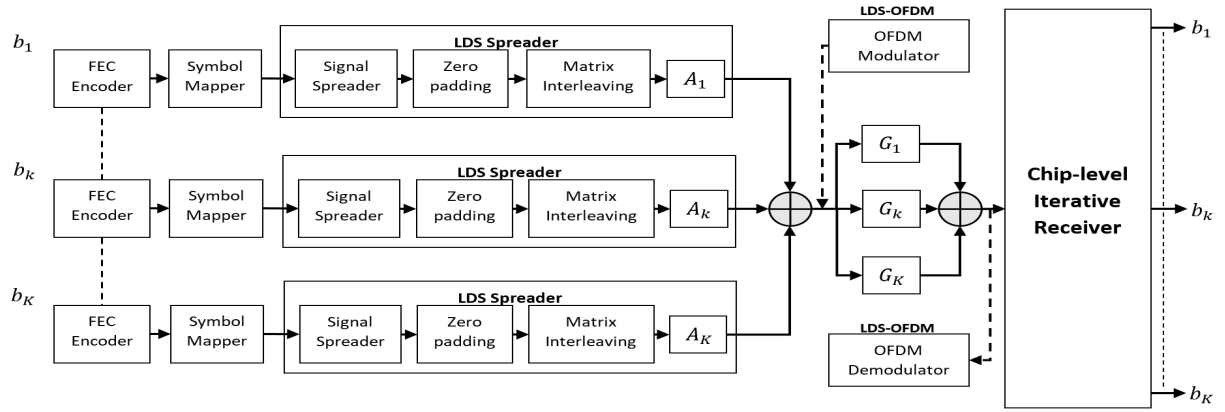


Fig. 1. LDS CDMA/OFDM System Model [2]

TABLE I
SIMULATION PARAMETERS FOR OFDM

Number of Users	16
Symbols Per Frame	53
FFT Length	64
Cyclic Prefix Length	16
Channel Model	AWGN
Modulation	16-QAM
Transmit Antennas	1

TABLE II
SIMULATION PARAMETERS FOR LDS-CDMA

Number of Users	16
Bits Per Signal	2
Hadamard Matrix Size	8x8
Channel Model	AWGN
Modulation	4-QAM

TABLE III
SIMULATION PARAMETERS FOR LDS-OFDM

Number of Users	3
Bits Per Signal	2
Symbols Per Frame	8
Hadamard Matrix Size	8x8
Channel Model	AWGN
Mapping	BPSK
Modulation	OFDM
Cyclic Prefix Length	16
FFT Length	19
Number of Symbols	6

TABLE IV
SIMULATION PARAMETERS FOR SCMA

Number of Users	6
Codebooks	6
Codewords Per Book	4
Bits Per Signal	2
Channel Model	AWGN

III. PERFORMANCE ANALYSIS UNDER AWGN CHANNEL

In this section, the performance of CD-NOMA techniques mentioned in Section II is analysed. OFDM is chosen as the OMA technique to be used as a base for comparison and is simulated in MATLAB using the predefined operator in the system library. The BER of each technique is measured over a range of SNR values while transmitting over an AWGN Channel. The simulation of each technique is carried out according to its respective simulation parameters. In order to ensure result accuracy, each technique runs one value of SNR for 5000 iterations. The bit/symbol error is computed for each iteration then averaged over the total number of iterations before moving on to the next SNR value. The total average error is then normalised to produce the error rate.

The SNR values range from 0 to a maximum of 20 dBs (with +1 increment). A system is considered to be in outage if even one user does not receive 50% of its message for CD-NOMA and more than or equal to 10.6 erroneous symbols

at the output for OFDM. As fig.2 shows, OFDM achieves a BER performance of about 0.01 at approximately 18 dBs of signal to noise power while every CD-NOMA technique achieves the same or lower error rate while requiring much less signal power. LDS-CDMA at 0.01 BER with around 11 dBs, LDS-OFDM at 0.005 BER with 10 dBs, and SCMA at approximately 0 BER with less than 6 dBs. The outage performance of OFDM, as shown in fig.3, remains inoperable until the 15 dB mark unlike CD-NOMA which achieves vastly superior performance while in outage. With LDS-CDMA and LDS-OFDM arriving at 0.45% and 0.3% OP respectively with about 11 dBs and SCMA at approximately 0% OP at less than 5 dBs.

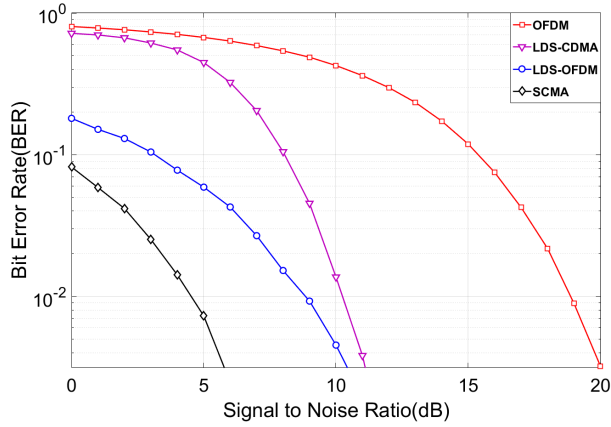


Fig. 2. BER for CD-NOMA Vs OMA

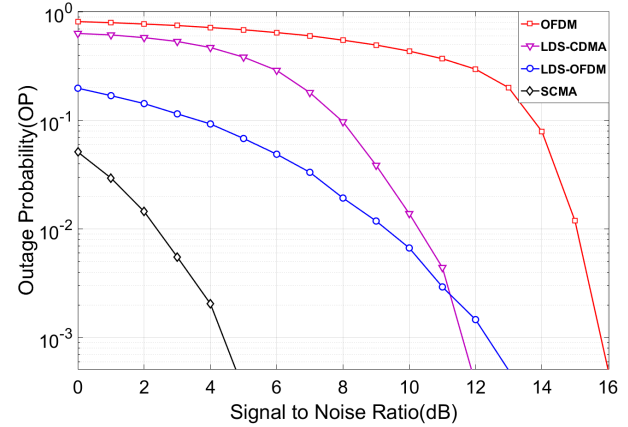


Fig. 3. Outage Performance for CD-NOMA Vs OMA

TABLE V
IMPLEMENTATION FEASIBILITY COMPARISON

	OFDM	LDS-CDMA	LDS-OFDM	SCMA
Encoding Complexity	Low	Low	Average	Very High
Decoding Complexity	Low	Average	Average	Average
Low-SNR Performance	Very Low	Average	High	Very High
High-SNR Performance	Very High	High	High	Very High
ISI	Very Low	Average	Low	Low
Receiver Complexity	Low	Low	Very High	Average
Overall Feasibility in Large Networks	High	Average	Low	Average

IV. CONCLUSION

This paper presented an overview of PD and CD NOMA, experimental simulations, and performance evaluations. NOMA superposes multiple users in the power domain, optimising the usage of available bandwidth by allowing subcarriers to accommodate multiple users, as opposed to dedicated frequency bands in OMA. The simulations have showcased the performance of NOMA and OMA scheme(s) in terms of their bit error rate and outage probability at a range of SNR values while transmitting over an AWGN channel. NOMA was revealed to be superior in terms of bit error as well as outage performance over OMA. Among the tested NOMA schemes, SCMA was shown to have the lowest bit error rate and outage probability at high interference channels and with low transmit power. Despite its highly complex design procedure for generating sparse spreading codewords, SCMA far outperforms other CD-NOMA schemes, making it the most likely candidate for focus in future research. Promising future directions for NOMA include: investigating receiver complexity in NOMA relative to OMA in order to improve on its implementation feasibility, combining the NOMA principle with MIMO in larger networks, investigating the efficacy of NOMA from an energy efficiency standpoint, and applying the co-op transmission scheme to NOMA in an attempt to increase diversity for each user which, in turn, may lead to a better outage performance.

REFERENCES

- [1] Z. Chen, Z. Ding, X. Dai, and R. Zhang, "A mathematical proof of the superiority of NOMA compared to conventional OMA," *IEEE Trans. Commun.*, (submitted) Available on-line at arXiv:1612.01069.
- [2] H. Sadia, M. Zeeshan and S. A. Sheikh, "Performance analysis of down-link power domain NOMA under fading channels," 2018 ELEKTRO, Mikulov, 2018, pp. 1-6.
- [3] S. M. Raizul Islam et al., "Power-Domain Non-Orthogonal Multiple Access (NOMA) in 5G Systems: Potentials and Challenges," *IEEE Communications Surveys and Tutorials*, vol. PP, no. 99, 25 Oct. 2016, pp. 1-1.
- [4] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, V. K. Bhargava, "A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends," *IEEE J. Select. Areas Commun.*, 35(10) pp. 2181-2195, Oct. 2017.
- [5] Anass Benjebbour et al., "Novel Low-Density Signature for Synchronous CDMA Systems Over AWGN Channel," in *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1616-1626, April 2008.
- [6] K. E. Ahmed and M. M. Farag, "Enhanced Overloaded CDMA Interconnect (OCI) Bus Architecture for On-Chip Communication," 2015 IEEE 23rd Annual Symposium on High-Performance Interconnects, Santa Clara, CA, 2015, pp. 78-87.
- [7] L. Welch, "Lower bounds on the maximum cross correlation of signals (Corresp.)," in *IEEE Transactions on Information Theory*, vol. 20, no. 3, pp. 397-399, May 1974.
- [8] R. Hoshyari, R. Razavi, and M. Al-Imari, "LDS-OFDM: an efficient multiple access technique," in *Proc. IEEE 71st Vehicular Technology Conf. (VTC 2010-Spring)*, pp. 1-5.
- [9] H. Nikopour and H. Baligh, "Sparse code multiple access," in *Personal Indoor and Mobile Radio Communications (PIMRC)*, 2013 IEEE 24th International Symposium on. IEEE, 2013, pp. 332-336.
- [10] Taherzadeh, M., Nikopour, H., Bayesteh, A., and Baligh, H. "SCMA Codebook Design," In *Proc. IEEE Vehicular Technology Conference (VTC Fall)*, 2014, pp. 1-5.



PEER REVIEWED PAPERS

Session 5: Novel Software

A high level abstraction approach for lattice Boltzmann simulations using future computing systems

Lokesh K. Ragta, Jianping Meng, Xiao-Jun Gu, and David R. Emerson

Scientific Computing Department, STFC Daresbury Laboratory
Warrington WA4 4AD, United Kingdom.

The lattice Boltzmann Method (LBM) [1] has become a commonly used method to simulate fluid flow problems. The popularity of using LBM is due to its algorithmic simplicity, ease of programming and parallelization. In spite of these advantages, researchers in LBM community nowadays face two major challenges. First challenge is due to the new and emerging computing hardware which often requires the researchers to rewrite their code, optimise the same and port them to each individual computing architecture. The second problem arises due to various mesh requirements in different parts of the domain. The problem under consideration might require to refine/de-refine the mesh during runtime in order to accurately capture the physics and save the computational time and cost.

To address these two challenges, we are developing a domain specific language (DSL) [2] for LBM. The proposed DSL is being linked with two libraries namely: Oxford parallel library for structured mesh solvers (OPS) [3] and AMReX library [4] for massively parallel block structured adaptive mesh refinement (AMR) applications. By incorporating the OPS library in the development of new DSL, the end user will just need to write the code only once for his application and the underlying library will perform the task of code generation, optimisation, translation and porting to various heterogeneous computing platforms such as Graphical Processing Units (GPUs), Xeon Phi, Central processing units (CPUs) + GPUs, etc. While the portability issues on heterogeneous computing systems would be resolved by the OPS library, the AMR capabilities will be provided the AMReX library.

In this talk, we will discuss the current progress of developing the DSL and demonstrate its performance on both single phase and multiphase flows.

References

- [1] Chen, S., and Doolen, G. D. (1998). *Lattice Boltzmann method for fluid flows*. Annual review of fluid mechanics, 30(1), 329-36
- [2] Fowler, M. with Parsons, R. *Domain-specific languages*. Addison-Wesley, (2011).
- [3] The Oxford Parallel Library for Structured-mesh solvers, <https://www.oerc.ox.ac.uk/projects/ops>, and <https://github.com/OP-DSL/OPS>.
- [4] The AMReX library, <https://amrex-codes.github.io/amrex/index.html>.

A constraint-based frequent pattern mining algorithm and its optimisation for multicore systems

1st Sofya Titarenko

School of Mathematics/LIDA
University of Leeds
Leeds, UK

S.Titarenko@leeds.ac.uk

2nd Valeriy Titarenko

School of Biological Science
University of Manchester
Manchester, UK

Valeriy.Titarenko@manchester.ac.uk

3rd Georgios Aivaliotis

School of Mathematics
University of Leeds
Leeds, UK

g.aivaliotis@leeds.ac.uk

4th Jan Palczewski

School of Mathematics
University of Leeds
Leeds, UK

j.palczewski@leeds.ac.uk

Abstract—Pattern mining is an important tool for analysing datasets. Time-dependent data represent a special case. Examples of temporal datasets can be found in environmental or medical monitoring, traffic or mobile applications (data streams or time-series datasets). There are also cases where data are recorded with timestamps, for example activities of internet users or a set of hospital treatments. Unfortunately, temporal records can contain systematic/random errors, which introduce challenges for pattern mining algorithms. Other than uncertainty, complications can be related to additional constraints put on the solution. For example, it might be required to find all frequent patterns with temporal length in a certain range, or patterns which do/do not include a particular item/items. In this work we present a novel constraint-based temporal frequent pattern mining algorithm. The algorithm allows uncertainty in time points as well as temporal and item-based constraints on a pattern. It is highly optimised for modern multicore systems and outperforms existing codes designed to work on sequential datasets (such as SPAM). In this work the algorithm is tested on a weather dataset which includes temperature and precipitation measurements over a set of European cities. The frequent patterns found give an insight into the dependence of different weather conditions between cities.

Index Terms—pattern mining, constraints, multithreading, optimisation

I. INTRODUCTION

Pattern mining is a data mining tool which aids retrieval of important information, or construction of a predictive model. A pattern can be defined as a set of items (sequences or events) which satisfies specific rules (or possesses specific features). The set of rules is defined by the type of dataset and problem to be solved.

In relation to the time-parameter, datasets can be divided into two main categories: 1) not time related 2) temporal datasets. Examples corresponding to the first group can be found in item-based datasets and sequential datasets, such as transactions in supermarkets, where the exact time of purchase is not important. For this group, a pattern consists of items (or a sequence of items). Examples of pattern mining through sequential datasets (itemsets) can be found in [1]–[5]. We can relate all medical, financial, environmental monitoring and hospital or internet user records to the second group. For this group we can define a pattern as a set of temporal events.

It is obvious that taking time into account makes searching algorithms more complex. A special case is timestamped data,

where each event possesses temporal longevity. In the worst case scenario, to describe the time relation between only two temporal events it is required to check all 13 Allen's type relations (see [6]). Therefore, for every found frequent pattern size of n it is necessary to store the corresponding upper triangular matrix of relations (often called a lexicographic order table, size of $n(n+1)/2$). Both storage space and calculative time grow very fast with increase of the length of the frequent pattern to be found. In some problems it is possible to reduce the number of relations which need to be checked (see examples in [7], [8]). In other cases the problem could be simplified by representing a timestamp as a set of its end points. This means mapping of an element $\{e_i, t_i\}$ (where e_i is an event code and t_i is its timestamp) onto the triplet $\{e_i, t_i^s, t_i^e\}$ (where t_i^s and t_i^e correspond to starting and ending points of timestamp t_i). Therefore, the whole timestamped dataset is mapped onto a time series dataset. Examples of this approach can be found in [9], [10]. A good review of time series algorithms is presented in [11].

An additional challenge represents uncertainty in the records. This can be due to faults in sensors, noise, sampling errors (in case of time series) or to a human factor (in the case of hospital treatment records). If this uncertainty is not taken into account, then applying traditional pattern mining algorithms can lead to incorrect results. Probabilistic models allowing uncertainty in pattern mining algorithms have been suggested in [10], [12]–[15] (these models are problem specific). However, introducing uncertainty into big datasets can result in computational problems which can be difficult to overcome. With uncertainty the number of frequent patterns increases dramatically, which makes it challenging in terms of storage space and calculative time. These issues become especially sensitive when working with confidential datasets where the use of remote clusters or clouds is undesirable. Examples of parallel implementation of pattern mining algorithms can be found in [16]–[18].

In this work we present two novel algorithms, c-FaRPaM1 and c-FaRPaM2 which can be applied to a wide range of temporal datasets. Both of them allow temporal uncertainty and temporal and item-based constraints on frequent patterns. The algorithms are highly optimised for multicore workstations. They take advantage of the multithreading and vectorisation

allowed on modern architectures. A new (more compact) data storage structure improves the speed of calculations. The second algorithm (c-FaRPaM2) exploits prior knowledge of the data structure to obtain further improvements in speed.

The algorithms have been tested on a weather dataset. The dataset consists of temperature and precipitation measurements ($\{T; P\}$) recorded daily over a 20 year period in a set of European cities. It is abstracted such that the change of $\{T; P\}$ for every city is found at every time point. For this particular dataset, item-based constraints become very important and allow the problem to be solved quickly and efficiently.

II. BASIC CONCEPTS

Let us consider a database E , consisting of a set of records $R = \{r_i, i = 1 \dots n\}$. Each record r_i contains a number of events $e_{ij} = \{e_j, t_j\}^i$, where e_j is an event code in the i^{th} record, t_j is the time point of its occurrence and the index j corresponds to the position of the event within record i . Suppose, that the precise time point t_j is unknown, but we are certain that the event e_j happen in time interval $[t_j^s, t_j^e]$ with probability $p_j = 1$.

We call a sequence of ordered events $\Pi_m = \{e_1, e_2, \dots, e_m; t_1^s < t_2^s < \dots < t_m^s\}$ a **pattern** Π_m of size m if

$$\forall j = 2, \dots, m: \quad t_j^e > \sup\{t_j^s; t_j^s \in \Pi_{m-1}\}, \quad (1)$$

where $\Pi_{m-1} = \{e_1, e_2, \dots, e_{m-1}\}$ is a sub-pattern of pattern Π_m .

We call a pattern Π_m **frequent with support** $\sigma \in [0; 1]$ if it has been met in $\sigma \cdot n$ number of records.

Sometimes it is desirable to exclude from the solution patterns with temporal length longer than a predefined value. For example, in the weather dataset we don't expect weather changes in different cities to be correlated if they are more then 2 days apart. Similar situations may occur in the processing of hospital treatments or internet query datasets (the value for a temporal constraint is defined from the data structure and problem settings).

We say that a pattern Π_m **has temporary constraints** if it is required that $t_m^e - t_1^s \leq \tau$, where τ is a temporal length.

Sometimes we are interested only in patterns which do/do not contain certain item/items. For example, in application to the weather dataset we may want to look only at patterns which contain characteristics from different cities. However, it is obvious that without a special constraint the large subset of found frequent patterns will contain temperature/precipitation changes located in the same city. This large number of unwanted patterns found in the first step will result in even larger numbers of candidate/frequent patterns in the next steps of the algorithm. Therefore we will quickly approach the storage space/calculative time limits and will not solve the problem we want. In this work we use the following item-based constraint:

Suppose all the codes from dataset E can be categorized in K groups. Let us label every code with index k accordingly. We say that a pattern Π_m is **item-based constrained** if $\forall e_i^k; i = 1 \dots m$ there's no coinciding k indexes.

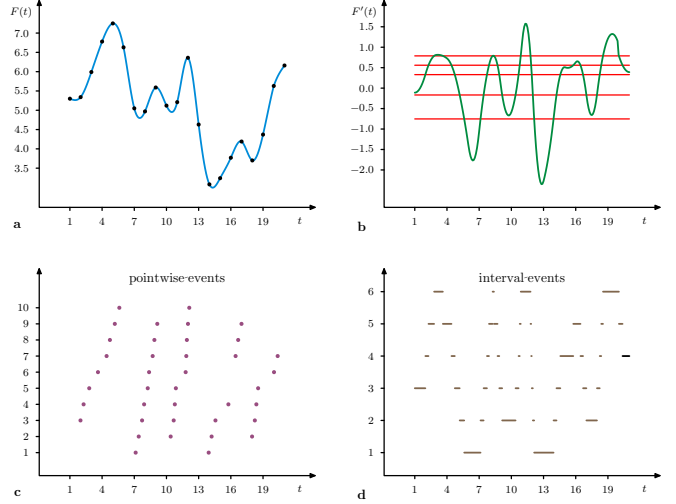


Fig. 1. Abstraction of measured weather parameters: a) measured values of temporal variable $F(t)$; b) $F'(t)$, lines show levels of abstraction; c) pointwise representation of b); d) introducing uncertainty.

III. WEATHER DATASET

In this work we used an open access data source provided by European Commission (via Agri4Cast Resources Portal of the Joint Research Centre, see <http://agri4cast.jrc.ec.europa.eu>). Gridded Agro-Meteorological database contains meteorological parameters from weather stations interpolated on a 25×25 km grid. Meteorological data are available on a daily basis from 1975 to the last calendar year completed, covering the EU Member States, neighbouring European countries, and the Mediterranean countries. The data provide a set of measurement such as mean temperature, mean daily wind speed, vapour pressure, etc..

We chose to work with mean daily temperature and precipitation measurements taken from 14 of European cities. The data have been abstracted according to the procedure shown in Figure 1. At the final step (step c) on the Figure 1 we have a series of events $e_j = \{e_j, t_j\}$ where code e_j corresponds to the change of a chosen parameter at time-point t_j . We chose the number of levels to be equal to 3.

IV. OPTIMISATION

A number of steps have been used to mine patterns quickly and efficiently.

Firstly, we have adopted a **new approach in database storage**. Suppose a record r_i has m events. To keep all the information concerning this record one can use three vectors of size m to store event codes e , starting time points t^s and ending time points t^e with uncertainty intervals τ . It is possible, however, instead of storing all the event codes e_i from the record i , to store only **unique event codes for this record**, and a vector which would say how many of each unique event the record i holds. For example, the record [abacdab] can be stored as vector [abcd] and the number of each of the unique records [3, 2, 1, 1]. This type of representation helps to reduce storage

space (especially for datasets with many repeated events) as well as reduce the time required for pattern searching.

Secondly, in our algorithms **the following property has been used**: all sub-patterns Π_{m-1} of frequent pattern Π_m must be also frequent (the reverse is not true!).

This requires that all the sub-patterns for every candidate pattern Π_m must be pre-checked. This procedure can be simplified if we store this information for every frequent pattern found previously. We have done this using **bitmap ID lists** (an approach similar to the one described in [3]). The use of compressed binary ID lists (bitmaps) also helps to reduce storage space, since every bit of the vector contains useful information. Use of binary logic operators also contributes to the overall speedup.

Thirdly, most time in the algorithm is taken up by the searching procedure. It is **parallelised** by dividing Π_{m-1} space into chunks and sending them to a number of openMP threads to process. After processing, the frequent patterns which have been found are collected, sorted and put in the frequent patterns data base. For example, suppose on the 3rd step the algorithm found frequent patterns [aab], [bdd], [bdb], [aad], [dba], [dca]. The algorithm sends patterns [[aab], [bdd], [bdb]] to thread N1 and patterns [[aad], [dba], [dca]] to thread N2. Each thread generates candidate patterns (by extending the patterns found by 1 item), checking candidates for frequency and returns the found frequent patterns. After that the patterns found are sorted and merged into the database of frequent patterns.

Fourthly, it can be important **to include available prior knowledge about the dataset** in the algorithm. For example, for some uncertain datasets it can be assumed that all uncertainty intervals τ are of the same length. This helps further optimisation of the searching pattern algorithm.

In this work we have implemented all the optimisation steps discussed above. Table I shows how the suggested algorithms are profiled against a Naive Apriori pattern mining algorithm (no bitmap ID lists, no openMP, traditional method of data storage) and the highly optimised sequential pattern mining algorithm SPAM [3]. To be able to profile against SPAM we had to set the uncertainty parameter to zero. However, from the Table I it is clear to see that our algorithms are far more efficient than SPAM (and much more efficient than the Naive Apriori approach), despite being more sophisticated and complex in nature.

TABLE I
RUN TIMES (IN SECONDS) FOR ALGORITHMS WITH ZERO UNCERTAINTY
 $\beta = 0$ FOR THE WEATHER DATASET (L3-D3-T14) MEASURED OVER 14
PLACES IN THE UK.

sup- port	max length	no. patterns	Apriori	SPAM	c-FaRPaM1	c-FaRPaM2
0.5	3	8332	120.1	14.7	1.19	1.21
0.4	4	46848	5942.1	50.4	3.66	3.87
0.3	5	157536	7519.8	219.4	7.80	7.85

V. RESULTS AND DISCUSSION

We emphasise that our research is aimed at developing generic algorithms for pattern mining, not understanding meteorological processes. However the weather dataset provides a testbed for an approach to the analysis of complex data where certain patterns are easily predicted, and we present a preliminary analysis of some results. We can identify three layers of data analysis in the weather dataset: 1) Single or multivariate analysis of temporal patterns at a single site. 2) Single variable spatio-temporal patterns over several sites. 3) Multivariate spatio-temporal patterns over the entire dataset. The third of these is the largest problem, and in accordance with our methodology, analysis of this may benefit from making various prior assumptions regarding the data. Here we present a preliminary analysis of the second layer above using variations in precipitation. Figure 2 shows how at a particular time rainfall may be increasing or decreasing in each locality. Three layers for abstraction of precipitation (temperature) dataset give the following set of changes which can be noticed on Figures:

- small increase;
- moderate increase;
- large increase;
- large decrease;
- moderate decrease;
- small decrease.

Similar categories used for temperature changes.

We have restricted the maximum pattern search time to 7, and divide the data into 7-day chunks giving a maximum length of a pattern in practice of 7 days. The red and green lines on Figure 2 show apparent spatio-temporal patterns in 83% of the records. For example a moderate increase in rainfall in Stockholm is frequently followed the next day by a small increase in Amsterdam, which is followed by a small decrease in Aberdeen. In the temperature data (Figure 3) a small increase in Munich is frequently followed by a small decrease in Montpellier and then a moderate increase in London. These patterns are significant in the data but the physical significance is not clear, since the normal pattern of weather front movement in Western Europe would be from west to east. We believe the patterns maybe related to the periodicity of the data at various scales (for example, seasonal variations on longer time scales but also the short-term periodicity in rainfall and temperature as weather systems pass through on a 1–3 day timescale). Further work will investigate reducing the search length and including temporal patterns at single sites.

VI. ACKNOWLEDGEMENTS

This work has been supported by EPSRC grant EP/N013980/1 (QuantiCode: Intelligent infrastructure for quantitative, coded longitudinal data).

REFERENCES

- [1] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases. VLDB '94, pp. 487–499. Mor-

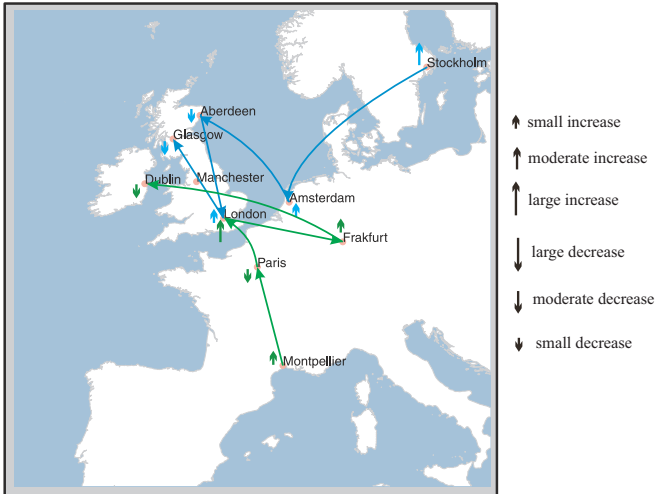


Fig. 2. Examples of found frequent patterns for precipitation (support 83%). Arrows indicate the changes in precipitation.

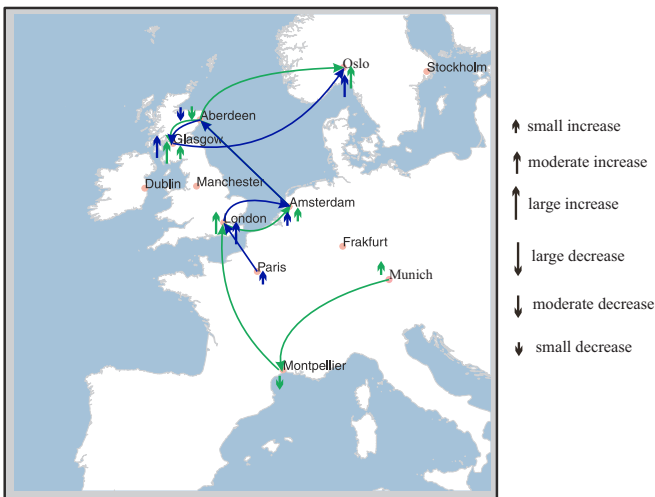


Fig. 3. Examples of found frequent patterns for temperature (support 83%). Arrows indicate the changes in temperatures.

- [7] Batal, I., Cooper, G.F., Fradkin, D., Harrison, J., Moerchen, F., Hauskrecht, M.: An efficient pattern mining approach for event detection in multivariate temporal data. *Knowledge and Information Systems* **46**(1), 115–150 (2016). doi:10.1007/s10115-015-0819-6
- [8] Moskovitch, R., Shahar, Y.: Classification of multivariate time series via temporal abstraction and time intervals mining. *Knowledge and Information Systems* **45**(1), 35–74 (2015). doi:10.1007/s10115-014-0784-5
- [9] Chen, Y.-C., Weng, J.T.-Y., Hui, L.: A novel algorithm for mining closed temporal patterns from interval-based data. *Knowledge and Information Systems* **46**(1), 151–183 (2016). doi:10.1007/s10115-014-0815-2
- [10] Palczewska, A., Palczewski, J., Aivaliotis, G., Kowalik, L.: RobustSPAM for inference from noisy longitudinal data and preservation of privacy. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 344–351 (2017). doi:10.1109/ICMLA.2017.0-137
- [11] Fu, T.-c.: A review on time series data mining. *Engineering Applications of Artificial Intelligence* **24**(1), 164–181 (2011). doi:10.1016/j.engappai.2010.09.007
- [12] Chen, J., Chen, P.: Sequential pattern mining for uncertain data streams using sequential sketch. *Journal of Networks* **9**(2), 252–258 (2014). doi:10.4304/jnw.9.2.252-258
- [13] Cuzzocrea, A., Leung, C.K.-S., MacKinnon, R.K.: Mining constrained frequent itemsets from distributed uncertain data. *Future Generation Computer Systems* **37**, 117–126 (2014). doi:10.1016/j.future.2013.10.026
- [14] Ge, J., Xia, Y., Wang, J.: Towards efficient sequential pattern mining in temporal uncertain databases. In: Cao, T., Lim, E.-P., Zhou, Z.-H., Ho, T.-B., Cheung, D., Motoda, H. (eds.) *Advances in Knowledge Discovery and Data Mining*, pp. 268–279. Springer, Cham (2015). doi:10.1007/978-3-319-18032-8_21
- [15] Wang, L., Cheung, D.W.L., Cheng, R., Lee, S.D., Yang, X.S.: Efficient mining of frequent item sets on large uncertain databases. *IEEE Transactions on Knowledge and Data Engineering* **24**(12), 2170–2183 (2012). doi:10.1109/TKDE.2011.165
- [16] Sutou, T., Tamura, K., Mori, Y., Kitakami, H.: Design and implementation of parallel modified prefixspan method. In: Veidenbaum, A., Joe, K., Amano, H., Aiso, H. (eds.) *High Performance Computing*, pp. 412–422. Springer, Berlin, Heidelberg (2003)
- [17] Qiao, S., Li, T., Peng, J., Qiu, J.: Parallel sequential pattern mining of massive trajectory data. *International Journal of Computational Intelligence Systems* **3**(3), 343–356 (2010). doi:10.1080/18756891.2010.9727705
- [18] Ruan, G., Zhang, H., Plale, B.: Parallel and quantitative sequential pattern mining for large-scale interval-based temporal data. In: 2014 IEEE International Conference on Big Data (Big Data), pp. 32–39 (2014). doi:10.1109/BigData.2014.7004410

gan Kaufmann Publishers Inc., San Francisco, CA, USA (1994). <http://dl.acm.org/citation.cfm?id=645920.672836>

- [2] Agrawal, R., Srikant, R.: Mining sequential patterns. In: *Proceedings of the Eleventh International Conference on Data Engineering*, pp. 3–14 (1995). doi:10.1109/ICDE.1995.380415
- [3] Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential pattern mining using a bitmap representation. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '02, pp. 429–435. ACM, New York, NY, USA (2002). doi:10.1145/775047.775109
- [4] Burdick, D., Calimlim, M., Gehrke, J.: Mafia: A maximal frequent itemset algorithm for transactional databases. In: *Proceedings of the 17th International Conference on Data Engineering*, pp. 443–452. IEEE Computer Society, Washington, DC, USA (2001). <http://dl.acm.org/citation.cfm?id=645484.656386>
- [5] Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery* **8**(1), 53–87 (2004). doi:10.1023/B:DAMI.0000005258.31418.83
- [6] Allen, J.F.: Maintaining knowledge about temporal intervals. *Commun. ACM* **26**(11), 832–843 (1983). doi:10.1145/182.358434

Clone-based Cloud Robotics System for Robot Teleoperation

1st Hamza Aagela

*School of Computing and Engineering
University of Huddersfield
Huddersfield, United Kingdom
Hamza.aagela@hud.ac.uk*

2nd Violeta Holmes

*School of Computing and Engineering
University of Huddersfield
Huddersfield, United Kingdom
v.holmes@hud.ac.uk*

Abstract—Robots are now able to assist humans in many demanding tasks. Teleoperation is one way to combine robot skills and human operator abilities, through remote automatic or manual control. However, modern applications require larger processing and memory resources than those currently available in most robotic systems. The main challenges associated with networked robots occur due to resource constraints, information and learning constraints, and communication constraints. In this paper we present our approach in dealing with teleoperation processes in multi-robot environments that attempt to tackle the Heterogeneity robot challenge. We have implemented clone-based cloud robotic platform (CCRP) which is designed to provide platform-as-a-service (PaaS) for the client robots. In this system, a virtual machine (VM) is assigned in a cloud for every robot. The platform uses Robot Operating System (ROS) as a middleware environment for robot development. The result show that the response time in teleoperation was on average 240ms for Turtlebot robot and 273ms for NAO robot.

Index Terms—Cloud robotics, Teleoperation, Turtlebot, NAO, ROS

I. INTRODUCTION

Although teleoperation was among the first applications in robotics back in the 1950s, it is still one of the most important robot application, posing many challenges to industrialists, researchers and scientists. The remote control of robots is a major task requiring complicated perceptions, decisions and actions to be taken by human operator or autonomous system, using limited environment and robot state information [15]. Nowadays, many teleoperated systems still employ human operators, but the inclusion of automatic control is now more and more common. However, automation rarely completely replaces human operators, because the current technology is still not able to replace fully the operators' actions. The real-time robot teleoperation applications have been implemented in various fields, such as, transportation, underwater exploration, and telesurgery [16]. These applications require a highly reliable platform that can securely manage the communication between a robot and its operator and translate the operator's command to the robot's meaningful signal. However, modern applications require larger processing capability and memory, much more than the resources currently available in robotic systems. The main challenges associated with networked robots occur due to resource limitation, lack of learning capability, and

communication. Cloud robotics is an emerging concept offering access to cloud services as a utility in robot applications [10]. The cloud allows the robots to extend on-board resources, therefore, enabling faster distributed processing and analysis of complex data and tasks [6]. More importantly, it improves the real-time communication between a robot and its operator. The motivation for this work is to deploy and assess efficacy

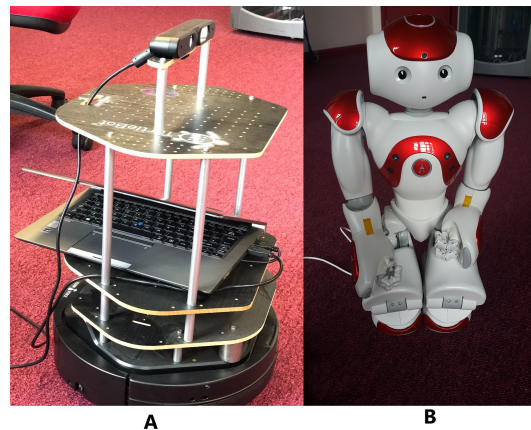


Fig. 1. A) Turtlebot robot B) NAO Humanoid robot

of the CCRP platform in controlling heterogeneous robots in the real-time teleoperation applications. Therefore, a cloud-based multi-robot environment was needed to address the challenges posed by the heterogeneity of robots and evaluate the implementation of a teleoperation task within the CCRP system. The paper is organized as follows: Section 2 outlines the results of related work; Section 3 describes the system architecture of the CCRP system; Section 4 defines the design and implementation of a teleoperation algorithm; Section 5 describes the experiment requirement and setup; Section 6 shows the experimental outcomes; and finally Section 7 provides a research summary and future work.

II. RELATED WORK

This section presents a review of recent efforts to create a range teleoperation platform for heterogeneous robots. The authors in [8] developed the humanoid robot OpenWoZ

framework, which is implemented as a HTTP server running on a robot operating system, and a cloud-backed multi-platform client. The OpenWoZ server uses representational state transfer (REST) protocol to manage the connection requests from a number of users simultaneously. It also allows the adjustment of parameters and behaviors during run-time [8]. A system proposed by [13] called CoWoOZ is based on Telescope project. The CoWoOZ is designed as a cloud-based teleoperation platform that has a web page supporting a number of robot behaviors. The robots receive control signal via HTTP protocol. The teleoperation process is an important part of other robotic tasks such as grasping, mapping and navigation [15], where in most of cases the execution of the tasks require a remote operator intervention. The existing cloud-based teleoperation platforms for controlling robots, which were proposed in reviewed publications, show that the research focus is on developing environment for the targeted types of robots and tasks. However, the multi-robot heterogeneous environments are not considered.

III. CCRP SYSTEM ARCHITECTURE

In this paper we present our approach in dealing with teleoperation task in multi-robot environments [9]. We have implemented clone-based cloud robotic platform (CCRP) as shown in Fig 1, with a NAO robot and a Turtlebot connected via a network with its VMs in the cloud. Likewise an operator can connect to the cloud VM master in order to control the linked robot. The platform is designed to provide platform-as-a-service (PaaS) for the client robots and the operator. The platform uses Robot Operating System (ROS) as a middleware environment for robot development. An operator can remotely connect to the targeted robot via its robot clone image (RCI). In addition, this system enhances the the security of the

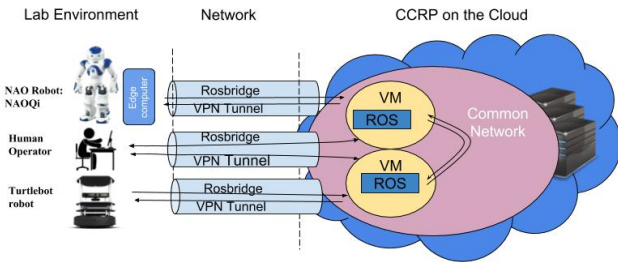


Fig. 2. CCRP System architecture

environment as the connection between the robots and its cloud VM is managed via virtual Private network "VPN" [7], [5], and Rosbridge [4] which establishes a secure tunnel between the robot or edge compute node and its clone VM in a cloud.

IV. TELEOPERATION ALGORITHM

The CCRP teleoperation algorithm is shown in Figure 1, where operator will be responsible of sending the tele-command signal over to the CCRP system, which has additional autonomous correction feature that used to synchronize the state between the operator side and the robot environment.

The robot actuators used to execute the received tele-command signal, which linked to the CCRP system and the sensors used to send the feedback such as video stream from a robot camera, and Odometry data that determines the robot's pose and next move. It was used in the experiments with a wheeled Turtlebot robot and a humanoid robot NAO robot.

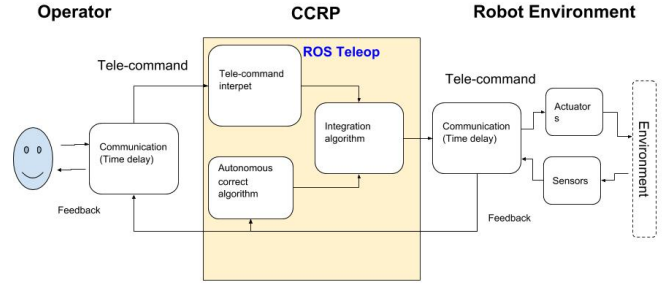


Fig. 3. The CCRP Teleoperation Algorithm

V. EXPERIMENT REQUIREMENTS AND SETUP

The experiments were conducted using a selection of hardware and software components. The hardware requirement are: operator's PC, one NAO robot, edge computer - a laptop for NAO robot, one Turtlebot robot and Rpi as edge device for Turtlebot. The software requirement are: ROS(ROS core, ROS NAO, ROS Turtlebot), OpenVPN, Rivz. The Google cloud platform was used to create two virtual machines (VM) for the experiments. Tutlebot II robot is shown in Fig 1 part A, uses open hardware developed by Willow Garage [3] and the robotic platform runs on a base of motorized wheel. The main sensor for the robot is Asus xtion pro sensor that is mounted on the robot. It allows the robot to capture video and get Odometry data, which is data from motion sensors, that define the changes of the robot position over time. [1]. The humonoid robot NAO is one of the most popular educational robots available since 2008 [2]. As shown in Fig 1 part B, the design of NAO robot resembles human appearance. The lower part of the robot has 11 degrees of freedom with an extra 14 degrees for its upper body. The NAO robot is fitted with a special set of sensors (such as camera, ultrasonic and tactile), the motor actuators for the joints, and LEDs as indicators. The communication with NAO is established using Wi-Fi connection and the Ethernet network. A number of software tools are used in the system. The CCRP platform is utilizing ROS, which is an open-source middleware that provides a number of packages supporting robot functions [14]. The ROS environment manages communication between the ROS nodes and the ROS master the ROS teleop package for both Turtlebot [11] and NAO [12] and ROS video server. A basic teleoperation tasks for heterogeneous robots are applied in an indoor environment. As shown in Fig 4 the robots are required to move from a point A to boint B with and avoid an obstacle placed 2.5m from their initial position.

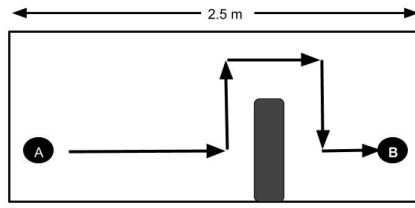


Fig. 4. Robot tasks

VI. EXPERIMENTAL RESULTS

To evaluate the CCRP teleoperation algorithm performance, the robot's response delay, video transmission delay and command delay were measured and data collected and processed from over 30 experiments for each robot. As shown in Fig. 5 the video transmission delay is 150ms demonstrating similar performance in both robots. The response delay represents a time delay between sending a tele-command signal to receiving a from a robot. It was on average approximately 250ms for Turtlebot robot and 273ms for NAO robot. The NAO robot shows slightly slower performance to complete six tele-command scenario. It takes around 70s compared to the Turtlebot which takes only around 25s to accomplish the given task. The difference in performance is due to the difference in both turning speed and forward speed between the two robots used. The results of the experiments show that the proposed cloud robotics solution with CCRP algorithm is effective in handling teleoperation processes, and can enable a real-time remote manipulation of different robot types in multi-robot environments [9].

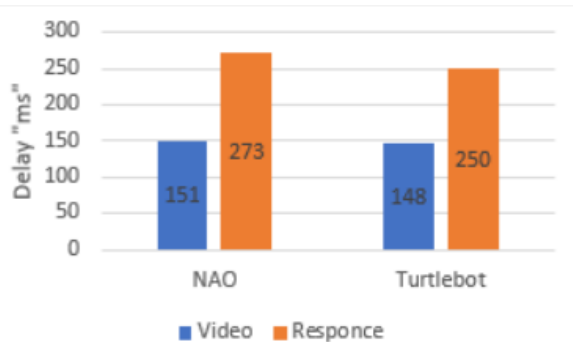


Fig. 5. OpenStack Private Cloud System

VII. CONCLUSION

The paper presents a novel cloud-based robot teleoperation algorithm, aiming to evaluate the capability of the CCRP platform in handling real-time teleoperation for different types of robots. The teleoperation is an effective approach in combining a robot skills and a human operator abilities, through remote automatic or manual control. As the modern teleoperation applications require larger processing power and memory resources, this synergy in of robot and human activities is

not possible for demanding applications, such as environment mapping and navigation. The proposed CCRP platform is able to resolve the lack of the resources in current available solution and overcome constraints posed by lack of information, communication constraints, and robots' ability to learn from previously completed tasks. The result of our experiments show that the CCRP is capable of supporting manual teleoperation application (with an operator's control) due to a low robot response time. In addition it can support heterogeneous robots within the multi-ROS environment. Future work will extend our study to further investigate various teleoperation scenarios aiming to combine an autonomous and operator-based control of remote robots in order to achieve optimal teleoperation performance.

ACKNOWLEDGMENT

We would like to acknowledge the support of the High-performance computing research group and the school of computing and engineering at the University of Huddersfield.

REFERENCES

- [1] Aagela, H., Al-Nesf, M., Holmes, V.: An asus_xtion_probased indoor mapping using a raspberry pi with turtlebot robot turtlebot robot. In: Automation and Computing (ICAC), 2017 23rd International Conference on. pp. 1–5. IEEE (2017)
- [2] Aagela, H., Holmes, V., Dhimish, M., Wilson, D.: Impact of video streaming quality on bandwidth in humanoid robot nao connected to the cloud. In: Proceedings of the Second International Conference on Internet of things and Cloud Computing. p. 134. ACM (2017)
- [3] Claessens, R., Müller, Y., Schnieders, B.: Graph-based simultaneous localization and mapping on the turtlebot platform (2013)
- [4] Crick, C., Jay, G., Osentoski, S., Pitzer, B., Jenkins, O.C.: Rosbridge: Ros for non-ros users. In: Robotics Research. pp. 493–504. Springer (2017)
- [5] Crist, E.F., Keijser, J.J.: Mastering OpenVPN. Packt Publishing Ltd (2015)
- [6] Doriya, R., Chakraborty, P., Nandi, G.: Robotic services in cloud computing paradigm. In: Cloud and Services Computing (ISCOS), 2012 International Symposium on. pp. 80–83. IEEE (2012)
- [7] Feilner, M.: OpenVPN: Building and integrating virtual private networks. Packt Publishing Ltd (2006)
- [8] Hoffman, G.: Openwoz: A runtime-configurable wizard-of-oz framework for human-robot interaction. In: 2016 AAAI Spring Symposium Series (2016)
- [9] Juan, S.H., Cotarello, F.H.: Multi-master ros systems. Institut de Robotics and Industrial Informatics pp. 1–18 (2015)
- [10] Kehoe, B., Patil, S., Abbeel, P., Goldberg, K.: A survey of research on cloud robotics and automation. IEEE Trans. Automation Science and Engineering **12**(2), 398–409 (2015)
- [11] Lee, J.: turtlebot teleop ros wiki (2015), http://wiki.ros.org/turtlebot_teleop
- [12] Lyubova, N.: nao teleop ros wiki (2016), http://wiki.ros.org/nao_teleop
- [13] Magyar, G., Sinčák, P., Magyar, J., Yoshida, K., Manzi, A., Cavallo, F.: Cowooza cloud-based teleoperation platform for social robotics. In: 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI). pp. 000049–000054. IEEE (2017)
- [14] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software. p. 5. No. 3.2, Kobe, Japan (2009)
- [15] Small, N., Lee, K., Mann, G.: An assigned responsibility system for robotic teleoperation control. International journal of intelligent robotics and applications **2**(1), 81–97 (2018)
- [16] Song, Y., Guo, S., Yin, X., Zhang, L., Wang, Y., Hirata, H., Ishihara, H.: Design and performance evaluation of a haptic interface based on mr fluids for endovascular tele-surgery. Microsystem Technologies **24**(2), 909–918 (2018)



PEER REVIEWED PAPERS

Session 6: Applications of Emerging Tech

AI to Facilitate Legal Analysis in the PESTLE Context

Mauro Vallati
School of Computing and Engineering
University of Huddersfield
Huddersfield, United Kingdom
m.vallati@hud.ac.uk

Alessia Grassi
Business School
University of Huddersfield
Huddersfield, United Kingdom
agrassi@hud.ac.uk

Abstract—PESTLE analysis has been used for decades to help companies in taking challenging and complex decisions with regards to aspects such as the development of new lines of products, or the expansion into new markets. Despite its complexity, PESTLE analysis is still performed manually, with issues related to the efficiency of the overall process, and the quality of the suggested actions.

In this work, leveraging on recent advances in Artificial Intelligence, we propose a framework for companies which can be used to support performing PESTLE analysis. In particular, we focus on the Legal aspect of the PESTLE acronym, that is one of the most complex to investigate.

Index Terms—Artificial Intelligence, Marketing, Applications

I. INTRODUCTION

PESTLE analysis is since 1960s the main model utilised by companies and professionals to analyse macroeconomic variables which might influence decision-making processes [5]. It is of pivotal importance when companies (or businesses in general) are assessing the viability of risks or the complexity of actions, such as the diversification of products or the expansion in a new market.

Notably, despite the significant advances of Artificial Intelligence (AI) techniques, PESTLE analysis is currently performed manually. Usually, a large number of human experts, with different background and expertise, have to collect, select, and analyse large amount of information in order to suggest the best course of action to perform in response to the enquiry at hand. This poses a significant burden on the experts, and drastically reduces companies' flexibility and their opportunities to act (and react) quickly; they have to identify the suitable experts and wait for their feedback. Furthermore, the quality of the feedback depends on the expertise of the people involved, and can be therefore hard to predict and to guarantee.

Recent developments in AI, particularly in the area of Knowledge Representation and Reasoning, suggest that a large part of the process of capturing knowledge and reasoning on top of it for the sake of performing a PESTLE analysis can be supported by AI-based agents.

This work has been partially supported by EU H2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 690974 for the project MIREL.



Fig. 1. The PESTLE model, which emphasises the Political, Economical, Social, Technological, Legal, and Environmental aspects to take into account.

In this work, we envisage a framework for exploiting emerging AI techniques for supporting the analysis of the Legal aspects of the model. The decision to focus the study on this specific variable depends on the relative facility in which is possible to categorise and classify elements belonging to the legal environment, thus apply AI techniques to investigate them.

II. PESTLE ANALYSIS

PESTLE analysis is the main model exploited to analyse variables which might influence decision-making processes [5]. In particular, the name of the model is an acronym for the six variables –the first version included only four– which are fundamental to consider by managers in developing their business: Political, Economical, Social, Technological, Legal, and Environmental (Figure 1). Companies utilise this model to investigate these macroeconomics changes which are uncontrollable and unavoidable [2]. By being able to identify,

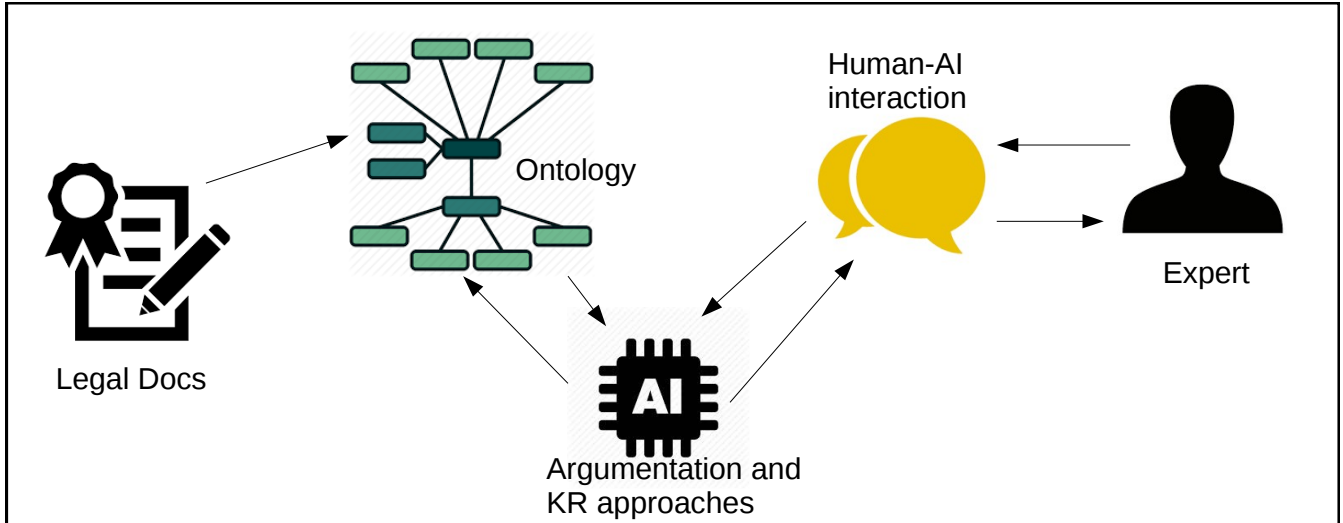


Fig. 2. An overview of the framework.

investigate, and classify the impact of all these variables, managers are facilitated in identifying eventual threats which are not possible to directly control, thus evaluate potential high risks [1]. By doing so they can conceptualise different scenarios, and develop potential alternatives. The model is particularly utilised when business are developing new products, or are expanding in new countries or new markets. However, it is also used on a regular base for understanding markets dynamics and cycles, and as such to eventually evaluate the position, potential and direction for a business [3].

PESTLE analysis is currently performed manually: human experts have to collect relevant information and analyse them in order to suggest the most promising strategy to adopt to achieve the goals of the company.

This paper is specifically focusing on one single element of the six considered by the PESTLE model, the legal environment. With legal environment is intended any law and regulation in force in the specific country (or industry) where a firm has decided to operate, and which might effect the business' decisions [3]. These regulations might effect different aspects of the business. To mention a few examples: health and safety; good practices for packaging and labelling; advertisements policies; codes of practices for market positioning (such as abuse of dominant position); product safety; price transparency; patents; copyright; and working conditions [1]. By being able to effectively investigate and identify all the regulations necessary for their businesses, companies can avoid paying fines and penalties, or in the worst case scenario, of being sued. There are several examples of companies obliged to pay billions of pounds because of legal infringement such as: GlaxoSmithKline forced to pay \$3 billion for misbranding drugs [6]; or more recently Google which was fined \$57 million for breaching Europe's data privacy laws (GDPR) [10]

III. PROPOSED FRAMEWORK

Noteworthy, the AI discipline has been increasingly turning its attention to the automated processing of complex information encoded in a non formal structure, as it is the case of laws and regulations. In fact, two main issues arise when dealing with such type of documents and knowledge: (i) a large body of rules and regulations are not electronically stored; and (ii) they strongly rely on (potentially very different) interpretations, that can depend on the context or on other involved aspects. The first issue does not present conceptual barriers to the application of AI –as it is merely a matter of capturing paper-based documents– and it is being tackled, as it would also lead to a more efficient exploitation of such documents by human experts. The latter issue is instead much more complex from an AI perspective, and is now object of significant research. Dedicated conferences and workshops, such as JURIX¹ and ICAIL², and journals such as *Artificial Intelligence and Law* (Springer) focus on the design and development of AI approaches fit for the purpose of analysing and processing legal documents.

Thanks also to the aforementioned venues, recent developments and emerging approaches in AI, particularly in the areas of Argumentation and Knowledge Representation, suggest that a large part of the process of capturing knowledge and reasoning on top of it for the sake of performing a PESTLE analysis can be supported by AI-based agents. As a matter of fact, the actual degree to which AI can support the legal reasoning is yet to be understood, and will be the focus of our future investigation. However, it is now possible to design an AI-based framework that would allow to support the PESTLE analysis via AI, with an emphasis on the legal aspects entailed by the model.

¹<http://jurix.nl>

²<https://icaill2019-cyberjustice.com>

The proposed framework is depicted in Figure 2. Legal documents can be mapped into an appropriate Ontology (see, e.g. [7], [8]). In a nutshell, an ontology provides a structured way to store, process, and search knowledge [9]. In a typical ontology, *entities* can be defined, and relations between entities can be described and established. Furthermore, characteristics and attributes of each entity can be specified, so that an overall structure can be designed and exploited for processing purposes.

The knowledge stored in the ontology can then be analysed using argumentation approaches [4], in order to provide an overview of the specific legal matter to the user, with pointers to related paragraphs and a first argumentative feedback. The field of argumentation provide means that can support automated reasoning, in terms of proposing arguments and counter-arguments to support or defeat a given statement, that is very similar to the way in which human experts would argue and debate. In that, conclusions reached by the approach can be easily investigated and explained, and the strength and validity of raised arguments can be assessed.

Notably, the human expert is still part of the process, and it is not envisaged to remove human expertise. This is for two main reasons. First, the human expert can make sure that the analysis performed by the framework, as well as the provided reasons and arguments, is sound. It may of course be the case that some notions have been misinterpreted by the framework, or that some conclusions are based on, for instance, debatable or controversial articles and bodies of text. Second, the human should be able to interact with the framework in order to explore different scenarios and possibilities, either by changing the posed query or by objecting on some steps of the argumentative process.

The proposed framework could not be extensively empirically tested, due to the required amount of information and to the required strong involvement of companies and marketing experts. However, we had some qualitative discussions with PESTLE and marketing experts. Such discussion clearly indicate that an AI-based support for dealing with legal aspects would be favourably received by companies, that are currently struggling in quickly comparing different strategies and are therefore forced to take complex decisions relying on limited amount of knowledge.

IV. CONCLUSION

In this work we introduced a framework that would allow to exploit emerging AI techniques for supporting the analysis of the Legal aspects of the PESTLE model. The focus on this one element is justified by its importance and complexity. The proposed framework leverages on recent advances in areas of AI that allows to represent and organise concepts and norms under the form of ontologies, and on automated reasoning techniques that allow to reason upon the structured knowledge.

We see several avenues for future work. First, we are interested in developing a prototype of the proposed framework, by focusing on synthetic legal data. Second, we are eager to

collaborate with companies in order to obtain real data to empirically test the proposed framework, and to gather additional insights from human experts. Finally, we plan to engage with potential users in order to design a suitable interface of the envisaged framework, that will play a significant role in the success of the approach.

REFERENCES

- [1] Baines, P., Fill, C., Rosengren, S., *Marketing* (Fourth ed.). Oxford: Oxford University Press, 2017.
- [2] Baines, P., Fill, C., Rosengren, S., Antonetti, P., *Fundamentals of marketing*. Oxford University Press, 2017.
- [3] Baker, M. J. (2014). *Marketing strategy and management* (Fifth ed.). Place of publication not identified: Palgrave Macmillan.
- [4] Bongiovanni G., Postema G, Rotolo A., Sartor G., and Valentini C., *Handbook of Legal Reasoning and Argumentation*, Springer, 2018.
- [5] Del Marmol T., Feys B., and Probert C., *Pestle analysis*, Namur: Lemaitre Publishing, 2015.
- [6] Marte, J. Hill, C. (2013, October 21). 5 of the biggest corporate fines ever. MarketWatch. Retrieved from <https://www.marketwatch.com/story/5-of-the-biggest-corporate-penalties-ever-2013-09-27>
- [7] Nguyen H., Nguyen V., and Vu V., *A knowledge representation for Vietnamese legal document system*, In Proceedings of the International Conference. on Knowledge and Systems Engineering, 2017.
- [8] Palmirani, M., Martoni, M., Rossi, A., Bartolini, C., Robaldo, L., *Legal Ontology for Modelling GDPR Concepts and Norms*, In Proceedings of the Conference on Legal Knowledge and Information Systems, 2018.
- [9] Staab, S., Studer, R. eds., *Handbook on ontologies*. Springer, 2010.
- [10] Wagner , K. (2019, January 22). Google fined for violating Europes new data privacy law. Retrieved from <https://www.recode.net/2019/1/22/18193250/google-gdpr-fine-france-privacy-57-million>

Holistic Approach to Energy and Power Management in HPC

Vadim Elisseev
IBM Research
United Kingdom
vadim.v.elisseev@ibm.com

Abstract—Exascale level of High Performance Computing (HPC) implies performance under stringent power constraints. Achieving power consumption targets for HPC systems requires energy efficiency optimizations throughout the whole HPC stack. Our approach to energy and power management in HPC systems is being presented.

Index Terms—High Performance Computing , Energy Efficiency, Energy Aware Scheduling, IoT, Machine Learning

I. MOTIVATION

Upcoming High Performance Computing (HPC) systems are on the critical path towards delivering the highest level of performance for large scale applications. As supercomputers become larger in the drive to the next levels of performance, energy efficiency has emerged as one of the foremost design goals. Relying upon contemporary technologies is simply not enough - as power demand for Exascale class systems would require hundreds of Megawatts of power. Currently, the most power efficient HPC system on the Green500 [1] list is Shoubu System B located at ACCC, RIKEN with 17GFlops/Watt. In order to achieve a sustainable power draw, future HPC systems will have to feature a power efficiency of around 50GFlops/Watt [2].

New approaches for energy optimization are being explored, which optimize throughout the whole HPC stack - from firmware and hardware through to the OS, applications and workload managers [3]. The challenge of optimizing for energy efficiency requires an orchestrated approach across different components of the infrastructure. Power and energy management needs to happen on a node level, job/task level and cluster/cloud level using various level schedulers all working in concert. Additionally, wide adoption of heterogeneous systems and evolving requirements for complex workflows present additional challenges to the performance and energy efficiency optimizations. Finally, there are efforts of augmenting schedulers by insights gained from data centre level telemetry data using cognitive analytics techniques.

We are presenting our approach to energy/power management, which can be described as Energy Aware Scheduling (EAS) and is illustrated in Fig. 1. EAS uses performance and power consumption models and software/hardware co-design for implementing various energy/power aware scheduling policies at the node, job and cluster levels.

II. IMPLEMENTATION

In this section we describe recent and current research activities in support of the EAS vision.

A. Cluster Level Policies

We have investigated and implemented a number of EAS policies in the IBM Spectrum LSF [4] batch scheduler. We have studied effects of the power management policies on supercomputer efficiency and power consumption using experimental as well as simulated data from scientific workloads on the BlueWonder supercomputer located at the Hartree Centre. Fig. 2 depicts simulated supercomputer power consumption with various energy budget scheduling policies.

We have observed energy saving of up to 12% [5]. Team from Leibniz Supercomputing Centre (LRZ) reported average savings 6%–8% on the SuperMUC supercomputer using the same prediction model implemented in IBM Load Leveler scheduler [6].

While allowing for respectful power savings, the employed performance and power prediction model had a number of limitations and was not able to maintain high accuracy of predictions over different workloads and microarchitectures. Some of these limitations have been addressed by a new model, which is described in the next section.

B. Power and Performance Prediction Models

We have further developed power and performance prediction models across different hardware microarchitectures using neural networks (NN) based models [7]. We have shown that NN based models improve accuracy of predictions. Fig. 3 illustrates improvement in accuracy of predicted power consumption of a neural networks based model compared to a linear regression based model. We are currently working on expanding NN based models to a broader range of workflows and heterogeneous architectures.

C. Job Level Policies

In our current implementation cluster level EAS policies are applied during scheduling phase before jobs are dispatched for execution. There is no feedback loop between scheduler and job during run time. Therefore certain opportunities for optimizing performance and power consumption are missed. To close the gap we need additional tools acting on a job level. Our research on job level scheduling policies is based

Energy Aware Scheduling

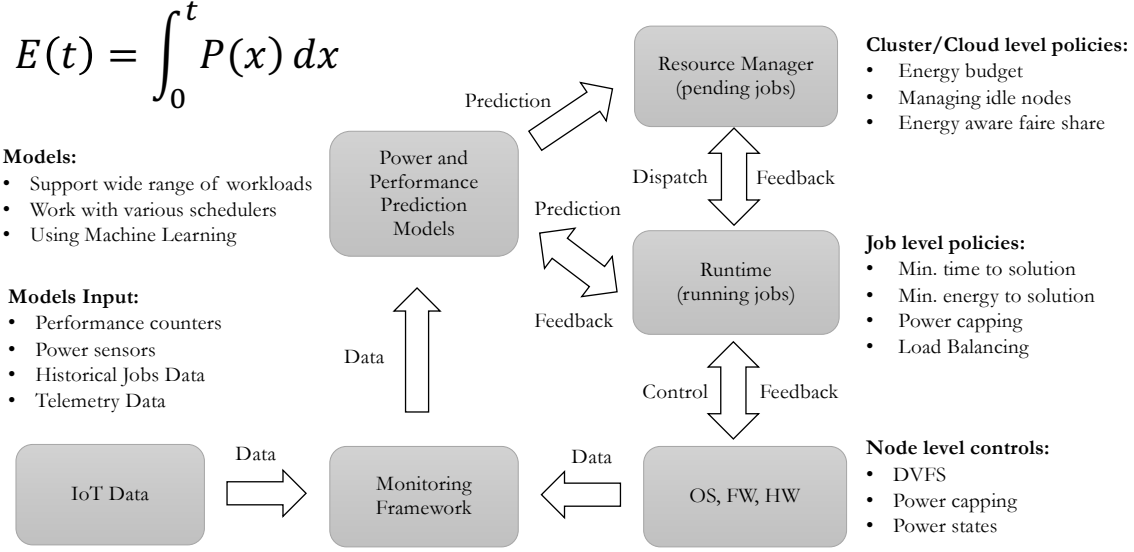


Fig. 1. Energy Aware Scheduling as a control loop across all components of the HPC systems software and hardware stack

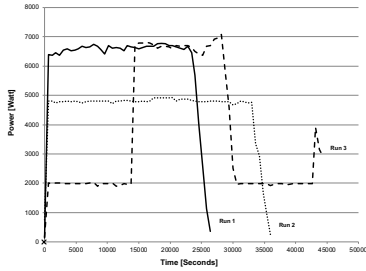


Fig. 2. Cluster power consumption under different energy budget policies.

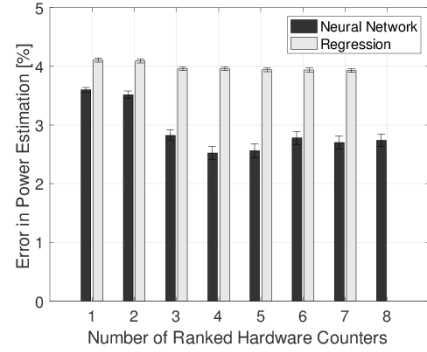


Fig. 3. Error in power estimation between linear regression and neural network based models on IBM® POWER™ microarchitecture.

on the open source Global Extensible Open Power Manager (GEOPM) framework, which allows for rapid prototyping of various power and performance optimization strategies for exascale workloads [8]. We have ported GEOPM to the OpenPower architecture [9] and are currently investigating benefits of GEOPM for various scientific applications on IBM POWER platform. GEOPM allows for finer grain dynamic control over application life time. It employs various hardware knobs and dials like power capping and dynamic voltage and frequency scaling (DVFS) to carry out different optimization strategies.

GEOPM can be integrated with higher level resource managers, thus allowing to coordinate job level and cluster level

EAS policies. As a proof of concept, we are looking into integration of GEOPM with IBM Spectrum LSF.

At present GEOPM is targeting MPI workloads, but generalizing to a broader range of workloads is possible.

D. Monitoring Framework

In addition to EAS policies and GEOPM, we are also developing an intelligent monitoring framework, which will allow us to collect additional power and performance metrics from compute nodes, storage and network subsystems as well as from IoT devices deployed in the data centre. Once obtained, such information can be correlated with workloads stats from a resource manager and analysed for developing

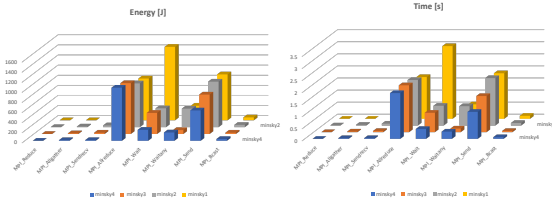


Fig. 4. Example of power and performance data from MPI regions collected by GEOPM.

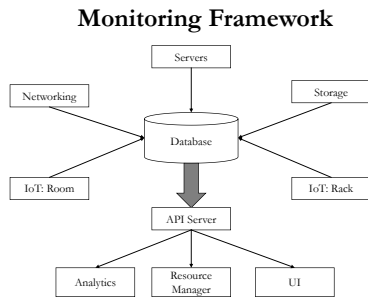


Fig. 5. Intelligent monitoring framework for EAS.

new EAS policies. Also, such framework can be used for cooling optimizations [10] and for anomalies detection [11]. High level block diagram of the monitoring framework is depicted in Fig. 5.

III. CONCLUSIONS

We presented our vision of the Holistic Approach to Energy and Power Management in HPC, which we describe as Energy Aware Scheduling (EAS). EAS uses performance and power consumption models and software/hardware co-design for optimizing energy efficiency policies at a node, job and cluster levels. We described our previous and ongoing research at different levels of the EAS stack: scheduling, runtime management, monitoring framework, performance and power prediction models.

Up until now our major focus was on homogenous HPC systems and MPI applications. We are now extending our research to more complicated data centric workflows and heterogeneous systems. We are also looking into Cloud environments, which present additional challenges related to multi tenant model of resources consumption and different scheduling paradigms [12].

IV. ACKNOWLEDGEMENTS

This work was supported by the STFC Hartree Centres Innovation Return on Research programme, funded by the

REFERENCES

- [1] THE GREEN500, “The Green Lists,” <https://www.top500.org/green500/lists/2017/11>, 2017, accessed: 2018-08-31.
- [2] “Scientific Grand Challenges: Architectures and Technology for Extreme Scale Computing - December 8-10, 2009, San Diego, CA. U.S. Department of Energy, Office of Science, Washington, D.C.”
- [3] THE HPC POWERSTACK, “The HPC PowerStack,” <https://powerstack.lrr.in.tum.de/>, 2019, accessed: 2019-03-10.
- [4] IBM, “Spectrum LSF,” <https://www.ibm.com/uk-en/marketplace/hpc-workload-management>, 2018, accessed: 2018-08-31.
- [5] V. Elisseev, J. Baker, N. Morgan, L. Brochard, and T. Hewitt, “Energy Aware Scheduling Study on BlueWonder,” in *4th International Workshop on Energy Efficient Supercomputing, E2SC@SC 2016, Salt Lake City, UT, USA, November 14, 2016*, 2016, pp. 61–68.
- [6] A. Auweter, A. Bode, M. Brehm, L. Brochard, N. Hammer, H. Huber, R. Panda, F. Thomas, and T. Wilde, “A Case Study of Energy Aware Scheduling on SuperMUC,” in *Supercomputing - 29th International Conference, ISC 2014, Leipzig, Germany, June 22-26, 2014. Proceedings*, 2014, pp. 394–409.
- [7] V. E. M. Puzovic, E.K Lee, “A Study on Cross-Architectural Modelling of Power Consumption Using Neural Networks,” *Supercomputing Frontiers and Innovations*, vol. 5, no. 4, pp. 24–41, 2018.
- [8] J. Eastep et al., “Global Extensible Open Power Manager: A Vehicle for HPC Community Collaboration on Co-Designed Energy Management Solutions,” in *ISC*, 2017.
- [9] M.Puzovic, V.Elisseev, K.Jordan, “Improving Performance and Energy Efficiency on OpenPower Systems Using Scalable Hardware-Software Co-Design,” in *High Performance Computing, ISC High Performance 2018 International Workshops, Frankfurt/Main, Germany, Revised Selected Papers*, 2018.
- [10] B. Acun, E. K. Lee, Y. Park, and L. V. Kalé, “Neural Network-Based Task Scheduling with Preemptive Fan Control,” in *International Workshop on Energy Efficient Supercomputing (E2SC)*. ACM, 2016.
- [11] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, and L. Benini, “Anomaly detection using autoencoders in high performance computing systems,” *CoRR*, vol. abs/1811.05269, 2018. [Online]. Available: <http://arxiv.org/abs/1811.05269>
- [12] KUBERNETES, “Kubernetes,” <https://kubernetes.io/>, 2018, accessed: 2018-08-31.

A Collaborative Cloud-based FR Approach for Humanoid robots

1st Hamza Aagela

School of Computing and Engineering
University of Huddersfield
Huddersfield, United Kingdom
Hamza.aagela@hud.ac.uk

2nd Violeta Holmes

School of Computing and Engineering
University of Huddersfield
Huddersfield, United Kingdom
v.holmes@hud.ac.uk

Abstract—The ability to recognize human faces in real time is an important requirement for most humanoid robots. One of the challenges in face recognition (FR) applications is the time it takes a robot to search through a large dataset of known faces. As a database of known images is increasing, a robots ability to store and process the data in real time is decreasing. In this paper we present a new cloud-based FR algorithm which will enable faster processing of data in face detection and recognition by humanoid robots. An improvement in robots performance will be achieved by offloading storage and processing tasks from limited on-board robot resources to the resources in the cloud. In the case of multi-robot systems, their performance can be further improved through cloud-based collaborative learning and information sharing. We created a new dataset containing over 300 trained facial images of 10 people. The result shows that the proposed FR can achieve 83% accuracy rate and exceeds the local FR performance in terms of the response time which is slightly increased in comparison to local system performance which sees significant increase when the dataset size grows. The system proved its capability to share knowledge between robots in the same multi-robot environment.

Index Terms—Edge computing, Cloud robotics, Face recognition, ROS, Humanoid-robot, NAO.

I. INTRODUCTION

The robots are increasingly present in human environments where robots and people need to collaborate and exchange knowledge and information. However, the robots are increasingly required to work in the same environment with other robots and share some of the tasks such as navigation, mapping, and image processing. Therefore, collaborative learning [7] between the robots is becoming more important [14]. It will optimize the robots' performance and use of resources by reducing common task repetitions. Collaborative learning will help the robots to inherit the knowledge that have been acquired previously by other robots. One of the most important tasks for humanoid robots is face detection and recognition [3]. However, they require extensive computing capability and storage, but developing a robot with high processing and storage characteristics can be expensive. Collaboration between robots can overcome the limitation of tasks processed using on-board resources. The main task in FR applications is defining unique features of the components in a human face. There are several FR approaches which have been developed, such as linear discriminant analysis (LDA) [13] and principal component

analysis (PCA) [11], which are implemented in various studies. The performance of these methods is acceptable, yet, they have limitations of being computationally intensive [4]. The main focus of this work is to overcome these limitations and build a real-time FR cloud robotic application that will allow NAO humanoid robots, which have limited computational capacity and low storage, to recognize people's faces with acceptable response time and accuracy. In addition, the proposed system aims to support sharing the knowledge between robots in a collaborative learning environment. The system will reduce the computational complexity in the robot and offload computation and storage to a cloud Virtual Machine (VM) by utilizing the clone-base cloud robotic platform (CCRP) shown in Fig. 1. We propose a new Collaborative Cloud-based Face Recognition Approach, which can be used with NAO humanoid robots. The rest of the paper is organized as follows:

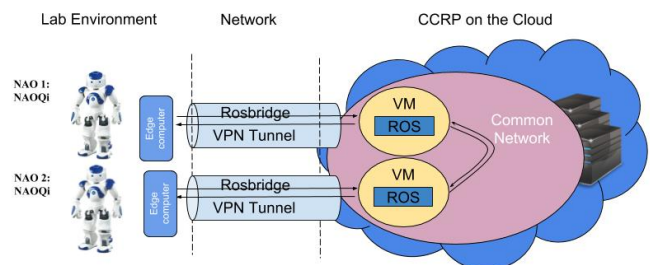


Fig. 1. CCRP Architecture and Methodology

Section 2 presents related work in the area of cloud robotics; Section 3 describes the architecture and methodology of the CCRP system; Section 4 defines the collaborative cloud-based face recognition algorithm; Sections 5 and 6 define the software environments, the experimental setup and presents the results of conducted experiments; Section 7 reports the analysis of the experimental results obtained in this research, and finally Section 8 concludes with the outcomes of our research and suggests possible future work.

II. RELATED WORK

The cloud robotic concept was introduced in mid 2009s by RoboEarth project. The main objective of the project was

to allow the robot to offload some of its computational tasks and provide a centralized knowledge-based environment that can help robots to share knowledge and collaborate with other robots in the system. The RoboEarth research group improved a number of cloud services and developed the cloud robotics network. [19]. [10] introduced Rapyuta, which is an extension of the RobotEarth, that installed Robot Operating System (ROS) in a virtual machine using a container. This established the connection with the robots over the websocket protocol, providing a full duplex connection link between the robots and the cloud. Authors in [18] developed a peer-based cloud robotic system which allows a single robot to recognize faces in a real-time by utilizing the intensive computational power of the cloud. The project used ROS as middleware and programming environment. However, a knowledge-sharing mechanism is not available in this system and there is a lack of published information on the performance of the system. Also, they used a peer-based cloud robotic model. A project called Cloudlet [17], which is a mobile fog computing system, allows a robot to send images to the cloud via a wireless connection established between the robots and a smartphone. [16] developed a mobile Cloudlet Cloud architecture, which simulates the process of offloading computationally intensive tasks to a number of cloud server VMs and the results show that the response time is reduced as the number of VMs increases.

III. CCRP ARCHITECTURE AND METHODOLOGY

In order to overcome shortcomings of the previous approaches a new CCRP platform is devised. As shown in Fig. 1, the architecture and methodology of the clone-based cloud robotic platform is designed to support a multi-robot system environment and is compatible with any robot supported by ROS. The CCRP is a cloud robotic solution that aims to provide a stable cloud robotic environment, and supports offloading of heavy computation over the network to the cloud. In addition, it provides a secure environment for accessing external resources and for a robot to collaborate and share knowledge with other robots. The connection between the robots and the clone instances is managed by OpenVPN software [8], [6], and rosbriidge [5] which establishes a secure tunnel between the robot edge node and its clone VM. Moreover, the addition virtual network created by the VPN will be used as a common network for a multi-ROS master that uses a ROS package called 'multimaster_fkie' [12], and the FR ROS package [20]. The performance of the system was evaluated by conducting the face detection and recognition in a case study using two NAO robots and a dataset stored on a universitys private OpenStack cloud.

IV. COLLABORATIVE CLOUD-BASED FR ALGORITHM

In this section we will describe our cloud-based FR algorithm shown in Fig. 2. The algorithm depends on moving the face detection and recognition ROS application to the cloud. The Humanoid robot NAO captures and streams the video data to the clone image on the CCRP platform. The algorithm has

several components, used for basic face awareness, face detection, and face recognition. These components are responsible for face detection and analysis. The Faceserver application in ROS will search through images, captured and streamed by a robot, in order to detect a face. It will compare it to the existing dataset stored in the cloud. If the face is known, the cloud sends back the personal details to the robot side; if the face is unknown, the algorithm starts a learning process, which will allow an operator to add a new record to the face dataset and start the learning mechanisms, which works by merging the newly-learned face to the existing records to be shared between robots. Each robot is capable of storing new faces in a cloud and sharing the data 'trained images' with other robots via the common network. In addition to capturing images, it is possible to send and process recorded video streams to the cloud. The FR algorithm was executed in 2D on the RGB face images captured by the NAO robot's main camera, with a Video Quality - KQVGA, which has resolution of 320x240. The video quality and frame rate, reported previously in our study [2], were used to enable an efficient data transfer.

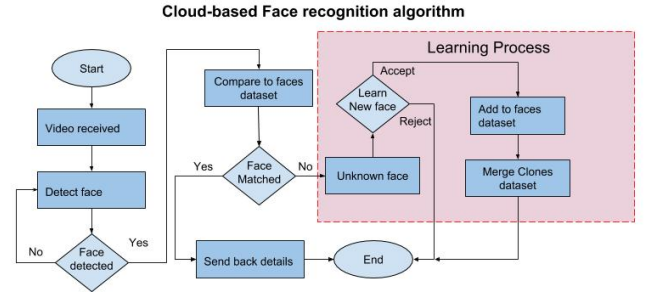


Fig. 2. The cloud-based face recognition algorithm

V. SYSTEM REQUIREMENTS

There are a number of hardware and software requirements needed to set up the environment for the multi-robot experiments. A cloud platform is needed to run the VMs, and host ROS stack middleware installed with a OpenVPN server. In the experiments we used private OpenStack cloud and two NAO humanoid robots.

A. Humanoid robot NAO

The humanoid robot NAO is one of the most popular educational robots available since 2008 [2]. As shown in figure 1 the design of a NAO robot resembles human appearance, weighing 4.5kg and is 0.5m tall. The NAO robot is fitted with a CPU GEODE 500 MHz, a set of sensors (ultrasonic and tactile) cameras, actuators (motors) for the joints, and indicators LEDs. Communication with NAO is established using Wi-Fi connection and the Ethernet network. The NAO robot is supported by ROS middleware which provides a naoqiridgemsgs package that is responsible for exchanging data between the NAO NAOQi 2.0 framework and ROS nodes. The package provides an access to all robot sensors, sending the commands to the actuators, reading sensors, and handling

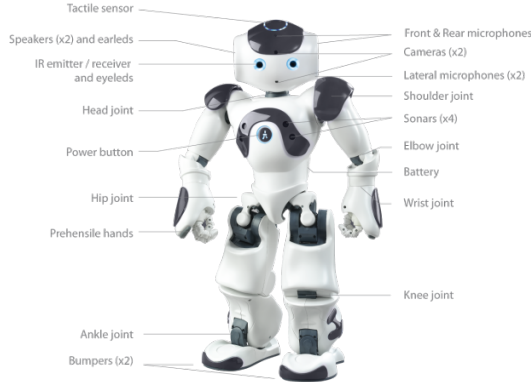


Fig. 3. Humanoid NAO robot [9]

Wi-Fi connections. NaoQis functions can be executed in C++, Python and Urbi [2].

B. Cloud platform "Openstack"

Cloud computing provides computing utility and storage capabilities as distributed resources, which are accessible remotely, such as Google Cloud, Amazon AWS or Microsoft Azure. Alternatively, a cloud environment can be deployed as a private cloud. In this research the OpenStack was used as a private cloud platform, deployed by using multi-Node Devstack installation. The system consists of one cloud controller and two compute and storage nodes, as shown in Fig 4.



Fig. 4. OpenStack Private Cloud System

C. Robot Operating System (ROS)

The ROS is an open-source middleware that provides a number of packages and software which support robot functions [15]. The ROS environment manage communication between the ROS nodes and the ROS master. In the ROS environment the sensors and actuators can be represented as topics and communication between them is managed by a master node. [1].

VI. EXPERIMENT SETUP

This section describes the setup of the experiment. University's private cloud was deployed using OpenStack. Two VMs were created on the cloud with the following infrastructure: Ubuntu 16.04 Operation system, 4 processing cores, 8 GByte of RAM and 100 GByte of hard drive storage. On the robots'

TABLE I
THE RESULT OF THE LOCAL FR APPROACH SHOWS THE ACCURACY AND FAILURE RATE.

No. of trained images	Confidence	Failure rate (%)
1	0.21	55%
5	0.33	35%
10	0.69	15%
20	0.76	0%
30	0.82	0%

side, two laptops with Ubuntu 16.04 were used to act as the edge computing systems for two NAO robots. The CCRP system was installed and configured in the cloud instances and robots' edge systems.

All the FR experiments have been done with the same environmental characteristics, such as the room lighting and the distance of the individuals' faces to the robots, because any differences in the setup can have an impact on the results of experiments. The distance between the robot and the person's face was about 50 to 70 cm. In the experiments, each robot captured face images of 5 persons (not just basic features), and the system synchronized and combined the data to be shared between the robots. The data gathered by the robots was stored in the ROS masters, leading to creation of over 300 images taken from 10 individuals. The experiments were conducted in two scenarios, first performing face recognition using local ROS environment (on the edge devices), and the second by using our cloud-based approach. The comparison between these two scenarios will determine the impact of moving the process of face recognition to the cloud, and will examine the difference in the algorithms' accuracy and the response time.

VII. EXPERIMENTAL RESULT

In this section we will examine the results of moving the FR task to the cloud in a multi-robot system. The NAO robots in our system are capable of acquiring knowledge of new faces, and recognizing faces of individuals in real time using created datasets, and sharing the knowledge of known faces among other robots, using ROS services deployed on OpenStack cloud. The results of the FR tasks in the multi-robot system with local ROS environments are shown in Table I. It is evident that the level of the accuracy increases when the number of the trained images increases, whilst the failure rate decreases. The results of the cloud-based FR approach are shown in Fig II. The accuracy of FR is very similar to that obtained using local (edge) FR. Therefore, the new cloud-based approach demonstrates that it is possible to perform FR tasks in the cloud without affecting the accuracy of the results. Each of the results presented in the tables is an average of 20 attempts in FR for a given scenario. The response times in FR using local (edge) and cloud approaches are shown in Fig 5. The response time using the local robot's resources is increasing as the number of image increases, from approximately 200ms with 1 trained image to more than 400ms with 30 trained images for 10 individuals. However,

TABLE II
THE RESULT OF THE CLOUD-BASED FR APPROACH SHOWS THE
ACCURACY AND FAILURE RATE.

No. of trained images	Confidence	Failure rate (%)
1	0.18	65%
5	0.32	40%%
10	0.63	15%
20	0.77	5%
30	0.83	0%

our cloud-based approach shows a smaller increase in the response time, from around 250ms to just above 300 ms, when processing 30 trained images. The cloud-based FR system outperforms the local FR system when there are more than 20 images per individual in the dataset. The proposed cloud-based system has an additional communication delay at the beginning of FR, due to establishing communication with a cloud, hence initially, it shows lower performance. In sharing the data between the robots, the exchange of the trained images was done successfully, updating the data every 5 minutes.

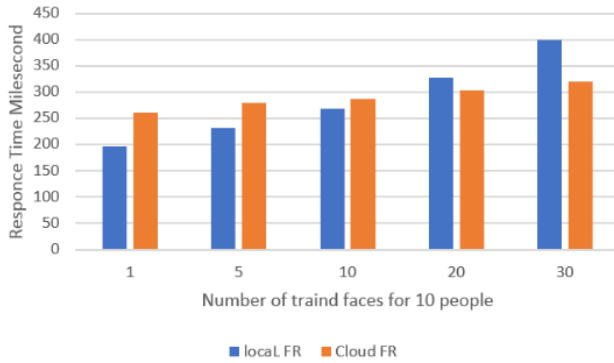


Fig. 5. The response time of the local FR and the cloud FR

VIII. CONCLUSION

This paper presented research work aimed to reduce the computational complexity in the robot system and to offload processing to the cloud VMs by utilizing clone-base cloud robotic platform (CCRP). We propose a new Collaborative Cloud-based Face Recognition Approach, which is implemented on the NAO humanoid robots, but can be deployed in other robots that support ROS and have the ability to capture images and video data. As the results section shows, the platform proved its capability in running the face detection and recognition tasks effectively and it exceeded the performance of the local system with limited computational power and storage. The robot (edge) solution will be less effective as the dataset increases, leading to longer response times. The new approach is also facilitating knowledge sharing between the robots without any difficulties, and is allowing the new learned facial images, captured by any robot in the system, to be shared with other robots within the same ROS multi-master environment. In future, further experiments will be conducted, using larger datasets, in order to verify and validate the

CCRP approach in the case of big data. Also, to improve the process of knowledge sharing among robots, an autonomous mechanisms will be provided to initiate an update of the dataset only when new trained data is available.

ACKNOWLEDGMENT

We would like to acknowledge the contribution of the HPC research group at the University of Huddersfield for providing the resources for this study.

REFERENCES

- [1] Aagela, H., Al-Nesf, M., Holmes, V.: An asus_xtion_probased indoor mapping using a raspberry pi with turtlebot robot turtlebot robot. In: Automation and Computing (ICAC), 2017 23rd International Conference on. pp. 1–5. IEEE (2017)
- [2] Aagela, H., Holmes, V., Dhimish, M., Wilson, D.: Impact of video streaming quality on bandwidth in humanoid robot nao connected to the cloud. In: Proceedings of the Second International Conference on Internet of things and Cloud Computing. p. 134. ACM (2017)
- [3] Abbas Shangari, T., Sadeghnejad, S., Baltes, J.: Importance of humanoid robot detection. *Humanoid Robotics: A Reference* pp. 1–9 (2016)
- [4] Bolotnikova, A., Demirel, H., Anbarjafari, G.: Real-time ensemble based face recognition system for nao humanoids using local binary pattern. *Analog Integrated Circuits and Signal Processing* **92**(3), 467–475 (Sep 2017). <https://doi.org/10.1007/s10470-017-1006-3>
- [5] Crick, C., Jay, G., Osentoski, S., Pitzer, B., Jenkins, O.C.: Rosbridge: Ros for non-ros users. In: *Robotics Research*, pp. 493–504. Springer (2017)
- [6] Crist, E.F., Keijser, J.J.: *Mastering OpenVPN*. Packt Publishing Ltd (2015)
- [7] Dillenbourg, P.: What do you mean by collaborative learning? (1999)
- [8] Feilner, M.: *OpenVPN: Building and integrating virtual private networks*. Packt Publishing Ltd (2006)
- [9] Francisco Miguel, R.M.: *Frivas pfc itis* (2013), <http://jderobot.org/Frivas-pfc-itis>
- [10] Hunziker, D., Gajamohan, M., Waibel, M., D'Andrea, R.: Rapyuta: The robearth cloud engine. In: *ICRA*. pp. 438–444. Citeseer (2013)
- [11] Ismail, L., Shamsuddin, S., Yussof, H., Hashim, H., Bahari, S., Jaafar, A., Zahari, I.: Face detection technique of humanoid robot nao for application in robotic assistive therapy. In: *2011 IEEE International Conference on Control System, Computing and Engineering*. pp. 517–521. IEEE (2011)
- [12] Juan, S.H., Cotarelo, F.H.: Multi-master ros systems. *Institut de Robotics and Industrial Informatics* pp. 1–18 (2015)
- [13] Li, M., Yuan, B.: 2d-lda: A statistical linear discriminant analysis for image matrix. *Pattern Recognition Letters* **26**(5), 527–532 (2005)
- [14] Ososky, S., Schuster, D., Jentsch, F., Fiore, S., Shumaker, R., Lebiere, C., Kurup, U., Oh, J., Stentz, A.: The importance of shared mental models and shared situation awareness for transforming robots from tools to teammates. In: *Unmanned Systems Technology XIV*. vol. 8387, p. 838710. International Society for Optics and Photonics (2012)
- [15] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: *ICRA workshop on open source software*. p. 5. No. 3.2, Kobe, Japan (2009)
- [16] Soyata, T., Muraleedharan, R., Funai, C., Kwon, M., Heinzelman, W.: Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. In: *2012 IEEE symposium on computers and communications (ISCC)*. pp. 000059–000066. IEEE (2012)
- [17] Stojmenovic, I.: Fog computing: A cloud to the ground support for smart things and machine-to-machine networks. In: *2014 Australasian Telecommunication Networks and Applications Conference (ATNAC)*. pp. 117–122. IEEE (2014)
- [18] Tian, S., Saitov, D., Lee, S.G.: Cloud robot with real-time face recognition ability. *Adv. Sci. Technol. Lett* **51**, 77–80 (2014)
- [19] Waibel, M., Beetz, M., Civera, J., d'Andrea, R., Elfving, J., Galvez-Lopez, D., Häussermann, K., Janssen, R., Montiel, J., Perzylo, A., et al.: Roboearth. *IEEE Robotics & Automation Magazine* **18**(2), 69–82 (2011)
- [20] Ziafati, P.: Face recognition ros wiki (2015), http://wiki.ros.org/face_recognition



POSTERS

Black-Box/Tracking System for Drones using LoRa

Alexandros Antoniadis, Hamza Aagela, Violeta Holmes

University of Huddersfield, School of Computing and Engineering



Abstract

This Drone Black-Box/ Tracking (DBB) system will be able to track the drones in flight by transmitting the geolocation data through the Low Power Wide Area Network protocol, LoRa™ to an operator's controller and/or to the cloud (Figures 1 and 2). It will enable the operator to choose if the drone controller can be used as a LoRa Client-Server for point-to-point communication (managed), as private connection, or as a LoRaWAN gateway (unmanaged), uploading the drones' sensor data to a cloud service such as TheThingsNetwork(TTN) or Cayenne LPP [2].

Research Objective

The objective of this project is to design and test a custom LoRa single-channel gateway which operates as a Black-Box transceiver when exchanging data. The gateway signal is to be analysed on a Spectrum Analyser to determine its power. The gateway should be able to exchange data with the Black-Box, performing a handshake within a specific interval to keep track of its status. TTN should also be used to confirm that the data received from the Black-Box are being pushed to the cloud on demand .

System requirements

Table 1 – System Requirements

Hardware	Tools
Heltec-ESP32-WIFI-LoRa Development Board	Arduino IDE
128x64 OLED Display	Atmel Studio 7.0
868 MHz SMA - Antenna	HMS-X Spectrum Analyser

Methodology

A proof-of-concept single-channel gateway was created, using an ESP32 microcontroller on a Heltec-WIFI-LoRa development board and an 868MHz omni-directional antenna (Figure 3). Code in C++ was designed to integrate AES256 Encryption, featuring End-to-End encoding/decoding . The gateway initialises the LoRa radio and starts "listening" to the LoRa channel, storing the data locally. If a network connection (over Wi-Fi) is present, the gateway forwards the data to the cloud. A spectrum analyser was used to measure the output characteristics of the antenna.



Figure 1 - DBB System Methodology

Systems Architecture

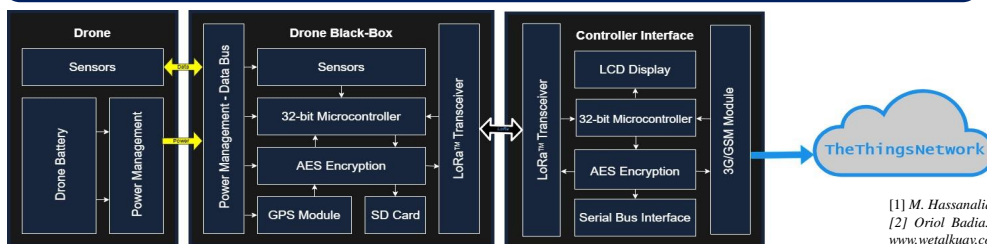


Figure 2 – Complete system architecture

Results



Figure 3 – Custom LoRa Gateway

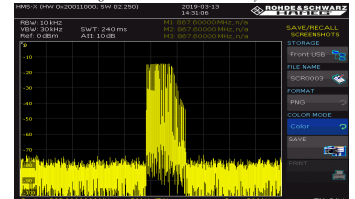


Figure 4 – Spectrum Analyser Measurements

Conclusions

The DBB device performed well in testing. The signal from the gateway was measured on a Spectrum Analyser (Figure 4). The screenshot displays the LoRa protocols FSK modulating nature, taken on the peak of the FSK shift. The packets received from the node were encrypted and encoded to JSON format to be pushed to the cloud.

Future Work includes: Swarm Robotics, Internet of Things, Low-Power Digital Signal Processing

References

- [1] M. Hassanalian, A. A. (2017). Classifications, applications, and design challenges of drones: A review.
- [2] Oriol Badia. (2016, November 10). Drone Black Boxes- Why do we need them? Retrieved from [www.wetalkuav.com: https://www.wetalkuav.com/drone-black-boxes/](https://www.wetalkuav.com/drone-black-boxes/)
- [3] Antoniadis, Alexandros. "SouthernEclipse/Encrypted-Single-Channel-Heltec-LoRa-Gateway". Github, 2019. <https://github.com/SouthernEclipse/Encrypted-Single-Channel-Heltec-LoRa-Gateway>.

SECURITY ORCHESTRATION, AUTOMATION AND RESPONSE (SOAR) IN HPC

Taha Al-Jody
taha.al-jody@hud.ac.uk
https://ta3design.com

Violeta Holmes
v.holmes@hud.ac.uk

ABSTRACT

Cybersecurity is becoming critical for many information and computing systems. In addition to standard security issues, there are additional requirements for secure access to High-Performance Computing (HPC) systems (1).

AIMS

- Security Orchestration Automation and Response (SOAR) system.
- Address current security issues in HPC systems.
- Based on familiar browser elements
- Supports secure operation and access to HPC systems without a need for specialist user training.
- User registration and authorisation based on two-factor authentication e.g time-based one-time password or/and universal second factor (U2F).
- Active monitoring, analysis and interpretation of data
- Continuous supervision of users' activity, server's requests, devices and geographic locations.

MOTIVATION

- The user management of HPC systems is often an add-on to existing institutional policies that do not take into account the need to access many different HPC systems.
- There are no common HPC security standards in research and education communities. Every institution has its own policy, often certified by the e-science certificate authority (CA).
- Most registered users use Simply Secure Shell (SSH) to access given HPC resource.
- The systems used to access HPC clusters are frequently insecure, such as Putty, or they are being used in unprotected networks.
- Currently, many HPC centres make use of Private Key Infrastructure (PKI) as a basic method of authentication. even with the PKI authentication in place, there are potential security issues for users who have minimal security awareness.

SOAR

Machine learning technology can offer solution to cybersecurity in HPC as part of SOAR. Component of the this system are:

- File Manager and Terminal (Fig 2 and 3).
- Access (Login and Registration).

Figure 1 shows the current architecture of the system.

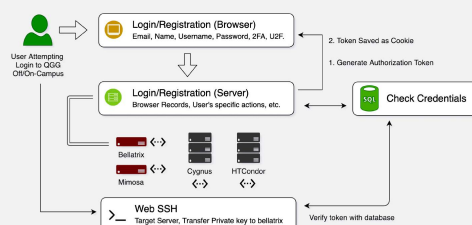


Fig 1 - System's Architecture

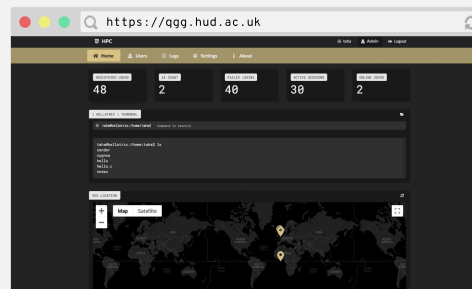


Fig 2 - Administrative Dashboard



Fig 3 - Web File Manager

TESTING

- The system was tested on Queens Gate Grid (OGG) system by administrators and student users at the University of Huddersfield.
- Initial administrator and user feedback is encouraging.
- From an administrative perspective, detailed information is available to detect possible security threats and to act promptly (Fig 4 and 5).
- From the users' perspective, the system is easy to learn and use.
- It provides repeatedly reliable access to resources, submitting and running jobs, and managing files (Fig 3).
- Current development of Artificial Intelligence system are being tested on the current users and administrators.

Time	Type	Comment
2019-01-12 14:01:00	File Manager	[Username] Command "tar" (Security) Both Manipulation: home/taha [Success] [Home/Taha]
2019-01-18 00:00:00	Terminal	[Username] Command "tar" Sent to "ballatrix" [Success] Command Run
2019-01-18 12:00:14	System	Refresh User
2019-01-18 12:00:14	System	Created User "VIOLETA"
2019-01-18 12:00:14	User	Deleted "VIO"

Fig 4 - User's Activity

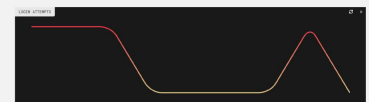


Fig 5 - User's Login Attempts

FUTURE WORK

- Future work will focus on:
- 1- Deploying Machine Learning and pattern recognition in SOAR to further improve HPC security.

REFERENCES

- 1- HPC Wire, Where Security Meets High Performance Computing, 2017

High End Compute

HEC can help you, whatever your level of experience:

- **reduce time-to-solution** by use of parallel programming & the best use of available hardware (multicore desktop, cloud, or supercomputing) & porting code to run well on new architectures (FPGA, GPU, Xeon Phi)
- Using the code we have ported to step you through so you understand how to apply the approach to your future codes and new problems
- **reduce “energy to solution”** by careful choices of code changes and use of appropriate tech. As well as being part of corporate social responsibility, this approach lowers energy bills, providing a greater return on investment.
- **sharing the skills** involved, by leading the client through the above steps and by bespoke training blending elements of online and face-to-face training your own codes. To **ensure efficient upskilling** HEC has developed its unique “follow up” approach with a visit to discuss options to further improve your application of the solutions provided.

Dr. David Topping, a UK National Centre for Atmospheric Sciences (NCAS) Fellow, says, “Working with HEC has enabled me to expand into areas that would otherwise be unavailable. Not only has Michael delivered technically, from research grade work through to outreach activities with schools, but also offered guidance on emerging roadmaps and where best to place myself to exploit opportunities. I would highly recommend working with HEC on all compute related matters.”

Paul Popelier, Professor of Computational Chemistry in the Manchester Institute of Biotechnology, noted of work that has realised over 50x speed up in each element (FEREBUS, MORFI & FFLUX) of their quantum chemical topology modelling, “Thanks to the work and support of HIGH END COMPUTE LTD, our homemade software is now highly parallelized and much less memory hungry. Thanks to these improvements, dispersion energies for much larger systems can be assessed and we are one step closer to our final goal: highly realistic biomolecular simulations using dispersion forces obtained through first principles.”



HEC has designed and run STEM outreach activities, including schoolchildren getting to grips with parallel programming using MPI on a cluster they built themselves using Raspberry Pi cards.

Proven experience in helping others to knowingly accelerate their research and R&D, across traditional and emerging technologies

- Member of the High Performance Computing Advisory Council
- Chair of Emerging Tech (EMiT) conference series; innovator of UoM GPU Club
- Author/coauthor on 18 papers; grants & journal reviewer; invited speaker at EuroPar and Cray User Group
- Excellent working relationships (& access to prototypes and experts) with key vendors: AMD, ARM, Atos, Intel, Maxeler, NVIDIA & Xilinx



SCAN ME!

<https://highendcompute.co.uk>



TWO WAY RANGING USING MM-WAVES FOR FUTURE WIRELESS NETWORKS

Adnan Farooq^a, Qasim Zeeshan Ahmed^a, Faheem Khan^a, & Temitope Alade^b

a. School of Computing and Engineering, University of Huddersfield, United Kingdom
b. Business School, University of Worcester, United Kingdom

Project Background

Mm-Waves are the future generation of wireless technology, using the spectrum ranging from 30GHz to 300GHz. Bands in this frequency have been used previously for small backhaul applications for small ranges, as long ranges can effect the signal and the signal strength. Objects such as trees and buildings can block these signals. Therefore, millimetre waves can be a great use for indoor applications, thus the initiation of this project.

This project will explore mm-waves using an EVM produced by TI. This EVM can be taken as input, and a decision can be made upon the input given to the processor by the EVM.

The wavelength of mm-waves can be calculated using the formula below:

$$\lambda = \frac{c}{f} \text{ Where } c = \text{Speed of light } (3 \times 10^8) \text{ and } f = \text{Frequency}$$
$$\lambda = \frac{3 \times 10^8}{30 \times 10^9} = 10 \text{mm at } 30 \text{ GHz}$$
$$\lambda = \frac{3 \times 10^8}{300 \times 10^9} = 1 \text{mm at } 300 \text{ GHz}$$

Aim

This project will aim to research and explore mm-Waves in depth, and show how they can be applied to short range applications.

Objectives

- Literature review of mm-Waves
- Issues with mm-Waves at long distances
- Application of mm-Waves
- Simulation on prototype of designed application

Method and Approach

1. Research on different types of applications of mm-Waves.
2. Built Prototype by TI found, distance measurements made using the EVM
3. Further coding done to use the EVM as in input
4. Further applications using EVM explored

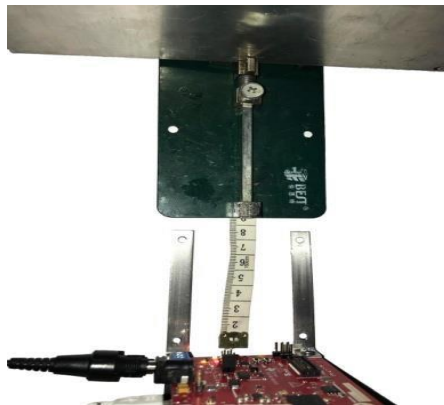


Figure 1: The setup used to make the distance measurements.

Results/Testing

With the EVM AWR1642, the initial testing was done by placing a metal sheet in front of the module. This sheet blocked the signal causing a Doppler effect. This effect is recognised by a program made by TI, which can then calculate which angle the object is at and how far it is.

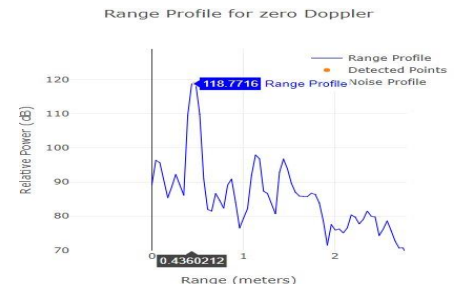


Figure 2: Range Profile Data

Figure 2 above is a capture from the demo visualizer, it shows that there is a peak power of 118.7dB at the distance 40CM. It shows that the distance is 43.6 CM, however this can be interpreted as an error. Distances from 20CM to 100CM were measured which all showed an error up to 6CM.

Further applications of the EVM are still under research. MATLAB will be used for these applications to process the data given by the EVM .

Conclusion

The tests showed there was a small error between the measured distances, however these errors can be eliminated by taking the SSE and subtracting it from the measured distances. Furthermore, the information found on the applications of mm-Waves shows that future wireless technology could be more advanced to the technology used currently. Significantly faster data transfer rates can be expected, which would result in a quicker response of a system, thus making vehicles much safer as well as general home security.

References

- J. I. R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems," IEEE Journal of Selected Topics in Signal Processing, vol. 10, no. 3, pp. 436–453, April 2016
- Texas Instruments Incorporated. (2019). MMWAVE-SDK 03_01_01_02 - TI.com. [online] Software-dl.ti.com. Available at: http://software-dl.ti.com/ra-processors/esd/MMWAVE-SDK/latest/index_FDS.html [Accessed 25 Jan. 2019]

Tunable Fault Tolerant Spiking Neural Networks on FPGAs



UNIVERSITY
of York



Ulster
University

Anju P. Johnson¹, Junxiu Liu², Alan G. Millard¹, Shvan Karim², Andy M. Tyrrell¹,
Jim Harkin², Jon Timmis¹, Liam McDaid² and David M. Halliday¹

¹Department of Electronic Engineering, University of York, York YO10 5DD, UK

²School of Computing, Engineering and Intelligent Systems, Ulster University, Derry BT48 7JL, UK



SPANER

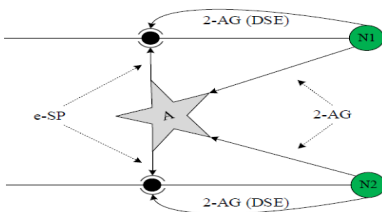
1: ABSTRACT

- Describes a novel methodology to address the problem of faulty synapses in Spiking Astrocyte Neural Networks (SANNs)
- Work is inspired by recovery processes in the brain
- A fault is considered as a reduction in synaptic transmission probability, leading to reduced spiking activity
- Field Programmable Gate Arrays (FPGAs) is used for demonstration
- Repair uses Dynamic Partial Reconfiguration (DPR) of the FPGAs
- Neuronal self-tuning is used to enhance repair capability

2: INTRODUCTION

- Homeostasis is the property of a biological system that helps maintain a constant internal environment
- For example, the nervous system monitors its activities and maintains its functional properties
- Astrocytes are star-shaped glial cells which coexist with neurons and regulate synaptic transmission
- Flexibility and rapid prototyping capabilities motivates the use of FPGAs in SANNs
- SRAM-based FPGAs are prone to hardware failures including Single Event Upsets
- DPR aims at modifying the existing circuit while other parts of the circuitry continue to function
- We use Dynamic clock alteration, a variant to the classical DPR

3: SANN NETWORK

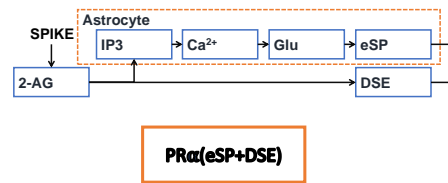


Signaling:

- DSE: Depolarization-induced Suppression Excitation (Direct Signaling)
- eSP: Endocannabinoid-mediated Synaptic Potentiation (Indirect Signaling)

4: ASTROCYTE PROCESS

Complete Model

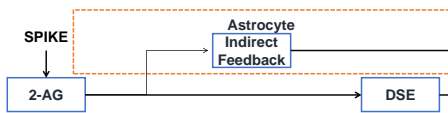


- e-SP and DSE are of opposite polarity

Repair Process

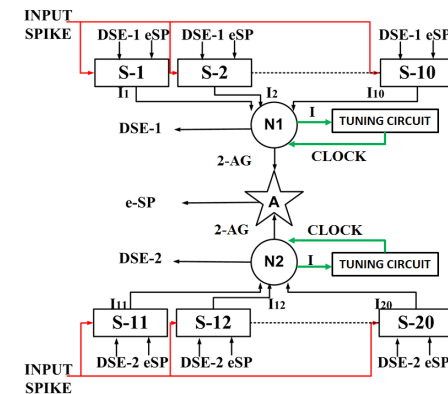
- When synapses fail DSE decreases
- e-SP via astrocytes increases PR
- This repairs the fault

Reduced Model



- Astrocyte process is reduced to a single differential equation

5: TUNABLE NEURON MODEL



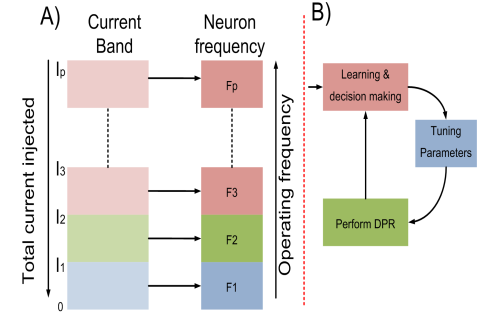
- Astrocyte based repair for faults <70%
- Neuronal self-tuning for faults >70%
- Neuron monitors synaptic input current
- Neuron parameters are adjusted
- Targeted Parameter - operating frequency
- This is established using Dynamic Partial Reconfiguration of FPGAs

synaptic fault %	Imax range	DCM		Frequency Neuron(MHz)
		M	D	
(0-70)%	(10. Isyn-4. Isyn)	2	2	100
(70-80)%	(4. Isyn-2. Isyn)	3	2	133
(80-100)%	(2. Isyn-0)	3	1	200

Isyn : Total synaptic current injected to the neuron

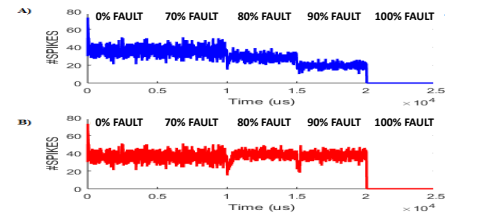
Isyn : Total synaptic current injected to the neuron

6: HOMEOSTATIC SELF-TUNING METHODOLOGY



7: EXPERIMENTAL RESULTS

Demonstration of Repair



HARDWARE UTILIZATION

Resource	Slice	Slice Reg	LUT	DSP	BUFG	DCM	PLL
Neuron Network	3139	1537	10403	20	0	0	0
Tuning Circuitry	26	36	37	0	9	2	1
Total	3165	1573	10440	20	9	2	1

PEARSON CORRELATION COEFFICIENT (spike Timing)

No fault vs 70% fault	No fault vs 80% fault	No fault vs 90% fault
0.999995	0.999995	0.999997

8: APPLICATIONS

Biomedical: Repairs faults in cardiac pacemakers such as over-sensing and under-sensing

Robotics: Regulates robot control signals working in noisy environments, with weak inputs

9: CONCLUSIONS

- The work gives insight into the capabilities of modern hardware to mimic brain-inspired homeostatic self-repair
- Future works:** Robots with self-repairing brains

ACKNOWLEDGEMENTS

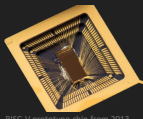
- Self-rePairing spiking Astrocyte Neural NETwork (SPANER) funded by EPSRC (EP/N007050/1, EP/N00714X/1)
- EPSRC platform grant(EP/K040820/1)

EPSRC
Engineering and Physical Sciences
Research Council



Background

- Traditionally microprocessors fall into two categories, **CISC** (Complex Instruction Set Computer) or **RISC** (Reduced Instruction Set Computer)
- Most CISC designs now utilise microoperations meaning they're essentially RISC designs at their lowest level
- More application specific microprocessor designs are needed to meet the needs of emerging technologies. However, economies of scale do not work in favour of these designs
- In an attempt to alleviate economies of scale RISC-V aims to replicate the Open Source model at the hardware level by providing a free and open **ISA** (Instruction Set Architecture)
- A proven method for gaining loyal community members is to use an open source project as an educational tool



RISC-V prototype chip from 2013



RISC-V development board 2018

Supervisor: Dr Violeta Holmes

Aim

The aim of this project is to design and implement a soft RISC-V core that could be used as an educational tool in computer architecture and organisation

Objectives

- ✓ Design a soft processor core capable of executing Integer Computation, Load/Store and Control Transfer instructions
- ✓ Implement the soft processor core using VHDL for use in FPGAs
- ✓ Test the implementation in simulation
- ✱ Test the implementation on an FPGA development board

Poster References

- RISC-V Foundation. (2017). Specifications. Retrieved from <https://riscv.org/specifications/>.
- Tilley, A.T. (2016.). This New Chip Startup Wants To Bring Open Source To A Stagnant Industry. Retrieved from <https://www.eetimes.com/this-new-chip-startup-wants-to-bring-open-source-to-a-stagnant-industry/>.
- Sifive. (N.d.). Boards & Software. Retrieved from <https://www.sifive.com/boards/>.

Instruction Set Architecture

The design adheres to a subset of the RV32I v2.0 base integer instruction set defined in the RISC-V user level specification. The main characteristics of this instruction set are:

- Fixed-length 32-bit instructions, aligned on a four-byte boundary in memory
- Has 31 general purpose registers x1-x31 with one register x0 hardwired to the constant 0
- Uniform decoding, split into four core formats (R/I/S/U) and two sub formats (B/J) which are variants of (S/U) respectively

31	20	15	12	8	7	6	5	4	3	2	1	0
opcode	rs2	rs1	funct3	rd	opcode	rs2	rs1	funct3	rd	opcode	rs2	rs1
imm[11:0]	rs2	rs1	funct3	rd	imm[11:0]	rs2	rs1	funct3	rd	imm[11:0]	rs2	rs1
imm[31]	imm[10:3]	imm[2:1]	imm[19:12]	rd	imm[31]	imm[10:3]	imm[2:1]	imm[19:12]	rd	imm[31]	imm[10:3]	imm[2:1]

- Little-Endian with the sign bit for all immediates held in bit 31 of the instruction, allowing sign-extension to proceed in parallel with instruction decoding

31	20	15	12	8	7	6	5	4	3	2	1	0
imm[31]	imm[30:25]	imm[24:19]	imm[18:13]	imm[12:7]	imm[6:1]	imm[31]	imm[30:25]	imm[24:19]	imm[18:13]	imm[12:7]	imm[6:1]	imm[31]
imm[31]	imm[30:25]	imm[24:19]	imm[18:13]	imm[12:7]	imm[6:1]	imm[31]	imm[30:25]	imm[24:19]	imm[18:13]	imm[12:7]	imm[6:1]	imm[31]

Author: Jack Parkinson

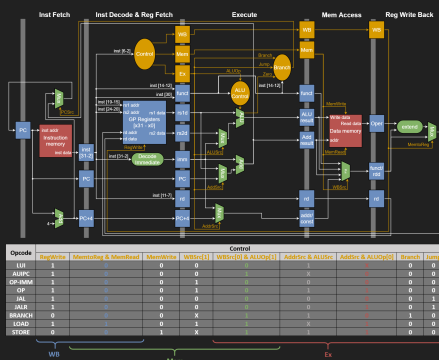
Student#: U1552044

RISC-V Implementation

Course: Electronic Engineering and Computer Systems BEng(Hons)

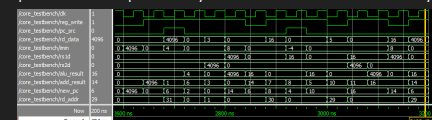
Core

- Critical path design that prioritised performance over size
- Classic 5-Stage RISC Pipeline
- Byte-aligned memory that supports misaligned access



Compiler & Results

Each VHDL entity was tested in RTL level simulation using custom testbenches also written in VHDL. To test the core as a whole, a testbench was created which acted as an assembly compiler and uploaded the compiled binary into the core's instruction memory



```

inst[31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[8, 7, 6, 5, 4, 3, 2, 1, 0]
inst[7, 6, 5, 4, 3, 2, 1, 0]
inst[6, 5, 4, 3, 2, 1, 0]
inst[5, 4, 3, 2, 1, 0]
inst[4, 3, 2, 1, 0]
inst[3, 2, 1, 0]
inst[2, 1, 0]
inst[1, 0]
inst[0]

```

A variety of assembly programs were run to test every instruction. This included one program which aimed to test every type of instruction acting on a single operand so any errors would affect the final result

Conclusion

The soft RISC-V core design manages to demonstrate many implementation techniques while still being relatively simple in its overall design. The VHDL implementation correctly executes all of the RV32I instructions it was designed to. The combination of the design, implementation and technical report could be used as an educational tool meaning this project can be deemed a success. However, given the additional time and resources it could greatly benefit from:

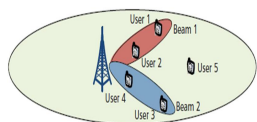
- Ensuring that the VHDL implementation can be used on FPGA by:
 - Completing timing simulation (50MHz target on Cyclone IV)
 - Testing on an FPGA development board
- Making the design and VHDL implementation compatible with existing open source compilers by:
 - Adding stall/flush logic & forwarding unit to the design to prevent data hazards
 - Adding logic to handle RISC-V hardware threads (harts) to the design, allowing it to execute the full RV32I instruction set
- Increasing the design's effectiveness as an education tool by breaking it up into smaller more easily digestible designs

Energy harvesting for advanced 5G wireless communication network IoT based storage devices

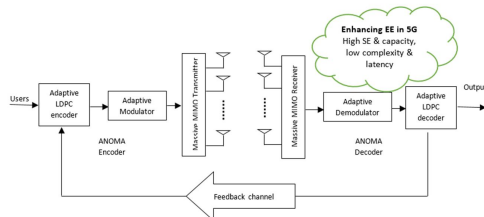
Introduction

Applications for wireless networks are growing despite the challenge of the energy costs. Further, the number of users is increasing within the basic wireless facilities which are depending on the energy efficient systems. Apparently, users need minimum energy cost influenced with the efficient design of wireless network such as multiple-input, multiple-output (MIMO). Analyzing the enhancement of energy efficiency (EE) through non-orthogonal multiple access (NOMA) scheme in wireless powered communication network (WPCN) a niche technique for Improving 5G system requirements and enhancing security.

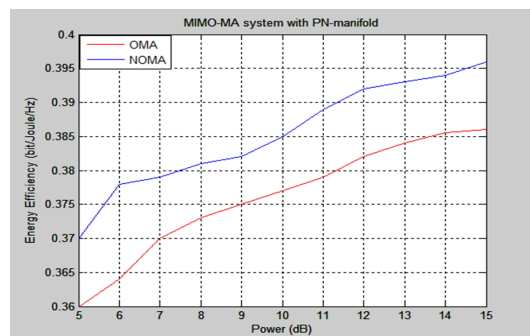
Development



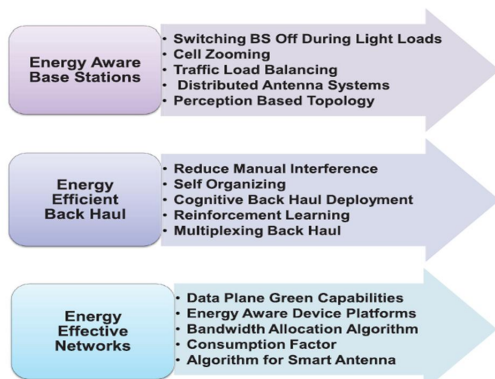
NOMA in Massive MIMO system



Downlink MIMO NOMA systems with manifolds (Pn-manifold)



EE in 5G wireless network



Open Research Problems

Analysis of EE with resource allocation under imperfect CSI.

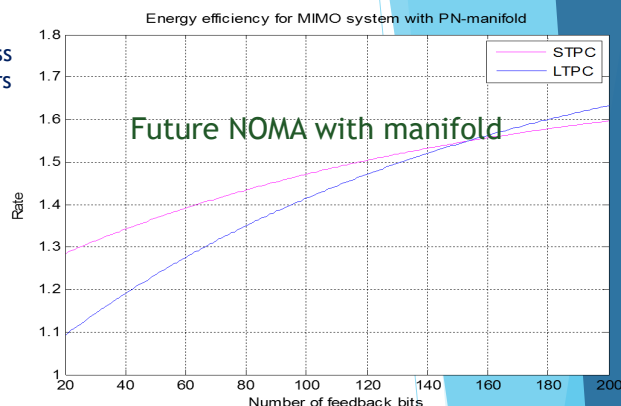
Analysis of EE with NOMA for the perfect CSI because perfect CSI is usually hard to obtain in fading channels
New network protocols using energy-efficient NOMA

Cyber security solution using software defined network (SDN), software define multiple access (SoDeMa) or SDMA

Security provisioning in NOMA

SDN provides simple abstractions to describe the components, the functions they provide, and the protocol to manage the forwarding plane from a remote controller via a secure channel. All layer security issues using SDN concepts.

Uses of NOMA for dynamic security in next-generation mobile networks.



The application of MIMO techniques to NOMA systems is important to enhance the performance gains of NOMA.

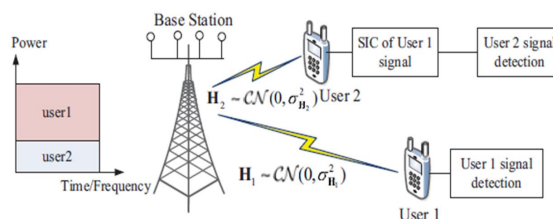
A general MIMO-NOMA framework is applicable to both downlink and uplink transmission

A larger diversity gain can be achieved, e.g., for a scenario in which all nodes are equipped with M antennas, a diversity order of M is achievable.

The MIMO-NOMA framework is more general, and also applicable to the case where the users have fewer antennas than the base station.

Downlink MIMO NOMA systems with statistical CSI

$$G = \frac{\text{MIMO}(N_r > 1)}{\text{MIMO}(N_r = 1)}$$



However, An important future direction is to study how MIMO-NOMA transmission can be realized with limited CSI feedback

in practice, the perfect CSI is usually hard to obtain in fading channels, the long term power control schemes (LTPC) with statistical CSI is preferred to reduce the CSI feedback overhead.

CONCLUSIONS

In this study, the future direction of renewable-energy systems based on a MIMO wireless network is analysed in various ways. Future energy systems and applications in the wireless environment depend on the optimized design of massive MIMO as the cost of energy is growing with a number of services. Energy harvesting for advanced 5G wireless communication network IoT based storage devices are exhibiting promising results and although the research is its infancy has potential for 5G implementations.

REFERENCES

1. J. Hoydis, S. T. Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks. How many antennas do we need?," IEEE Journal on Selected Areas in Communications, Vol. 31, 2013, pp. 160-171.
2. Alzaharani, Ahmed, and Vijey Thayanathan. "Analysis of Energy Efficiency for MIMO Wireless Network using Manifold Techniques." Journal of Information Science & Engineering 35, no. 2 (2019).
3. V. Thayanathan and F. M. Bahazq, "Analytical model of energy saving approach in a wireless sensor network," in Proceedings of IEEE 16th International Conference on Computer Modelling and Simulation, 2014, pp. 504-509.
4. Wang, Yingmin, Bin Ren, Shaohui Sun, Shaoli Kang, and Xinwei Yue. "Analysis of non-orthogonal multiple access for 5G." China Communications 13, no. 2 (2016): 52-66.

Dr Javad Yazdani

University of Central Lancashire, Faculty of Science and Technology, School of Engineering, Preston, Lancashire, PR1 2HE, UK

Dr Vijey Thayanathan

Department of computer science, King Abdulaziz University, Jeddah, KSA.

AUTHOR INDEX

- Aagela, H., 62, 72, 77
Ahmed, Q., 36, 52, 81
Aivaliotis, G., 58
Al-Jofy, T., 78
Al-Riyami, S., 25
Alade, T., 81
Alattal, H., 36
Antoniades, A., 77
Arinto, C., 79
Ashworth, M., 12
Attwood, A., 12
Audouin, Y., 41
Bane, M.K., 80
Bogdan, P.A., 20
Coenen, F., 25
Dal Palu, A., 45
Davidson, S., 20
Elisseev, V., 69
Elsaraf, Z., 52
Emerson, D., 41, 57
Farooq, A., 81
Furber, S., 20
Garcia, G.P., 20
Grasset, J., 41
Grassi, A., 66
Gu, X., 57
Halliday, D.M., 82
Harkin, J., 82
Hoefer, T., 10
Holmes, V., 28, 32, 62, 72, 77, 78
Hopkins, R., 20
James, R., 20
Johnson, A.P., 82
Karim, S., 82
Khan, F., 36, 52, 81
Kondratyuk, N., 46
Kynigos, M., 48
Lant, J., 16
Lisitsa, A., 25
Liu, J., 82
Longshaw, S., 41
Macfarlane, K., 28
Mawer, J., 12
McDaid, L., 82
Meng, J., 57
Millard, A.G., 82
Milutinovic, V., 9
Moulinec, C., 41
Navaridas, J., 16, 48
Palczewski, J., 58
Parkinson, J., 83
Pascual, J.A., 48
Qurashi, A.W., 32
Ragta, L., 57
Riley, G., 12
Seedall, M., 28
Smirnov, G., 46
Stegailov, V., 46
Thayananthan, V., 84
Timmis, J., 82
Timofeev, A., 46
Titarenko, S., 58
Titarenko, V., 58
Turchetto, M., 45
Tyrrell, A.M., 82
Vacondio, R., 45
Vallati, M., 66
Yazdani, J., 84

Copyright © Michael Bane, EMiT (Emerging Tech) Conference series

Published by
EMiT/University of Huddersfield/High End Compute Ltd/University of Manchester

Proceedings of the 2019 Emerging Technology Conference
9-11 April 2019, University of Huddersfield, Huddersfield, U.K.



ISBN: 978-0-9933426-4-6