Science & Technology Facilities Council

Accelerating performance of the HPC electron collisions R-matrix code PFARM on the Xeon Phi

AG Sunderland(1), G Corbett(1), M Lysaght(2) and M Plummer(1)

(1) Scientific Computing Department, STFC Daresbury Laboratory, UK

(2) Novel Technologies Activity and the Intel Parallel Computing Centre, ICHEC Dublin, Ireland

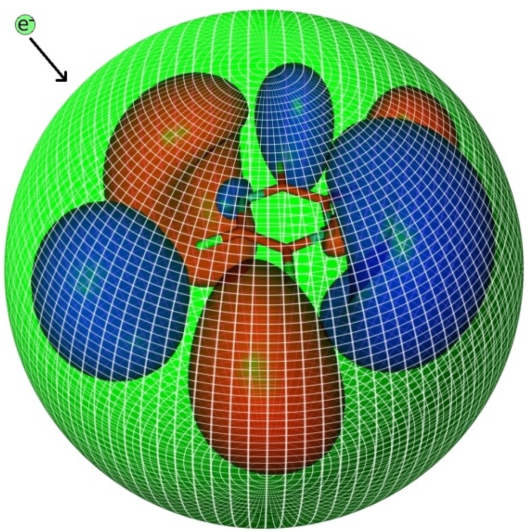
R-matrix Theory

- Basis of computer programs that describe a wide range of atomic, molecular and optical processes. Numerically very stable
- Ab initio solution of full Schrodinger equation using CI
- Successful in treating a wide range of collision phenomena
  - Scattering of electrons, positrons or photons with atomic and molecular targets
  - Multiphoton interactions with atoms (and now/soon molecules)
- The PFARM code developed for atoms has recently been adapted for molecular codes
- Developed by CCP2/CCPQ. Optimization projects - dCSE, PRACE.
- **Real world applications include:**
  - **Astrophysics: stars, interstellar medium (shocks)**
  - **Atmospheres, atomic and molecular plasmas (nuclear fusion, laser-produced plasmas, lighting)**
  - **radiation damage to DNA (electron collisions with DNA bases)**

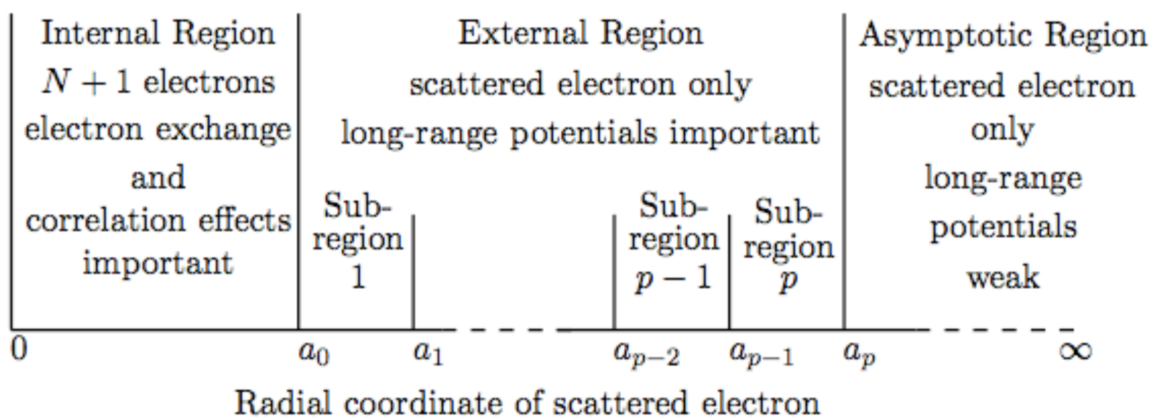
Science & Technology Facilities Council

The R-matrix method

•Configuration space divided into ‘inner’ and ‘outer’ regions by a sphere
•Inside: all electron (lepton) calculation, CI, exchange, spherical tensor algebra, Hamiltonian formation and diagonalization (with non-vanishing orbitals on the boundary)
•Outside: multipole potentials (from ‘inside’), coupled differential equations, propagation to asymptotic region, possible frame transformations
•Inside: energy-independent; outside: energy-dependent



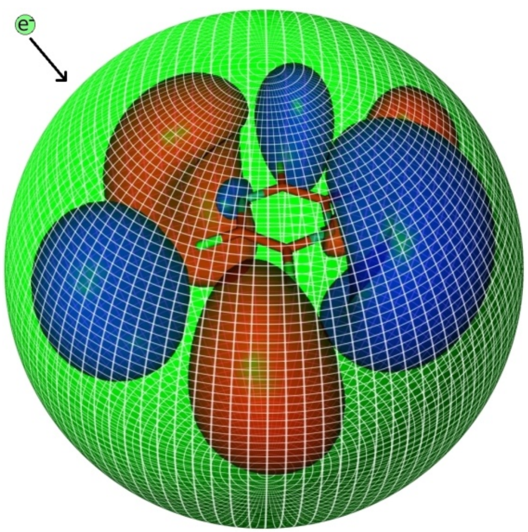
Partition of Configuration Space



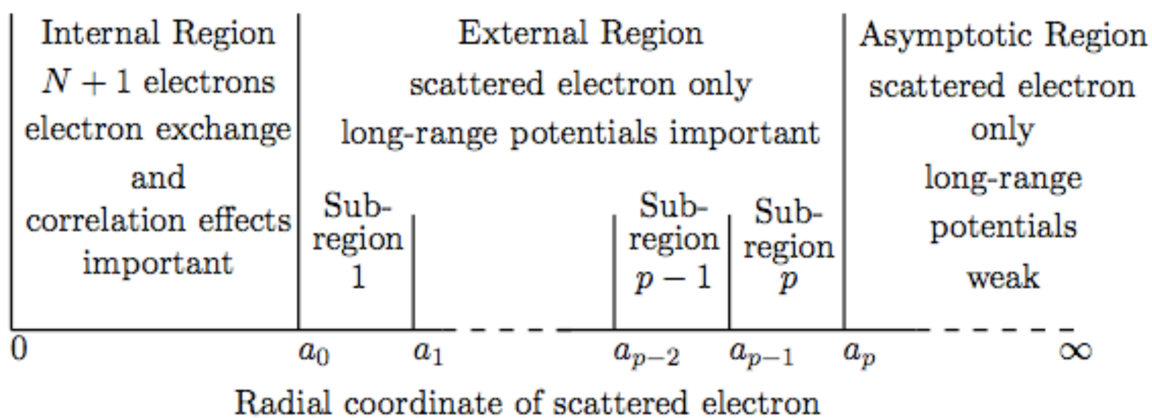
Science & Technology Facilities Council

The R-matrix method

- Configuration space divided into ‘inner’ and ‘outer’ regions by a sphere
- Inside: all electron (lepton) calculation, CI, exchange, spherical tensor algebra, Hamiltonian formation and diagonalization (with non-vanishing orbitals on the boundary)
- Outside: multipole potentials (from ‘inside’), coupled differential equations, propagation to asymptotic region, possible frame transformations
- Inside: energy-independent; outside: energy-dependent



Partition of Configuration Space



The parallelization of the code maps closely to this partitioning

PFARM: external and asymptotic regions

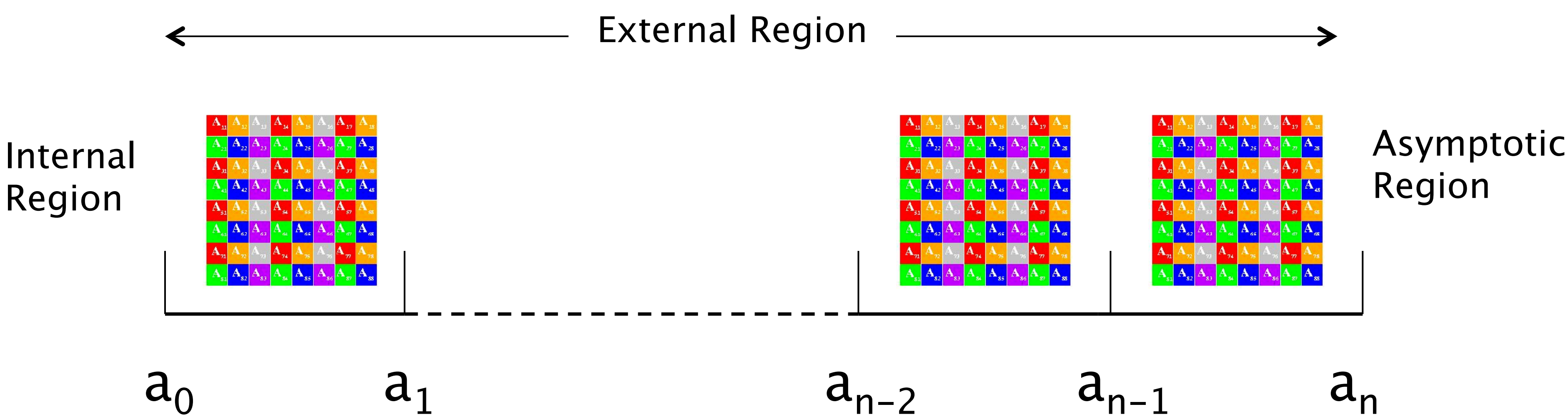
Baluja-Burke-Morgan (BBM)-based Implementation

2 Stage Parallelization of BBM approach in the external region:

- EXDIG Program (Modern Fortran):
  - Diagonalize Sector Hamiltonian matrices using ScaLAPACK **PDSYEVD** (Blacs-based Data decomposition).
- EXAS Program (Modern Fortran):
  - For each scattering energy propagate using 3 functional groups:
  - Generate initial R-Matrix **PDGEMM** (Data decomposition).
  - Propagate R-Matrix across each sector in pipeline (Control decomposition). **DGEMM, DGETRF, DGEMM**
  - Calculate thermally averaged collision strengths. Serial S.V.D. (Task Farmed).

EXAS Stage

Serial, OpenMP and MPI versions

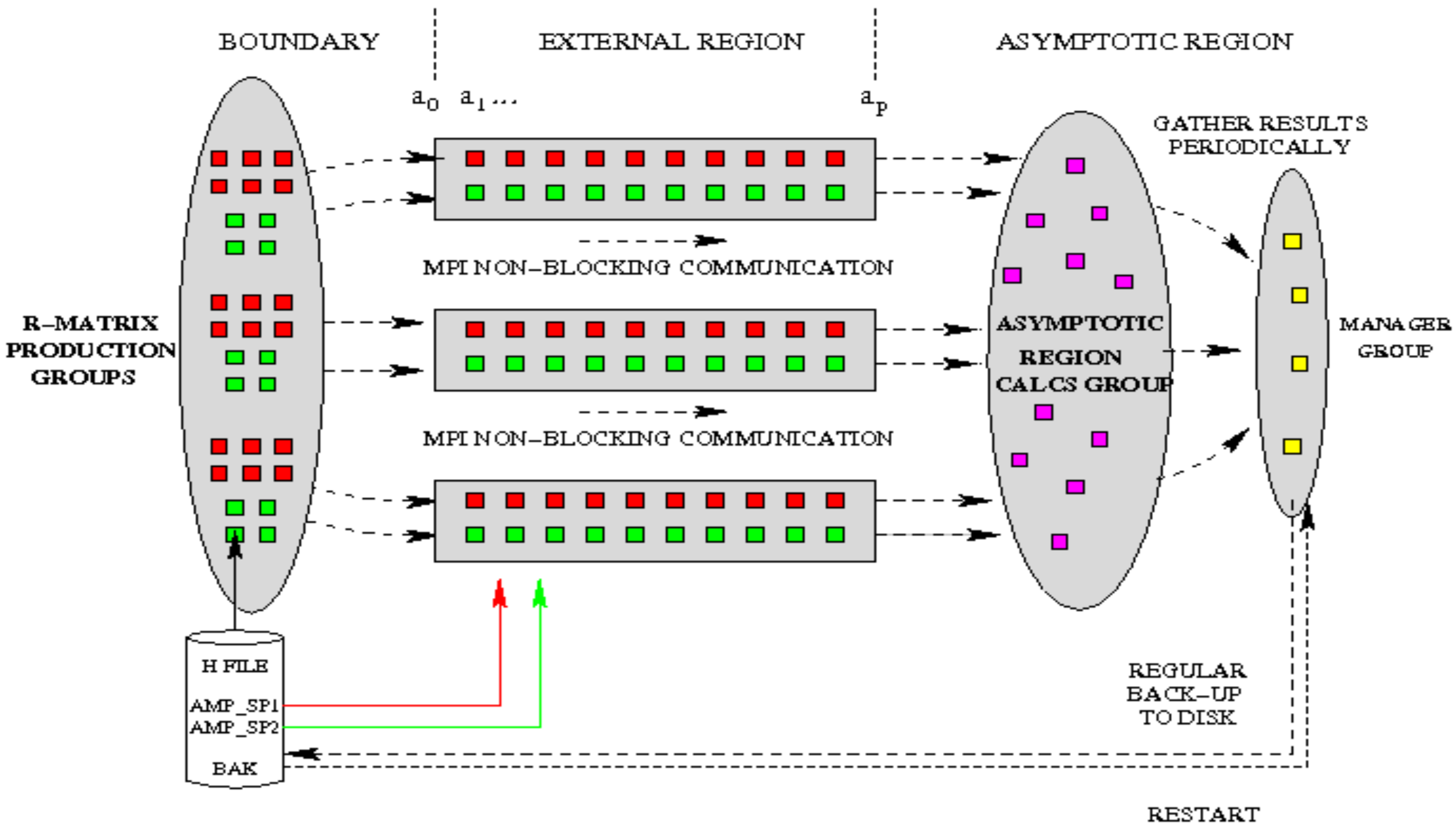


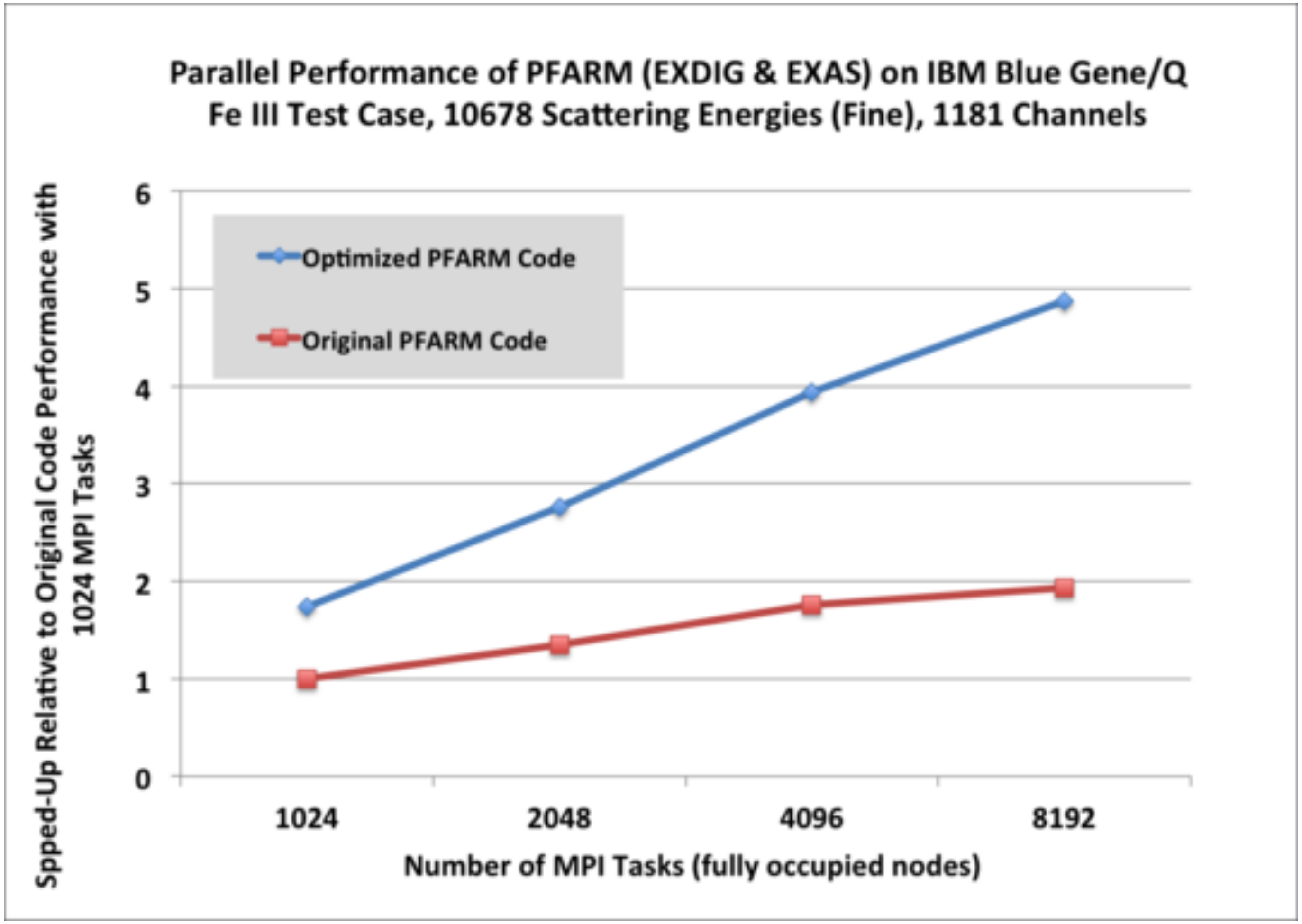
Parallel Diagonalizations of Large Symmetric Sector Hamiltonian Matrices

EXAS Stage

•Outer code PFARM, scales to 10000s of cores: now used with both atomic inner region and UKRmol

- full parallel diagonalization (ScaLAPACK), multiple MPI task propagation and pipelining:





Optimized code – overall 150% performance improvement on 8132 cores (I/O and diag improvement)

Candidates for Offloading

- Four dense linear algebra operations identified as candidates for offloading to Xeon Phi:
  - Matrix Multiply in EXAS (dgemm)
  - Linear Solver in EXAS (dgetrf)
  - Singular Value Decomposition in EXAS (dgesvd)
  - Symmetric Eigensolver in EXDIG (dsyevd)

MKL & MAGMA

- Intel® Math Kernel Library (MKL)
  - A library of highly optimized, extensively threaded math routines including BLAS library, LAPACK, ScaLAPACK, sparse solvers, Fast Fourier Transforms library, vector math, and more.
- Matrix Algebra on GPU and Multicore Architectures (MAGMA)
  - similar to LAPACK but for heterogeneous/hybrid architectures, starting with current "Multicore+GPU" systems.

http://icl.cs.utk.edu/magma/index.html

https://software.intel.com/en-us/tools-for-math-processing

Offloading in MKL

- Code with highly parallel phases
- Code runs on Xeon Host until a sufficiently computationally heavy region reached
- Data transfer to Phi and execution runs there
- Data transferred back to Host
- Auto or user defined

Image modified from:
•Slidecast 3/3 – PRACE Summer School on Code Optimisation for Multi-Core and Intel MIC Architectures – Workshop on MIC
•Intel MIC Architecture – Intel MIC HW/SW Architecture

Science & Technology Facilities Council

Offloading in MAGMA

- Client/Server model
- Server must be active on the Phi before offloading can occur
- Small non-parallelizable tasks are scheduled on the host, whilst larger, more parallelizable tasks, (e.g. Level 3 BLAS), are scheduled on the Intel Xeon Phi.
- Unlike MAGMA-GPU, no supplier-provided Fortran interfaces

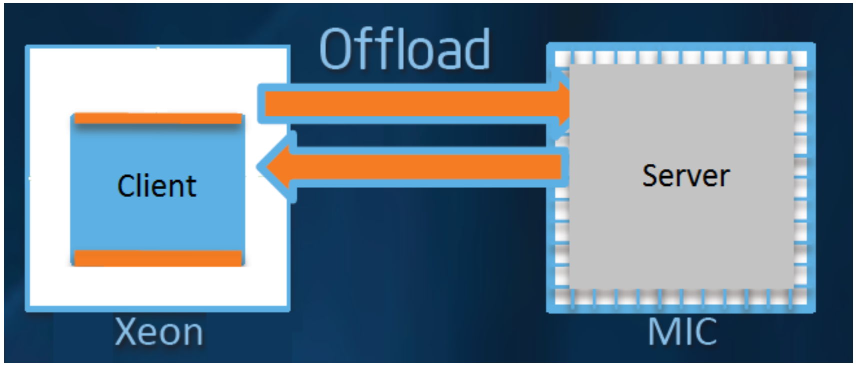
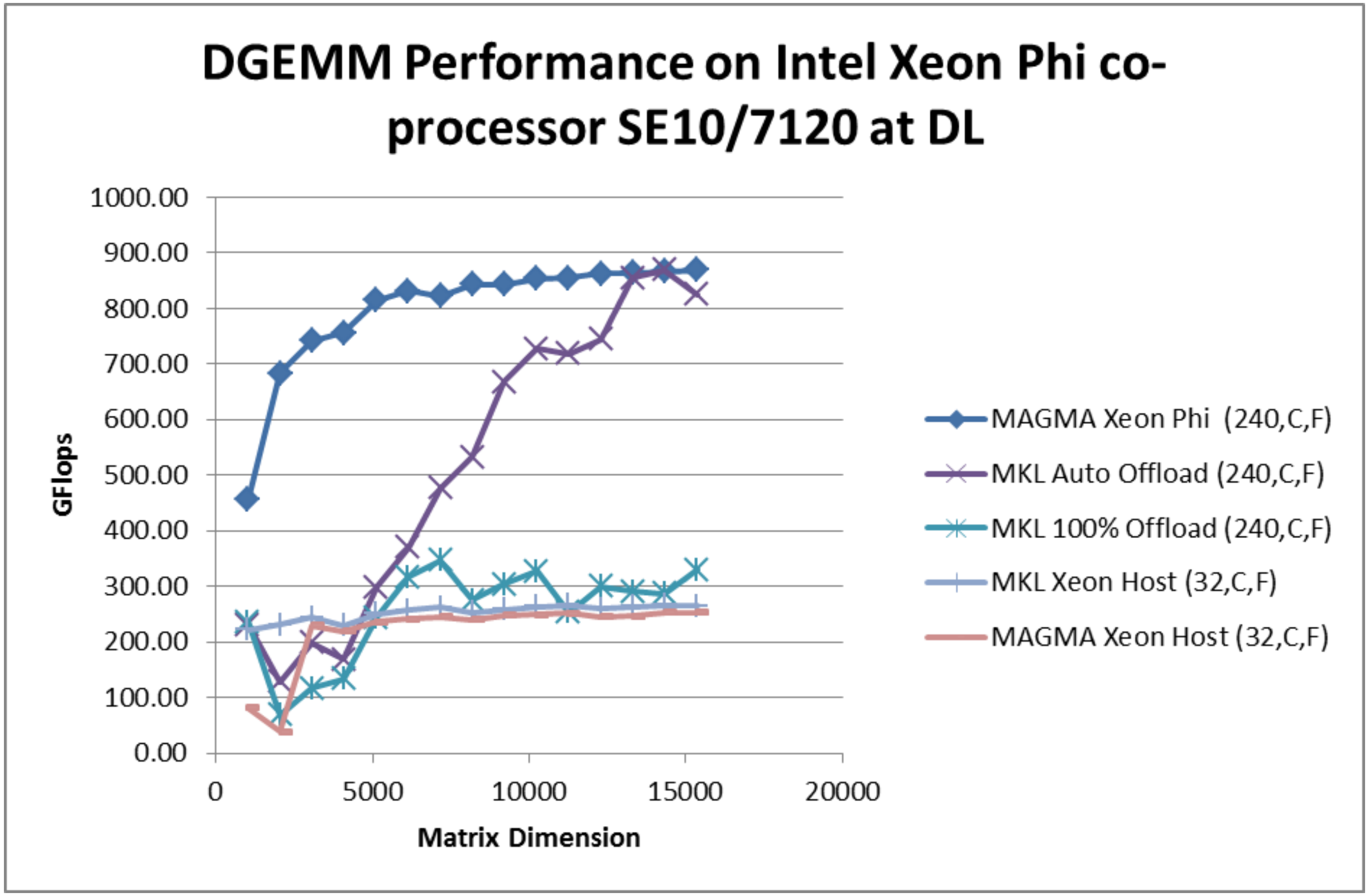


Image modified from:
•Slidecast 3/3 – PRACE Summer School on Code Optimisation for Multi-Core and Intel MIC Architectures – Workshop on MIC
•Intel MIC Architecture – Intel MIC HW/SW Architecture

Science & Technology Facilities Council

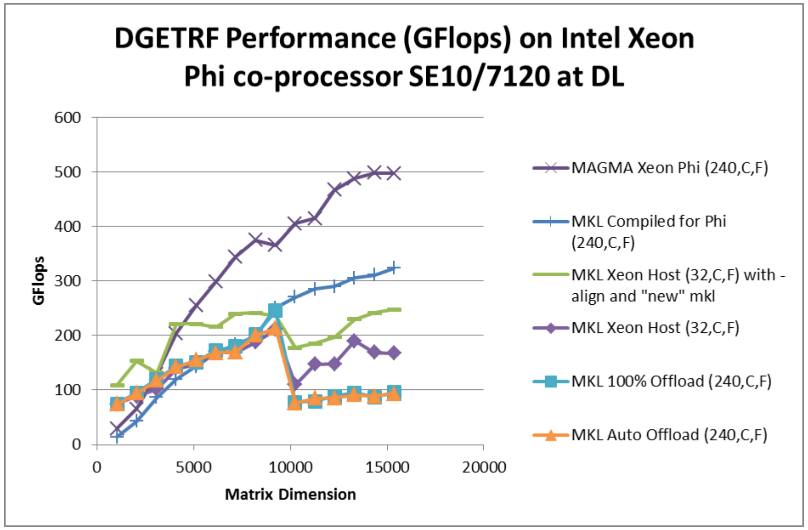
DGEMM Performance on Intel Xeon Phi

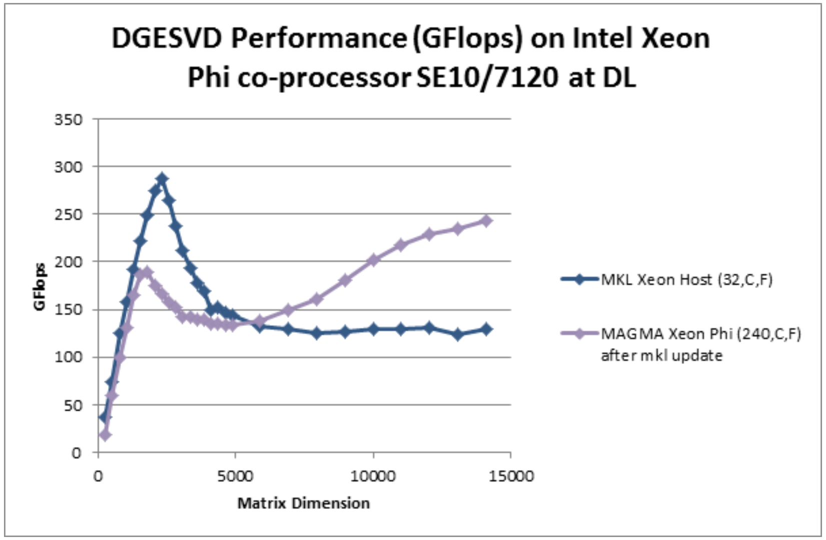


DL Xeon Phi co-processor SE10/7120
2x8 cores on Host
61 cores on Xeon Phi

Science & Technology Facilities Council

DGETRF & DGESVD Performance on Intel Xeon Phi



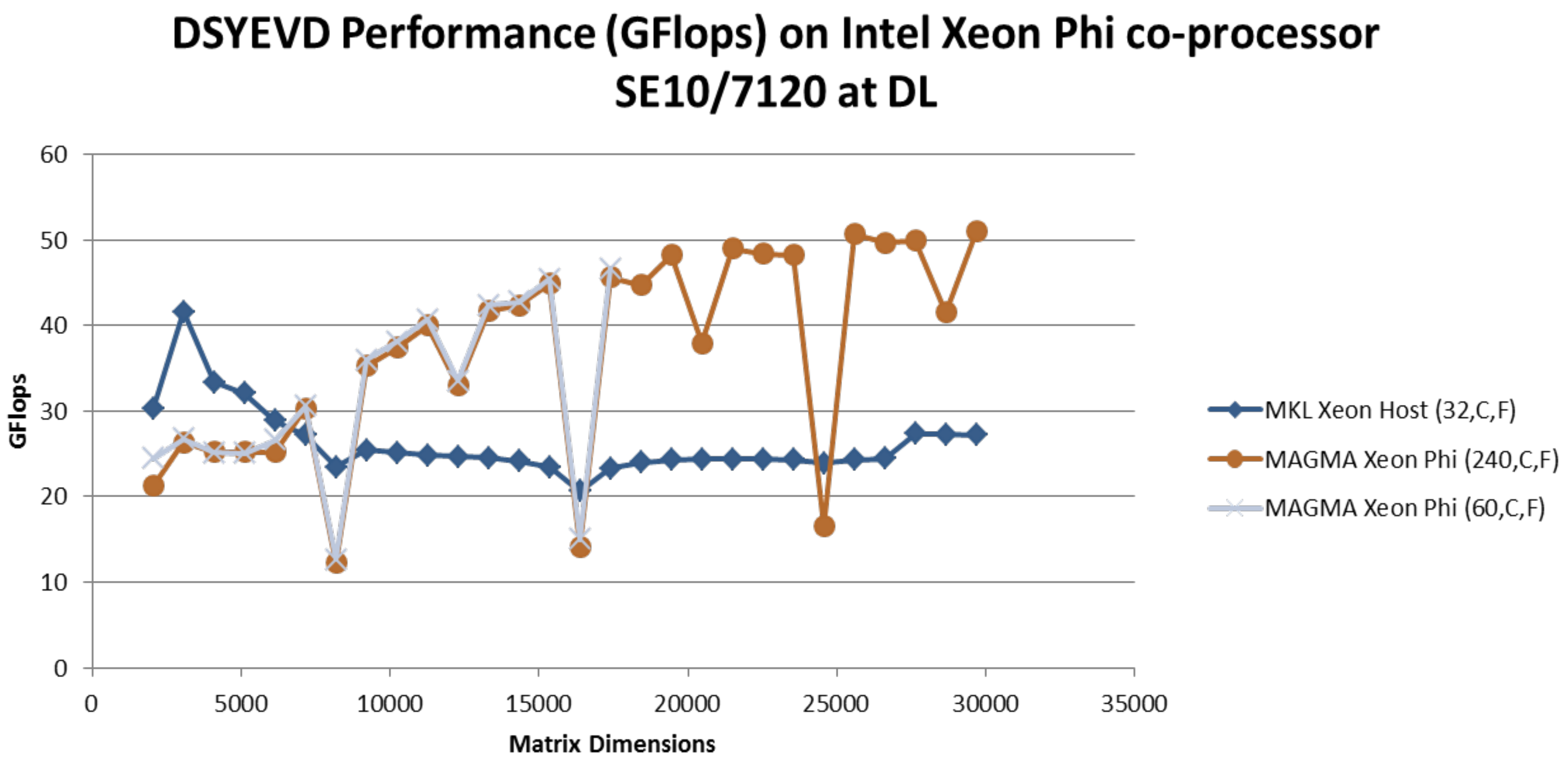


EXAS matrices too small at present to gain advantage form offloading DGETRF & DGESVD

DL Xeon Phi co-processor SE10/7120
2x8 cores on Host
61 cores on Xeon Phi

Science & Technology Facilities Council

DSYEVD (Eigensolver) Performance on Intel Xeon Phi

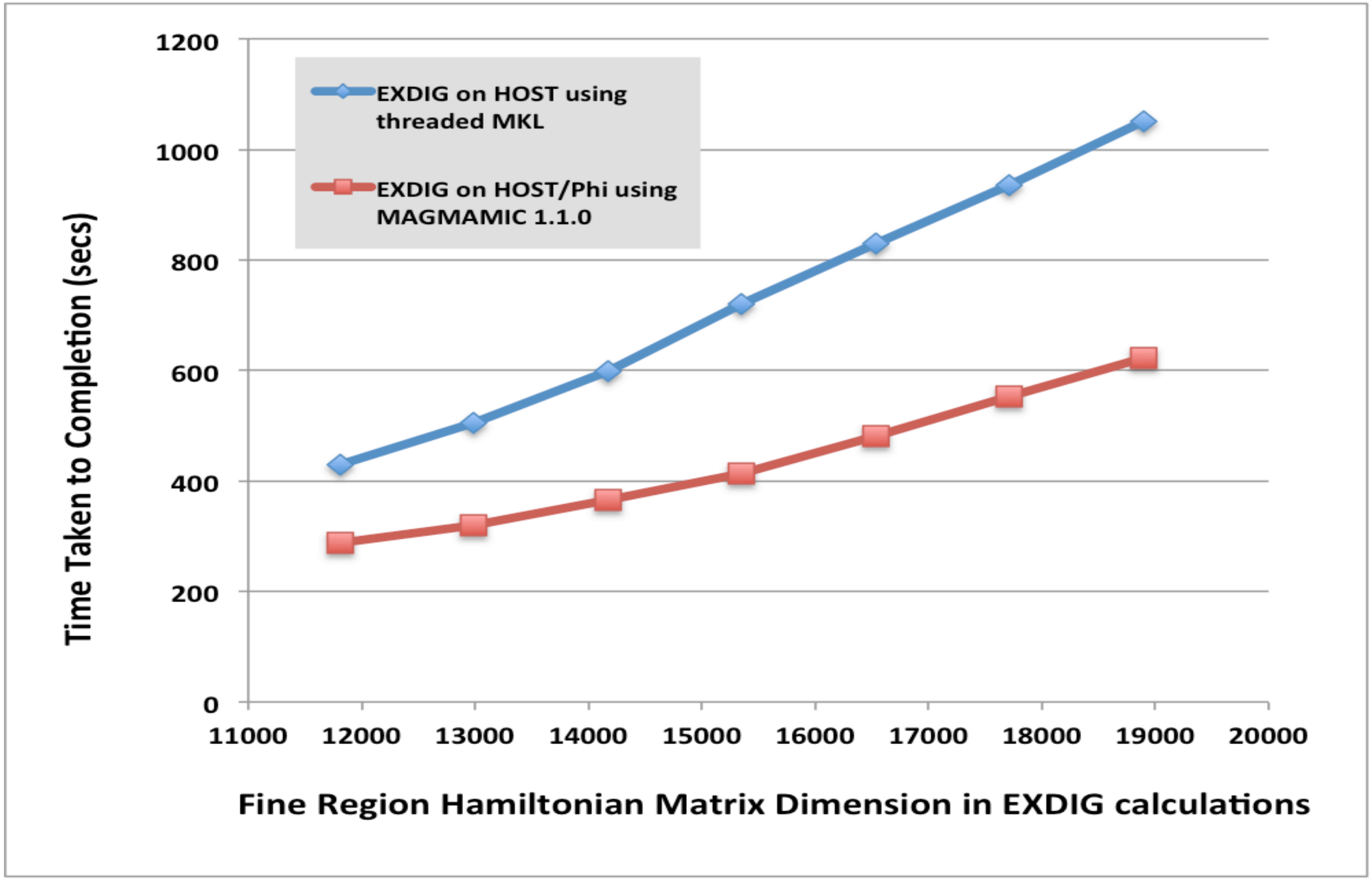


DL Xeon Phi co-processor SE10/7120
2x8 cores on Host
61 cores on Xeon Phi

Science & Technology Facilities Council

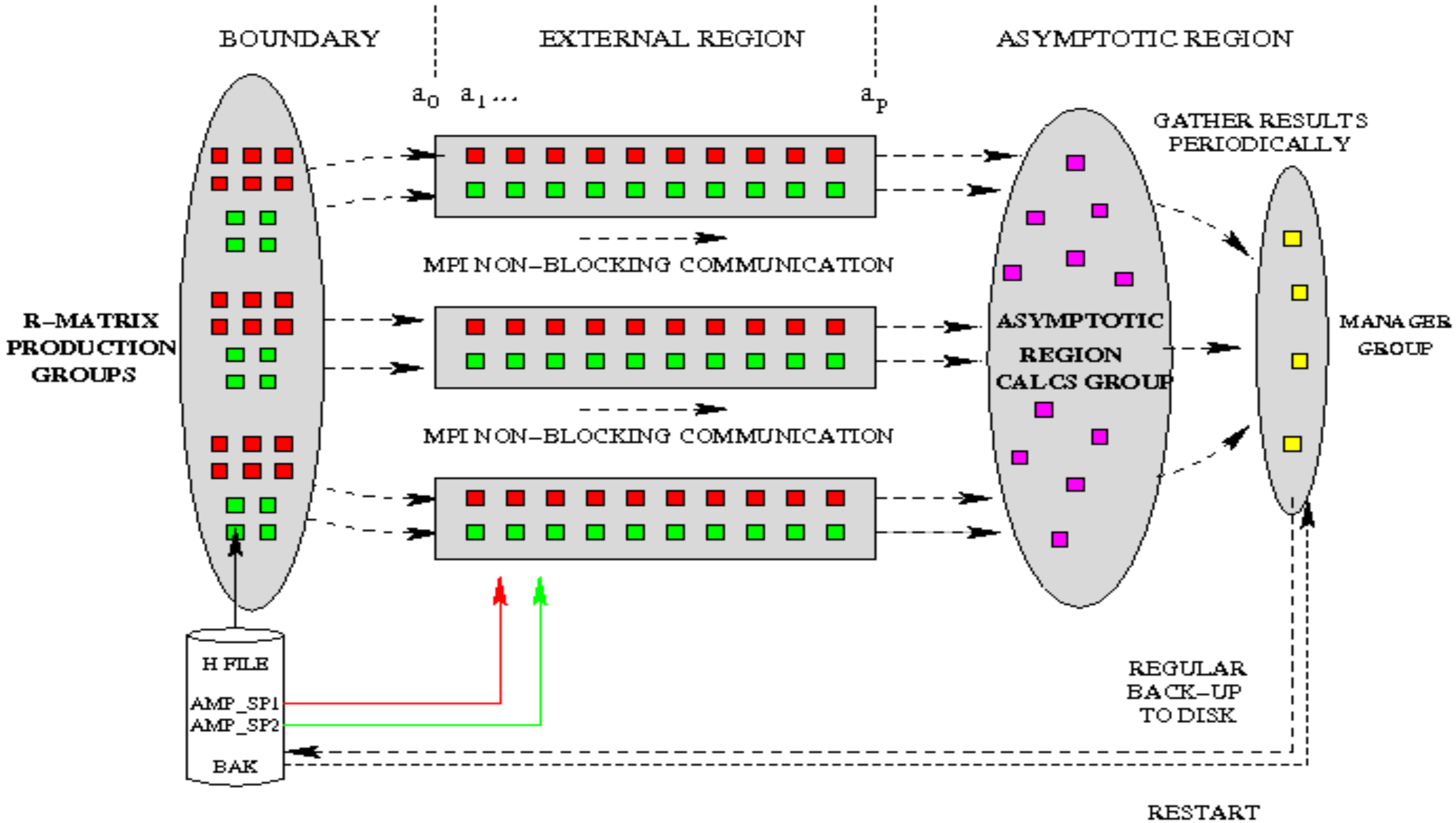
Performance of EXDIG with Xeon Phi acceleration using MAGMA

The Xeon host calculations are run using MKL v 1.1 with 32 threads and the Xeon Phi MAGMA v1.1.0 calculations use 240 threads.



Science & Technology Facilities Council

EXAS Stage



EXAS on Xeon Phi

- Matrices generally too small for effective offloading

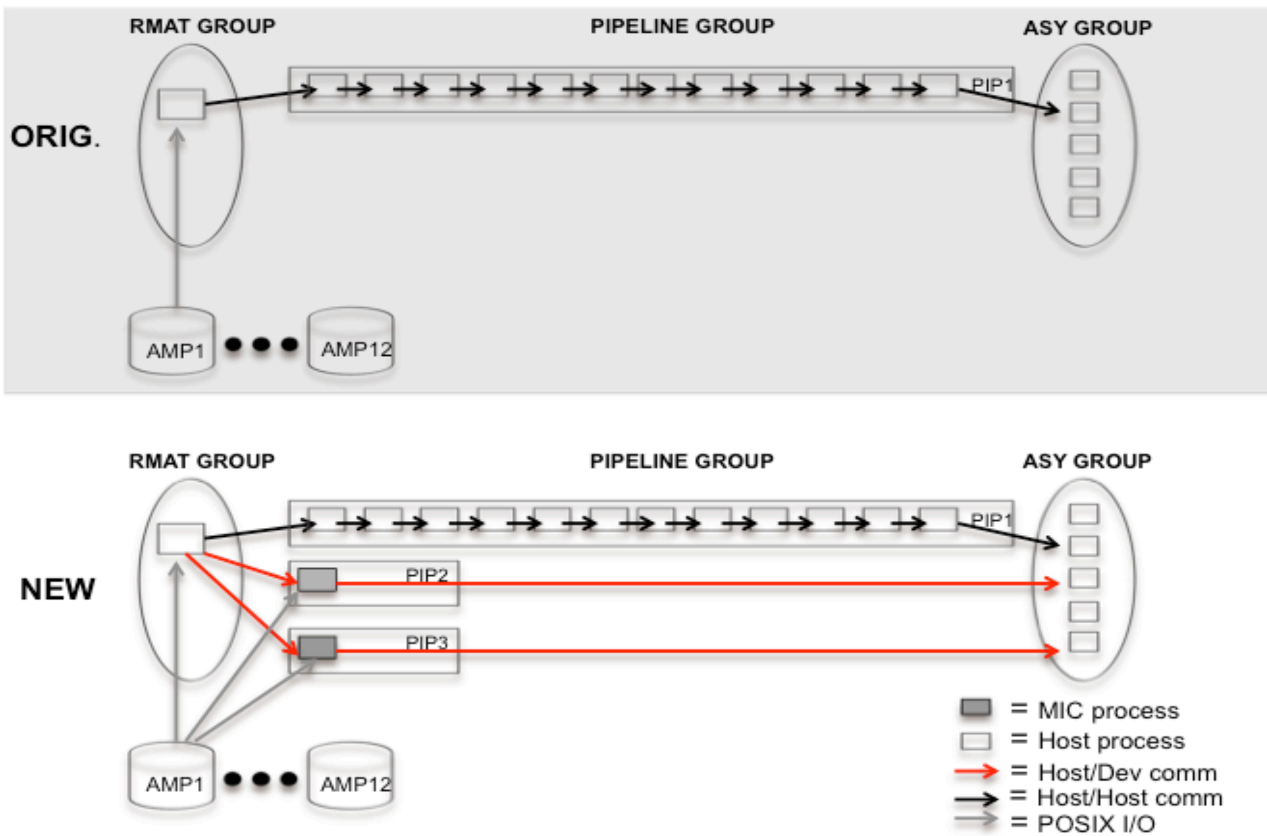
- Collapse a pipeline communicator and do the work for the pipeline on the Phi (shared memory parallelism), hosts can continue to run alongside use standard distributed pipelines
  - Replacement coding strictly localized with a clear interface to the main code within the pipelining modules

- Dominated by dense linear algebra operations. Originally undertaken with MKL (serial, shared memory tasks, distributed memory)

- The new version of EXAS is fully heterogeneous, ie hosts and Intel Phis perform separate work simultaneously

Science & Technology Facilities Council

Schematic of original EXAS implementation for a single pipeline (top) and schematic of new EXAS implementation enabled for Fionn Xeon Phi machine (bottom)



Performance analysis of original implementation of EXAS (left) and new implementation of EXAS on Xeon Phi (right) using the Intel Trace Analyzer and Collector (ITAC) profiler

Flat Profile | Load Balance | Call Tree | Call Graph

Children of Group All_Processes

Name	TSelf	TSelf	TTotal	#Calls	TSelf /Call
Group MPI	7.37452e+3 s		7.37452e+3 s	8105	909.873e-3 s
Group Application	7.4129e+3 s		14.7874e+3 s	20	370.645 s
Process 0	424.606 s		739.362 s	1	424.606 s
Process 1	565.53 s		739.37 s	1	565.53 s
Process 2	570.265 s		739.372 s	1	570.265 s
Process 3	569.291 s		739.372 s	1	569.291 s
Process 4	569.861 s		739.372 s	1	569.861 s
Process 5	569.036 s		739.371 s	1	569.036 s
Process 6	568.069 s		739.372 s	1	568.069 s
Process 7	565.308 s		739.371 s	1	565.308 s
Process 8	561.372 s		739.372 s	1	561.372 s
Process 9	560.036 s		739.372 s	1	560.036 s
Process 10	556.76 s		739.371 s	1	556.76 s
Process 11	559.524 s		739.372 s	1	559.524 s
Process 12	512.158 s		739.371 s	1	512.158 s
Process 13	44.6289 s		739.371 s	1	44.6289 s
Process 14	45.9804 s		739.37 s	1	45.9804 s
Process 15	46.7251 s		739.371 s	1	46.7251 s
Process 16	47.2213 s		739.37 s	1	47.2213 s
Process 17	47.5858 s		739.371 s	1	47.5858 s
Process 18	417.497e-3 s		739.371 s	1	417.497e-3 s
Process 19	28.5248 s		739.371 s	1	28.5248 s

Flat Profile | Load Balance | Call Tree | Call Graph

Children of Group All_Processes

Name	TSelf	TSelf	TTotal	#Calls	TSelf /Call
Group MPI	25.5177e+3 s		25.5177e+3 s	4746	5.37667 s
Group Application	6.38296e+3 s		31.9006e+3 s	22	290.135 s
Process 0	894.348 s		1.45e+3 s	1	894.348 s
Process 1	462.845 s		1.44998e+3 s	1	462.845 s
Process 2	371.256 s		1.44998e+3 s	1	371.256 s
Process 3	371.24 s		1.44998e+3 s	1	371.24 s
Process 4	371.36 s		1.44998e+3 s	1	371.36 s
Process 5	371.53 s		1.44999e+3 s	1	371.53 s
Process 6	370.523 s		1.44999e+3 s	1	370.523 s
Process 7	371.445 s		1.44999e+3 s	1	371.445 s
Process 8	370.817 s		1.44999e+3 s	1	370.817 s
Process 9	371.023 s		1.44999e+3 s	1	371.023 s
Process 10	370.331 s		1.44999e+3 s	1	370.331 s
Process 11	371.993 s		1.44999e+3 s	1	371.993 s
Process 12	370.935 s		1.44999e+3 s	1	370.935 s
Process 13	82.9509 s		1.44999e+3 s	1	82.9509 s
Process 14	85.245 s		1.44999e+3 s	1	85.245 s
Process 15	86.219 s		1.44999e+3 s	1	86.219 s
Process 16	87.0879 s		1.44999e+3 s	1	87.0879 s
Process 17	88.1475 s		1.44999e+3 s	1	88.1475 s
Process 18	1.15865 s		1.44999e+3 s	1	1.15865 s
Process 19	79.0817 s		1.44999e+3 s	1	79.0817 s
Process 20	216.994 s		1.45045e+3 s	1	216.994 s
Process 21	216.434 s		1.45046e+3 s	1	216.434 s

Science & Technology Facilities Council

Summary

- Optimised Intel Xeon Phi port of PFARM (EXDIG) incorporating MAGMA MIC for accelerated parallel eigensolvers. (~2x speed-up overall)

- A new version of PFARM (EXAS), restructured for accelerated R-matrix propagation pipelining. Tested and tuned on the Intel Xeon Phi and also applicable to GPUs (M.L. expects speed-up once communication bottlenecks reduced)

- Detailed analyses of MKL and MAGMA MIC numerical library routines performance on Intel Xeon Phi architectures.

- EXAS undergoing further optimization: currently a host in one functional group offloads to a Phi in another group with slow comms
  - Fully flexible MPI/OpenMP version – Distribute complete multiple functional groups efficiently across Host/Phi, exploit OpenMP 4.0 task model
  - Preparation for Knights Landing
  - MAGMA MIC, MKL Offloading for PDGEMM (large rectangular matrices)