Science & Technology Facilities Council

Towards Performance Portability in GungHo and GOcean

M. Ashworth, **R. Ford**, A. Porter - STFC Daresbury,
J. Holt, H. Liu - NOC Liverpool,
C. Maynard, T. Melvin - Met Office,
G. Riley - Manchester University,
+ GungHo team

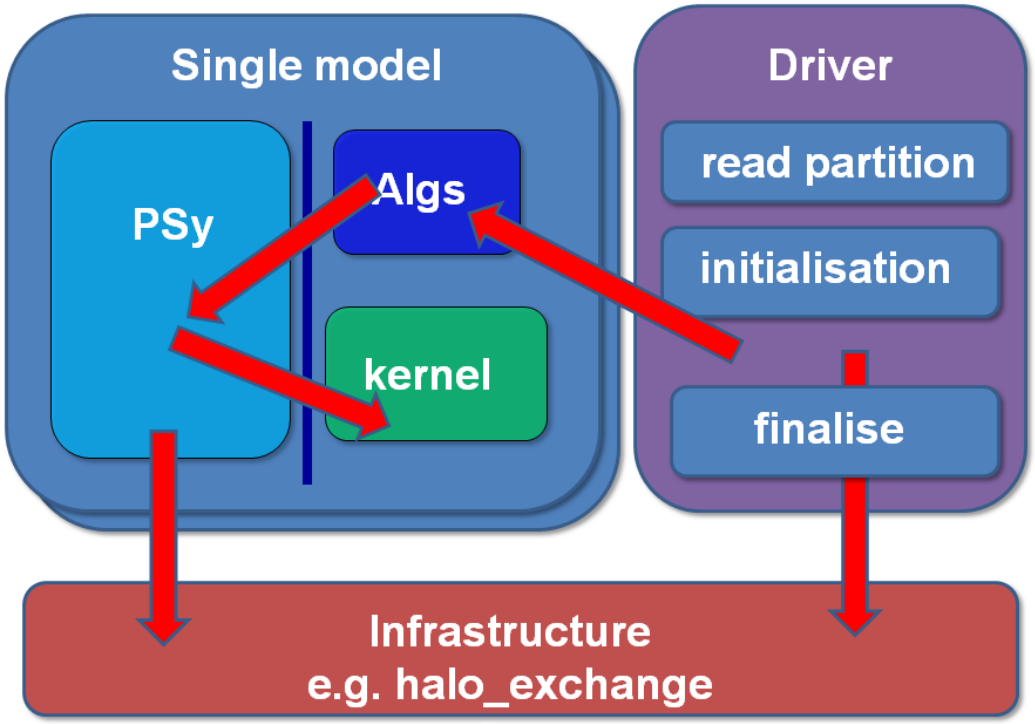
GungHo

- GungHo is a Met Office/NERC/STFC project developing the Met Office's next generation dynamical core – Quasi-uniform mesh, finite element based
- Computer architectures are in a state of flux with a variety of competing technologies
  - developing code for computers that do not yet exist
  - Many cores, accelerators (PCIe or socket), FPGAs…
- How can we produce maintainable, scientifically-correct code that will perform well on a range of future architectures?

工合

Science & Technology Facilities Council

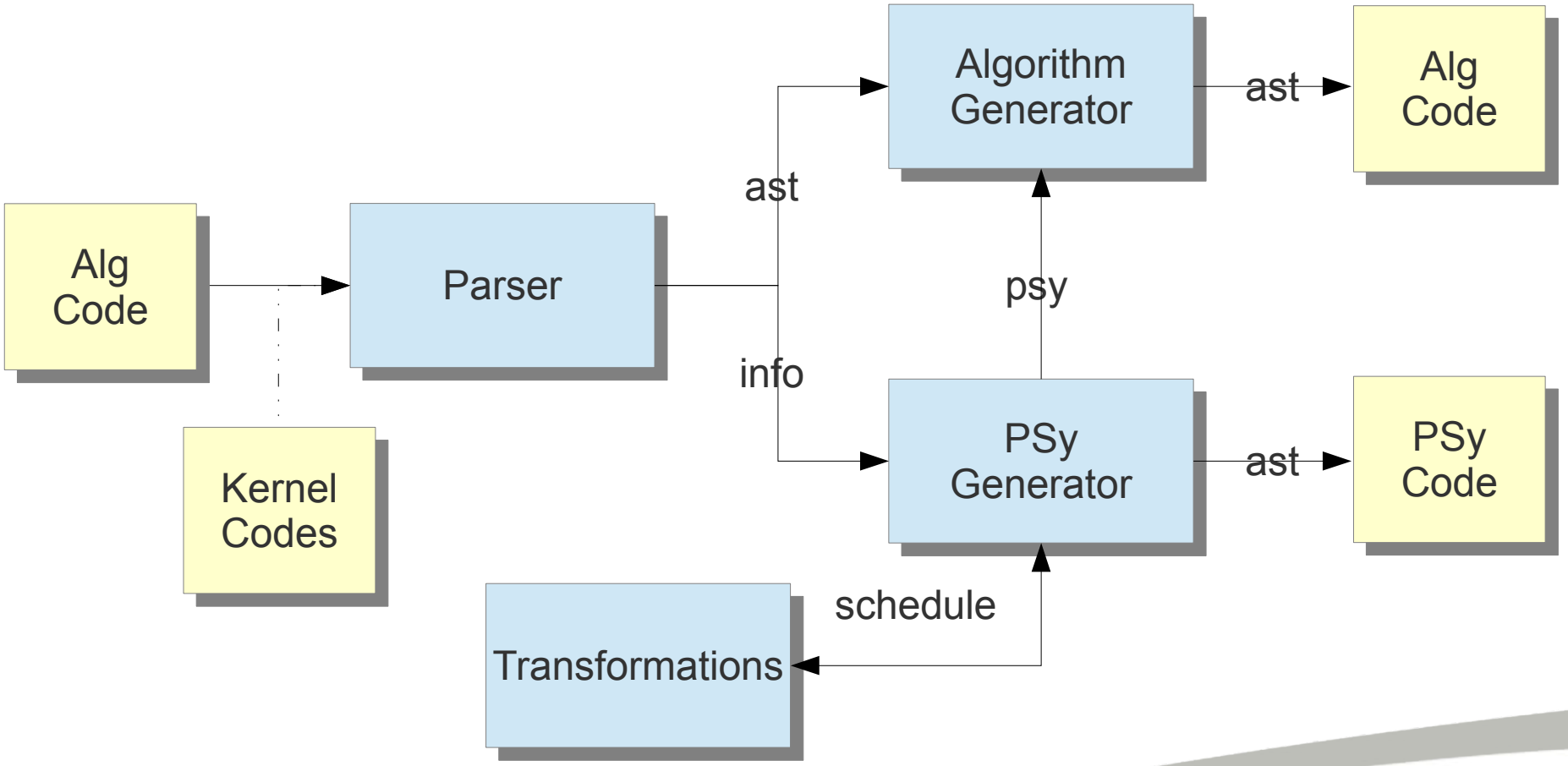
Separation of Concerns



The Parallel System, Kernel, Algorithm (PSyKAl) Approach…

- Scientist writes the algorithm (top) and kernel (bottom) layers, following certain rules
  - no need to worry about relative indexing of various fields
  - no need to worry about parallelism (algorithm layer deals with logically global field quantities)
- A code-generation system (PSyclone) generates the PSy middle layer
  - glues the algorithm and kernels together
  - incorporates **all code related to parallelism**

PSyclone Structure



Science & Technology Facilities Council

GOcean

- NERC funded, 03/2014 – 02/2015
- Collaboration between NOC and STFC
- Investigate the feasibility of applying technology from the GungHo project to Ocean modelling
- Extend the developing Gung-Ho infrastructure to support finite difference on regular, lat-lon grids

工合海洋 GOcean

Science & Technology Facilities Council

Two shallow-water codes…

- We have applied the PSyKAl approach to two codes:
  - 'shallow:' originally written by Swarztrauber, NCAR
  - 'NEMOLite2D:' 2D, free-surface part of NEMO extracted by NOC
- Both use Finite Difference on Arakawa C grid
- But there are important differences:
  - Boundary conditions (periodic vs. forced/closed)
  - Relative indexing of variables on the grid
- Understanding and expressing these differences is essential for correct code generation

Two benchmarks…

- Have re-structured both Shallow and NEMOLite2D following PSyKAl separation of concerns
- Optimise Shallow while obeying PSyKAl rules
  - Serial benchmark
- Optimise NEMOLite2D and (manually) implement OpenMP parallelisation
  - Parallel benchmark
- These benchmarks have also been supplied to vendors (IBM, NVIDIA) for them to optimise for their hardware while obeying PSyKAl rules

Body of time-stepping loop consists of kernel calls:

```
call invoke(                                                    &
            continuity(ssha_t, sshn_t, sshn_u, sshn_v,          &
                       hu, hv, un, vn, rdt),                    &

            momentum_u(ua, un, vn, hu, hv, ht,                  &
                       ssha_u, sshn_t, sshn_u, sshn_v),         &

            momentum_v(va, un, vn, hu, hv, ht,                  &
                       ssha_v, sshn_t, sshn_u, sshn_v),         &

            bc_ssh(istp, ssha_t),                               &

            bc_solid_u(ua),                                     &
            bc_solid_v(va),                                     &

            bc_flather_u(ua, hu, sshn_u),                       &
            bc_flather_v(va, hv, sshn_v),                       &

            copy(un, ua),                                       &
            copy(vn, va),                                       &
            copy(sshn_t, ssha_t),                               &

            next_sshu(sshn_u, sshn_t),                          &
            next_sshv(sshn_v, sshn_t)                           &
           )
```

Science & Technology Facilities Council

A kernel looks like:

```
subroutine continuity_code(ji, jj,                          &
                           ssha, sshn, sshn_u, sshn_v, &
                           hu, hv, un, vn, rdt, e12t)
  implicit none
  integer,                    intent(in)  :: ji, jj
  real(wp),                   intent(in)  :: rdt
  real(wp), dimension(:,:), intent(in)  :: e12t
  real(wp), dimension(:,:), intent(out) :: ssha
  real(wp), dimension(:,:), intent(in)  :: sshn, sshn_u, sshn_v, &
                                           hu, hv, un, vn
  ! Locals
  real(wp) :: rtmp1, rtmp2, rtmp3, rtmp4

  rtmp1 = (sshn_u(ji  ,jj  ) + hu(ji  ,jj  )) * un(ji  ,jj  )
  rtmp2 = (sshn_u(ji-1,jj  ) + hu(ji-1,jj  )) * un(ji-1,jj  )
  rtmp3 = (sshn_v(ji  ,jj  ) + hv(ji  ,jj  )) * vn(ji  ,jj  )
  rtmp4 = (sshn_v(ji  ,jj-1) + hv(ji  ,jj-1)) * vn(ji  ,jj-1)

  ssha(ji,jj) = sshn(ji,jj) + (rtmp2 - rtmp1 + rtmp4 - rtmp3) * &
                   rdt / e12t(ji,jj)

end subroutine continuity_code
```

Science & Technology Facilities Council

Kernel meta-data

- meta-data specifies what quantities a kernel requires from the infrastructure:

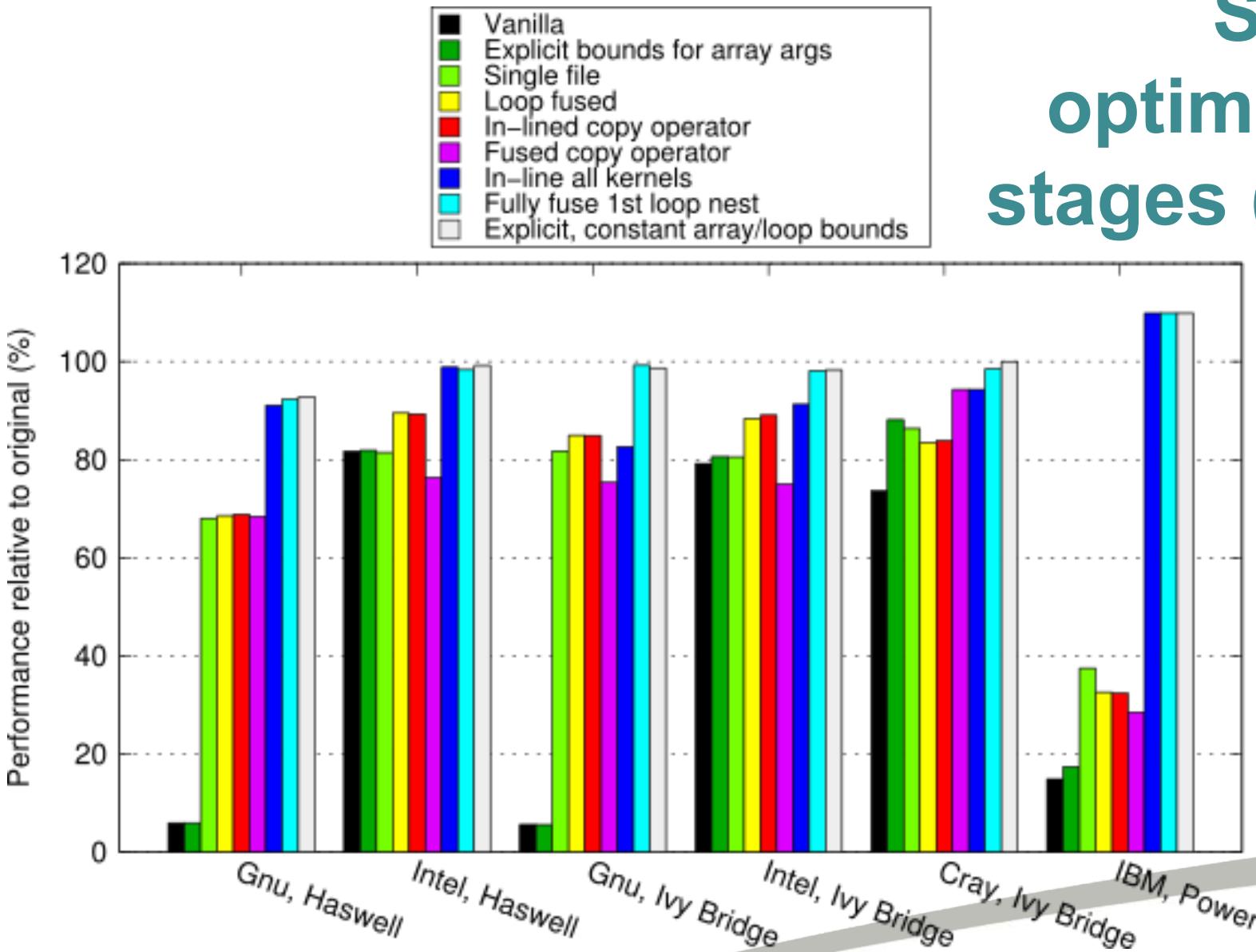
```
type, extends(kernel_type) :: next_sshu
   type(arg), dimension(5) :: meta_args =  &
          (/ arg(READWRITE, CU, POINTWISE),  &
             arg(READ,      CU, POINTWISE),  &
             arg(READ,      GRID_MASK_T),    &
             arg(READ,      GRID_AREA_T),    &
             arg(READ,      GRID_AREA_U)     &
           /)
```

- PSyclone then supplies these quantities from the generated middle layer

What about performance?

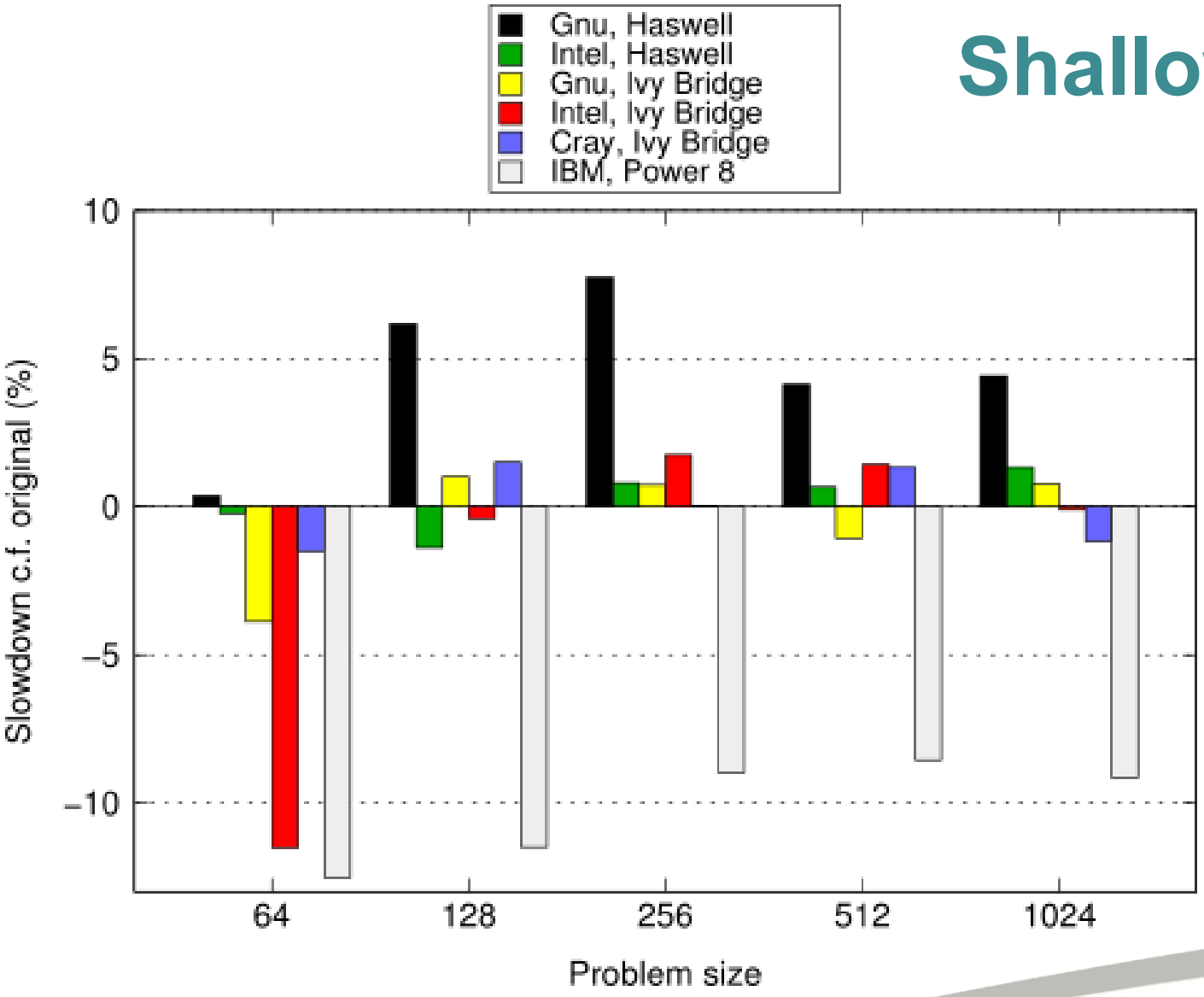
Science & Technology
Facilities Council

Shallow optimisation stages (serial)



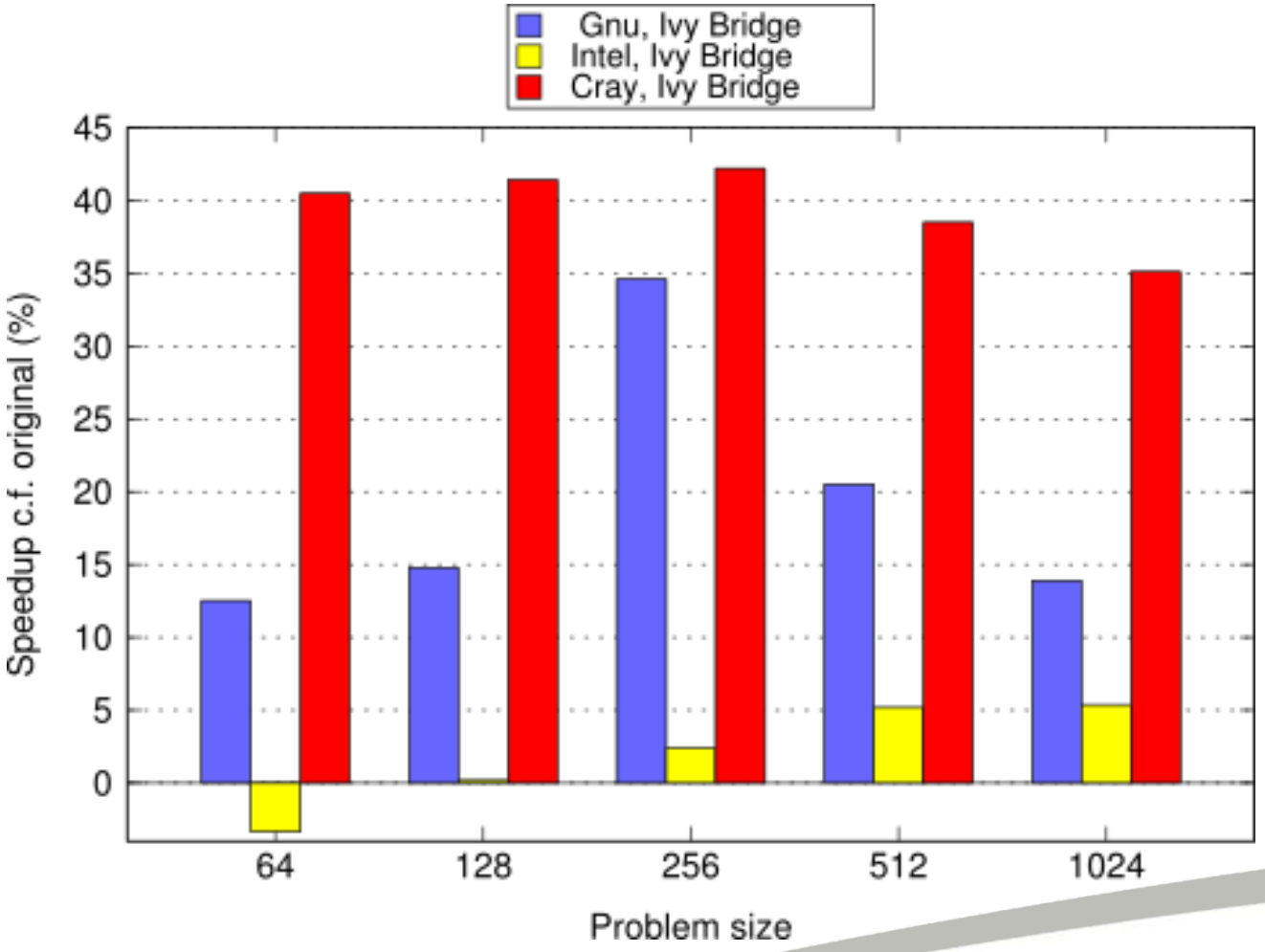
Science & Technology Facilities Council

Shallow (serial)



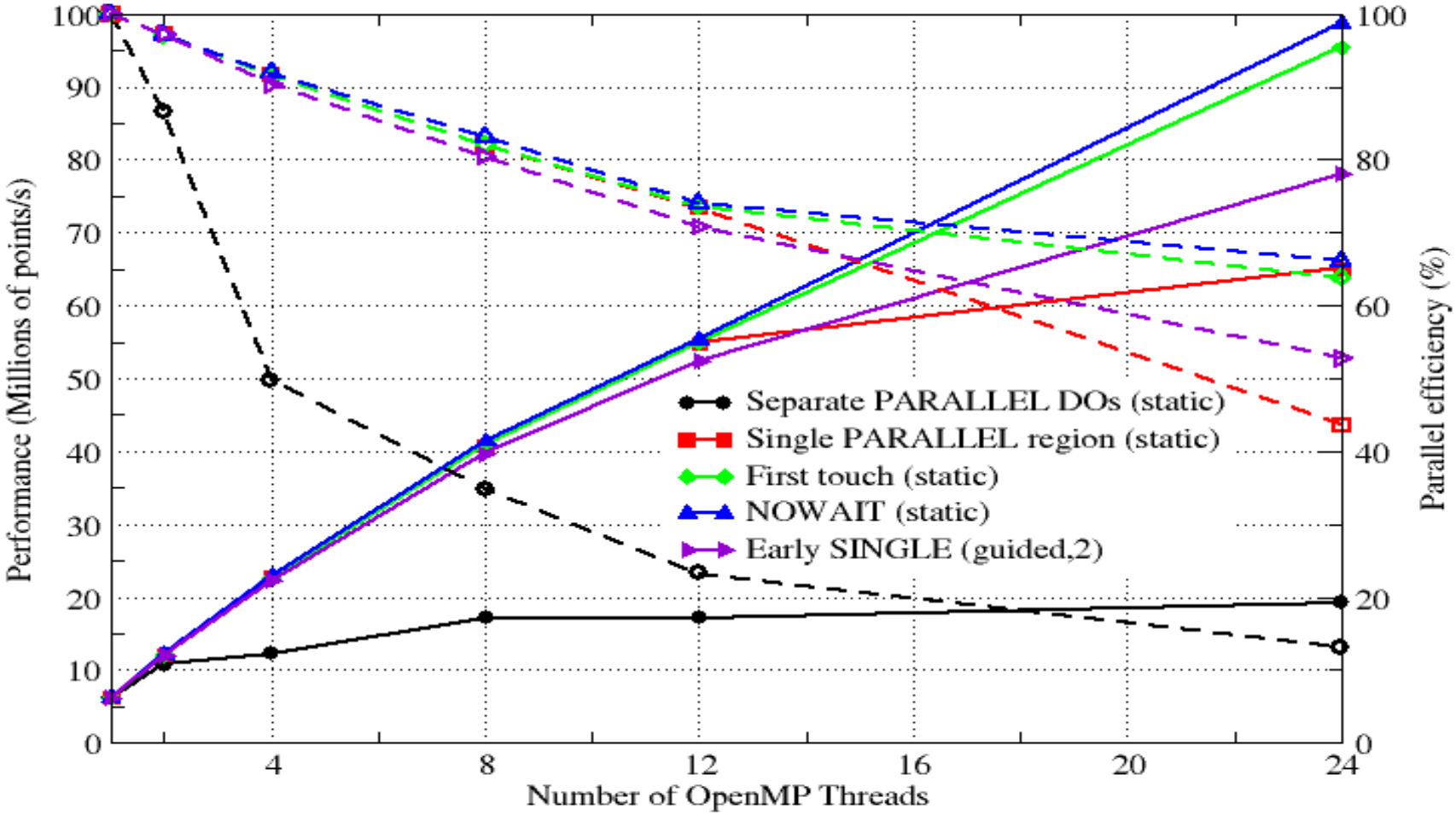
Science & Technology Facilities Council

NEMOLite2D (serial)



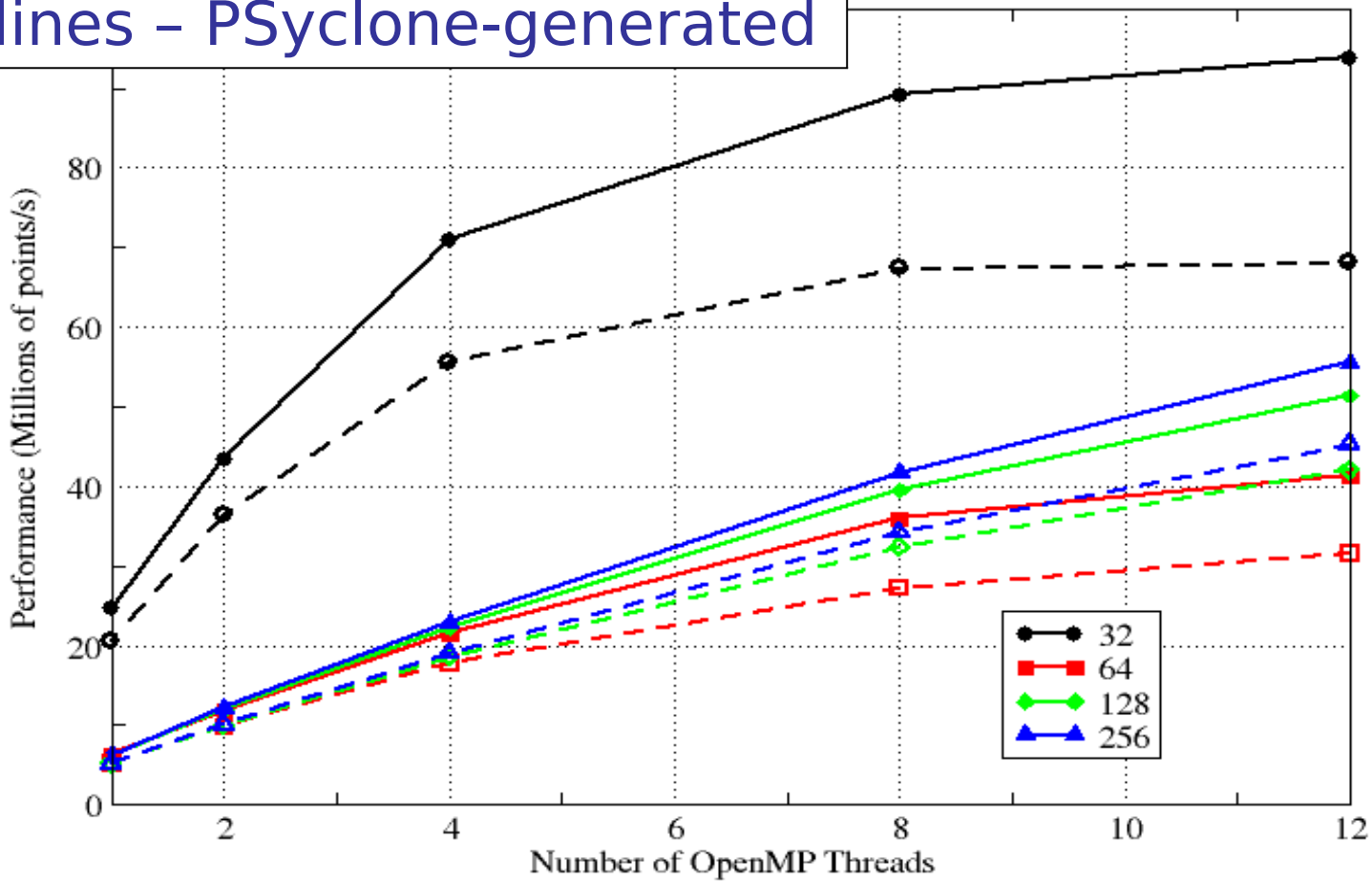
Science & Technology Facilities Council

NEMOLite2D OpenMP optimisation stages

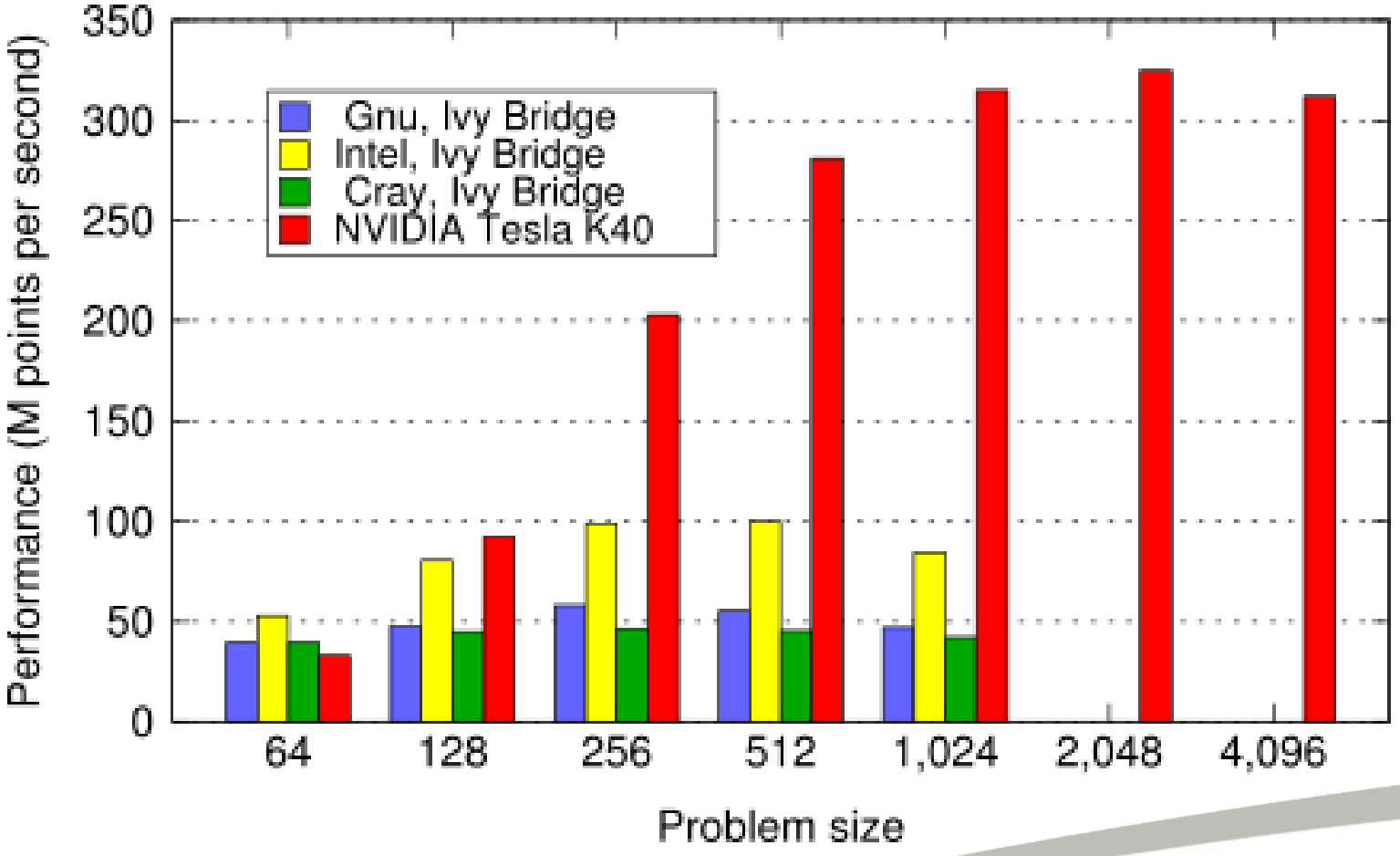


NEMOLite2D, OpenMP

Solid lines – manual OpenMP
Dashed lines – PSyclone-generated



NEMOLite2D, GPU



Science & Technology Facilities Council

Summary

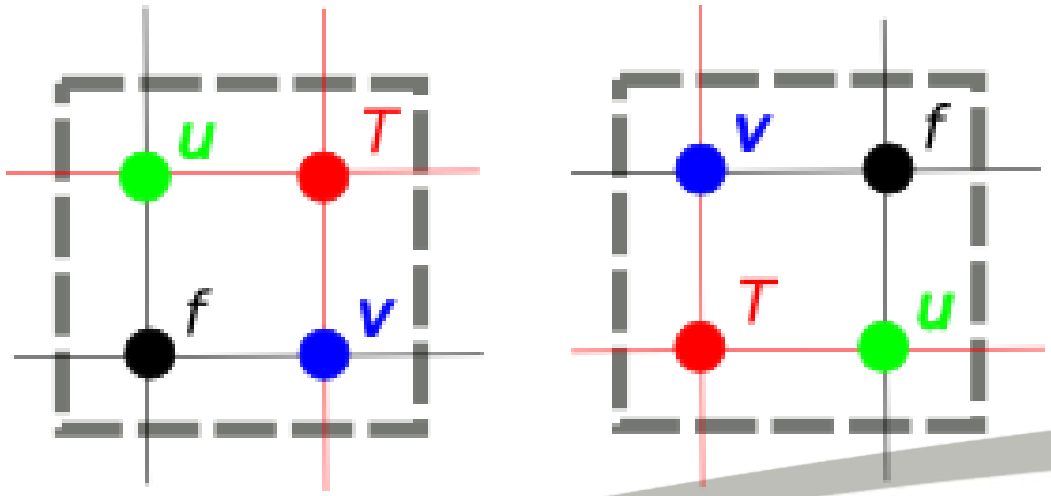
- Compilers are complex
  - Recovering performance is not straightforward
  - More transformations required in PSyclone
- Currently not exploring the optimisation space
  - Only attempting to recover original code structure
- Separation of Concerns: Introduces flexibility needed to achieve performance on different architecture
  - Potentially enables *e.g.* OpenMP or OpenACC to be used, depending on target hardware
  - No need to modify source code
  - Similar performance to hand optimised code
- Work continuing on PSyclone in the GungHo project
  - Including MPI parallelisation

Extras…

Offset choice

- A *developer* can choose how the different grid-point types are indexed relative to T
- shallow defines {u,v,f} points to the South and West of the T point to have same (i,j) index while NEMO uses those to the North and East:

- We call this choice the 'offset' of the grids
- Specified in kernel meta-data



Science & Technology Facilities Council