Durham University
Department of Computer Science

Intelligent algorithms on intelligent networks

SIAM PP 24

With help from M. Bader[°], A. Basden[x], R. Graham[+], J. Moore[□ *], P. Samfass[° ⊤], M. Turner[⊥ □]

[°] Technische Universität München
[x] DiRAC, Durham University
[+] NVIDIA Networking
[□] Computer Science, Durham University
[*] Computer Science, University of Bristol
[⊤] AMD, Germany
[⊥] Advanced Research Computing (ARC), Durham University

March 5, 2024

Outline

Durham University
Department of Computer Science

The science case

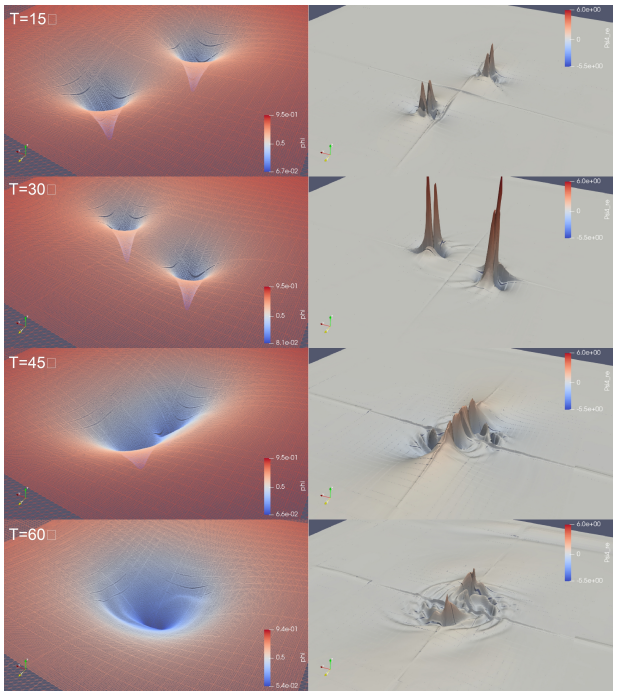
A smart team of MPI ranks and three smart ideas

Intelligent network environments

Next-gen software architecture

The “search” for gravitational waves

Durham University
Department of Computer Science



Rotating binary black holes; collaboration with Baojiu Li and Han Zhang (Institute for Computational Cosmology

- Numerical scheme
  - CCZ4 formulation
  - ADER-DG with Finite Volume limiting (first order)
  - RK1FD4 (Euler) and RK4FD4
  - Dynamically adaptive Cartesian meshes
- Benchmark against GRChombo
  - Stability
    (artificial smoothing through KO dissipation)
  - Numerical diffusion
  - Rotation speed

  $\Rightarrow$ Fundamental numerical insight

  $\Rightarrow$ Understand FO impact

Challenge for my group

Durham University
Department of Computer Science

American English

> Chris Johnson: “Before every great invention was the discovery of a new tool”[+].

[+] From an interview in the video “The Golden Age of Computing”.

British English

> Humphry Davy: “Nothing tends so much to the advancement of knowledge as the application of a new instrument. The native intellectual powers of men in different times are not so much the causes of the different success of their labours, as the peculiar nature of the means and artificial resources in their possession.”

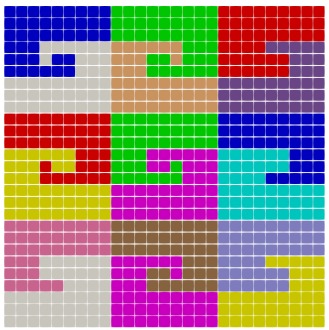
Thanks to David E. Keyes for pointing to this quote.

ExaHyPE—a tool with MPI+X+X+X

Durham University
Department of Computer Science

$$\frac{\partial}{\partial t} Q + \nabla \cdot \mathbf{F}(Q) + \sum_i \mathcal{B}_i \frac{\partial Q}{\partial x_i} - \underbrace{\nabla \mathcal{K} \nabla Q}_{\text{not with FO}} = \mathbf{S} + \sum \delta$$

- Various numerical schemes & adaptive Cartesian grids
- Strict separation of concerns through Python API
- Parallelisation & technical details hidden
- ...



SFC-based domain partitioning of regular grid for 18 ranks/cores.

MPI SFC-based non-overlapping domain decomposition

Threads SFC-based non-overlapping domain decomposition
(OpenMP, TBB, C++, SYCL)

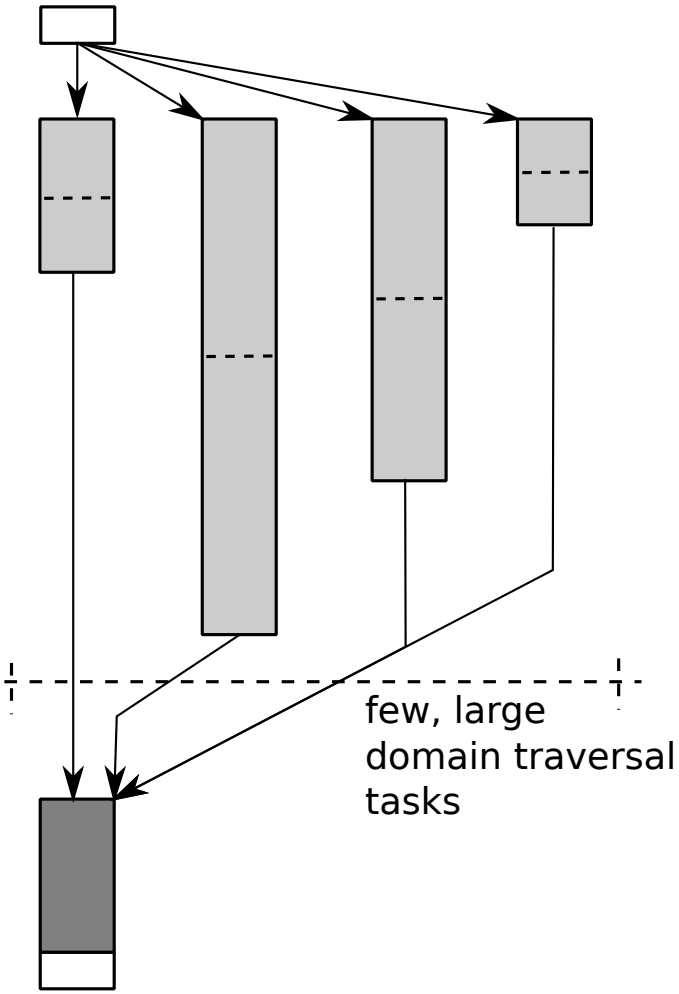
Threads — next slide —
(OpenMP, TBB, C++)

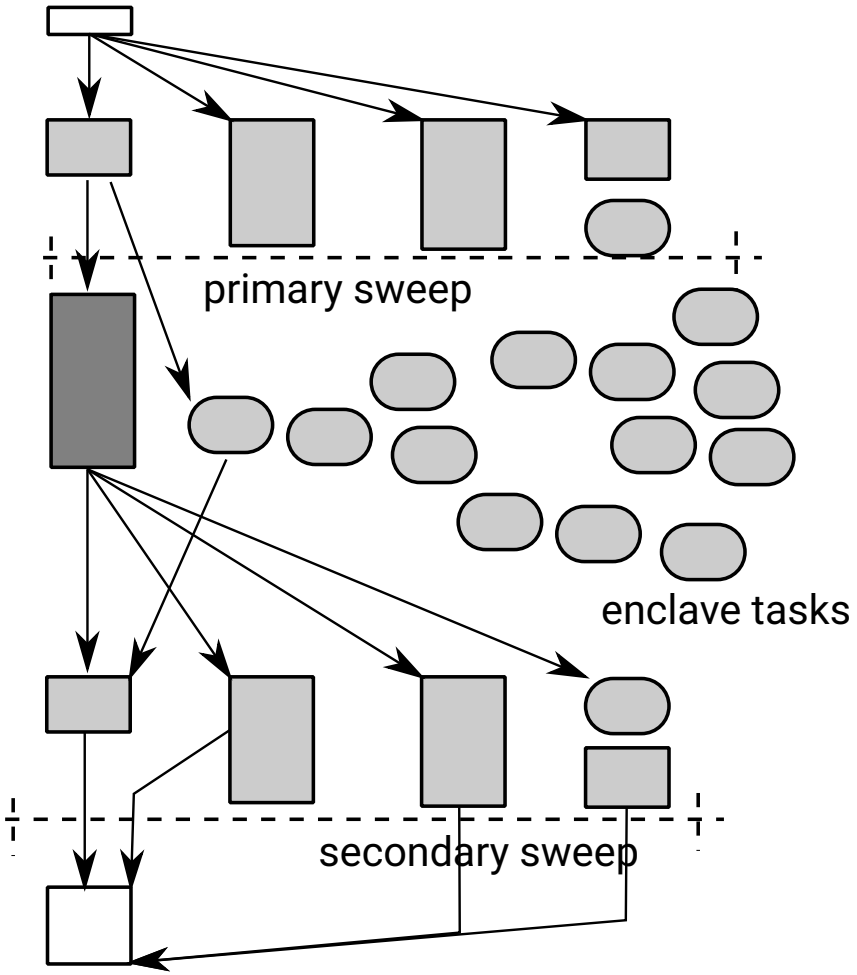
GPUs GPU offloading by different tasks
(OpenMP, SYCL, (NVIDIA's) C++ extensions)

⇒ See Chung's talk in paper session CP 5 on Wednesday 11:30am

Fork-join is intrinsically problematic: tasking

Durham University
Department of Computer Science





D.E. Charrier, B. Hazelwood, T. Weinzierl: Enclave Tasking for DG Methods on Dynamically Adaptive Meshes. SISC 42(3) (2020)

Research questions

Durham University
Department of Computer Science

- *Efficacy* Can we roll this out over many ranks?
- *Efficiency* Can we benefit from multi-rank tasking?
- *Robustness* Can we make it robustly fast?

Outline

Durham University
Department of Computer Science

The science case

A smart team of MPI ranks and three smart ideas

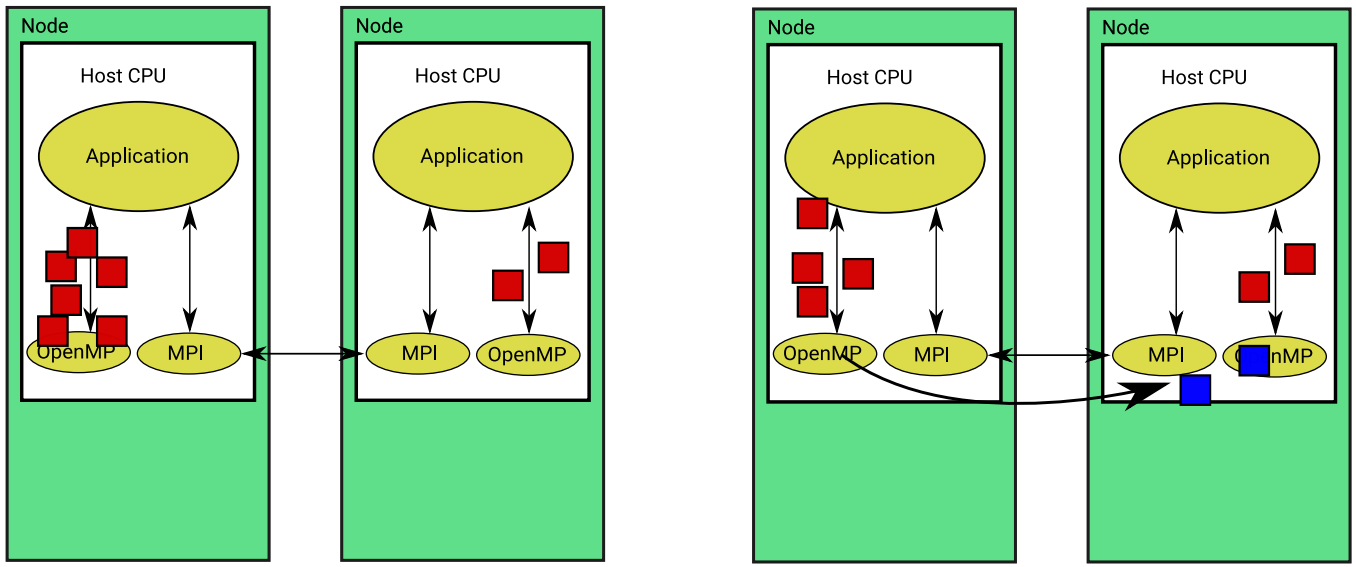
Intelligent network environments

Next-gen software architecture

Idea #1: Tasks to load-balance between MPI ranks

Durham University
Department of Computer Science

Research vision: Use non-persistent task migration to compensate for ill-balances.



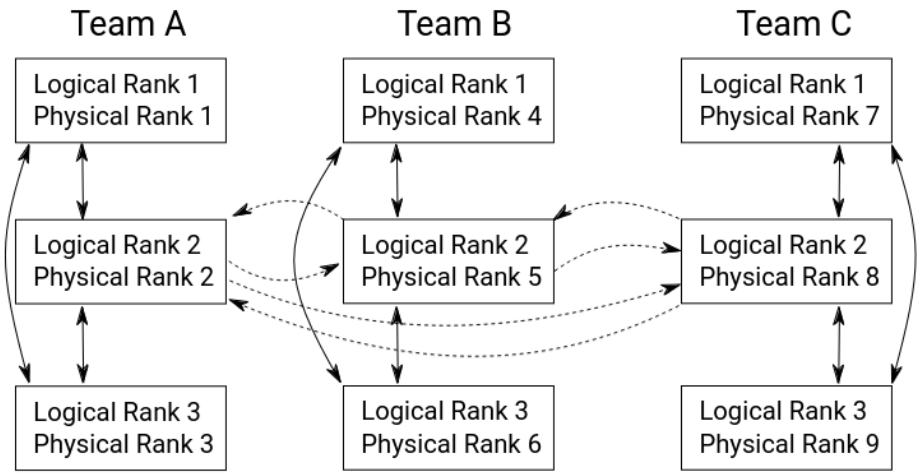
- Measure MPI imbalance (reduce MPI wait times)
- Derive task distribution matrix
- Migrate tasks to other rank
- Bring results back

$\Rightarrow$ Idling ranks take over work from heavy ranks

Idea #2: Replicate tasks to become resilient

Durham University
Department of Computer Science

Research vision: Replicate ranks but exchange task outcomes as “shortcuts”.



- Each rank exists multiple times ($N \geq 2$) (split global ranks into N teams)
- Permute task execution slightly (randomised scheduler)
- Share task outcomes
- Cancel task if result arrives on time, i.e. already known

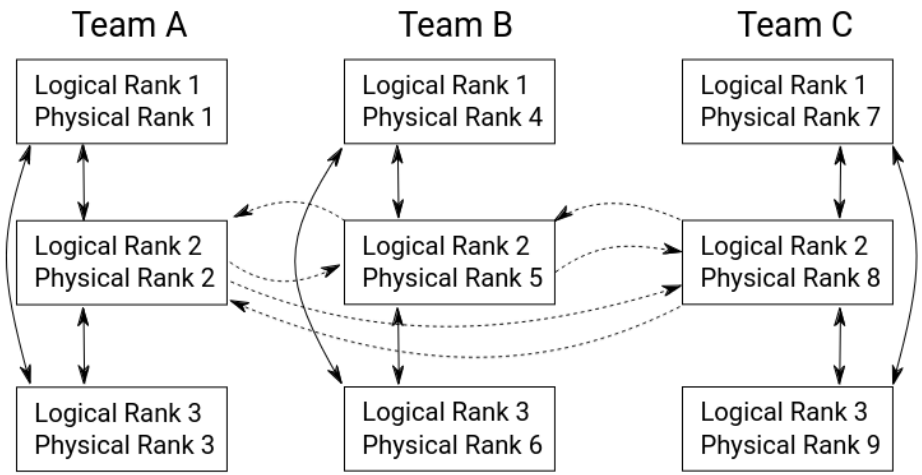
$\Rightarrow$ Rank drop-out does not terminate application

$\Rightarrow$ Replication is not (too) expensive

Idea #3: Compare task outcomes

Durham University
Department of Computer Science

Research vision: Compare redundantly computed task outcomes to flag potential problems.

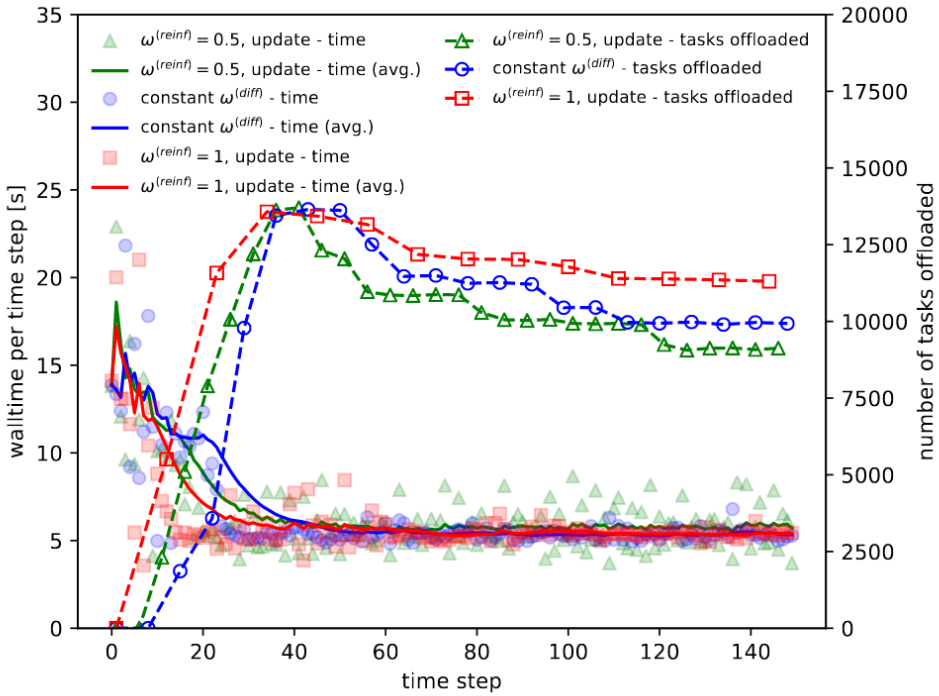


- Replicate some task computations
- Compare outcomes/hashes as flagging mechanism
- Get in third opinion if required

$\Rightarrow$ Spot some soft errors on-the-fly

Does all of this work?

Durham University
Department of Computer Science



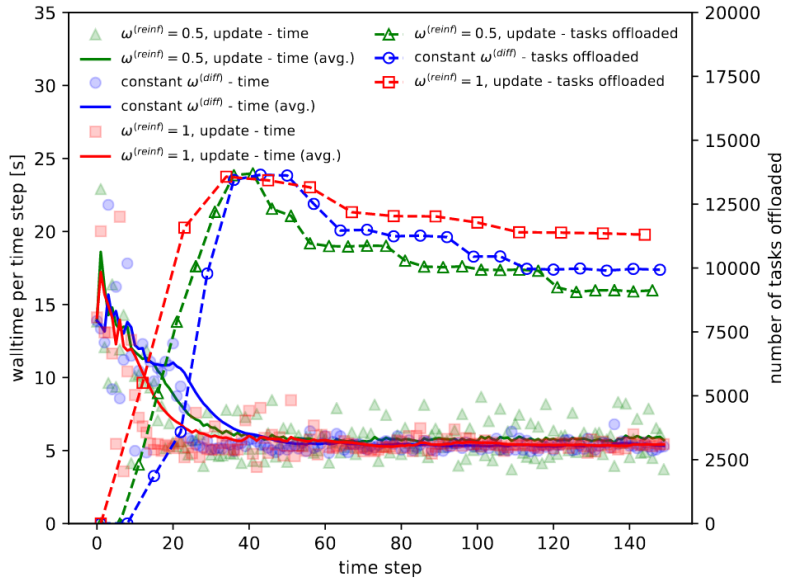
From “Lightweight task offloading . . . ”: After few iterations, the task migration adopts and improves the runtime.

Good “news”:

- P. Samfass, T. Weinzierl, D.E. Charrier, M. Bader: Lightweight task offloading exploiting MPI wait times for parallel adaptive mesh refinement, CCPE 32(24) (2020)
- P. Samfass, T. Weinzierl, B. Hazelwood, M. Bader: TeaMPI—Replication-Based Resilience Without the (Performance) Pain. ISC 20 (2020)
- P. Samfass, T. Weinzierl, A. Reinarz, M. Bader: *Doubt and Redundancy Kill Soft Errors—Towards Detection and Correction of Silent Data Corruption in Task-based Numerical Software*, Supercomputing 21 (2021)

Does all of this work?

Durham University
Department of Computer Science



From “Lightweight task offloading . . . ”: After few iterations, the task migration adopts and improves the runtime.

Bad “news”:

- Overlap does not happen
  (MPI progression issue)
- Overhead too high
  (unexpected message arrival, tag matching, . . .)
- Orchestration interrupts CPU’s work
  (cycles sacrificed for core MPI/scheduling work)
- Network stress increases
  (vicious cycle of congestion and additional offloading)

Outline

The science case

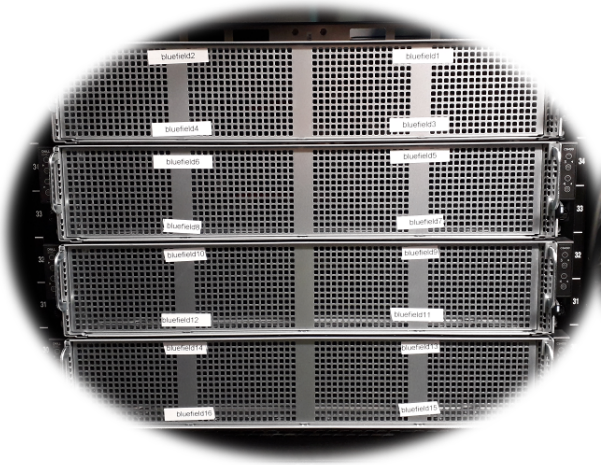
A smart team of MPI ranks and three smart ideas

Intelligent network environments

Next-gen software architecture

DINE 1.0

Durham University
Department of Computer Science



DINE (name due to Alastair Basden)

Durham Intelligent Network Environment:

- Seedcorn funding by Strategic Investment of Durham CS
- 16 nodes AMD EPYC
- 16 BlueField-1 cards (Ethernet)
- Installed in cooperation with ICC

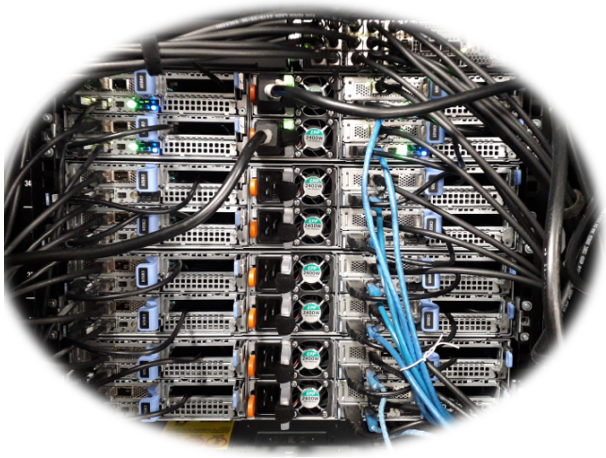
Lessons learned:

- Ethernet not “good” enough
- Software stack immature
- But: Basic configuration needs became clear

  (Slurm configuration, file system mounting, OS incompatibilities, . . .)

DINE 2.0

Durham University
Department of Computer Science



DINE (name due to Alastair Basden)

Funding:

- ExCALIBUR's H&ES testbed funding
- DiRAC
- Championed by Alastair Basden (DiRAC/ICC)

System changes:

- Infiniband
- 24 nodes

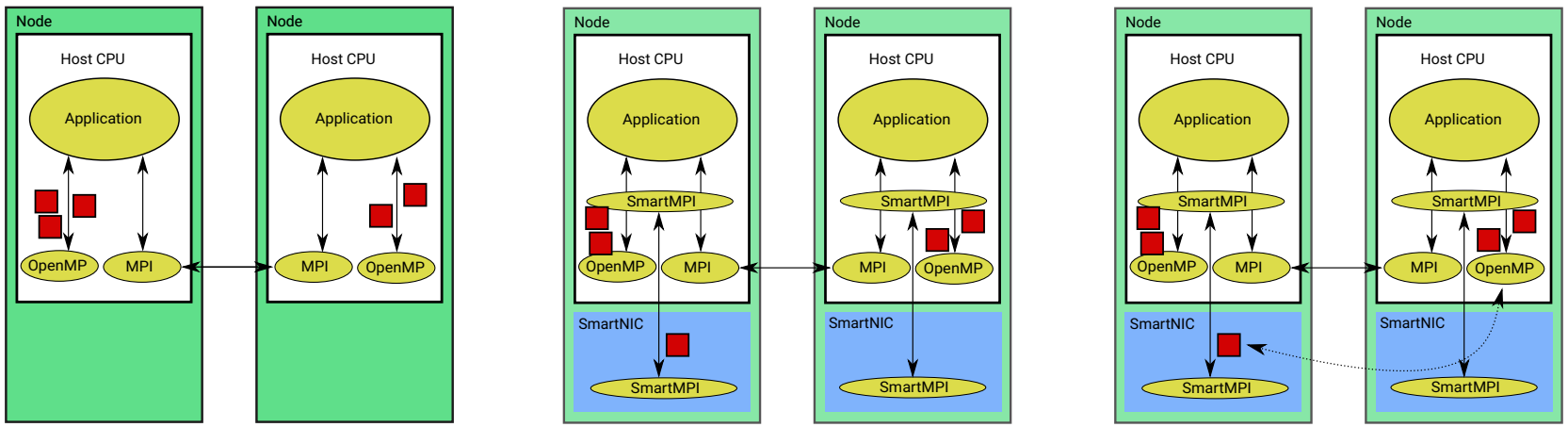
Machine usage:

- Free for any UK/EPSRC scientist
- Ask for access otherwise
- Used for prototyping within ExCALIBUR
- Used for collaboration with NVIDIA Networking

Smart fixes on smart hardware

(by smart guys)

Durham University
Department of Computer Science



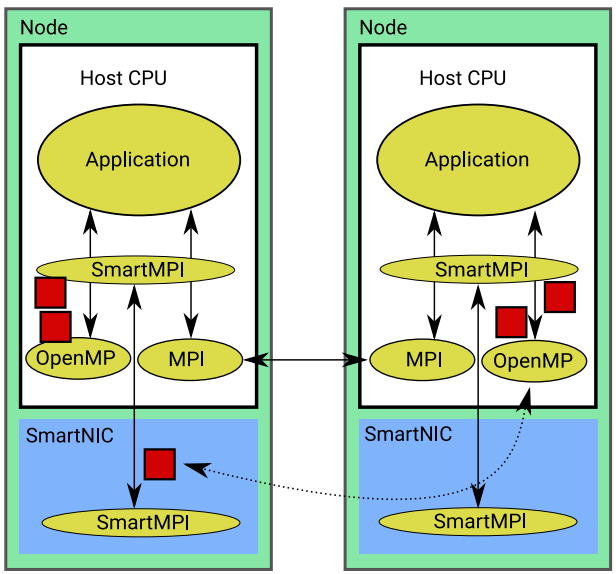
- Nodes offload (some) ready tasks to smart device
  (non-ready tasks reside on node)
- Smart network deploys tasks to underutilised nodes
- Smart device gets task outcomes back and injects them back into results queue

$\Rightarrow$ Task migration hidden

$\Rightarrow$ Network-as-compute-server paradigm

Software architecture 0.1

Durham University
Department of Computer Science



Static design:

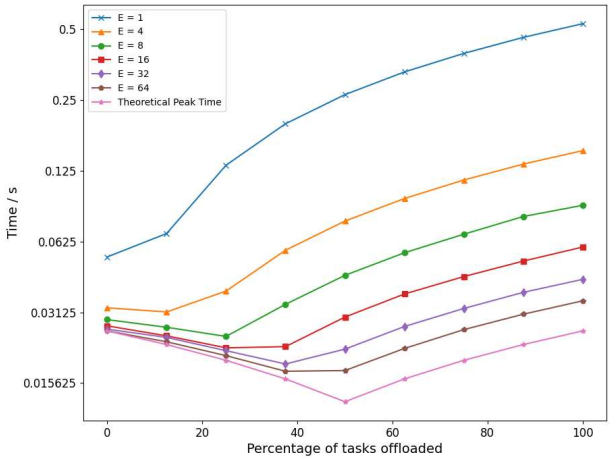
- Dedicated MPI ranks on BlueFields
  (heterogeneous MPI)
- SPMD technically, 2x SPMD logically
  - One huge if at start of code
  - Different MPI communicators
    (host-host, bf-bf, host-bf)
  - Impose 1:1 correlation bf-host
- MPI everywhere
  - Host-to-host
  - BlueField-to-BlueField
  - Host-to-its-BlueField

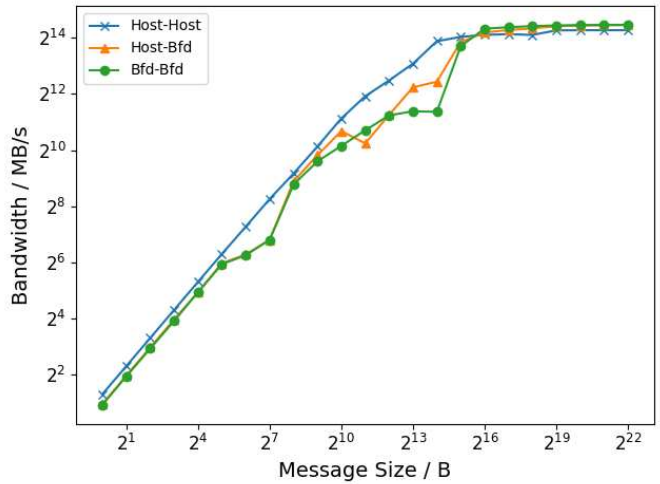
Dynamic design:

- BlueField polls its host (`Iprobe`)
- Non-blocking allreduce for load balancing on host
  (decide how many tasks to throw into network)
- BlueField uses allreduce outcome to route/deploy tasks
- Idling hosts poll their Blue Field

Software architecture 0.1—break down

Durham University
Department of Computer Science





+ Coordination can be hidden
+ Additional data movements can be hidden
- Relative latency problematic, as tasks tiny
- PCI bus speed
o Heterogeneous MPI runs challenging (OS, libraries, . . .)
o Have to bundle multiple tasks (E in graphs)

$\Rightarrow$ Unlikely viable solution

Outline

Durham University
Department of Computer Science

The science case

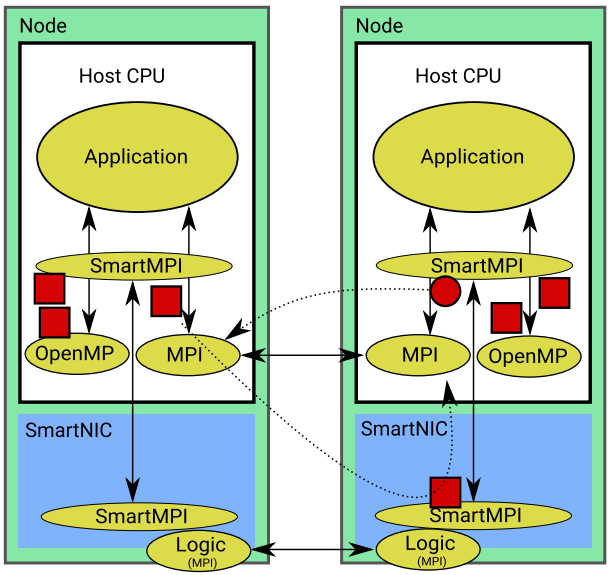
A smart team of MPI ranks and three smart ideas

Intelligent network environments

Next-gen software architecture

No single paradigm

Durham University
Department of Computer Science



- Paradigm
  - Bluefield owns/orchestrates communication
  - Bluefield owns decision logic
  - Bluefield does **not** move (deterministic) data
- Realisation on BlueField
  - Bookkeeping (non-blocking reduction, routing)
  - Trigger data transfer on host
  - Receive and cache unexpected messages (resiliency)
- Software stack
  - Remote procedure calls
  - Persistent threads using MPI between BlueFields
  - ...

  $\Rightarrow$ Not as clean anymore

Summary

Durham University
Department of Computer Science

- Can we roll our tasking out over many ranks?

⇒ Yes

- Can we benefit from multi-rank tasking?

⇒ Possible, but requires a lot of tweaking atm

- Can we make it robustly fast?

⇒ Maybe if network accepts responsibility

Financial support:

Durham University
Department of Computer Science

ExCALIBUR

DiRAC