Durham University
Department of Computer Science

Intelligent algorithms on intelligent networks—experiences and challenges using NVIDIA's BlueField technology

ISC 2023—DPU Workshop

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

[x] DiRAC, Durham University
[+] NVIDIA Networks
[*] Computer Science, UCL
[+] Computer Science, Durham University
[⊤] AMD, Germany
[⊥] Advanced Research Computing (ARC), Durham University

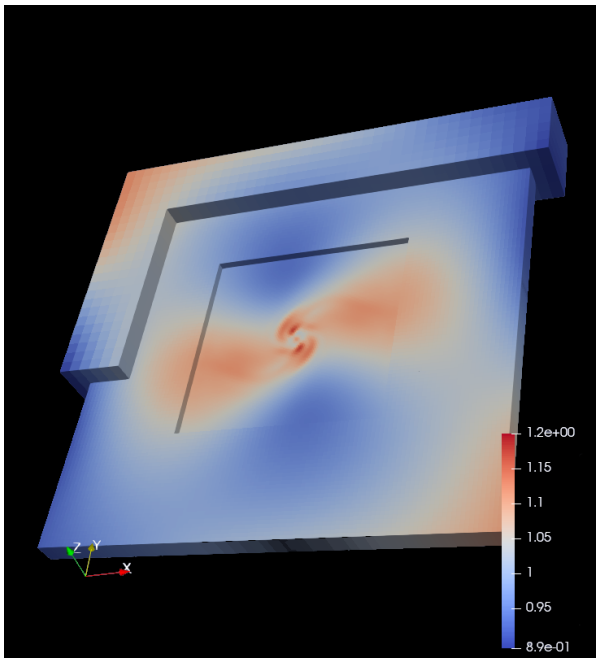
May 25, 2023

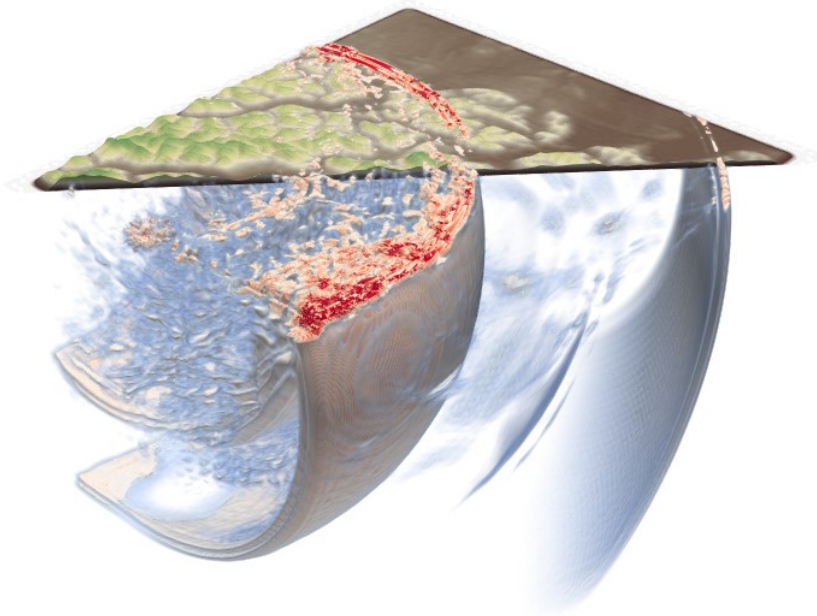
Outline

Durham University
Department of Computer Science

Application challenge

Durham University
Department of Computer Science





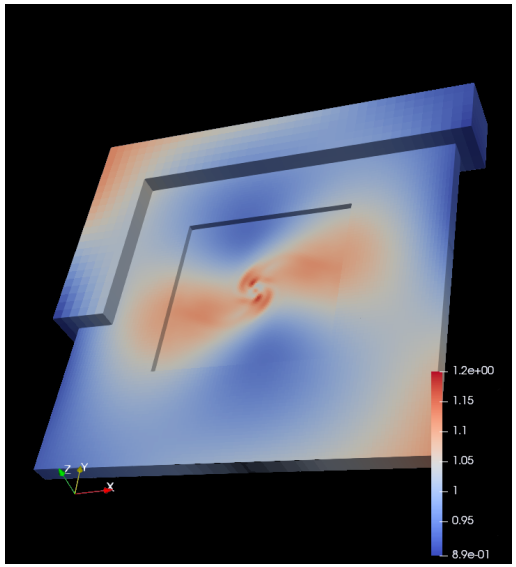
Work by groups of Baojiu Li and Han Zhang (Institute for Computational Cosmology)

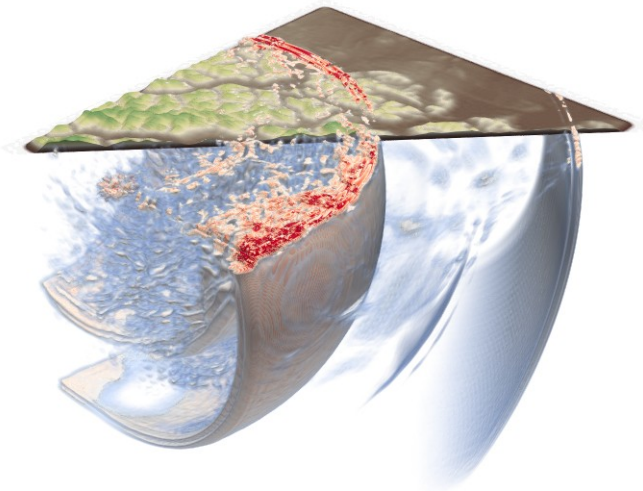
Chris Johnson: “Before every great invention was the discovery of a new tool”[+].

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

Application challenge in British English

Durham University
Department of Computer Science





Work by groups of Baojiu Li and Han Zhang (Institute for Computational Cosmology)

> 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.”

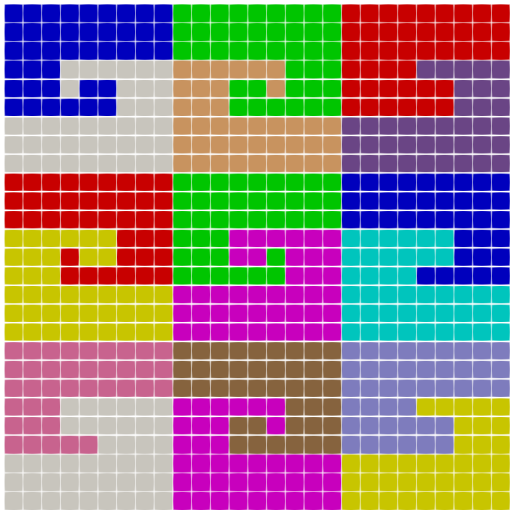
Thanks to David E. Keyes for pointing to this quote.

ExaHyPE—a tool with MPI+X+X+X

Durham University
Department of Computer Science

$$\frac{\partial}{\partial t} Q + \nabla \cdot \mathbf{F}(Q) + \sum_i \mathcal{B}_i \frac{\partial Q}{\partial x_i} = \mathbf{S} + \sum \delta$$

- Various numerical schemes
- Adaptive Cartesian grids
- Strict separation of concerns
- Python API
- Parallelisation hidden
- ...



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

MPI SFC-based non-overlapping domain decomposition

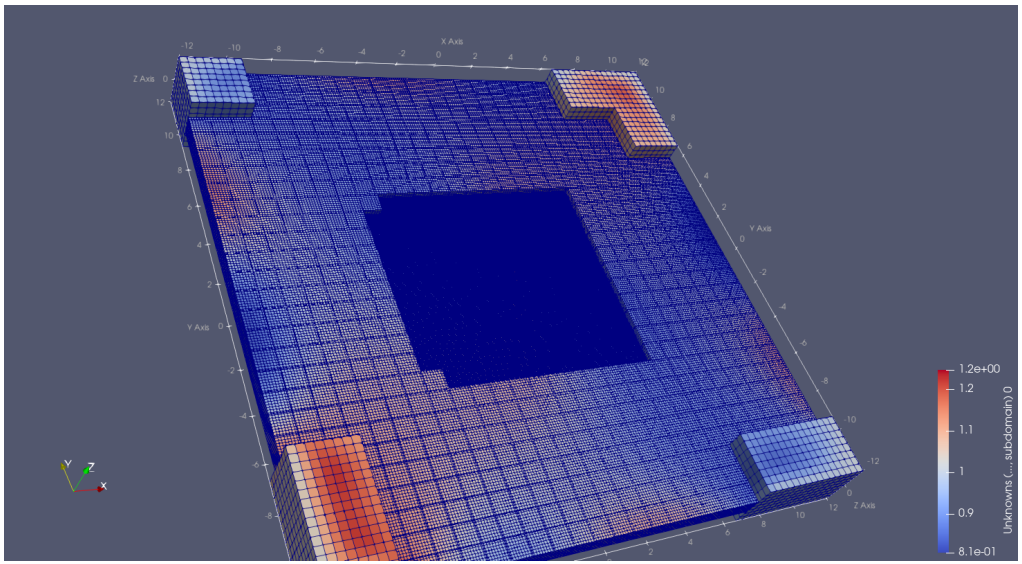
Threads SFC-based non-overlapping domain decomposition
(OpenMP, TBB, C++, SYCL)

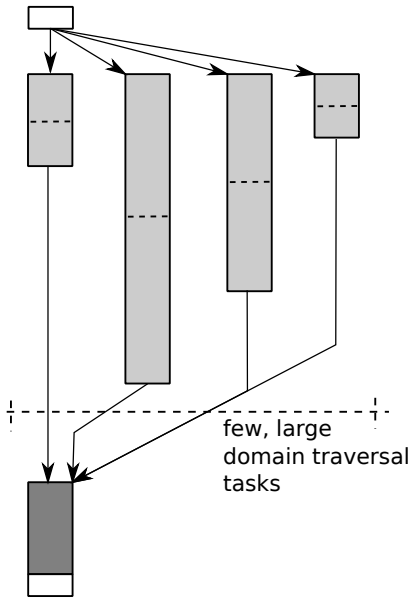
Threads Additional tasks spawned per SFC partition
(OpenMP, TBB, C++, SYCL)

GPUs GPU offloading by different tasks
(OpenMP, SYCL, experimental (NVIDIA) C++)

Fork-join is intrinsically problematic

Durham University
Department of Computer Science



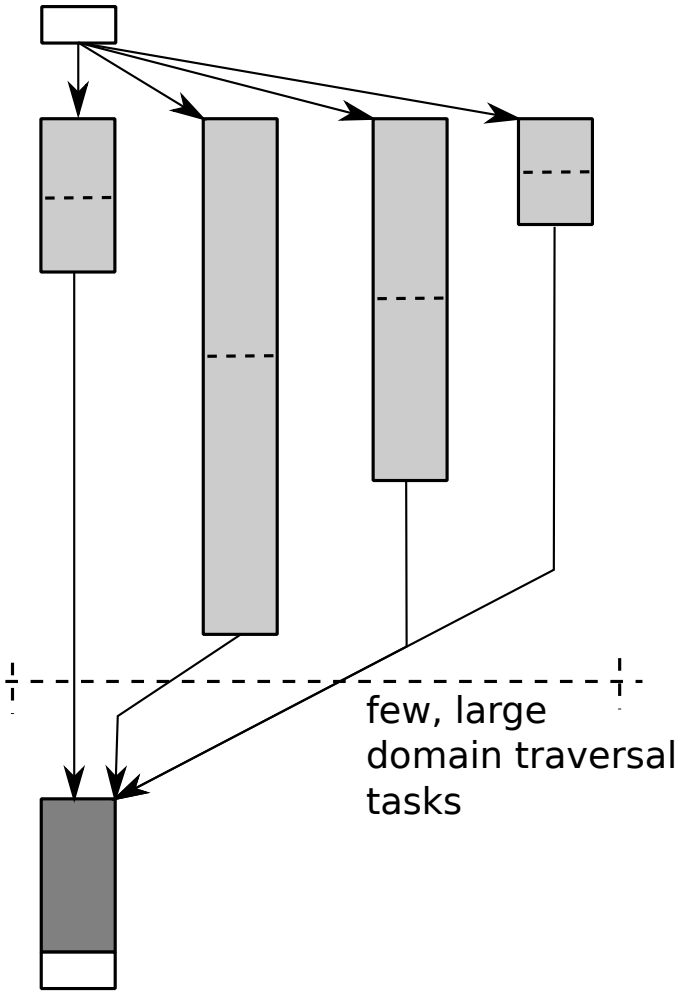


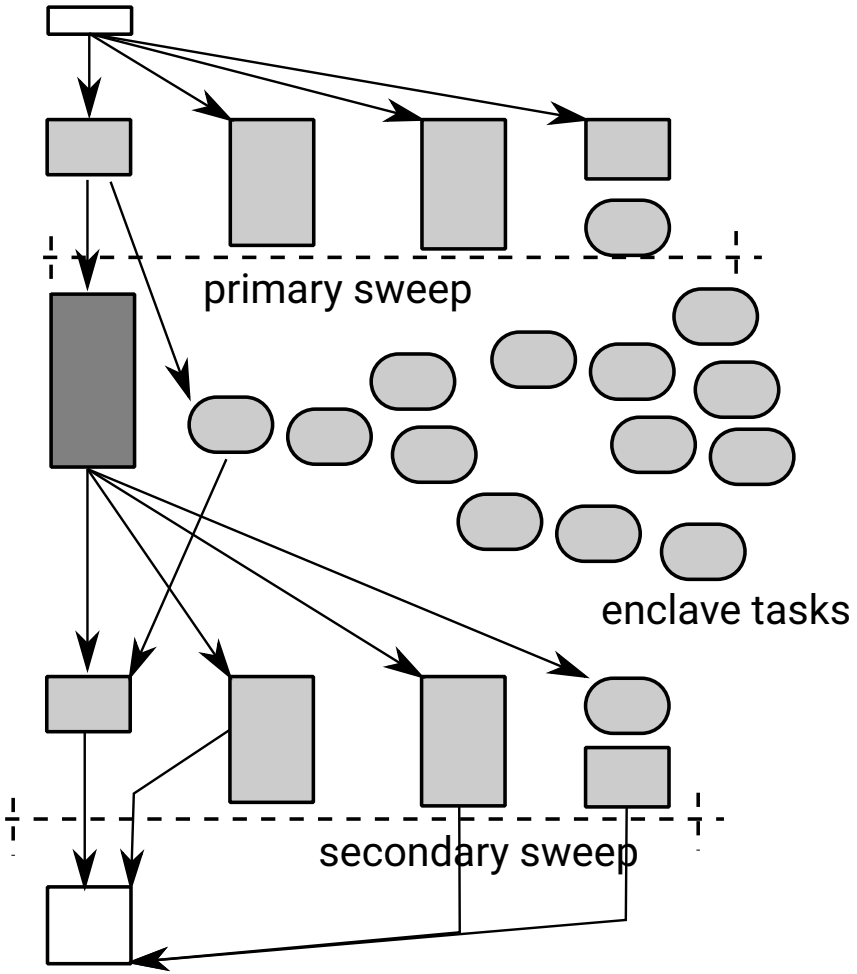
Left: Snapshot from a debugging session for rotating binary black holes (Han Zhang, Baojiu Li)

Right: From the SISC paper on enclaves; cmp. next slide (Dominic E. Charrier)

The enclave task concept

Durham University
Department of Computer Science





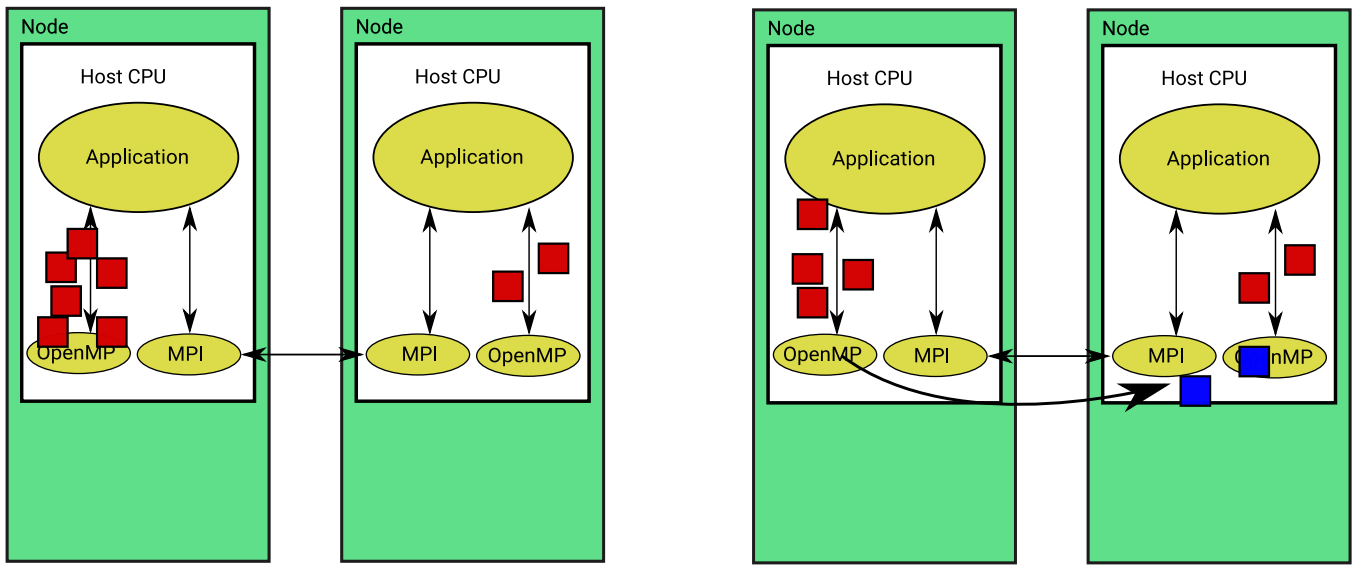
D.E. Charrier, B. Hazelwood, T. Weinzierl: Enclave Tasking for DG Methods on Dynamically Adaptive Meshes. SISC 42(3) (2020)

Outline

Tasking in a distributed world

Durham University
Department of Computer Science

Research vision: Migrate tasks to other ranks, bring results back (non-persistent load balancing).



Can tasks work in MPI environment?

Durham University
Department of Computer Science

> Research vision: Migrate tasks to other ranks, bring results back (non-persistent load balancing).

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)

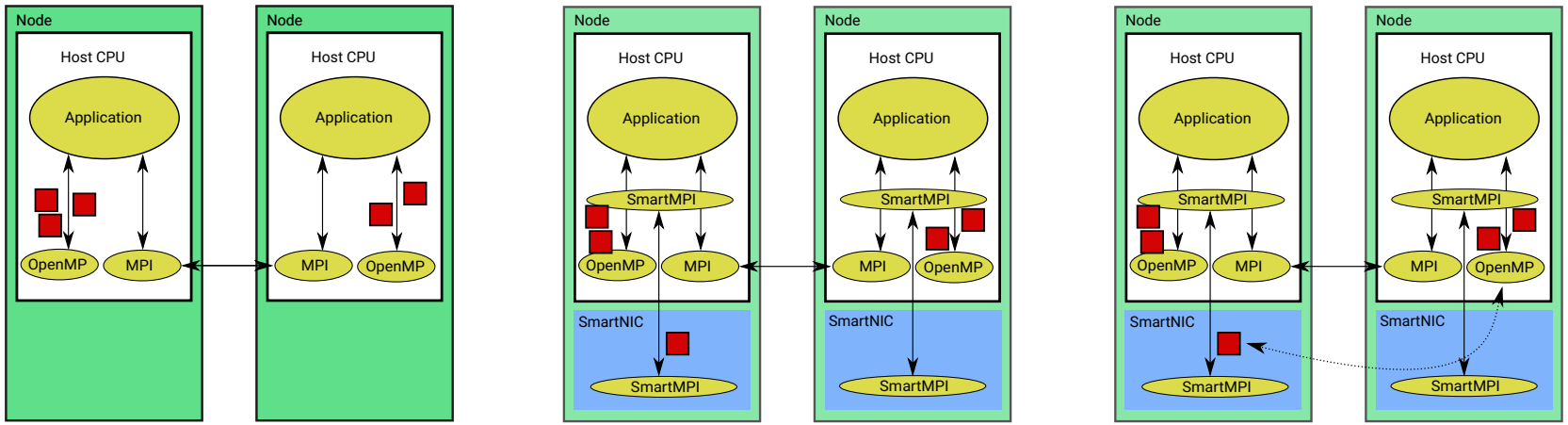
Bad “news”:

- Overlap does not happen
  (MPI progression issue)
- Overhead too high
  (task administration)
- Orchestration interrupts CPU’s work
  (cycles sacrificed for core MPI/scheduling work)
- Network stress increases
  (vicious cycle of congestion and additional offloading)

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 into results queue

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

Alternative: Node-as-accelerator paradigm (see work by Legg and Yates on work how to place scheduler into network: network champions algorithm and CPUs become compute servers; cmp. suggestions of M. Schulz earlier today)

Outline

Durham University
Department of Computer Science

The science case

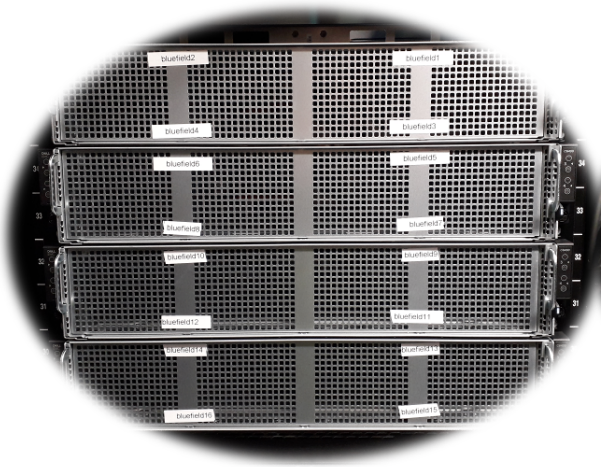
Tasking in a distributed world

Durham's Intelligent Network Environment

Software architecture and usage model

DINE 1.0

Durham University
Department of Computer Science



DINE (name due to Alastair Basden)

Durham Intelligent NIC Environment:

- Trigger: ISC 2016
  (at Mellanox booth)
- Seedcorn funding by Strategic Investment of Durham CS
- Team up with ICC/DiRAC staff to install nodes
  (you need brains and racks)
- 16 nodes AMD EPYC
- 16 BlueField-1 cards (Ethernet)

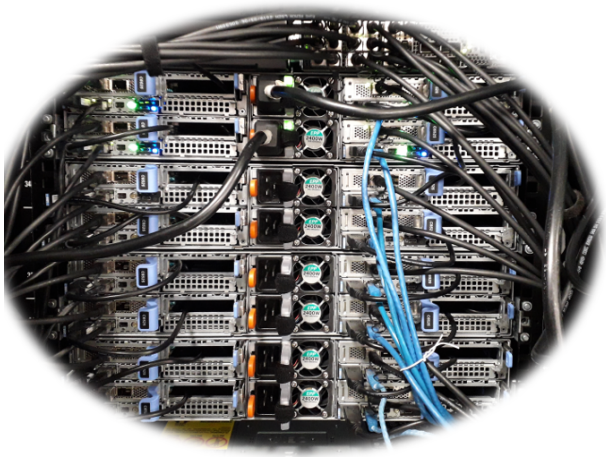
Lessons learned:

- Ethernet not “good” enough
- Software stack immature
- Basic configuration needs became clear
  (Slurm configuration, file system mounting, OS incompatibilities, . . .)

$\Rightarrow$ Eventually used as “normal” prototyping cluster

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

Outline

Durham University
Department of Computer Science

The science case

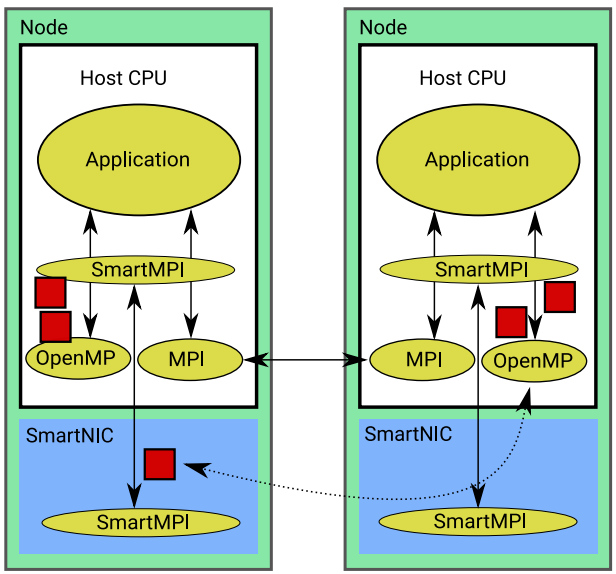
Tasking in a distributed world

Durham's Intelligent Network Environment

Software architecture and usage model

Usage model

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

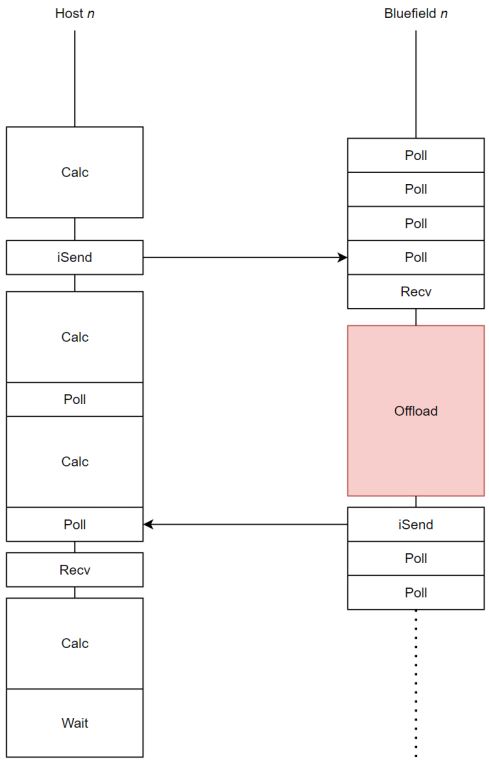
Dynamic design:

- 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 bf

Data flow

Durham University
Department of Computer Science

- Enclave tasking ⇒ D.E. Charrier
  ⇒ works only within node
- Task migration ⇒ P. Samfass
  ⇒ MPI progression a nightmare and software tied to ExaHyPE
- Black-box library ⇒ M. Turner
  ⇒ code cleaned up but not yet smart
- BF port ⇒ J. Moore



Schematic data flow sketch lacking recursion and offloading from BF to under-subscribed node

Status quo and research questions

Durham University
Department of Computer Science

Does smart technology work?

- Ingredients all there
- Progression issues eliminated
- Different protocols do not work (yet)

Can idea work?

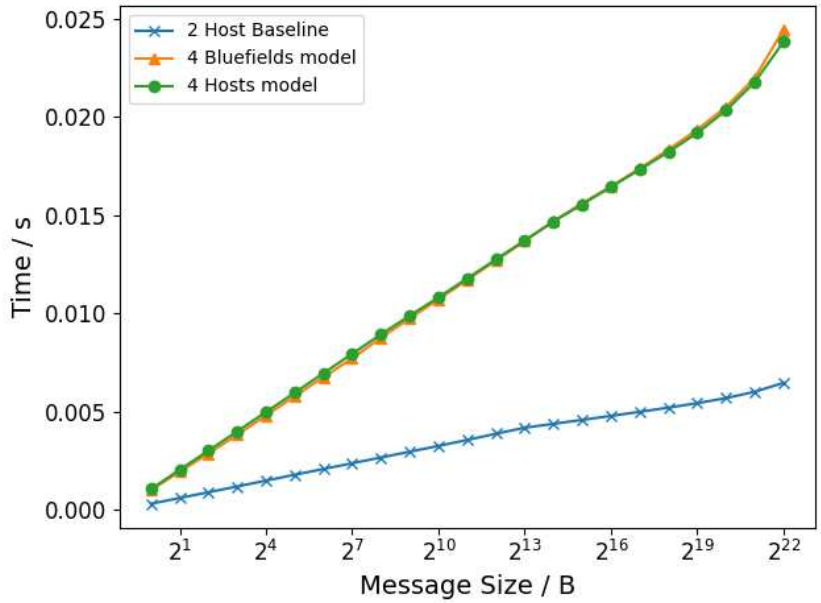
- Simple performance model

$$
\begin{aligned}
T &= \sum_{i=1}^{N} T_i^{(\text{task})} \approx N \cdot T^{(\text{task})} \\
T &= N^{(local)} \cdot T^{(\text{task})} + (N - N^{(local)}) \cdot \left(T^{(\texttt{isend})} + T^{(\texttt{recv})} \right) \\
&\quad + K \cdot T^{(\texttt{iprobe})}, \quad N^{(local)} \leq N
\end{aligned}
$$

- Calibrate entries
- Identify “working” hardware and task properties

Overheads

Durham University
Department of Computer Science



x-axis: total data size split up into tasks or given size

Setup:

- Baseline: Two compute nodes
- Four BlueFields: Study only bf-bf part of (host-bf-bf-host)
- Fake machine: two AMD nodes as compute nodes, two as “SmartNICs”

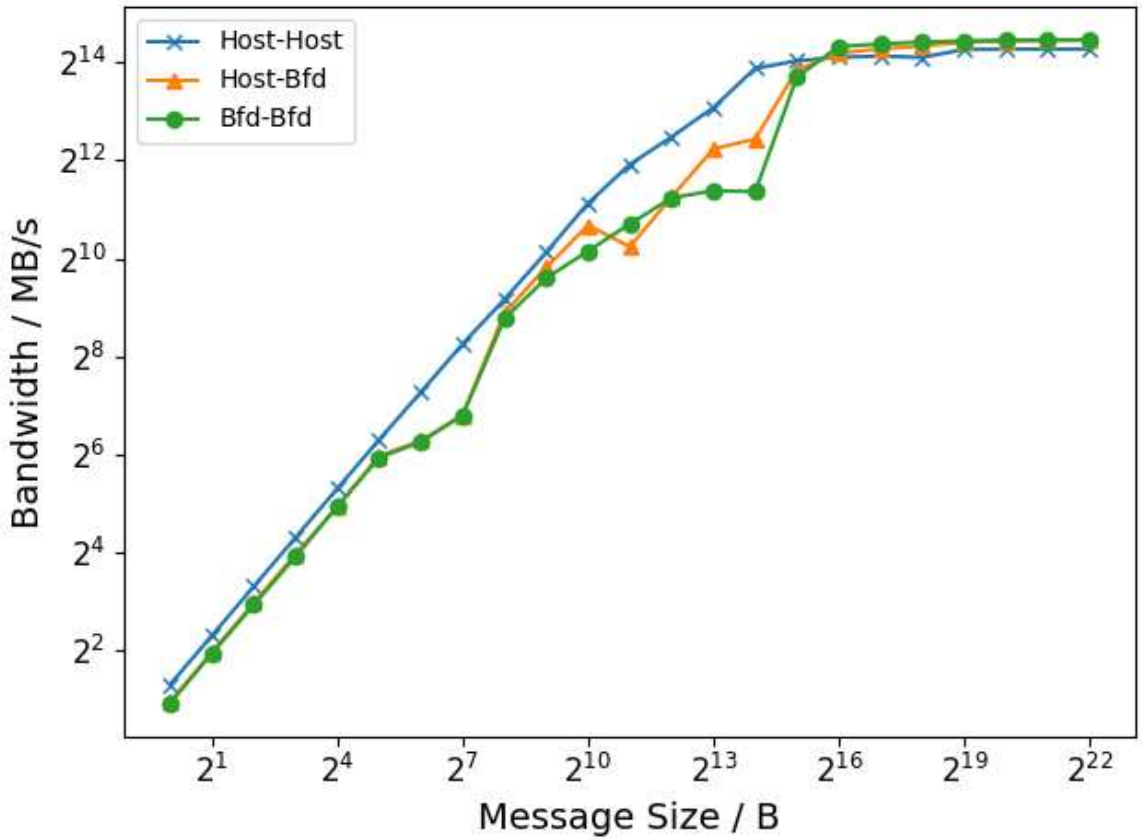
Insight:

- Additional two hops harm
- BF-BF is not to blame

$\Rightarrow$ but we want to manage to hide this extra cost anyway

Calibration

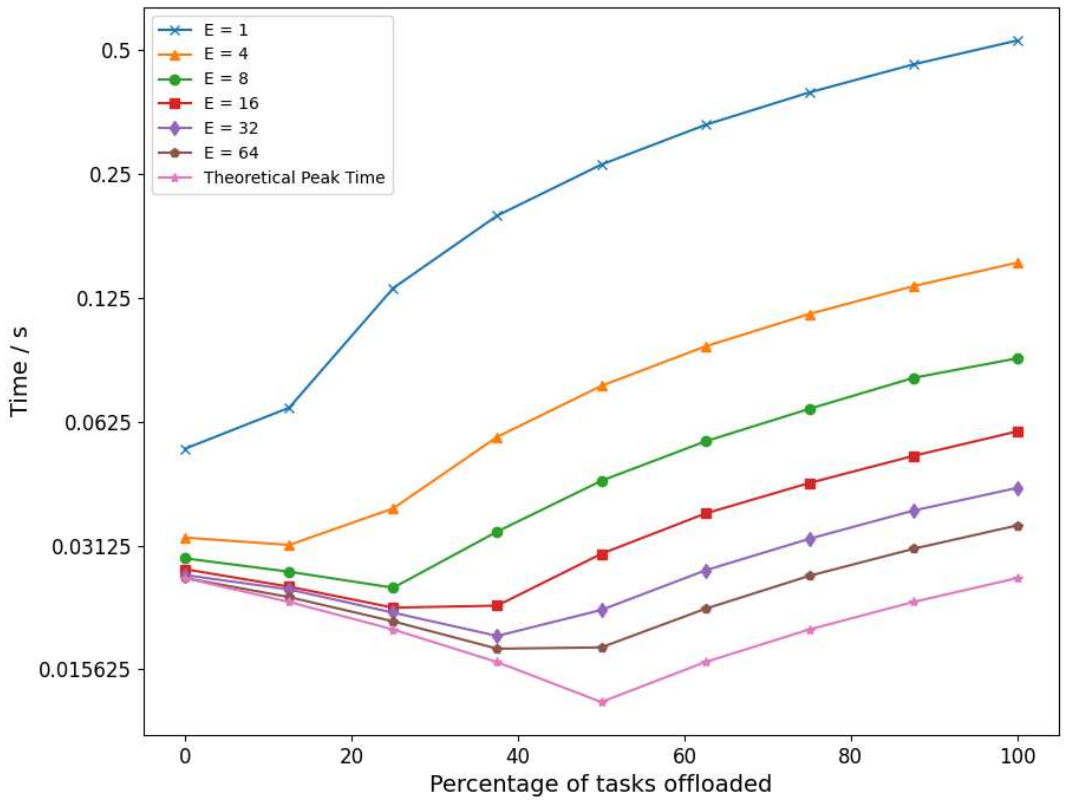
Durham University
Department of Computer Science



- BF is only slightly weird

Big picture

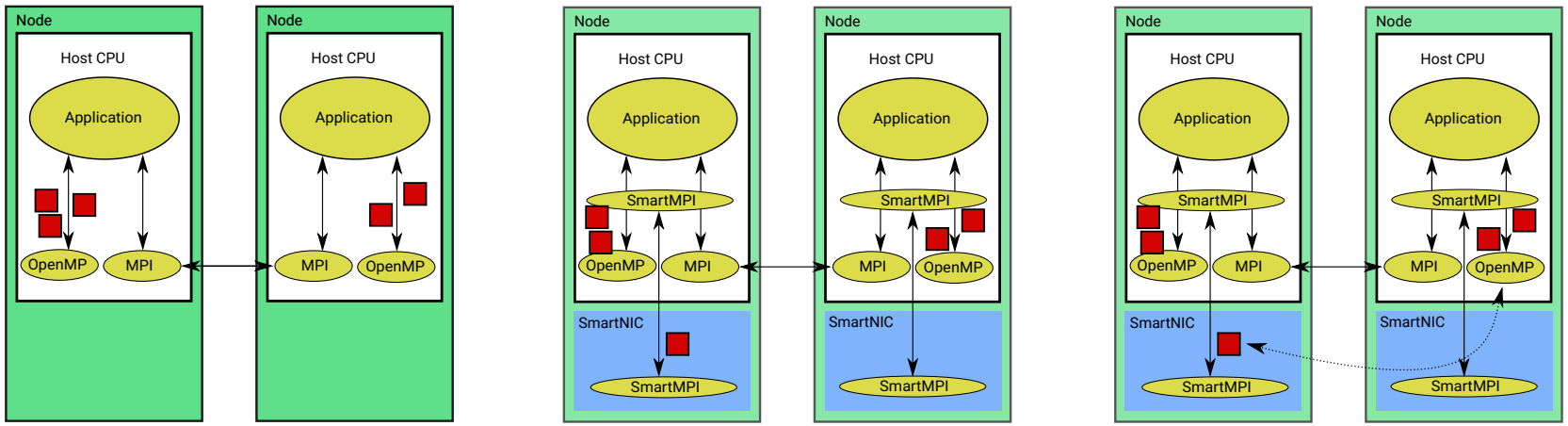
Durham University
Department of Computer Science



- Two node setup (host-victim scenario): Optimal speed-up is 50% runtime reduction
- Ratio of task cost to transfer time key
- Don't poll too often (have enough local work)

Smart solutions on smart hardware

Durham University
Department of Computer Science

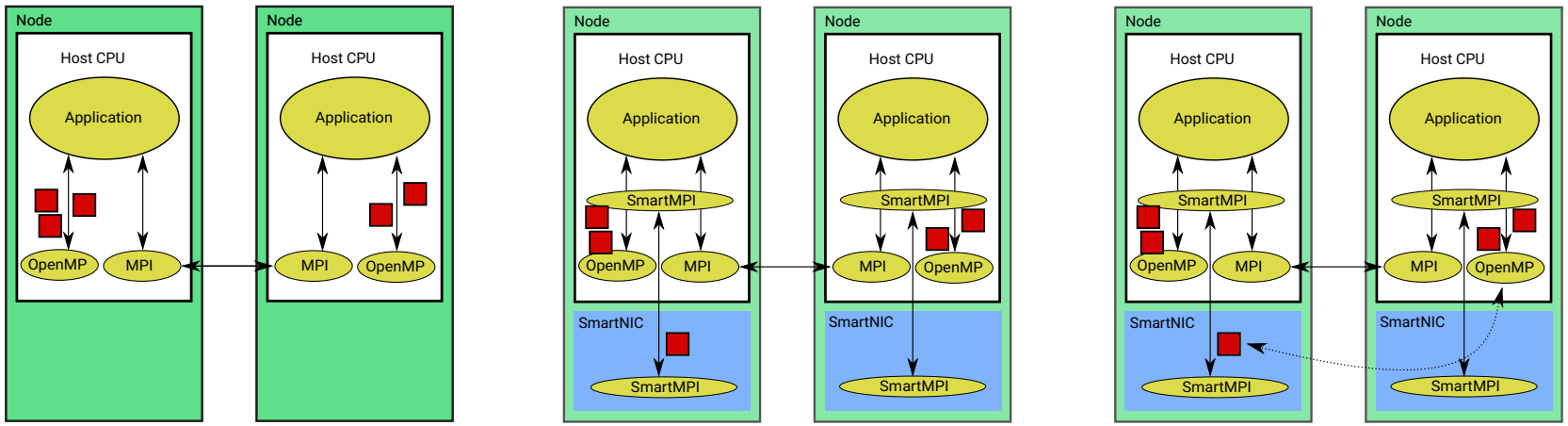


- Good:
  - Software ingredients all in place
  - Performance benchmarking suggests is should work
- Bad:
  - Integration not yet possible
  - Enormous man power needs
  - Standard USPs not relevant to us (reduction, security, I/O, . . .)
  - DOCA programming model not 1:1 fit to our vision

  $\Rightarrow$ we hvae to rewrite our code
- Ugly:
  - Surprises around the corner
  - My budget

Smart solutions on smart hardware

Durham University
Department of Computer Science



- Good:
  - Software ingredients all in place
  - Performance benchmarking suggests is should work
- Bad:
  - Integration not yet possible
  - Enormous man power needs
  - Standard USPs not relevant to us (reduction, security, I/O, . . .)
  - DOCA programming model not 1:1 fit to our vision
  - $\Rightarrow$ we hvae to rewrite our code
- Ugly:
  - Surprises around the corner
  - My budget
- But: All changes once
  - network can actually compute (resiliency, floating-point compression, load balancing, . . .)
  - $\Rightarrow$ maybe abandon sole offloading paradigm eventually
  - network can dynamically grow (GPU) cluster
  - $\Rightarrow$ if data parallelism stagnates, functional decomposition has to ride to our rescue