Offloading tasks to intelligent hardware

M. Turner (mark.turner@durham.ac.uk), T. Weinzierl, H. Schulz, P. Samfass

Motivation

Many HPC codes suffer from:

(1) bandwidth/latency constraints
(2) lacking message progression
(3) Limited reactivity to performance flaws at runtime

smarTeaMPI

We introduce smarTeaMPI:

Users spawn tasks within the library which will either:

(1) run the task locally
(2) **project the task intelligently elsewhere** in the cluster

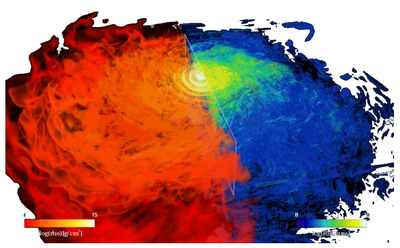
smarTeaMPI = a generic library for ***nonpersistent loadbalancing*** (Samfass et a., 2020) of task based codes at runtime

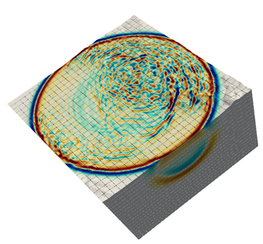
Where is smarTeaMPI used?

ExaHyPE (Weinzierl, 2019)

Engine for hyperbolic PDEs

Major applications in astrophysics and seismology





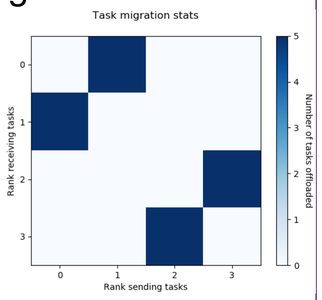
From the ExaHyPE gallery (www.exahype.eu)

smarTeaMPI: the basics

API outline:

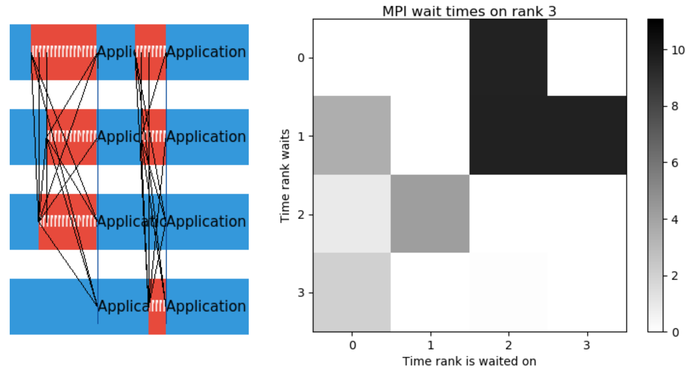
(1) Implement the smartmpi::Task interface to make 'smart' tasks
(2) Spawn smart tasks within smarTeaMPI
(3) Reports MPI wait times to smarTeaMPI
(4) call smarTeaMPI's 'progress' functions

smarTeaMPI encapsulates the complexity of task migration (right: the lib's visualisation of a simple swapping scheduler) and the deduction of optimal task routing patterns



MPI Wait Times

Ranks maintain MPI wait time statistics:



Following Samfass et al. (2020), smarTeaMPI finds:

(1) critical ranks: delay but don't wait
(2) victim ranks: never delay but wait

Smart ExaHyPE

Objective #1:

Use smarTeaMPI at runtime to intelligently migrate tasks nonpersistently
--- address load imbalance at run time

Setup:

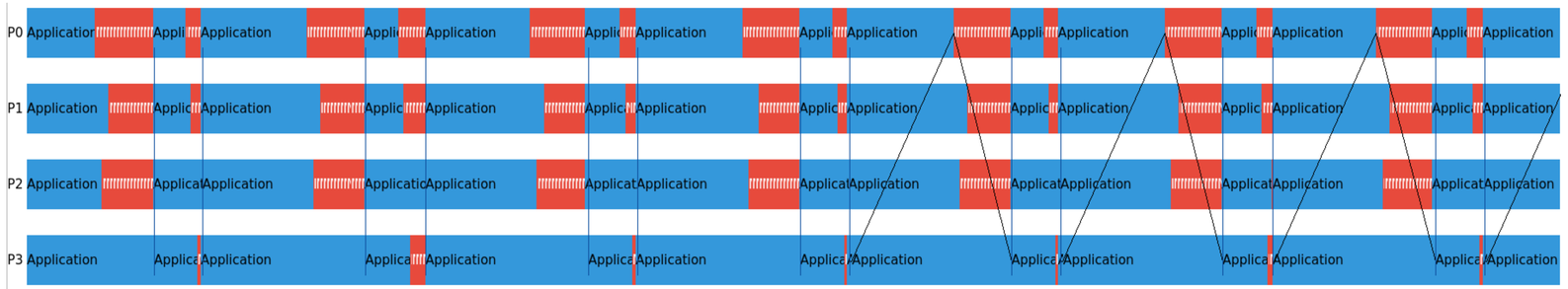
Euler 3D + Adaptive Mesh Refinement + artificially imposed load imbalance

Early results: case with 10% speed up

How does smarTeaMPI respond?

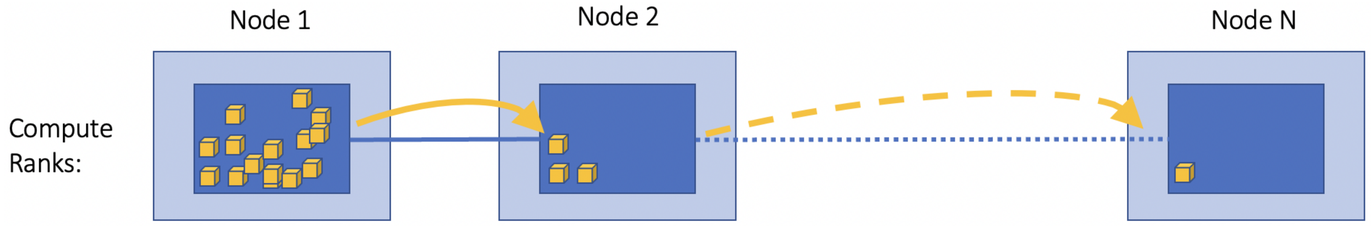
In ITAC's view below, smarTeaMPI:

(1) Spends initial time steps **learning** about the setup
(2) **Observes** rank 3 = critical rank
(3) **Offloads** tasks from rank 3
(4) Ensures tasks are computed on rank 0 (the 'optimal victim')
(5) **Coordinates** the returning of results to rank 3 for bookmarking

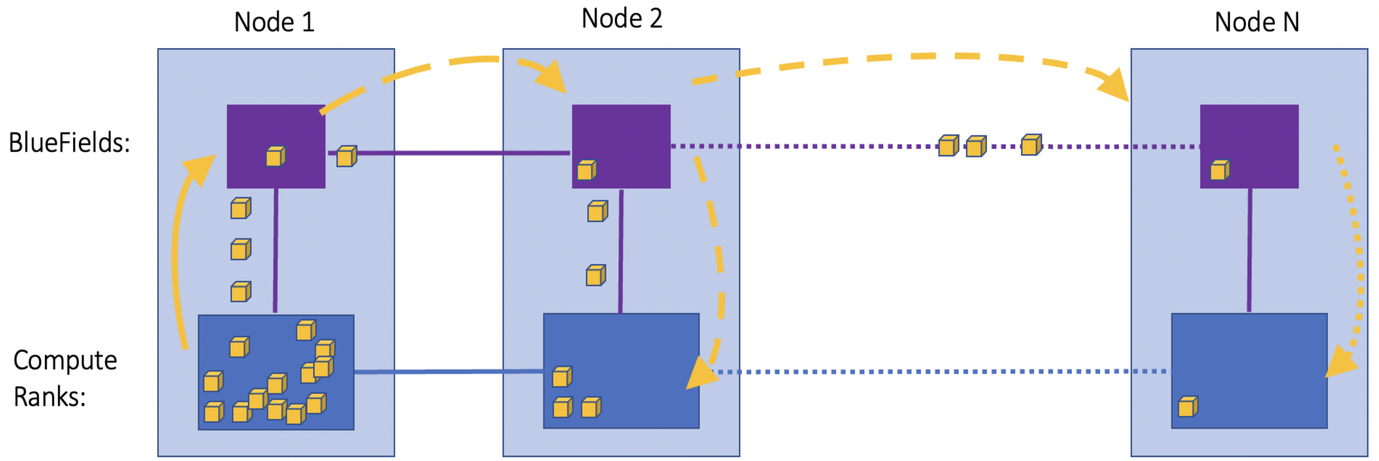


Smarter MPI with smarter hardware

Task migration on traditional compute clusters:



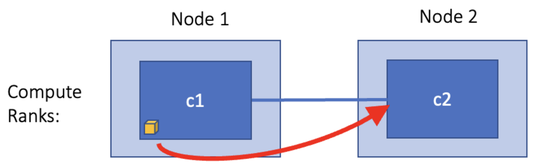
Task migration on smart clusters with smarTeaMPI's alternating topology:



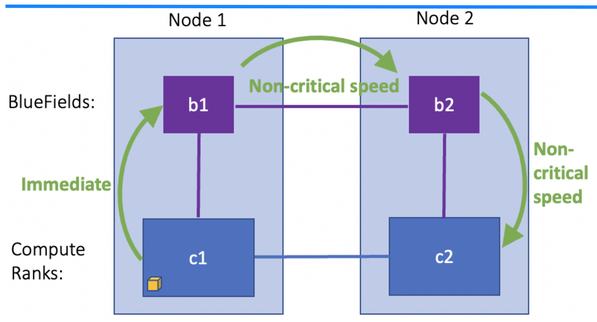
smarTeaMPI leverages evolving Smart NIC technology (NVIDIA **BlueFields**) to increase code **resilience** by projecting a highly reactive software layer into the **fabric** of cluster supercomputers

Aim: driving increased reactivity with smarter fabrics

Comparison: sending tasks between compute ranks with/without BFs



Stresses MPI on c1 and c2: c1 has to trigger an Isend
---> **Risk**: this becomes rendezvous
---> Scheduling/analytics/migration has to fit around

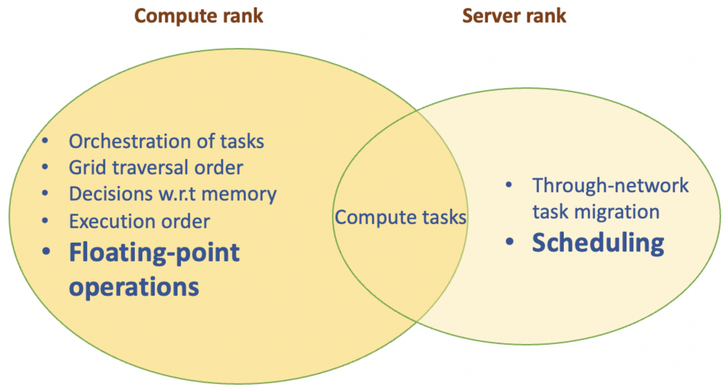


By design, BFs are always ready to receive so compute ranks can offload instantly. Now a compute rank only delays a server if they are slow to receive a smart task
---> The disposibility of the NIC = a key advantage in that it **leaves the compute ranks uninterrupted in their focus on computational throughput**

Functional differentiation

smarTeaMPI encapsulates **topological information**, distinguishing server and compute ranks. The library uses this schema to organise MPI processes over **heterogeneous compute clusters** that are equiped with programmable network cards - e.g. NVIDIA BlueFields.

Result = separation of concerns:



Compute ranks **--->** throughput
Server ranks (on BFs) **--->** reactivity

We envisage the server ranks taking on greater compute responsibilities as NVIDIA DPUs become more powerful (and acquire GPUs of their own)

References

[1] Samfass et al. Lightweight task offloading exploiting MPI wait times for parallel adaptive mesh refinement. Concurrency and computation : practice and experience, 32(24), 2020.

[2] Tobias Weinzierl. The Peano Software - Parallel, Automation-Based, Dynamically Adaptive Grid Traversals. TOMS, 45(2), 2019