Hardware Topology Working Group

WG Statement:

As hardware is rapidly evolving, a wide spectrum of hardware features became available: complex memory hierarchies, accelerators, many-core processors, FPGAs, MPI on hardware, non-coherent memory, heterogeneous architectures, etc. As a consequence, understanding such features and exploiting them at the MPI application level (e.g. data locality) is a challenge that applications envision to address directly. Indeed, MPI still relies on (or enforces) a flat programming model, which is became less relevant with the advent of the heterogeneous manycore era. Despite recent modifications to the standard to address these scalability issues (e.g. distributed graph interface, neighborhood collectives, etc.), the challenges of hardware topology discovery and management have been left unaddressed by the MPI Forum so far.

The main question this Working Group proposes to address is the following: how can hardware resources (I/O, cores, caches, I/O proxies, etc.) be discovered, queried upon and distributed between execution flows? Indeed, MPI processes are bound to span multiple cores and no portable primitives exist outside, nor inside, MPI to explore and take advantage of the hardware topology either at the node or that is made available at the process level.

Assuming the need to expose such information to the application, here is a more detailed list of questions to be addressed in the working group:

- What level of abstraction is desirable in order to keep the hardware-independence of MPI effective?
- Which MPI objects/constructs could benefit and exploit such a knowledge (e.g groups, communicator, shared-memory windows, etc)?
- How can we envision interactions between MPI and some other programming model/interface (e.g. OpenMP) with regards to hardware topology utilization?
- How can this WG interact with other WGs that already started to address the same kind of issues, especially for shared-memory (e.g. the MPI_COMM_TYPE_ADDRESS_SPACE proposal)?
- How does this impact other discussed proposals, such as sessions and endpoints for instance?
- More generally speaking, what kind of interactions are expected with the hybrid WG?
- What should be the relationship between the proposed interface and the mapping/binding mechanisms provided by process managers and/or RJMS? Can we (should we?) standardize this also?

Despite such a wide range of questions, answering them is critical for the understanding of hierarchical notions at the application level. We plan to first address the hierarchical communicators interface proposal, as presented informally at the MPI Forum meeting in Portland (Feb 2017) by Inria, and from there open the discussion to the other topics mentioned above.

We envision open and explorative discussions around the hardware topology discovery and management idea as members will have the opportunity to provide their existing

knowledge and feedback on the related topics:

- Inria: hwloc and topology discovery knowledge, MPI development issues, interface prototyping, use-cases
- UTK: integration with MPI implementation, integration with different other MPI concepts (process placement, collective communications)
- CEA: knowledge in hybrid MPI+threads programming model to explore further ideas on model mixing (in terms of pure locality)
- ParaTools SAS: knowledge in profiling tools and possible scenarios including collocation
- ATOS/BULL: knowledge in MPI implementations (hierarchical aware collectives, shared memory communication with high NUMA factor) and future architectures including hardware MPI implementations, bringing constraints from the integrator, use-cases

Institutions supporting WG creation

So far, the following institutions/vendors are supporting this WG creation:
• Atos/BULL
• CEA
• Inria (proposed coordinator/animator of the WG)
• Paratools SAS
• University of Tennessee, Knoxville

Current Resources

• Interface prototype: the hsplit library: http://mpi-topology.gforge.inria.fr
• Research report: https://hal.inria.fr/hal-01538002v3