ISC 2018 05/26/2018

Hardware Topologies Working Group

Motivations

- New architectures become increasingly complex
  - Memory hierarchy
  - Numa effects
- Application developpers need features to:
  - Deal with hardware characteristics (Caches, Interconnect, Cores, NUMA nodes, etc.)
  - Deal with low level tools (hwloc, libnuma, etc.)
- Expected performance improvements
  - Improved locality
  - Improved communication performance

Working Group Statement

- MPI is hardware-agnostic
  - And should remain so
  - But doesn’t prevent nor encourages the application to access the underlying HW
- Issues
  - How to discover HW resources in a MPI application?
  - How to leverage the HW resources?
- Questions
  - What is the right level of abstraction?
  - Which MPI constructs could leverage HW topologies?
  - What are the interactions with other programming models?

Three Directions Discussed

- The *implicit* access to the HW topology
  - The HW topology can be accessed through MPI abstractions
- The *explicit* access to the HW topology
  - A HW topology description can be accessed by the user directly
- The mapping and binding of MPI processes
  - Borderline, but a very important point
  - Related to process managers/RJMS

Implicit access to HW topologies

- Current proposal:
  - Creation of so-called hierarchical communicators
    - A communicator corresponds to a specific level in the HW hierarchy
  - Based on the `MPI_Comm_split_type` function
    - Introduce a new split_type value: **`MPI_COMM_TYPE_PHYSICAL_TOPOLOGY`**
- Prototype implementation available: Hsplit
  - External library (for now)
    - Available at : http://mpi-topology.gforge.inria.fr/
  - hwloc/netloc-based implementation

Explicit access

- Determination of processes coordinates and neighborhoods
  - MPI_T interface
  - Dedicated functions (E.g. Fujitsu’s extensions)

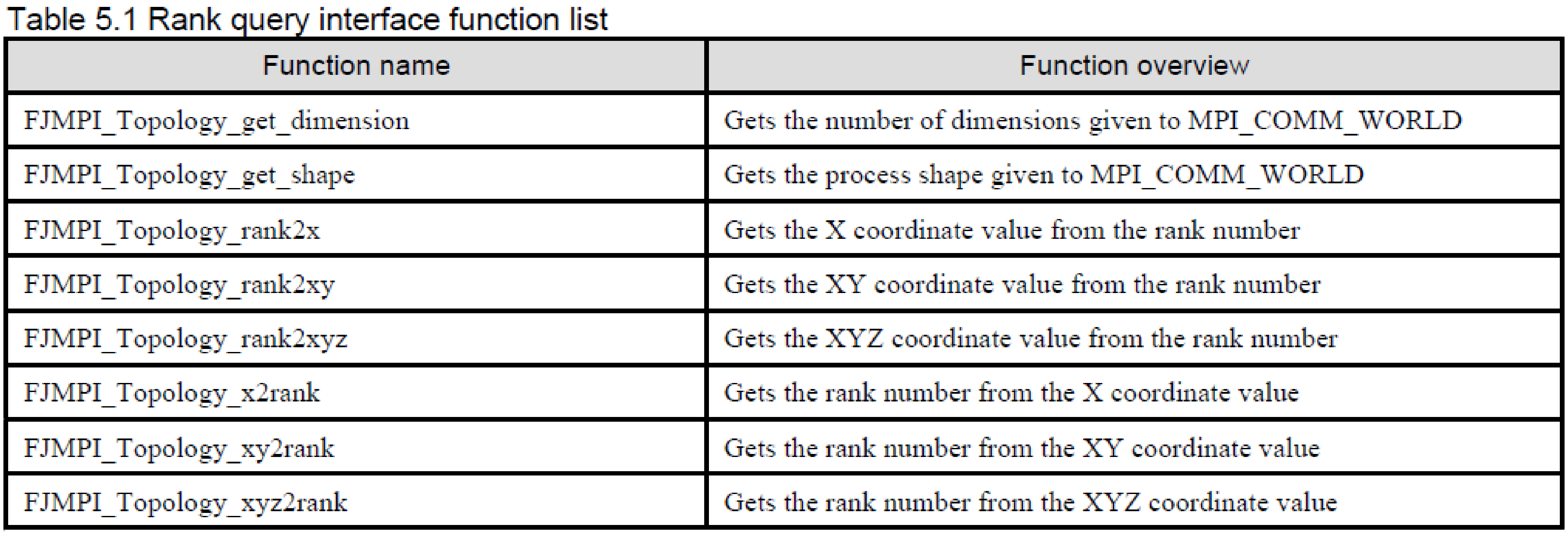

Table 5.1 Rank query interface function list

Function name	Function overview
FJMPI_Topology_get_dimension	Gets the number of dimensions given to MPI_COMM_WORLD
FJMPI_Topology_get_shape	Gets the process shape given to MPI_COMM_WORLD
FJMPI_Topology_rank2x	Gets the X coordinate value from the rank number
FJMPI_Topology_rank2xy	Gets the XY coordinate value from the rank number
FJMPI_Topology_rank2xyz	Gets the XYZ coordinate value from the rank number
FJMPI_Topology_x2rank	Gets the rank number from the X coordinate value
FJMPI_Topology_xy2rank	Gets the rank number from the XY coordinate value
FJMPI_Topology_xyz2rank	Gets the rank number from the XYZ coordinate value

Mapping/binding

- Difficult issue
  - “Outside the scope of the standard”
  - Involves RJMS, process managers, MPI applications
    - At what level (e.g MPI_Bind)?
    - Identify the possible interactions
  - Binding is easy, mapping not so
    - Even worse in hybrid, dynamic cases
- Not very user-friendly
  - Changes from one implementation version to the other
  - Not portable
- Standardize mpiexec/mpirun parameters?

Join us!

- Github: https://github.com/mpiwg-hw-topology
  - Teleconferences on regular basis
  - The minutes are available
- We need
  - Feedback from application developers
  - More use cases