Making the Case for Portable MPI Process Pinning

Balazs Gerofi <bgerofi@riken.jp>, Rolf Riesen <rolf.riesen@intel.com> and Yutaka Ishikawa <yutaka.ishikawa@riken.jp>

**Motivation and Background**
- Architectural complexity of node resources in high-performance computing (HPC) keeps increasing
- Many-core CPUs, non-uniform memory architectures (NUMA), deep memory hierarchies are becoming commonplace (e.g., Intel Xeon Phi, AMD EPYC, etc.)
- Topology aware MPI process pinning is utmost important for efficiently utilizing the underlying hardware

**Issues**
- Existing process pinning APIs are MPI implementation specific, non-standard and often overly intricate
- Non-portable job scripts across different MPI environments
- Difficulty of comparing MPI implementations due to pinning differences

**Design and Implementation**
1. fork()
   - Processes synchronize
   - Leader process elected
   - Leader creates shared memory region and collects topology information
   - Resources are divided
   - Each rank pins itself to its corresponding partition (i.e., sched_setaffinity())
   - Execute application binary (i.e., exevp())

**Evaluation Environment**
- Oakforest-PACS Supercomputer at JCAHPC (hosted by The University of Tokyo)
- Intel Xeon Phi (KNL) CPU 7250 (1 socket, 68 cores, 4 HW threads / core)
- Intel Xeon CPU E5-2690 v4 (2 sockets, 14 cores / socket, 2 HW threads / core)

**Demonstration and Experiments**
Intel MPI, Version 2018 Update 3 Build 20180411 (id: 18329):
```bash
$ mpirun -env I_MPI_PIN_DOMAIN=14 -env I_MPI_PIN_ORDER=compact \
  -n 4 -ppn 4 -host <host> app
```

**mpipin:** an MPI Implementation Agnostic Process Pinning Tool
$ mpirun -hostfile ~/hosts -n <N> -ppn <PPN> \ 
  mpipin --ranks-per-node <PPN> app

**Command Line Options**
- --processes-per-node, --ranks-per-node, --ppn:
  - Specifies the number of MPI processes per node
- --threads-per-process, --cores-per-process, --tpp:
  - Specifies the number of threads (logical CPUs) per MPI process
- --compact:
  - Follow compact process layout (default)
- --scatter:
  - Follow scattered process layout
- --exclude-cpus, --exclude-cores:
  - Specifies a list of logical CPUs to be excluded from resource partitioning

**Default Pinning Policy**
- An MPI implementation agnostic process pinning tool
- Simple and intuitive interface
- Transparent, topology aware process placement
- Not an API extension to mpirun!