How to bind MPI tasks/ OpenMP threads to logical/physical cpus in SLURM

MPI tasks

By default SLURM uses the auto-affinity module to bind MPI tasks to cpus. In some cases, you may want to modify this default, in order to ensure optimal performances.

You can find below the guidelines to do it.

1) On "exclusive" nodes if you require for the single node fewer tasks than the available cpus, you have to specify the following slurm option as a parameter of the "srun" command line:

```
--cpu-bind=cores
```

Alternatively, you can request more cpus for the single task so that you will use all the cpus of the node. First you have to define the environment variable S LURM_CPUS_PER_TASK with the directive:

```
--cpus-per-task=<n° cpus per node / n° tasks per node>
```

```
export SRUN_CPUS_PER_TASK=$SLURM_CPUS_PER_TASK
```

If the environment variable is not specified, it is not inherited (by design) by srun, so the default of --cpus-per-task is 1, and it is necessary to define it in the srun command:

```
 srun --cpus-per-task
```

If the “n° tasks per node” **is not a divisor** of “n° cpus per node“, then “n° cpus per task” should be equal to:

```
--cpus-per-task=<floor (n° cpus per node/ n° tasks per node)>
export SRUN_CPUS_PER_TASK=$SLURM_CPUS_PER_TASK
```

You have to remember that if hyperthreading is enabled, thus the “ n° cpus per node” refers to logical rather than physical cpus (logical cpus = physical cpus * n° hyper threads):

```
--cpus-per-task=<n° logical cpus per node / n° tasks per node>
```

```
--cpus-per-task=floor (n° physical cpus per node/ n° tasks per node) * n° hyper threads
```

```
export SRUN_CPUS_PER_TASK=$SLURM_CPUS_PER_TASK
```

In case the number of tasks is not a divisor of the number of cpus per node, covering all the tasks is not an alternative, **and you should add the srun directive "--cpu-bind=cores" anyway**.

2) On "hyperthreading nodes" if you require for a single node more tasks of available physical cpus you have to specify the following SLURM option:

```
--cpu-bind=threads
```

in order to ensure the binding between mpi tasks and logical cpus and to avoid the overload of physical cpus.

Alternatively, you can request more cpus for the single task until you use all logical cpus of the node defining:

```
--cpus-per-task=<n° logical cpus per node / n° tasks per node>
```

```
export SRUN_CPUS_PER_TASK=$SLURM_CPUS_PER_TASK
```

3) If you require for a single node a number of tasks that are equal to a number of physical cpus or number of logical cpus sthere is no need for adding --cpu-bind or --cpus-per-task slurm options. Each task is assigned a CPU in sequential order.

OpenMP threads

For OpenMP codes you have to make explicit the number of cpus to allocate for each single task to be used from OpenMP threads. In order to do it you can use the following slurm option to define SLURM_CPUS_PER_TASK:

```
--cpus-per-task=<n° cpus>
```

```
export SRUN_CPUS_PER_TASK=$SLURM_CPUS_PER_TASK
```

On nodes without hyperthreading the cpu concept coincides with physical cpu (core) and consequently the n° of cpus for single task (--cpus-per-task) can be up the maximum number of physical cpus of the node. For example, on BDW and SKL nodes the n° of cpus for single task can be up to 36/ 48 cpus respectively.

On hyperthreading nodes, it coincides with logical cpu (thread) and consequently, the n° of cpus for a single task can be up to the maximum number of logical cpus of the node.

In order to define if the OpenMP threads have to bind physical (core) or logical cpus (thread) you can use the following variable:

```
export OMP_PLACES= <cores|threads>
```

For example, if you have to run on a node with 32 cores and 8 hyper threads for single core, you can require up to 256 logical cpus for single task. If you require 256 openmp threads for a single task, each one binds necessary a logical cpus. If you require 32 openmp threads they can bind logical or physical cpus.

In order to bind physical cpus: 1 task, 32 openmp threads for a single task, 32 physical cpus (=256 logical cpus) for a single thread

```
--cpus-per-task=256
```

```
export SRUN_CPUS_PER_TASK=$SLURM_CPUS_PER_TASK
export OMP_NUM_THREADS=32
```

```
export OMP_PLACES= <cores>
```

In order to bind logical cpus: 1 task, 32 openmp threads for single task, 32 logical cpus for single thread

```
--cpus-per-task=256
```

```
export SRUN_CPUS_PER_TASK=$SLURM_CPUS_PER_TASK
export OMP_NUM_THREADS=32
```

```
export OMP_PLACES= <threads>
```

If you are using intel you can set KMP_AFFINITY variable to "compact" value in order to bind the threads to available cpus consecutively:

```
export KMP_AFFINITY=compact
```

You can modify this default in “scatter” way :

```
export KMP_AFFINITY=scatter
```

```
Alternatively, you can use non intel options:
```

```
export OMP_PROC_BIND=<close|spread|true>
```

```
A safe default setting is
```

```
export OMP_PROC_BIND=true
```

```
For all the details see the intel web pages:
```

https://software.intel.com/en-us/node/522691

You can find, at the following web page, some MPI and MPI/OpenMP jobs scripts examplest:

UG2.6.1: Batch Scheduler SLURM