

Testing Parallel Models using MPI-Start

V. Sipkova, J. Astalos, M. Dobrucky
Institute of Informatics SAS
Bratislava, Slovakia



- **The main goal of testing:**
to found out how MPI-Start supports various parallel models – relations between
 - user defined JDL attributes
 - CpuNumber
 - HostNumber, SMPGranularity, WholeNodes
 - user defined mpi-start options
 - -np n, -nnode m
 - mapping of MPI processes to allocated Nodes/Cores
 - mpiexec options – generated by MPI-Start

- **UMD based cluster** (ce2.ui.savba.sk)
 - number of nodes: 16
 - number of cores per node: 2
 - Compilers: gcc/gfortran v. 4.1.2 and v. 4.4.4
 - Open MPI v. 1.4
 - MPI-Start v. 1.2
- **Job management**
 - CREAM client commands were used

User specifications (JDL and start-script)	Results (from MPI-Start and running)
CpuNumber=6; mpi-start ... -np 6 ...	mpiexec ... -np 6 ... MPI processes: 6 Result: OK
HostNumber=3; SMPGranularity=2; WholeNodes=true; mpi-start ... -np 6 -nnode 2 ...	mpiexec ... -nnode 2 ... MPI processes: 6 (2 processes per node) Result: OK

User specifications (JDL and start-script)	Results (from running)
<pre>HostNumber=1; SMPGranularity=2; WholeNodes=true; export OMP_NUM_THREADS=2</pre>	<pre>OpenMP threads: 2 Result: OK</pre>

User specifications (JDL and start-script)	Results (from MPI-Start and running)
<pre>HostNumber=4; SMPGranularity=2; WholeNodes=true; #export OMP_NUM_THREADS=2 mpi-start ... -np 4 -nnode 1...</pre>	<pre>mpiexec ... -npernode 1 ... MPI processes: 4 (1 MPI process per node) OpenMP threads: 2 (per MPI process) Result: OK</pre>
<pre>HostNumber=2; SMPGranularity=4; WholeNodes=true; export OMP_NUM_THREADS=2 export MPI_USE_OMP=0 mpi-start ... -np 4 -nnode 2 ...</pre>	<pre>mpiexec ... -npernode 2 ... MPI processes: 4 (2 MPI process per node) OpenMP threads: 2 (per MPI process) Result: OK</pre>

- **All tests (MPI, OpenMP, MPI+OpenMP) terminate with correct results**
- **Hooks framework**
 - user hooks “pre-run” and “post-run” are working properly (including the compilation)
 - OpenMP hook (local site hook) controls setting of `OMP_NUM_THREADS`
It would be beneficial if this variable can be fixed also by the user
(`export MPI_USE_OMP=0`)