Testing MPI applications using MPI-Start

The main goal of these experiments was to find out how the MPI-Start supports various parallel models, that is, which relations are between the user defined JDL-attributes specifying the number of CPUs/Nodes (CpuNumber, HostNumber), the user defined options in "mpi-start" (-np, -npnode), and the way how MPI-Start maps MPI processes to allocated resources (options of mpiexec set up by MPI-Start).

Test-programs represent the implementation of the graph partitioning known as the combinatorial optimization problem “Max-Cut”.

Experimental results described below were obtained by running test-programs on our UMD based testing cluster: “ce2.ui.savba.sk” added to MPI kickstart VO.
For the job management the relevant CREAM client commands were utilized.

The configuration of the testing cluster	
Number of nodes	16
Number of cores per node	2
Open MPI	version 1.4
MPI-Start	version 1.2
Compilers	gcc and gfortran, versions 4.1.2 and 4.4.4
MPI SHARED HOME	available

MPI Tests	
User specifications (in JDL and start-script)	**Result** (from MPI-Start and running)
CpuNumber = 6; mpi-start ... -np 6 ...	mpiexec ... -np 6 … MPI processes launched: 6 Result: OK
HostNumber = 3; SMPGranularity = 2; WholeNodes = true; mpi-start ... -np 6 -npnode 2 ...	mpiexec ... -npernode 2 … MPI processes launched: 6 (2 processes per node) Result: OK
CpuNumber = 6; mpi-start ... -np 4 -npnode 2 … Not fully correct example!	mpiexec ... -npernode 2 … MPI processes launched: 8 (2 processes per node) Result: unpredictable (CPUs across 4 nodes were allocated)

OpenMP Tests	
User specifications (in JDL and start-script)	**Result** (from running)
HostNumber = 1; SMPGranularity = 2; WholeNodes = true; export OMP_NUM_THREADS=2	OpenMP threads launched: 2 Result: OK

MPI+OpenMP Tests	
User specifications (in JDL and start-script)	**Result** (from MPI-Start and running)
HostNumber = 4; SMPGranularity = 2; WholeNodes = true; #export OMP_NUM_THREADS=2 mpi-start ... -np 4 -npnode 1 ...	mpiexec ... -npernode 1 … MPI processes launched: 4 (1 process per node) OpenMP threads launched: 2 (per MPI process) Result: OK
HostNumber = 2; SMPGranularity = 4; WholeNodes = true; export OMP_NUM_THREADS=2 export MPI_USE_OMP=0 mpi-start ... -np 4 -npnode 2 ...	mpiexec ... -npernode 2 … MPI processes launched: 4 (2 processes per node) OpenMP threads launched: 2 per MPI process Result: OK

Summary (from the user point of view)

- **OpenMP tests terminate with correct results.**
- **MPI tests terminate with correct results** (including also pre-run and post-run hooks).
  The number of MPI processes launched, which is defined by the user in the "mpi-start" command through the option "-np" does not appear in the "mpiexec" (except the simplest case). The number of MPI processes is dependent on the temporary created machine file. As the strategy of the generating the machine file is hidden from the user, it would useful if in the debug information also the content of the actual machine file is listed.
- **Hybrid MPI+OpenMP tests are running properly**
  1. if the OpenMP hook (local site hook) is enabled, that is,the variable MPI_USE_OMP is set to 1, then the OMP_NUM_THREADS is determined by MPI-Start automatically;
  2. if the variable MPI_USE_OMP is set to 0 and the user defines the number of threads by himself (dynamically in program or by exporting the OMP_NUM_THREADS);
     the variable MPI_USE_OMP can be set and exported also by the user, but is it legal?
  3. else if the variable MPI_USE_OMP is not set the result is undefined (OMP_NUM_THREADS was set to 1).
- **Hooks Framework:** user hooks "pre-run" and "post-run" are working correctly.