MPI Working Group update

D. H. van Dok

Berkeley's 'seven dwarfs'

Seven classes of parallel computing methods.

1. dense linear algebra
2. sparse linear algebra
3. spectral methods
4. *n*-body methods
5. structured grids
6. unstructured grids
7. monte carlo

What seems to be the problem

1. Not enough resources available for parallel computing
2. not enough users run their parallel problems on the grid

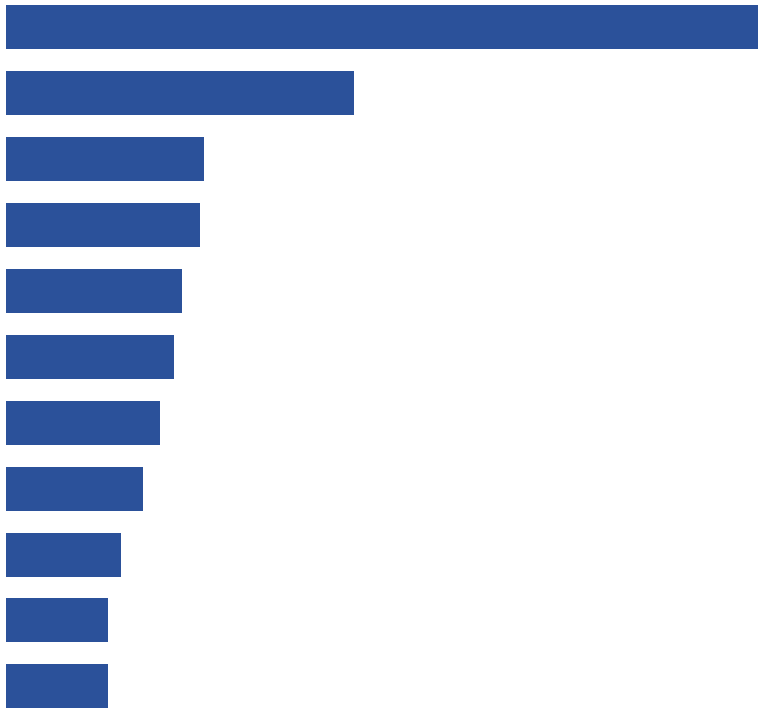
The question is whether this impression is correct.

findings from the information system

About **126** out of **484** listed SubClusters list anything like 'mpi' in the `GlueHostApplicationSoftwareRunTimeEnvironment`, or roughly **25%.**

The most frequently reported tags:

MPICH	171
MPI-START	79
MPICH-1.2.7p1	45
OPENMPI	44
MPICH-1.2.7	40
MPI_SHARED_HOME	38
MPICH2	35
MPI_NO_SHARED_HOME	31
MPICH2-1.0.4	26
VO-balticgrid-E-PARALLEL-MPICH2	23
OPENMPI-1.1	23



(65 more)

- SAM tests were discontinued due to too many failures
- supporting software is out of date, not maintained?

User Survey

Questions to the users:

- Can you give an estimate how intense the processing is, in terms of total CPU hours and the period over which these were used?
- Where have you found resources to run your programs in the past?
- Please estimate the size of previously used resource provider (total CPUs)
- Are you using the Grid to do multi-core computations, using MPI, OpenMP or similar techniques?
- Did you ask the Grid sites for support?
- Did you seek help in the EGEE community?
- What were the answers you got?
- Did you have to convince the system operators to install MPI?
- Was it easy to find documentation for running in parallel on the Grid?

A virtual tour of a parallel job

A user would typically visit the following waypoints:

- discovery of resources
- matchmaking
- (porting the application)
- passing job requirements
- scheduling
- initializing the system
- running and collecting output

Publishing tags in the IS

The new recommendation tentatively states that

```
GlueHostApplicationSoftwareRunTimeEnvironment: Parallel
GlueHostApplicationSoftwareRunTimeEnvironment: ⟨flavour⟩-⟨version⟩
GlueHostApplicationSoftwareRunTimeEnvironment: ⟨flavour⟩
```

should be used. Perhaps we should establish the canonical names for each possible flavour as well, to prevent varieties like OpenMPI, OPENMPI and OPEN-MPI from cropping up.
The matchmaking process will match the job requirements to those resources that have the right tag; unfortunately you can only specify an exact version, so matchmaking a la 'greater-than-or-equal' is not possible.

scheduling to whole nodes

There is a need to schedule to whole nodes:

- MPI jobs don't want to share the CPU with other work
- shared memory communication is faster than across the network
- OpenMP or simple multi-threaded applications
- memory-intensive applications

Whole-node scheduling is demanding on the configuration of the local batch system. Some systems are more attuned to such use than others.

Granularity

As a more general system than whole-node scheduling, SMPGranularity can be used in the JDL;

```
NodeNumber=12
SMPGranularity=4
```

means that 12 cores in total, with a distribution of 4 cores per node, are to be allocated.
This new attribute requires some changes at least to the WMS and CREAM JDL. These have to be worked out further for the LCG-CE and for BLAH; the technicalities were just recently discussed within JRA1.

for administrators

- The current collection of mpi software for the worker nodes is outdated, and causes dependency issues with mpiexec. The packages need to be updated, build for different MPI implementations, tailored for use with different batch systems.
- Source RPMs should be provided to allow the site admin to make a local version if needed.
- Reviving and updating the SAM tests, basically checking how well the installation reflects the new recommendations. This both helps users and site admins.
- updating documentation
- updating the yaim functions, providing easy 'default' setup for simple clusters.

last but not least...

Documentation should be reviewed, updated, improved and augmented with examples, validation scripts, etc.

Summary of recommendations

- update packages, provide RPMs and source RPMs
- update yaim functions
- revitalize SAM tests
- SMPGranularity, with examples of integrating this with popular batch systems
- publishing flavour and version
- updating and improving documentation
- more structural support beyond the WG