

gLite MPI support on BalticGRID

(C) 2007.01.16 K. Paulikas, KTU

Overview

- MPI status on BalticGrid
- MPI implementations
- Running MPI jobs via PBS
- Running MPI jobs via Grid
- Samples
- Performance

Status

- ~10 clusters support MPI
- All clusters have MPICH 1.x and “mpiexec” installed
- Custom MPICH2, OpenMPI, LAM-MPI versions available on some clusters
- All clusters on BalticGRID use torque as local PBS.

MPI implementations

- MPI: mpich
- MPI2: mpich2, LAM-MPI, OpenMPI
- Interoperability. Binaries do not work with different MPI library.
- Startup. Different MPI implementations support different startup methods: rsh/ssh, tm, globus ...

PBS (torque/maui) and MPI

- How real cluster CPUs are selected (torque interpretation of nodes and ppn)?
- Maui. ENABLEMULTIREQJOBS
(nodes=2:ppn=2+3:ppn=1)
- Allocated nodes: \$PBS_NODEFILE.
- Startup
 - ssh/rsh. Used by mpirun.
 - TM API. Torque v1.2 and later has TM API support. TM API may be used by LAM-MPI and OpenMPI.
 - mpiexec
- Reservations

gLite and MPI

- Rewriting JDL “NodeNumber” to PBS “nodes” and “ppn”
- Default resource limits: 48:00 cpu time, 72:00 wall time.
- CPU time is calculated for all processes of the job
- I/O waiting time is not cpu time.

[NON]SHARED /home

- Both cluster types available.
- There is no accurate test to check if HOME is shared.
- SMP machines.
- SHARED: easier to use for custom apps.
- NONSHARED: higher local I/O performance.

Notes

- **\$TMPDIR** is set to path, which is not available on other nodes.
- On clusters where CE and SE are on the same host You can access Your files while it is running (also shared /home required).
- **mpicxx** not available on ITPA cluster, use **mpicc** to compile C++ sources. *[Untested]*

Sample Job

job.jdl

```
Type = "Job";
JobType = "MPICH";
NodeNumber = 2;
Executable = "check.sh";
StdOutput = "check.out";
StdError = "check.err";
InputSandbox = {"check.sh", "mpihello.c", "Makefile"};
OutputSandbox = {"check.err", "check.out"};
#Requirements = (other.GlueCEInfoLRMSType == "PBS");
RetryCount = 0;
Lrms_Type = "PBS";
```

Makefile

```
mpihello: mpihello.c
    mpicc -o mpihello mpihello.c
```

```
#!/bin/sh -x
```

check.sh

```
make
mpiexec -kill mpihello
```

```
echo "==== End ====="
```

/* mpihello.c */

```
#include <mpi.h>
#include <stdio.h>
int main(int argc, char *argv[]){
    int numprocs;
    int procnum;
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &procnum);
    MPI_Comm_size(MPI_COMM_WORLD, &numprocs);
    printf ("Hello world! from processor %d out of %d\n", procnum, numprocs);
    MPI_Finalize();
    return 0;
}
```

mpiexec

Run MPI program

```
$ mpiexec -kill mpiprogram
```

Run non-MPI program

```
$ mpiexec -comm none -pernode -kill -nostdin hostname
```

copy file to other nodes

```
$ mpiexec -comm none -pernode -kill -nolocal -allstdin cat \> myfile < myfile
```

```
$ export SOMEVAR=somevalue
```

```
$ mpiexec -comm none -kill echo \${SOMEVAR} \${HOSTNAME}
```

```
somevalue host1
```

```
somevalue host2
```

```
somevalue host3
```

```
somevalue host4
```

To control individual nodes:

```
node1 node2 node3 : command1
```

```
node4 : command2
```

```
mpiexec -verbose -pernode -comm none -kill -nostdin -config mpiexec.conf
```

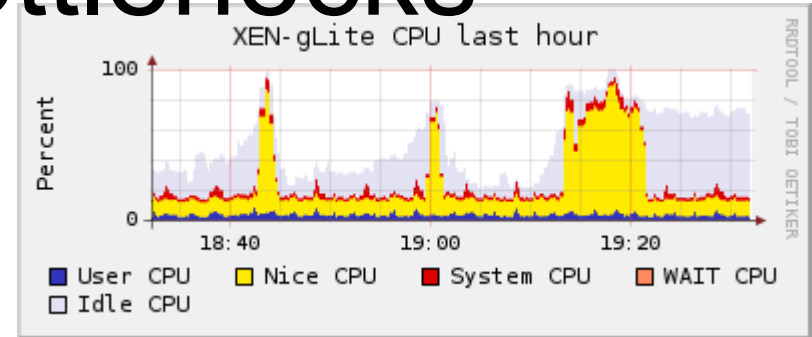
mpirun

- MPICH tries to start processes via rsh/ssh.
- for typical use can be replaced by mpiexec.
 - symlink or wrapper script in \$PATH before real mpirun
 - patch Your program to invoke mpiexec
 - use ssh replacement
- MPI2 implementations have their own mpiexec, which is different tool.
- mpirun from OpenMPI or LAM is usable if compiled with TM support.

Bugs/problems

- Some clusters use *torque* v1.0 (gLite default).
- missing *tm.h* ant *libpbs libtorque* (request to install submitted)
- missing *libpbs* dependencies (*libelf*)
- gLite default *mpiexec* v0.77 does not work with *torque* v2.x.
- ...

Performance bottlenecks



- Heavy disk usage
 - Allocated RAM for job processes exceeds available RAM on WN - swapping occurs
 - Random disk reads/writes (seek)
 - Concurrent use of the same HDD by different jobs
- Bandwidth and latency of interconnections between nodes (SMP, LAN, InfiniBand ...)
- MultiCPU nodes
 - Some clusters allow more jobs than CPUs per node
 - Performance of SMP nodes isn't always $N \times \text{CPU}$

Conclusions

- Detailed information about clusters is required to use resources efficiently.
- User ability to monitor job execution is too limited.

References

- http://www.balticgrid.org/Grid_Operations/technicalguides/Advices_for_application_developers/MPI_in_BalticGrid
- <http://www.osc.edu/~pw/mpiexec/index.php>
- <http://kopustas.elen.ktu.lt/help/grid>
- gLite User Manual
- http://goc.grid.sinica.edu.tw/gocwiki/MPI_Support_with_Torque