

How to submit MPI jobs

Come lanciare job MPI su GRID@Trieste.

I sistemi Abilitati all'esecuzione di JOB MPI sono due, *egrid-ce-01.egrid.it* e *baciuco.hpc.sissa.it*

Descrizione egrid-ce-01.egrid.it

Questo cluster utilizza la tecnologia di *LCG* quindi le home degli utenti di griglia *non sono condivise*, perciò bisogna utilizzare un wrapper che provvede a copiare via ssh l'eseguibile sui vari nodi allocati per l'esecuzione. Inoltre il sistema non supporta MPI su Myrinet.

Nota Bene: causa BUG del RB https://savannah.cern.ch/bugs/?func=detailitem&item_id=7875 Bisogna indicare esplicitamente il sistema di code durante la sottomissione opzione '-lrms pbs':

```
edg-job-submit --vo egrid -r egrid-ce-01.egrid.it:2119/jobmanager-lcgpbs-egrid -lrms pbs MPItest.
```

Procedura di sottomissione:

Compilare il proprio eseguibile con mpi senza supporto per Myrinet Tale informazione ? pubblicata sul BDII con il *software tag* p4

Creare un job che effettuata i seguenti passi

Carica in griglia l'eseguibile nell'esempio seguente 'cpi' e il wrapper 'MPItest.sh' che provvede a copiare l'eseguibile sui vari nodi allocati e a lanciare il comando 'mpirun'

File JDL:

```
Type = "Job";
JobType = "MPICH";
NodeNumber = 4;
Executable = "MPItest.sh";
Arguments = "cpi 4";
StdOutput = "test.out";
StdError = "test.err";
InputSandbox = {"MPItest.sh", "cpi"};
OutputSandbox = {"test.err", "test.out", "executable.out"};
Requirements = other.GlueCEInfoLRMSType == "PBS" || Member("p4", other.GlueHostApplicationSoftware
```

Wrapper::

```
#!/bin/sh
#
# this parameter is the binary to be executed
EXE=$1

# this parameter is the number of CPU's to be reserved for parallel execution
CPU_NEEDED=$2

# prints the name of the master node
echo "Running on: $HOSTNAME"

echo "*****"
if [ -f "$PWD/.BrokerInfo" ] ; then
    TEST_LSF=`edg-brokerinfo getCE | cut -d/ -f2 | grep lsf`
else
```

```

TEST_LSF=`ps -ef | grep sbatchd | grep -v grep`
fi

if [ "x$TEST_LSF" = "x" ] ; then
# prints the name of the file containing the nodes allocated for parallel execution
echo "PBS Nodefile: $PBS_NODEFILE"
# print the names of the nodes allocated for parallel execution
cat $PBS_NODEFILE
echo "*****"
HOST_NODEFILE=$PBS_NODEFILE
else
# print the names of the nodes allocated for parallel execution
echo "LSF Hosts: $LSB_HOSTS"
# loops over the nodes allocated for parallel execution
HOST_NODEFILE=`pwd`/lsf_nodefile.$$
for host in ${LSB_HOSTS}

do
echo $host >> ${HOST_NODEFILE}
done

fi
echo "*****"
# prints the working directory on the master node
echo "Current dir: $PWD"
echo "*****"

for i in `cat $HOST_NODEFILE` ; do
echo "Mirroring via SSH to $i"
# creates the working directories on all the nodes allocated for parallel execution
ssh $i mkdir -p `pwd`
# copies the needed files on all the nodes allocated for parallel execution
/usr/bin/scp -rp ./ * $i:`pwd`
# checks that all files are present on all the nodes allocated for parallel execution
echo `pwd`
ssh $i ls `pwd`
# sets the permissions of the files
ssh $i chmod 755 `pwd`/$EXE
ssh $i ls -alR `pwd`
echo "#####"
done

# execute the parallel job with mpirun

echo "*****"
echo "Executing $EXE"
chmod 755 $EXE
ls -l
#
# Mpi exectution
#
mpirun -np $CPU_NEEDED -machinefile $HOST_NODEFILE `pwd`/$EXE > executable.out
echo "*****"

```

Descrizione baciuco.hpc.sissa.it

Questo cluster utilizza le home condivise e pubblica come LMRS pbs, per cui non c'è bisogno di nessun *wrapper*, in quanto viene eseguito direttamente l'eseguibile tramite *mpirun*. L'eseguibile deve essere compilato con il supporto per mpi su Myrinet. Il sistema informativo di baciuco pubblica il *SoftwareTag gm* che viene utilizzato per effettuare il matchmaking.

In questo caso l'eseguibile viene caricato in griglia ed eseguito direttamente dal job wrapper tramite *mpirun* senza nessun wrapper che si occupi di copiarlo tra i vari workernode assegnati.

File JDL:

```
Type = "Job";
JobType = "MPICH";
NodeNumber = 4;

Executable = "hello-mpi";
StdOutput="hello-mpi.out";
StdError="hello-mpi.err";

InputSandbox={"hello-mpi"};
OutputSandbox={"hello-mpi.out", "hello-mpi.err"};

Requirements      = Member("gm",other.GlueHostApplicationSoftwareRunTimeEnvironment);
```

Sottomissione:

```
edg-job-submit --vo egrid hello-mpi.jdl
```