Klastrové a Gridové Aplikácie

Viera Šipková
Ústav informatiky SAV
Bratislava, Dúbravska cesta 9

Obsah

- **Vývoj aplikácie**
  - pre klaster
  - pre grid
- **Vykonávanie aplikácie**
  - na lokálnom klastri
  - na gride

Klastrové počítanie

- **Klaster s viac-jadrovými CPUs** – počítačový systém s distribuovanou a zdieľanou pamäťou
  - množina nezávislých viac-jadrových procesorov (uzlov), navzájom spojených prostredníctvom prepojovacej siete, ktorá umožňuje medzi-uzlovú komunikáciu
  - každý uzol má svoju vlastnú pamäť (Distributed Memory - DM)
  - všetky jadrá v rámci uzla zdieľajú jednu spoločnú pamäť (Shared Memory - SM)
- **Klastrové počítanie** ⇒ hybridný programovací model
  - uplatnenie technológií pre DM a technológií pre SM (prípadne aj GPU technológií)

Klastrové počítanie

- **Vytvorenie úlohy**
  - vývoj kódu, kompilácia (Linux, C, Fortran, MPI, OpenMP)
  - vývoj skriptov (Shell, Python, ...)
    - pre spustenie úlohy na lokálnom klastri
    - pre manažovanie procesu vykonávania úlohy
- **Vykonanie úlohy**
  - prostredníctvom systému PBS (Portable Batch System)
    - qsub - predloží úlohu batch-serveru na vykonanie
    - qstat - požiada batch-server o výpis stavu úlohy
    - qdel - požiada batch-server o predčasné ukončenie úlohy
- **Popis úlohy**
  - Shell-skript obsahujúci PBS príkazy
    - vstup pre PBS “qsub” príkaz

Klastrové počítanie

- **PBS skript** (Seq. program)

start-test1.sh

```
#!/bin/sh
#PBS -l nodes=1
#PBS -N test1
#PBS -o tstd1.out
#PBS -e tstd1.err
#PBS -q local
cd $PBS_O_WORKDIR
. . . Shell commands . . .
echo "Testing start: "`date`
./test1.exe  input_par
echo "Testing end: "`date`
exit $?
```

Klastrové počítanie

- **PBS skript** (MPI program)

start-test2.sh

```
#!/bin/sh
#PBS -l nodes=8
#PBS -N test2
#PBS -o tstd2.out
#PBS -e tstd2.err
#PBS -q local
cd $PBS_O_WORKDIR
. . . Shell commands . . .
npr=`wc -l  < $PBS_NODEFILE`
echo "Testing start: "`date`
mpirun  -np $npr  test2.exe input_par
echo "Testing end: "`date`
```

Klastrové počítanie

- **PBS skript** (MPI+OpenMP program)

start-test3.sh

```
#!/bin/sh
#PBS -l nodes=4:ppn=2
#PBS -N test3
#PBS -o tstd3.out
#PBS -e tstd3.err
#PBS -q local
#PBS -v OMP_NUM_THREADS=4
cd $PBS_O_WORKDIR
. . . Shell commands . . .
echo "Testing start: "`date`
mpiexec  --bynode -np 4  test3.exe input_par
echo "Testing end: "`date`
```

- **Manažér skript**

manager1.sh

```
#!/bin/sh
if [ $# -ne 2 ]; then
   echo "Incorrect number of input arguments"
   exit 1
fi
# Generating PBS submission script
cat > start-test.sh << EOF
. . . PBS-Shell commands (using input arguments) . . .
EOF
# Submitting job
qsub  start-test.sh
exit 0
```

Klastrové počítanie

- **Manažér skript** (vykonanie viac úloh)

managerN.sh

```
#!/bin/sh
. . . Generating PBS submission script: start-test0.sh . . .
# Submitting job test0
qsub  start-test0.sh  > qoutput
# qoutput: 74324.ce2.ui.savba.sk
jobName=`cat  qoutput`
jobID=${jobName%%\.*}
# Submitting set of jobs dependent on successful termination of test0
for (( n=1; n<=NN; n++ )) ; do
  . . . Generating PBS submission scripts: start-test$n.sh . . .
  qsub  -W depend=afterok:$jobID  start-test$n.sh
  sleep 2
done
```

Gridové počítanie

- **Grid** – distribuovaný paralelný systém
  - umožňuje zdieľanie, výber a zoskupenie geograficky distribuovaných autonómnych prostriedkov (výpočtových, úložných, softvérových, a iných) dynamickým spôsobom, v závislosti od ich dostupnosti, vybavenia, výkonnosti, ceny a používateľských požiadaviek na kvalitu služieb
  - súčasné výpočtové prostriedky: výkonné počítačové klastre

- **Gridové počítanie** ⇒ hybridný programovací model
  - uplatnenie technológií pre klastrové počítanie a technológií webových a gridových služieb

Gridové počítanie

- **Vytvorenie úlohy**
  - vývoj kódu – ako pre klaster (Linux, C, Fortran, MPI, OpenMP)
  - vývoj skriptov
    - pre spustenie úlohy na gride
    - pre manažovanie procesu vykonávania úlohy
- **Vykonanie úlohy**
  - prostredníctvom gridového middlewaru
    - gLite, Globus, EMI (gLite+ARC+UNICORE)
  - gLite WMS (Workload Management System)
    - glite-wms-job-submit – predloženie úlohy na vykonanie
    - glite-wms-job-status – výpis stavu vykonávanej úlohy
    - glite-wms-job-output – výber výsledkov ukončenej úlohy
    - glite-wms-job-cancel – predčasné ukončenie úlohy

Gridové počítanie

- **Popis úlohy** (gLite WMS)
  - skript v jazyku JDL (Job Description Language)
    - JDL je vytvorený na báze “Condor classified advertisement”, umožňuje popisovať úlohy a množiny úloh, špecifikovať ich vlastnosti, požiadavky a obmedzujúce podmienky, ktoré sú uplatňované pri výbere najvhodnejšieho dostupného výpočtového prostriedku
  - vstup pre “glite-wms-job-submit” príkaz

Gridové počítanie

- **Typ úlohy** (gLite WMS)
  - Job – jednoduchá úloha
    - Normal: štandardné vykonateľné skripty a programy, skripty spúšťajúce MPI programy
    - Parametric: množina identických úloh, ktoré sa vykonávajú s rôznymi vstupnými parametrami resp. dátami)
    - MPICH: MPI programy vykonateľné priamo (zastaraný!)
    - Interactive: štandardný vstup/výstup úlohy je priamo pripojený ku klientovi, ktorý spúšťa úlohu, bez presmerovania z/do súboru
  - DAG – graf závislých úloh
  - Collection – množina nezávislých úloh

Gridové počítanie

- **Jednoduchá úloha**

simple-job.jdl

```
Type= "Job";
JobType= "Normal";
Executable= "test.exe";
Arguments= "input.dat";
StdOutput= "std.out";
StdError= "std.err";
InputSandbox= { "test.exe", "input.dat" };
OutputSandbox= { "std.out", "std.err", "output.dat" };
RetryCount= 0;
ShallowRetryCount= 3;
Requirements= other.GlueCEUniqueID==
   "ce2.ui.savba.sk:2119/jobmanager-pbs-esr";
```

- **MPI úloha** (zastaraný spôsob!)

mpi-job.jdl

```
Type= "Job";
JobType= "MPICH";
CpuNumber= 4;
Executable= "mpi-test.exe";
Arguments= "input.dat";
StdOutput= "std.out";
StdError= "std.err";
InputSandbox= { "mpi-test.exe", "input.dat" };
OutputSandbox= { "std.out", "std.err", "output.dat" };
RetryCount= 0;
ShallowRetryCount= 0;
```

Gridové počítanie

- **MPI úloha** (spustená cez wrapper-skript)

wmpi-job.jdl

```
Type= "Job";
JobType= "Normal";
CpuNumber= 4;
Executable= "run-mpi-test.sh";
Arguments= "4 mpi-test.exe input.dat";
StdOutput=" std.out";
StdError=" std.err";
InputSandbox= { "run-mpi-test.sh", "mpi-test.exe", "input.dat" };
OutputSandbox= { "std.out", "std.err", "mpistd.out", "output.dat" };
RetryCount= 0;
ShallowRetryCount= 0;
Requirements= Member("OPENMPI",
   other.GlueHostApplicationSoftwareRunTimeEnvironment);
```

Gridové počítanie

- **Parametrická úloha**

par-job.jdl

```
Type= "Job";
JobType= "Parametric";
Executable= "par-test.exe";
Arguments= "_PARAM_ input_PARAM_.dat";
Parameters= 10;
ParameterStart= 0;
ParameterStep= 1;
StdOutput= "std_PARAM_.out";
StdError= "std_PARAM_.err";
InputSandbox= { "par-test.exe", "input_PARAM_.dat" };
OutputSandbox= { "std_PARAM_.out", "std_PARAM_.err",
    "output_PARAM_.dat" };
```

- **DAG úloha**

dag-job.jdl

```
Type= "Dag";
DefaultRetryCount= 0;
InputSandbox= { "input.dat" };
max_running_nodes= 2;
nodes = [
   nodeA = [ description = [
      JobType= "Normal";
      Executable= "testA.exe";
      InputSandbox= { "testA.exe", root.InputSandbox[0] };
      . . .
   ]; ];
   nodeB = [ description = [ . . . ]; ];
  dependencies= { nodeA, nodeB };
];
```

- **Manažér skript**

manager.sh

```
#!/bin/sh
if [ $# -ne 2 ]; then
   echo "Incorrect number of input arguments" ; exit 1
fi
# Generating JDL job-submission script
cat > test.jdl << EOF
. . . JDL specification (using input arguments) . . .
EOF
# Submitting job
glite-wms-job-submit  -a -o jobID test.jdl
# Testing status:  glite-wms-job status -i jobID
# Retrieving output results:  glite-wms-job-output --dir output -i jobID
```

Linky

- **MPI** http://www.mcs.anl.gov/research/projects/mpi
- **OpenMP** http://openmp.org/wp
- **PBS** http://www.pbsworks.com
- **gLite** http://glite.web.cern.ch/glite
- **EGEE** http://www.eu-egee.org
- **EGI** http://www.egi.eu
- **SlovakGrid** http://www.slovakgrid.sk

Ďakujem za pozornosť !