Interfacing BLAHP* with LSF – Status Report

Ulrich Schwickerath
CERN/IT/FIO

Francesco Prelz
INFN

(*) BLAHP = Batch System ASCII Helper Protocol

Description of the problem

Current situation:

- Users can specify some a priori knowledge about their jobs in the JDL
  - Memory requirements
  - Approximate length of the job
  - Special requirements like CPU architecture
  - ...
- This information is currently **not passed** down to the level of batch system
- Different batch systems use different ways of specifying requirements

Work - around at CERN: special Grid queues:

- one queue per VO
- highest priority (higher than local queues)
- longest possible (one week)
- use only recent nodes with at least 2GB of memory

Description of the problem (2)

Disadvantages:

- non-optimal use of available computing resources
  - older nodes are not being used even if the jobs would not need much memory
  - big jobs may get killed because the execution node runs out of memory
  - local users suffer since Grid Jobs have highest priority

- Grid Jobs may be very short but are still run in a long queue
  - Hard to predict turn around time for jobs
  - Need to close queues already one week in advance of a maintenance event

Prototype for better solution:
passing **memory** and **OS requirements**
to the batch system

Glite starts jobs via BLAHP on the CE. A new version of BLAHP will be able to pass the following information to a **new batch system dependent** module:

```
GlueHostApplicationSoftwareRunTimeEnvironment="APP3"
GlueHostMainMemoryRAMSize_Min=2000
VirtualOrganisation=DTEAM
GlueHostOperatingSystemName=SLC3
...
```

Which is mapped to (in the case of LSF):

```
#BSUB -R "select[mem>=2000&&type==SLC3]"
```

Prototype for better solution:
passing **run time requirements** to batch system

Information given by BLAHP (min of kSI2K):

```
GlueCEPolicyMaxCPUTime_Min=10
GlueCEPolicyMaxCPUTime_Max=100
```

Run time restrictions in LSF are defined on the queue level:

LSF setup: replace VO-based Grid queues by one set of time based queues

```
QUEUE_NAME PRIO STATUS
grid_dteam  21  Open:Active
grid_cms    20  Open:Active
grid_atlas  20  Open:Active
grid_lhcb   20  Open:Active
grid_alice  20  Open:Active
```

```
QUEUE_NAME      PRIO STATUS
grid_8nm         7   Open:Active
grid_1nh         6   Open:Active
grid_8nh         5   Open:Active
grid_1nd         4   Open:Active
grid_1nw         3   Open:Active
```

Note: Local queues for non-Grid users are still required to allow fair share

Prototype for better solution:
passing run time requirements to batch system

With this setup the mapping should result in :

```
#BSUB -q grid_1nd
```

Remarks:

- Adjust queue priorities to make short jobs faster
- All pool accounts are allowed to run in all these queues
- Make the new queues look like single one when reporting status information
- Local Queue match making:
  - Select all queues that provide at least the required CPU time limit
  - Among those, select the shortest one that provides the run time limit
  - If no matching queue, use the largest available queue
- Accounting can still be done based on group membership of pool accounts
- Second set of non-Grid queues may still be needed for local users

Implementation details: Script plugin requirements

Requirements:

- As little site specific as possible
  - Read available queue names from LSF subsystem itself
  - Read queue configuration from LSF itself
  - Extract site specific things into configuration file (eg. Queues that may be used
- Read BLAHP input as shell variables
- Write results to STDOUT

Prototype implementation features:

- Perl script:
  - read out LSF configuration
  - make queue matching
  - Write out results in LSF format to STDOUT
- Shell script wrapper: export only those variables that we can deal with

Prototype for better solution: first results

No special resources requested: head of created submission script

```
#!/bin/bash
# LSF job wrapper generated by lsf_submit.sh
# on Thu Mar 30 20:12:41 CEST 2006
#
# LSF directives:
#BSUB -L /bin/bash
#BSUB -N
#BSUB -u blahp_sink@mi.infn.it
#BSUB -J blahjob_V17238
#BSUB -f "/home/grid/egee004/Condor_glidein/local.eb981.../StandardOutput <StandardOutput.124..."
#BSUB -f "/home/grid/egee004/Condor_glidein/local.eb/spool/cluster16.proc0.subproc0/Sta..."
#BSUB -f "/opt/glite/bin/BPRserver > BPRserver.13484.17144.1143742361"
#BSUB -f "/home/grid/egee004/Condor_glidein/local.eb981f6a5113696b9cbe5c4c8/spool/clust...X8Lw.sh"
#BSUB -q grid_glite
#BSUB -f "/home/grid/egee004/Condor_glidein/local.eb9user.proxy.[...]lmt > blahjob_V17238.2361.proxy"

# Check whether we need to move to the LSF original CWD:
...
```

Prototype for better solution: first results

Memory requirement: > 750 MB
very long job

```
#!/bin/bash
# LSF job wrapper generated by lsf_submit.sh
# on Thu Mar 30 20:28:42 CEST 2006
#
# LSF directives:
#BSUB -L /bin/bash
#BSUB -N
#BSUB -u blahp_sink@mi.infn.it
#BSUB -J blahjob_xz3783
#BSUB -R "select[mem>=750]"
#BSUB -q grid_1nw
#BSUB -f "/home/grid/egee004/Condor_glidein/local.eb981f6a5113696b9cbeeb929e95c4c8/spool/cluster[...]
#BSUB -f "/home/grid/egee004/Condor_glidein/local.eb981f6a5113696b9cbeeb929e95c[...]
#BSUB -f "/opt/glite/bin/BPRserver > BPRserver.13484.3737.1143743322"
#BSUB -f "/home/grid/egee004/Condor_glidein/local.eb981f6a5113696b9cbeeb929e95c4c8/spool/c[...]
#BSUB -q grid_glite
#BSUB -f "/home/grid/egee004/Condor_glidein/local.eb981f6a5113696b9cbeeb929e95c4c8/s[...]

#
```

Conclusions and Open issues

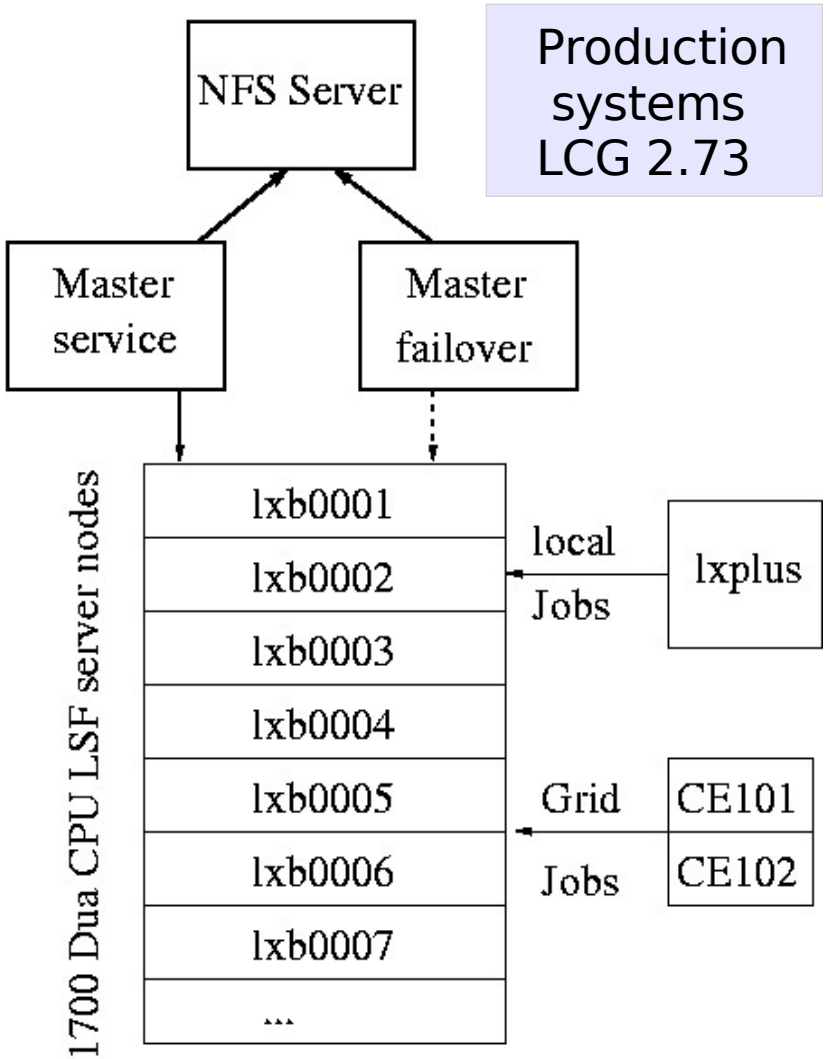
passing job requirements down to the batch system level with BLAHP

Proof of concept works!

Looking back at last HEPiX: http://hepix.caspur.it/afs/hepix.org/project/batch

CPU Time required	OK
Specific operating system	OK
Wall Clock Time required	OK
Total RAM required	OK
Swap space required	virtual memory is given by BLAHP
Job Name	need to check if BLAHP default can be overwritten
Temporary disk space	not yet implemented (should be possible for CERN)
Speed of processor required	different sites use different conventions here

Prototype for better solution:
Overview LSF batch system at CERN



Production systems LCG 2.73

glite/BLAHP test system