ICM - where and who we are?
Dominik Bartkiewicz, Marcin Stolarek

ICM – Interdisciplinary Center for Mathematical and Computational Modeling

Five supercomputers:
- halo2 (Sun Constellation, x86 cluster ~8k cores)
- hydrA (x86 cluster ~ 5k cores)
- notos (BGP ~2k cores)
- nostromo (BGQ ~16k cores)
- boreasz (Power 775 ~2k cores)

www.icm.edu.pl
ICM - where and who we are?
Dominik Bartkiewicz, Marcin Stolarek

ICM – Interdisciplinary Center for Mathematical and Computational Modeling

Five supercomputers:
- halo2 (Sun Constellation, x86 cluster ~8k cores)
- hydra (x86 cluster ~ 5k cores)
- notos (BGP ~2k cores)
- nostromo (BGQ ~16k cores)
- boreasz (Power 775 ~2k cores)

Where is Warsaw?

Hydra cluster - overview
- 4 kind of nodes (12,16,20,48,54 cores), 3 kind of GPUs
  (both Nvidia and AMD)
- disk-less setup based on NFS root and customized init
- nodes with and without infini-band (complex topology)
- 3 kinds of users: local (ICM), plgrid federation, WLCG

Middleware

Healthcheck

JobSubmit Plugin

/tmp and bindtmp.so plugin

unshare.so plugin

www.icm.edu.pl

www.plgrid.pl
ICM - where and who we are?
Dominik Bartkiewicz, Marcin Stolarek

ICM – Interdisciplinary Center for Mathematical and Computational Modeling

Five supercomputers:
- halo2 (Sun Constellation, x86 cluster ~8k cores)
- hydra (x86 cluster ~ 5k cores)
- notos (BGP ~2k cores)
- nostromo (BGQ ~16k cores)
- boreasz (Power 775 ~2k cores)

www.icm.edu.pl
www.plgrid.pl
Where is Warsaw?
ICM - where and who we are?
Dominik Bartkiewicz, Marcin Stolarek

ICM – Interdisciplinary Center for Mathematical and Computational Modeling

Five supercomputers:
- halo2 (Sun Constellation, x86 cluster ~8k cores)
- hydra (x86 cluster ~ 5k cores)
- notos (BGP ~2k cores)
- nostromo (BGQ ~16k cores)
- boreasz (Power 775 ~2k cores)

www.icm.edu.pl
www.plgrid.pl
**Hydra cluster - overview**

- 4 kinds of nodes (12, 16, 20, 48, 64 cores), 3 kinds of GPUs (both Nvidia and AMD)
- Disk-less setup based on NFS root and customized init
- Nodes with and without infini-band (complex topology)
- 3 kinds of users: local (ICM), plgrid federation, WLCG

**Middleware**

*We support three middleware:*
- gLite (first production site in WLCG)
- UNICORE
- QCG (slurm-drmaa)

**Healthcheck**

- Check functionality rather than state
- First idea: don’t change anything, just check and live information in “Reason=” field
- (Actually we try to heal some problems)
- Maybe add this to job epilogue

**Black hole detection**

- Simple cron job, trying to submit jobs to non-allocated nodes
- Write an email, drain node

**Reboot nodes**

- Look for nodes drained by “blackhole” or healthcheck
- Now it’s cron, but we will probably switch to strigger

Slurmmmon, OStrich, account backup generation

Thank you for your attention!

Our questions:
- How do you perform slurm profiling?
- Is it possible to use dynamic from user application (API documentation?)

/tmp and bindtmp.so plugin
Middleware

We support three middleware:

- gLite (first production site in WLCG)
- UNICORE
- QCG (slurm-drmaa)
Healthcheck
• check functionality rather than state
• first idea: don't change anything, just check and live information in "Reason=" field
• (actually we try to heal some problems)
• maybe add this to job epilogue

Black hole detection
• simple cron job, trying to submit jobs to non-allocated nodes
• write an email, drain node

Reboot nodes
• look for nodes drained by "blackhole" or healthcheck
• now it's cron, but we will probably switch to strigger
source ./chk_dir
#Small random sleep to reduce load on lustre
time="0."$[ $RANDOM%100 ]
sleep $time

#Check if specified user can create/delete file in spec. location
cchk_dir /mnt/lustre/temp/plgmonitoring/ plgmonitoring
cchk_dir /icm/home/mstol mstol
cchk_dir /icm/tmp/mstol mstol
cchk_dir /mnt/unicore/monitoring plgmonitoring

#Check filesystems - parse df output
source ./chk_fs
cchk_fs /tmp/
cchk_fs /var/

#Check if there are processes not connected to running jobs (try to clean this)
#Remove empty cpusets
source ./chk_cgroup
cchk_cgroup
Healthcheck
- check functionality rather than state
- first idea: don't change anything, just check and live information in "Reason=\" field
- (actually we try to heal some problems)
- maybe add this to job epilogue

Black hole detection
- simple cron job, trying to submit jobs to non-allocated nodes
- write an email, drain node

Reboot nodes
- look for nodes drained by "blackhole" or healthcheck
- now it's cron, but we will probably switch to strigger
/tmp and bindtmp.so plugin

- create per job (in job prolog) on node and delete in the job epilog
- ext4 filesystem in file on Lustre (loopback mount in: /tmp/lls-${SLURM_JOB_ID})
- size determined by "--tmp="
- spank plugin: unshare and bind to /tmp

unshare.so plugin

- Unshare and unmount filesystem if job doesn't specify licenses
- We use this plugin to limit number of job using Lustre

display plugstack.conf:
optional bindtmp.so /tmp/lls_
optional unshare.so lustre /mnt/lustre
Job Submit Plugins

What they provide:
- prevent submission of insane jobs
- check if user have active account

Programming:
- we do it in C; don't know Lua and are afraid of lack of libraries (for example libldap).
- more examples available in C.
- we develop spank plugin wrapping shell scripts
Configuration file example:

```
feature istanbul 12
feature westmere 12
feature magnycours 48
feature interlagos 64
feature magnycours_2300mhz_fermi_480 16
feature ivybridge 20
user plgmonitoring
```

What we check:

- Does the job use more than one node?
- Does the user specified constraint for this job?
- Does the job use full nodes? (if not - reject, if yes add needed features)

Plugin source:

https://github.com/cinek810/misc/tree/master/jobs-submit/job_sane
Job Submit Plugins

What they provide:
- prevent submission of insane jobs
- check if user have active account

Programming:
- we do it in C; don't know Lua and are afraid of lack of libraries (for example libldap).
- more examples available in C.
- we develop spank plugin wrapping shell scripts
Account verification in job_submit plugin:

What does it work:
- connect to LDAP server
- set default grant (if user haven't specified)
- check if user hasn't permissions to user this grant, if grant is active etc.

Pros of this solution:
- changes in account management portal have immediate effect on job submission
- the logic can be complicated and different for particular partitions

Cons:
- we are not able to use QoS, since we don't have account enforcement enabled in slurm.conf
Job Submit Plugins

What they provide:
- prevent submission of insane jobs
- check if user have active account

Programming:
- we do it in C; don't know Lua and are afraid of lack of libraries (for example libldap).
- more examples available in C.
- we develop spank plugin wrapping shell scripts
Thank you for your attention!

Our questions:

- How do you perform slurm profiling?
- Is it possible to use dynalloc from user application (API documentation)?

Personal: bart@icm.edu.pl, mstol@icm.edu.pl
Admin list: hydra-admins@icm.edu.pl
ICM - where and who we are?
Dominik Bartkiewicz, Marcin Stolarek

ICM – Interdisciplinary Center for Mathematical and Computational Modeling

Five supercomputers:
- halo2 (Sun Constellation, x86 cluster ~8k cores)
- **hydra** (x86 cluster ~ 5k cores)
- notos (BGP ~2k cores)
- **nostromo** (BGQ ~16k cores)
- boreasz (Power 775 ~2k cores)

www.icm.edu.pl

www.plgrid.pl