

Site report: ICM Warsaw University

ICM - where and who we are?

Dominik Bartkiewicz, Marcin Stolarek

ICM – Interdisciplinary Center for Mathematical and Computational Modeling

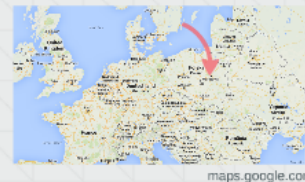
Five supercomputers:

- 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

Where is Warsaw ?



maps.google.com



www.plgrid.pl

Hydra cluster - overview

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

Middleware

- We support three middleware:
- jll Site (first production site in WLCG)
 - UNICOR
 - QCG (slurm-drmaa)



Healthcheck

- check functionality rather than state
 - first check - don't change anything, just check and log information in "Healthcheck" file
 - actually we try to find some problems
 - maybe add this to job epilog
- Black hole detection**
- simple cron job, trying to submit jobs to non allocated nodes
 - send an email, drain node
- Reboot nodes**
- look for nodes drained by "blackhole" or healthcheck
 - give it's cores, but we will probably switch to straggler

slurmon, OSTree, account backup generation



Thank you for your attention!

Our applications:

What they provide:

Programming:

Job Submit Plugins

What they provide:

Programming:

Job Submit Plugins

What they provide:

Programming:

/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/lfs_\$(SLURM_JOB_ID)
- size determined by "--tmp="
- spack 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

example pluginstack.conf:
optional bindtmp.so /tmp/lfs
optional unshare.so lustre /mnt/lustre

Site report: ICM Warsaw University

ICM - where and who we are?

Dominik Bartkiewicz, Marcin Stolarek

ICM – Interdisciplinary Center for Mathematical and Computational Modeling

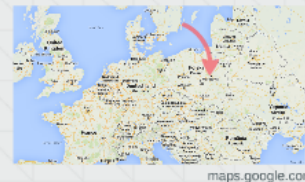
Five supercomputers:

- 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

Where is Warsaw ?



maps.google.com



www.plgrid.pl

Hydra cluster - overview

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

Middleware

- We support three middleware:
- jll Site (first production site in WLCG)
 - UNICOR
 - QCG (slurm-drmaa)



Healthcheck

- check functionality rather than state
 - first check - don't change anything, just check and log information in "Healthcheck" file
 - actually we try to find some problems
 - maybe add this to job epilog
- Black hole detection**
- simple cron job, trying to submit jobs to non allocated nodes
 - send an email, drain node
- Reboot nodes**
- look for nodes drained by "blackhole" or healthcheck
 - give it's cores, but we will probably switch to straggler

slurmon, OSTree, account backup generation



Thank you for your attention!

One application

What they provide:

Progressing:

Job Submit Plugins

optional bindtmp.so /tmp/lustre

optional unshare.so /lustre

optional unshare.so /lustre

/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/lustre/\$SLURM_JOB_ID
- size determined by "--tmp="
- spunk 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

example pluginstack.conf:
optional bindtmp.so /tmp/lustre
optional unshare.so /lustre

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 ?

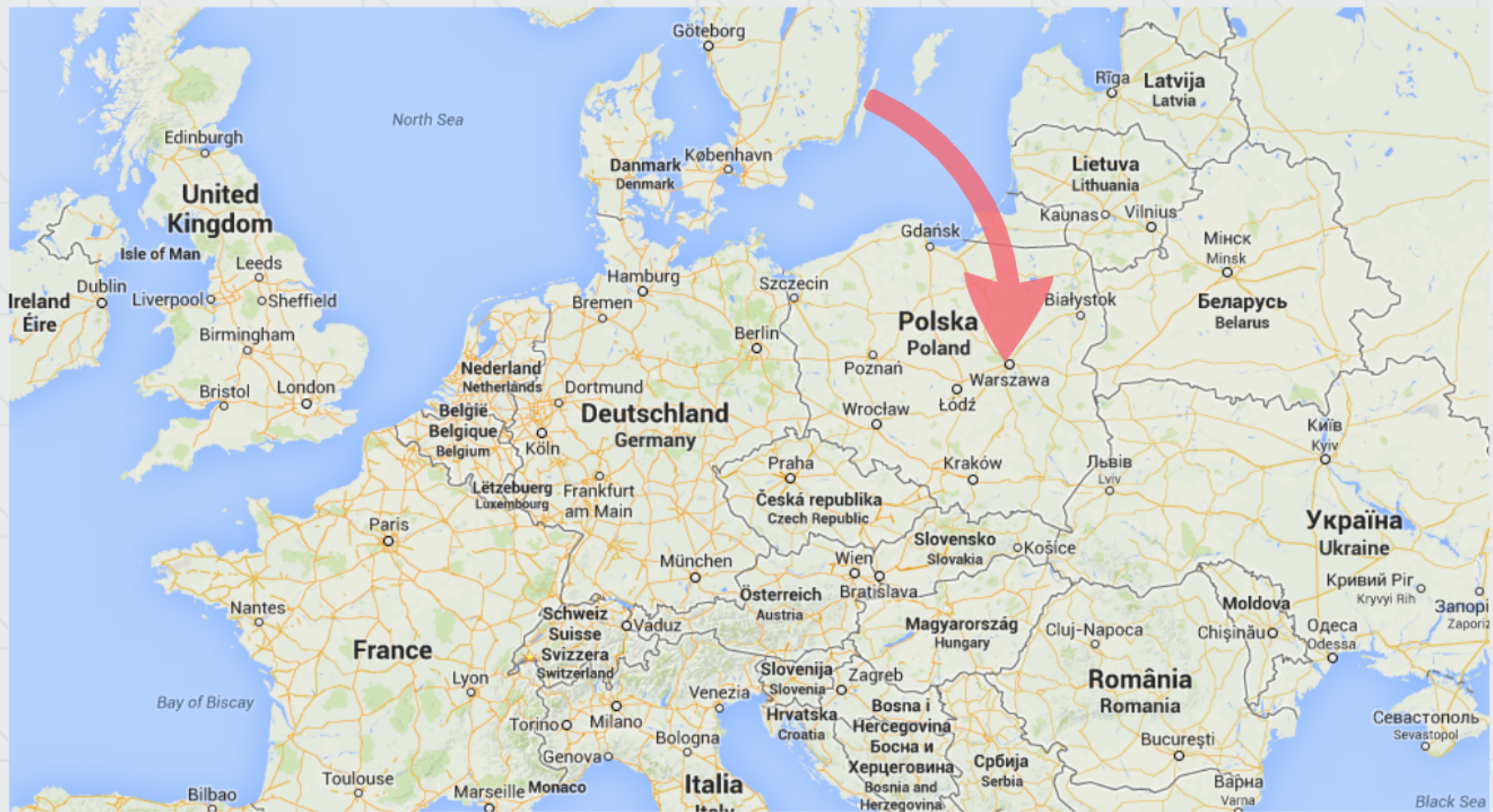


www.icm.edu.pl



www.plgrid.pl

Where is Warsaw ?



maps.google.com

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 ?



www.icm.edu.pl



www.plgrid.pl

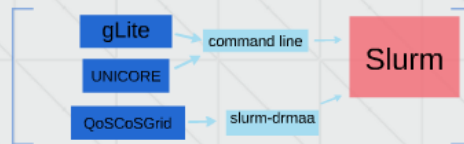
Hydra cluster - overview

- 4 kind of nodes (12,16,20,48,64 cores), 3 kind of GPUs (both Nvidia and AMD)
- disk-less setup based on NFS root and customized init
- nodes with and without infiniband (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-drmmaa)



slurmmon, OStrich, account backup generation

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



Thank you for your attention!



Our questions:

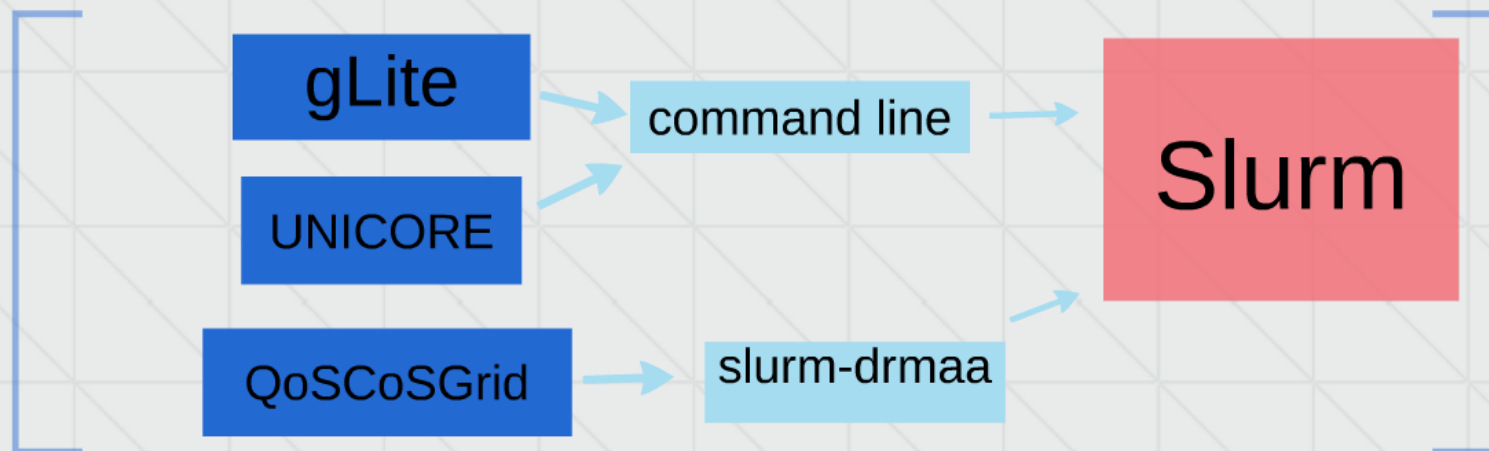
- How do you perform slurm profiling?
- Is it possible to use dynalloc from user application (API documentation)?

[/tmp and bindtmp.so plugin](#)

Middleware

We support three middleware:

- gLite (first production site in WLCG)
- UNICOR
- 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

How does it look like?

```
error: AKA, do
Worst random sleep to reduce load on state
sleep=$((RANDOM*2))

#Check if specified user can create/delete file in spec. location
chk, do findmnt -t xfs -o noexec,nowarn | grep noexec
chk, do findmnt -t xfs -o noexec,nowarn | grep noexec
chk, do findmnt -t xfs -o noexec,nowarn | grep noexec

#Check if filesystem - journal of output
error: AKA, do
chk, do findmnt -t xfs -o noexec,nowarn | grep noexec

#Check if there are processes not connected to running job (try to clean this)
error: AKA, do
chk, do findmnt -t xfs -o noexec,nowarn | grep noexec
```


How does it look like?

```
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
chk_dir /mnt/lustre/temp/plgmonitoring/ plgmonitoring
chk_dir /icm/home/mstol mstol
chk_dir /icm/tmp/mstol mstol
chk_dir /mnt/unicore/monitoring plgmonitoring

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

#Check if there are processes not connected to running jobs (try to clean this)
#Remove empty cpusets
source ./chk_cgroup
chk_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

How does it look like?

```
error: AKA, do
Worst random sleep to reduce load on state
sleep=$((RANDOM%30))

#Check if specified user can create/delete file in spec. location
chk, do mkdir -p /tmp/$(hostname)
chk, do touch /tmp/$(hostname)
chk, do rm /tmp/$(hostname)

#Check if filesystem - partition of output
error: AKA, do
chk, do df -h
chk, do echo

#Check if there are processes not connected to running job (try to clean this)
error: AKA, do echo
chk, do echo
```

/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/lfs_ \${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

example plugstack.conf:

```
optional bindtmp.so /tmp/lfs_  
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 istambul 12
feature westmere 12
feature magyrcours 48
feature interlagos 64
feature magyrcours_2300mhz_ferni_480 16
feature ivybridge 20
user plgnonitring

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/cinciek610/misc/tree/master/jobsubmit/job_sane

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

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/jobsubmit/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

Configuration file example:
feature istanbul 12
feature westmere 12
feature magyrcours 48
feature interlagos 64
feature magyrcours_2300mhz_ferni_480 16
feature ivybridge 20
user plgnonitring

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/cinciek610/misc/tree/master/jobsubmit/job_sane

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

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

Configuration file example:
feature istanbul 12
feature westmere 12
feature magyrcours 48
feature interlagos 64
feature magyrcours_2300mhz_ferni_480 16
feature ivybridge 20
user plgnonitring

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/cinciek610/misc/tree/master/jobsubmit/job_sane

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

Thank you for your attention!



Our questions:

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

Personal: bart@icm.edu.pl, mstol@icm.edu.pl
Admin list: hydra-admins@icm.edu.pl

Site report: ICM Warsaw University

ICM - where and who we are?

Dominik Bartkiewicz, Marcin Stolarek

ICM – Interdisciplinary Center for Mathematical and Computational Modeling

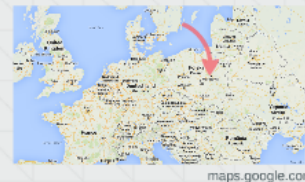
Five supercomputers:

- 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

Where is Warsaw ?



maps.google.com



www.plgrid.pl

Hydra cluster - overview

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

Middleware

- We support three middleware:
- jll Site (first production site in WLCG)
 - UNICOR
 - QCG (slurm-drmcas)



Healthcheck

- check functionality rather than state
 - first check - don't change anything, just check and log information in "thesauri" files
 - actually we try to find some problems
 - maybe add this to job epilog
- ### Black hole detection
- simple cron job, trying to submit jobs to non allocated nodes
 - send an email, drain node
- ### Reboot nodes
- look for nodes drained by "blackhole" or healthcheck
 - give it's cores, but we will probably switch to straggler

slurmon, OSTree, account backup generation



Job Submit Plugins

- What they provide:
- provide job resource of submit jobs
 - check if user have correct account
- Programming:
- we do it in C, don't know use and we don't have a lot of time for example that's why we decided to use in C++
 - we don't have enough time to write that's why we decided to use in C++

/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/lfs_\$(SLURM_JOB_ID)
- size determined by "--tmp="
- spack 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

example pluginstack.conf:
optional bindtmp.so /tmp/lfs
optional unshare.so lustre /mnt/lustre