

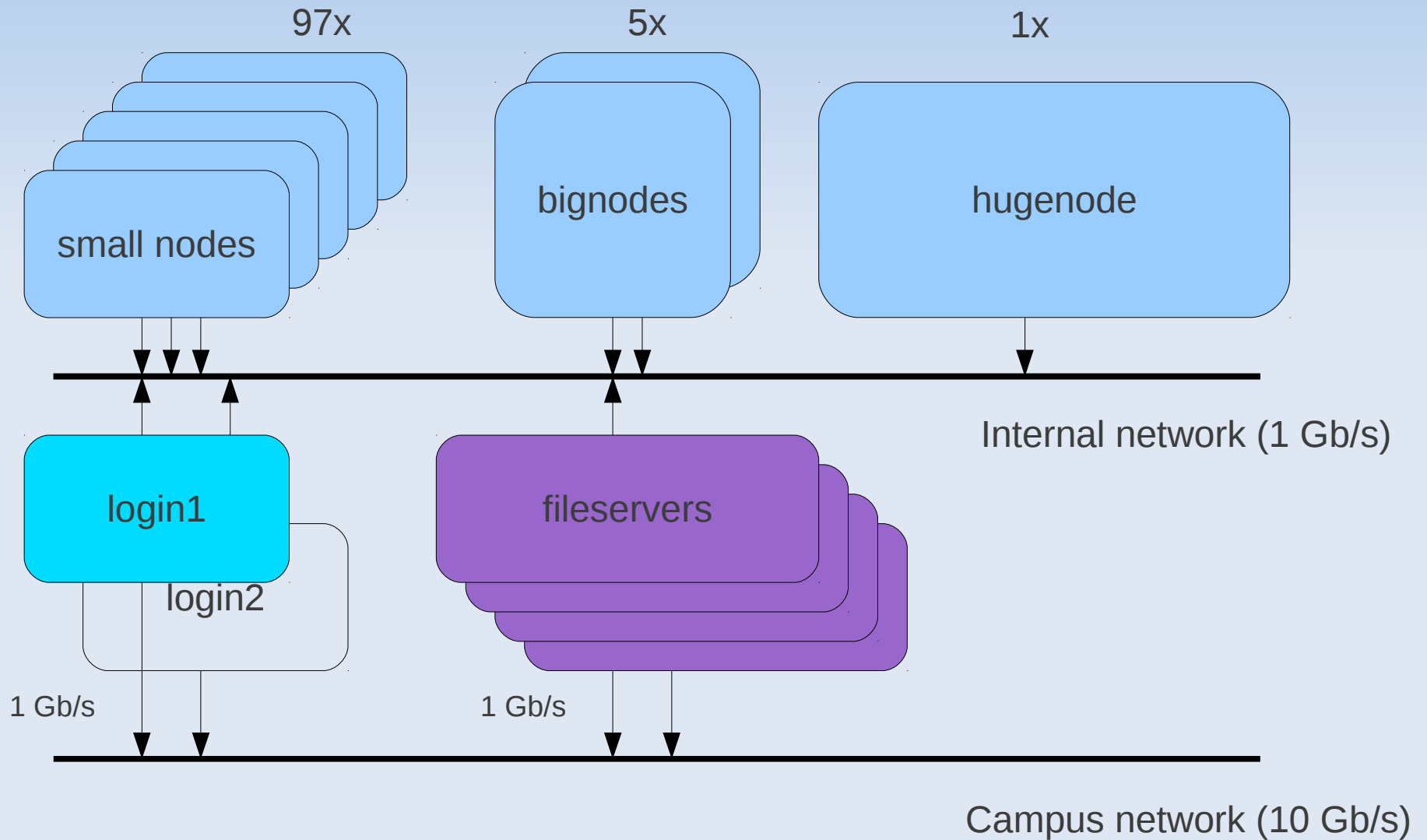
Cluster Intro

BIMSB cluster
How to use and not to use it

What is a cluster

- “... consists of a set of loosely connected or tightly connected computers that work together so that in many respects they can be viewed as a single system.” (Wikipedia)
- Each component (“node”) runs its own instance of operating system
- HPC clusters vs. HA clusters

Our cluster



Our cluster hardware

- Small nodes (node001-node097)
 - 2x quad-core AMD CPU
 - 16 – 64 GB RAM
 - ~ 200GB slow local disk (/tmp)
- Bignodes
 - 8 x dual-core AMD
 - 256 GB RAM
 - ~ 7TB fast local disk (/scratch)

Our cluster hardware

- Hugenode
 - 8 x 10 core Intel Xeon 2.4 GHz
 - 1 TB RAM
 - Medium speed local /scratch
 - Access restricted
- Storage
 - /home, /data/... ~ 2 PB (with online compression)

Grid Engine

- Queueing and resource management system
- Handles “best distribution” of compute jobs to the cluster nodes
- Univa Grid Engine, Sun Grid Engine, Oracle Grid Engine ...

Grid Engine concepts

- Queues:
 - NOT: list of jobs waiting to start
 - BUT: list of “slots” available for running jobs per machine
 - Aggregation of configuration settings
 - Memory available for jobs
 - Runtime limit
 - Access lists

Queues

	standard	interactive	longrun	high
Slots per node (small, big, huge)	8 / 16 / 80	1 / 1 / 0	1 / 1 / 0	2 / 4 / 0
Memory limit	RAM – 0.5 GB	8 GB / 64 GB @big	7 GB / 50 GB @big	62 GB
Runtime limit	96 h	48 h	250 d	21 d
Interactive usage	N	Y	N	Y
Restricted	N (Y 20 slots / user @huge)	3 slots per user	N	Y
			- Suspended if load high - Resource “longrun”	

Grid Engine concepts cont.

- Queue instances
 - Queue on a single node
 - “standard@node001”
- Resource
(aka. “Complex” or “Complex Resource”)
 - Represents an available resource
 - Physical (memory, swap ...)
 - Logical (“has scratch”, Matlab license ...)
- Slots
 - “slice” of computing power
 - Think “cores”

Important resources

What	GE Name	default	consumable	enforced
Virtual memory	h_vmem	1G	Y	Y
Runtime	h_rt	0:0:0	N	Y
Longrun queue	longrun	FALSE	N	N
Hugenode	huge	FALSE	N	N
Has /scratch	scratch	FALSE	N	N

Grid Engine concepts cont.

- Parallel Environments
 - Enable scheduling of jobs with “builtin parallelism”
 - “smp”
 - Normal multi-thread / multi-process jobs
 - Single node
 - “orte”
 - Multi-node for MPI-enabled programs
 - Not much experience
 - “make”
 - Multi-node for Makefile based workflows
 - Often inefficient because of network

GridEngine commands

- `qrsh`: interactive usage
- `qsub`: submit batch jobs
- `qalter`: alter settings of already submitted jobs
- `qstat`: get status of currently running and waiting jobs
- `qacct`: get accounting info / info on finished jobs
- `qmon`: Graphical interface to Grid Engine

qrsh

- Normal case: **1** slot in “interactive” queue
- `qrsh -now no`
 - Does not need to start immediately
 - Runs in standard queue
 - Can request more than 1 slot
 - Calling shell blocked until qrsh finished
- For X11 programs:
 - `ssh with X11 forwarding to login1`
 - “`qrsh <options> <program name>`” or
 - “`qrsh <options> bash`” → **node** → “program”

qsub

- Submits a batch job
- Lots of options → “man qsub”
- “binary” vs. job script:
 - “-b y” / “-b n”
 - Binary only transmits path to command / script
 - Job script can set job parameters (“special comments”)
 - Binary is more effective for short running jobs

qstat

- Without options → list of all waiting or running jobs of current user
- `qstat -j <job_id>` gives details of job and scheduler decisions
- `qstat -g c` : Overview of all queues
- `man qstat`

qacct

- Analyzes “accounting” file
- Useful for error finding: `qacct -j <job_id>`

```
qname          standard
hostname       bignode04.mdccluster
group          ohler
owner          scordoba
project        NONE
department     ohler
jobname        scordoba_Cuffdiff_rRNA_replicates.jb
jobnumber      2628966
taskid         undefined
account        sge
priority       0
qsub_time      Mon Jul 29 11:12:49 2013
start_time     Tue Jul 30 00:04:59 2013
end_time       Tue Jul 30 00:04:59 2013
```


qacct -j

```
granted_pe    smp
slots        8
failed       0
exit_status   0
...
maxvmem      0.000
```

qmon

- Graphical interface to many GridEngine functions
 - `qstat`, `qsub`, `qconf`
- Needs X11 tunnel (`ssh -X login1`)



qmon

The screenshot shows the QMON Job Control window. The title bar reads "QMON +++ Job Control". The window features the "grid engine" logo and a "Job Control" label. Below the logo are three tabs: "Pending Jobs", "Running Jobs" (which is selected), and "Finished Jobs". The main area contains a table with the following columns: JobId, Priority, JobName, Owner, Status, and Queue. The table lists 20 jobs, all with a status of "r". To the right of the table is a vertical panel of control buttons: Refresh, Submit, Tickets, Force (with a checkbox), Suspend, Resume, Delete, Reschedule, Select All, Why?, Hold, Priority, Qalter, Clear Error, Customize, Done, and Help.

JobId	Priority	JobName	Owner	Status	Queue
2629172	1.05253	QRLOGIN	ikel	r	interactive@node
2629933	1.05003	bigwigs.sh	rbeagri	r	standard@node057
2605032	0.88335	cuff	marodrig	r	longrun@bignode0
2629166	0.55275	QRLOGIN	ekadikow	r	interactive@node
2629598	0.55210	flexbar.sh	ekadikow	r	P:standard@bigno
2601803	0.35000	QLOGIN	mschuele	r	standard@hugenod
2622430	0.28539	R	mschuele	r	standard@hugenod
2625784	0.26469	R	mschuele	r	standard@hugenod
2628032	0.26118	R	mschuele	r	standard@hugenod
2628393	0.25419	runs_bayes	trito	r	standard@node074
2628396	0.25418	runs_bayes	trito	r	standard@node096
2628968	0.25288	QLOGIN	mschuele	r	P:standard@hugen
2629163	0.25282	runs_bayes	trito	r	standard@node037
2629164	0.25282	runs_bayes	trito	r	standard@node082
2629165	0.25282	runs_bayes	trito	r	standard@node022
2629951	0.19286	log.u2r.s6	binzhang	r	P:standard@node0
2629953	0.19286	log.u2r.WC	binzhang	r	P:standard@bigno

Array jobs

- Same program on different inputs
- `qsub -t n-[m[:s]] <program>`
`(-tc max_running_tasks)`
- `$SGE_TASK_ID` **inside job**
- `qstat -t`
- Easier to handle for GridEngine and user

SMP jobs

- Multi-core computation
(multiple processes, pthreads, OpenMP ...)
- `qsub -pe smp n [-m]`
- `h_vmem` **per slot!**
- User is responsible for setting correct number of threads in application
- `$NSLOTS`

Holds

- Submitting job with hold : `qsub -h ...`
- Releasing hold: `qrls -j <job_id>`
- Waiting for other job:
`qsub -hold_jid wc_job_list`
- Waiting for corresponding task of other job:
`qsub -hold_jid_ad wc_job_list`

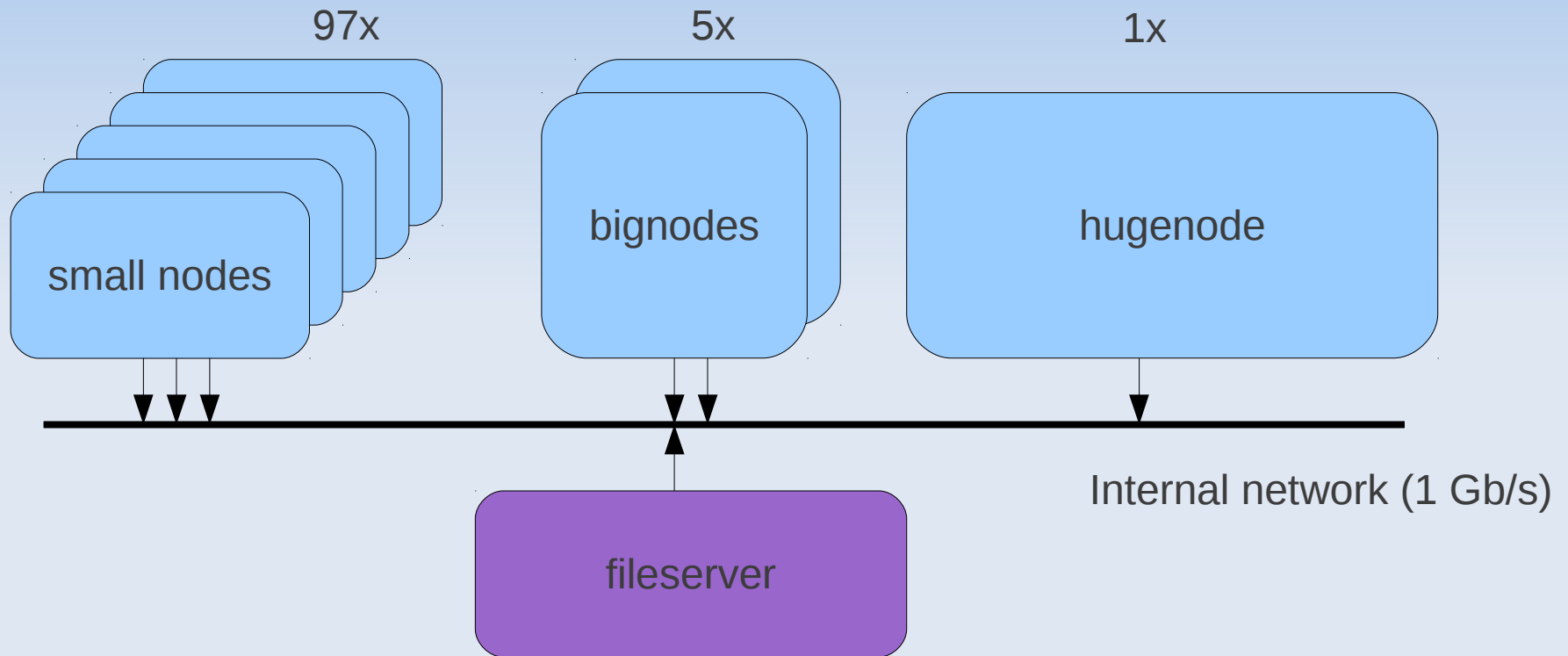
Why ...

- ... did my job fail?
 - Look at jobs error file
(default `<job_name>.e<job_id>`)
 - Look at `qacct -j <job_id>` :
 - Runtime above limit ?
 - Maxvmem above requested limit?
 - Exit status ? (e.q. 137 means SIGKILL)
 - What host ?
 - At host: `grep` for `job_id` in
`/var/spool/sge/nodeXXX/messages`
 - Ask Sysadmin (with all informations)

Why ...

- ... do my jobs not start?
 - Cluster full ? (“`qstat -u *`” / Ganglia)
 - Needed resources not available (`qstat -j ...`)
 - Needed resources not available at the moment
 - `qstat -j ...`
 - `qhost -q -F h_vmem`
 - Job on hold?

NFS bottleneck



NFS bottleneck

- Hard to avoid
- Monitor Wait-IO on cluster (ganglia)
- Reduce number of tasks / jobs, work with holds
- Staging (/scratch or /tmp)

Ganglia

- Monitoring load on cluster nodes / servers
- Can give you an idea about Wait-IO
- <http://141.80.186.22/ganglia>

Rules

- No computation on login node!
- No computation outside of GridEngine!
 - (background jobs, `screen` command)
- Try to avoid NFS bottleneck!
- Monitor your job!
- Use brain!
- If you have questions, ask!

Further information

- Man-pages
- <http://bbc.mdc-berlin.de/howto/cluster>

Outlook

- End of 2013 new cluster:
 - ~ 100 nodes
 - Partition with Infiniband interconnect (for MPI jobs)
 - Fileservers connected with 10 Gbs
 - Fast distributed filesystem to avoid NFS-bottleneck
 - Broader audience (whole MDC)

Questions

???