X Grid Engine

Where
X
stands for
Oracle
Univa
Open
Son of
more to come....?!?

Carsten Preuss
on behalf of
Scientific Computing
High Performance Computing

Scheduler candidates

- LSF
  - too expensive
- PBS / Torque
  - not really stable
- Condor
  - too complicated
  - HTC scheduler (HPC one needed)
- (S)GE
  - future unknown (initial developer Sun was bought by Oracle during the test phase)
- Slurm

Clusters

- LSF cluster
  - 240 nodes, 8 cores, 16/24/32 GB
  - 2200 cores at all
    - Slot based scheduling
  - SGE cluster for Alice Grid jobs on top of it
    - Load based scheduling
  - /u, /d, /misc and /user/local

Clusters II

- GE cluster
  - 100 nodes, 16 cores each, 32GB
    - Resource based scheduling
- new GE cluster
  - 400 nodes, 24 cores, 64 GB
    - 9600 cores at all
    - Infiniband instead of Ethernet
    - resource based scheduling

- both cluster don't have any NFS system (not even /u) mounted
  - only Lustre for data
  - Software comes from CVMFS (CERN Virtual Maschine File System)
    - HTTP based read only file system
    - software will be installed via build servers
      - will be administrated by the experiments

New cluster

- MiniCube build up inside the Testinghalle
  - new cooling concept
  - will have it's own Lustre cluster
  - everything connected via Infiniband
  - will be operational in April (this year ;-))

Queue concept

- the new cluster(s) do not have experiment specific queues anymore
- all queues are resource based ones

- old queue system was sometimes reason for underutilization of the farm even if jobs were waiting for scheduling
- the slots per experiment are now done via Fair Share

Cluster overview

- qstat -g c
  - will give an overview over the available queues and the associated slots
- qhost
  - shows all available batch nodes with load indices
- qhost -j -q (-h lxbXXX)
  - shows all hosts with their queue instances and jobs running on them

User groups

- Users are differed into 3 groups
  - Default
    - can use up to 100 slots at a time
  - Advanced
    - 400 slots
  - Power
    - can use more or less all slots and resources

  - for the new cluster the rules have to be adapted

CVMFS

- software is installed on so called build servers
- by the IT department
  - /cvmfs/it.gsi.de/oracle/..
- experiments
  - /cvmfs/alice.gsi.de/aliroot/....
  - /cvmfs/hades.gsi.de/...
  - /cvmfs/fairroot.gsi.de/...
  - ….

Submitting a job

- login to lxsqueezelust64 (pool)
- qsub /bin/hostname
  - Unable to run job: Script length does not match declared length. Exiting.
- What's this?
  - GE expects you to submit a script not a binary
- qsub -b(inary) y(es) /bin/hostname
- But wait! What about the output? It will be written to /u/cpreuss which does not exists on the cluster! The job will fail....
- qsub -wd /lustre/rz/cpreuss/ -b y /bin/hostname
- Or go to your target directory and use the -cwd option to qsub

Submitting a job II

- differences in submitting scripts and binaries
  - GE will copy the submitted script to the batch node
    - but only the submitted script, be carefull if your script calls another script!
  - a submitted binary will not be copied, so you have to use the path/name on the target batch host
  - if you have a script which is already on the cluster you can use it via the -b y option, GE will not mind.

Submitting jobs III

- in LSF you had to define a queue which your job should go to
- in GE you can't define queues (they exist but only for administrative usage)
- what you should define are the resource requirements

Submitting jobs IV

- h_cpu CPU time
- h_rt Runtime
- h_rss real memory
- h_vmem virt. Memory
- and many more, one can use (qconf -sc will show all variables)
- be carefull these requirements are needed to sort your jobs to queues which fits to the job needs
- these requirements are limits, too!
- exceeding them and your jobs will be killed by GE
- qsub -l h_rt=1:0:0,h_vmem=2G ….
  - 1 hour runtime and 2 GB of virt. memory

qsub V

- -wd /tmp OR /lustre/....
- -cwd
  - if you are already in the target directorie
- -terse
  - will return only the jobID
    - nice for automated job submitting and controlling
- -w e | w | v
  - will verify your jobs and report
    - Errors
    - Warnings
    - just verify, don't submit anything

qsub VI - Job Arrays

- qsub -t 1-5:2 …..
  - more or less the same as in LSF
    - (but has to start with 1 or greater, not 0)
  - starts one job with 3 tasks
    - task IDs will be 1,3,5
    - complete job is selected by the jobID
    - tasks by jobID.taskID (not jobID[taskID] like in LSF)
- qsub -t 1-5 -tc 2 ….
  - will run only 2 tasks at a time
  - for job throttling

qsub VII – Dependency Jobs

- dependency jobs are jobs which are waiting for others to finish
- in GE the dependency jobs can only wait till the defined job is finished complete
  - no possibility to define the job status (Exit/Done)
  - also not possible to define a number of tasks to be finished if the job is waiting for an array
- qsub -hold_jid jobID ….
  - will pend in state 'hqw' untill job jobID has finished

qsub VIII – Parallel environments

- in GE we support parallel environments
  - qsub -pe openmpi 10-20 …
  - will submit a job with min. 10 and max. 20 tasks, depending on how full the cluster is
  - the tasks will be clustered on single nodes according to the allocating strategy used by the PE openmpi

Controlling jobs

- qstat -j jobID
  - will show details for the job
  - if the job is pending GE will show you all reasons why
    - a very, very longish output.....
  - if the job is running
    - qstat -j jobID | grep usage
    - shows the current used resources
    - can be used to monitor the jobs in real time
  - or
    - qstat -j jobID -ext
  - Done jobs can be shown only via
    - qstat -s z
      - it is NOT Zombie as stated in the manual!
      - GE does not know the states Zombie and UNKWN like LSF

Modifying jobs

- if you see that the given parameters for your jobs don't make sense you can just change them, no need to kill and submit everything again

- qalter -j h_rss=4G jobID
  - will change the memory requriement

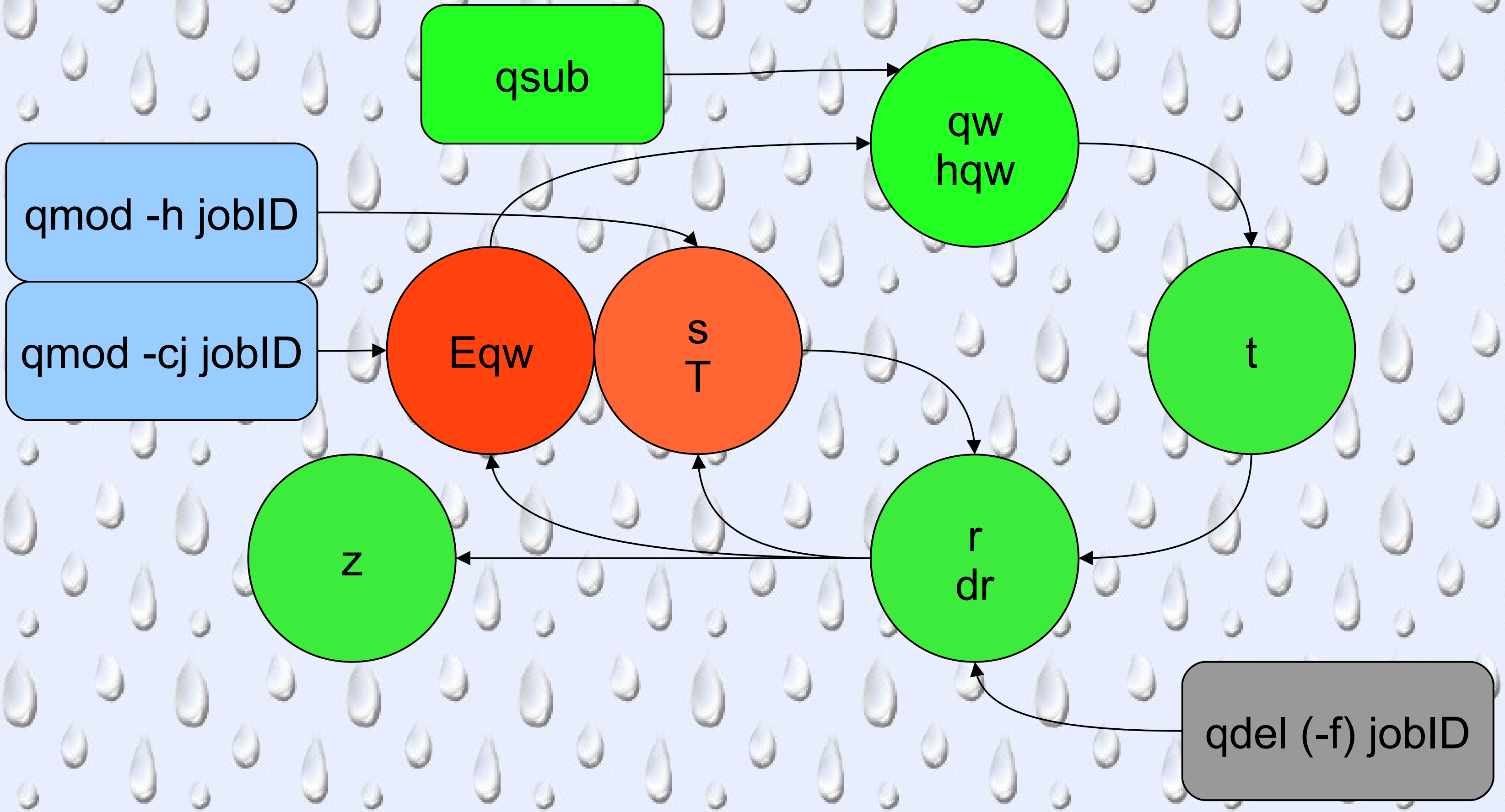
- all parameters can be altered, but some only on pending jobs

Suspending/deleting/resubmiting jobs

- qmod -sj jobID
  - job will go to state 's'
- qmod -usj jobID
  - and the job will run
- qdel (-f) jobID
  - will delete a job, using '-f' will force this action
- qresub jobID
  - this will resubmit an exact copy of a job

Job lifecycle

The jobs can (and will) have several states :



Prioritys and Fair Share

- as written some pages before the experiments don't have a dedicated amount of slots for their usage
- instead of this the complete cluster is now allocated to the users via Fair Share
- advantages
  - users can use big parts of the cluster (if it is empty)
  - large consumers are penalized

Prioritys and Fair Share

- qsub -P highprio ….
  - will account your resource usage to the Alice project 'highprio'
  - without giving a project the resource usage is accountet to the default project
- jobs are sorted in order to their priority
  - qstat -u '*'
    - shows in the second column the job priority
- the spreading of the shares over the different departments will/should be discussed by the department haeds in the future.....

Infos about done jobs

- qstat -s z
  - shows only a list of done jobs
  - no additional infos
- qacct -j jobID
  - should be used to retrieve more detailed infos like exit status and used resources
  - also a resource accumulation is available
  - unfortunately qacct can't be used by users at the moment

Some remarks...

- the help for a command is always 'qsub -help' not -h nor –help
- it is 'qstat -u ….' but 'qacct -o ….'
- using qstat shows the submit time/date on pending jobs but the start time/date on running jobs
- qacct can't be used at the moment
- do not use queus
- give fair resource requirements to your jobs, not too much but also not too less
- there are no user preexecution scripts anymore
- selecting all jobs/users/hosts is done via '*' not 0 (like in LSF)

...and some Do's and Dont's

- submit test jobs before starting large production
  - checking if the job will run at all
  - checking the resource requirements
- if submitting several equal jobs, use job arrays instead of single jobs
- don't get output via mail
- use the local disk /tmp with care
  - renember that there will be 24 jobs using a single disk, no RAID5
- have an eye on your production
  - don't assume that everything will go smooth
- if there is something going wrong – tell us!
  - don't wait until everything is going down the drain