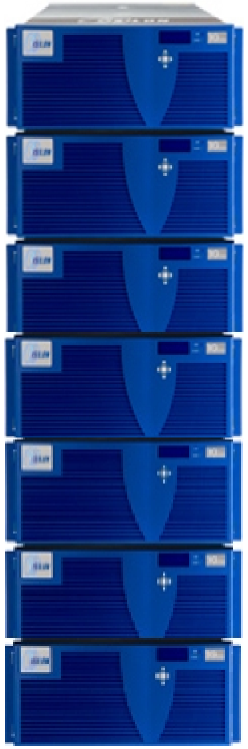
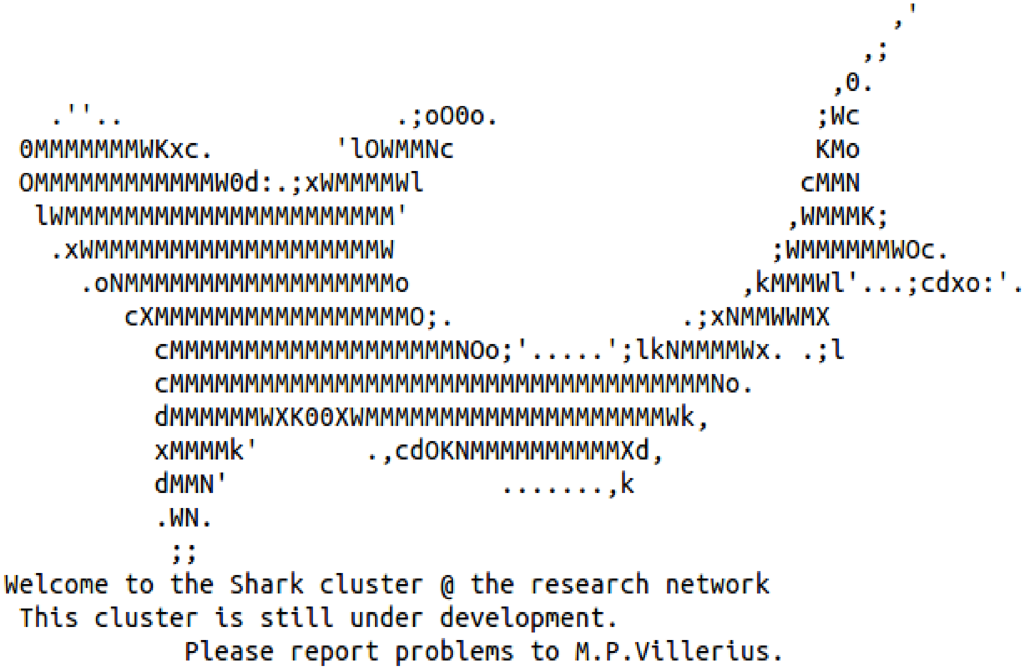
Shark Cluster Overview



36 Execution Nodes
1 Head Node (shark)
1 Graphical login node (rivershark)
596 Cores = slots
437 TB Storage



```
                                                           ,'
                                                         ,;
                                                       ,0.
   .''..                  .;oO0o.                     ;Wc
 0MMMMMMMWKxc.        'lOWMMNc                        KMo
 OMMMMMMMMMMMMW0d:.;xWMMMMWl                         cMMN
  lWMMMMMMMMMMMMMMMMMMMMMM'                         ,WMMMK;
   .xWMMMMMMMMMMMMMMMMMMMW                         ;WMMMMMMWOc.
     .oNMMMMMMMMMMMMMMMMMMo                      ,kMMMWl'...;cdxo:'.
        cXMMMMMMMMMMMMMMMMMO;.               .;xNMMWWMX
          cMMMMMMMMMMMMMMMMMMMNOo;'.....';lkNMMMMWx. .;l
          cMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMNo.
          dMMMMMMWXK00XWMMMMMMMMMMMMMMMMMMMMWk,
          xMMMMk'       .,cdOKNMMMMMMMMMMXd,
          dMMN'                  .......,k
          .WN.
           ;;
Welcome to the Shark cluster @ the research network
 This cluster is still under development.
            Please report problems to M.P.Villerius.
```

Introduction

What is a cluster?

- A cluster is a group of computers connected with a tcp/ip-network.
  - The group can work as one large computer or as singe computers called nodes.
  - The cluster is controlled by a resource manager Open Grid Scheduler

There are different type of nodes

- Head node
  - Uses a resource manager to manage all the resources
- Compute node
  - A single blade server
- Login node
  - Shark.researchlumc.nl / rivershark.researchlumc.nl
- Storage node
  - Isilon storage

Key Terms

Cluster = are a groups of PC's coupled together
Blade server = a compact like workstation (PC)
Chassis = where we store max. 16 blade servers together
Node = is a blade server attached to a cluster
Head node = is the controller/master node of a cluster
Execute node = is the blade that calculate your jobs
Worker node = same as execution node
Storage node = is a special server that will store your data on hard disks
CPU = Central Processing Unit
Core = is another name for the CPU
Slot = in a cluster one CPU is also called a SLOT
Job = your job that needs to be calculated
Queue = a waiting line for all jobs (jobs that cannot be processed)
Fair share = everyone gets CPU power according to what the paid for.
SSH = Secure Shell, a protocol to connect to a server
OGS = Open Grid Scheduler/Grid Engine batch-queuing system for distributed resource management.

Open Grid Scheduler

- OGS is a scheduler which divides the available resources(CPU,mem,etc) into queues
- OGS places the compute tasks (jobs) in a large list.
- OGS searches the right queue for the requested resources in a job and starts that job.
- When the resources are full, the jobs are placed in a waiting list
- Jobs with a higher priority are placed higher in the list
- The priority is based on the priority resource and the fair-share policy

Where can I place my data?

Directory	**Description**	**Size**
/exports/	Mount point for Isilon evey department has his own bought isilon export mounted here	715TB
/home/	Home dir, can hold: scripts, R library, compiled code	2TB
/bam-export	Everyone can save BAM files here that can be reached from the UCSC genome browser. External address : https://barmsijs.lumc.nl/..........	1TB

Location of important directories

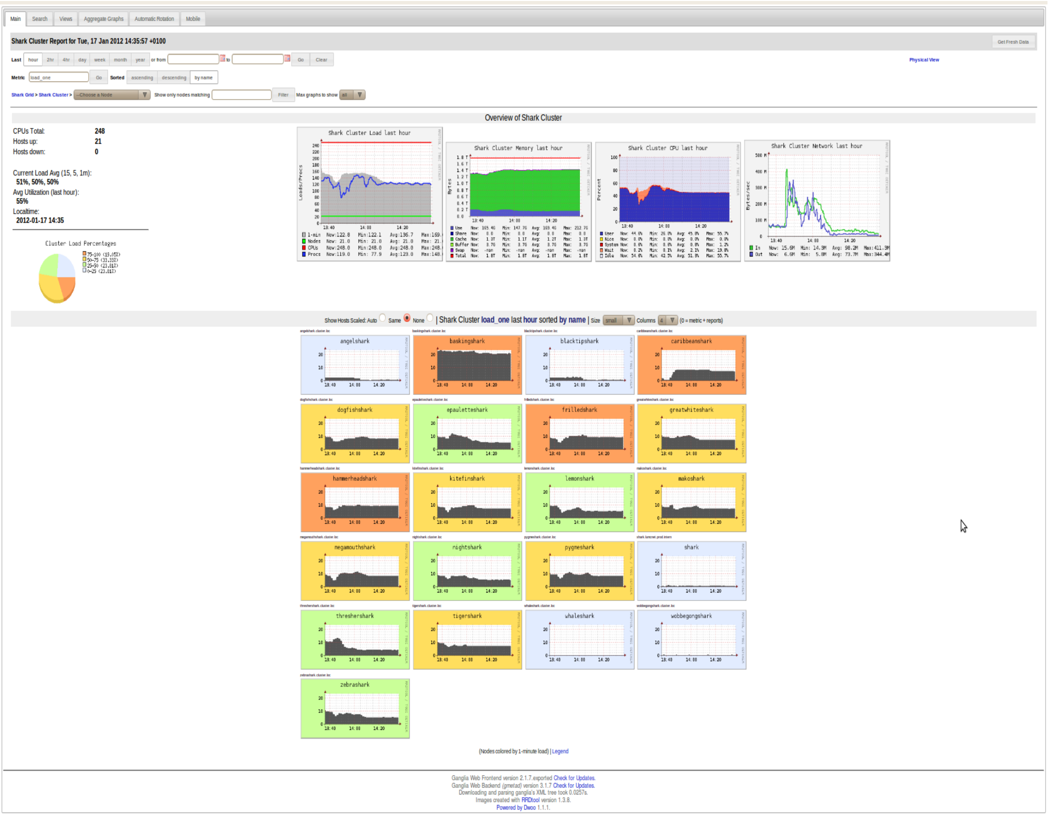
- /usr/local *Where programs are located*
- /usr/local/Genomes *Where the reference genomes are located*

What is your responsibility

- Describe what resources are necessary to ensure success for your jobs (memory, run time, disk space, number of cores, CPU usage, network)
- Not to overload the systems resources, CPU, mem, disk space, network, run time
- Kill the job if the system is getting overloaded
- Keep track of jobs when you are not sure what resources the job uses

Important Note :
qdel “JOB_ID” to kill a job
qdel -u “your username” to kill all your jobs

http://shark.lumcnet.prod.intern/ganglia/



Shark Cluster Report for Tue, 17 Jan 2012 14:35:57 +0100

Overview of Shark Cluster

CPUs Total: 248
Hosts up: 21
Hosts down: 0

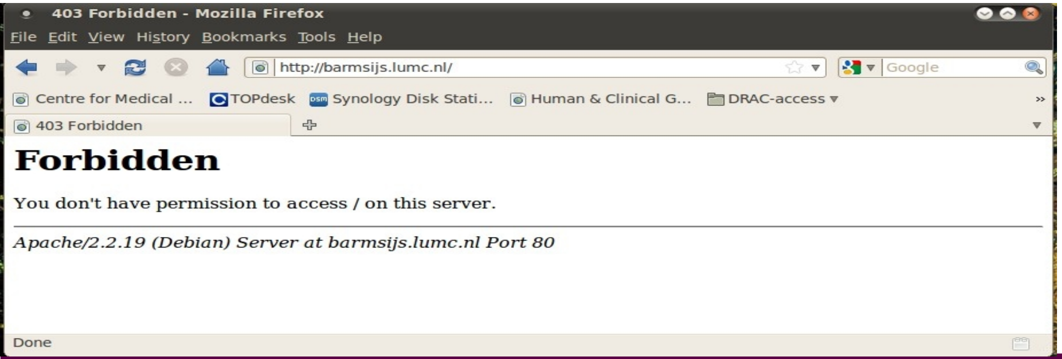
Current Load Avg (15, 5, 1m):
51%, 50%, 50%
Avg Utilization (last hour):
55%
Localtime:
2012-01-17 14:35

| Shark Cluster load_one last hour sorted by name |

(Nodes colored by 1-minute load) | Legend

Ganglia Web Frontend version 2.1.7.exported Check for Updates.
Ganglia Web Backend *(gmetad)* version 3.1.7 Check for Updates.
Downloading and parsing ganglia's XML tree took 0.0257s.
Images created with RRDtool version 1.3.8.
Powered by Dwoo 1.1.1.

https://barmsijs.lumc.nl/

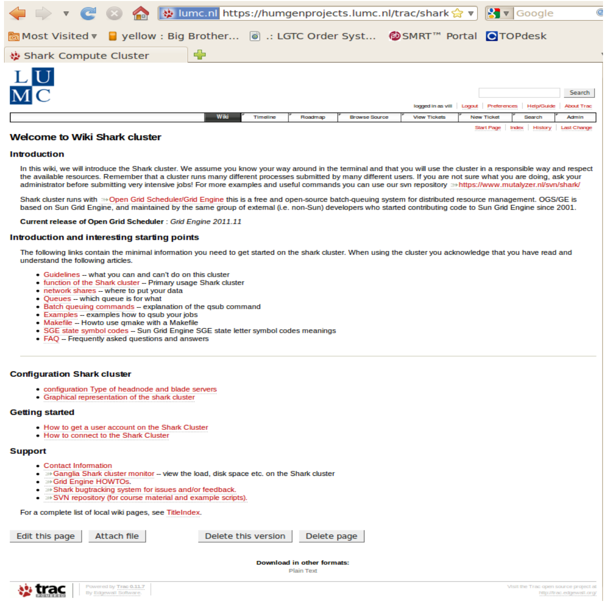


Forbidden

You don't have permission to access / on this server.

Apache/2.2.19 (Debian) Server at barmsijs.lumc.nl Port 80

https://humgenprojects.lumc.nl/trac/shark/



Welcome to Wiki Shark cluster

Introduction

In this wiki, we will introduce the Shark cluster. We assume you know your way around in the terminal and that you will use the cluster in a responsible way and respect the available resources. Remember that a cluster runs many different processes submitted by many different users. If you are not sure what you are doing, ask your administrator before submitting very intensive jobs! For more examples and useful commands you can use our svn repository ⇒https://www.mutalyzer.nl/svn/shark/

Shark cluster runs with ⇒Open Grid Scheduler/Grid Engine this is a free and open-source batch-queuing system for distributed resource management. OGS/GE is based on Sun Grid Engine, and maintained by the same group of external (i.e. non-Sun) developers who started contributing code to Sun Grid Engine since 2001.

Current release of Open Grid Scheduler : *Grid Engine 2011.11*

Introduction and interesting starting points

The following links contain the minimal information you need to get started on the shark cluster. When using the cluster you acknowledge that you have read and understand the following articles.

- Guidelines -- what you can and can't do on this cluster
- function of the Shark cluster -- Primary usage Shark cluster
- network shares -- where to put your data
- Queues -- which queue is for what
- Batch queuing commands -- explanation of the qsub command
- Examples -- examples how to qsub your jobs
- Makefile -- Howto use qmake with a Makefile
- SGE state symbol codes -- Sun Grid Engine SGE state letter symbol codes meanings
- FAQ -- Frequently asked questions and answers

Configuration Shark cluster

- configuration Type of headnode and blade servers
- Graphical representation of the shark cluster

Getting started

- How to get a user account on the Shark Cluster
- How to connect to the Shark Cluster

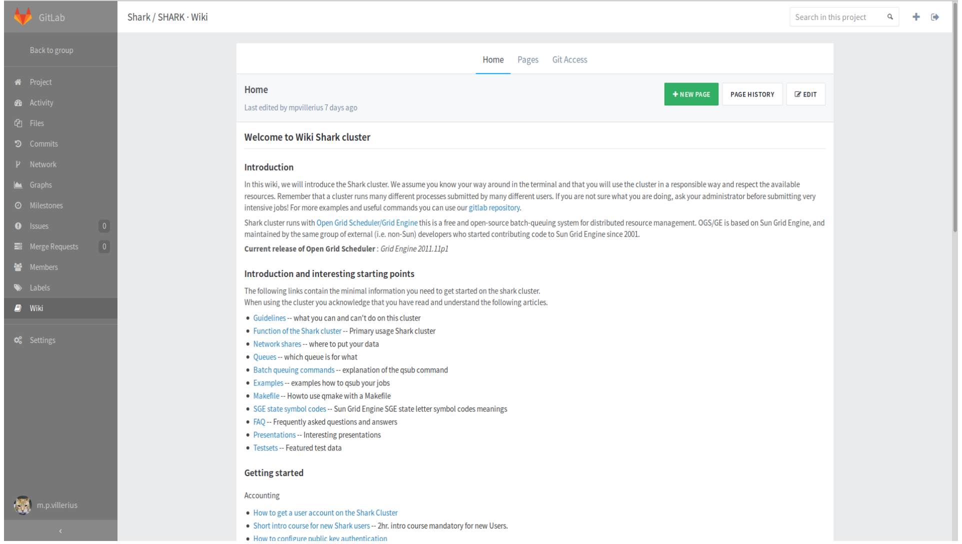
Support

- Contact Information
- ⇒Ganglia Shark cluster monitor -- view the load, disk space etc. on the Shark cluster
- ⇒Grid Engine HOWTOs.
- ⇒Shark bugtracking system for issues and/or feedback.
- ⇒SVN repository (for course material and example scripts).

For a complete list of local wiki pages, see TitleIndex.

Download in other formats:
Plain Text

https://git.lumc.nl/shark/SHARK/wikis/home



Home

Last edited by mpvillerius 7 days ago

Welcome to Wiki Shark cluster

Introduction

In this wiki, we will introduce the Shark cluster. We assume you know your way around in the terminal and that you will use the cluster in a responsible way and respect the available resources. Remember that a cluster runs many different processes submitted by many different users. If you are not sure what you are doing, ask your administrator before submitting very intensive jobs! For more examples and useful commands you can use our gitlab repository.

Shark cluster runs with Open Grid Scheduler/Grid Engine this is a free and open-source batch-queuing system for distributed resource management. OGS/GE is based on Sun Grid Engine, and maintained by the same group of external (i.e. non-Sun) developers who started contributing code to Sun Grid Engine since 2001.

Current release of Open Grid Scheduler : *Grid Engine 2011.11p1*

Introduction and interesting starting points

The following links contain the minimal information you need to get started on the shark cluster.
When using the cluster you acknowledge that you have read and understand the following articles.

- Guidelines -- what you can and can't do on this cluster
- Function of the Shark cluster -- Primary usage Shark cluster
- Network shares -- where to put your data
- Queues -- which queue is for what
- Batch queuing commands -- explanation of the qsub command
- Examples -- examples how to qsub your jobs
- Makefile -- Howto use qmake with a Makefile
- SGE state symbol codes -- Sun Grid Engine SGE state letter symbol codes meanings
- FAQ -- Frequently asked questions and answers
- Presentations -- Interesting presentations
- Testsets -- Featured test data

Getting started

Accounting

- How to get a user account on the Shark Cluster
- Short intro course for new Shark users -- 2hr. intro course mandatory for new Users.
- How to configure public key authentication

How to connect to Shark from LUMC

- You need to use your RESEARCHLUMC username and password to connect to the Shark cluster

- Linux / MAC use SSH protocol
  - Command in terminal : ssh researchlumcusername@145.88.65.151
- Windows
  - use putty, cygwin or mobaxterm
    host : shark.researchlumc.nl
    port : 22

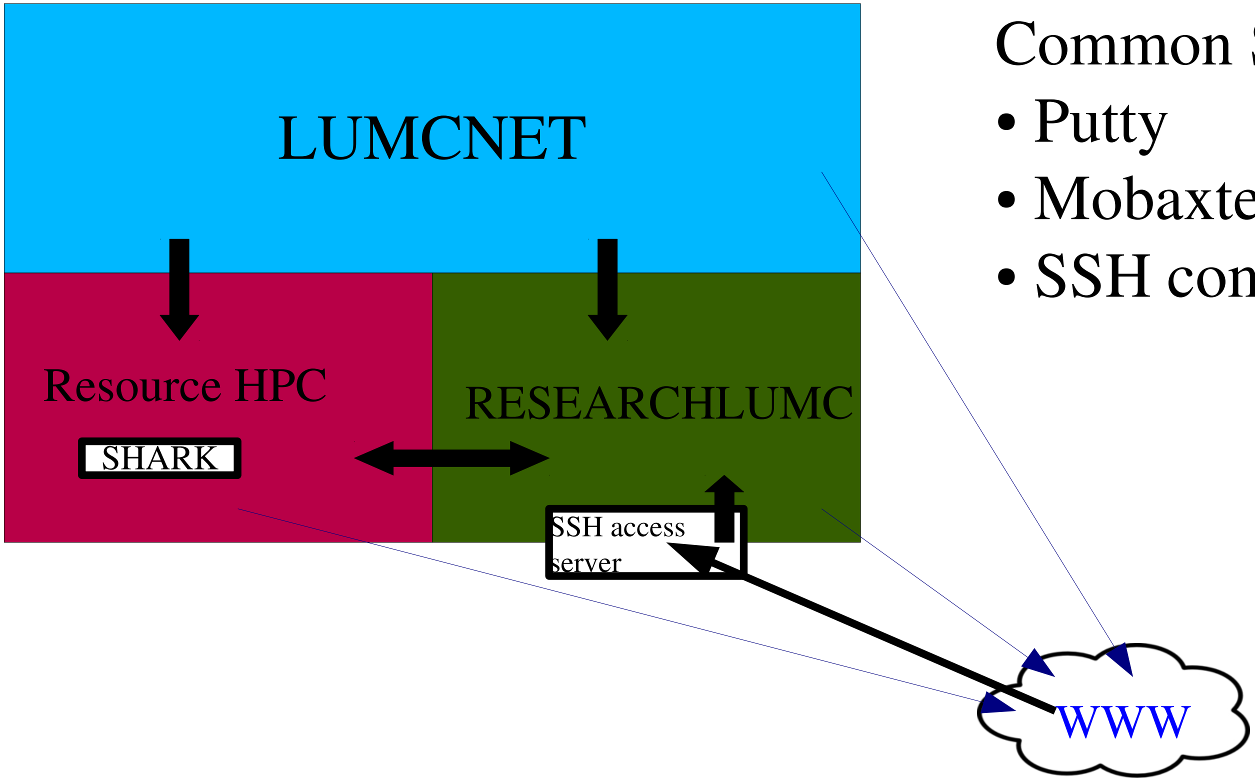
- Login from outside the LUMC
  - Use the SSH jump server : ip = 145.88.35.10

SSH schema

- SSH   Default SSH port : 22
- SFTP
- SCP
- SSHFS

SSH server properties

- 3 times wrong user name / session : ip banned
- 3 times wrong password / session : ip banned



Common SSH programs

- Putty
- Mobaxterm
- SSH command

Getting started

Important Note : default memory of every job is set to 3GB / slot

qlogin.q and notebook.q maximum run time 12 hrs.

QUEUES

```
username@shark:~$ qstat -g c
CLUSTER QUEUE       CQLOAD   USED       RES   AVAIL      TOTAL      aoACDS      cdsuE
--------------------------------------------------------------------------------------
all.q               0.26     195        0     368        563        0           0
notebook.q          0.63     1          0     8          9          0           0
qlogin.q            0.20     12         0     8          20         0           0
```

Parallel environments

```
username@shark:~$ qconf -spl
Name      allocation rule
BWA       pe_slots
make      pe_slots
mpich2    round_robin
```

Interactive use of SGE

```
username@shark:~$ qlogin
Your job 822125 ("QLOGIN") has been submitted
waiting for interactive job to be scheduled ...
Your interactive job 822125 has been successfully scheduled.
Establishing /usr/local/Scripts/qlogin_wrapper session to host angelshark.researchlumc.nl ...

username@angelshark:~$ qstat
Job-ID     prior   name       user     state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
 822125 0.50075 QLOGIN     username  r     06/04/2014 13:17:25 qlogin.q@angelshark.researchlumc.nl      1

username@shark:~$ qlogin -pe BWA 4 -q all.q
Your job 110997 ("QLOGIN") has been submitted
waiting for interactive job to be scheduled ...
Your interactive job 110997 has been successfully scheduled.
Establishing built in session to host angelshark.researchlumc.nl ...
vill@angelshark:~$

username@angelshark:~$ qstat
job-ID  prior      name     user     state submit/start   at       queue                          slots ja-task-ID
-------------------------------------------------------------------------------------------------------------------
 110997 0.53227 QLOGIN     vill        r     08/03/2011 03:13:59 all.q@angelshark.researchlumc.nl     4
```

Important Note : default memory of every job is set to
3GB / slot

Qsub your job

Only a script can be submitted with qsub !!!!!!!!

Syntax qsub command:

qsub [options] [scriptfile | -- [script args]]

Binary files can not be submitted directly to SGE.

Example submit "date" command to SGE (with script):

```
#!/bin/bash
/bin/date
```

Submit this script with:

```
$ qsub sge-date
```

SGE will then run the program, and place two files in your home directory:

sge-date.e   sge-date.o

The most common options:

-N [name]	-- The name of the job
-l h_rt=hr:min:sec	-- Maximum walltime for this job
-l h_vmem	-- Maximum memory your job can use
-r [y,n]	-- Should this job be re-runnable (default y)
-pe [type] [num]	-- Request [num] amount of [type] nodes.
-cwd	-- Place the output files (.e,.o) in the current working directory. The default is to place them in the users home directory.
-S [shell path]	-- Specify the shell to use when running the job script

Important Note : default memory of every job is set to 3GB / slot
-l h_vmem=3G

Qsub your job

Qsub options can be placed inside your script.

```
vill@shark:/home/vill/shark/trunk# cat run_my_first_job.sh
#!/bin/bash
#$ -S /bin/bash
#$ -q all.q <= use with care
#$ -N my_first_job
#$ -pe BWA 4 <==== the number 4 is only if you need 4 CPU's
#$ -l h_vmem=4g <== maximum memory that can be used.
#$ -cwd
#$ -j Y
#$ -V
#$ -m be
#$ -M email@address.lumc

echo Start time : `date`
/bin/hostname
echo End time : `date`
```

```
username@shark:~/shark/trunk$ qsub run_my_first_job.sh
Your job 220282 ("my_first_job") has been submitted
```

```
vill@shark:~/shark/trunk$ qstat
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
 220282 0.60500 my_first_j vill         r     09/07/2011 09:26:55 all.q@kitefinshark.cluster.loc     4
```

Important Note : default memory of every job is set to 3GB
-l h_vmem=4G

```
vill@shark:~/shark/trunk$ qacct -j 220282
==============================================================
qname        all.q
hostname     kitefinshark.cluster.loc
group        NexGenSeq
owner        vill
project      NONE
department   defaultdepartment
jobname      my_first_job
jobnumber    220282
taskid       undefined
account      sge
priority     20
qsub_time    Wed Sep  7 09:26:53 2011
start_time   Wed Sep  7 09:26:54 2011
end_time     Wed Sep  7 09:27:15 2011
granted_pe   BWA
slots        4
failed       0
exit_status  0
ru_wallclock 21
ru_utime     0.040
ru_stime     0.010
ru_maxrss    1500
ru_ixrss     0
ru_ismrss    0
ru_idrss     0
ru_isrss     0
ru_minflt    2332
ru_majflt    0
ru_nswap     0
ru_inblock   16
ru_oublock   192
ru_msgsnd    0
ru_msgrcv    0
ru_nsignals  0
ru_nvcsw     148
ru_nivcsw    5
cpu          0.050
mem          0.000
io           0.000
iow          0.000
maxvmem      30.523M    <= note the maximum memory that has been used
arid         undefined
```

Important Note : default memory of every job is set to 3GB
Job has been submitted with 4GB times 4 slots = 16GB can be submitted with 15MB
-l h_vmem=15m

Configure a task dependency

```
#qsub test1.sh
#qsub -hold_jid (jobID for test1.sh) test2.sh
```

test2.sh will wait for test1.sh to complete

Run 3rd job that depends on test2.sh

```
#qsub -hold_jid (jobID for test2.sh) test3.sh
```

What can I run where ?

- HEAD Node
  - Small non memory/cpu intensive jobs
  - vi, vim, nano
  - scp, sftp
  - Compile small programs

- Graphical Login Node
  - Firefox, fslview, gedit, etc

- Execution Nodes
  - Large jobs
  - R(run only on here!)
  - Pipelines
  - Matlab

    Do not run Qlogin and go to lunch while this slot is doing nothing.

Evironment Modules

- The Environment Modules package provides for the dynamic modification of a user's environment via modulefiles.

```
# module list
Currently Loaded Modulefiles:
  1) OpenGridScheduler   2) History            3) R/3.0.2

#module avail
--------------------------------------------- /usr/local/Modules/versions --------------------
3.2.10
-------------------------------- /usr/local/Modules/3.2.10/modulefiles-------------------------------------
dot               java/Oracle-JRE-1.7 module-info        OpenGridScheduler   R/2.15.3          smrtanalysis
History             java/Oracle-JRE-1.8 modules            python/python-3.4.1 R/3.0.2           use.own
java/openjdk-7      module-git         null             qiime            R/3.1.1


#module load R/3.0.2
#module unload R/3.0.2
#module switch R/3.1.1
```

Summary

- **qsub** submit a job to the batch scheduler
- **qlogin** submit an interactive login session to Grid Engine
- **qstat** examine the job queue
- **qdel** delete a job from the queue
- **qacct** report and account for Grid Engine usage
- **unique-user-slots**
- /home Home dir do not store DATA here
- /data/... for storage of your data
- /usr/local/Genomes/ dir where ref. Files are located
- /bam-export dir for temp. bam files and share files.
- Do not abuse QLOGIN use qsub or qmake
- http://barmsijs.lumc.nl/