Short introduction to usage of batch cluster

1. Connection to our batch system

Basics

Our HPC cluster is Linux based (Scientific Linux). If you are not registered for it please send an emal to support@gwdg.de (subject: registration for HPC cluster). If you are asked to subscribe HPC-announcement list (https://listserv.gwdg.de/mailman/listinfo/hpc-announce) please do so!
Be sure to have enough disk quota for your work. You can check your quota by typing "Quota" (beginning with a capital letter!) at command-line. Please pay attention: Your quota is given in kB. If you need more disk space request for it by sending an email to support@gwdg.de (subject: Quota of Linux home directory).
Last but not least: You **need** some basics in Linux to work with a Linux system! If you haven't any experiences in Linux let me know, I will help you and give you a short introduction to Linux.

Frontends

Batch system can be connected by two submit hosts (= batch system frontends) called "gwdu101.gwdg.de" und "gwdu102.gwdg.de". Both are accessible via gwdu05.gwdg.de (or gwdu06.gwdg.de) only. That means connection to frotends maybe done by connecting to gwdu05 followed by command: **`ssh -Y gwdu101`**.
Connecting software for Linux and UNIX PCs (or Notebooks) is not necessary, Mac needs an XWindow terminalserver like XQuartz (http://xquartz.macosforge.org/landing), MSWindows an XWindow terminalserver like XWin32 (http://www.gwdg.de/index.php?id=1320).

2. Our batch system is controlled by the following commands:

bqueues *queue_name1 queue_name2 queue_name3 ...*

Example:

```
gwdu102:19 09:13:17 /home/uni05/rbohrer2 > bqueues fat fat-short fat-long
QUEUE_NAME      PRIO STATUS          MAX JL/U JL/P JL/H NJOBS  PEND   RUN  SUSP
fat              30  Open:Active       -    -    -    -  4217   841  3073     0
fat-short        40  Open:Active     128    -    -    -     0     0     0     0
fat-long         30  Open:Active     128    -    -    -   296   168   128     0
gwdu102:19 09:13:27 /home/uni05/rbohrer2 >
```

Command "bqueues" without any parameter shows all available queues at HPC cluster. The here listed queues are of especially interest for bioinformaticians because of there high memory and large number of cores.
The meaning of several colums are: "NJOBS" (total number of jobs), "PEND" (total number of pending jobs), "RUN" (total number of running jobs), "SUSP" (total number of suspended jobs)

bjobs

This command shows your jobs only and their batchjob IDs (this is **not** process-id of linux operating system!).

bkill *batchjob-ID*

This command must be used if you want to stop your job after submission to the batch system (regardless if the job is pending or running). The string "batchjob-ID" must be replaced by your real batchjob ID. You will get a message containing this job-id after submitting job to batch system or you ask for it by **`bjobs`** command.

bsub < *your_batch_submit_command_file*

This submits your job to the batch system. The string "your_batch_submit_command_file" must be replaced with the **filename** of your batch submit command (see next topic).

3. The batch submit command file (syntax):

```
#!/bin/sh
#BSUB -R scratch                                  → requests for slots connected to /scratch which is based upon a very fast Fraunhofer file system and suitable for shared data
#BSUB -R big                                      → requests for slots with the bigest number of cores (currently 64)
#BSUB -R 'span[hosts=1]'                          → limits the number of hosts for the requested cores to one host (= all cores must be of the same host/node)
#BSUB -q fat                                      → specifies requested queue [fat = max. 48 h walltime, fat-long = max. 120 h walltime, fat-short = max. 4 h walltime]
#BSUB -W hh:mm                                    → specifies requested walltime in "hh:mm"
#BSUB -n number_of_requested_cores
#BSUB -o absolute_path_to_batch_process_outfile   → this is a logfile of your batch process and contains important data (start time, end time, used host, used command etc.)
#BSUB -e absolute_path_to_batch_process_error_outfile   → this is a error logfile of your batch process and contains error output if an error occurred (otherwise it is a zero byte file)
your_program_file -parameter1 value_of_parameter1 -parameter2 value_of_parameter2 .....
```

Example:

```
#!/bin/sh
#BSUB -R scratch
#BSUB -R big
#BSUB -R 'span[hosts=1]'
#BSUB -q fat
#BSUB -W 02:10
#BSUB -n 6
#BSUB -o /usr/users/rbohrer2/test.seq-nr.out
#BSUB -e /usr/users/rbohrer2/test.seq-nr.err
blastx -db /usr/users/rbohrer2/BLAST_dbs/nr -out /usr/users/rbohrer2/test.seq-nr.blastx -num_threads 64 -query /usr/users/rbohrer2/test.seq
```

Please pay attention to the following:

a) Do not run any jobs without defining standard output and error output files, because these are very essential in error search if your job crashes.

b) Use for such a command file for bsub (see above) a linux editor like emacs, nano or joe. If you are using a MS Windows system and are not familiar with such Linux editors install a freeware editor for windows like "EditPad lite" (http://www.editpadlite.com) and save your file in UNIX ASCII format **before** you transfer it to your Linux homedirectory (e.g. using our SAMBA service: http://www.gwdg.de/index.php?id=663).

Furthermore it is possible to create the example file above (without using an editor!) from command-line as follows (but be sure to use apostrophe characters at the beginning and end of each echo command!):

```
echo '#!/bin/sh' > batch_cmd.file
echo '#BSUB -R scratch' >> batch_cmd.file
echo '#BSUB -R big' >> batch_cmd.file
echo "#BSUB -R 'span[hosts=1]'" >> batch_cmd.file
echo '#BSUB -q fat' >> batch_cmd.file
echo ' -W 02:10' >> batch_cmd.file
echo ' -n 6' >> batch_cmd.file
echo ' -o /usr/users/rbohrer2/test.seq-nr.out' >> batch_cmd.file
echo ' -e /usr/users/rbohrer2/test.seq-nr.err' >> batch_cmd.file
echo '/usr/product/bioinfo/ncbi/bin/blastx -db /usr/users/rbohrer2/BLAST_dbs/nr -out /usr/users/rbohrer2/test.seq-nr.blastx -num_threads 64 -query /usr/users/rbohrer2/test.seq' >> batch_cmd.file
```

c) Queue "fat" offers slots with 64 cores (of 256 GB in total) or 48 cores (of 128 GB RAM in total). With the option **`#BSUB -R big`** (or **`#BSUB -R np64`**) you request for the (newer) 64 cores nodes only. If you need the complete memory you had to request for 64 or 48 resp. cores - regardless if you need these cores.
That means: number of requested cores determines maximum of usable memory (e.g.)**:**
request of 4 cores of 64 core nodes (using additional parameter "**`-R big`**") results in (256 / 64 x 4 =) 16GB RAM
request of 4 cores of 48 core nodes results in (128 / 48 x 2,5 =) 10GB RAM
and in both cases you request for queue fat!

d) Please request for distinctly more than enough walltime because if you reach limit of your requested walltime your job will be **automatically killed**!
fat-long is of 120 hours, **fat** of 48 hours, and **fat-short** of 4 hours walltime maximum.

4. Launching of programs (***before*** submission to batch system!) (syntax examples):

```
module avail                          # shows all available programs
module load NCBI-BLAST                # loads module NCBI-BLAST (incl. batch_blast.script for intensive BLAST runs)
module load MIRA/3.9.18               # loads module version 3.9.18 of MIRA
module load OASES VELVET              # loads OASES and VELVET
module unload OASES                   # unloads OASES only
module purge                          # unloads all loaded modues
module help                           # help function of module
man module                            # manpage of module
```

5. More detailed information you will find in:

http://wiki.gwdg.de/index.php/Scientific_Compute_Cluster

http://wiki.gwdg.de/images/2/29/Parallelkurs.pdf (slides of our course for usage of HPC cluster)

http://wiki.gwdg.de/index.php/Running_Jobs or more detailed in http://wiki.gwdg.de/index.php/Running_Jobs_%28for_experienced_users%29

please subscribe to HPC maling list (https://listserv.gwdg.de/mailman/listinfo/hpc-announce) and use our course no.1391 "Using the GWDG Scientific Compute Cluster - An Introduction"!