

Kurze Anleitung zur Nutzung des Batch-Systems im HPC-Cluster

1. Verbindung zum Batch-System

Allgemeine Vorbemerkung

Unser HPC-Cluster ist Linux basiert. Wenn Sie noch keine Zulassung für den HPC-Cluster haben, senden Sie bitte eine entsprechende E-Mail an support@gwdg.de. Achten Sie bitte darauf immer ausreichend Plattenplatz für Ihre Arbeit zu haben. Wie hoch Ihre Disk-Quota ist, und wie viel Sie bereits davon verbraucht haben, erfahren Sie durch die Eingabe von "**Quota**" (erster Buchstabe ist groß!) auf der Kommandozeile. Falls Sie mehr Platz brauchen, schicken Sie eine entsprechende E-Mail an support@gwdg.de. Last but not least: Sie benötigen selbstverständlich ein Minimum an Linux-Kenntnissen, um mit dem HPC-Cluster arbeiten zu können. Wenn Sie noch keine Erfahrung mit Linux haben, melden Sie sich bei uns per E-Mail und wir helfen Ihnen weiter.

Die Frontend-Rechner

Es gibt 3 Frontend-Rechner von denen aus Sie Jobs zum Batch-System schicken können: "gwdu101.gwdg.de", "gwdu102.gwdg.de" und "gwdu103.gwdg.de". Alle drei sind nur vom GÖNET aus erreichbar. Wenn Sie sich außerhalb des GÖNETs befinden (z.B. zu Hause oder unterwegs) müssen Sie vor der Verbindung entweder einen VPN-Tunnel zur GWDG öffnen (https://info.gwdg.de/dokuwiki/doku.php?id=de:services:network_services:vpn:anyconnect), um eine IP-Adresse aus dem GÖNET zu erhalten, oder Sie verbinden sich mit dem weltweit erreichbaren Server **login.gwdg.de** [unter Linux mit: `ssh -Y username@login.gwdg.de`] und geben dann dort auf der Kommandozeile z.B. ein:

```
ssh -Y username@gwdu101.gwdg.de
```

Von einem Linux-PC **im GÖNET** können Sie den vorstehenden Befehl direkt ausführen, bei einem Mac oder MSWindows-Rechner brauchen Sie zunächst einen XWindow-Terminalserver [MacOS: XQuartz (<https://www.xquartz.org>); MSWindows: XWin32 (https://info.gwdg.de/dokuwiki/doku.php?id=de:services:general_services:software_and_license_management:xwin32lics), beide sind für Sie kostenlos erhältlich]. **Bitte führen Sie auf den Frontends keine Rechnungen aus** (höchstens um mal kurz zu testen, ob der Job korrekt anläuft).

2. Hier die wichtigsten Befehle für das Batchsystem (LSF):

```
bsub < jobscript
```

Damit schicken Sie den Inhalt von "jobscript" zum Batchsystem, wobei der String "jobscript" stellvertretend steht für den Pfad und den Namen des Skripts.

```
bjobs
```

Dieser Befehl gibt im Linux-Fenster alle Ihre Jobs, deren Status (laufend, wartend, beendet) und JobID aus. Letztere ist nicht die ProzessID, sondern wird vom LSF vergeben!

```
bkill jobid
```

Nur mit diesem Kommando können Sie einen Job abrechnen, unabhängig davon, ob er schon läuft oder noch wartet. Für den String "jobid" muss die tatsächliche JobID (s. vorherigen Befehl) eingesetzt werden. Die JobID wird Ihnen auch automatisch angezeigt, wenn Sie einen Job an das Batch-System abgeschickt haben

```
bqueues queue_name1 queue_name2 queue_name3 ...
```

Beispielausgabe:

```
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 >
```

Ohne weitere Parameter eingegeben zeigt Ihnen dieses Kommando für **alle** verfügbaren Batch-Queues an, wie stark die Auslastung ist. Die 3 hier ausgewählten Queues sind für Bioinformatiker von besonderem Interesse wegen des großen Hauptspeichers und der Anzahl der Cores. Die Bedeutung der Spalten ist wie folgt: "NJOBS" (alle Jobs), "PEND" (alle wartenden Jobs), "RUN" (alle laufenden Jobs), "SUSP" (alle angehaltenen Jobs)

3. Syntax für das Job-Skript:

```
#!/bin/bash
#BSUB -R keyword                → fordert bestimmte Ressourcen an
#BSUB -q queue_name             → benennt die Queue [fat = max. 48 h walltime, fat-long = max. 120 h walltime, fat-short = max. 4 h walltime]
#BSUB -W hh:mm                 → legt die max. Verweildauer beim Rechnen in der Queue fest ("hh:mm"), die natürlich nicht höher sein kann als die max. (s.o.)
#BSUB -n cores                 → fordert eine bestimmte Menge an Cores an
#BSUB -o absoluter_pfad_zum_batch_process_logfile → absoluter Pfad zum Logfile des Batch-Prozesses. Das Logfile enthält wichtige Daten über den Batch Prozess (u.a. Beginn, Ende, benutzer Rechenknoten) und die Ausgabe auf der Kommandozeile (falls vorhanden)
#BSUB -e absoluter_pfad_zum_batch_process_errorfile → absoluter Pfad zum Error-File des Batch-Prozesses. Falls Problem aufgetreten sind, sind diese dort vermerkt, ansonsten hat es 0 Bytes.
anwendungsprogramm -parameter1 wert_von_parameter1 -parameter2 wert_von_parameter2 .....
```

Diese letzte Zeile - die einzige ohne führendes #-Zeichen - enthält den eigentlichen Aufruf des Jobs. Wenn Sie ein bisschen Linux gelernt haben, dann können Sie hier auch ein Skript aufrufen. In dem haben Sie dann alle nur denkbaren Möglichkeiten den Job zu steuern bis hin zu einem kompletten mehrstufigen Arbeitsablauf (Pipeline).

Beispiel für ein Job-Skript: (Start des Programms blastx unter Verwendung der Datenbank NR und der Sequenzdatei "test.seq"):

```
#!/bin/bash
#BSUB -R scratch          → fordert Batch-Slots an in denen das Filesystem /scratch verfügbar ist (ein sehr schnelles BeeGFS-Filesystem zum temporären Speichern von Daten)
#BSUB -R 'span[hosts=1]'  → begrenzt die Anzahl der Rechenknoten auf 1, d.h. alle angeforderten Cores müssen auf demselben Rechenknoten liegen (notwendig, da kein echtes Parallelrechnen geplant ist)
#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/product/bioinfo/DATABASES/BLAST/NR -out /usr/users/rbohrer2/test.seq-nr.blastx -num_threads 6 -query /usr/users/rbohrer2/test.seq
```

Bitte die folgenden Hinweise besonders beachten:

- Starten Sie bitte keinen Job ohne Batch-Ausgabe-File und Batch-Error-File zu definieren**, denn darin stehen wichtige Daten für eine etwaige Fehleranalyse.
- Zum Editieren** Ihres Job-Skriptes (s. o. hellblau unterlegt) benutzen Sie bitte einen Editor unter Linux (z.B. emacs, nano oder joe). Für MSWindows-Systemen ist z.B. der freie Editor "EditPad lite" (<http://www.editpadlite.com>) gut zu verwenden. Speichern Sie mit diesem die Datei als **UNIX-ASCII-Format** (das ist nicht dasselbe wie das Microsoft ASCII Format!) **bevor** Sie die Datei auf Ihrem Linux Homedirectory speichern (z.B. unter Nutzung des SAMBA-Service: https://info.gwdg.de/dokuwiki/doku.php?id=en:services:server_services:compute_server:unix_linux_dialog_server).
- Wenn Sie den gesamten Hauptspeicher eines Rechenknotens benötigen, müssen Sie alle Cores diese Knoten anfordern, auch wenn sie die Cores nicht alle brauchen. **Das bedeutet: die Anzahl der angeforderten Cores bestimmt die Größe des zur Verfügung stehenden Hauptspeichers** (z.B.):
Wenn Sie 6 von 64 Cores eines Knoten anfordern (solche Knoten erhalten Sie mit dem Parameter "-R big") stehen Ihnen (256 / 64 x 6 =) 24 GB RAM zur Verfügung (falls der Knoten 256 GB RAM hat).
Wenn Sie 64 von 64 Cores eines Knoten anfordern (solche Knoten erhalten Sie mit dem Parameter "-R big") stehen Ihnen (256 / 64 x 64 =) 256 GB RAM zur Verfügung
Anmrk: Geben Sie einem Java-Programm intern (also über seine Javaaufruf-Parameter) niemals das komplette Memory Ihres Batch-Slots, da das Batch-System auch noch etwas Platz zum Rechnen benötigt, also im obigen Beispiel sollte Java nicht mehr als 250 GB RAM bekommen.
- Fordern Sie immer etwas großzügig Rechenzeit (Walltime) an, da am Ende der Walltime der Prozess **rücksichtslos abgebrochen** wird. Die unterschiedlich langen max. Rechenzeiten der Queues kann man an dem Suffix "-long" oder "-short" ablesen kann, z.B. **fat-long** (120 Std.), **fat** (48 Std.) und **fat-short** (4 Std.).

Weitere, detailliertere Informationen finden Sie hier:

https://info.gwdg.de/dokuwiki/doku.php?id=en:services:application_services:high_performance_computing:running_jobs
https://info.gwdg.de/dokuwiki/doku.php?id=en:services:application_services:high_performance_computing:running_jobs_for_experienced_users
https://info.gwdg.de/docs/doku.php?id=en:services:application_services:high_performance_computing:new_nodes

e) Tipp: Wenn Sie einen sehr großen Hauptspeicherbedarf haben (> 512 GB) können Sie dies auch direkt anfordern (müssen es also nicht über die Menge an Cores definieren):

```
#BSUB -q fat+          → Queue mit besonders großem Hauptspeicher
#BSUB -x
#BSUB -R "maxmem < 1600000 && maxmem > 600000"  → dies fordert Batch-Slots mit einem Hauptspeicher zwischen 600000 und 1600000 MB (damit keine Knoten mit nur 512 GB, aber alle mit 1,5 TB) an.
[Anmerk.: Im HPC-Cluster stehen z.Z. 5 Knoten mit 1,5 TB und 1 Knoten mit 2 TB zur Verfügung]
```

4. Start der Anwendungsprogramme (eine gute Übersicht über die Anwendungsprogramme der *Bioinformatik* findet sich unter: <https://gwdu100.gwdg.de>):

Vor jeder Ausführung von Anwendungsprogrammen müssen für diese immer erst das erforderliche Environment geladen werden (e.g. **module load OASES**)

(ausgenommen ist nur das Standard-R-Programm, das mit dem Betriebssystem mitgeliefert wird). Beim Abschicken eines Batchjobs von der Kommandozeile wird das aktuelle Environment des betreffenden Linux-Fensters in das Batch-System übernommen. Ist der entsprechende Modul-Befehl bereits ausgeführt worden, kann im Job-Skript das Anwendungsprogramm (letzte Zeile!) direkt benannt werden (andernfalls muss dort auf ein Skript verwiesen werden in dem dann der Modul-Befehl ausgeführt wird). Hier weitere Module-Parameter:

```
module avail          # zeigt alle verfügbaren Programme an
module load MIRA/3.9.18  # lädt das Environment der Version 3.9.18 von MIRA
module load OASES VELVET  # lädt die (jeweils neuesten) Environments von OASES und VELVET
module unload OASES     # entlädt das Environment von OASES
module purge          # entlädt alle Environments
module help           # ruft die Hilefunktion von module auf [Die Eingabe von man module listet die kompletten Hilfeseiten von "module" auf.]
```

5. Weitere und ausführlichere Informationen zur Nutzung des HPC-Clusters:

Übertragung großer Datenmengen (> 10 GB) in den HPC-Cluster (https://info.gwdg.de/dokuwiki/doku.php?id=en:services:application_services:high_performance_computing:transfer_data)

Übersicht über alle HPC-Knoten (https://info.gwdg.de/dokuwiki/doku.php?id=en:services:application_services:high_performance_computing:start)

Kurs zu Nutzung des HPC-Clusters (https://info.gwdg.de/dokuwiki/doku.php?id=en:services:application_services:high_performance_computing:courses)

Konfiguration der eigenen bash-Shell (https://info.gwdg.de/dokuwiki/doku.php?id=en:services:application_services:high_performance_computing:bashrc)