

## **Athene HPC-Cluster 2013/14 Uni-Regensburg**

[http://www.uni-regensburg.de/EDV/kurs\\_info/brf09510/hpc/hpc13.html](http://www.uni-regensburg.de/EDV/kurs_info/brf09510/hpc/hpc13.html)

[http://www.uni-regensburg.de/EDV/kurs\\_info/brf09510/hpc/hpc13.pdf](http://www.uni-regensburg.de/EDV/kurs_info/brf09510/hpc/hpc13.pdf)

[http://www.uni-regensburg.de/EDV/kurs\\_info/brf09510/hpc/hpc13.dvi](http://www.uni-regensburg.de/EDV/kurs_info/brf09510/hpc/hpc13.dvi)

9. Dezember 2014



## Schnellinformationen

### 1. What's New — Neuigkeiten

Der aktuelle Stand und die aktuellen Neuigkeiten findet man nach dem login auf der Athene im *message of the day*-Text und mit dem Kommando `wip`.

**12'2014:** Infotools

**12'2014:** 4 neue Knoten; QDR-IB-Switch (Ports können noch gemietet werden)

**8'2014:** Intel MKL 11 (siehe `/software/iNTEL/mkl`)

**7'2014:** Intel Cluster Studio XE 2013; Intel MKL 10.3.12.361 (systemweit)

**7'2014:** AMD OpenCL

**7'2014:** Matlab R2014a

**7'2014:** BUPC, PetSC, SlepC, Octave, Python

**4'2014:** Hardware-Erweiterung

**3'2014:** Umstellung auf Debian (AthOS - Wheezy) und openMPI abgeschlossen

**2014:** Neue Knoten 71 × IvyBridge

**2014:** Neue Graphikkarten 2 Tesla M2090

**2013:** PETSc und SLEPc

**2013:** Umstellung auf Debian und openMPI abgeschlossen (inklusive Athene2)

**2013:** Neue Knoten 4 × SandyBridge

**2013:** Neue Knoten 2 × Nehalem

**2013:** Neue Knoten 4 × SandyBridge

**10'2012:** Neue Kommandos: `wip`, `qparams common`

**10'2012:** Neue Graphikkarten Tesla M2090

**2012:** Neuinstallation Squeeze - AthOS (Wartungsvertragablauf)

**13.6.2011:** Neuer Knoten mit 2 Tesla M2070 GPGPU-Karten

**10.1.2011:** Neue Knoten 16 × Nehalem

**11.2.2010:** Neue Knoten 14 × Nehalem

**16.11.2009:** Chapel von Cray, eine PGAS-Sprache mit Ziel hoher Produktivität, funktioniert.

[http://www.uni-regensburg.de/EDV/kurs\\_info/brf09510/hpc/hpccpl.html](http://www.uni-regensburg.de/EDV/kurs_info/brf09510/hpc/hpccpl.html)

**10.11.2009:** UPC, eine PGAS-Sprache, funktioniert.

[http://www.uni-regensburg.de/EDV/kurs\\_info/brf09510/hpc/pgasupc.html](http://www.uni-regensburg.de/EDV/kurs_info/brf09510/hpc/pgasupc.html)

**30.10.2009:** Titanium, eine PGAS-Sprache, funktioniert.

[http://www.uni-regensburg.de/EDV/kurs\\_info/brf09510/hpc/pgastit.html](http://www.uni-regensburg.de/EDV/kurs_info/brf09510/hpc/pgastit.html)

**29.4.2009:** Einweihung **1.4.2009:** Inbetriebnahme **Februar 2009:** Testbetrieb **26.2.2009:** Abnahme

### 2. Schnelleinstieg

Zum Rechnen auf der Athene benötigt man

- einen Athene-Account
- den Login-Knoten Athene (Athene1)
- einen Compiler (FSF/Gnu, Intel, UPC)
- eine Parallel-Bibliothek (Open-MPI)
- weitere parallele Software (PETSc, SLEPc)
- indirekt: Torque-Batch-System zum Starten der Jobs auf den Knotenrechnern
- indirekt: Torque-Erweiterung (dyn. Knotenmanagement)

**2.1. Account.** Bitte stellen Sie einen Antrag auf Clusterbenutzung im Rechenzentrum. Ihr RZ-Account wird dann für die Athene übernommen.

**2.2. Login.** Athene (früher Athene1) ist der Loginknoten: hier müssen Sie sich mit ihrem normalen RZ-Account einloggen.

Beispiel mit ssh unter Linux (Rex1, Phy301...):

```
ssh bbbxxxx@athene (aus dem Intranet)
```

Eine direkte Einwahl aus dem Internet ist verboten. Studenten benutzen vorzugsweise die rex1, wissenschaftliche Mitarbeiter und Bedienstete die rex2, von da geht es dann weiter auf die Athene:

```
ssh bbbxxxx@rex1.uni-regensburg.de
ssh athene
```

Bildschirmprotokoll:

```
1 pc1011006189:~$ ssh bbbnnnnn@rex1.uni-regensburg.de
2 ...
3 Are you sure you want to continue connecting (yes/no)? yes
4 bbbnnnnn@rex1.uni-regensburg.de's password:
5 ...
6 rex1:~> ssh athene
7 ...
8 Are you sure you want to continue connecting (yes/no)? yes
9 ...
10 bbbnnnnn@athene's password:
11 Welcome to AthOS @ UR HPC-Cluster Athene
12
13 * NEW COMMANDs "athinv" - lists hardware available global or in a given
14 * queue,
15
16                               shows bookings (maintenance, courses, ...)
17 "jps" - print process snapshot of a job [+ shm segments]
18 "lqueues" - list enabled queues w/ access rights
19 "lpkgs" - list packages (matching a given pattern)
20 available
21
22                               @compute-nodes
23
24 * use "wip" to read actual installation work in progress news
25 * if you are new here, read the qsub/qstat and cognate man-pages;
26 reading man-pages is MANDATORY in unix like environments!
27 * use "pbstop" to get the actual node utilization (press "h" for help)
28 * use "showstart" to see when our scheduler assumes that there are enough
29 free resources to start your waiting job(s)
30 * use "qparams <queue>" to get the predefined and maximum possible resource
31 attributes for a given queue, "qstat -q" lists all available queues
32 * look in "/undelivered" if you are missing saved job stdout/stderr files
33 from your finished but likely broken job(s) - moved in here once a day
34 * if a job exits early, keep sure that it doesn't tried to use more
35 resources (number of processes/threads, memory per proc, ...) as requested
36 - if you can't find the reason for an odd behavior, mail all relevant
37 informations (jobnumber, jobscript, queue, etc.) to randy.rueckner@ur.de
38 * make sure, that you never run multiple io intensive jobs at the same time,
39 "serialize" them with "-W depend:afterok=$JOBID" or use the local /scratch
40 * athene1 is ONLY for editing/compiling sources, rearranging input data
41 and job submitting, everything running else can be killed at any time
42
43 > all your past jobs consumed 32 cpu days together...
44 > cpu time used/WASTED under AthOS-Wheezy so far: 42 hours / 2 hours
```

43

44 `bbnnnnn@athene1:~$`

([http://www.uni-regensburg.de/EDV/kurs\\_info/brf09510/hpc/ssh13.txt](http://www.uni-regensburg.de/EDV/kurs_info/brf09510/hpc/ssh13.txt))

Die Welcome-message enthält mit \* markierte statische Information und benutzerbezogene dynamische Information (>).

**2.3. Compiler und MPI-Library.** Als Compiler wird vorzugsweise FSF/Gnu oder Intel benutzt. Beide Compiler beherrschen openMP und coarray-Fortran direkt. UPC kann mit Berkeley-UPC übersetzt werden. Als MPI-Bibliothek wird openMPI verwendet. Bibliotheken sind einheitlich in `/usr/lib` installiert.

**2.4. Übersetzung.** Die Übersetzung erfolgt mit `mpicc` oder unter Angabe der benötigten Libraries:

```
mpicc -O3 hellompi.c -o hello Übersetzung mit Optimierung
mpicc --showme hellompi.c -o hello Ausgabe der Compiler-Flags
gcc -I /usr/lib/openmpi/include/ -O3 hellompi.c -L /usr/lib/openmpi/lib/ -lmpi -o hello
icc -I /usr/lib/openmpi/include/ -O3 hellompi.c -L /usr/lib/openmpi/lib/ -lmpi -o hello
```

**2.5. Start von HPC-Jobs.** Auf der Athene sind mehrere Queues eingerichtet. Nicht alle Queues sind für alle Programme oder alle Benutzer verfügbar. Die Queues erlauben verschiedene maximale Job-Limits.

```
qstat Information über die Queues
qparams Maximale Limits
lqueues Liste aktiver Queues mit Zugangsberechtigungen
qstat -Q Liste aller Queues
qparams express Limits in der express-Queue
qstat -u bbbnnnnn Jobs von User bbbnnnnn
```

Das übersetzte Programm wird als Batchjob zur parallelen Ausführung auf den Knotenrechnern in einer Queue mit Torque gestartet. Torque erzeugt Batch-Jobs oder interaktive Jobs (z.B. zum Debuggen). Torque ist aus dem älteren OpenPBS hervorgegangen. Deshalb taucht das Kürzel PBS in den Submit-Scripts auf. Der eigentlich Job-Scheduler heißt Maui.

Beschreibung:

```
http://docs.adaptivecomputing.com/index.htm
http://docs.adaptivecomputing.com/torque/4-2-8/help.htm
http://www.adaptivecomputing.com/resources/docs/torque/4-2-8/torqueAdminGuide-4.2.8.pdf
http://docs.adaptivecomputing.com/suite/8-0/basic/help.htm#topics/torque/0-intro/introduction.htm
http://www.adaptivecomputing.com/products/open-source/torque/
http://docs.adaptivecomputing.com/maui/index.php
http://www.adaptivecomputing.com/resources/docs/maui/pdf/mauiadmin.pdf
```

Der Batchjob wird in einem modifizierten Shell-Script, einem sog. PBS-Script aufgerufen. Es eignen sich alle Script-Prozessoren (bash, csh, sh,...).

```
1 #!/bin/bash
2 #PBS -l nodes=4:ppn=8
3 #PBS -l walltime=00:10:0
4 #PBS -l pmem=1g
5 #PBS -q express
6
7 #          cput=00:01:00 (hh:mm:ss) CPU-Zeit
8
9
10 mpiexec -n 32 /home/brf09510/svn/hpc/trunk/mssgpssg/mpi/hallo/hello
```

([http://www.uni-regensburg.de/EDV/kurs\\_info/brf09510/hpc/batch13.pbs](http://www.uni-regensburg.de/EDV/kurs_info/brf09510/hpc/batch13.pbs))

Mit der ersten Pseudokommentarzeile (Shebang `#!/bin/bash`) wird `bash` zur Ausführung gestartet. Die folgenden Pseudokommentarzeilen (`#PBS directive`) sind mit dem Schlüsselwort `PBS` als `PBS-Direktiven` gekennzeichnet.

Zeile 2 fordert 4 Rechenknoten (`nodes`) mit je 8 Kernen (`ppn - processors per node, cores`), also insgesamt 32 Rechenkerne an.

Zeile 3 stellt eine maximale Verweilzeit (`walltime - Wanduhr`) des Jobs von 10 Minuten ein. Verweilt der Job länger als die hier angegebene Zeit im System, wird er automatisch abgebrochen. Kleine Werte führen andererseits zu einem früheren Jobstart als größere. Wählen Sie also diesen Wert *so klein wie möglich, so groß wie nötig*, und mit einem moderaten Sicherheitszuschlag nach oben (Empfehlung: ca 25%). Diese Angabe wird vom Batchsystem verwendet, um den Zeitpunkt des Jobstarts zu bestimmen und um fehlerhafte Jobs nach einer Maximalzeit abzuberechnen.

Zeile 4 fordert Kerne mit mindestens 1 GB RAM an. Zu große Werte führen auch hier zu einer Verzögerung des Jobstarts, weil auf das Freiwerden geeigneter Kerne gewartet werden muss.

Zeile 5 bestimmt die Queue *express*.

Der folgende Kommentar würde als `PBS-Direktive` die Rechenzeit (`cput - akkumulierte CPU Time`) auf 1 Minute begrenzen. Dieser Wert kann angegeben werden, spielt jedoch auf der Athene beim Scheduling *keine* Rolle. Der entscheidende Wert ist `walltime`.

Der Rest der Datei besteht aus normalen Shellscript-Kommandos. Im Beispiel startet `mpiexec` das Programm `hallo` auf 32 Cores (die Datei `hallo` wurde wie in den Übersetzungen weiter oben erstellt).

Der Job wird mit dem Kommando

```
qsub batch.pbs
```

in die Queue gestellt und abgearbeitet, sobald Knoten frei sind und der Job unter allen mit ihm konkurrierenden die höchste Priorität erhalten hat. Während der Ausführung belegt der Job die zugewiesenen Knoten.

Auch im `qsub`-Kommando können Limit-Optionen angegeben werden. Im folgenden Beispiel wird mit der Option `-l (limit)` die maximale `walltime` begrenzt. Von mehreren konkurrierenden Angaben gilt zunächst die `qsub`-Angabe, dann die `PBS`-Angabe und zuletzt die Voreinstellung.

```
qsub -l walltime=2:0:0 batch.pbs (beim Start)
#PBS -l walltime=2:00:00 (im PBS-Script)
```

Mit dieser Beispiel-Option wird die maximale Verweilzeit (`walltime`) auf 2 Stunden begrenzt. Begrenzung der CPU-Zeit (`cput`) hat bei der Athene keine Wirkung.

Über den Status von Jobs, Queues und Server gibt das Kommando `qstat` Auskunft.

```
qstat
qstat job-id
qstat -f job-id
lqueues Queue-Übersicht
```

Versehentlich gestartete Jobs können mit `qdel` wieder gelöscht werden:

```
qdel job-id
qdel all
```

Nach dem normalen Ende eines Jobs werden die nach `stdout` und `stderr` geschriebenen Programmergebnisse in zwei Dateien geschrieben, deren Namen mit dem `PBS-Scriptnamen` beginnen und die Jobnummer enthalten. Im Beispiel sind das

```
1 brf09510@athene1:~$ ls -altr
2 total 324
3 ...
4 -rw----- 1 brf09510 rz      0 Aug  8 14:17 batch.pbs.e63404
5 -rw----- 1 brf09510 rz    1140 Aug  8 14:17 batch.pbs.o63404
6 drwxr-xr-x 2 brf09510 rz    4096 Aug  8 15:33 .
7 brf09510@athene1:~$
```

([http://www.uni-regensburg.de/EDV/kurs\\_info/brf09510/hpc/eofiles.txt](http://www.uni-regensburg.de/EDV/kurs_info/brf09510/hpc/eofiles.txt))

Ein interaktiver Job wird gestartet mit

```
qsub -I
```

Queue-Übersicht (Jul'2014):

**express** Testjobs; Programmentwicklung; kleinere Jobs

**merkl**

**ziegler**

**serial** Maximal 1 Knoten; MPI und openMP für alle Kerne eines Knotens erlaubt

**horinek**

**longs**

**spang**

**common** größere Jobs

**nvidia** Jobs für die nvidia-Graphikkarten

**striegnitz**

**condmat**

**dick**

**fabian**

**2.6. MKL.** Die Montage von Programmen mit MKL ist komplizierter und wird von Intel mit dem MKL-Link-Line-Adviser unterstützt:

<https://software.intel.com/en-us/articles/intel-mkl-link-line-advisor>

(JavaScript erforderlich!)

Nach Eingabe der gewünschten Parameter kann man per Copy-and-Paste die notwendigen Compiler-Optionen entnehmen.

### 3. Kleine Helfer

Auf der Athene stehen eine Reihe kleiner Tools bereit (Autor, H. Rückner):

**wip** Stand der Installation (work in progress)

**athinv queue** Hardware-Übersicht

**jps jobid** Job Process Snapshot

**lpkgs** Software-Übersicht (lokale Software auf den Knoten)

**lqueues** Queue-Übersicht (grün: Nutzung erlaubt)

**jobeff jobid** Effizienz laufender Jobs

**liar** Aktuelle verfügbare Knoten und erlaubte Ressourcen

**cusage** Ausnutzung der Queueresourcen

**showstart** Worst case Wartezeit





## Hardware

In der veralteten und zu groben Taxonomie nach Flynn ist die Athene als MIMD-Architektur aufgebaut. Die Struktur des gesamten Clusters ist eine Distributed Memory Architektur. Jeder Knoten hat seinen eigenen Speicher, die Kommunikation erfolgt über eines der beiden Netze (Infiniband, GigE). Jeder einzelne Knoten verfügt über zwei oder mehr Prozessoren mit je eigenem Speicher, der jedoch prinzipiell von beiden Prozessoren genutzt werden kann, also eine ccNUMA-Architektur. Jeder der beiden Prozessoren besteht aus vier oder mehr Rechnerkernen mit gemeinsamem Speicher, bildet also eine SMP-Architektur. Ähnliches gilt auch für die Erweiterungen.

### 1. Rechner

Bei der Erstinstantation war der Cluster fast homogen. Bei Erweiterungen kann diese Homogenität nicht erhalten bleiben. Auch die aktuelle Netzstruktur ist inhomogen geworden und macht paralleles Rechnen auf alten und neuen Knoten mit demselben Programm schwierig bis unmöglich.

#### 1.1. Übersicht.

(Erstellt am 14.8.2014 mit `athinv`)

Login Node (Athene 1)

Master Node (Athene 2)

File Server (Athene 3-4)

1a: AMD K10:

2: K10-2354 ( 8 cores), 16149 MiB RAM, 130 GB local scratch

6: K10-2354 ( 8 cores), 16149 MiB RAM, 457 GB local scratch

1: K10-2354 ( 8 cores), 16149 MiB RAM, 937 GB local scratch

1b: AMD K10:

3: K10-2354 ( 8 cores), 32311 MiB RAM, 457 GB local scratch

1c: AMD K10:

120: K10-2354 ( 8 cores), 16149 MiB RAM, 216 GB local scratch, add-ons: ddrIB

1: K10-2354 ( 8 cores), 16149 MiB RAM, 457 GB local scratch, add-ons: ddrIB

1d: AMD K10:

42: K10-2354 ( 8 cores), 32311 MiB RAM, 457 GB local scratch, add-ons: ddrIB

2: AMD K10 (Condmatt)

13: K10-2354 ( 8 cores), 16149 MiB RAM, 457 GB local scratch, add-ons: ddrIB

2: Nehalem X

1: X5650 (12 cores), 24150 MiB RAM, 216 GB local scratch

3: Nehalem (condmat)

16: X5650 (12 cores), 24157 MiB RAM, 216 GB local scratch

4: Nehalem (condmat)

4: X5650 (12 cores), 24157 MiB RAM, 216 GB local scratch

5: Nehalem (Dick)

2: X5650 (12 cores), 24150 MiB RAM, 216 GB local scratch

6: Sandybridge (Condmatt)

1: X5650 (12 cores), 24157 MiB RAM, 216 GB local scratch

3: X5650 (12 cores), 24157 MiB RAM, 457 GB local scratch

Nvidia

1: X5660 (12 cores), 24157 MiB RAM, 457 GB local scratch, add-ons: qdrIB,Tesla-M2070,Tesla-M2070

1: X5660 (12 cores), 24157 MiB RAM, 457 GB local scratch, add-ons: qdrIB,Tesla-M2090,Tesla-M2090

5: E5530 ( 8 cores), 24155 MiB RAM, 216 GB local scratch

8: E5530 ( 8 cores), 24155 MiB RAM, 457 GB local scratch

6: Ivybridge

1: E5-2430 (12 cores), 32166 MiB RAM, 456 GB local scratch  
 4: E5-2650 (16 cores), 64505 MiB RAM, 457 GB local scratch  
 4: E5-2650v2 (16 cores), 64506 MiB RAM, 937 GB local scratch, add-ons: qdrIB  
 26: E5-2650v2 (16 cores), 64506 MiB RAM, 937 GB local scratch  
 28: E5-2650v2 (16 cores), 64506 MiB RAM, 937 GB local scratch, add-ons: qdrIB  
 1: E5-2650v2 (16 cores), 64518 MiB RAM, 264 GB local scratch, add-ons: Phi-5110P  
 2: E5-2650v2 (16 cores), 129146 MiB RAM, 553 GB local scratch  
 4: E5-2650v2 (16 cores), 129146 MiB RAM, 937 GB local scratch  
 3: E5-2650v2 (16 cores), 258426 MiB RAM, 937 GB local scratch, add-ons: qdrIB

Zusammen 302 Knoten, 3040 Cores

Gesamtsspeicher: 9865 GiB (Damit kann grob im gesamten Cluster eine dichte  $n \times n$ -Matrix,  $n = 1110000$  gespeichert werden. Die Größenordnung der Rechenzeit für eine Gausselimination bei dieser Matrix ist bei RPeak = 40 Tflops rechnerisch etwa  $t = 4.5h$ .)

## 2. Netz

Gigabit Ethernet (Broadcom Corporation NetXtreme Gigabit Ethernet PCI Express (rev 21))

Infiniband (DDR 4x, 20 GBit/s, InfiniBand: Mellanox Technologies MT25204 [InfiniHost III Lx HCA] (rev 20))

Zwei unabhängige Netze, die beide für Message Passing verwendet werden können. Das Ethernet wird eher für Verwaltungszwecke, Infiniband eher für Message Passing eingesetzt. Message Passing mit Ethernet bremst Programme um einen Faktor von etwa 8.

Infiniband (QDR).

## 3. Dateisystem

### 3.1. Dateiserver.

Athene3

Athene4

Zwei NFS-Server für die Dateisysteme  $\$HOME$  (quotiert, 3GiB/15000 files), `/scratch` und `/data`.

Derzeit existiert kein paralleles Filesystem.

**3.2. Lokale Filesysteme: NFS.** `/tmp` kann auch für lokale temporäre Daten verwendet werden; ist bei verteilter Anwendung mit vielen Knoten unter Umständen schneller als `/scratch`.

### 3.3. Umgang mit den Scratch-Verzeichnissen.

- Lokales Scratch wird nach jedem Jobende gelöscht. Benutzung unter `/scratch/${PBS_JOBID}`
- NFS-Scratch1 und -Scratch2: Hier existiert pro Benutzer ein Unterverzeichnis. Bei 80% Füllstand werden älteste Dateien automatisch gelöscht, bis ein Füllstand von 60% erreicht wird.

Achtung: Es ist unfair und wird überwacht, ob Dateien mit `touch` aufgefrischt werden.

## KAPITEL 3

# System

### 1. Betriebssystem

Linux version 3.2.0-4-amd64, Debian GNU/Linux 7 (wheezy)

### 2. Torque und PBS

#### 2.1. Übersicht. siehe man pbs

Kommando	Beispiel	Script
qalter	Ändern der Jobattribute	
qdel	Beenden (Abwürgen) eines Jobs	

#### 2.2. qsub. qsub jobScript options

Option	Beispiel	Script
-l resource-list	-l walltime=30.pmem=1gb	#PBS -q express
-q queue	-q express	
-I Interaktive		

#### 2.3. Ressourcen. siehe man pbs\_resources

Ressource	Wert	Bedeutung
arch		
cput	[[hh:]mm:]ss	Maximale CPU-Zeit pro core des Jobs (ohne Wirkung)
file		
host		
nice		
nodes	nodes=4	Anzahl der Knoten
nodes:ppn	nodes=4:ppn=2	Anzahl der Knoten und Kerne pro Knoten
other		
pccput		
pmem	pmem=1gb	Einheiten: b, kb, mb, gb
pvmem		
software		
vmem		
walltime	[[hh:]mm:]ss[.ms]	Maximale Verweilzeit des Jobs

#### 2.4. qstat. qstat jobid queue options

Option	Wirkung	Beispiel
-f	full	
-n	nodes	
-u bbbnnnnn	user filter	

#### 2.5. qdel. qdel jobid

## 2.6. PBS Umgebungsvariable.

Die folgenden Umgebungsvariable werden von der Umgebung des qsub-Kommandos weitergereicht:

Variable	Wert (Bsp)	Beschreibung
PBS_ENVIRONMENT	PBS_BATCH	set to PBS_BATCH to indicate that the job is a batch job; otherwise set to PBS_INTERACTIVE to indicate that the job is a PBS interactive job
PBS_JOBID	165139.Athene2	the job identifier assigned to the job by the batch system
PBS_JOBNAME	batch.pbs	the job name supplied by the user
PBS_NODEFILE	/var/spool/torque/aux//165139.Athene2	the name of the file that contains the list of the nodes assigned to the job
PBS_QUEUE	common	the name of the queue from which the job is executed
PBS_TMPDIR		

Die Umgebungsvariable X aus der Umgebung des qsub-Kommandos wird als Variable PBS\_O\_X weitergereicht.

Variable	Wert (Bsp)	Beschreibung
PBS_O_HOME	/home/brf09510	HOME variable in the evvironm. in which qsub was executed
PBS_O_LANG	en_US.UTF-8	LANG variable in the evvironm. in which qsub was executed
PBS_O_LOGNAME	brf09510	LOGNAME variable in the evvironm. in which qsub was executed
PBS_O_PATH	/usr/bin:/bin:/	PATH variable in the evvironm. in which qsub was executed
PBS_O_MAIL	/var/mail/brf09510	MAIL variable in the evvironm. in which qsub was executed
PBS_O_SHELL	/bin/tcsh	SHELL variable in the evvironm. in which qsub was executed
PBS_O_TZ		TZ variable in the evvironm. in which qsub was executed
PBS_O_HOST	Athene1.uni-regensburg.de	the name of the host upon which the qsub command is running.
PBS_SERVER	Athene1.uni-regensburg.de	the hostname of the pbs_server which qsub submits the job to.
PBS_O_QUEUE	batch	the name of the original queue to which the job was submitted.
PBS_ARRAYID		each member of a job array is assigned a unique identifier (see -t)
PBS_O_WORKDIR	/home/brf09510	the abs. path of the current working dir. of the qsub command

Zusätzlich werden die folgenden Umgebungsvariable definiert:

Variable	Wert (Bsp)	Beschreibung
PBS_O_HOST	Athene1.uni-regensburg.de	the name of the host upon which the qsub command is running
PBS_O_QUEUE	batch	the name of the original queue to which the job was submitted
PBS_O_WORKDIR	/home/brf09510	the absolute path of the current working directory of the qsub command

## Inhalt



## Inhaltsverzeichnis

Kapitel 1. Schnellinformationen	3
1. What's New — Neuigkeiten	3
2. Schnelleinstieg	3
3. Kleine Helfer	7
Kapitel 2. Hardware	9
1. Rechner	9
2. Netz	10
3. Dateisystem	10
Kapitel 3. System	11
1. Betriebssystem	11
2. Torque und PBS	11
Inhalt	13

TTH-Seite:

<http://hutchinson.belmont.ma.us/tth/>