01010011 Scientific
01000011 Computing
01000111 Group

Leibniz Universität Hannover

Handbuch zum Clustersystem

Leibniz Universität IT Services
Scientific Computing Group

3. August 2018

Inhaltsverzeichnis

Vorwort

Dieses Handbuch ist als Hilfsmittel zur Arbeit mit dem Clustersystem der Leibniz Universität Hannover gedacht. Um die Arbeit mit dem Werkzeug Clustersystem zu erleichtern, haben wir die wichtigsten Themen in diesem Kompendium zusammengestellt. Sollten im Laufe der Lektüre oder darüber hinaus Fragen entstehen, stehen wir unter cluster-help@luis.uni-hannover.de zur Verfügung. Wer überhaupt keine Zeit für jegliche Lektüre hat sollte sicherstellen, für die Cluster-News Mailingliste eingetragen zu sein.

Euer Cluster-Team

Ich bin neu hier – Schnelleinstieg

Wichtig: Wer absolut keine Zeit hat, sollte wenigstens diese eine Seite lesen. Bitte stellen Sie sicher, dass die folgenden Punkte erfüllt sind.

- Haben Sie eine Benachrichtigung bekommen, dass Sie in der Cluster-News Mailing Liste aufgenommen sind? Über diese Liste werden wichtige Ankündigungen, z.B. Wartungszeiten, bekannt gemacht.
- Haben Sie Ihr Passwort geändert? Ihr Passwort ändern Sie mit dem Befehl `passwd`.
- Begreifen Sie das Clustersystem bitte als Werkzeug, welches Ihre Forschung erleichtert. Ein Werkzeug zu beherrschen benötigt Zeit. Überlegen Sie sich, ob Sie an einem Einführungsvortrag teilnehmen möchten und dieses Handbuch lesen.

Über das Clustersystem

Um den Grundbedarf der Universität an Rechenkapazität, CPU-Leistung und Arbeitsspeicher, abzudecken, betreibt das LUIS im Service *Scientific Computing* ein Clustersystem. Dieses Clustersystem steht allen Mitarbeiterinnen und Mitarbeitern der Leibniz Universität Hannover für ihre Forschung kostenfrei zur Verfügung.

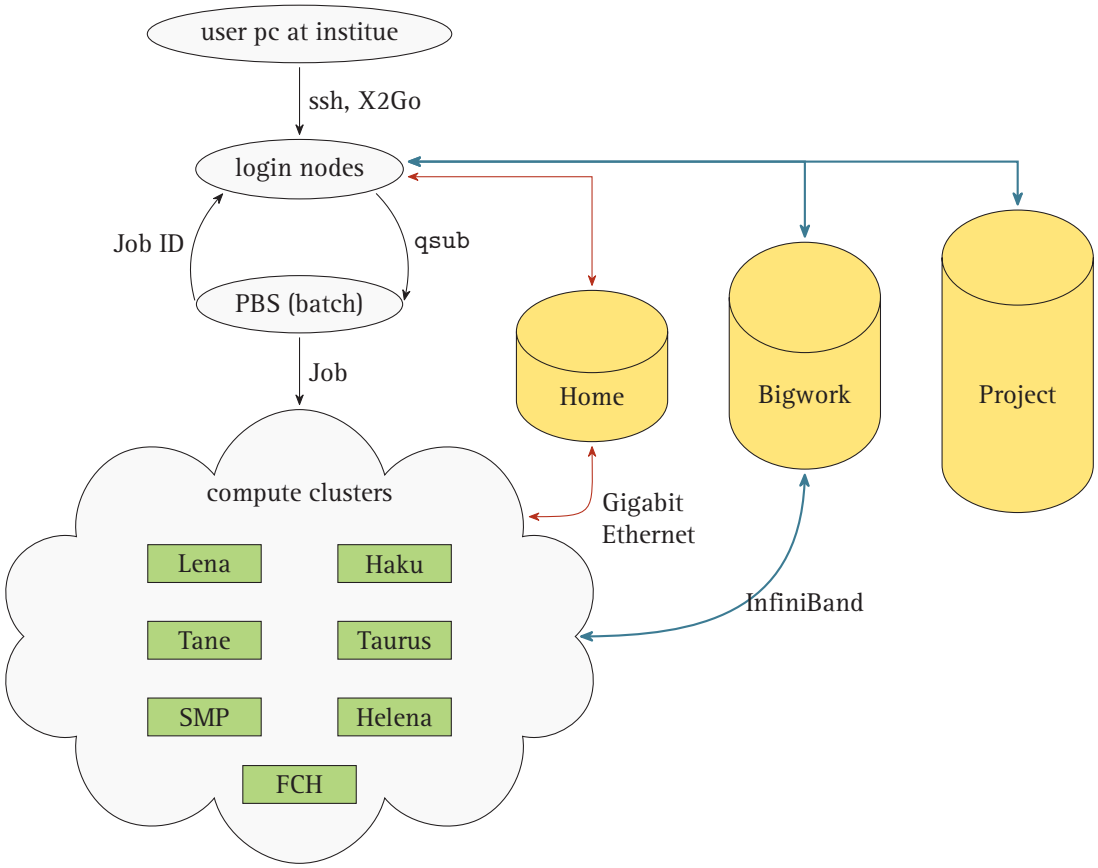

Abbildung 2.1: Aufbau des Clustersystems mit seinen Einzelkomponenten

Die Rechenressourcen des Clustersystems sind größtenteils als DFG Forschungsgroßgerät beschafft worden. Somit gelten bei Benutzung des Clustersystems die Regeln[1] der DFG für solche Forschungsgroßgeräte. Die Verantwortung zur Einhaltung obliegt den jeweiligen Leitenden eines EDV-Projektes, unter welchem das Clustersystem genutzt wird.

2.1 Zugang beantragen

Der Rahmen, in dem Sie Ihre Arbeiten auf dem Clustersystem durchführen, ist ein EDV-Projekt. Wahrscheinlich gibt es an Ihrem Institut bereits einen Projektleiter eines EDV-Projekts. Diese Person kann für Sie eine Benutzerkennung für das Clustersystem in BIAS[2] erstellen. Sollten Sie ein neues EDV-Projekt eröffnen wollen, so verwenden Sie bitte das Formular ORG.BEN 4[2].

Wichtig: Bitte lediglich eine Nutzerkennung pro Person anlegen.

[1] www.dfg.de
[2] Die Benutzerverwaltung BIAS ist nicht Teil des Service *Scientific Computing* und somit nicht Teil des Clustersystems

2.2 Wozu das Clustersystem genutzt werden darf

Teile des Clustersystems sind als DFG Forschungsgroßgerät beschafft worden, wodurch die Regeln für solche DFG Großgeräte für das gesamte Clustersystem gelten, da es ausschließlich im Verbund aller Komponenten genutzt werden kann. Des Weiteren sind die zur Verfügung stehenden Softwarelizenzen für Lehre und Forschung gültig. Das Clustersystem darf entsprechend lediglich für Zwecke der Forschung und Lehre verwendet werden.

2.3 Vorhandene Rechenressourcen

Zurzeit besteht das Clustersystems aus folgenden Teilclustern:

- 4 Cluster für MPI Berechnungen, die viele CPUs benötigen
  - **Lena**: 80 Knoten; jeweils 16 Kerne @ 2.4 GHz; 64 GB RAM
  - **Haku**: 20 Knoten; jeweils 16 Kerne @ 2.10 GHz; 64 GB RAM
  - **Tane**: 96 Knoten; jeweils 12 Kerne @ 2.9 GHz; 48 GB RAM
  - **Taurus**: 54 Knoten; jeweils 12 Kerne @ 2.66 GHz; 48 GB RAM
- 22 SMP Rechner für Berechnungen, die viel RAM benötigen
  - **SMP**: 4 Knoten; jeweils 32 Kerne @ 2.5 GHz; 256 GB RAM
  - **SMP**: 9 Knoten; jeweils 32 Kerne @ 2.0 GHz; 256 GB RAM
  - **SMP**: 9 Knoten; jeweils 24 Kerne @ 2.0 GHz; 256 GB RAM
- 3 Rechner mit jeweils 1 TB RAM
  - **Helena**: 3 Knoten; jeweils 32 Kerne @ 2.13 GHz; 1028 GB RAM
- Forschungscluster-Housing
  - **FCH**: 41 Rechner in verschiedenen Konfigurationen

Wichtig: Die einzelnen Teilcluster haben unterschiedlich lange Wartungsverträge. Wer für die nächsten Jahre eine garantierte, absolut identische Hardwarebasis benötigt, wählt Lena.

2.4 Forschungscluster-Housing

Im Service Forschungscluster-Housing (FCH) können Institute eigene Hardware ins Clustersystem integrieren. Diese Hardware kann tagsüber, von acht Uhr morgens bis acht Uhr abends, ausschließlich vom jeweiligen Institut genutzt werden. In den Nachtstunden und am Wochenende steht die Hardware der gesamten Nutzerschaft zur Verfügung, sofern Ressourcen frei sind.

Wenn Sie während dieser Zeit auf Maschinen rechnen, welche vom Namen her nicht in das Schema unsere großen Teilcluster passen, so handelt es sich vermutlich um Forschungscluster-Housing Teilnehmer. Bei Interesse zur Teilnahme an diesem Service mit Ihrer eigenen Hardware, beraten wir Sie gerne.

2.5 Kontakt & Hilfe

Hilfe bei Fragen bezüglich des Clustersystems kann bei cluster-help@luis.uni-hannover.de angefordert werden.

Verbinden mit dem Clustersystem & Dateitransfer

Die folgenden Adressen sollten für Verbindungen zum Clustersystem verwendet werden. Unter:

login.cluster.uni-hannover.de erreicht man die Login-Knoten, um Rechenjobs abzuschicken.

transfer.cluster.uni-hannover.de erreicht man den Transfer-Knoten, für Datentransfer mit dem Clustersystem.

Wichtig: Nach 30 Minuten werden Prozesse auf den Login-Knoten abgebrochen.

Aus diesem Grund brechen Dateitransfers auf den Login-Knoten ab, wenn das Limit erreicht ist. Auf der anderen Seite ist die Laufzeit für Prozesse auf dem Transfer-Knoten unbegrenzt. Es nicht jedoch nicht möglich vom Transfer-Knoten aus Rechenjobs abzuschicken.

3.1 Von Linux und Mac OS aus verbinden

Um sich mit dem Clustersystem von Linux oder Mac OS aus zu verbinden ist ein ssh Client nötig, der bei den allermeisten Distributionen zur Standardausstattung gehört. Das folgende Kommando baut eine Verbindung mit dem Clustersystem auf, username muss durch den eigenen Nutzernamen ersetzen.

```
ssh username@login.cluster.uni-hannover.de
```

Sollen grafische Programme auf dem Clustersystem verwendet werden, so muss die Option `-X`, welche X11-Weiterleitung einschaltet, verwendet werden.

```
ssh -X username@login.cluster.uni-hannover.de
```

3.2 Dateitransfer von Linux und Mac OS aus

Es existiert eine Maschine, welche für Dateitransfer mit dem Clustersystem bestimmt ist. Wann immer Daten mit dem Clustersystem ausgetauscht werden, sollte folgende Adresse verwendet werden:

`transfer.cluster.uni-hannover.de`

Für den Dateifransfer mit dem Clustersystem stehen verschiedene Programme zur Verfügung. Für einzelne Datein kann `scp` verwendet werden. Wir empfehlen den Einsatz von `rsync`. Außerdem kann FileZilla genutzt werden, wenn ein grafisches Werkzeug bevorzugt wird. Informationen zur Einrichtung von FileZilla sind in Abschnitt 3.4 zu finden.

Wichtig: Verwenden Sie den dedizierten Transfer-Knoten für Dateitransfers, da prozesse auf den Login-Knoten nach 30 Minuten abgebrochen werden.

3.3 Von Windows aus verbinden

Um sich mit dem Clustersystem von Windows aus zu verbinden ist zusätzliche Software nötig. Dadurch ist es möglich, grafische Anwendungen auf Login-Knoten oder Compute-Knoten von Ihrem PC aus zu nutzen. Voraussetzung dafür ist ein sogenannter X-Window-Client. Im Folgenden wird die Installation und Nutzung des Clients `X2Go` unter Windows 7 beschrieben. Dieser Anleitung liegt der X2Go-Client in Version `4.0.2.1+hotfix1` zu Grunde, welchen man über die folgende URL beziehen kann. Nach dem Download installiert man das Programm in einen beliebigen Ordner.

3.3.1 X2Go für den Login auf dem Clustersystem konfigurieren

Nach dem Start des X2Go-Client, entweder über eine Verknüpfung auf dem Desktop oder über das Startmenü, erscheint ein Konfigurationsdialog. In diesem Dialog kann der Sitzungsname vergeben werden und es müssen die folgenden vier Einträge gesetzt werden.

1. Host: login.cluster.uni-hannover.de
2. Login: Ihr Benutzername
3. SSH-Port: 22
4. Sitzungsart: XFCE

Der ausgefüllte Konfigurationsdialog ist in Abbildung 3.1 dargestellt. Die Einträge in den roten Kästen müssen gesetzt werden. Anschließend kann der Konfigurationsdialog mit einem Klick auf **OK** verlassen werden.

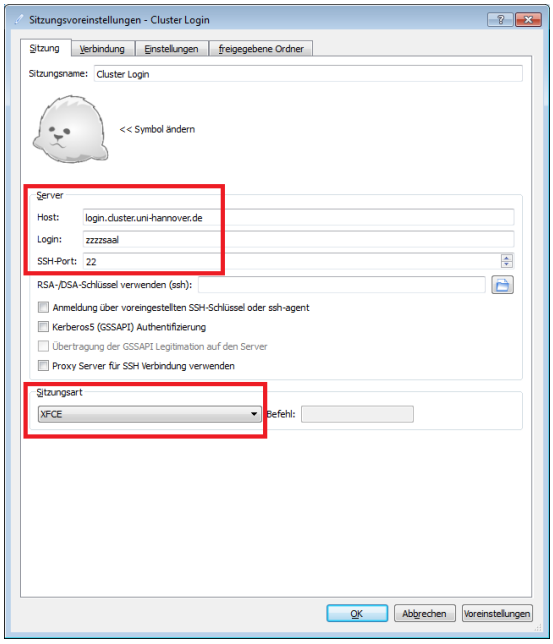

Abbildung 3.1: X2Go Konfigurationsdialog, die Einträge in den roten Kästen müssen gesetzt werden.

Auf der rechten Seite des Hauptfensters steht nun die neu erstellte Verbindung zur Verfügung, siehe Abbildung 3.2. Man startet diese mit einem Klick auf den Sitzungsnamen (rechts oben in Abbildung 3.2) oder durch Eingabe des Namens im Feld **Sitzung** (mittig in Abbildung 3.2).

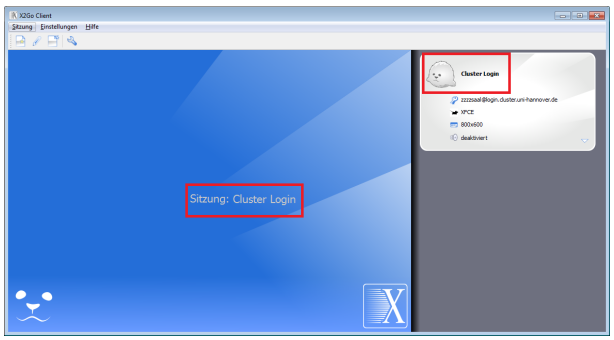

Abbildung 3.2: Verbindung zum Clustersystem mit einem Klick oder durch Eingabe des Sitzungsnamens öffnen.

Wenn zum ersten Mal eine Verbindung zu den Login-Knoten aufgebaut wird, ist der Host-Key der Login-Knoten noch unbekannt. Es erscheint ein Hinweisdialog, in welchem mit **Ja** der Host-Key akzeptiert werden kann, siehe Abbildung 3.3.

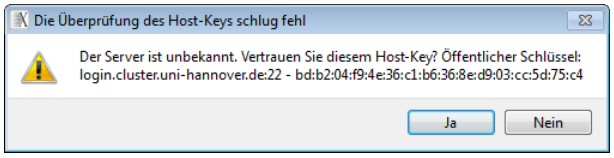

Abbildung 3.3: Dialog zur Überprüfung des Host-Key

Nach erfolgreichem Verbindungsaufbau steht ein XFCE Desktop zur Verfügung, wie in Abbildung 3.4 dargestellt.

Abbildung 3.4: XFCE Desktop mit Menü unten links

Über das *Applications Menu* unten links kann z.B. ein Terminal-Fenster geöffnet und Module geladen, `showq` benutzt oder Jobs in die Queue absetzt werden. Ebenfalls können Editoren geöffnet werden, um Batchskripte zu schreiben. Insbesondere interaktive Jobs mit grafischen Fenstern sind möglich. Wenn die Sitzung beendet werden soll, geht dies über den untersten Eintrag im *Applications Menu* oder über das grüne Männchen ganz rechts in der XFCE-Leiste am unteren Bildschirmrand.

3.4 Dateitransfer unter Windows mit FileZilla

Für den Dateitransfer unter Windows kann z. B. der FileZilla Client verwendet werden. Dieser Anleitung liegt der FileZilla Client in Version `3.14.1_win64` zu Grunde, welcher über diese URL bezogen werden kann. Nach dem Download kann das Programm in einen beliebigen Ordner installiert werden.

Nach der Installation kann im Servermanager ein neuer Server angelegt werden, zu dem sich anschließend eine Verbindung aufbauen lässt. Die folgenden Einstellungen müssen vorgenommen werden, siehe auch rote Kästen in Abbildung 3.5.

- Server: transfer.cluster.uni-hannover.de
- Protokoll: SFTP - SSH File Transfer Protocol
- Verbindungsart: Nach Passwort fragen

- Benutzer: Ihr Benutzername

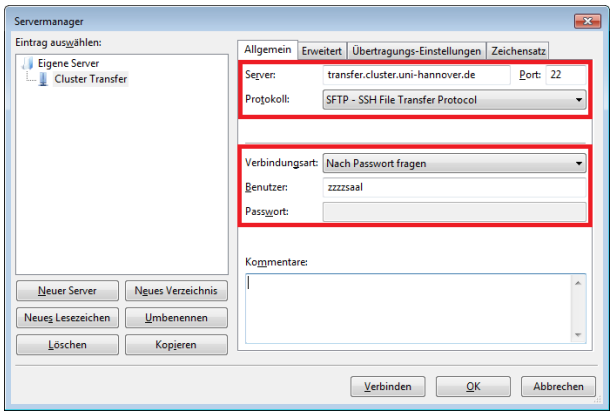

Abbildung 3.5: Servermanager allgemeine Einstellungen

Des Weiteren ist es möglich, sich direkt nach $BIGWORK verbinden zu lassen. Standardmäßig wird eine Verbindung zu $HOME aufgebaut. Um diese Option zu konfigurieren, muss dies im Reiter *Erweitert* unter *Standard-Verzeichnis auf Server* vorgenommen werden, siehe Abbildung 3.6.

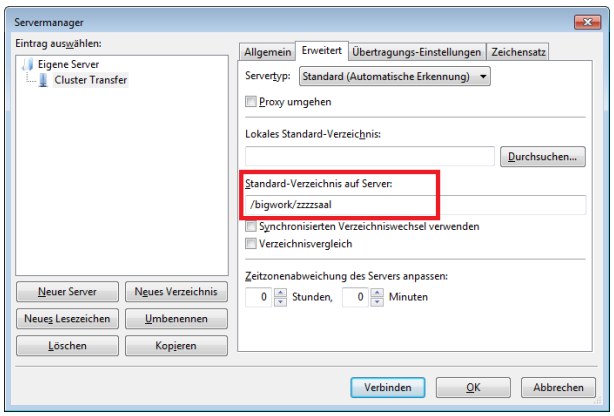

Abbildung 3.6: Servermanager erweiterte Einstellungen

Bei der ersten Verbindung zum Transfer-Knoten wird der Host-Key überprüft, siehe Abbildung 3.7.

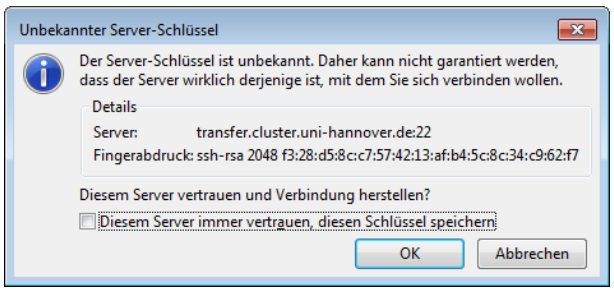

Abbildung 3.7: Überprüfung des Host-Key bei erstmaliger Anmeldung

Nach erfolgreichem Verbindungsaufbau, siehe Abbildung 3.8, kann man Dateien mit dem Clustersystem austauschen.

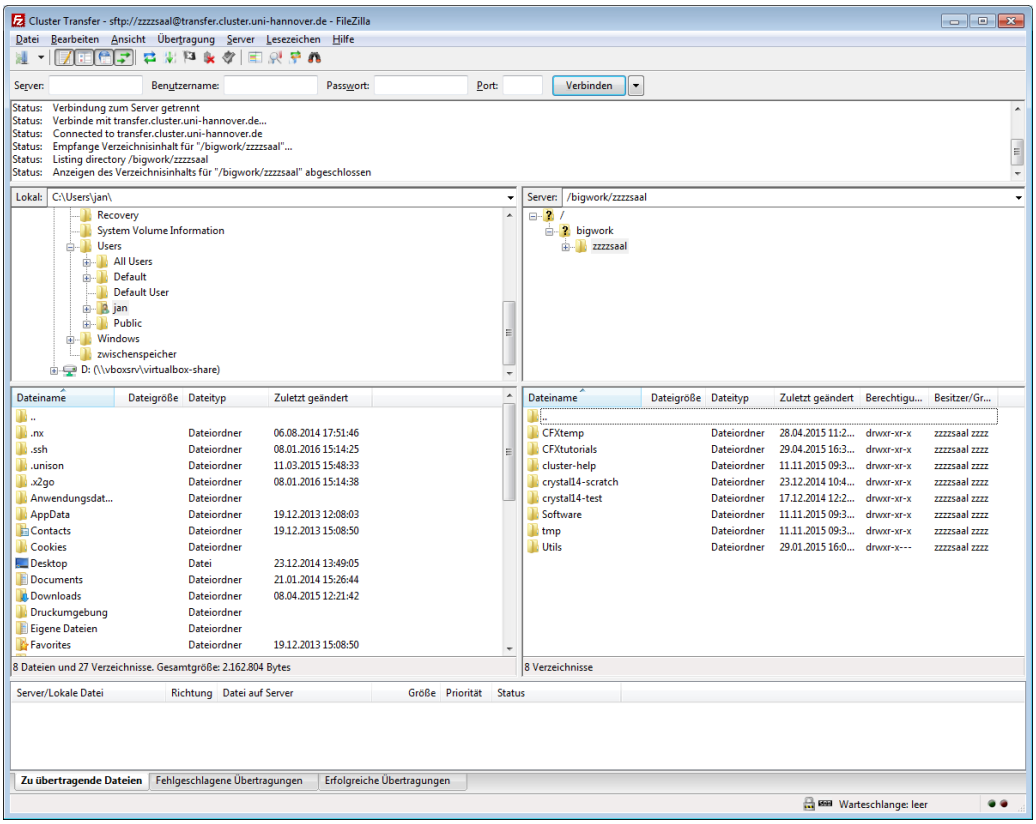

Abbildung 3.8: Verbindung zum Transfer-Knoten mit FileZilla hergestellt

3.5 Verbinden von außerhalb des Netzes der Universität

Von außerhalb des Netzes der Universität können keine Verbindungen zum Clustersystem aufgebaut werden. Zuerst müssen Sie z.B. per VPN eine Verbindung in das Netz der Universität aufbauen. Einen VPN Service stellen die Kollegen des Netz Teams bereit.

LUIS VPN Service

Nachdem Sie im Netz der Universität eingewählt sind, können Sie sich wie gewohnt mit dem Clustersystem verbinden.

Wichtig: Die Geschwindigkeit von Verbindungen von außerhalb ist meist deutlich langsamer, als von Ihrem Arbeitsplatz aus. Dadurch wird vor allem bei Anwendungen, welche eine graphische Benutzeroberfläche bereit stellen, die Arbeit erschwert. Verhalten sich Fenster von graphischen Benutzeroberflächen träge, so ist dies technisch bedingt und es liegt kein Fehler des Clustersystems vor.

Dateisysteme

Im Clustersystem stellen zwei Speichersysteme global, sprich auf jedem Knoten, zwei Dateisysteme zur Verfügung. Diese sind in Abbildung 4.1 dargestellt.

$HOME Ihr home-Verzeichnis. Hier steht eine vergleichsweise geringe Menge Speicherplatz zur Verfügung, dafür wird ein tägliches Backup dieser Verzeichnisse angefertigt. Demzufolge sollten hier lediglich wichtigste Dateien gespeichert werden. $HOME ist per Gigabit Ethernet, und somit im Vergleich zu $BIGWORK vergleichsweise langsam, angebunden.

$BIGWORK Ihr bigwork-Verzeichnis. Hier steht eine vergleichsweise große Menge Speicherplatz zur Verfügung, es gibt kein Backup. Dieses Verzeichnis ist als Arbeitsverzeichnis gedacht. Rechnungen sollten grundsätzlich hier schreiben. $BIGWORK ist per InfiniBand, und somit im Vergleich zu $HOME deutlich schneller, angebunden. $BIGWORK wird auch als scratch- oder work-Verzeichnis bezeichnet. Das bedeutet, hier können Ideen, wie auf Schmierpapier, ausprobiert und anschließend wieder ausradiert werden. Es ist ein Bereich für jegliche Art von Arbeitsskizze.

$PROJECT Ihr Projekt-Verzeichnis. Jedem Projekt im BIAS wird ein Projektverzeichnis mit entsprechend großem und schnellen(lustre, InfiniBand) Speicherplatz für die längerfristige Speicherung der Projektdaten zugeordnet. Alle Mitglieder des Projekts haben Lese- und Schreibzugriff auf diesen Projektbereich unter `/project/<your-cluster-groupname>` (die shell-Variable $PROJECT zeigt auf denselben Pfad). Die Quota (Speicherplatzbeschränkung) des Projektverzeichnis beträgt zuerst 10TB und steht nur auf den login- und transfer-Knoten zur Verfügung. Das bedeutet, dass Sie das Projektverzeichnis nicht als Ein- oder Ausgabe für Ihre Jobs auf unseren Rechenknoten benutzen können (verwenden Sie dafür $BIGWORK). Sie können auf den login- oder transfer-Knoten ihre Rechen-Daten von $BIGWORK nach $PROJECT und zurück kopieren. Generell ist der Projekt-Speicher nicht als Rechenspeicher geeignet, vielmehr als eine langfristige Ablage im Rechennetz mit guter Netzanbindung an $BIGWORK. In Bezug auf IOPs ist $PROJECT auch relativ langsamer als $BIGWORK. Der Projekt-Speicher wird nicht im Backup gesichert.

Auf $HOME sollten wichtige Dateien gelagert werden, welche bei Verlust nicht einfach wieder herzustellen sind. Zwischenergebnisse von Simulationsrechnungen sollten nach $BIGWORK geschrieben werden, da diese durch erneutes Starten einer Rechnung wieder erzeugt werden können. Hier steht nicht nur mehr Speicherplatz zur Verfügung, die Anbindung des Speichersystems ist durch Verwendung von InfiniBand auch deutlicher schneller. Wird $HOME als Speichersystem während Simulationen genutzt, so wird die Simulationsrechnung sehr wahrscheinlich dadurch ausgebremst. Deshalb sollte darauf geachtet werden, alle automatisch von Anwendungssoftware gesetzten Verzeichnisse, dies sind vor allem temporäre Verzeichnisse, auf $BIGWORK zeigen zu lassen.

Keinesfalls sollte ein Link auf $BIGWORK im home-Verzeichnis angelegt werden. Dabei würden Dateien, welche auf $BIGWORK geschrieben werden sollen, durch $HOME geleitet und erst von dort aus weiter nach $BIGWORK. Beim Lesen entsprechend andersherum. Stattdessen kann die Umgebungsvariable $BIGWORK verwendet werden, siehe Übung in Kapitel 4.4.

Wichtig: Regelmäßiges sichern Ihrer Daten von $BIGWORK auf dem $PROJECT-Speicher oder auf die Rechner des Instituts ist unerlässlich, da $BIGWORK als Scratch-Speichersystem konzipiert ist.

4.1 Quota und grace time

Auf beiden Speichersystemen steht jeweils nur ein Teil des gesamten Speicherplatzes zur persönlichen Verfügung. Dieses Kontingent wird mit dem englischen Begriff *quota* bezeichnet. Es wird unterschieden in *soft quota* und *hard quota*. Das *hard quota* ist eine obere Grenze. Es kann nicht mehr Speicherplatz verbraucht werden,

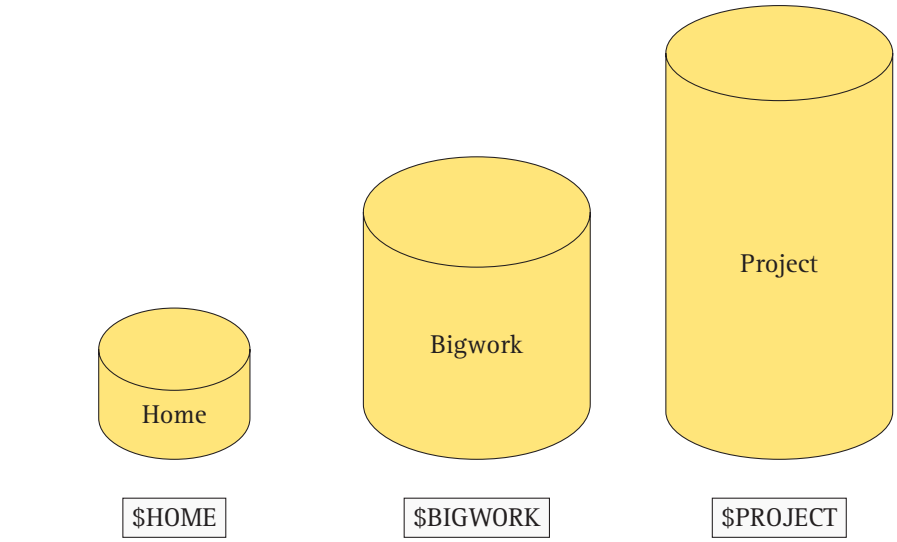

Abbildung 4.1: Dateisysteme im Clustersystem mit ihren Spezifikationen

als durch das *hard quota* vorgegeben. Das *soft quota* hingegen kann überschritten werden. Überschreitet der Speicherplatzverbrauch den durch das *soft quota* vorgegebenen Wert, so beginnt die *grace time*. Dies ist eine Frist, während der man das *soft quota* überschreiten darf. Nach Ablauf dieser Frist muss der Speicherplatzverbrauch unter die Schwelle des *soft quota* gesenkt werden, ansonsten ist kein weiteres Schreiben auf das Speichersystem möglich. Durch unterschreiten des *soft quota* wird die *grace time* zurückgesetzt.

Der Mechanismus des *quota* dient uns dazu, den Speicherplatzverbrauch einzudämmend und das System für die Nutzerschaft möglichst performant zu halten. Generell sollten nicht mehr benötigte Dateien gelöscht werden. Ein niedriger Füllstand sichert insbesondere bei $BIGWORK einen optimalen Betrieb. Die Abfrage des eigenen Speicherplatzverbrauchs und *quota* erfolgt über folgende Befehle, siehe auch die Übung in Kapitel 4.4.

fquota Zeigt den Speicherplatzverbrauch und *quota* von $HOME an.

lquota Zeigt den Speicherplatzverbrauch und *quota* von $BIGWORK un $PROJECT an.

Wichtig: Ist auf $HOME kein Speicherplatz mehr verfügbar, scheitert jede grafische Anmeldung. Eine Anmeldung per ssh (ohne -X) ist weiterhin möglich.

4.2 $BIGWORK's Dateisystem Lustre und stripe count

Wichtig: Alle Angaben in diesem Abschnitt gelten auch für den $PROJECT-Speicher

Technisch besteht $BIGWORK aus mehreren Einzelkomponenten, die zusammen das Speichersystem ergeben, dargestellt in Abbildung 4.2. Grundsätzlich kann $BIGWORK mit den Standardeinstellungen verwendet werden. Es kann jedoch sinnvoll sein, einen Parameter, den *stripe count* anzupassen. Dadurch kann eine höhere Leistung erzielt werden und das Gesamtsystem wird ausgeglichener ausgelastet, was der gesamten Nutzerschaft zugute kommt.

Auf $BIGWORK werden die Daten auf OSTs, *object storage targets*, gespeichert. Jedes OST besteht seinerseits aus einer Anzahl von Festplatten. Standardmäßig werden Dateien, unabhängig von ihrer Größe, auf einem einzigen OST gespeichert. Dies entspricht einem *stripe count* von eins. Der *stripe count* gibt an, auf wie viele OSTs eine Datei aufgeteilt wird. Sprich, in wie viele Streifen eine Datei gedanklich aufgeteilt wird. Durch Aufteilung auf mehrere OSTs können höhere Zugriffsgeschwindigkeiten erreicht werden, da die Schreib- und Lesegeschwindigkeiten mehrerer OSTs und somit einer Vielzahl von Festplatten genutzt werden.

Wichtig: Wer mit Dateien arbeitet, welche größer als 100 MB sind, bitte *stripe count* anpassen, siehe Kapitel 4.5.

Der *stripe count* wird als Zahl der zu verwendenden OSTs gesetzt, wobei -1 für alle verfügbaren OSTs steht. Es bietet sich an, ein Verzeichnis unterhalb von $BIGWORK anzulegen, welches einen *stripe count* von -1 erhält und dort alle Dateien, welche größer als 100 MB sind, zu speichern. Für kleine Dateien, solche kleiner als 100 MB, ist ein *stripe count* von eins, welcher voreingestellt ist, ausreichend.

Wichtig: Um für bestehende Dateien den *stripe count* anzupassen, müssen diese kopiert werden, siehe Kapitel 4.6. Ein verschieben mit `mv` reicht nicht aus.

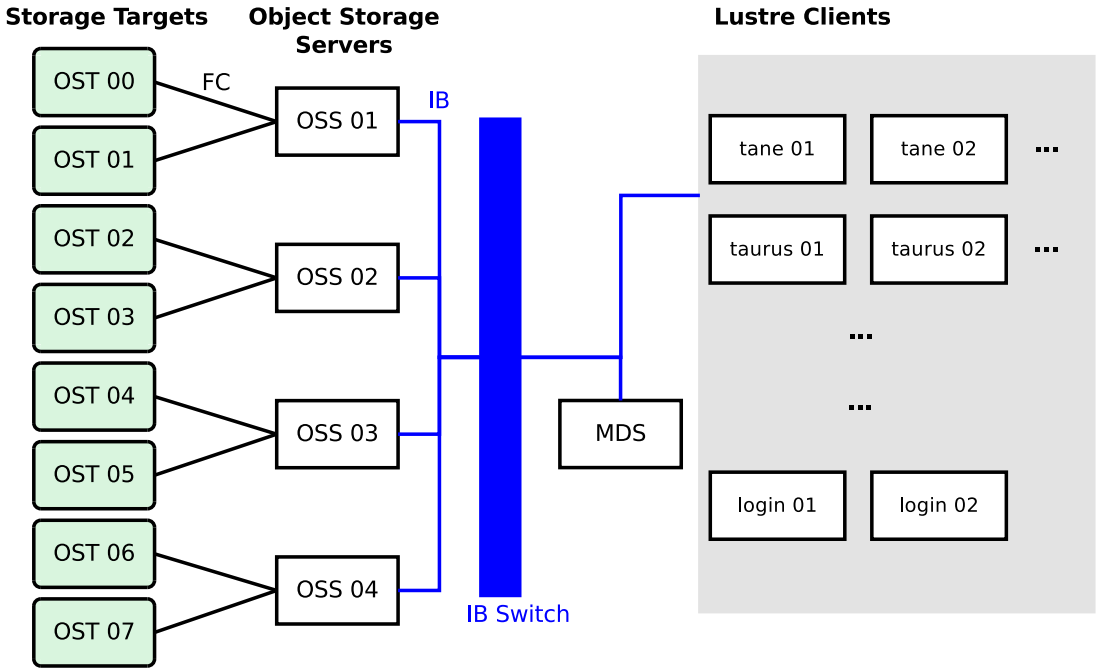

Abbildung 4.2: $BIGWORK's technische Komponenten mit *object storage targets* (OST), *object storage servers* (OSS), *metadata server* (MDS) und InifiniBand (IB) Switch.

4.3 $TMPDIR

Innerhalb eines Jobs steht unter der Variable $TMPDIR ein Verzeichnis auf den Knoten, welche am Job teilnehmen, lokal zur Verfügung. Wenn Speicherplatz lokal benötigt wird sollte man dieses Verzeichnis benutzen.

Wichtig: Nach Ende des Jobs werden alle in $TMPDIR gespeicherten Dateien automatisch gelöscht.

Wer denkt, auf $TMPDIR könnte grundsätzlich schneller geschrieben werden, als auf $BIGWORK, sollte dies unbedingt durch Tests verifizieren. $TMPDIR bietet sich für temporäre Dateien während Rechnungen an bei Anwendungssoftware, die zwingend ein temporäres Verzeichnis benötigt.

4.4 Übung: Dateisysteme benutzen

```
# where are you? lost? print working directory!
pwd
# change directory to your bigwork directory
cd $BIGWORK
# what is the absolute path of your bigwork directory?
pwd
# change directory to your project directory
cd $PROJECT
# what is the absolute path of your project directory?
pwd
# change directory to your home directory
cd $HOME
# display the absolute path of your home directory
pwd
# display your home, bigwork & project quota
checkquota
# make personal directory in your group's project storage
mkdir -m 0700 $PROJECT/$USER
# copy the directory mydir from bigwork to project
cp -r $BIGWORK/mydir $PROJECT/$USER
```

Verbrauchter Speicherplatz wird in Kilobyte angezeigt.
Note: tera, giga, mega, kilo ...

Used space: 000 123 456 789

4.5 Übung: stripe count setzen

```
# get overall bigwork usage, note different fill levels
lfs df -h
# get current stripe settings for your bigwork
lfs getstripe $BIGWORK
# change directory to your bigwork
cd $BIGWORK
# create a directory for large files (anything over 100 MB)
mkdir LargeFiles
# get current stripe settings for that directory
lfs getstripe LargeFiles
# set stripe count to -1 (all available OSTs)
lfs setstripe --count -1 LargeFiles
# check current stripe settings for LargeFiles directory
lfs getstripe LargeFiles
# create a directory for small files
mkdir SmallFiles
# check stripe information for SmallFiles directory
lfs getstripe SmallFiles
```

Use newly created LargeFiles directory to store large files

4.6 Übung: stripe count anpassen

Falls Sie nicht wissen, wie groß die Dateien werden, welche Ihre Anwendung erzeugt, können Sie *stripe size* nachträglich setzen. Als erstes wird eine 100 MB große Datei erstellt.

```
# enter the directory for small files
cd SmallFiles
# create a 100 MB file
dd if=/dev/zero of=100mb.file bs=10M count=10
# check filesize by listing directory contents
ls -lh
# check stripe information on 100mb.file
lfs getstripe 100mb.file
# move the file into the large files directory
mv 100mb.file ../LargeFiles/
# check if stripe information of 100mb.file changed
lfs getstripe ../LargeFiles/100mb.file
# remove the file
rm ../LargeFiles/100mb.file
```

Um den stripe zu ändern, müssen Dateien kopiert (cp) werden. Verschieben (mv) reicht nicht aus.

Erste Methode:

```
# from within the small files directory
cd $BIGWORK/SmallFiles
# create a 100 MB file
dd if=/dev/zero of=100mb.file bs=10M count=10
# copy file into the LargeFiles directory
cp 100mb.file ../LargeFiles/
# check stripe in the new location
lfs getstripe ../LargeFiles/100mb.file
```

Zweite Methode:

```
# create empty file with appropriate stripe count
lfs setstripe --count -1 empty.file
# check stripe information of empty file
lfs getstripe empty.file
# copy file "in place"
cp 100mb.file empty.file
# check that empty.file now has a size of 100 MB
ls -lh
# remove the original 100mb.file and work with empty.file
rm 100mb.file
```

Modules & Anwendungssoftware

Wichtig: Lmod ist seit Dezember 2016 das Software Modulsystem und automatisch nach jeder Anmeldung aktiv

Mit dem Betriebssystem der Maschinen im Clustersystem wird von uns eine schlanke Paketauswahl, sprich Software, installiert. Dadurch bedingt steht Ihnen erst einmal eine geringe Auswahl an Programmpaketen zur Verfügung. Zusätzliche Pakete und Anwendungssoftware werden von uns in Modulen bereit gestellt. Durch die Benutzung von Modulen können Sie Ihre Arbeitsumgebung auf dem Clustersystem auf einfache Art und Weise modifizieren. Global bieten Module den Vorteil, dass z.B. mehrere Versionen einer Anwendung nebeneinander angeboten werden können. Je nach Bedarf können Sie die benötigte Version laden. Beim Laden eines Moduls werden spezifische Operationen für die jeweilige Anwendung durchgeführt, zum Beispiel das Erweitern der Umgebungsvariablen `PATH`, `LD_LIBRARY_PATH` und `MANPATH`.

Das Modulsystem, welches im Clustersystem zum Einsatz kommt, nennt sich `Lmod`. Im Anhang findet sich die Dokumentation zu dem Vorgängersystem, welches vorerst noch der Standard im Clustersystem ist. `Lmod` bietet eine hohe Leistung und bringt eine Vielzahl zeitgemäßer Funktionen mit. Dennoch funktionieren die gängigsten Modulkommandos fast wie gewohnt. Zusätzlich wird mit `EasyBuild` eine systematische Namensstruktur und Versionierung der Anwendungssoftware erzielt. Mehr Informationen sind auf der Lmod Homepage zu finden.

Die `Lmod` Installation benutzt ein hierarchisches Software Modul Namensschema. Dadurch liefert der Befehl `module avail` keine komplette Liste aller zur Verfügung stehenden Software zurück. Lediglich die von der aktuellen Ebene verfügbare Software wird angezeigt. Nach dem Laden grundlegender Module ist zusätzliche Software verfügbar. Speziell das Laden eines Compilers oder einer MPI Implementation bewirkt, dass anschließend alle jene Software Verfügbar ist, welche auf diesen aufgebaut.

Auf der obersten Ebene der Modulhierarchie stehen Module für Compiler, Toolchains und Anwendungssoftware, welche als Binary ausgeliefert wird und somit nicht auf einen Compiler angewiesen ist. In einer Toolchain sind Compiler, MPI Implementierung und numerische Bibliotheken zusammengefasst. Momentan existieren die folgenden Toolchains:

- Compiler only toolchains
  - GCC: GCC + updated binutils
  - iccifort: Intel compilers + GCC
- Compiler + MPI toolchains
  - gompi: GCC + OpenMPI
  - iimpi: Intel compilers + Intel MPI
  - iompi: Intel compilers + OpenMPI
- Compiler + MPI + numerical libraries toolchains
  - foss: gompi + OpenBLAS + FFTW + ScaLAPACK
  - intel: iimpi + Intel MKL
  - iomkl: iompi + Intel MKL

In Tabelle 5.1 sind die GNU basierten Toolchains aufgelistet und in den Tabellen 5.2 und 5.3 die Intel basierten.

Tabelle 5.1: GNU basierte Toolchains

Name	GCC	OpenMPI	FFTW	OpenBLAS	ScaLAPACK	LAPACK	Sub-Toolchain
foss/2014b	4.8.3	1.8.1	3.3.4	0.2.9	2.0.2	3.5.0	gompi/2014b
foss/2015b	4.9.3	1.8.8	3.3.4	0.2.14	2.0.2	3.5.0	gompi/2015b
foss/2016a	4.9.3	1.10.2	3.3.4	0.2.15	2.0.2	3.6.0	gompi/2016a
foss/2016.04	5.3.0	1.10.2	3.3.4	0.2.18	2.0.2	3.6.0	gompi/2016.04
foss/2016b	5.4.0	1.10.3	3.3.4	0.2.18	2.0.2	3.6.1	gompi/2016b

Tabelle 5.2: Intel basierte Toolchains mit Intel MPI

Name	ICC	IFort	IMKL	IMPI	Note
intel/2016a	2016.1.150	2016.1.150	11.3.1.150	5.1.2.150	compiled by GCC-4.9.3-2.25
intel/2016b	2016.3.210	2016.3.210	11.3.3.210	5.1.3.181	compiled by GCC-5.4.0-2.26

Tabelle 5.3: Intel basierte Toolchains mit Open MPI

Name	ICC	IFort	IMKL	OpenMPI	Note
iomkl/2016b	2016.3.210	2016.3.210	11.3.3.210	1.10.3	compiled by GCC-5.4.0-2.26

5.1 Module benutzen

Die Benutzung der Modulumgebung erfolgt mit folgenden Kommandos.

Eine Übersicht aller Module

```
module spider
```

Eine Übersicht aller Module in kompakterer Form

```
module -t spider
```

Nach Modulen suchen, bei denen „string“ im Namen vorkommt

```
module spider string
```

Detaillierte Informationen zu einem Modul mit Informationen, wie dieses geladen werden kann

```
module spider name/version
```

Module anzeigen, welche sofort, sprich von der jetzigen Ebene aus, geladen werden können

```
module avail
```

Manche Module sind vor den Befehlen `avail` und `spider` versteckt. Dies sind vorrangig Systembibliotheken, von denen Anwendungssoftware abhängt. Um versteckte Module anzuzeigen, muss der Parameter `--show_hidden` verwendet werden.

```
module --show_hidden avail
module --show_hidden spider
```

Versteckte Module haben einen Punkt (`.`) vor der Versionsnummer (z.B. `zlib/.1.2.8`).

Momentan geladene Module

```
module list
```

Eine spezifische Version eines Moduls laden

```
module load name/version
```

Falls lediglich der Name des Moduls ohne Versionsnummer angegeben wird, so wird die default-Version, gekennzeichnet mit (D), geladen. Durch das Laden eines Moduls können andere Module als Abhängigkeit mit geladen werden.

Zwei unterschiedliche Versionen des selben Moduls können nicht gleichzeitig geladen werden.

Wechseln zwischen zwei Modulen

```
module swap old new
```

Das angegebene Modul wird aus der aktuellen Umgebung entfernt

```
module unload name
```

Alle geladenen Module aus der aktuellen Umgebung löschen

```
module purge
```

Anzeigen lassen, welche Variablen ein Modul setzt

```
module show name/version
```

Speichert die momentane Auswahl an Modulen unter „name“ ab, um Sie später wieder herzustellen

```
module save name
```

Stellt die unter „name“ abgespeicherte Auswahl an Modulen wieder her

```
module restore name
```

Listet alle abgespeicherten Konfigurationen auf

```
module savelist
```

Liste aller Optionen, welche Lmod für den Befehl `module` bereit stellt

```
module help
```

5.2 Übung: Module benutzen

Als Beispiel für die Arbeit mit Lmod wird im folgenden gezeigt, wie ein Modul für gnuplot geladen wird.

Momentan geladene Module anzeigen

```
module list

No modules loaded
```

Verfügbare gnuplot Versionen anzeigen

```
module -t spider gnuplot

gnuplot/4.6.0
gnuplot/5.0.3
```

Herausfinden wie `gnuplot/5.0.3` geladen werden kann

```
module spider gnuplot/5.0.3

----------------------------------------------------------------------------
  gnuplot: gnuplot/5.0.3
----------------------------------------------------------------------------
    Description:
      Portable interactive, function plotting utility - Homepage: http://gnuplot
          .sourceforge.net/
```

```
    This module can only be loaded through the following modules:

      GCC/4.9.3-2.25  OpenMPI/1.10.2

    Help:
      Portable interactive, function plotting utility - Homepage: http://gnuplot
          .sourceforge.net/
```

Die, laut vorheriger Ausgabe, benötigten Module laden

```
 module load GCC/4.9.3-2.25  OpenMPI/1.10.2

Module for GCCcore, version .4.9.3 loaded
Module for binutils, version .2.25 loaded
Module for GCC, version 4.9.3-2.25 loaded
Module for numactl, version .2.0.11 loaded
Module for hwloc, version .1.11.2 loaded
Module for OpenMPI, version 1.10.2 loaded
```

Schließlich kann das angestrebte `gnuplot` Modul geladen werden

```
 module load gnuplot/5.0.3

Module for OpenBLAS, version 0.2.15-LAPACK-3.6.0 loaded
Module for FFTW, version 3.3.4 loaded
Module for ScaLAPACK, version 2.0.2-OpenBLAS-0.2.15-LAPACK-3.6.0 loaded
Module for bzip2, version .1.0.6 loaded
Module for zlib, version .1.2.8 loaded
.............
.............
```

Um die Prozedur des Ladens zu vereinfachen, kann die momentane Auswahl an Modulen abgespeichert werden. Wir speichern hier unter dem Namen ”mygnuplot”, dieser Name ist willkürlich gewählt. Später kann diese Auswahl an Modulen wieder hergestellt werden.

Abspeichern der Modulauswahl unter dem Namen ”mygnuplot”

```
 module save mygnuplot

Saved current collection of modules to: mygnuplot
```

Wird kein Name angegeben, so wird der Name „default“ verwendet.

Alle geladenen Module entfernen, oder einen neue Shell öffnen

```
 module purge

Module for gnuplot, version 5.0.3 unloaded
Module for Qt, version 4.8.7 unloaded
Module for libXt, version .1.1.5 unloaded
............
............
```

Momentane Auswahl anzeigen lassen. Diese ist jetzt lerer.

```
 module list

No modules loaded
```

Gespeicherte Konfigurationen auflisten

```
 module savelist

Named collection list:
  1) mygnuplot
```

Modulauswahl `gnuplot` wieder herstellen

```
 module restore mygnuplot

Restoring modules to user's mygnuplot
Module for GCCcore, version .4.9.3 loaded
Module for binutils, version .2.25 loaded
Module for GCC, version 4.9.3-2.25 loaded
Module for numactl, version .2.0.11 loaded
.............
.............
```

Arbeiten mit dem Clustersystem

Der Grund, weshalb Sie das Clustersystem nutzen möchten, ist wahrscheinlich die Rechenkapazität. Die Rechenknoten sind allerdings nicht direkt erreichbar. Bei ca. 250 Personen pro Jahr, die das Clustersystem für ihre Forschung nutzen, ist eine Instanz nötig, die die Ressourcen zuteilt. Diese Instanz ist das Batchsystem, auch PBS (portable batch system) genannt. Im Clustersystem setzen wir *TORQUE* als resource manager und *Maui* als scheduler ein, welche auf Anfrage Ressourcen innerhalb eines Jobs zur Verfügung stellen.

Die meisten Arbeiten am Clustersystem werden Sie innerhalb von Jobs durchführen. Jobs bilden den Rahmen für die Nutzung der Rechenkapazität des Clustersystems und werden mit dem qsub-Befehl gestartet. Der qsub-Befehl hat die folgende Form.

```
qsub <Optionen> <Name des Jobscripts>
```

Die Handbuchseite für qsub ist folgendermaßen zu erreichen.

```
man qsub
```

Die Handbuchseite kann durch drucken des 'q' Taste verlassen werden.

6.1 Login-Knoten

Mit der Anmeldung am Clustersystem ist eine Verbindung zu einem Login-Knoten hergestellt. Diese sind nicht für Berechnungen gedacht. Um diese Maschinen zugänglich, sprich die Last möglichst gering zu halten, werden alle Prozesse automatisch abgebrochen, die 30 Minuten CPU-Zeit verbraucht haben. Damit soll verhindert werden, dass Berechnungen auf den Login-Knoten ausgeführt und die Maschinen blockiert werden. Für alle aufwendigen Arbeiten wie Pre- oder Post-Processing und auch manche Kompilierung kann es passieren, dass der Prozess automatisch abgebrochen wird, sprich einfach aus geht. Um diese frustrierende Erfahrung zu vermeiden, sollten Sie auf interaktive Jobs ausweichen.

6.2 Interaktive Jobs

Die einfachste Art Rechenkapazität des Clustersystems zu nutzen, ist einen interaktiven Job zu starten. Hierfür wird auf einem Login-Knoten das qsub-Kommando mit der Option -I verwendet.

```
zzzzsaal@login02:~$ qsub -I
ACHTUNG / WARNING:
'mem' parameter not present; the default value will be used (1800mb)
'walltime' parameter not present; the default value will be used (24:00:00)
'nodes' parameter not present; the default value will be used (nodes=1:ppn=1)
qsub: waiting for job 1001152.batch.css.lan to start
qsub: job 1001152.batch.css.lan ready

zzzzsaal@lena-n080:~$
```

In diesem Beispiel setzt die Nutzerkennung zzzzsaal das qsub-Kommando von der Maschine login02 aus ab. Das Batchsystem gibt Warnmeldungen aus und startet den Job mit der JobID 1001152.batch.css.lan, oder kurz 1001152. Anschließend befindet sich die Nutzerkennung auf der Maschine lena-n080, was an der Eingabeaufforderung durch @lena-n080 zu erkennen ist. Dies ist der Knoten mit der Nummer 80 im Teilcluster Lena. Von jetzt an wird die Rechenleistung dieser Maschine verwendet.

Diese einfachste Form eines interaktiven Jobs benutzt Standardwerte für alle Ressourcenanforderungen. In der Praxis sollte die Anforderung stets an die eigenen Bedürfnisse angepasst werden. Dafür werden dem

qsub-Kommando weitere Optionen mitgegeben. Eine Auflistung möglicher Optionen sind in Kapitel 6.4 zu finden. Im folgenden Beispiel startet die Nutzerkennung zzzzsaal von login02 aus erneut einen interaktiven Job. Für diesen interaktiven Job werden für eine Stunde Zugriff auf einen CPU-Kern auf einer Maschine und 2 GB Arbeitsspeicher angefordert. Zusätzlich wird die Weiterleitung von Fenstern (X window forwarding), durch die Option `-X`, aktiviert. Nun können Anwendungen mit grafischen Benutzeroberflächen verwendet werden.

```
zzzzsaal@login02:~$ qsub -I -X -l nodes=1:ppn=1 -l walltime=01:00:00 -l mem=2GB
qsub: waiting for job 1001154.batch.css.lan to start
qsub: job 1001154.batch.css.lan ready

zzzzsaal@lena-n079:~$
```

Nachdem der Job mit der JobID 1001154 gestartet ist, steht die Maschine lena-n079 zur Benutzung bereit. Ein erweitertes Beispiel zu interaktiven Jobs ist in Abschnitt 6.9 zu finden.

6.3 Batch Jobs

Als Vorbereitung eines batch-Jobs sollten immer interaktive Jobs dienen. Innerhalb interaktiver Jobs können Sie alle Befehle eingeben, welche Sie später auch automatisch in Ihrem batch-Skript ausführen lassen. Diese Befehle sollten Sie ausschließlich dann, wenn alles korrekt innerhalb interaktiver Jobs funktioniert, Zeile für Zeile in ein Skript schreiben. Dieses Skript ist dann das Jobskript. Sollten Sie ein batch-Skript von anderen Personen erhalten haben, nehmen Sie sich die Zeit, alle Befehle in einem interaktiven Job einzugeben. Auf diese Weise verstehen Sie, was die einzelnen Anweisungen bewirken.

Um die selben Ressourcen in einem batch-Job anzufordern, wie in dem Beispiel eines interaktiven Jobs aus Abschnitt 6.2, kann folgendes in eine Datei geschrieben werden.

```
#!/bin/bash -login

# resource specification
#PBS -l nodes=1:ppn=1
#PBS -l walltime=01:00:00
#PBS -l mem=2GB

# commands to execute
date
```

Grob lassen sich Jobskripte in zwei Sektionen unterteilen. Ressourcenanforderungen und auszuführende Befehle. Grundsätzlich sind Zeilen, welche mit # beginnen Kommentarzeilen. Es gibt allerdings zwei Ausnahmen, wie hier zu sehen. Die erste Zeile gibt die Shell an, in der das Script interpretiert wird, hier die bash. Ebenfalls sind die mit #PBS beginnenden Zeilen Direktiven für PBS, das portable batch system und keine Kommentarzeilen. Durch die vorliegenden Ressourcenanforderungen wird für eine Stunde Zugriff auf einen CPU-Kern auf einer Maschine und 2 GB Arbeitsspeicher angefordert. Die Sektion zur Ausführung von Befehlen enthält lediglich einen einzigen Befehl. Das date Kommando gibt das aktuelle Datum inklusive Zeit aus.

Diese Datei wird als batch-job-example.sh abgespeichert. Anschließend kann das Jobskript mit dem qsub-Kommando angeschickt werden.

```
zzzzsaal@login02:~$ qsub batch-job-example.sh
1001187.batch.css.lan
```

Nach dem Abschicken gibt das Batchsystem eine JobID zurück, in diesem Fall 1001187. Ist der Job beendet, so befinden sich zwei vom Batch-System generierte Dateien im Ordner, aus dem der Job abgeschickt wurde.

```
zzzzsaal@login02:~$ ls -lh
total 12K
-rw-r--r-- 1 zzzzsaal zzzz 137 19. Apr 12:54 batch-job-example.sh
-rw------- 1 zzzzsaal zzzz   0 19. Apr 12:59 batch-job-example.sh.e1001187
-rw------- 1 zzzzsaal zzzz  30 19. Apr 12:59 batch-job-example.sh.o1001187
```

Zum einen eine Datei mit der Endung .e1001187, welche mögliche Fehler während der Ausführung enthält. Diese ist in diesem Fall leer. Zum anderen eine Datei mit der Endung .o1001187, welche alle Ausgaben enthält, die während des Jobs auf die Kommandozeile geschrieben worden wären und hierhin umgeleitet wurden. Dies wird überprüft, in dem der Inhalt der .o-Datei ausgegeben wird.

```
zzzzsaal@login02:~$ cat batch-job-example.sh.o1001187
Tue Apr 19 12:59:18 CEST 2016
```

Die Datei enthält die Ausgabe des `date` Kommandos.

Wichtig: Unter Windows geschriebene Jobskripte müssen konvertiert werden, siehe Abschnitt 6.3.1.

6.3.1 Unter Windows geschriebene Jobskripte konvertieren

Wird ein Jobskript unter Windows erstellt und anschließend auf das Clustersystem kopiert, kann es zu folgender Fehlermeldung kommen beim Versuch dieses Jobskript mit `qsub` abzuschicken.

```
zzzzsaal@login02:~$ qsub WindowsDatei.txt
qsub:  script is written in DOS/Windows text format
```

Überprüfen Sie die Datei mit dem `file` Kommando.

```
zzzzsaal@login02:~$ file WindowsDatei.txt
WindowsDatei.txt: ASCII text, with CRLF line terminators
```

Wandeln Sie die Datei in das Unix Format um.

```
zzzzsaal@login02:~$ dos2unix WindowsDatei.txt
dos2unix: converting file WindowsDatei.txt to UNIX format ...
```

Überprüfen Sie die Datei erneut, um sicherzustellen, dass die Konvertierung erfolgreich verlief.

```
zzzzsaal@login02:~$ file WindowsDatei.txt
WindowsDatei.txt: ASCII text
```

6.4 PBS Optionen

Im Folgenden sind einige ausgewählte Optionen angegeben, mit denen Sie Ihre Jobs steuern können. Diese Optionen können sowohl für interaktive, wie für batch-Jobs verwendet werden.

`-N` *name*
: gibt als Job Namen *name* an

`-j oe`
: verbinde (join) Standard Output und Standard Error

`-l nodes=`*n*`:ppn=`*p*
: fordere *n* Knoten und *p* Prozessorkerne pro Knoten an

`-l walltime=`*time*
: maximale Wallclock Zeit im Format hh:mm:ss

`-l mem=`*value*
: Hauptspeicheranforderung, Werte können mit kb, mb oder gb angegeben werden

`-M` *email address*
: schicke Job-relevante Mails an *mailadresse*

`-m abe`
: schicke Mail bei (eine oder mehrere Angaben): *a* - Job Abbruch, *b* - Job Beginn, *e* - Job Ende

`-V`
: exportiert momentan gültige Umgebungsvariablen in den Job

`-q` *queue*
: Queue in welcher der Job eingereiht wird, siehe Abschnitt 6.7

`-W x=PARTITION:`*name*
: Partition, welche verwendet werden soll, siehe Abschnitt 6.7

`-I`
: Interaktiver Job

Mehr Optionen können Sie auf der Handbuchseite finden, die Sie mit folgendem Kommando aufrufen.

```
man qsub
```

6.5 PBS Umgebungsvariablen

Unter `PBS` stehen Umgebungsvariablen zur Verfügung, über die in einem Job auf wichtige Informationen zugegriffen werden kann. Es folgt ein Auszug der möglichen Variablen. Mehr sind auf den Webseiten von Adaptive Computing zu finden oder auf der Handbuchseite von `qsub`.

`PBS_O_WORKDIR`
: Absoluter Pfadname des Verzeichnisses, aus dem qsub aufgerufen wurde

`PBS_NODEFILE`
: Name der Datei, die die Liste der Knoten enthält auf denen der Job läuft

`PBS_QUEUE`
: Name der queue, in der der Job läuft

`PBS_JOBNAME`
: Name des Jobs (von der `-N` Option im Batchscript)

`PBS_JOBID`
: Einzigartige Identifikationsnummer eines Jobs

6.6 PBS Kommandos

`qsub script`
: Übermittelt ein Jobskript an PBS

`showq`
: listet die Job Queue

`qdel` *jobid*
: beendet laufenden Job mit der ID *jobid* bzw. löscht ihn aus der Warteschlange

Alle diese Befehle haben auch sehr ausführliche Handbuchseiten, die sich mit folgenden Befehl aufrufen lassen:

```
man <command>
```

Zum Verlassen der Handbuchseite die Taste `q` drücken.

6.6.1 pbsnodes

Auf den Login-Knoten steht das Kommando `pbsnodes` zur Verfügung, mit dem Sie selbst prüfen können, welche Ressourcen zur Verfügung stehen. Es lässt sich beispielsweise nachschauen, wie viel RAM eine der „Terabyte-Maschinen“ in der queue helena hat.

```
pbsnodes smp-n031
```

Die Ausgabe ist, bei der ersten Betrachtung, etwas unübersichtlich. Sie zeigt unter anderem folgenden Parameter an.

```
physmem=1058644176kb
```

Diese Angabe kann noch in gb umgerechnet werden. 1024kb sind 1mb, 1024mb sind 1gb, etc. ... Auf diese Weise können Sie in Erfahrung bringen, dass Sie maximal 1009 gb RAM pro Maschine in der queue helena anfordern können.

6.7 Queues & Partitionen

Cluster	Knoten	Prozessoren	Kerne /Knoten	Speicher /Knoten (GB)	Partition	Queue
Lena	80	2x Intel Haswell Xeon E5-2630 v3 8-cores, 2.40GHz, 20MB Cache	16	64	lena	all
Tane	96	2x Intel Westmere-EP Xeon X5670 6-cores, 2.93GHz, 12MB Cache	12	48	tane	all
Taurus	54	2x Intel Westmere-EP Xeon X5650 6-cores, 2.67GHz, 12MB Cache	12	48	taurus	all
SMP	9	4x Intel Westmere-EX Xeon E7-4830 8-cores, 2.13GHz, 24MB Cache	32	256	smp	all
	9	4x Intel Backton Xeon E7540 6-cores, 2.00GHz, 18MB Cache	24	256	puresmp	all
	3	4x Intel Westmere-EX Xeon E7-4830 8-cores, 2.13GHz, 24MB Cache	32	1024	–	helena
FCH	33				.[1]	all

Es gibt drei Jobqueues:

`all`
: Die Standard-Queue. Sie muss nicht explizit angegeben werden, weil sie als *default* eingestellt ist. PBS schickt einen Job je nach angegebenen Ressourcen auf passende Knoten.

`helena`
: Für Programme, die sehr große Hauptspeicheranforderungen haben (bis 1 Terabyte).

`test`
: Für Tests. Es steht ein Knoten mit bis zu 12 Prozessoren und 48 GB RAM zur Verfügung. Maximale Walltime ist 6 Stunden.

Zusätzlich zu den Queues gibt es verschiedene Partitionen, wodurch sich z.B. die Anforderung innerhalb der Queue all feiner steuern lässt, siehe Tabelle. Forschungscluster-Housing Maschinen sind pro Institut in eine Partition eingeteilt.

[1] Jedes Institut hat eine eigene Partition

6.8 Maximale Ressourcen Anforderungen

Es existieren einige Maximalwerte, welche nicht überschritten werden können. Die maximale Laufzeit pro Job ist begrenzt, genauso wie die maximale Anzahl gleichzeitig laufender Jobs. Auch die Anzahl gleichzeitig verwendeter CPUs ist begrenzt. Alle diese Grenzen gelten pro Nutzerkennung.

Walltime Die maximale Walltime beträgt 200 Stunden pro Job

Jobs Die maximale Anzahl gleichzeitig laufender Jobs pro Benutzerkennung ist 64

CPUs Die Gesamtanzahl an CPUs (ppn), welche alle laufenden Jobs nutzen können, ist 768 pro Benutzerkennung

6.9 Übung: Interaktiver Job

```
# start an interactive job, what happens?
qsub -I
# exit this interactive job
exit
# specify all resource parameters, so no defaults get used
qsub -I -X -l nodes=1:ppn=1 -l walltime=01:00:00 -l mem=2GB
# load module for octave
module load octave/3.8.1
# start octave
octave
# inside octave the following commands create a plot
octave:1> x = 0:10;
octave:2> y = x.^2;
octave:3> h = figure(1);
octave:4> plot(x,y);
octave:5> print('-f1','-dpng','myplot')
octave:6> exit
# display newly created image
display myplot.png
```

Interactive jobs are useful for debugging, always use interactive first

6.10 Übung: Batch Job

Create a file named MyBatchPlot.m

```
x = 0:10;
y = x.^2;
h = figure(1);
plot(x,y);
print('-f1','-dpng','MyBatchPlot');
```

Create a file named MyFirstBatchJob.sh

```
#!/bin/bash -login
#PBS -l nodes=1:ppn=1
#PBS -l walltime=01:00:00
#PBS -l mem=2GB
# load octave module
module load octave/3.8.1
# start octave
octave MyBatchPlot.m
```

Submit the job script

```
qsub MyFirstBatchJob.sh
```

Check files MyFirstBatchJob.sh.o* and MyFirstBatchJob.sh.e*

Anwendungssoftware

Wichtig: Wir versuchen unsere Dokumentation aktuell zu halten, bei Software Anwendungen ist dies jedoch sehr schwer. Der Befehl `module spider` auf dem Clustersystem liefert eine umfassende Liste verfügbarer Anwendungen. Dieser Abschnitt kann nicht die anwendungseigene Dokumentation ersetzen. Bitte lest auch die Dokumentation Eurer Anwendungen.

Im Clustersystem steht eine breite Auswahl an Anwendungssoftware zur Benutzung bereit. Diese als Modul verfügbaren Anwendungen liegen auf einem per NFS eingebundenen Speichersystem. Sollten Sie eine andere Version einer vorhandenen Anwendung benötigen, oder eine noch nicht vorhandene Anwendung, nehmen Sie bitte Kontakt auf. Voraussetzung für die Bereitstellung einer Software im Clustersystem ist, dass diese für Linux verfügbar ist. Sollte es sich um eine lizenzpflichtige Anwendung handeln, müssen außerdem weitere Punkte geklärt werden.

Einige Windows Anwendungen können auf dem Clustersystem mit Hilfe von `wine` oder `Singularity` Containern ausgeführt werden. Informationen zu `Singularity` finden sich in Kapitel 7.2, oder nehmen Sie bei Fragen Kontakt auf.

Wichtig: Wir haben uns dazu entschlossen, die folgenden Abschnitte vor allem in englisch bereit zu stellen. Falls Sie eine Übersetzung benötigen, nehmen Sie bitte Kontakt auf.

7.1 Build software from source code on the cluster

Sub-clusters of the cluster system have different CPU architectures. The command `lcpuarchs` issued on the login nodes lists all available CPU types.

```
login01 ~ $ lcpuarchs -v
CPU arch names       Cluster partitions
--------------       ------------------
haswell              fi,haku,iqo,isd,iwes,lena,pci,smp
nehalem              nwf,smp,tane,taurus,tfd
sandybridge          bmwz,iazd,isu,itp

CPU of this machine: haswell

For more verbose output type: lcpuarchs -vv
```

Therefore, if a software executable built with the target specific compiler options runs on a machine with not suitable CPUs, then the „Illegal instruction“ error message will be triggered. For example, if you compile your program on `haswell` node (eg. `lena` sub-cluster) with the `gcc` compiler option `-march=native`, then the program will not run on `nehalem` nodes (eg. `tane` sub-cluster).

This section explains how to build a software on the cluster system to avoid the mentioned problem.

In your `HOME` (or `BIGWORK`) directory create build/install directories for every available CPU architecture and also the directory `source` for storing software installation sources.

In the exmple below we want to compile an example software `my-soft` of version `3.1`.

```
login03~$ mkdir -p ~/sw/source
login03~$ mkdir -p ~/sw/nehalem/my-soft/3.1/{install,build}
login03~$ mkdir -p ~/sw/haswell/my-soft/3.1/{install,build}
login03~$ mkdir -p ~/sw/sandybridge/my-soft/3.1/{install,build}
```

Copy software installation archive to the source directory.

```
login03~$ mv my-soft-3.1.tar.gz ~/sw/source
```

Build `my-soft` for every available CPU architecture by submitting an interactive job to node with the proper CPU type. For example, to compile `my-soft` for `nehalem` nodes.

```
login03~$ qsub -I -l nodes=1:nehalem:ppn=4,walltime=6:00:00,mem=16gb
```

Then unpack and build the software.

```
taurus-n034~$ tar -zxvf ~/sw/source/my-soft-3.1.tgz -C ~/sw/$ARCH/my-soft/3.1/
    build
taurus-n034~$ ./configure --prefix=~/sw/$ARCH/my-soft/3.1/install && make &&
    make install
```

Finally, use the environment variable `$ARCH` in your jobs scripts to access the right installation path of `my-soft` executable for the current work node.

```
login03~$ cat job-my-soft.sh
#PBS -N my-soft
#PBS -l nodes=4:ppn=8,walltime=12:0:0,mem=128gb,

# change to work dir
cd $PBS_O_WORKDIR

# run my_soft
~/sw/$ARCH/my-soft/3.1/install/bin/my-soft.exe file.input
```

7.2 Singularity Container

Wichtig: Diese Anleitung wurde für Singularity 2.4.2 geschrieben.

„Singularity enables users to have full control of their environment. Singularity containers can be used to package entire scientific workflows, software and libraries, and even data.“[1]

7.2.1

Mit Singularity ist es möglich Container im Clustersystem auszuführen, als wären es native Programme auf den Rechenknoten. Zum Beispiel ist es möglich unter Scientific Linux eine Ubuntu Anwendung laufen zu lassen. Dazu wird ein Container Image erstellt, darin Ubuntu und die entsprechende Anwendung installiert und zum Clustersystem kopiert. Anschließend kann die Anwendung mit Hilfe von Singularity in ihrer nativen Ubuntu Umgebung ausgeführt werden.

Der größte Vorteil von Singularity ist die Ausführung der Container mit Nutzerrechten auf dem Clustersystem. Gleichzeitig besteht Zugriff auf alle Netzwerkressourcen des Clustersystems, wie `HOME`, `BIGWORK` and `PROJECT`.

Zusätzlich kann mit Singularity das Message Passing Interface (MPI) verwendet werden, genauso wie unsere InfiniBand Fabric oder Omni-Path Architektur.

7.2.2 Singularity Container auf dem Clustersystem

Vor der Arbeit mit Singularity empfehlen wir die Singularity Dokumentation zu lesen. Nachfolgend sind die grundlegenden Schritte aufgeführt, um mit der Arbeit beginnen zu können.

Wichtig: Neuere Distributionen wie z.B. Ubuntu 18.04 oder Fedora 28 sind momentan nicht auf dem Clustersystem als Singularity Container ausführbar. Scientific Linux 6 verwendet einen Kernel, welcher zu alt für die glibc dieser Distributionen ist.

7.2.3 Singularity auf dem eigenen Rechner installieren

Zunächst sollte Singularity auf einem Rechner installiert werden, auf dem Sie administrative Rechte haben. Root- oder sudo-Rechte sind nötig für den bootstrap-Prozess, oder wenn ein ein bestehendes Image verändert werden soll.

```
VERSION=2.4.2
wget
https://github.com/singularityware/singularity/releases/download/$VERSION/
    singularity-$VERSION.tar.gz
tar xvf singularity-$VERSION.tar.gz
cd singularity-$VERSION/
./configure --prefix=/usr/local
make
sudo make install
```

Um zu überprüfen, ob Singularity installiert ist, folgenden Befehl absetzen

```
singularity
```

Eine detaillierte Installationsanleitung findet sich online.

7.2.4 Singularity Container mit recipe-Datei erstellen

Um reproduzierbare Container zu erhalten, empfehlen wir die Verwendung von recipe-Dateien. Auf dem Clustersystem ist es nicht möglich bestehende Container zu verändern oder neue per bootstrap zu erzeugen, da

[1] Singularity homepage

Ihnen die root-Rechte fehlen. Hier können Sie lediglich Container ausführen. Das folgende recipe erstellt einen `CentOS 7` Container.

Erstellen Sie die Datei mit dem Namen `centos7.def`.

```
BootStrap: yum
OSVersion: 7
MirrorURL:
http://mirror.centos.org/centos-%{OSVERSION}/%{OSVERSION}/os/$basearch/
Include: yum wget

%setup
  echo "This section runs on the host outside the container during bootstrap"

%post
  echo "This section runs inside the container during bootstrap"

  # install packages in the container
  yum -y groupinstall "Development Tools"
  yum -y install vim python epel-release
  yum -y install python-pip

  # install tensorflow
  pip install --upgrade pip
  pip install --upgrade tensorflow

  # enable access to BIGWORK and PROJECT storage on the cluster system
  mkdir -p /bigwork /project

%runscript
  echo "This is what happens when you run the container"

  echo "Arguments received: $*"
  exec /usr/bin/python "$@"

%test
  echo "This test will be run at the very end of the bootstrapping process"

  python --version
```

Diese recipe Datei benutzt das `YUM` bootstrap Modul, um das Betriebssystem `CentOS 7` in einem Container zu installieren. Für andere Arten der bootstrap Installation, z.B. `docker`, oder Details zu recipe Dateien, siehe die online Dokumentation.

Nun kann ein Container erstellt werden. Dies muss auf einem Rechner durchgeführt werden, auf dem Sie root-Rechte haben.

```
sudo singularity build centavo.img centos7.def
```

Der neu erstellte Container kann mit folgendem Befehl betreten werden.

```
singularity shell centos7.img
```

Standardmäßig werden Singularity Container als squashfs Imagedateien erstellt, welche lediglich lesbar sind. Um Container verändern zu können, z.B. um zusätzliche Pakete zu installieren, muss der Container erst in eine schreibbare Sandbox umgewandelt werden.

```
sudo singularity build --sandbox centos7-sandbox centos7.img
```

Es entsteht ein Sandbox Verzeichnis namens *centos7-sandbox*, welches mit folgendem Befehl betreten und Änderungen vorgenommen werden können.

```
sudo singularity shell --writable centos7-sandbox
```

Wir empfehlen Änderungen in die recipe Datei zu übernehmen, damit der Container reproduzierbar ist.

Andere Singularity Optionen werden durch den folgenden Befehl ausgegeben.

```
singularity help
```

7.2.5 Singularity Container vom Docker oder Singularity Hub laden

Eine andere Möglichkeit einen Container zu erhalten, ist du Nutzung von Docker Hub oder Singularity Hub image repositories. Nähere Informationen finden sich in der Singularity online documentation.

7.2.6 Upload des Containers ins BIGWORK Verzeichnis auf dem Clustersystem

Unter Linux kann der entstandene Container mit folgendem Befehl zum Clustersystem übertragen werden. Ansonsten können Sie gerne Ihre bevorzugte Methode verwenden.

```
scp centos7.img username@transfer.cluster.uni-hannever.de:/bigwork/username
```

7.2.7 Ausführen eines Containers

Wichtig: Um Singularity Container ausführen zu können, müssen diese in Ihrem `BIGWORK`-Verzeichnis liegen.

Einloggen auf dem Clustersystem.

```
ssh username@login.cluster.uni-hannever.de
```

Die nötigen Module laden und den Container ausführen.

```
username@login01:~$ cd $BIGWORK
username@login01:~$ module load GCC/4.9.3-2.25 Singularity/2.4.2
username@login01:~$ singularity run centos7.img --version
```

Der `run` Befehl führt alle Befehle des `%runscript` Abschnittes der recipe Datei aus. Durch Benutzung des `exec` Befehls können beliebige Befehle innerhalb des Containers ausgeführt werden. Zum Beispiel kann der Inhalt der Datei `/etc/os-release` von innerhalb des Containers ausgegeben werden.

```
username@login01:~$ singularity exec centos7.img cat /etc/os-release
```

Wichtig: Es besteht Zugriff (lesend & schreibend) auf Ihre `HOME`, `BIGWORK` und `PROJECT` (nur auf login-Knoten) Verzeichnisse innerhalb des Containers. Zusätzlich werden die <Verzeichnisse `/tmp` und `/scratch` automatisch eingebunden. Letzteres lediglich auf Rechenknoten.

7.2.8 Singularity & parallele MPI Anwendungen

Um Ihre parallele MPI Anwendung innerhalb eines Containers auf dem Clustersystem zu betreiben, muss die passenden MPI Bibliotheken innerhalb des Containers zur Verfügung stehen. Außerdem sollte, um die InfiniBand Fabric des Clustersystems für die MPI Kommunikation nutzen zu können, der passende Treiber installiert sein.

Das folgende Beispiel recipe `ubuntu-openmpi.def` holt einen Container aus dem `Docker Hub` und installiert die benötigten MPI und InfiniBand Pakete.

```
BootStrap: docker
From: ubuntu:xenial

%post
# install openmpi & infiniband
apt-get update
apt-get -y install openmpi-bin openmpi-common libibverbs1 libmlx4-1
```

```
# enable access to BIGWORK and PROJECT storage on the cluster system
mkdir -p /bigwork /project

# enable access to /scratch dir. required by mpi jobs
mkdir -p /scratch
```

Nachdem Sie eine Imagedatei `ubuntu-openmpi.img` erstellt und zum Clustersystem übertragen haben, wie im vorangegangenen Abschnitt beschrieben, können Sie Ihre MPI Anwendung wie folgt aufrufen (vorausgesetzt, Sie haben bereits einige Rechenknoten reserviert).

```
module load foss/2016a
module load GCC/4.9.3-2.25 Singularity/2.4.2
mpirun singularity exec ubuntu-openmpi.img /path/to/your/parallel-mpi-app
```

Die obigen Befehle können sowohl innerhalb eines interaktiven Jobs, als auch in einem batch-Job ausgeführt werden.

7.2.9 Weitere Informationen

- Singularity home page
- Singularity Hub
- Docker Hub

7.3 MATLAB

MATLAB is a technical computing environment for high performance numeric computation and visualization. MATLAB integrates numerical analysis, matrix computation, signal processing and graphics. MATLAB toolboxes are collections of algorithms that enhance MATLAB's functionality in domains such as signal and image processing, data analysis and statistics, mathematical modeling, etc.

7.3.1 Versions and Availability

You can use `module spider matlab` to get a list of available versions. Feel free to contact cluster team if you need other versions. MATLAB is proprietary software and can be used on the cluster for non-commercial, academic purposes only. As a free alternative to MATLAB you might consider GNU Octave, which is mostly compatible with MATLAB and provides most of the features as well.

7.3.2 Running MATLAB

As MATLAB by default utilizes all available CPU cores, excessive use of MATLAB on the login nodes would prevent other logged in users from using resources. The recommended way to work with MATLAB is to submit a job (interactive or batch) and start MATLAB from within the dedicated compute node assigned to you by the batch system.

To submit an interactive job, you can use the command (you may adjust the numbers per your need).

```
login03~$ qsub -I -X -l nodes=1:ppn=12,walltime=02:00:00,mem=36gb
```

The -X flag enables X11 forwarding on the compute node if you need to use the graphical MATLAB user interface.

Once you are assigned a compute node, you can run MATLAB interactively by loading the MATLAB module. To load the default version of MATLAB module, use `module load MATLAB`. To select a particular MATLAB version, use `module load MATLAB/version`. For example, to load MATLAB version 2017a, use `module load MATLAB/2017a`.

To start MATLAB interacively with graphical user interface (GUI), after having loaded MATLAB module, type the command.

```
smp-n010~$ matlab
```

This requires the -X flag for the qsub command as well as X11 forwarding enabled for your SSH client or using X2GO (recommended) for logging in onto the cluster system. The following command will start an interactive, non-GUI version of MATLAB.

```
smp-n010~$ matlab -nodisplay -nosplash
```

Type `matlab -h` for a complete list of command line options.

In order to run MATLAB non-interactively via the batch system, you will require a batch submission script. Below is an example batch script (`matlab-job-serial.sh`) for a serial run that will execute MATLAB script (`hello.m`) in a single thread.

```
#!/bin/bash
#PBS -N matlab_serial
#PBS -M my@email.address
#PBS -m bae
#PBS -j oe
#PBS -l nodes=1:ppn=1
#PBS -l walltime=00:10:00
#PBS -l mem=4gb

# Compute node the job ran on
echo "Job ran on:" $HOSTNAME
```

```
# Load modules
module load MATLAB/2017a

# Change to work dir:
cd $PBS_O_WORKDIR

# Log file name
LOGFILE=$(echo $PBS_JOBID | cut -d"." -f1).log

# The program to run
matlab -nodesktop -nosplash < hello.m > $LOGFILE 2>&1
```

Example MATLAB script, `hello.m`:

```
% Example MATLAB script for Hello World
disp 'Hello World'
exit
% end of example file
```

To run `hello.m` via the batch system, submit the `matlab-job-serial.sh` file with the following command:

```
login03~$ qsub matlab-job-serial.sh

2232617.batch.css.lan
```

Output from the running of the MATLAB script will be saved in the `$LOGFILE` file, which in this case expands to `2232617.log`.

7.3.3 Parallel Processing in MATLAB

Please note: MATLAB parallel computing across multiple compute nodes is not supported by the cluster system at the moment.

MATLAB supports both implicit multithreding as well as explicit parallelism provided by the Parallel Computing Toolbox (PCT) which requires specific commands in your MATLAB code in order to create threads.

Implicit multithreading allows some functions in MATLAB, particularly linear algebra and numerical routines such as `fft`, `eig`, `svd`, etc, to distribute the workload between cores of the node that your job is running on and thus run faster than on a single core. By default, all of the current versions of MATLAB available on the cluster have multithreading enabled. **A single MATLAB session will run as many threads as there are cores on a compute node reserved for your job by the batch system.** For example, if you request `nodes=1:ppn=4`, your MATLAB session will spawn four threads.

```
#!/bin/bash
#PBS -N matlab_multithread
#PBS -M my@email.address
#PBS -m bae
#PBS -j oe
#PBS -l nodes=1:ppn=4
#PBS -l walltime=00:20:00
#PBS -l mem=16gb

# Compute node the job ran on
echo "Job ran on:" $HOSTNAME

# Load modules
module load MATLAB/2017a

# Change to work dir:
cd $PBS_O_WORKDIR
```

```
# Log file name
LOGFILE=$(echo $PBS_JOBID | cut -d"." -f1).log

# The program to run
matlab -nodesktop -nosplash < hello.m > $LOGFILE 2>&1
```

However, if you want to disable multithreading you may either request `nodes=1:ppn=1` (see an example job script above) or add the option `-singleCompThread` when running MATLAB.

7.3.4 Using the Parallel Computing Toolbox

MATLAB Parallel Computing Toolbox (PCT) (parallel for-loops, special array types, etc.) lets you make explicit use of multicore processors, GPUs and clusters by executing applications on workers (MATLAB computational engines) that run locally. At the moment, the cluster system does not support parallel processing across multiple nodes (MATLAB Distributed Computing Server). As such, parallel MATLAB jobs are limited to a single compute node with the „local“ pool through use of the MATLAB PCT.

Specific care must be taken when running multiple MATLAB PCT jobs. When you submit multiple jobs that are all using MATLAB PCT for parallelization, all of the jobs will attempt to use the same defaut location for storing information about the MATLAB pools that are in use, thereby creating a race condition where one job modifies the files that were put in place by another. The solution is to have each of your jobs that will use the PCT set a unique location for storing job information. An example batch script `matlab-job-pct.sh` is shown below.

```
#!/bin/bash
#PBS -N matlab_multithread_pct
#PBS -M my@email.address
#PBS -m bae
#PBS -j oe
#PBS -l nodes=1:ppn=12
#PBS -l walltime=00:40:00
#PBS -l mem=40gb

# Compute node the job ran on
echo "Job ran on:" $HOSTNAME

# Load modules
module load MATLAB/2017a

# Change to work dir:
cd $PBS_O_WORKDIR

# Log file name
LOGFILE=$(echo $PBS_JOBID | cut -d"." -f1).log

# The program to run
matlab -nodesktop -nosplash < pi_parallel.m > $LOGFILE 2>&1
```

And the corresponding MATLAB script `pi_parallel.m`, which in addition starts the correct number of parallel MATLAB workers depending on the requested cores.

```
% create a local cluster object
pc = parcluster('local')

% explicitly set the JobStorageLocation to
% the temp directory that is unique to each cluster job
pc.JobStorageLocation = getenv('TMPDIR')

% start the matlabpool with maximum available workers
parpool (pc, str2num(getenv('PBS_NP')))
```

```
R = 1
darts = 1e7
count = 0
tic
parfor i = 1:darts
  x = R*rand(1)
  y = R*rand(1)
  if x^2 + y^2 <= R^2
    count = count + 1
  end
end

myPI = 4*count/darts
T = toc
fprintf('The computed value of pi is %8.7f\n',myPI)
fprintf('The parallel Monte-Carlo method is executed in %8.2f seconds\n', T)
delete(gcp)
exit
```

For further details on MATLAB PCT refer to the online documentation.

7.3.5 Build MEX File

See e.g. online documentation for details on how to build mex files. This section is a straight transscript of that website adapted to fit the cluster system's module system. First, get hold of the example file `timestwo.c` which comes with MATLAB. This can be done from within MATLAB.

```
copyfile(fullfile(matlabroot,'extern','examples','refbook','timestwo.c'),'.','f
    ')
```

Each MATLAB version needs a specific version of GCC in order to build mex files. This is the crucial part. See MATLAB documentation for details.

```
$ module load MATLAB/2018a
$ module load GCC/6.3.0-2.27
$ ls
timestwo.c
$ mex timestwo.c
Building with 'gcc'.
MEX completed successfully.
$ ls
timestwo.c timestwo.mexa64
```

Notice the file `timestwo.mexa64` was created. You can now use it like a function.

```
$ matlab -nodesktop -nosplash
>> timestwo(6)

ans =

    12
```

7.3.6 Toolboxes and Features

On the cluster system you can use the university's MATLAB campus licence. Please see for details and a list of available toolboxes and features.

7.3.7 Further Reading

- MATLAB online documentation

- MATLAB Parallel Computing Toolbox

7.4 NFFT

Please note: These instructions were written for NFFT 3.4.1.

NFFT is available as a module on the cluster system. However, there may be situations where you need to compile your own version. For example if you need the MATLAB interface as mex file. When compiling from source, take into account that the cluster system's CPU architecture is heterogeneous and you best compile a version for every architecture you will be using in order to avoid problems, see section 7.1.

Execute the following commands to load prerequisites.

```
$ module load foss/2016a FFTW/3.3.4 gompi/2016a
```

Download the NFFT sources and change to the directoty containing nfft-3.4.1 sources. Execute the following commands in order to build NFFT with MATLAB 2018a interface. Substitute zzzzsaal with your own user name.

```
$ ./configure --prefix=/bigwork/zzzzsaal/nfft/nfft-3.4.1-compiled
--with-matlab=/sw-eb/apps/software/haswell/Core/MATLAB/2018a/
$ make
$ make install
```

You end up with libnfft.mexa64 located in the lib directory. Use it as detailed here. In order to test your build, start the MATLAB version you built for.

```
$ module load MATLAB/2018a
$ matlab -nodisplay -nosplash
```

Inside MATLAB issue the following commands. Change zzzzsaal to your user name.

```
>> addpath(genpath('/bigwork/zzzzsaal/nfft/nfft-3.4.1-compiled/lib'))
>> cd /bigwork/zzzzsaal/nfft/nfft-3.4.1-compiled/share/nfft/matlab/nfft/
>> simple_test
```

Daten ins Archiv kopieren

Wichtig: Das Archiv wird vom Service Archivierung betrieben und ist nicht Teil des Clustersystems.

Das Archiv dient dazu, Ergebnisse und wichtige Simulationsdaten dauerhaft zu speichern. Um es zu verwenden, muss jeder Account einmalig freigeschaltet werden. Dazu besucht man die BIAS-Webseite und loggt sich dort mit Benutzername und Passwort ein. Nach Anklicken des Links *Ihren Benutzernamen für die Nutzung des Archivsystems zulassen* dauert es ca. eine Stunde, bis das Archivsystem verwendet werden kann.

8.1 Quota

Die Archivierung von Dateien im Archivsystem der Leibniz Universität Hannover unterliegt einer Quotierung auf Dateianzahl und Speichermenge. Bitte informieren Sie sich über die generellen Quotagrenzen auf der Webseite des Archivierungsdiensts unter `http://www.luis.uni-hannover.de/archivierung.html`.

8.2 Daten ins Archiv übertragen

Für den Austausch von Dateien mit dem Archivsystem der Leibniz Universität Hannover empfehlen wir die Verwendung des dafür vorgesehenen, dedizierten Transfersknotens, siehe Abschnitte 3.2 und 3.4.

8.3 Login mit lftp

Das Archiv ist unter dem Servernamen `archiv.luis.uni-hannover.de` erreichbar. Das Aufschalten wird mit dem `lftp`-Kommando erreicht.

```
username@clustertransfer:~$ lftp <username>@archiv.luis.uni-hannover.de
```

Man wird nach dem Cluster-Passwort gefragt. Nach Eingabe des Passworts bekommt man den `lftp`-Prompt:

```
lftp <username>@archiv.luis.uni-hannover.de:~>
```

Jetzt kann man z.B. den Befehl `ls` verwenden, um sich den Verzeichnisinhalt im Archiv anzusehen. Gleichzeitig prüfen wir damit, ob eine Verbindung erfolgreich aufgebaut werden konnte:

```
 lftp <username>@archiv.luis.uni-hannover.de:~> ls
```

Beim ersten Besuch des Archivsystems mit Ihrem neu freigeschalteten Account enthält das Verzeichnis keine Dateien. Ein `ls`-Befehl erzeugt in diesem Fall keine Ausgabe. Beendet wird die Verbindung mit `exit`.

```
lftp <username>@archiv.luis.uni-hannover.de:~> exit
<username>@clustertransfer:~$
```

Als Alias für `exit` lassen sich die Befehle `quit` und `bye` verwenden.

8.4 Kopieren von Dateien in das Archivsystem

Auf dem Transferknoten des Clustersystems eingeloggt wechselt man in das Verzeichnis, in dem die zu kopierenden Daten liegen.

```
clustertransfer:~$ cd $BIGWORK/my_data_dir
clustertransfer:/bigwork/username/my_data_dir$
```

Nach dem Login via `lftp` verwenden wir den `put`-Befehl.

```
clustertransfer:/bigwork/username/my_data_dir$ lftp
<username>@archiv.luis.uni-hannover.de:~>
lftp <username>@archiv.luis.uni-hannover.de:~> put myfile.tar.gz
```

Die Datei `myfile.tar.gz` liegt in diesem Beispiel in dem Verzeichnis, in das wir vorher gewechselt haben. Durch die Übertragung per `put` ist die Datei nun auch im Archivsystem zu finden. Die aus der Arbeit mit den Unix-Shells bekannte `TAB`-Vervollständigung für Datei- und Ordnernamen funktioniert auch in `lftp`.

Die Speicherung vieler kleiner Dateien im Archivsystem ist nicht erwünscht. Weil die Daten in mindestens einer Kopie auf Mangnetband gelagert werden ist es sinnvoller, einen konstanten Datenstrom zu speichern. Daher sind wenige große Dateien zu bevorzugen. Bevor Sie die Übertragung in das Archivsystem durchführen empfehlen wir, kleine Dateien zu einer großen Datei zusammen zu fassen, z.B. mit den Kommandos `tar` oder `zip`. Dies kann sich auch durch eine bessere Ausnutzung der Quota (siehe Abschnitt **Quota**) Ihres Accounts bemerkbar machen.

Um mehrere (große) Dateien auf einmal zu übertragen verwendet man den `mput` Befehl (kurz für *multiple put*). Dieser Befehl versteht die Wildcard `*` wie sie auch in der Bash verwendet wird.

```
lftp <username>@archiv.luis.uni-hannover.de:~> mput mydata*.tar.gz
```

8.5 Dateien aus dem Archivsystem holen

Um Dateien vom Archivsystem zurückzuholen benutzt man den `get` Befehl.

```
lftp <username>@archiv.luis.uni-hannover.de:~> get myfile.tar.gz
```

Der Befehl legt die Datei an dem Ort ab, an dem man sich befand, bevor man mit `lftp` ins Archiv ging. Für mehr als eine Datei existiert `mget` (*multiple get*). Das Zurückholen dauert u.U. ein paar Minuten, bis der Transfer beginnt. Der Bandroboter benötigt diese Zeit, um das Magnetband zu holen und an die Stelle zu spulen, an der sich die gesuchte Datei befindet.

8.6 Noch mehr nützliche Befehle

Um den Inhalt des aktuellen *lokalen* Verzeichnisses zu sehen verwendet man `!ls`; ein Ausrufezeichen führt das Kommando auf dem Rechner aus, auf dem `lftp` gestartet wurde. Die Ausgabe des aktuellen *lokalen* Verzeichnisses erfolgt durch eingabe von `lpwd` am Prompt von lftp.

Man kann auf dem Archivsystem auch Verzeichnisse anlegen. Das Kommando lautet `mkdir`.

```
lftp <username>@archiv.luis.uni-hannover.de:~> mkdir myDir
```

Ein Verzeichnis wechselt man wie gewohnt mit `cd`.

```
lftp <username>@archiv.luis.uni-hannover.de:~> cd myDir
```

Zurück nach oben geht es so:

```
lftp <username>@archiv.luis.uni-hannover.de:~> cd ..
```

Um das lokale Verzeichnis zu wechseln, benutzt man den `lcd`-Befehl (kurz für *local cd*).

```
lftp <username>@archiv.luis.uni-hannover.de:~> lcd /bigwork/<username>/datadir
```

8.7 Zum Weiterlesen

- Man-Page von `lftp`

```
clustertransfer:~$ man lftp
```

(Bild hoch/runter, Leertaste und Pfeiltasten zum Blättern, ’q’ zum Beenden)

- Service *Archivierung*:
  http://www.luis.uni-hannover.de/archivierung.html

Zitieren des Clustersystems

Es freut uns sehr, wenn die Verwendung des Clustersystems im Paper oder Vortrag auch gewürdigt wird (z.B. im Abschnitt *Acknowledgements*). Damit die Wortwahl leichter fällt, haben wir ein paar Beispiele auf Englisch und Deutsch gesammelt. Diese können gerne per Copy-und-Paste eingefügt und angepasst werden. Vielen Dank dafür.

Beispiel 1

> Die Ergebnisse/Teile der Ergebnisse, die in dieser Arbeit vorgestellt sind, wurden mithilfe des Clustersystems an der Leibniz Universität Hannover berechnet.
>
> The results presented here were (partially) carried out on the cluster system at the Leibniz University of Hannover, Germany.

Beispiel 2

> Diese Arbeit wurde vom Team des Clustersystems der Leibniz Universität Hannover unterstützt.
>
> We acknowledge the support of the cluster system team at the Leibniz University of Hannover, Germany in the production of this work.

Beispiel 3

> Diese Arbeit wurde unterstützt vom Compute-Cluster, welches von der Leibniz Universität Hannover, vom Niedersächsischen Ministerium für Wissenschaft und Kultur (MWK) und der Deutschen Forschungsgemeinschaft (DFG) getragen wird.
>
> This work was supported by the compute cluster, which is funded by the Leibniz Universität Hannover, the Lower Saxony Ministry of Science and Culture (MWK) and the German Research Association (DFG).

Das Ende Ihrer Arbeit

Wenn das Ende Ihrer Arbeit mit dem Clustersystem gekommen ist, würden wir uns freuen, wenn Sie ein paar Dinge erledigen, bevor Sie das Clustersystem verlassen.

1. Verzeichnisse aufräumen
    - $HOME aufräumen
    - $BIGWORK aufräumen
2. Nehmen Sie Kontakt mit uns auf, um einen Beitrag für den nächsten Clusterjahresbericht einzureichen
3. Ihr Projektleiter sollte Ihre Benutzerkennung löschen