

bwUniCluster Tutorial

Access, Data Transfer, Compiling, Modulefiles, Batch Jobs

Frauke Bösert, SCC, KIT





1

Material: Slides & Scripts

- https://indico.scc.kit.edu/indico/event/263/
- @bwUniCluster/ForHLR I/ForHLR II:

/pfs/data1/software_uc1/bwhpc/kit/workshop/2016-12-06

How to read the following slides

Abbreviation/Colour code	Full meaning
\$ command -option value	<pre>\$ = prompt of the interactive shell The full prompt may look like: user@machine:path\$ The command has been entered in the interactive shell session</pre>
<integer> <string></string></integer>	<> = Placeholder for integer, string etc
foo, bar	Metasyntactic variables



Login

<UserID>

username

prefix

- = <prefix_username>
- = your user account at your university.

= organization's token
KIT: ka → but NO ka_ for bwUniCluster/ForHLR I&II
Uni Stuttgart: st, Uni Hohenheim: ho,
HS Esslingen: es, Uni Freiburg: fr

Host <host>

- bwUniCluster: uc1.scc.kit.edu
- bwFileStorage: bwfilestorage-login.lsdf.kit.edu

Linux / macOS

3

- open terminal:
 - \$ ssh <UserID>@<host>

Windows

- use SSH-Client, e.g. PuTTY
- connect to <host>: Login as: <UserID>



Revision: basic linux commands

Cheat sheet: http://ryanstutorials.net/linuxtutorial/cheatsheet.php

\$ pwd	show path of working directory
<pre>\$ mkdir <dirname></dirname></pre>	make directory
<pre>\$ cd <dirname></dirname></pre>	change directory
\$ ls	list directory content
<pre>\$ cp <sourcefile> <targetfile></targetfile></sourcefile></pre>	copy file
<pre>\$ mv <sourcefile> <targetfile></targetfile></sourcefile></pre>	move file
<pre>\$ rm <filename></filename></pre>	remove file
<pre>\$ nano <filename></filename></pre>	edit file
\$ man <command/>	show command's manual



Exercise 1: Login+Linux

TASK/ToDo: 10 min

Login to bwUniCluster

- 1. Generate folder with today's date (use syntax YYYY-MM-DD) unter \$HOME
- 2. Generate empty file with name *moab1* + today's date Hint: use command *touch* For use of *touch* read manual via command *man touch*
- 3. Move file to \$HOME/<YYYY-MM-DD>; Rename file by adding the extension .sh
- Find under /opt/bwhpc/kit/workshop/2016-12-06 (*man find*) the PDF of the talk starting with the string 04_2016-12-06 ; Copy it to your folder generated under 1.
- 5. Go to folder \$HOME/<YYYY-MM-DD> and open the PDF with *evince* \rightarrow it will fail: find the option of the SSH command in the manual *man ssh* \rightarrow look at *"*Enables X11 forwarding"
- Exit login

5

Re-login to bwUniCluster with the SSH + new option and open PDF with evince

Data Transfer

From localhost to cluster:

- use scp (secure copy) or sftp (secure file transfer program)
- Read manual for options/syntax questions (man scp, man sftp)
- From Linux to Linux Systems you can also use rsync
- Linux / OS X
 - Open terminal at your computer:
 - \$ scp <sourcefile> <username>@<host>:<targetfile>
 - or
 - \$ sftp <username>@<host>:<targetdir>
 - \$ put <sourcefile>

Windows

06/12/2016

- use SCP/SFTP-Client, e.g. WinSCP
- connect to <username>@<host>
- copy data by drag&drop mechanism

From \$HOME to Workspaces

- Compute nodes read&write in workspaces much faster than in \$HOME directory.
- **DO NOT COMPUTE IN \$HOME !!**
- SHOME:
 - For:
 - Source code
- Workspaces:
 - HowTo: compare previous talk slide no 9
 - For:

9

- Program input (e.g. initial and boundary conditions)
- Program output



Exercise 2: Data Transfer

TASK/ToDo: 10 min

- 1. Download 04_2016-12-06_bwHPC_course_suppl.tar.gz from https://indico.scc.kit.edu/indico/event/263/material/0/ to your course computer
- 2.Copy *tar.gz to \$HOME/<YYYY-MM-DD> at bwUniCluster
- 3. Login to bwUniCluster
- 4. Generate workspace with lifetime of 10 days and name "workshop"
- 5. Copy *tar.gz to your workspace
- 6. Unpack the archive *tar.gz in your workspace Hint: for unpack use command + options: *tar xvBfz*

Module Environment

Users require different software in different versions.

Software is installed and can be used by loading corresponding modules.

\$ module	avail	show all installed software packages
\$ module	avail compiler	show all available compilers
\$ module	<pre>load <modulepath></modulepath></pre>	load a module in list
\$ module	unload <modulepath></modulepath>	remove a module from list
\$ module	list	show all loaded modules
\$ module	<pre>show <modulepath></modulepath></pre>	show environment variables of module
\$ module	help <modulepath></modulepath>	show usage information of module



Exercise 3: Editors/Create script

TASK/ToDo: 10 min

1. Choose:

- a) vim → cheat sheet: http://vim.rtorr.com/
- b) nano → cheat sheet: http://alturl.com/wbtrk
- C) emacs → login to bwUniCluster via X-forwarding, i.e., ssh -X cheat sheet: http://www.rgrjr.com/emacs/emacs_cheat.html
- 2. Insert in file moab1_2016-12-06.sh, save it, print content to screen

#!/bin/bash
which icc
module load compiler/intel
which icc

- 3. Change file permission to make moab1_2016-12-06.sh executable
- 4. Run script



Environment variables in job scripts

Details: http://www.bwhpc-c5.de/wiki/index.php/Batch_Jobs#Environment_Variables_for_Batch_Jobs

MOAB variables and own environment variables

	Using MOAB Variables	Defining own variables
Header	#MSUB -o \$(JOBNAME).o\$(JOBID)	#MSUB -v EXEC=./hello
Execution Part	<pre>echo "Job \${MOAB_JOBNAME} is running (ID=\${MOAB_JOBID})"</pre>	export EXEC=./hello



Keep track of a job

Submit job script

- \$ msub <jobscript>
- If a job (script) is accepted the <jobid> appears at screen.

\$ checkjob <jobid></jobid>	show job details
\$ showq	list all my running, idling and blocked jobs by <jobid></jobid>
\$ showq -n	list all my running, idling and blocked jobs by <jobname></jobname>
\$ showq -c	list my completed jobs
\$ <pre>canceljob <jobid></jobid></pre>	cancel job



Exercise 4: Batch job

TASK/ToDo: 10 min

17

- 1. Copy moab1_2016-12-06.sh to the subfolder in your workspace containing hello.c
- 2. Go to your workspace; load intel compiler
- 3. Compile existing code and execute hello.x Hint for compiling: \$ icc -o hello.x hello.c
- 4. Replace all in moab1_2016-12-06.sh with the following lines & save it

```
#!/bin/bash
#MSUB -l nodes=1:ppn=1,walltime=00:01:00
#MSUB -l mem=100mb
#MSUB -l advres=workshop.8
#MSUB -N firstjob
./hello.x
sleep 10
```

5. Submit script to queueing system + watch progress

