R in an HPC environment

Introduction to Kebnekaise

Birgitte Brydsø

HPC2N, Umeå University

14-15 December 2022

UMEÅ UNIVERSITET

Using Kebnekaise

Connecting to HPC2N's systems, thinlinc

- Download the client and install it:
  https://www.cendio.com/thinlinc/download
  - Alternately, run ThinLinc directly in your browser. See link under "More information"
- Start the client. Enter the servername:
  kebnekaise-tl.hpc2n.umu.se.
  Enter your username under "Username".
- Go to "Options" -> "Security" and check that authentication method is set to password.
- Go to "Options" -> "Screen". Uncheck "Full screen mode".
- Enter your HPC2N password. Click "Connect"
- More information here:
  https://www.hpc2n.umu.se/documentation/guides/thinlinc

Using Kebnekaise

Connecting to HPC2N's systems, other

- **Linux, OS X:**
  - `ssh username@kebnekaise.hpc2n.umu.se`
  - Use `ssh -Y ....` if you want to open graphical displays.
- **Windows:**
  - Get SSH client (MobaXterm, PuTTY, Cygwin ...)
  - Get X11 server if you need graphical displays (Xming, ...)
  - Start the client and login with your HPC2N username to

    `kebnekaise.hpc2n.umu.se`
  - More information here:

    https://www.hpc2n.umu.se/documentation/guides/windows-connection
- **Mac/OSX:** Guide here:

  https://www.hpc2n.umu.se/documentation/guides/mac-connection

Editing your files

- Various editors: vi, vim, nano, emacs ...
- Example, vi/vim:
  - `vi <filename>`
  - Insert before: i
  - Save and exit vi/vim: `Esc :wq`
- Example, nano:
  - `nano <filename>`
  - Save and exit nano: `Ctrl-x`
- Example, Emacs:
  - Start with: emacs
  - Open (or create) file: Ctrl-x Ctrl-f
  - Save: Ctrl-x Ctrl-s
  - Exit Emacs: Ctrl-x Ctrl-c

The File System

More info here: http://www.hpc2n.umu.se/filesystems/overview

	Project storage	**$HOME**	**/scratch**
Recommended for batch jobs	Yes	No (size)	Yes
Backed up	No	Yes	No
Accessible by batch system	Yes	Yes	Yes (node only)
Performance	High	High	Medium
Default readability	Group only	Owner	Owner
Permissions management	chmod, chgrp, ACL	chmod, chgrp, ACL	N/A for batchjobs
Notes	Storage your group get allocated through the storage projects	Your home-directory	Per node

R on Kebnekaise at HPC2N

The Module Environment

Most programs are accessed by first loading them as a 'module'

Modules are:

- used to set up your environment (paths to executables, libraries, etc.) for a particular (set of) software package(s)
- for helping users manage their Unix/Linux shell environment, allowing groups of related environment-variable settings to be made or removed dynamically
- allows having multiple versions of a program or package available by just loading the proper module
- installed in a hierarchial layout; thus some modules are only available after loading a specific compiler and/or MPI version

Compiler toolchains load software-bundles for a complete environment (compiling, using prebuilt software). Includes some/all of: compiler suite, MPI, BLAS, LAPACK, ScaLapack, FFTW, CUDA.

- **foss**: GCC, OpenMPI, OpenBLAS/LAPACK, FFTW, ScaLAPACK
- **intel**: icc, ifort, IntelMPI, IntelMKL

Check existing modules (versions): `ml spider R`

- We recommend R 4.0.4 on Kebnekaise
- Checking how to load this and any prerequisites:
  `ml spider R/4.0.4`
- Loading:
  `ml GCC/10.2.0 OpenMPI/4.0.5 R/4.0.4`
- Loading (with CUDA support):
  `ml GCC/10.2.0 CUDA/11.1.1 OpenMPI/4.0.5 R/4.0.4`
- You start R by typing `R` on the command line.

- parallel
- Rmpi
- foreach
- doparallel
- cluster
- knor (now called clusternor)
- boot

Most of these packages are available as extensions to the R module on Kebnekaise. They are thus loaded together with the R module.

R on Kebnekaise at HPC2N

R packages needed for this course - getting access

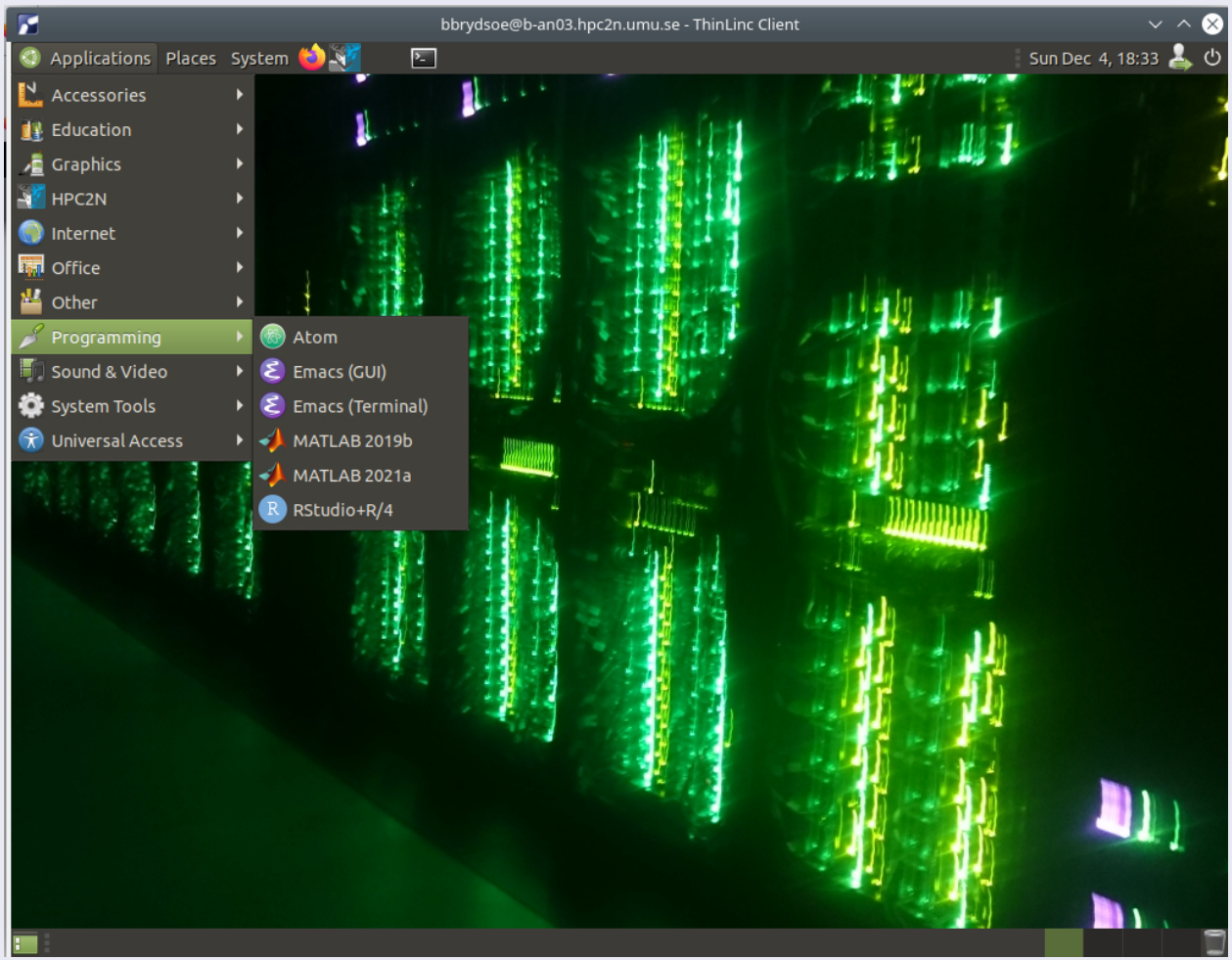
- To see which packages are available, start R and then enter these four lines:
  - `ip <- as.data.frame(installed.packages()[,c(1,3:4)])`
  - `rownames(ip) <- NULL`
  - `ip <- ip[is.na(ip$Priority),1:2,drop=FALSE]`
  - `print(ip, row.names=FALSE)`

- Two modules are **not** available and needs to be installed by you:
  - parallel
  - knor (clusternor)
- Installing R packages on Kebnekaise is explained here: https://www.hpc2n.umu.se/resources/software/user_installed/r

R on Kebnekaise at HPC2N

Rstudio

To use Rstudio on Kebnekaise, you need to connect using ThinLinc as it is not installed on the regular login nodes. It matches R 4.0.4.



- Large/long/parallel jobs **must** be run through the batch system
- SLURM is an Open Source job scheduler. It provides three key functions
  - Keeps track of available system resources
  - Enforces local system resource usage and job scheduling policies
  - Manages a job queue, distributing work across resources according to policies
- To run a batch job, you need a SLURM submit file (batch submit file, batch script, job script ...)
- When submitting jobs to the batch system, you **must** use the course project!
- Guides and documentation at: http://www.hpc2n.umu.se/support

- Submit job: `sbatch <jobscript.sh>` (successful submission returns a job-id number)
- As default, output/errors are found in `slurm-<job-id>.out`
- Get list of all jobs: `squeue`
- Get list of only your jobs: `squeue -u <username>`
- Adding the flag `--start` to `squeue` gives the estimated job start time. This can change depending on other people's jobs.
- Check on a specific job: `scontrol show job <job id>`
- Delete a specific job: `scancel <job id>`
- Delete all your jobs: `scancel -u <username>`

R on Kebnekaise at HPC2N

SLURM batch script for a serial R job

```
#!/bin/bash
#SBATCH -A SNIC2022-22-1012 #Project id
#SBATCH -J my-serial-R-job #Name of job
#SBATCH --time=00:10:00 #Jobtime (HH:MM:SS) Max:  168H
#SBATCH -o Rjob_%j.out #Naming the output file
#SBATCH -e Rjob_%j.err #Naming the error file
#SBATCH -n 1 #Number of tasks.

ml purge < /dev/null 2>&1
ml GCC/10.2.0 OpenMPI/4.0.5 R/4.0.4

R --no-save --quiet < input.R > Rexample.out
```

Submit the above script with `sbatch myRjob.sh`

NOTE that you need to load the Rmpi library within your R script for this to work

```
#!/bin/bash
#SBATCH -A SNIC2022-22-1012
#SBATCH -n 8
#SBATCH --time=00:30:00

ml purge < /dev/null 2>&1
ml GCC/10.2.0 OpenMPI/4.0.5 R/4.0.4

mpirun R -q -f <program>.R
```

R on Kebnekaise at HPC2N

SLURM batch script for a parallel R job, using doParallel

Assume we have a small R program, "doParallel.R":

```
library(doParallel)
cl <- makeCluster(4)
registerDoParallel(cl)

# code that we want executed in parallel

stopCluster(cl)
```

Batch script to submit the above R program:

```
#!/bin/bash
#SBATCH -A SNIC2022-22-1012
#SBATCH -t 00:10:00
#SBATCH -N 1
#SBATCH -c 4

ml purge < /dev/null 2>&1
ml GCC/10.2.0 OpenMPI/4.0.5 R/4.0.4

R -q --slave -f doParallel.R
```

Various useful info

- A project has been set up for the workshop:
  `SNIC2022-22-1012`
- You use it in your batch submit file by adding:

  ```
  #SBATCH -A SNIC2022-22-1012
  ```
- Kebnekaise ThinLinc login node:
  `kebnekaise-tl.hpc2n.umu.se`
- Kebnekaise "regular" login node:
  `kebnekaise.hpc2n.umu.se`