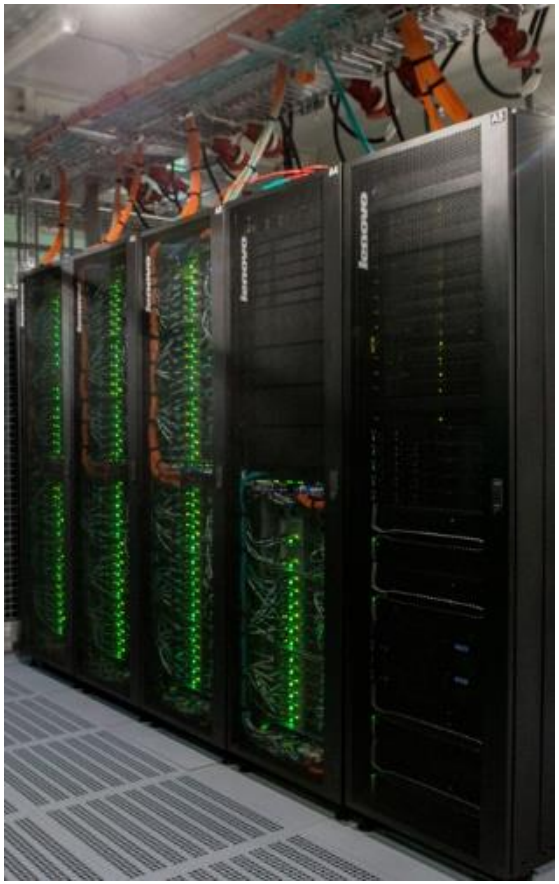
Using R on the HPC Cluster CARL

CARL and EDDY

- close to 600 compute nodes, CARL has
  - 158/128 standard/low-memory nodes (24 cores, 2.2GHz 128/256GB RAM)
  - 30 big nodes (16 cores, 3.2 GHz, 512GB RAM)
  - 2 PP nodes (40 cores, 2,8 GHz, 2TB RAM)
  - 9 GPU nodes (24 cores, 2.2GHz 256GB RAM, P100 GPU)
- FDR Infiniband Interconnect
- 900 TB GPFS storage
- few 100TB NFS storage
- peak performance 460 Tflop/s
- funded by DFG, MWK and BMWi (see Acknowledging HPC Cluster)



https://wiki.hpcuser.uni-oldenburg.de/index.php?title=HPC_Facilities_of_the_University_of_Oldenburg_2016

Login to the HPC Cluster

http://wiki.hpcuser.uni-oldenburg.de/index.php?title=Login

- Linux
  - use ssh as before with **carl** or **eddy** as login nodes

```
ssh -X abcd1234@carl.hpc.uni-oldenburg.de
```

- Windows
  - use MobaXterm (recommended) or PuTTY
  - with Windows 10 you can also use Windows Subsystem for Linux (WSL, see https://docs.microsoft.com/en-us/windows/wsl/install-win10)

- login host names

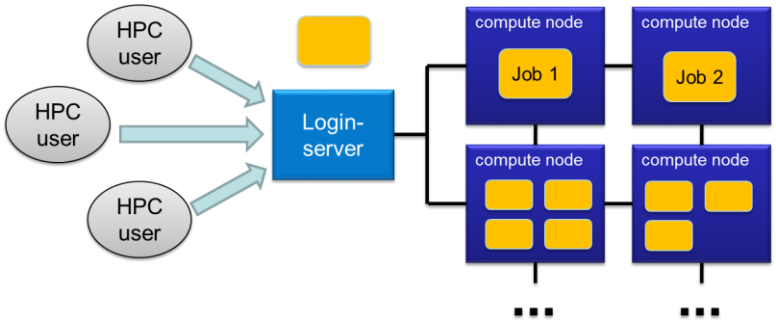
```
hpcl00[1-4].hpc.uni-oldenburg.de
```

  - can be used instead of **carl** or **eddy** (for login to specific node)
  - no difference between **carl** and **eddy** as login

- from outside of the campus network use VPN connection
  - see instructions at http://www.itdienste.uni-oldenburg.de/21240.html

Basic HPC Cluster Usage

- users can login to the head node only
- jobs are defined in form of a **bash** script
- job schedulers provides functionality to submit and manage jobs
- scheduler also takes care of fair sharing and efficient usage of the resources




```
$ cat HelloWorld_v2.job
#!/bin/bash

####### SLURM options begin

### general settings
#SBATCH --partition=carl.p
#SBATCH --job-name=HelloWorld
#SBATCH --output=HelloWorld.o%j

### requested resources
#SBATCH --time=0:10:00      # max runtime
#SBATCH --mem=1G            # max memory

####### SLURM options end

# execute these commands
sleep 10
echo "Hello World from `hostname`"
```

HelloWorld_v2.sge

special comments allow to specify settings and requirements for a job

commands in the script define the job the user wants to be executed

```
$ cat HelloWorld_v2.job
#!/bin/bash

####### SLURM options begin

### general settings
#SBATCH --partition=carl.p
#SBATCH --job-name=HelloWorld
#SBATCH --output=HelloWorld.o%j

### requested resources
#SBATCH --time=0:10:00      # max runtime
#SBATCH --mem=1G            # max memory

####### SLURM options end

# execute these commands
sleep 10
echo "Hello World from `hostname`"
```

Important SLURM Commands

Command	Used for
`sinfo`	information about SLURM nodes and partitions
`squeue`	overview of jobs in the scheduler queue
`sacct`	accounting information about jobs
`sbatch`	submit jobs to the scheduler
`srun`	allocate resources if needed and launch a job step within an job allocation
`scancel`	delete queued or running jobs
`scontrol`	manage jobs (limited) and more

to get information about commands visit https://slurm.schedmd.com/documentation.html or use

```
$ man <command>
```

Options for SBATCH

https://slurm.schedmd.com/sbatch.html

Option	Short Form	Description
`--job-name=JobName`	`-J JobName`	sets a name for job which is display in the queue
`--partion=<partition>`	`-p <partition>`	(comma-separated list of) partition(s) where the job should run, no default
`--output=<filename>` `--error=<filename>`	`-o <filename>` `-e <filename>`	output files for STDOUT and STDERR, default is join in slurm-%j.out
`--ntasks=<n>`	`-n <n>`	number of tasks (e.g. for MPI parallel jobs)
`--cpus-per-task=<n>`	`-c <n>`	CPU cores per task (parallel threads)
`--mem-per-cpu=<m>`		memory per core/task, optional
`--mem=<m>`		memory per node, exclusive with above
`--mail-type=<MT>` `--mail-user=...`		mail settings

File Systems

File System	Env. Variable	Path	Used for
Home	`$HOME`	`/user/abcd1234`	critical data that cannot easily be reproduced (program codes, initial conditions, results from data analysis)
Data	`$DATA`	`/nfs/data/abcd1234`	important data from simulations for on-going analysis and mid term (project duration) storage
Work	`$WORK`	`/gss/work/abcd1234`	data storage for simulation runtime, pre- and post-processing, short term (weeks) storage
Scratch	`$TMPDIR`	`/scratch/<job-dir>`	temporary data storage during job runtime
Offsite	`$OFFSITE`	`/nfs/offsite/user/abcd1234`	long term storage for inactive data, only available on login nodes

- HOME, DATA and OFFSITE have backup for disaster recovery and daily snapshots for file recovery
- special quota rule for WORK

Software and Modules

- software is installed centrally on the cluster
  - /cm/shared/uniol/software
  - user can use preinstalled software
  - software can be optimized for system
  - own software can be installed too

- installed software includes
  - compilers
  - libraries (MPI, numerical libraries,…)
  - scientific application
  - overview and help in the HPC wiki

Module Commands

https://wiki.hpcuser.uni-oldenburg.de/index.php?title=User_environment_-_The_usage_of_module_2016

- find modules

```
module available [module-name]
module spider [module-name]
```

  - list all modules [with given module name]
  - spider is case-insensitive and understands reg-exp

- load/unload

```
module load <module-name>
module remove <module-name>
```

  - to return to a default state

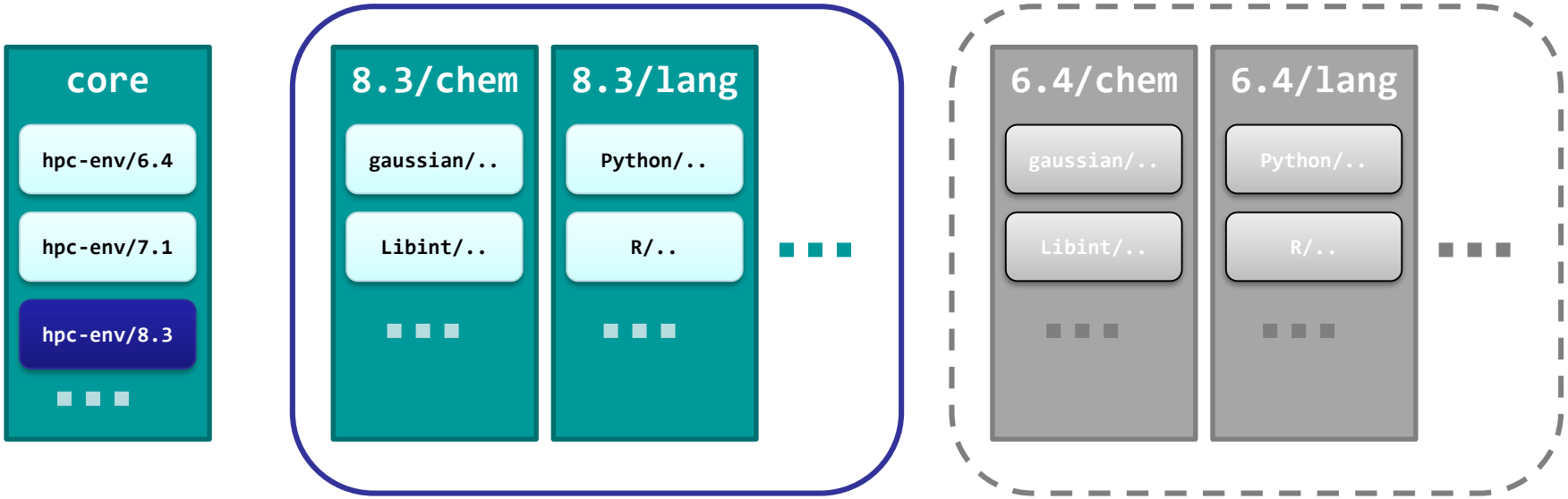
```
module restore
```

- information about modules

```
module list
module help <module-name>
module spider <module-name>
```

Module Hierarchy

- modules are organized in categories (e.g. **`chem`**, **`lang`**, …)
- category core includes special modules hpc-env (environments)
  - module in other categories only become visible if environment is loaded
  - different modules/versions are available in different environments



Finding Installed Versions of R

- the commands **`module spider`** and **`module available`** can be used to list available **`R`** modules
  - the single letter makes the search a bit difficult but regular expressions can help

```
[abcd1234@carl]$ module -r spider ^R/      # finds modules starting with R/

----------------------------------------------------------------------------------------------------------------------
  R:
----------------------------------------------------------------------------------------------------------------------
    Description:
      R is a free software environment for statistical computing and graphics. - Homepage:
      http://www.r-project.org/

     Versions:
        R/3.3.1
        R/3.4.4-intel-2018a
        R/3.5.2-intel-2018a
        R/3.6.1-intel-2018a
        R/4.0.2-foss-2019b

----------------------------------------------------------------------------------------------------------------------
  For detailed information about a specific "R" module (including how to load the modules) use the module's full name.
  For example:

     $ module spider R/3.3.1
----------------------------------------------------------------------------------------------------------------------
```

Finding Installed Versions of R

- using **`module spider <name>/<version>`** gives details
  - including which environment has to be loaded first

```
[abcd1234@carl]$ module spider R/4.0.2-foss-2019b

----------------------------------------------------------------------------------------------------
  R: R/4.0.2-foss-2019b
----------------------------------------------------------------------------------------------------
    Description:
      R is a free software environment for statistical computing and graphics.


    You will need to load all module(s) on any one of the lines below before the "R/4.0.2-foss-2019b" module is availab
e to load.

      hpc-env/8.3

    Help:

      Description
      ===========
      R is a free software environment for statistical computing
       and graphics.
```

Finding Installed Versions of R

- the command **`module available`** only lists modules that can be loaded (because the required environment is loaded)
  - with **`module spider`** all modules are found

```
[abcd1234@carl]$ module load hpc-env/8.3
[abcd1234@carl]$ module -r av ^R/

------------------------------ /cm/shared/uniol/modules/8.3/lang -------------------------------
   R/4.0.2-foss-2019b

Use "module spider" to find all possible modules.
Use "module keyword key1 key2 ..." to search for all possible modules matching any of the "keys".
```

Using R (Login Node/Interactive)

- on the cluster the R command-line can be used
  - first load the R module
  - can be used for testing or small calculations on the login node

```
[abcd1234@carl]$ module load R/4.0.2-foss-2019b
[abcd1234@carl]$ R                    # starting the R command-line

R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> q()
Save workspace image? [y/n/c]: n
[abcd1234@carl]$
```

Using R (Login Node/Script)

- alternatively, execute a script
  - script needs to be executable, set permissions with **`chmod`**

```
[abcd1234@carl]$ cat HelloWorld.R
#!/usr/bin/env Rscript

# print a message
message <- "Hello World!"
print(message)

# get a command-line argument
args <- commandArgs(trailingOnly=TRUE)
if (length(args)==1) {
   print(paste("Hello", args[1]))
}

# get information from the environment
jid <- Sys.getenv("SLURM_JOB_ID")
if (jid != "") {
   print(paste("My job id is", jid))
} else {
   print("I do not have a job id")
}
[abcd1234@carl]$ chmod u+x HelloWorld.R
```

shebang for calling Rscript

Using R (Login Node/Script)

- alternatively, execute a script
  - script demonstrate the use of command-line arguments and how to get information from the environment

```
[abcd1234@carl]$ ./HelloWorld.R $USER
[1] "Hello World!"
[1] "Hello abcd1234"
[1] "I do not have a job id"
[abcd1234@carl]$
```

  - there are other ways of executing a script with subtle differences (e.g. whether or not the code is printed in addition to output)

R Extensions

- the module R includes a large number of extensions
  - the command **`module help R`** will show which ones

```
[abcd1234@carl]$ module help R

---------------------------------- Module Specific Help for "R/4.0.2-foss-2019b" ---------

Description
===========
R is a free software environment for statistical computing
 and graphics.


More information
================
 - Homepage: https://www.r-project.org/


Included extensions
===================
abc-2.1, abc.data-1.0, abe-3.0.1, abind-1.4-5, acepack-1.4.1, adabag-4.2,
ade4-1.7-15, ADGofTest-0.3, aggregation-1.0.1, akima-0.6-2, AlgDesign-1.2.0,
animation-2.6, aod-1.3.1, ape-5.3, argparse-2.0.1, arm-1.11-1, askpass-1.1,
asnipe-1.1.12, assertthat-0.2.1, AUC-0.3.0, audio-0.1-7, b-a, backports-1.1.6,
bacr-1.0.1, bartMachine-1.2.4.2, bartMachineJARs-1.1, base64-2.0,
base64enc-0.1-3, BatchJobs-1.8, BayesianTools-0.1.7, bayesm-3.1-4,
BayesPen-1.0, bayesplot-1.7.1, BB-2019.10-1, BBmisc-1.11, bbmle-1.0.23.1,
BCEE-1.3.0, BDgraph-2.62, bdsmatrix-1.3-4, beanplot-1.2, beeswarm-0.2.3,
```

R Extensions

- the module R includes a large number of extensions
  - the command **`module help R`** will show which ones
- additional extensions can be installed
  - by the user, requires a directory and the file **`$HOME/.Renviron`**

```
[abcd1234@carl]$ mkdir -p $HOME/R/lib/
[abcd1234@carl]$ cat $HOME/.Renviron
R_LIBS_USER="~/R/lib"
[abcd1234@carl]$
```

R Extensions

- the module R includes a large number of extensions
  - the command **`module help R`** will show which ones
- additional extensions can be installed
  - by the user, requires a directory and the file **`$HOME/.Renviron`**
  - execute the command **`install.package("quanteda")`** in R which also installs dependencies
  - might fail when additional software is needed, e.g. for PDFTools you need poppler
    - if a module is available it might help to load the module
  - alternatively send an installation request to hpcsupport@uol.de

Using R (Compute Nodes)

- in order to use the compute nodes a job script is needed
  - alternatively, an interactive session can be started using **`srun`**
  - example job script **`HelloWorld_Rjob.sh`**

```
#!/bin/bash

# Slurm settings
#SBATCH --partition carl.p
#SBATCH --ntasks 1

# modules
module load hpc-env/8.3
module load R/4.0.2-foss-2019b

# executing script
echo "Job started on $(date)"
./HelloWorld.R
status=$?
echo "Job finished on $(date) with status $status"
exit $status
```

Using R (Compute Nodes)

- in order to use the compute nodes a job script is needed
  - job is submitted with the command **`sbatch <jobscript>`**
  - returns a jobid which can be used e.g. to get job info from **`sacct`**
  - output is written to **`slurm-<jobid>.out`** (unless otherwise specified)

```
[abcd1234@carl]$ sbatch HelloWorld_Rjob.sh
Submitted batch job 21787740
[abcd1234@carl]$ cat slurm-21787740.out
Job started on Thu Jul  9 10:14:35 CEST 2020
[1] "Hello World!"
[1] "My job id is 21787740"
Job finished on Thu Jul  9 10:14:35 CEST 2020 with status 0
```

Parallelization with R or Job Skripts

- there are different parallelizations possible for R
  - using e.g. Rmpi to parallelize the R script
  - using thread-parallel libraries in R
  - by submitting a job array for a task-parallel problem

```
[abcd1234@carl]$ sbatch HelloWorld_Rjob_array.sh
Submitted batch job 21787975
[abcd1234@carl]$ squeue -u $USER
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
       21787975_1    carl.p HelloWor lees4820  R       0:01      1 mpcs093
       21787975_2    carl.p HelloWor lees4820  R       0:01      1 mpcs093
       21787975_3    carl.p HelloWor lees4820  R       0:01      1 mpcs093
       21787975_4    carl.p HelloWor lees4820  R       0:01      1 mpcl079
[abcd1234@carl]$ cat slurm-21787975_1.out

The following have been reloaded with a version change:
  1) LibTIFF/4.0.10-GCCcore-8.3.0 => LibTIFF/4.1.0-GCCcore-8.3.0

Job started on Thu Jul  9 11:10:44 CEST 2020
Executing task 1
[1] "Hello World!"
[1] "Hello Esmerelda Weatherwax"
[1] "My job id is 21787976 and my task id is 1"
Job finished on Thu Jul  9 11:10:44 CEST 2020 with status 0
```

Using a CSV File

- to work with a CSV file you need to copy the file to the cluster
  - e.g. go to **$WORK** and create a subdir for R (likewise for **$HOME**)

```
[abcd1234@carl]$ cd $WORK
[abcd1234@carl]$ mkdir -p R
```

  - on Linux use **scp**, on Windows MobaXterm (drag‘n‘drop) to copy file to or from cluster
  - CSV files from a Windows system may have **CRLF** line terminators, can be changed with **dos2unix**

```
[abcd1234@carl]$ file data.csv
data.csv: ASCII text, with CRLF line terminators
[abcd1234@carl]$ dos2unix data.csv
dos2unix: converting file data.csv to Unix format ...
[abcd1234@carl]$ file data.csv
data.csv: ASCII text
```

Using a CSV File

- to work with a CSV file you need to copy the file to the cluster
  - CSV files from a German computer may use a decimal comma instead of a decimal point (R would read numbers as strings)
  - problem can be solve with the **sed** command

```
[abcd1234@carl]$ head data.csv
x;y
0,288487818;0,092591244
0,058707592;0,260476579
0,148140717;0,226443709
0,112170217;0,189693498
[abcd1234@carl]$ sed -i "s/,/./g" data.csv
[abcd1234@carl]$ head -n 2 data.csv
x;y
0.288487818;0.092591244
```

Reading and Writing Files

- files can be read and written as usual in R
  - working directory is where R was started unless changed with e.g. **`setwd()`**
  - for jobs, the working directory is the directory where the job was started from (in which **`sbatch`** was issued)
  - recommended directory for file I/O is **`$WORK`**, **`$HOME`** or **`$DATA`** should not be used (see HPC Wiki on File I/O)
  - if you have very high I/O demands (especially random I/O) you can also use **`$TMPDIR`**, but remember to copy needed files back to **`$WORK`** before end of job

Reading and Writing Files

- files can be read and written as usual in R
  - example **`kmeans.R`** reads and CSV from the same directory and write a PDF file

```
#!/usr/bin/env Rscript

# read data from csv file (if needed add setwd() call before)
data <- read.csv(file="data.csv", sep=";")

# do a kmeans analysis to find 3 clusters
cl <- kmeans(data, 3)

# write a plot to a PDF file
require(graphics)
pdf("kmeans_plot.pdf")
plot(data, col=cl$cluster)
points(cl$centers, col = 1:3, pch = 8, cex = 2)
```