

# Biostat M280

## Cluster Computing at UCLA

Raffaella D'Auria, PhD

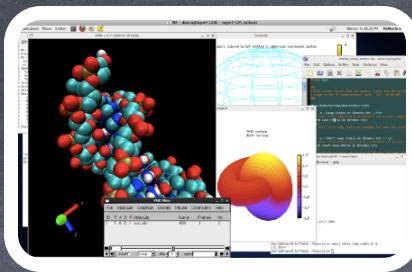
## Learning outcomes:

- ▷ what is the Hoffman2 cluster
- ▷ getting an account on Hoffman2
- ▷ connecting to the Hoffman2 cluster w/ X11-forwarding enabled (from different OSs)
- ▷ copying your data to and from Hoffman2
- ▷ unix command line survival quick & dirty guide
- ▷ getting an interactive session
- ▷ how to find out what applications are on Hoffman2?
- ▷ loading/unloading applications in your environment
- ▷ run a batch job
- ▷ using R on Hoffman2

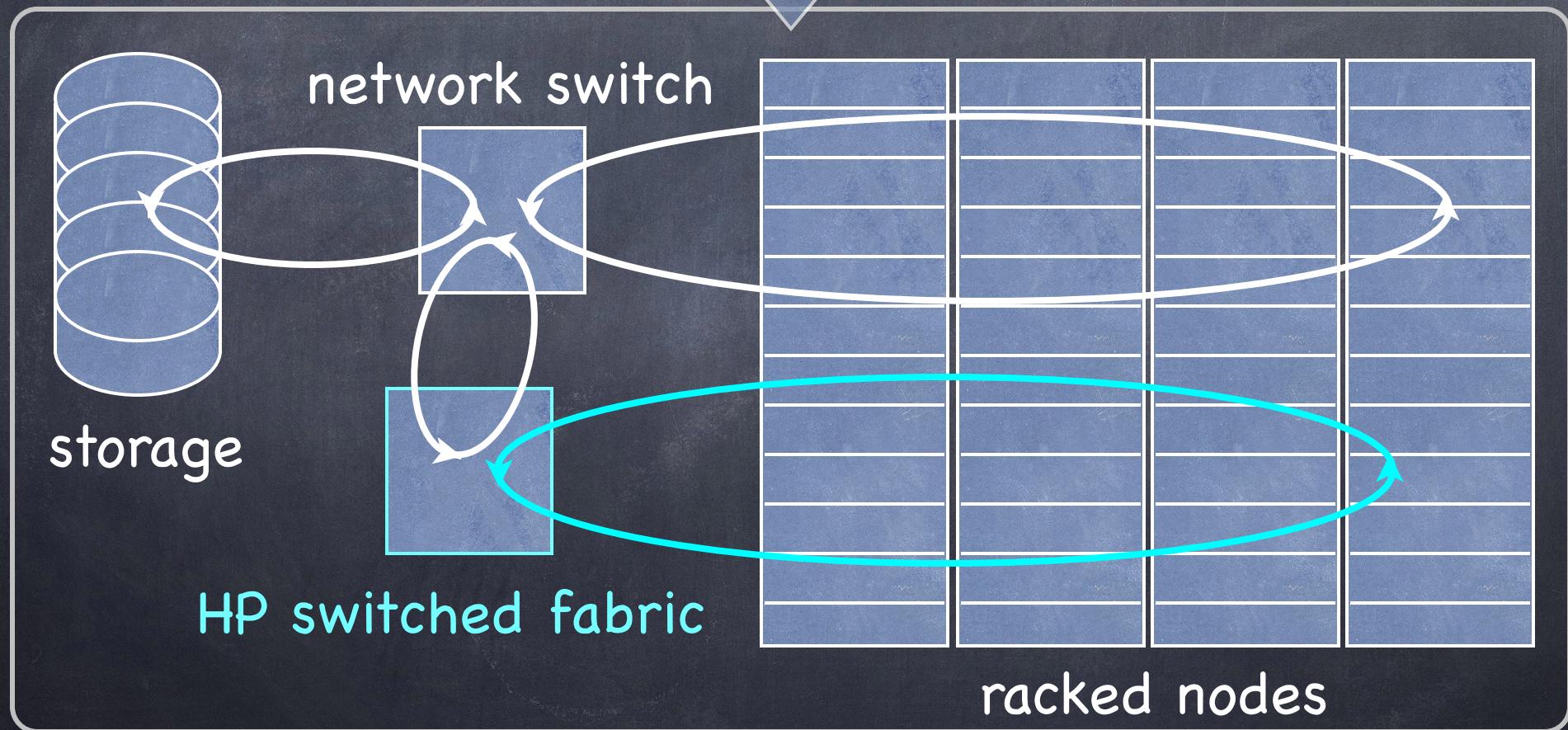


# The Hoffman2 cluster

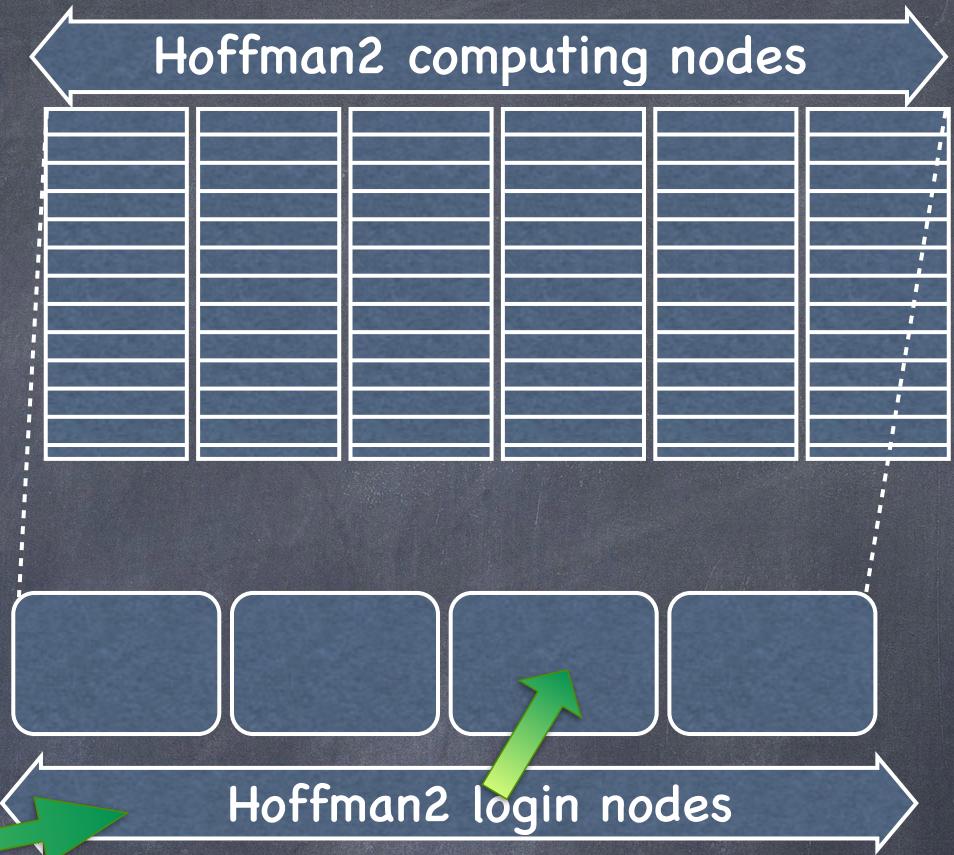
User running  
applications



Hoffman2 cluster



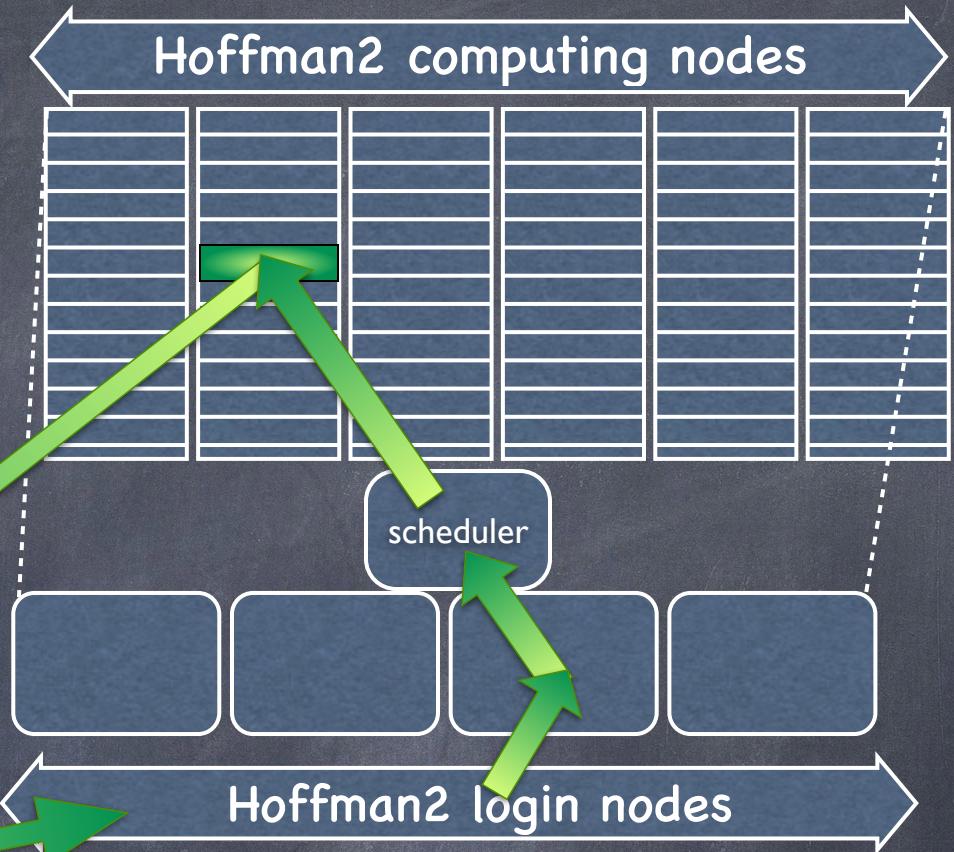
# Using the Hoffman2 cluster: starting an interactive session



# Using the Hoffman2 cluster: starting an interactive session



```
[rdtest@login2 ~]$ qrsh -l h_rt=2:00:00,h_data=4g -pe shared 2  
[rdtest@n9860 ~]$
```



Q?: Why is the command "qrsh" understood by the unix-shell?

A?: b/c it is in your \$PATH

# How to learn more about qrsh...

The screenshot shows a web browser displaying the Hoffman2 Cluster User Guide. The URL in the address bar is <https://www.hoffman2.idre.ucla.edu>. The page features a dark header with the UCLA and idre logos. A navigation bar at the top includes links for ACCESS, COMPUTING (which is highlighted with a green oval), DATA STORAGE, FILE TRANSFER, and SOFTWARE. On the left, a sidebar lists various user support links: About, User Support, FAQ, Getting started, News, Security policy, Status, How to acknowledge the Hoffman2 cluster program, and How to use this site. The main content area shows the "Home" page with a welcome message, recent announcements about free classes and Stata 15, and a search bar.

Report Typos and Errors

UCLA

Hoffman2 Cluster User Guide

Search this website

ACCESS COMPUTING DATA STORAGE FILE TRANSFER SOFTWARE

You are here: Home

## Home

Welcome to the Hoffman2 Cluster User Guide. To look up usage information, click on one of the navigation tabs above or the sidebar items on the left.

### Recent Announcements

**Free Classes** IDRE Research Technology Group is pleased to present Fall Quarter classes. There is no charge or fee to attend these classes. See [IDRE Fall 2017 HPC Classes](#)

**Stata 15** A new version of Stata has been installed. See [Stata 15 available](#) for further information.

About

User Support

FAQ

Getting started

News

Security policy

Status

How to acknowledge the Hoffman2 cluster program

How to use this site

# How to learn more about qrsh...

The screenshot shows a web browser displaying the Hoffman2 Cluster User Guide. The URL in the address bar is <https://www.hoffman2.idre.ucla.edu/computing/>. The page content includes a navigation menu with links like ACCESS, COMPUTING, DATA STORAGE, FILE TRANSFER, and SOFTWARE. On the left, there's a sidebar with links for About, User Support, FAQ, Getting started, News, Security policy, Status, How to acknowledge the Hoffman2 cluster program, and How to use this site. The main content area shows the 'Computing' page. Under 'General information', there's a list with 'Interactive jobs' highlighted by a green circle. Under 'Interactive jobs', there's a link 'How to Get an Interactive Session' also highlighted by a green circle.

https://www.hoffman2.idre.ucla.edu/computing/

120% C ☆ Search Report Typos and Errors

idre UCLA Hoffman2 Cluster User Guide

Search this website

ACCESS COMPUTING DATA STORAGE FILE TRANSFER SOFTWARE

You are here: Home / Computing

## Computing

Contents [hide]

General information

Interactive jobs (highlighted)

Batch Jobs

GPU Computing

Additional Information

How to ...

### General information

- Job Queues and Policy

### Interactive jobs

- How to Get an Interactive Session (highlighted)

### Batch Jobs

- Submitting a Job
- Commonly-Used UGE Commands

Apps on H2: intro R. D'Auria, PhD

# How to learn more about qrsh...

The screenshot shows a web browser window with a green oval highlighting the URL bar. The URL is <https://www.hoffman2.idre.ucla.edu/computing/interactive-session/>. The page content is from the idre UCLA Hoffman2 Cluster User Guide.

**Page Headers:**

- Page title: Hoffman2 Cluster User Guide
- Search bar: Search this website
- Report Typo and Errors link

**Page Navigation:**

- ACCESS
- COMPUTING
- DATA STORAGE
- FILE TRANSFER
- SOFTWARE

**Left Sidebar (About section):**

- About
- User Support
- FAQ
- Getting started
- News
- Security policy
- Status
- How to acknowledge the Hoffman2 cluster program
- How to use this site

**Page Content:**

You are here: Home / Computing / Getting an interactive session

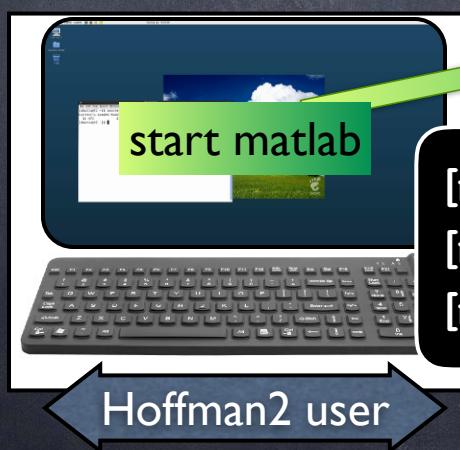
## Getting an interactive session

Normally, you would submit your jobs to the scheduler to be executed on the compute nodes (using the "qsub" command). However, at times it is convenient to do certain things interactively. For this purpose, some compute nodes are made available exclusively (certain limitations apply; see below) for interactive use via the "qrsh" command.

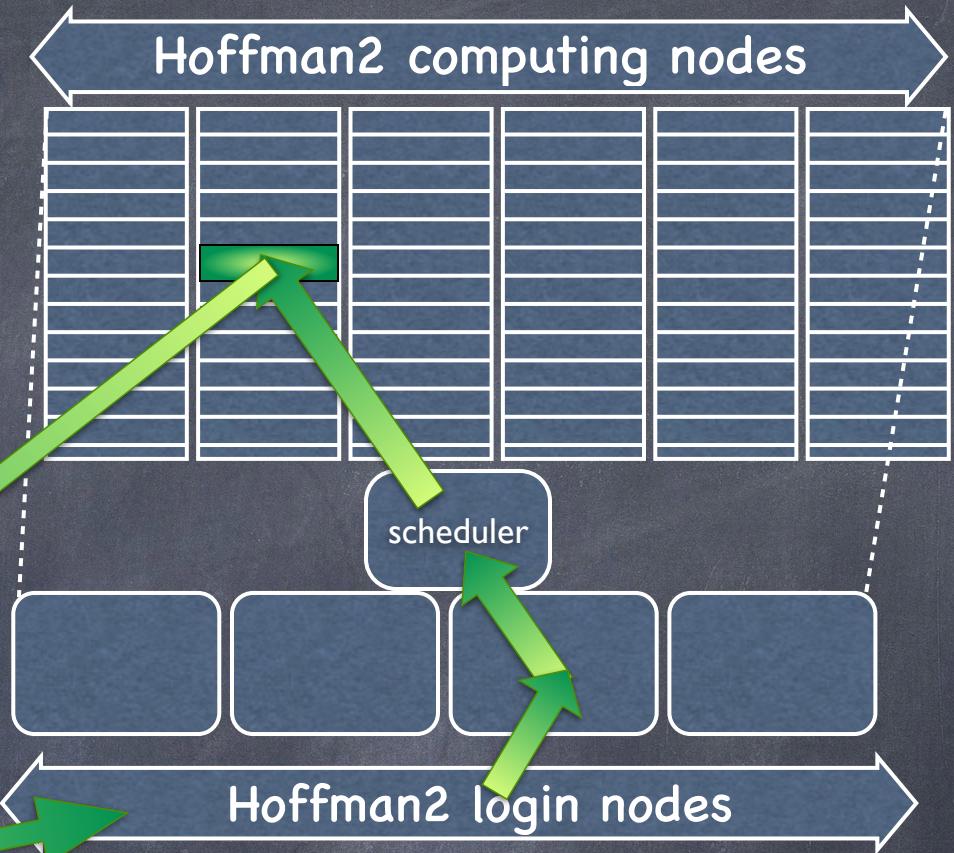
**Contents [hide]**

- Basic Usage
  - A word about h\_data
- Resource limitation
- Running on your group's nodes
- qrsh examples
- qrsh startup time
- Interpreting error messages
- Running MPI with qrsh
  - Set up the environment
  - Launch your MPI program
- Additional tools
  - get\_pe\_hostfile
  - get\_sge\_jobid
  - get\_sge\_env

# Using the Hoffman2 cluster: starting an interactive application



```
[rdtest@login2 ~]$ qrsh -l h_rt=2:00:00,h_data=4g -pe shared 2  
[rdtest@login2 ~]$ module load matlab  
[rdtest@login2 ~]$ matlab &
```



Side note: “module load <app>” places the <app> in your \$PATH

# How to learn more about environmental modules...

The screenshot shows a web browser displaying the UCLA Hoffman2 Cluster User Guide. The URL in the address bar is <https://www.hoffman2.idre.ucla.edu>. The page features a header with the idre logo, the UCLA logo, and a search bar. Below the header, there is a navigation menu with tabs: ACCESS, COMPUTING, DATA STORAGE, FILE TRANSFER, and SOFTWARE, with the SOFTWARE tab circled in green. On the left side, there is a sidebar with links for About, User Support, FAQ, Getting started, News, Security policy, Status, How to acknowledge the Hoffman2 cluster program, and How to use this site. The main content area displays the "Home" page, which welcomes visitors to the Hoffman2 Cluster User Guide and provides links to recent announcements about free classes and Stata 15.

Report Typos and Errors

UCLA

Hoffman2 Cluster User Guide

Search this website

ACCESS COMPUTING DATA STORAGE FILE TRANSFER SOFTWARE

You are here: Home

## Home

Welcome to the Hoffman2 Cluster User Guide. To look up usage information, click on one of the navigation tabs above or the sidebar items on the left.

### Recent Announcements

**Free Classes** IDRE Research Technology Group is pleased to present Fall Quarter classes. There is no charge or fee to attend these classes. See [IDRE Fall 2017 HPC Classes](#)

**Stata 15** A new version of Stata has been installed. See [Stata 15 available](#) for further information.

About [About](#)

User Support

FAQ

Getting started

News

Security policy

Status

How to acknowledge the Hoffman2 cluster program

How to use this site

# How to learn more about environmental modules...

The screenshot shows a web browser window with the following details:

- URL Bar:** https://www.hoffman2.idre.ucla.edu/software/ (highlighted by a green circle).
- Page Title:** UCLA Hoffman2 Cluster User Guide
- Header:** idre (with a blue molecular logo), Search this website, Report Typos and Errors.
- Navigation:** ACCESS, COMPUTING, DATA STORAGE, FILE TRANSFER, SOFTWARE (highlighted by a blue bar).
- Left Sidebar:** About, User Support, FAQ, Getting started, News, Security policy, Status, How to acknowledge the Hoffman2 cluster program, How to use this site.
- Current Location:** You are here: Home / Software
- Section Title:** Software
- Text:** This page provides a summary of most of the software available on the Hoffman2 cluster
- Contents:** Compilers, Editors, Environmental Modules (highlighted by a green circle), Installations requests, Applications Packages
  - Bioinformatics and Biostatistics
  - Chemistry and Chemical Engineering
  - Engineering and Mathematics
  - Statistics
  - Visualization

# How to learn more about environmental modules...

The screenshot shows a web browser window with the URL <https://www.hoffman2.idre.ucla.edu/computing/modules/> highlighted with a green oval. The page content is as follows:

**idre** Report Typos and Errors

**UCLA**  
Hoffman2 Cluster User Guide

**Search this website**

**ACCESS COMPUTING DATA STORAGE FILE TRANSFER SOFTWARE**

You are here: [Home](#) / [Computing](#) / [modules](#)

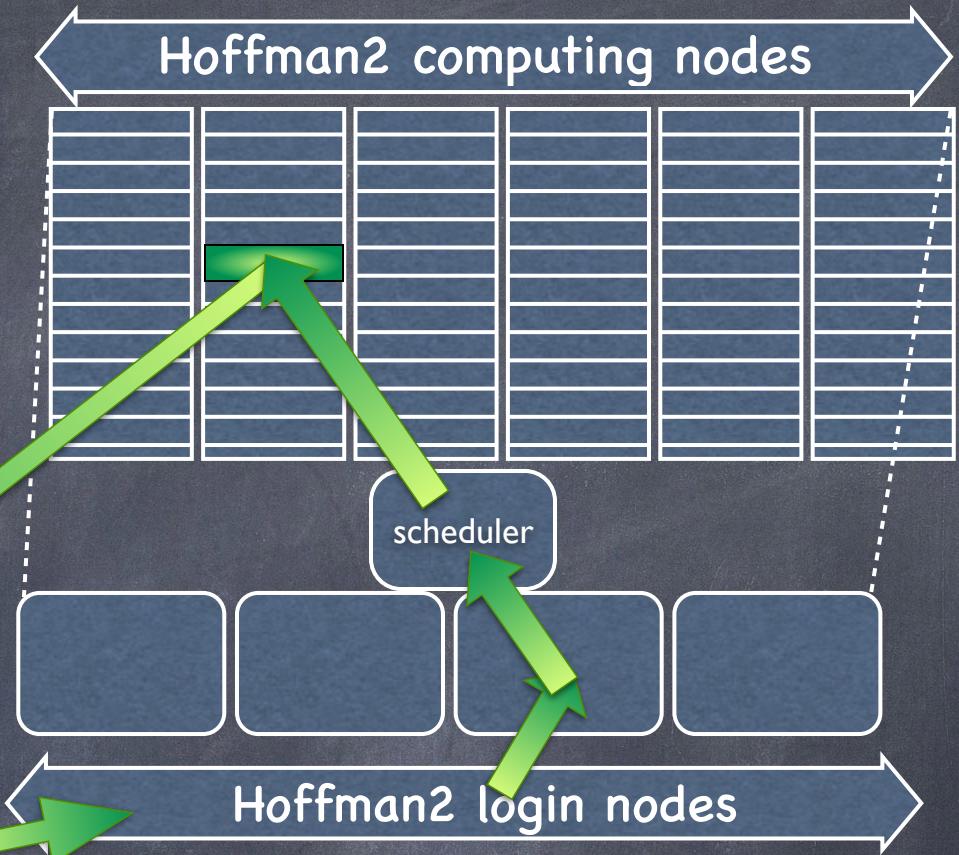
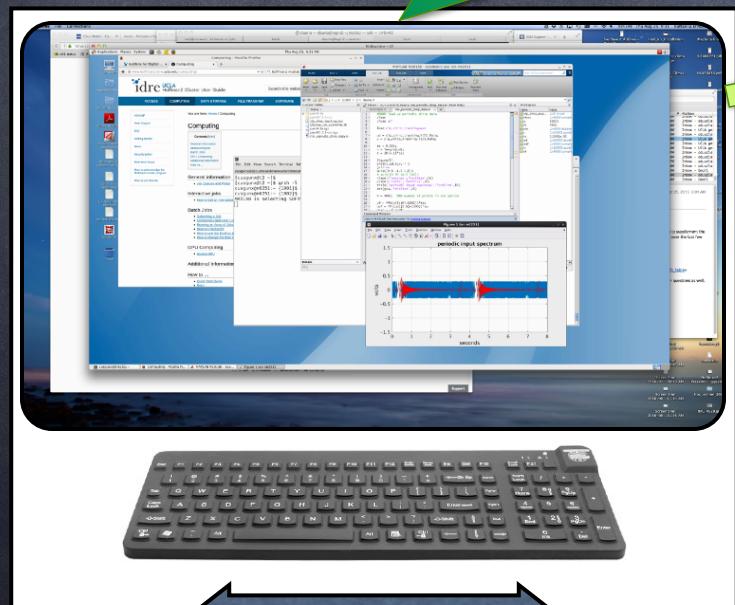
## modules

[Environmental modules](#) is a utility that allows users to dynamically modify their shell environment (e.g., \$PATH, \$LD\_LIBRARY\_PATH, etc.) in order to support a number of compilers and applications installed on the Hoffman2 Cluster.

[Contents](#) [hide]

- [Environmental modules: Basic commands](#)
- [How to use the module command in interactive sessions](#)
- [How to use the module command in scripts for batch execution](#)
- [How to use the module command in scripts for batch execution of serial jobs](#)
- [How to use the module command in scripts for batch execution of parallel jobs](#)
- [How to use the module command in scripts for batch execution of parallel jobs using shared memory](#)
- [How to use the module command in scripts for batch execution of parallel jobs using IntelMPI library](#)
- [How to use the module command in scripts for batch execution of parallel jobs using IntelMPI library and](#)

# Using the Hoffman2 cluster: running Matlab



What applications are  
available on the  
Hoffman2 cluster?

# Applications already available on Hoffman2

The screenshot shows a web browser window with the URL <https://www.hoffman2.idre.ucla.edu>. The page title is "Hoffman2 Cluster User Guide". The header includes the UCLA and idre logos, a search bar, and navigation icons. A green circle highlights the "SOFTWARE" tab in the top navigation bar. The left sidebar contains links for About, User Support, FAQ, Getting started, News, Security policy, Status, How to acknowledge the Hoffman2 cluster program, and How to use this site. The main content area displays the "Home" page with a welcome message, recent announcements about free classes and Stata 15, and a sidebar with a "Report Typos and Errors" link.

Report Typos and Errors

UCLA

Hoffman2 Cluster User Guide

Search this website

ACCESS COMPUTING DATA STORAGE FILE TRANSFER SOFTWARE

About

User Support

FAQ

Getting started

News

Security policy

Status

How to acknowledge the Hoffman2 cluster program

How to use this site

You are here: Home

## Home

Welcome to the Hoffman2 Cluster User Guide. To look up usage information, click on one of the navigation tabs above or the sidebar items on the left.

### Recent Announcements

**Free Classes** IDRE Research Technology Group is pleased to present Fall Quarter classes. There is no charge or fee to attend these classes. See [IDRE Fall 2017 HPC Classes](#)

**Stata 15** A new version of Stata has been installed. See [Stata 15 available](#) for further information.

Report Typos and Errors

# Applications already available on Hoffman2

The screenshot shows a web browser displaying the UCLA Hoffman2 Cluster User Guide. The URL in the address bar is <https://www.hoffman2.idre.ucla.edu/software/>. A green oval highlights the URL. The page title is "Software". The main content area lists various software categories, with "Compilers" highlighted by a green rectangle.

UCLA Hoffman2 Cluster User Guide

Search this website

ACCESS COMPUTING DATA STORAGE FILE TRANSFER SOFTWARE

About  User Support FAQ Getting started News Security policy Status How to acknowledge the Hoffman2 cluster program How to use this site

You are here: Home / Software

## Software

This page provides a summary of most of the software available on the Hoffman2 cluster

Contents [hide]

- Compilers**
- Editors
- Environmental Modules
- Installations requests
- Applications Packages
  - Bioinformatics and Biostatistics
  - Chemistry and Chemical Engineering
  - Engineering and Mathematics
  - Statistics
  - Visualization
  - Languages and Debuggers
  - Integrated development environments (IDEs)
  - Programming Libraries
  - Misellaneous

# Applications already available on Hoffman2

The screenshot shows a web browser displaying the UCLA Hoffman2 Cluster User Guide. The URL in the address bar is https://www.hoffman2.idre.ucla.edu/software/. A green oval highlights the URL. The page title is "Software". The main content area lists various software categories, with "Installations requests" highlighted by a green rectangle.

https://www.hoffman2.idre.ucla.edu/software/

Report Typos and Errors

idre

UCLA Hoffman2 Cluster User Guide

Search this website

ACCESS COMPUTING DATA STORAGE FILE TRANSFER SOFTWARE

About 

User Support

FAQ

Getting started

News

Security policy

Status

How to acknowledge the Hoffman2 cluster program

How to use this site

You are here: Home / Software

## Software

This page provides a summary of most of the software available on the Hoffman2 cluster

Contents [hide]

- Compilers
- Editors
- Environmental Modules
- Installations requests**
- Applications Packages
  - Bioinformatics and Biostatistics
  - Chemistry and Chemical Engineering
  - Engineering and Mathematics
  - Statistics
  - Visualization
  - Languages and Debuggers
  - Integrated development environments (IDEs)
  - Programming Libraries
  - Misellaneous

# Applications already available on Hoffman2

The screenshot shows a web browser displaying the UCLA Hoffman2 Cluster User Guide. The URL in the address bar is https://www.hoffman2.idre.ucla.edu/software/. A green circle highlights the URL. The page title is "Software". A search bar says "Search this website". A sidebar on the left lists various links: About, User Support, FAQ, Getting started, News, Security policy, Status, How to acknowledge the Hoffman2 cluster program, and How to use this site. A green box highlights the "Applications Packages" section under the "Contents" heading, which includes links to Bioinformatics and Biostatistics, Chemistry and Chemical Engineering, Engineering and Mathematics, Statistics, Visualization, Languages and Debuggers, Integrated development environments (IDEs), Programming Libraries, and Misellaneous.

https://www.hoffman2.idre.ucla.edu/software/

Report Typos and Errors

idre

UCLA Hoffman2 Cluster User Guide

Search this website

ACCESS COMPUTING DATA STORAGE FILE TRANSFER SOFTWARE

You are here: Home / Software

## Software

This page provides a summary of most of the software available on the Hoffman2 cluster

Contents [hide]

- Compilers
- Editors
- Environmental Modules
- Installations requests
- Applications Packages**
- Bioinformatics and Biostatistics
- Chemistry and Chemical Engineering
- Engineering and Mathematics
- Statistics
- Visualization
- Languages and Debuggers
- Integrated development environments (IDEs)
- Programming Libraries
- Misellaneous

# Applications already available on Hoffman2

```
[rdtest@login2 ~]$ module av
```

```
----- /u/local/Modules/modulefiles -----
```

ATS	gcc/4.4(default)	netcdf/4.1.3(default)
R/2.12.0	gcc/4.7.2	netcdf/4.1.3-shared
R/2.12.1	gcc/4.9.3	netcdf/4.1.3_i
R/2.12.2	gcc/6.3.0	netcdf/4.2.3-c
R/2.13.2	gcta/0.93.9	netcdf/4.4.2-fortran
R/2.15.1	gdal/1.9.2(default)	ngsplot/2.47
R/2.9.1	gdal/2.2.0	nwchem/6.5(default)
R/3.0.1	gdc/0.30	octave/3.6.1
R/3.1.1	gdc/4.9.3(default)	octave/3.6.4(default)
R/3.2.1	geant4/10.00.p02_wGDML	oommf/12b0
R/3.2.3(default)	geant4/4.9.3.p02(default)	openbabel/2.3.1(default)
R/3.3.0	geant4/4.9.4.p02	openblas/0.2.18(default)
R/3.3.3	geant4/4.9.5	openblas/0.2.18-threaded
R/3.4.0	geant4/9.6.2	opencv/2.4.11
R/3.4.1	geant4/9.6.p02_wGDML	openmpi/1.6_gcc-4.4
Rstudio/0.98(default)	geos/3.4.2(default)	opensees/2.4.1

...

...

...

& many more...

# Applications already available on Hoffman2: Which R packages?

```
[rdtest@n9860 ~]$ module load R/3.4.1
```

```
[...]
```

```
[rdtest@n9860 ~]$ R
```

```
R version 3.4.1 (2017-06-30) -- "Single Candle"
```

```
Copyright (C) 2017 The R Foundation for Statistical Computing
```

```
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
[...]
```

```
Type 'q()' to quit R.
```

```
> library()
```

```
Packages in library '/u/home/r/rdtest/R/x86_64-pc-linux-gnu-library/3.4':
```

**bbmle**

Tools for General Maximum Likelihood Estimation

**BH**

Boost C++ Header Files

**Biobase**

Biobase: Base functions for Bioconductor

**... & more...**

# Applications already available on Hoffman2: Which R packages?

```
[rdtest@n9860 ~]$ module load R/3.4.1
```

```
[rdtest@n9860 ~]$ R
```

```
R version 3.4.1 (2017-06-30) -- "Single Candle"  
Copyright (C) 2017 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)  
[...]  
Type 'q()' to quit R.
```

```
> install.packages("xlsx", dependencies = TRUE)
```

# Applications already available on Hoffman2: Which R packages?

```
[rdtest@n9860 ~]$ module load R/3.4.1
```

```
[rdtest@n9860 ~]$ R
```

```
R version 3.4.1 (2017-06-30) -- "Single Candle"  
Copyright (C) 2017 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)  
[...]  
Type 'q()' to quit R.
```

```
> install.packages("xlsx", dependencies = TRUE)  
Installing package into '/u/home/r/rdtest/R/x86_64-pc-linux-gnu-library/3.4'  
(as 'lib' is unspecified)  
--- Please select a CRAN mirror for use in this session –
```

# Applications already available on Hoffman2: Which R packages?

```
[rdtest@n9860 ~]$ module load R/3.4.1
```

```
[rdtest@n9860 ~]$ R
```

```
R version 3.4.1 (2017-06-30) -- "Single Candle"  
Copyright (C) 2017 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)  
[...]  
Type 'q()' to quit R.
```

```
> install.packages("xlsx", dependencies = TRUE)  
Installing package into '/u/home/r/rdtest/R/x86_64-pc-linux-gnu-library/3.4'  
(as 'lib' is unspecified)  
--- Please select a CRAN mirror for use in this session –  
[...]  
** testing if installed package can be loaded  
* DONE (xlsx)
```

# Running a batch job

Why?

# Running a batch job

The screenshot shows a web browser window with the following details:

- Address Bar:** https://www.hoffman2.idre.ucla.edu
- Toolbar:** Includes back, forward, search, and other standard browser icons.
- Header:** The page title is "Running a batch job". The header also includes the UCLA IDRE logo and a "Report Typos and Errors" link.
- Content Area:**
  - UCLA Hoffman2 Cluster User Guide:** A sidebar on the left contains links to "About", "User Support", "FAQ", "Getting started", "News", "Security policy", "Status", "How to acknowledge the Hoffman2 cluster program", and "How to use this site".
  - Search:** A "Search this website" input field.
  - Navigation Tabs:** A blue navigation bar with tabs: ACCESS, COMPUTING (circled in green), DATA STORAGE, FILE TRANSFER, and SOFTWARE.
  - Page Content:** The main content area displays the "Home" page of the user guide. It includes a breadcrumb trail ("You are here: Home"), the title "Home", a welcome message, and sections for "Recent Announcements" (mentioning free classes and Stata 15) and "Free Classes".

# Running a batch job

The screenshot shows a web browser displaying the 'Computing' page of the idre Hoffman2 Cluster User Guide. The URL in the address bar is <https://www.hoffman2.idre.ucla.edu/computing/>. The page has a dark blue header with navigation links for ACCESS, COMPUTING, DATA STORAGE, FILE TRANSFER, and SOFTWARE. On the left, there's a sidebar with links for About, User Support, FAQ, Getting started, News, Security policy, Status, How to acknowledge the Hoffman2 cluster program, and How to use this site. The main content area shows the 'Computing' page with a breadcrumb trail: You are here: Home / Computing. Below this is a 'Contents [hide]' box containing links for General information, Interactive jobs, Batch Jobs (which is circled in green), GPO Computing, Additional Information, and How to ... . Under 'General information', there's a bullet point for Job Queues and Policy. Under 'Interactive jobs', there's a bullet point for How to Get an Interactive Session. Under 'Batch Jobs', there are two bullet points: Submitting a Job and Commonly-Used UGE Commands (also circled in green). At the bottom, there's a footer for Apps on H2: Case Studies and a copyright notice for R. D'Auria, PhD.

https://www.hoffman2.idre.ucla.edu/computing/

120% C ☆ Search

Report Typos and Errors

idre UCLA

Hoffman2 Cluster User Guide

Search this website

ACCESS COMPUTING DATA STORAGE FILE TRANSFER SOFTWARE

About User Support FAQ Getting started News Security policy Status How to acknowledge the Hoffman2 cluster program How to use this site

You are here: Home / Computing

## Computing

Contents [hide]

General information  
Interactive jobs  
**Batch Jobs**  
GPO Computing  
Additional Information  
How to ...

### General information

- Job Queues and Policy

### Interactive jobs

- How to Get an Interactive Session

### Batch Jobs

- Submitting a Job
- Commonly-Used UGE Commands

Apps on H2: Case Studies

R. D'Auria, PhD

# Running a batch job

```
[rdtest@login3 ~]$ cp /u/local/apps/submit_scripts/submit_job.sh ./
[rdtest@login3 ~]$ ls -lat submit_job.sh
-rwxr--r-- 1 rdtest systems 575 Oct 11 14:05 submit_job.sh
[rdtest@login3 ~]$ █
```

If `submit_job.sh` had not been executable we should have issued:

```
[rdtest@login3 ~]$ chmod u+x submit_job.sh
```

To submit the job:

```
[rdtest@login3 ~]$ qsub submit_job.sh
Your job 865688 ("submit_job.sh") has been submitted
[rdtest@login3 ~]$ █
```

# Checking the status of a job in the queue

```
[rdtest@login3 ~]$ myjob
job-ID prior name user state submit/start at queue slots ja-task-ID
-----
865688 10.000000 submit_job rdtest r 10/11/2017 14:24:34 c2_smp.q@n9860 1
[rdtest@login3 ~]$
```

If `submit_job.sh` had been pending its “state” would have been “qw”

```
[rdtest@login3 ~]$ qstat -s p | head -n 6
job-ID prior name user state submit/start at queue slots ja-task-ID
-----
861033 5.75286 mayo_cruce caroartc qw 10/10/2017 22:59:11 4
862844 2.87643 QRLOGIN caroartc qw 10/11/2017 07:20:43 4
121156 1.91801 roms_TEST_ hfrenzel qw 07/14/2017 11:24:01 96
740935 1.91771 in.ThreeD_ tjones19 qw 09/27/2017 12:37:24 24
```

# Anatomy of a batch job script

```
#!/bin/bash
#$ -cwd
# error = Merged with joblog
#$ -o joblog.$JOB_ID
#$ -j y
#$ -l h_rt=1:00:00,h_data=1G
# Email address to notify
#$ -M $USER@mail
# Notify when
#$ -m bea

# echo job info on joblog:
echo "Job $JOB_ID started on: `hostname -s`"
echo "Job $JOB_ID started on: `date`"
echo " "

# load the job environment:
. /u/local/Modules/default/init/modules.sh
module load <APP>
module li
echo " "

# run <APP>
echo '<execution command>'<execution command>

# echo job info on joblog:
echo "Job $JOB_ID ended on: `hostname -s`"
echo "Job $JOB_ID ended on: `date`"
echo " "
```

# Anatomy of a batch job script

Shell that will be used

```
#!/bin/bash
#$ -cwd
# error = Merged with joblog
#$ -o joblog.$JOB_ID
#$ -j y
#$ -l h_rt=1:00:00,h_data=1G
# Email address to notify
#$ -M $USER@mail
# Notify when
#$ -m bea

# echo job info on joblog:
echo "Job $JOB_ID started on: `hostname -s`"
echo "Job $JOB_ID started on: `date`"
echo ""

# load the job environment:
. /u/local/Modules/default/init/modules.sh
module load <APP>
module li
echo ""

# run <APP>
echo '<execution command>'<execution command>

# echo job info on joblog:
echo "Job $JOB_ID ended on: `hostname -s`"
echo "Job $JOB_ID ended on: `date`"
echo ""
```

# Anatomy of a batch job script

Shell that will be used

```
#!/bin/bash
```

```
#$ -cwd
```

```
# error = Merged with joblog
```

```
#$ -o joblog.$JOB_ID
```

```
#$ -j y
```

```
#$ -l h_rt=1:00:00,h_data=1G
```

```
# Email address to notify
```

```
#$ -M $USER@mail
```

```
# Notify when
```

```
#$ -m bea
```

#\$ scheduler commands

```
# echo job info on joblog:
```

```
echo "Job $JOB_ID started on: " `hostname -s`
```

```
echo "Job $JOB_ID started on: " `date`
```

```
echo "
```

# comments

```
# load the job environment:
```

```
. /u/local/Modules/default/init/modules.sh
```

```
module load <APP>
```

```
module li
```

```
echo "
```

```
# run <APP>
```

```
echo '<execution command>'
```

```
<execution command>
```

```
# echo job info on joblog:
```

```
echo "Job $JOB_ID ended on: " `hostname -s`
```

```
echo "Job $JOB_ID ended on: " `date`
```

```
echo "
```

# Anatomy of a batch job script

Shell that will be used

```
#!/bin/bash  
#$ -cwd  
# error = Merged with joblog  
#$ -o joblog.$JOB_ID  
#$ -j y  
#$ -l h_rt=1:00:00,h_data=1G  
# Email address to notify  
#$ -M $USER@mail  
# Notify when  
#$ -m bea
```

Submission script  
preamble

#\$ scheduler commands

```
# echo job info on joblog:  
echo "Job $JOB_ID started on: " `hostname -s`  
echo "Job $JOB_ID started on: " `date`  
echo "
```

```
# load the job environment:  
. /u/local/Modules/default/init/modules.sh  
module load <APP>  
module li  
echo "
```

# comments

```
# run <APP>  
echo '<execution command>'  
<execution command>
```

```
# echo job info on joblog:  
echo "Job $JOB_ID ended on: " `hostname -s`  
echo "Job $JOB_ID ended on: " `date`  
echo "
```

# Anatomy of a batch job script

```
[rdtest@login3 ~]$ myjob
```

You do not have any active SGE jobs currently.

```
[rdtest@login3 ~]$ cat joblog.865688
```

```
Job 865688 started on: n9860
```

```
Job 865688 started on: Wed Oct 11 14:24:35 PDT 2017
```

The 'gcc/4.9.3' module is being loaded

```
pwd; sleep 500
```

```
/u/home/r/rdtest
```

```
Job 865688 ended on: n9860
```

```
Job 865688 ended on: Wed Oct 11 14:32:55 PDT 2017
```

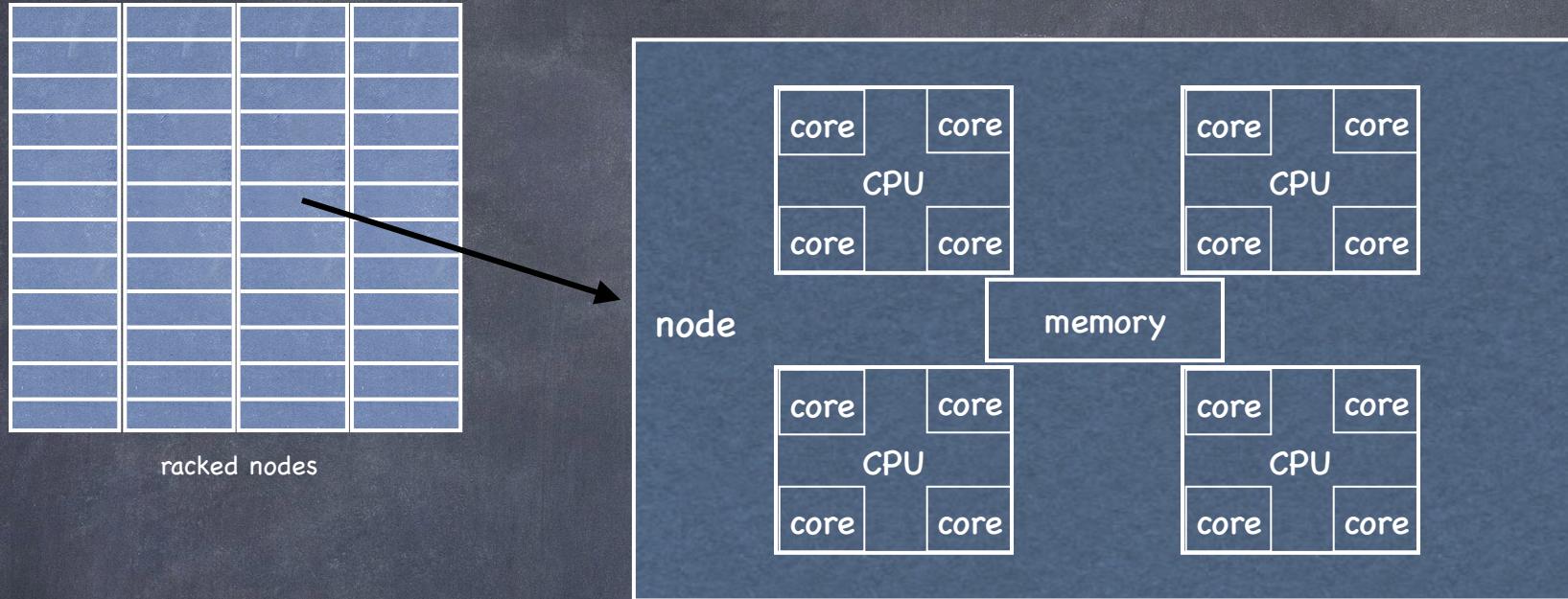
```
[rdtest@login3 ~]$
```

# Running a batch job

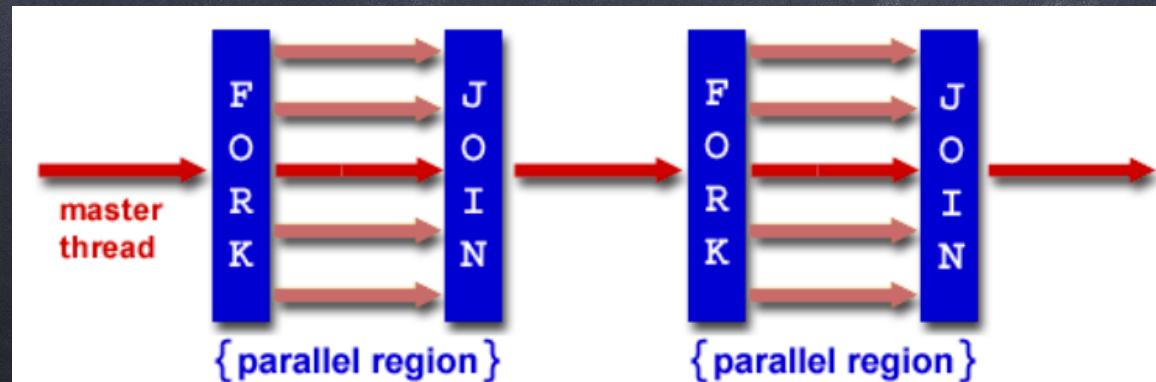
You can also use our own queue scripts, such as:

- job.q (serial jobs)
- openmp.q (multiple threads/shared memory jobs)
- intelmpi.q (distributed memory jobs w/ intelmpi)
- openmpi.q (distributed memory jobs w/ openmpi)
- parallel.q (distributed memory jobs DIY)
- <APP>.q:
  - stata.q. (allows serial/multiple threads)
  - R.q (serial only version of R set)
  - etc.
- jobarray.q. (embarrassingly parallel jobs)

# Shared Memory jobs



- Use more than one core (slot in queue jargon)
- All cores are from the same node (server/computer)
- The program spawns threads each of which runs on a separate core



# Shared Memory jobs

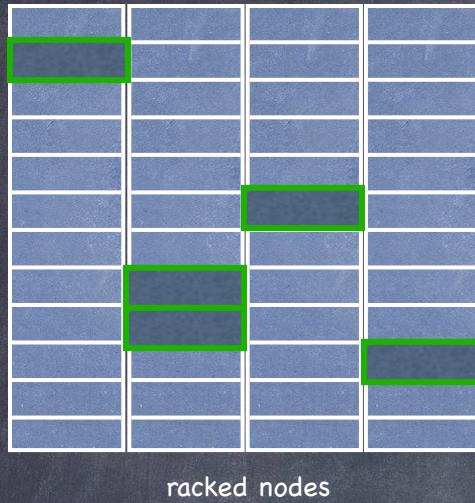
You may be running in shared memory jobs b/c:

- You link to multi-threaded libraries (MKL, blas, lapak, etc.)
- You have explicitly included threads/openmp chunks of code
- You are running an app, developed by a third party, which spawns threads
  - You should accordingly request multiple cores
    - Either with openmp.q
    - Or generalize your submission script preamble:

```
#!/bin/bash
#$ -cwd
# error = Merged with joblog
#$ -o joblog.$JOB_ID
#$ -j y
#$ -l h_rt=1:00:00,h_data=1G
#$ -pe shared 6
# Email address to notify
#$ -M $USER@mail
# Notify when
#$ -m bea
```

```
#!/bin/bash
#$ -cwd
# error = Merged with joblog
#$ -o joblog.$JOB_ID
#$ -j y
#$ -l h_rt=1:00:00,h_data=1G,exclusive
# Email address to notify
#$ -M $USER@mail
# Notify when
#$ -m bea
```

# Distributed memory jobs



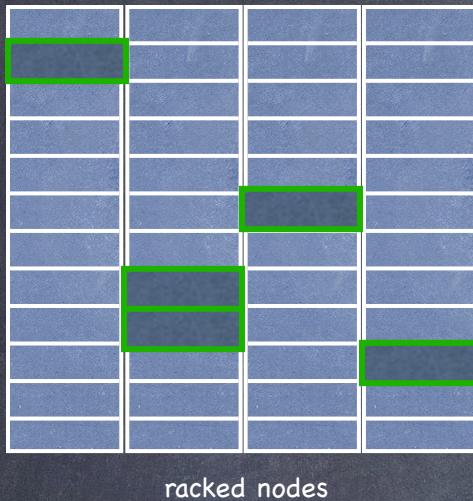
- Each parallel worker has its own private memory space
- All communications is done with message passing (MPI)
- Your code contains parallel instructions

As you need multiple cores possibly coming from multiple nodes

- Either use intelmpi.q/openmpi.q/parallel.q
- Or generalize your job script preamble

```
#!/bin/bash
#$ -cwd
# error = Merged with joblog
#$ -o joblog.$JOB_ID
#$ -j y
#$ -l h_rt=1:00:00,h_data=1G
#$ -pe dc* 24
# Email address to notify
#$ -M $USER@mail
# Notify when
#$ -m bea
```

# Hybrid distributed memory/shared memory jobs



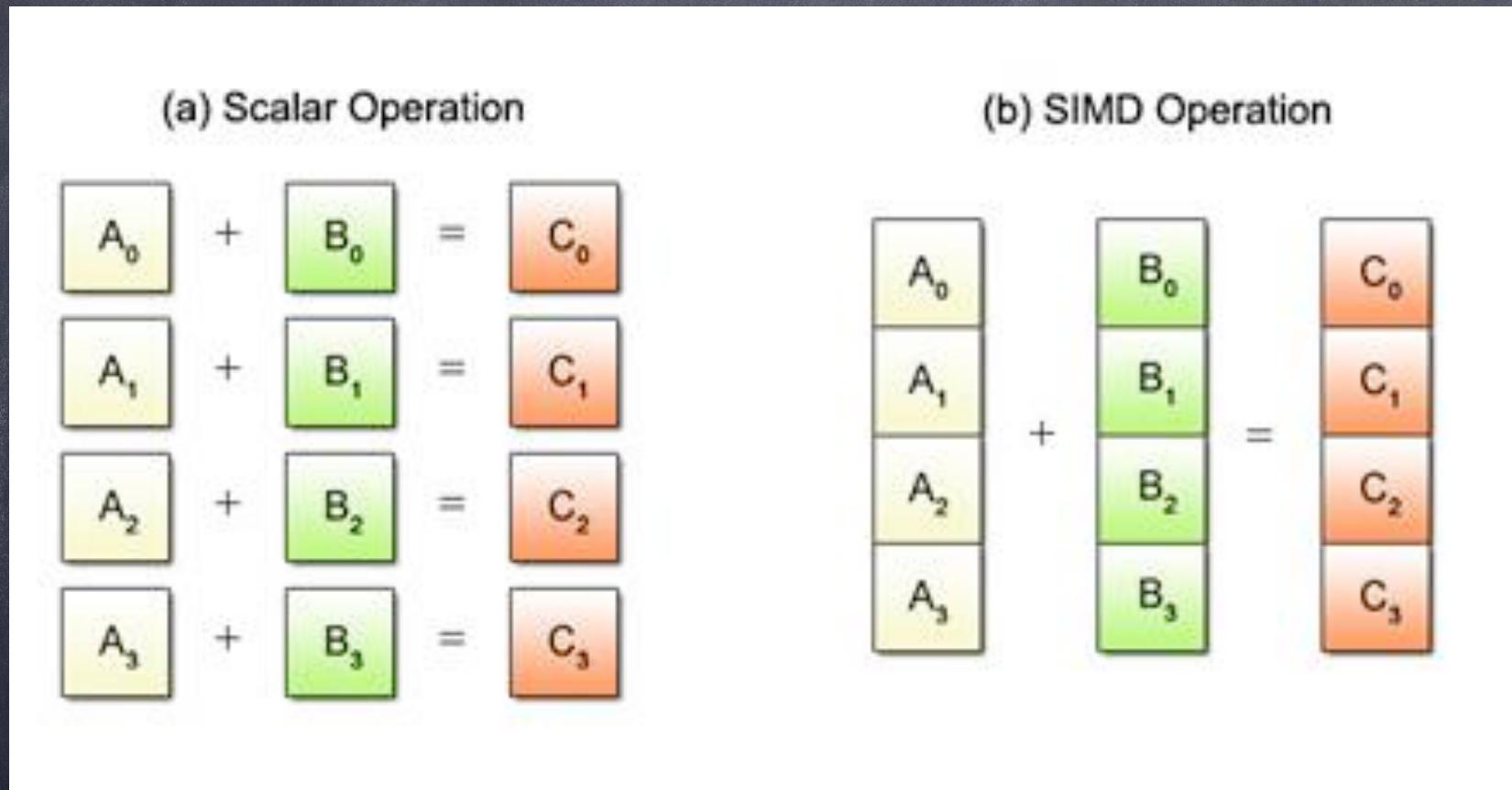
- Message passing between the nodes
- Shared memory within each node
- Used by:
  - comsol.q (sorry no public licenses)
  - gaussian.q

As you need multiple cores possibly coming from multiple nodes  
→ generalize your job script preamble

```
#!/bin/bash
#$ -cwd
# error = Merged with joblog
#$ -o joblog.$JOB_ID
#$ -j y
#$ -l h_rt=1:00:00,h_data=1G,exclusive
#$ -pe node 3
# Email address to notify
#$ -M $USER@mail
# Notify when
#$ -m bea
```

# Embarrassingly Parallel

"In parallel computing, an embarrassingly parallel workload or problem (also called perfectly parallel or pleasingly parallel) is one where little or no effort is needed to separate the problem into a number of parallel tasks." [Wikipedia]



Instead of looping through (scalar operation) we can give each processor the elements of the arrays to sum and all the sums happen concurrently

# Embarrassingly Parallel

What is the most efficient way to tackle these problems on a cluster?

- if the problem can fit on a node (there are enough cores for the number of threads we would need to have)

```
#!/bin/bash
#$ -cwd
# error = Merged with joblog
#$ -o joblog.$JOB_ID
#$ -j y
#$ -l h_rt=1:00:00,h_data=1G
#$ -pe shared 6
```

```
#!/bin/bash
#$ -cwd
# error = Merged with joblog
#$ -o joblog.$JOB_ID
#$ -j y
#$ -l h_rt=1:00:00,h_data=1G,exclusive
```

- we could distribute it across several machines with:

```
#!/bin/bash
#$ -cwd
# error = Merged with joblog
#$ -o joblog.$JOB_ID
#$ -j y
#$ -l h_rt=1:00:00,h_data=1G
#$ -pe dc* 24
```

Wait? Work that could have been done totally independently has now to wait for resources to become available concurrently...

# Job Arrays

https://www.hoffman2.idre.ucla.edu

| c |  |  |  | → | ↓ | ⌂ | ⌄ | ⌄

  
Report  
Typos  
and  
Errors

**UCLA**  
Hoffman2 Cluster User Guide

array job

ACCESS COMPUTING DATA STORAGE FILE TRANSFER SOFTWARE

You are here: [Home](#)

## Home

Welcome to the Hoffman2 Cluster User Guide. To look up usage information, click on one of the navigation tabs above or the sidebar items on the left.

### Recent Announcements

**Free Classes** IDRE Research Technology Group is pleased to present Fall Quarter classes. There is no charge or fee to attend these classes. See [IDRE Fall 2017 HPC Classes](#)

**Stata 15** A new version of Stata has been installed. See [Stata 15 available](#) for further information.

About   
User Support  
FAQ  
Getting started  
News  
Security policy  
Status  
How to acknowledge the Hoffman2 cluster program  
How to use this site

# Job Arrays

The screenshot shows a web browser window with the URL [https://www.hoffman2.idre.ucla.edu/computing/job\\_arrays/](https://www.hoffman2.idre.ucla.edu/computing/job_arrays/). The page title is "Job Arrays". The header includes the idre logo, the UCLA logo, and a search bar. A navigation menu at the top has categories: ACCESS, COMPUTING, DATA STORAGE, FILE TRANSFER, and SOFTWARE. On the left, a sidebar contains links for About, User Support, FAQ, Getting started, News, Security policy, Status, How to acknowledge the Hoffman2 cluster program, and How to use this site. The main content area shows the breadcrumb "You are here: Home / Computing / Running an Array of Jobs Using UGE" and the title "Running an Array of Jobs Using UGE". A "Contents [hide]" section lists "UGE Job Arrays", "File: hellotask.sh", "File: data1.in", "File: data2.in", and "Using the Queue Script to Submit a Job Array". Below this, a section titled "UGE Job Arrays" explains the UGE job array option. It states: "The Univa Grid Engine (UGE) has a job array option, which lets you submit multiple jobs with a single command file. It is convenient to use job arrays when you need to conduct a parametric study to optimize the various parameters in an experiment. Computing the optimum solution involves repeated execution of same program over a range of input values and this is exactly what the job array option lets you do." It also says: "To run using job arrays, you need to insert an instruction of the form:" followed by the code "#\$ -t lower-upper:interval" which is circled in green. The text continues: "into your UGE command file."

You are here: Home / Computing / Running an Array of Jobs Using UGE

## Running an Array of Jobs Using UGE

**Contents [hide]**

- [UGE Job Arrays](#)
- [File: hellotask.sh](#)
- [File: data1.in](#)
- [File: data2.in](#)
- [Using the Queue Script to Submit a Job Array](#)

### UGE Job Arrays

The Univa Grid Engine (UGE) has a job array option, which lets you submit multiple jobs with a single command file. It is convenient to use job arrays when you need to conduct a parametric study to optimize the various parameters in an experiment. Computing the optimum solution involves repeated execution of same program over a range of input values and this is exactly what the job array option lets you do.

To run using job arrays, you need to insert an instruction of the form:

#\$ -t lower-upper:interval

into your UGE command file.

# Array jobs: submit multiple jobs with a single command file

- can your job be broken down in a series of tasks that are totally independent one from the other?
  - generally not if you are predicting tomorrow's weather...
  - great if you are doing a parametric study

```
#!/bin/sh
echo "Task id is $SGE_TASK_ID"
if [ -e $HOME/JOBARRAY/data$SGE_TASK_ID.in ]; then
while read file
do
echo "hello $file"
done < $HOME/JOBARRAY/data$SGE_TASK_ID.in
fi
```

# Array jobs: submit multiple jobs with a single command file

```
[rdtest@login1 ~]$ mkdir $HOME/JOBARRAY
[rdtest@login1 ~]$ cd $HOME/JOBARRAY
[rdtest@login1 ~/JOBARRAY]$ cp /u/local/apps/submit_scripts/array_jobs/hellotask.sh ./
[rdtest@login1 ~/JOBARRAY]$ cp /u/local/apps/submit_scripts/array_jobs/data*.in ./
[rdtest@login1 ~/JOBARRAY]$ jobarray.q
```

Enter to continue.

```
jobarray Serial UGE job script
Functions (acceptable abbreviations are shown in CAPS)
Menu:      Display this menu
Info:      Display help information
Build:     Build a UGE .cmd file for jobarray
Submit:    Submit a UGE .cmd file for execution
Status:    Display the status of UGE jobs for rdtest
SYsstat:   Display the status of UGE jobs for the system
Hold:      Hold a UGE job
RElease:   Release a UGE job that is held
RESet:     Reset the priority of a UGE job
Cancel:    Cancel a UGE job
Quit:     Exit this script
```

Command: b

```
Enter the name of the Job array program or script to be executed
<or quit>: hellotask.sh
[...]
```

# Array jobs

```
[rdtest@login1 ~/JOBARRAY]$ head -n 35 hello task.sh.cmd
```

```
#!/bin/csh -f
# hellotask.sh.cmd
#
# UGE job for hellotask.sh built Wed Oct 11 20:51:21 PDT 2017
#
# The following items pertain to this script
# Use current working directory
#$ -cwd
# input          = /dev/null
# output         = /u/home/r/rdtest/matlab/ARRAYJOBS/hellotask.sh.joblog.$JOB_ID.$TASK_ID
#$ -o /u/home/r/rdtest/matlab/ARRAYJOBS/hellotask.sh.joblog.$JOB_ID.$TASK_ID
# error          = Merged with joblog
#$ -j y
# The following items pertain to the user program
# user program   = /u/home/r/rdtest/matlab/ARRAYJOBS/hellotask.sh
# arguments      =
# program input  = Specified by user program
# program output = Specified by user program
# Resources requested
#
#$ -l h_data=1024M,h_rt=1:00:00
# #
# Name of application for log
#$ -v QQAPP=job
# Email address to notify
#$ -M rdtest@mail
# Notify at beginning and end of job
#$ -m a
# Job is not rerunable
#$ -r n
#
# Job array indexes
#$ -t 1-2:1
[...]
```

# Array jobs

```
[rdtest@login1 ~/JOBARRAY]$ ls -lt
total 24
-rw-r--r-- 1 rdtest systems 547 Oct 11 20:52 hellotask.sh.joblog.868129.2
-rw-r--r-- 1 rdtest systems 149 Oct 11 20:52 hellotask.sh.output.868129.2
-rw-r--r-- 1 rdtest systems 544 Oct 11 20:52 hellotask.sh.joblog.868129.1
-rw-r--r-- 1 rdtest systems 149 Oct 11 20:52 hellotask.sh.output.868129.1
-rwxr--r-- 1 rdtest systems 3165 Oct 11 20:51 hellotask.sh.cmd
-rw-r--r-- 1 rdtest systems 13 Oct 11 20:50 data2.in
-rw-r--r-- 1 rdtest systems 13 Oct 11 20:50 data1.in
-rwxr--r-- 1 rdtest systems 183 Oct 11 20:50 hellotask.sh
[rdtest@n9860 ~/JOBARRAY]$ head -n3 hellotask.sh.output.868129.1
"Task id is 1"
"hello first"
"hello second"
```

If your job can be broken in 100s, or 1000s or 10,000s independent tasks you are in business!!!

# Array jobs R example

```
[rdtest@login3 R]$ cp /u/local/apps/submit_scripts/array_jobs/R/test_array.R ./
[rdtest@login3 R]$ cp /u/local/apps/submit_scripts/array_jobs/R/submit_R_array.sh./
[rdtest@login3 R]$ cat test_array.R
```

```
options(echo=TRUE) # if you want see commands in output file
args <- commandArgs(trailingOnly = TRUE)
print(args)
n <- length(args)
print(n)
suff <- args[n]
outputfile <- paste("myout", ".", suff, sep="")
write(args, file=outputfile, ncolumns = n,
      append = FALSE, sep = " ")
Sys.sleep(120)
```

# Array jobs R example

```
[rdtest@login3 R]$ cat submit_R_array.sh
```

```
#!/bin/bash
#$ -cwd
#$ -o test.joblog.$JOB_ID.$TASK_ID
#$ -j y
# Resources requested:
#$ -l h_data=1g,h_rt=2:00:00
# Email address to notify
#$ -M $USER@mail
#$ -m bea
#$ -t 1-3:1

echo ""
echo "test started on:  "` hostname -s `"
echo "test started at:  "` date `"
echo ""

#
. /u/local/Modules/default/init/modules.sh
module load R

echo "Task id is \$SGE_TASK_ID"

#
# Run the user program
#
R CMD BATCH --no-save --no-restore "--args \$SGE_TASK_ID" ./test_array.R \
test_array.out.${JOB_ID}.${SGE_TASK_ID}
```

# Array jobs in R

```
[rdtest@login3 R]$ chmod u+x submit_R_array.sh
[rdtest@login3 R]$ qsub submit_R_array.sh
Your job-array 868598.1-3:1 ("submit_R_array.sh") has been submitted
[rdtest@login3 R]$ ls -lt
total 32
-rw-r--r-- 1 rdtest systems 1093 Oct 11 21:40 test_array.out.868598.3
-rw-r--r-- 1 rdtest systems 1093 Oct 11 21:40 test_array.out.868598.2
-rw-r--r-- 1 rdtest systems 1093 Oct 11 21:40 test_array.out.868598.1
-rw-r--r-- 1 rdtest systems    2 Oct 11 21:38 myout.3
-rw-r--r-- 1 rdtest systems    2 Oct 11 21:38 myout.1
-rw-r--r-- 1 rdtest systems    2 Oct 11 21:38 myout.2
-rw-r--r-- 1 rdtest systems   94 Oct 11 21:38 test.joblog.868598.3
-rw-r--r-- 1 rdtest systems   94 Oct 11 21:38 test.joblog.868598.2
-rw-r--r-- 1 rdtest systems   94 Oct 11 21:38 test.joblog.868598.1
-rwxr--r-- 1 rdtest systems  529 Oct 11 21:29 submit_R_array.sh
-rw-r--r-- 1 rdtest systems  295 Oct 11 21:28 test_array.R
```

# Explicit MPI parallelism in R: Rmpi

Rmpi is a package for R, a free software environment for statistical computing and graphics. It is an interface (wrapper) to MPI (Message-Passing Interface). MPI is a standardized and portable message-passing system designed by a group of researchers from academia and industry to function on a wide variety of parallel computers such as Beowulf clusters. [<http://www.stats.uwo.ca/faculty/yu/Rmpi/>]

# Rmpi

```
[rdtest@login3 ~]$ qrsh -l h_rt=1:00:00,h_data=1g -pe dc* 14
```

```
[rdtest@n2190 ~]$ . /u/local/bin/set_qrsh_env.sh
```

```
[rdtest@n2190 ~]$ cat $PE_HOSTFILE
```

```
n2190 8 inter_msa.q@n2190 UNDEFINED
```

```
n2187 6 inter_msa.q@n2187 UNDEFINED
```

```
[rdtest@n2190 ~]$ module li
```

Currently Loaded Modulefiles:

- 1) ATS
- 2) intel/13.cs(default)

```
[rdtest@n2190 ~]$ module av R
```

```
----- /u/local/Modules/modulefiles -----
```

R/2.12.0	R/2.15.1	R/3.2.1	R/3.4.0	Rstudio/R-2.15.1
----------	----------	---------	---------	------------------

R/2.12.1	R/2.9.1	R/3.2.3(default)	R/3.4.1	
----------	---------	------------------	---------	--

R/2.12.2	R/3.0.1	R/3.3.0	R/3.4.2	
----------	---------	---------	---------	--

R/2.13.2	R/3.1.1	R/3.3.3	Rstudio/0.98(default)	
----------	---------	---------	-----------------------	--

```
[rdtest@n2190 ~]$ module load R/3.4.2 > /dev/null
```

```
[rdtest@n2190 ~]$ module li
```

Currently Loaded Modulefiles:

- 1) ATS
- 3) bzip2/1.0.6
- 5) pcre/8.39
- 7) intel/17.0.1
- 9) R/3.4.2
- 2) zlib/1.2.8
- 4) xz/5.2.2
- 6) curl/7.49.1
- 8) gcc/4.9.3

# Rmpi

```
[rdtest@n2190 ~]$ module av openmpi
```

```
----- /u/local/gcc/4.9.3/modulefiles -----
```

```
openmpi/1.10
```

```
----- /u/local/Modules/modulefiles -----
```

```
openmpi/1.6_gcc-4.4 openmpi/2.0.2      openmpi/3.0.0
```

# Rmpi

```
[rdtest@n2190 ~]$ module av openmpi
```

```
----- /u/local/gcc/4.9.3/modulefiles -----
```

```
openmpi/1.10
```

```
----- /u/local/Modules/modulefiles -----
```

```
openmpi/1.6_gcc-4.4 openmpi/2.0.2      openmpi/3.0.0
```

```
[rdtest@n2190 ~]$ module load openmpi/1.10
```

```
[rdtest@n2190 ~]$ R
```

```
R version 3.4.2 (2017-09-28) -- "Short Summer"
```

```
[...]
```

# Rmpi

```
[rdtest@n2190 ~]$ module av openmpi
```

```
----- /u/local/gcc/4.9.3/modulefiles -----  
openmpi/1.10
```

```
----- /u/local/Modules/modulefiles -----  
openmpi/1.6_gcc-4.4 openmpi/2.0.2      openmpi/3.0.0
```

```
[rdtest@n2190 ~]$ module load openmpi/1.10
```

```
[rdtest@n2190 ~]$ R
```

```
R version 3.4.2 (2017-09-28) -- "Short Summer"
```

```
[...]
```

```
> install.packages("Rmpi",type="source",configure.args=c('--with-mpi=/u/local/mpi/openmpi/1.10.2/gcc-4.9.3','--with-Rmpi-type=OPENMPI'),dependencies=TRUE)
```

```
Installing package into '/u/home/r/rdtest/R/x86_64-pc-linux-gnu-library/3.4'  
(as 'lib' is unspecified)
```

```
--- Please select a CRAN mirror for use in this session ---
```

```
[...]
```

```
* DONE (Rmpi)
```

```
* >
```

# Rmpi

```
[rdtest@n2190 ~]$ mkdir Rmpi_test
[rdtest@n2190 ~]$ cd Rmpi_test
[rdtest@n2190 ~/Rmpi_test]$ cp /u/home/r/rdtest/R/x86_64-pc-linux-gnu-library/3.4/Rmpi/Rprofile .Rprofile
[rdtest@n2190 ~/Rmpi_test]$ cp /u/local/apps/submit_scripts/R/Rmpi/*.R .
[rdtest@n2190 ~/Rmpi_test]$ ls -lta
total 44
-rw-r--r-- 1 rdtest systems 356 Jan 23 17:37 test_Rmpi.R
drwxr-xr-x 2 rdtest systems 4096 Jan 23 17:37 .
-rw-r--r-- 1 rdtest systems 714 Jan 23 17:37 simplePI.R
-rw-r--r-- 1 rdtest systems 2070 Jan 23 17:27 .Rprofile
drwxr-x--- 112 rdtest systems 24576 Jan 23 17:25 ..
[rdtest@n2190 ~/Rmpi_test]$ cat $PE_HOSTFILE
n2190 8 inter_msa.q@n2190 UNDEFINED
n2187 6 inter_msa.q@n2187 UNDEFINED
[rdtest@n2190 ~/Rmpi_test]$ time `which mpirun` -np $NSLOTS R CMD BATCH ./test_Rmpi.R
[...]
dauria@n2187_hoffman2:~> top -u rdtest
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
727	rdtest	20	0	702m	63m	8920	R	39.0	0.2	0:01.18	/u/local/apps/R/3.4.2/gcc-4.9.3_MKL-2017.0/lib64/R/bin/exec/R -f ./test_Rmpi.R --rest
711	rdtest	20	0	702m	63m	8924	R	38.7	0.2	0:01.17	/u/local/apps/R/3.4.2/gcc-4.9.3_MKL-2017.0/lib64/R/bin/exec/R -f ./test_Rmpi.R --rest
720	rdtest	20	0	702m	65m	8912	R	38.7	0.2	0:01.17	/u/local/apps/R/3.4.2/gcc-4.9.3_MKL-2017.0/lib64/R/bin/exec/R -f ./test_Rmpi.R --rest
724	rdtest	20	0	702m	62m	8916	R	38.7	0.2	0:01.17	/u/local/apps/R/3.4.2/gcc-4.9.3_MKL-2017.0/lib64/R/bin/exec/R -f ./test_Rmpi.R --rest
725	rdtest	20	0	702m	63m	8916	R	38.7	0.2	0:01.17	/u/local/apps/R/3.4.2/gcc-4.9.3_MKL-2017.0/lib64/R/bin/exec/R -f ./test_Rmpi.R --rest
723	rdtest	20	0	702m	65m	8920	R	37.7	0.2	0:01.14	/u/local/apps/R/3.4.2/gcc-4.9.3_MKL-2017.0/lib64/R/bin/exec/R -f ./test_Rmpi.R --rest

# Rmpi

```
[rdtest@login3 ~/Rmpi_test]$ cp /u/local/apps/submit_scripts/R/Rmpi/run_Rmpitest.sh ./
[rdtest@login3 ~/Rmpi_test]$ qsub run_Rmpitest.sh
JSV: No h_data is set; setting default h_data=1G (if this value is too small, the job will fail)
Your job 218797 ("run_Rmpitest.sh") has been submitted
[rdtest@login3 ~/Rmpi_test]$ cat Rmpitest.Rout
[...]
master (rank 0 , comm 1) of size 12 is running on: n7243
slave1 (rank 1 , comm 1) of size 12 is running on: n7243
slave2 (rank 2 , comm 1) of size 12 is running on: n7273
slave3 (rank 3 , comm 1) of size 12 is running on: n7273
...
slave10 (rank 10, comm 1) of size 12 is running on: n7207
slave11 (rank 11, comm 1) of size 12 is running on: n7207
[...]
> # Tell all slaves to close down, and exit the program
> mpi.close.Rslaves()
[1] 1
> mpi.quit()
[rdtest@login3 ~/Rmpi_test]$
```

# R parallel package

Let's start w/ lapply:

```
[rdtest@n2192 test]$ which R  
/u/local/apps/R/3.4.2/gcc-4.9.3_MKL-2017.0/bin/R  
[rdtest@n2192 test]$ R  
[...]  
> lapply(c(1,4), function(x) c(x, x^2, x^3))  
[[1]]  
[1] 1 1 1  
  
[[2]]  
[1] 4 16 64
```

`lapply` takes one parameter (a vector/list), feeds that variable into the function, and returns a list

# R parallel package

```
[rdtest@login1 ~]$ qrsh -l exclusive
[rdtest@n9768 ~]$ module load R/3.4.2
[rdtest@n9768 ~]$ which R
/u/local/apps/R/3.4.2/gcc-4.9.3_MKL-2017.0/bin/R
[rdtest@n9768 ~]$ R
[...]
> library(parallel)
> no_cores <- detectCores() - 1
> cl <- makeCluster(no_cores, type="FORK")
> parLapply(cl, c(2,3), function(exponent) 2^exponent)
[[1]]
[1] 4

[[2]]
[1] 8

> stopCluster(cl)
```

parLapply takes one parameter (a vector/list), feeds that variable into the function, and returns a list while splitting the work on the cluster

# R implicit parallelism

- Most recent versions of R on the cluster are built w/ gcc/4.9.3 & Intel MKL/2017
- take advantage of multiple cores available on a single cluster node (i.e., computer)
- The default number of threads will be chosen by the OpenMP software, but can be controlled by setting OMP\_NUM\_THREADS or MKL\_NUM\_THREADS

```
[rdtest@login1 ~]$ cp /u/local/apps/submit_scripts/R/R-benchmark-25.R .
[rdtest@login1 ~]$ qrsh -l exclusive
[rdtest@n9768 ~]$ module load R/3.4.2
[rdtest@n9768 ~]$ which R
/u/local/apps/R/3.4.2/gcc-4.9.3_MKL-2017.0/bin/R
[rdtest@n9768 ~]$ R
[...]
> source("R-benchmark-25.R")
```

# R implicit parallelism

- Most recent versions of R on the cluster are built w/ gcc/4.9.3 & Intel MKL/2017
- take advantage of multiple cores available on a single cluster node (i.e., computer)
- The default number of threads will be chosen by the OpenMP software, but can be controlled by setting OMP\_NUM\_THREADS or MKL\_NUM\_THREADS

```
[rdtest@login1 ~]$ cp /u/local/apps/submit_scripts/R/R-benchmark-25.R .
[rdtest@login1 ~]$ qrsh -l exclusive
[rdtest@n9768 ~]$ module load R/3.4.2
[rdtest@n9768 ~]$ which R
/u/local/apps/R/3.4.2/gcc-4.9.3_MKL-2017.0/bin/R
[rdtest@n9768 ~]$ R
[...]
> source("R-benchmark-25.R")
```

# R on jupyter notebooks

```
dauria@alba:~/M280> wget https://gitlab.idre.ucla.edu/dauria/jupyter-notebook/raw/master/h2jupynb  
-bash: wget: command not found  
dauria@alba:~/M280> curl -O https://gitlab.idre.ucla.edu/dauria/jupyter-notebook/raw/master/h2jupynb  
[...]  
dauria@alba:~/M280> chmod u+x h2jupynb  
dauria@alba:~/M280> ./h2jupynb -u ic9g001  
Your Hoffman2 user name is ic9g001  
The time in hours is 2  
The memory in GB per slots is 1  
The number of slots is 1  
The version of python for the notebook is 2.7.13  
The port is 9300  
The directory on Hoffman2 is $HOME  
ic9g001@hoffman2.idre.ucla.edu's password:  
[...]  
[ic9g001@login1 ~]$ jupyter notebook --port=9300
```

```
[ic9g001@login1 ~]$ cp /u/local/apps/submit_scripts/R/  
jupyter_R_example_approx_of_pi.ipynb ./  
[ic9g001@login1 ~]$
```

# R on jupyter notebooks



# R on jupyter notebooks



A method for computing  $\pi$  is to draw a circle inscribed in a square, and randomly place dots in the square. The ratio of dots inside the circle to the total number of dots will approximately equal  $\pi/4$

```
In [2]: n <- 1000000
x <- runif(n); y <- runif(n)
inside <- x^2+y^2 <= 1
pi <- 4*sum(inside)/n
idx <- sample.int(trunc(n/(log(n))), replace=FALSE)
x <- x[idx]; y <- y[idx]; inside <- inside[idx]
plot(x,y, col=ifelse(inside,"blue","red"), cex = 0.5, main = sprintf("Bootstrap approx of pi using\n n= %s"))
print(pi)
```

```
[1] 3.1425
```

Bootstrap approx of pi using  
n= 1e+06 random samples, pi ~ 3.14250

