



A Quick Introduction to the Amarel Cluster

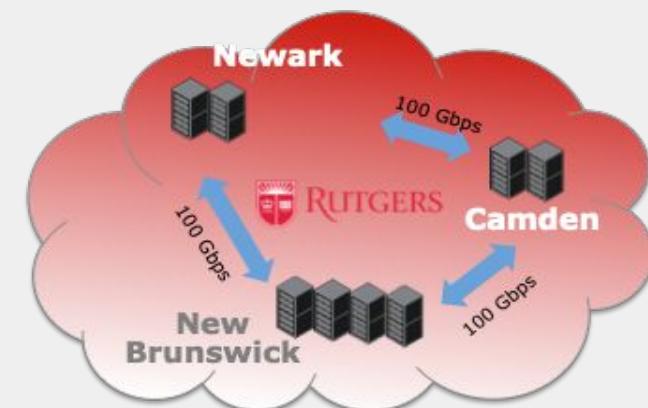
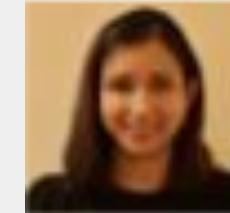
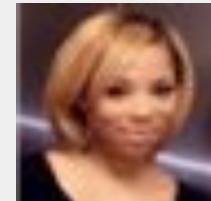
Bala Desinghu

Office of Advanced Research Computing (OARC)
Rutgers University



Office of Advanced Research Computing (OARC)

Bala Desinghu, PhD
OARC, Rutgers



Amarel for Rutgers

Saul Amarel (1928 - 2002) D.Sc.
Alan M. Turing Prof. of Computer
Science at Rutgers



Caliburn for NJ



OARC Support Channels

Bala Desinghu, PhD
OARC, Rutgers

One-on-One help (email, zoom, office hours)

help@oarc.rutgers.edu

Regular workshops on all three locations

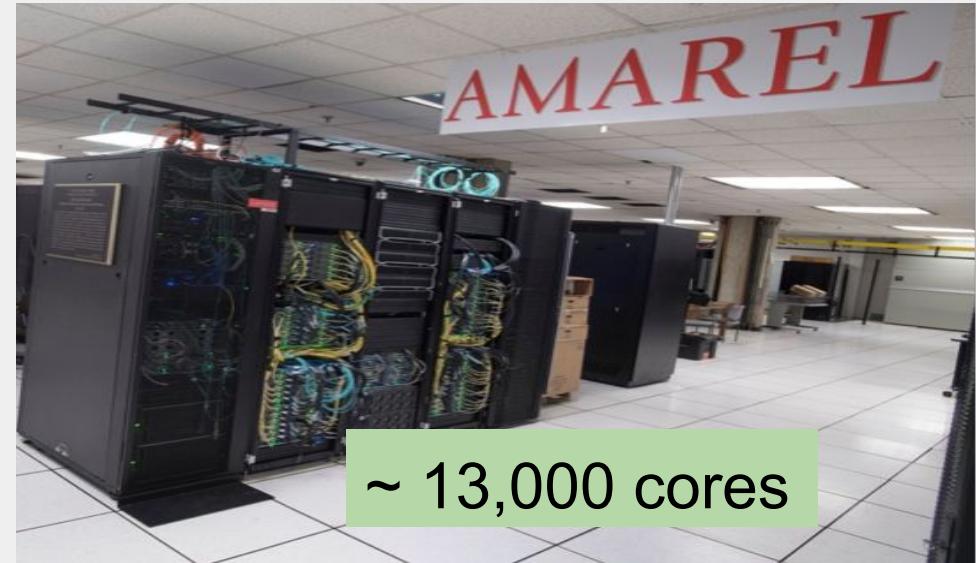
Camden, Newark, Piscataway

Training materials and domain specific examples



Why you need HPC?

Bala Desinghu, PhD
OARC, Rutgers

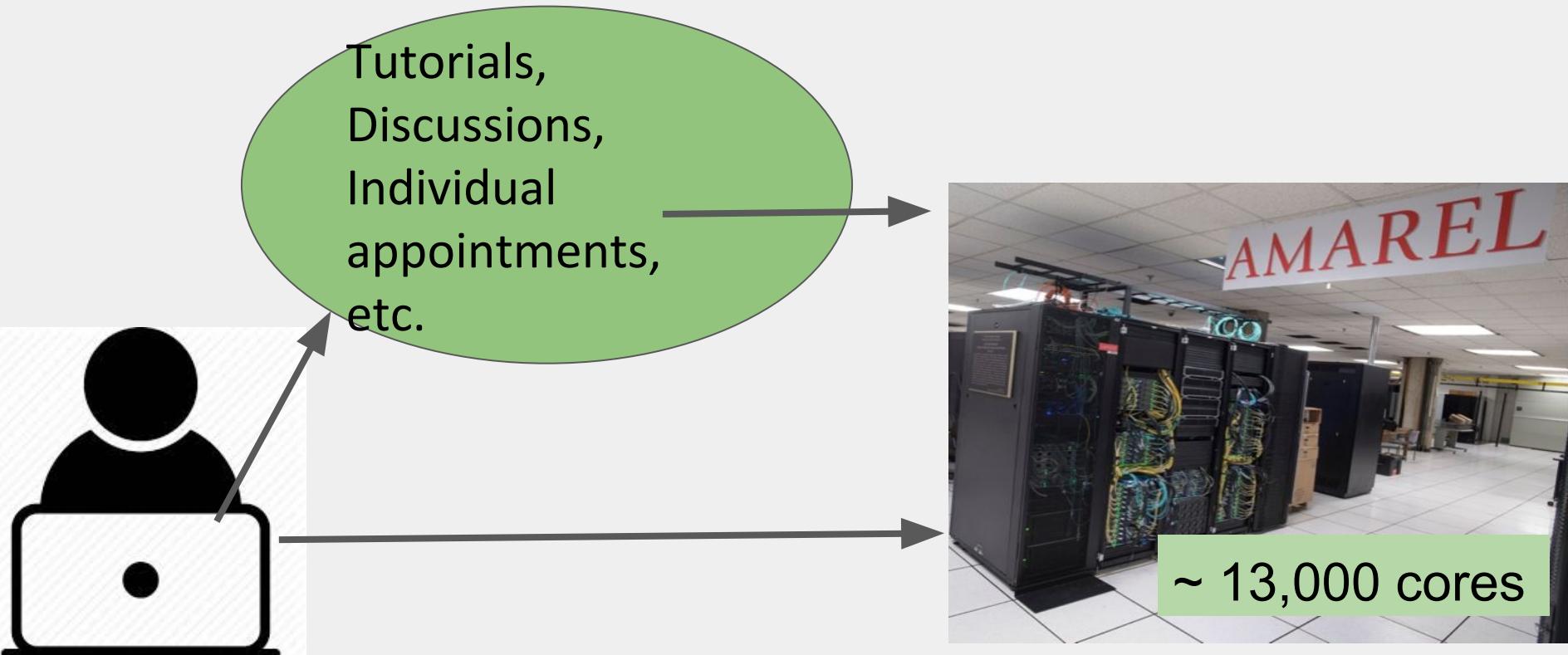


Computation done on Amarel in one day will run for about 15 years on a laptop (dual core).



How to go from laptop to HPC?

Bala Desinghu, PhD
OARC, Rutgers





Common Challenges

Bala Desinghu, PhD
OARC, Rutgers

Getting started (this is our focus today)

How to connect with the HPC?

What programs are available?

How much disk space can I have on the cluster?

How secure is my data?

How to run a job?...



More Involved Questions

Bala Desinghu, PhD
OARC, Rutgers

Questions specific to research project

How to scale my application?

Can I automate my workflow?

How to build portable a application?

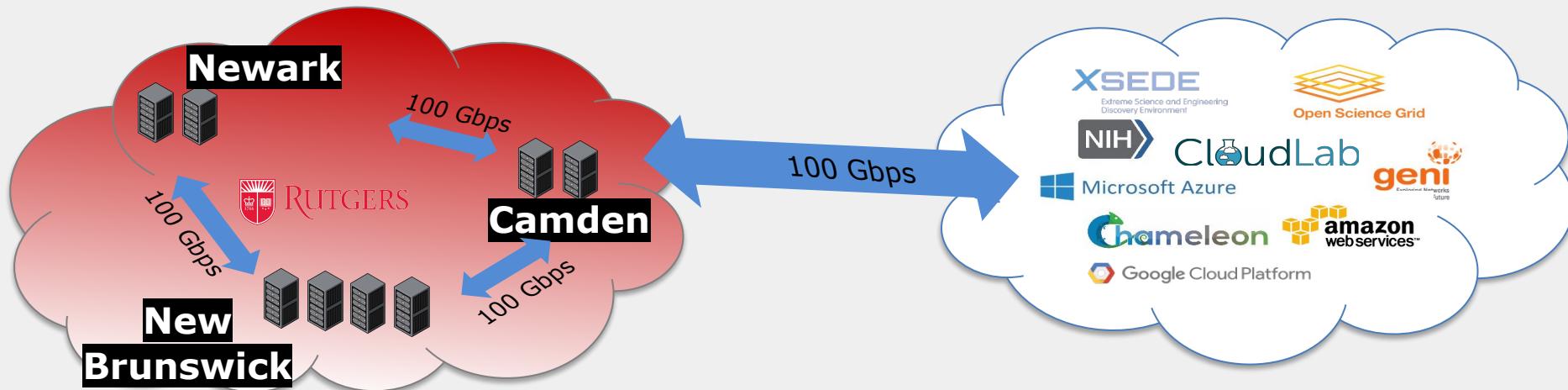
How to use the hardware accelerators?

....



One-Rutgers ACI Eco System

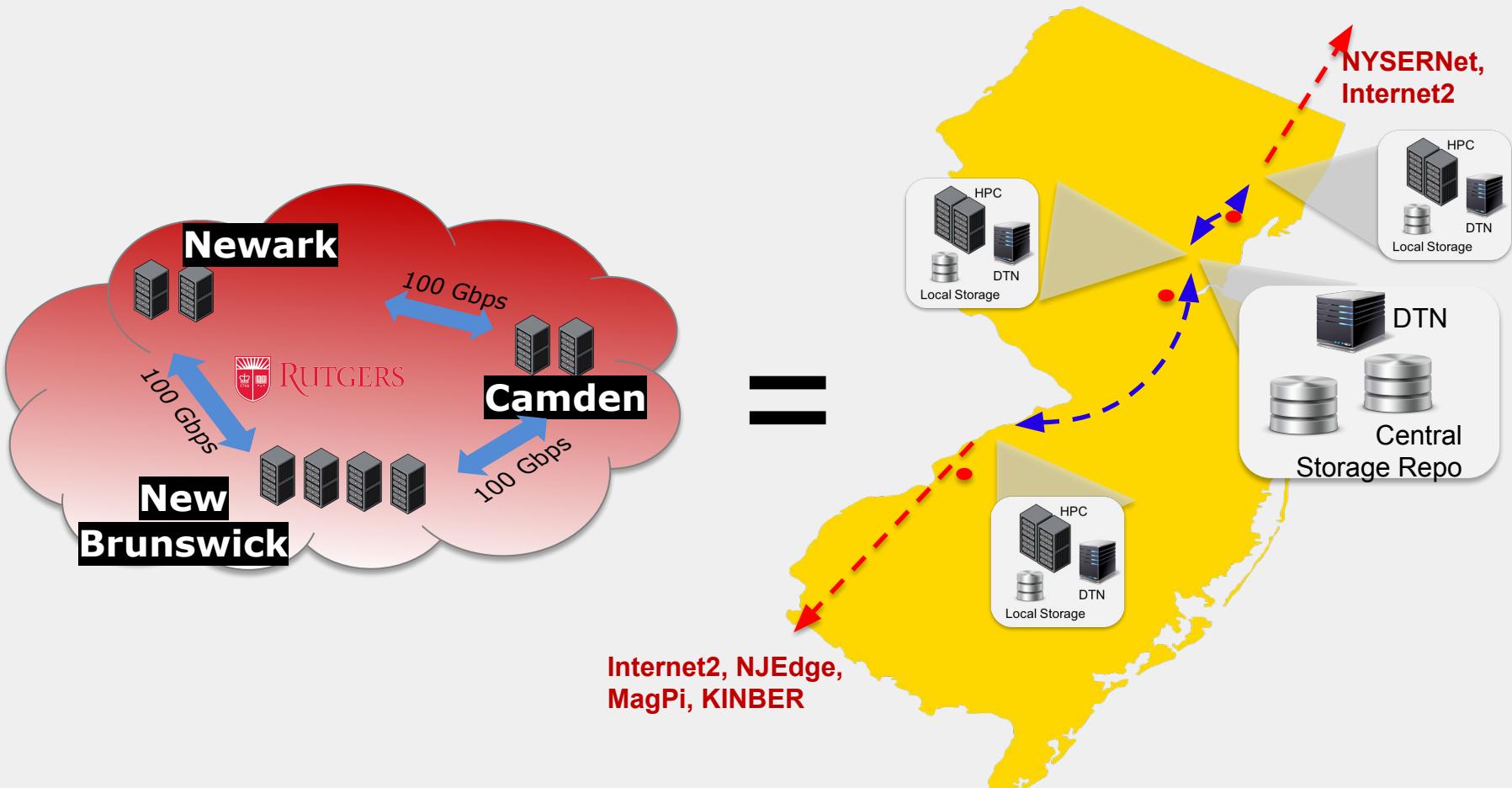
Bala Desinghu, PhD
OARC, Rutgers





Amarel Cluster - Geo View

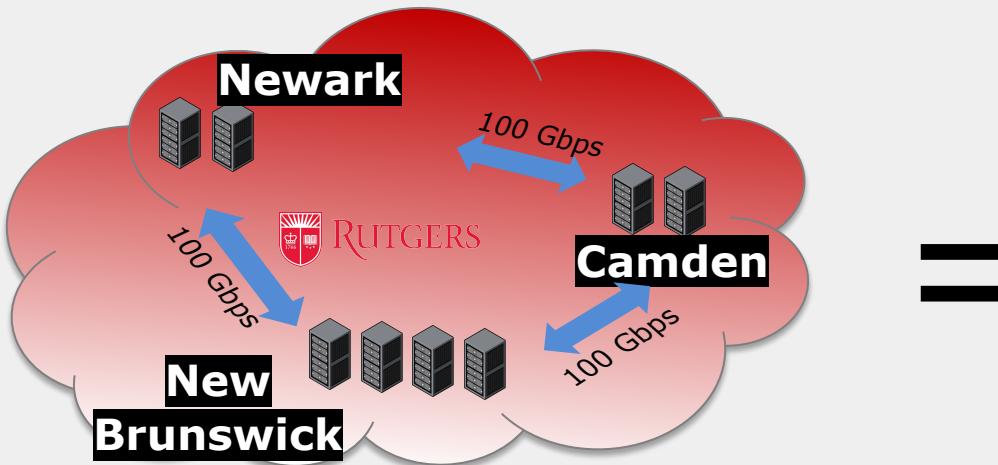
Bala Desinghu, PhD
OARC, Rutgers



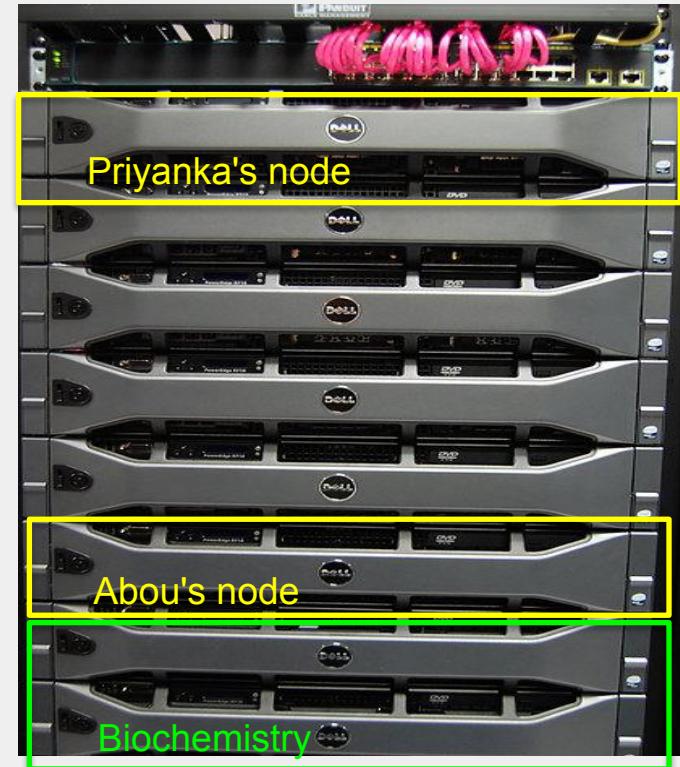


Amarel Cluster - Community Model

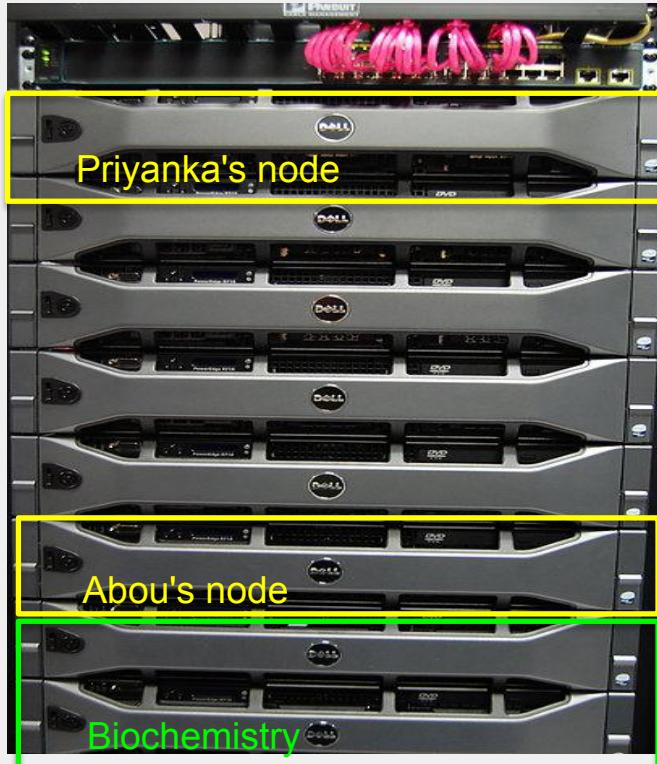
Bala Desinghu, PhD
OARC, Rutgers



HPC distributed on multiple campuses



Community Model



- Using Amarel is FREE, but about 50% of the cluster's compute nodes are "owned" by researchers
- Owners buy dedicated (highest priority) access to a nodes/GPUs using an low-cost 4-year plan
- OARC pays for power, cooling, repair or replacement, system administration, networking...
- When an owned node is not in use, general/main partition jobs can use idle CPU cores
- General/main partition jobs can be preempted by owner jobs, but that should occur rarely because only about 50% owned



Amarel Cluster - User's View

Bala Desinghu, PhD
OARC, Rutgers

CPU, GPU, FPGA

- Intel (Xeon, Cascade Lake, Skylake, Haswell, Broadwell, Sandy Bridge, Ivy Bridge...) nodes
- Memory (128 GB, 192 GB, 256 GB, 1.5 TB, ...)
- CPUs (16, 24, 40, 56 cores)

Storage (~5 PB General Parallel File System)

- 100 GB storage in /home with weekly snapshots
- 20 TB temporary, high I/O storage in /scratch
- 250 to 1000 GB fast local storage at /mnt/scratch





Technical Details (Sample)

- 2 40-core Xeon 6230 nodes each with 2 Nvidia Volta V100 GPUs onboard
- 4 24-core Xeon 4116 nodes each with 8 Nvidia RTX 2080Ti GPUs onboard
- 40+ CPU-only nodes, each with 40 Xeon 6230 (Cascade Lake) cores + 192 GB RAM
- 70+ CPU-only nodes, each with 32 Xeon 6130 (Skylake) cores + 192 GB RAM
- 52 CPU-only nodes, each with 28 Xeon e5-2680v4 (Broadwell) cores + 128 GB RAM
- 20 CPU-only nodes, each with 28 Xeon e5-2680v4 (Broadwell) cores + 256 GB RAM
- 4 28-core Xeon e5-2680v4 nodes each with 2 Nvidia Pascal P100 GPUs onboard
- 2 high-memory nodes, each with 56 e7-4830v4 (Broadwell) cores + 1.5 TB RAM
- 53 CPU-only nodes, each with 16 Xeon e5-2670 (Sandy Bridge) cores + 128 GB RAM
- 5 CPU-only nodes, each with 20 Intel Xeon e5-2670 (Ivy Bridge) cores + 128 GB RAM
- 26 CPU-only nodes, each with 24 Intel Xeon e5-2670 (Haswell) cores + 128 GB RAM
- 4 CPU-only nodes, each with 16 Intel Xeon e5-2680 (Broadwell) cores + 128 GB RAM
- 3 12-core Xeon e5-2670 nodes with 8 Nvidia Tesla M2070 GPUs onboard
- 2 28-core Xeon e5-2680 nodes with 4 Quadra M6000 GPUs onboard
- InfiniBand FDR (56 Gbps) and EDR (100 Gbps)
- 100 GB storage in /home with weekly snapshots
- 20 TB temporary, high I/O storage in /scratch
- 250 to 1000 GB fast local storage at /mnt/scratch

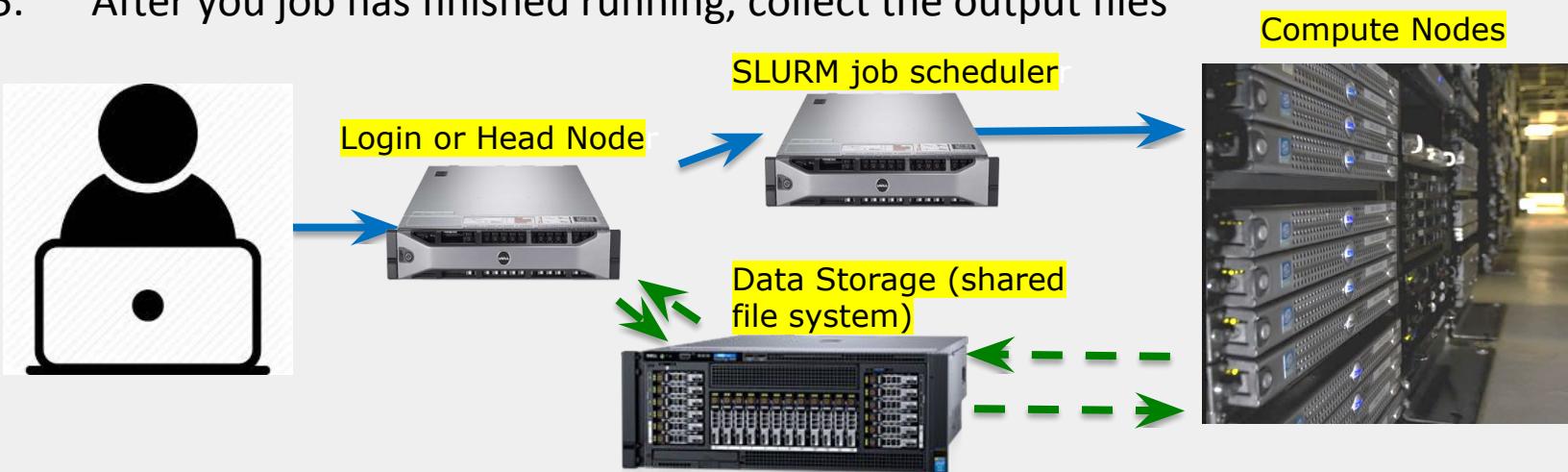




How to run a compute job?

Bala Desinghu, PhD
OARC, Rutgers

1. Connect to a cluster (SSH or web access), setup your software to run there
2. Move input files/data to the cluster (rsync, scp, sftp)
3. Create a job script (requesting only the hardware you need)
4. Submit your job script to the cluster's resource manager
5. After your job has finished running, collect the output files





Log-in to Amarel

Connect to your Amarel cluster account:



In OS X, launch a Terminal and SSH to Amarel:

Applications > Utilities > Terminal

ssh [your NetID]@amarel.rutgers.edu



In Linux or Unix, launch a Terminal and SSH to Amarel:

(in Ubuntu, Ctrl-Alt-T)

ssh [your NetID]@amarel.rutgers.edu



In Windows, launch an SSH client (e.g., PuTTY or MobaXterm):

Host Name: amarel.rutgers.edu

Login as: [your NetID]

Off campus? Connect to the campus VPN first:

<https://soc.rutgers.edu/vpn>



Shells and Bash

Bash (Bourne-again shell) = default shell (command processor) in Linux.

```
[gc563@amarell ~]$ which bash
/usr/bin/bash
[gc563@amarell ~]$ bash --version
GNU bash, version 4.2.46(1)-release (x86_64-redhat-linux-gnu)
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
[gc563@amarell ~]$
```

Bash uses 2 configuration files, .bashrc & .bash_profile and only .bashrc is processed every time a new shell instance is started.

Every time to connect to a different computer/server/node, you're starting a new shell.

Compute Nodes





Storage

/home/<netid> = 100 GB (backed Up)

mmquota -u <netid> --block-size=auto cache

/scratch/<netid> = 20 TB (no backup and 90 days purge policy)

mmquota -u <netid> --block-size=auto scratch

/project/<yourgroupdir> = purchased storage (backed up)

mmquota --block-size=auto -j <urgroupdir>

projectsp(n,c)

The Scratch Filesystem



- Located at `/scratch/NetID`
- Physically located near execution nodes (varies)
- Temporary work directory for all jobs
- Specialized high-performance hardware
- Designed to handle very high I/O activity and large files
- 20 TB storage space
- Move files in, run jobs, move files out



Protecting Important Data

Bala Desinghu, PhD
OARC, Rutgers

Free, backed-up storage options available to Rutgers students, staff, and faculty:

- Unlimited storage in `box.rutgers.edu`, <https://box.rutgers.edu> for all students, staff, faculty (15 GB file size limit, but larger files can be split before uploading)
- 5 TB of space in Office365 OneDrive,
<https://it.rutgers.edu/rutgers-connect/knowledgebase/onedrive-for-business> for all staff, faculty, and RBHS students (10 GB file size limit, but larger files can be split before uploading)
- Unlimited storage in Google Drive for ScarletApps users,
<https://it.rutgers.edu/scarletapps> (5 TB file size limit, but larger files can be split before uploading)
- Backed-up /projects storage on Amarel, \$150/TB for 4 years for compute node owners

Also, be mindful of where it's safe to store your data:

<https://box.rutgers.edu/data-classification-and-storage-matrix>



Software Support

Bala Desinghu, PhD
OARC, Rutgers

- System wide Installations (/opt/sw/...)
 - Commercial and open source packages
- Community Installations (/project/community/...)
 - Open source packages
- Personal installations (/home, /project/<urgroupdir>)
 - Commercial and open source packages

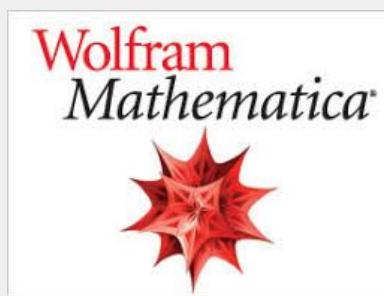
Type of Installations

- Compile from source
- Binaries
- Singularity images



Existing packages

BLAST, Bowtie, BWA, CUDA Toolkit, Gaussian, GCC compilers, GROMACS, Intel compilers, LAMMPS, MATLAB, NAMD, PGI compilers, SAMtools, TopHat, TrinityRNAseq, and many more...





Software Environment

- To get a list of available software:
 - **module avail**
 - **module spider** (for more options & details)
- Check for prerequisites:
 - **module spider intel/17.0.4**
- Load (add) modules:
 - **module load intel/17.0.4 mvapich2/2.1**
- List your loaded modules:
 - **module list**
- To clear-out your added modules:
 - **module purge**

You can add your 'module load ...' command(s) to your .bashrc file

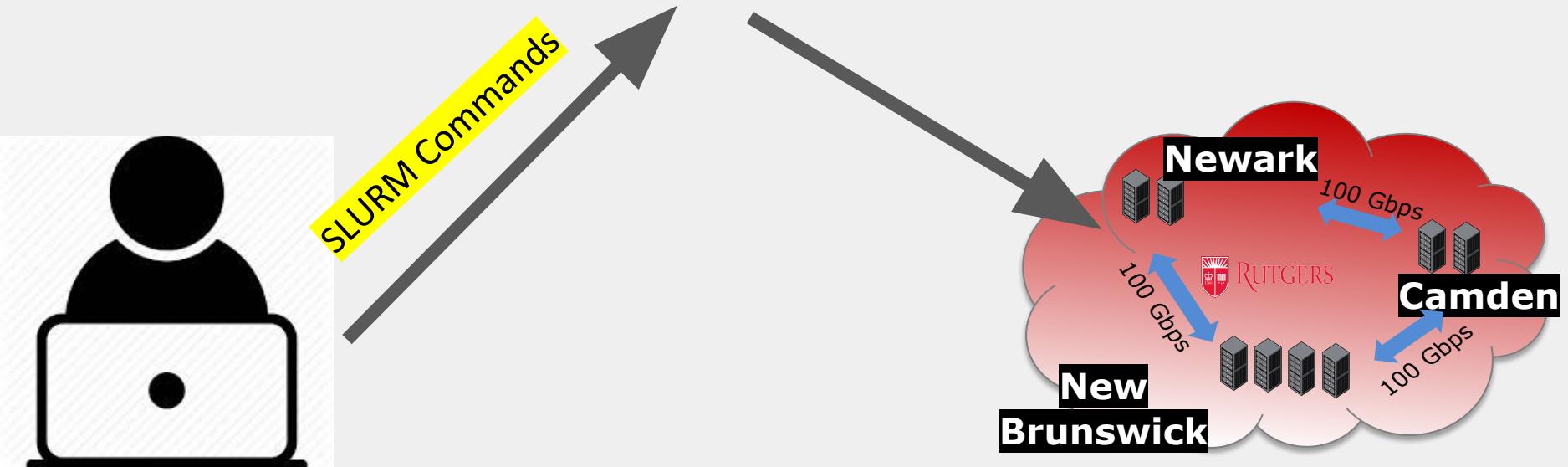


Resource Management

Bala Desinghu, PhD
OARC, Rutgers

SLURM

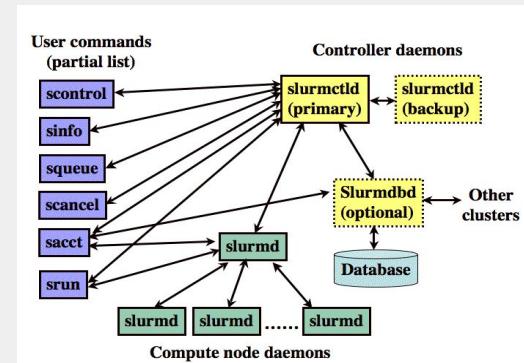
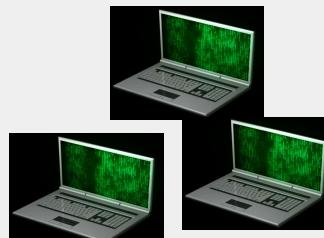
resource manager / job scheduler



Resource Management

SLURM = resource manager / job scheduler

- Enables scripting of computational tasks
- SLURM runs these tasks on compute nodes and returns the results (output files)
- If the cluster is full, SLURM holds your tasks and runs them when the resources are available
- SLURM ensures fair sharing of cluster resources (policy enforcement)





Basic SLURM commands

Bala Desinghu, PhD
OARC, Rutgers

<u>Command(s)</u>	<u>Description</u>
sinfo -a --summarize	View nodes and partition info
sbatch job-script [options]	Submit/setup a batch job
srun [options] program_name	Run a program (exe, application)
squeue -u NetID	Check status of job submissions
sstat -u jobID	Check status of a running job
sacct --format [options] -j jobID	See accounting details of current and completed jobs



<https://slurm.schedmd.com/pdfs/summary.pdf>



Copy an Example Job

Copy example files to your /home directory:

cd /home/[NetID]

cp -r /projects/oarc/users/training/intro.amarel .

cd intro.amarel

Don't forget the “.”

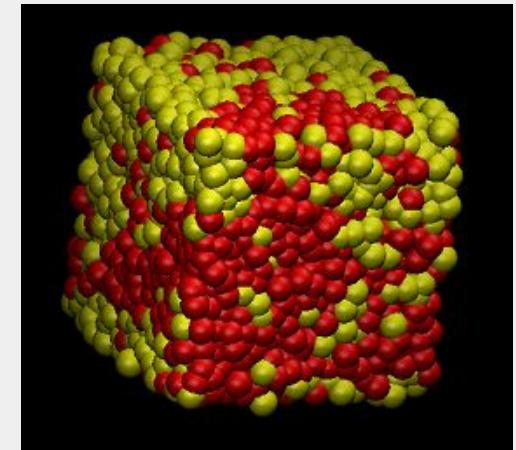
ls

It means copy the files “here” (to my current location)

```
[gc563@amarel1 intro.amarel]$ ls
amber.example          gethostname.mpi.c      matlab.example
bashrc.modified.example hello_world_mpi.c    namd.example
bashrc.new               hello_world_openmp.c pi.integ.c
gaussian.example        lammps.example       pi.integ.mpi.c
gethostname.c           mathematica.example run.hello_world_openmp
```

Run an Example Job

- LAMMPS is a molecular dynamics simulation program
- Cooling of a binary mixture
- 5,000 atoms (L-J interactions)
- 500,000 0.005 ps steps



```
cd intro.amarel/lammps.example
```

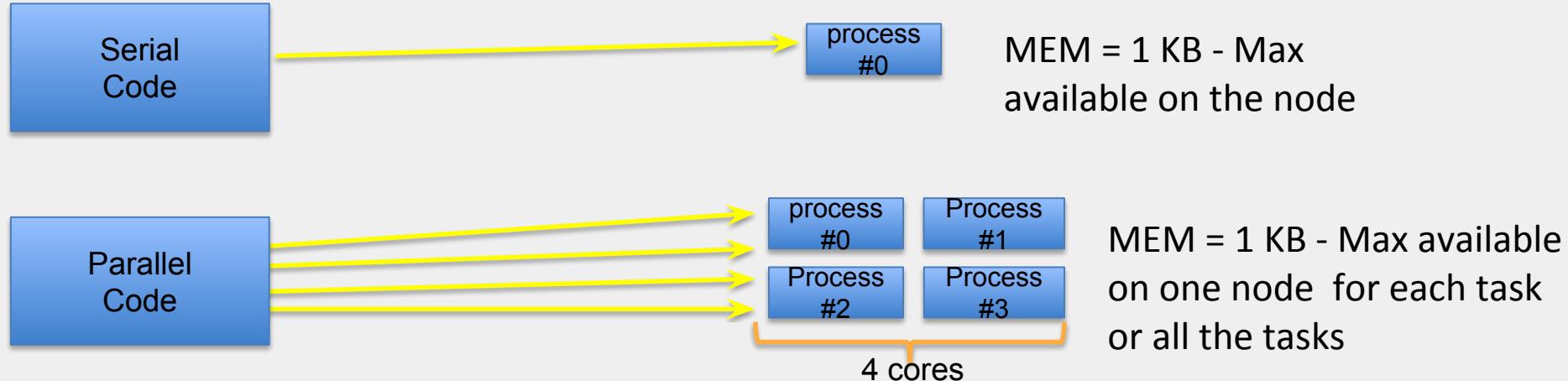
Edit run.lammps.binary (optional), then run the job:

```
sbatch run.lammps.binary
```

```
squeue -u [NetID]
```

```
sstat --format=MaxRSS,MaxDiskRead,MaxDiskWrite -j [JobID]
```

Serial vs Parallel Computing



- Nearly all software is becoming increasingly parallel
- When you run a parallel program, copies of the parallel parts of the code are distributed amongst the available compute cores
- Each separate task can send & receive data among the other tasks (**MPI & OpenMP**)
- If tasks on separate nodes must communicate for the overall program to proceed, high-speed networking is needed (only MPI)



Job Script

```
#!/bin/bash
#SBATCH --partition=main
#SBATCH --job-name=CH3Phx
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=1
#SBATCH --mem=12G
#SBATCH --time=1:00:00
#SBATCH --output=slurm.%N.%j.out
#SBATCH --error=slurm.%N.%j.err
#SBATCH --mail-user=[NetID]@rutgers.edu
#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH --requeue
#SBATCH --export=ALL

./my-exe input > output
sacct --format MaxRSS,Elapsed -j $SLURM_JOBID
```

This is a “long” example. You likely will not need all of these options.



Job Script (MPI + GPUs)

```
#!/bin/bash
#SBATCH --partition=gpu
#SBATCH --job-name=CH3Phx
#SBATCH --nodes=2
#SBATCH --ntasks=16
#SBATCH --gres=gpu:2
#SBATCH --cpus-per-task=1
#SBATCH --mem=120G
#SBATCH --time=5:00:00
#SBATCH --output=slurm.%N.%j.out
#SBATCH --error=slurm.%N.%j.err
#SBATCH --mail-user=[NetID]@rutgers.edu
#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH --requeue
#SBATCH --export=ALL

srun --mpi=pmi2 my-exe input > output
sacct --format MaxRSS,Elapsed -j $SLURM_JOBID
```

This is a “long” example. You likely will not need all of these options.



Job Script (OpenMP)

```
#!/bin/bash
#SBATCH --clusters=amarel
#SBATCH --partition=nonpre
#SBATCH --job-name=Efxea
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --mem=120G
#SBATCH --time=5:00:00
#SBATCH --output=slurm.%N.%j.out
#SBATCH --error=slurm.%N.%j.err
#SBATCH --mail-user=[NetID]@rutgers.edu
#SBATCH --mail-type=BEGIN,END,FAIL
#SBATCH --export=ALL

OMP_NUM_THREADS=16
srun my-exe -t 16 input > output
sacct --format MaxRSS,Elapsed -j $SLURM_JOBID
```

Note: --mem=0 means "use all available RAM" but that's only the RAM that's not allocated to other jobs



Defaults and Limits

Default settings (in case you don't specify something):

--ntasks=1

--cpus-per-task=1

--mem-per-cpu=4G (per CPU core requested)

--time=00-00:02:00

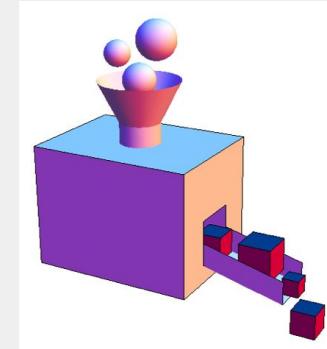
Limits / upper bounds:

- Max # CPUs in-use per user for 'main' partition jobs = 504
- Max # of jobs a user can submit = 500 (this also applies to job arrays)
- Currently no limit for owners, but system-wide limit of 10,000 jobs creates a variable limit⁶²

Job Arrays

Processing many input files with the same exe & job requirements

```
#SBATCH --partition=main
#SBATCH --job-name=ARRx1
#SBATCH --array=1-240%100
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=4
#SBATCH --mem=6000
#SBATCH --time=8:00:00
#SBATCH --requeue
```



Input files:
1.in, 2.in, 3.in, ...

Output files:
1.out, 2.out, 3.out, ...

```
OMP_NUM_THREADS=4
srun my-exe ${SLURM_ARRAY_TASK_ID}.in > ${SLURM_ARRAY_TASK_ID}.out
```

Note: sacct & sstat commands won't work for array jobs



Setup E-mail Notifications

Bala Desinghu, PhD
OARC, Rutgers

```
#SBATCH --mail-user=[NetID]@rutgers.edu
```

```
#SBATCH --mail-type=[type]
```

Where [type] can be one of the following:

NONE, BEGIN, END, FAIL, REQUEUE, ALL

Note: only [NetID]@rutgers.edu will work

Non-Rutgers e-mail addresses or
[NetID]@xxxxx.rutgers.edu will not work





Killing Batch Jobs

Kill just one, or just a few jobs:

```
scancel [jobID] [jobID] [jobID] ...
```

Kill all of your jobs:

```
scancel --user=[NetID]
```

Kill all of your *queued* jobs:

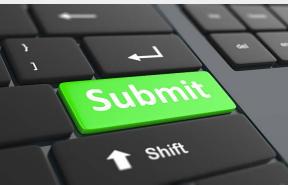
```
scancel --user=[NetID] --state=PENDING
```

Kill all of your *running* jobs:

```
scancel --user=[NetID] --state=RUNNING
```



Batch vs. Interactive



Batch Job

Interactive Job

sbatch *job-script*

Starts when requested resources are available

Runs “in the background”

Use **srun** to launch tasks inside your script

Terminate batch jobs using
scancel *jobID*

Useful for jobs that will run for a long time, and for jobs that don’t require interaction or supervision

srun [required resources] *your.exe*

Starts when requested resources are available

You are actively logged-in (running a shell) on a compute node

Terminate interactive jobs by simply logging-out (using **exit** or **CTRL-D**)

Useful for testing, compiling code, computational steering, etc.



Interactive Jobs

Useful for debugging applications or running short tests

Single-core / serial task example:

```
srun --time=2:00:00 --pty bash
```

```
exit (when finished)
```

Multi-core / parallel task example:

```
srun --ntasks=4 --mpi=pmi2 -time=2:00:00 --pty bash
```

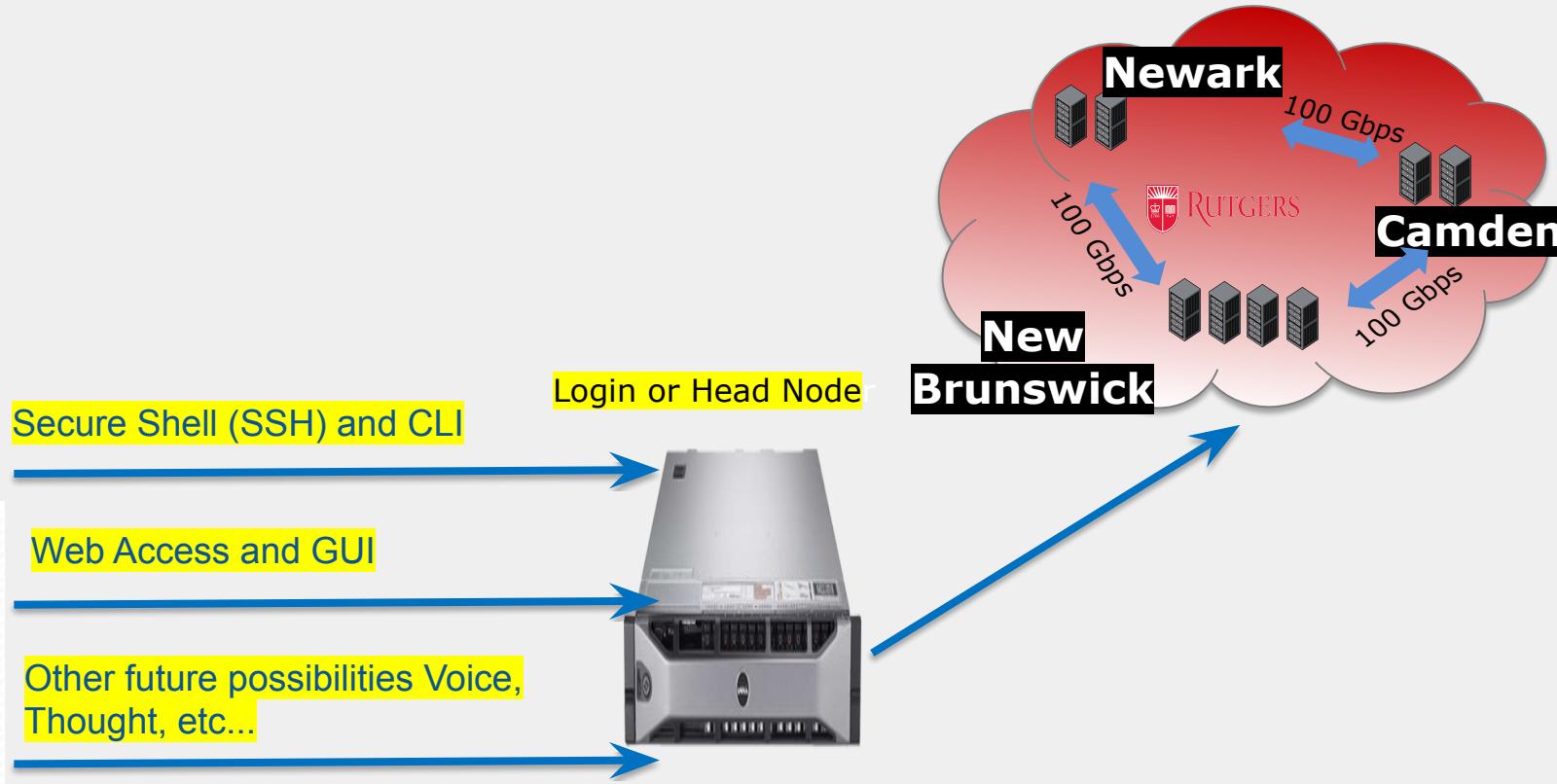
```
exit (when finished)
```





User Interface

Bala Desinghu, PhD
OARC, Rutgers

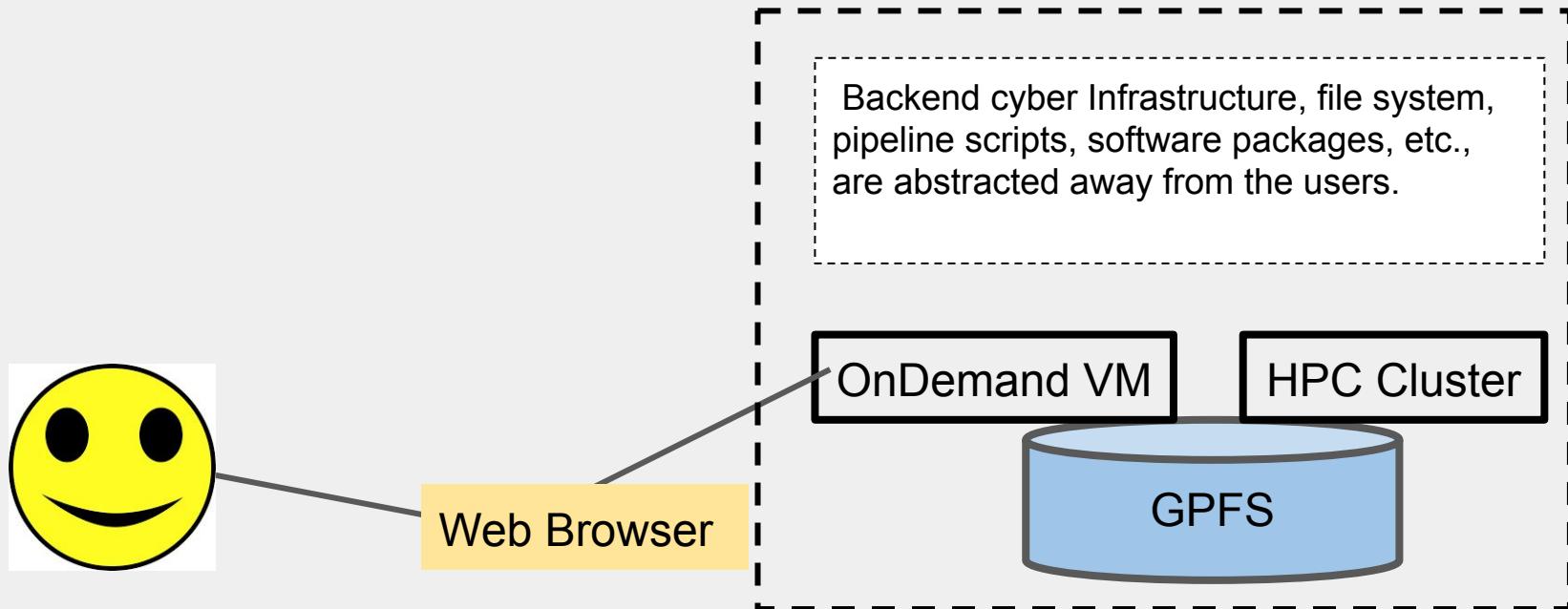




Web Access - Open OnDemand

Bala Desinghu, PhD
OARC, Rutgers

<https://ondemand.hpc.rutgers.edu/>





Web Access - Open OnDemand

Bala Desinghu, PhD
OARC, Rutgers

<https://ondemand.hpc.rutgers.edu/>

The screenshot shows the main dashboard of the Open OnDemand web interface. At the top, there is a navigation bar with links for "Open OnDemand", "Files", "Jobs", "Clusters", "Interactive Apps", and a search icon. Below the navigation bar, the "OnDemand" logo is prominently displayed. A sub-header below the logo reads "OnDemand provides an integrated, single access point for all of your HPC resources." The main content area is currently empty, showing a large white space.

The screenshot shows the "Interactive Apps" menu of the Open OnDemand web interface. The menu is organized into several sections: "Clusters", "Interactive Apps", and "Desktops". Under "Interactive Apps", there is a list of available applications, each represented by a blue circular icon and a label. Under "Desktops", there are two listed: "Amarel Desktop" and "Perceval Desktop". Under "Servers", there are four listed: "Comp. Gen. Jupyter", "Jupyter Notebook 2", "Jupyter Notebook 3", and "RStudio Server".

Clusters	Interactive Apps	Desktops	Servers
	<ul style="list-style-type: none">Example - Silly BatchRNA Seq Batch Jobcellrangercellranger-aggrcellranger-mkfastq	<ul style="list-style-type: none">Amarel DesktopPerceval Desktop	<ul style="list-style-type: none">Comp. Gen. JupyterJupyter Notebook 2Jupyter Notebook 3RStudio Server



Web Access - fastx

Go to <https://amarel.rutgers.edu:3443>

Confirm/accept the security exception

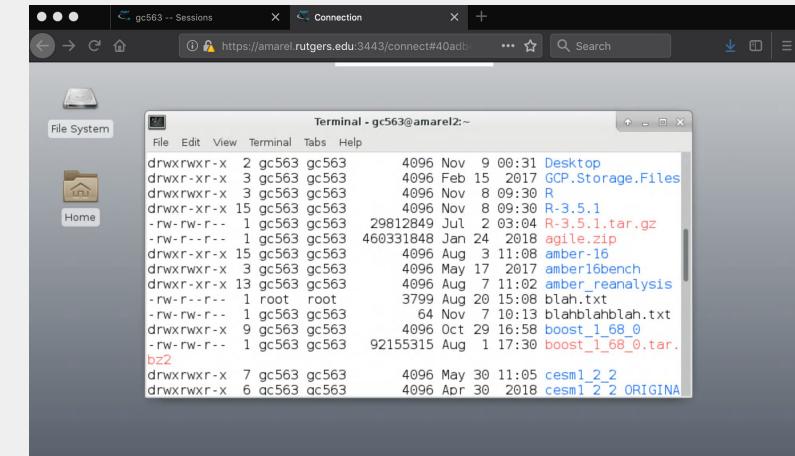
Log-in to Amarel

Choose ‘Launch Session’ and select an XFCE Desktop, then click the Launch button. You’ll be launching a GUI on the Perceval login node using FastX.

On the desktop of your FastX session, right-click and choose ‘Open Terminal Here’

In that terminal window, launch an interactive session

```
srun --time=1:00:00 --pty bash
```



Use ALT-TAB to find minimized windows



Software Support

- System wide Installations (/opt/sw/...)
 - Commercial and open source packages
- Community Installations (/project/community/...)
 - Open source packages
- Personal installations (/home, /project/<group>)
 - Commercial and open source packages

Type of Installations

- Compile from source
- Binaries
- Singularity images



Software Installation

Most programs that you download and install in a Linux environment can be installed using a procedure similar to these steps:

```
wget https://www.python.org/ftp/python/3.6.8/Python-3.6.8.tgz  
tar -zxf Python-3.6.8.tgz  
cd Python-3.6.8  
.configure --prefix=/home/[NetID]/python/3.6.8  
make -j 4  
make install
```



The actual steps may vary a little, but the overall concept should be the same.



Software Installation

Configure your environment for using a new package:

These are just examples, you may not need all of these environment variables. Also, each long line here is a separate “export...” command.

```
export PATH=/home/[NetID]/python/3.6.8/bin:$PATH
export LD_LIBRARY_PATH=/home/[NetID]/python/3.6.8/lib:$LD_LIBRARY_PATH
export LIBRARY_PATH=/home/[NetID]/python/3.6.8/lib:$LIBRARY_PATH
export C_INCLUDE_PATH=/home/[NetID]/python/3.6.8/include:$C_INCLUDE_PATH
export CPLUS_INCLUDE_PATH=/home/[NetID]/python/3.6.8/include:$CPLUS_INCLUDE_PATH
export MANPATH=/home/[NetID]/python/3.6.8/share/man:$MANPATH
```

Add these lines to the end of your `~/.bashrc` file to make the changes persist for all shells & subshells (interactive jobs)

Note: omit the `:-----` part if a variable isn't already set



Singularity

I can't install my software on Amarel (CentOS 7) because

1. it requires newer core system libs (e.g., GLIBCXX_3.4.20+),
2. my software was designed for a different OS,
3. I need root/admin privileges and apt-get or yum to install my software and a ton of dependencies.



Singularity



DOCKER, or SINGULARITY on your laptop. Or install singularity inside a virtual box.

1. Use a Linux workstation or VM where you have root/admin privileges to create a Singularity image,
2. Install your software in that OS image,
3. Run your customized image as a container in a job run on Amarel



Best Practices

Bala Desinghu, PhD
OARC, Rutgers

- Job Considerations
- Guidelines on a shared resource (login node)
- Documentations and training

Help Jobs Start Sooner

- Keep your jobs small, use minimum cores & memory
- Keep your jobs short (shorter wall time / time limit)
- Be flexible: highly specific hardware requirements may take longer to allocate
- Think about memory request boundaries (ask for about 8 GB less than the max)





Running Test Jobs

- Start with a small example to verify that your job will run successfully
- Test parallelization on 1 node (multiple cores) first
- How much memory does my job need? Run some test jobs and look at the detailed job output using

```
sacct --format=MaxRSS,MaxDiskRead,MaxDiskWrite,Elapsed --units=G -j [JobID] to  
see how much memory, I/O, and time your job used
```

- *My job just dies without producing output!* Maybe you didn't allocate enough memory for it to run properly



Sharing Resources

- The log-in / head / master node (amarel1, amarel2, etc.) is a shared system
- Avoid compute-intensive or memory-intensive operations there
- Do not run applications (R, Python, Perl, MATLAB, etc.) on the log-in node... that's what compute nodes are for
- On compute nodes, remember that you may not be the only user on a given node





Amarel User Guide

Bala Desinghu, PhD
OARC, Rutgers

<https://rutgers-oarc.github.io/training>

The screenshot shows a web browser displaying the "Cluster Guides - OARC-Rutgers" page at <https://rutgers-oarc.github.io/training>. The page has a dark-themed header with the OARC-Rutgers logo and a search bar. The main content area shows the "General Information" section of the "Cluster Guides" documentation. The sidebar on the left lists various sections such as Home, Cluster Guides (General Information, Cluster Resources, Basic operations, Listing available resources, Loading software modules, Running slurm jobs, Monitoring the status of jobs, Killing/cancelling/ terminating jobs, Installing your own software), Singularity (Troubleshooting, Acknowledging Amarel), Examples, Community, HOWTO, Jupyter, and GitHub. The footer includes navigation links for "Previous" and "Next".

General Information

There are several clusters administered by OARC. Each cluster is using the resource scheduler [Slurm](#) to allocate users the requested resources.

- Amarel - a general access cluster. [Request access or become an owner](#)
- Perceval - similar to Amarel, but paid for, and to be used by exclusively by, NIH grants and grantees
- NM3 - [Newark Massive Memory Machine](#)
- Sirius - a single big machine with no resource scheduler, acts as a big desktop with multiple concurrent users

Cluster Resources

Amarel

Amarel is a CentOS 7 Linux compute cluster that is actively growing through the combination of separate computing clusters into a single, shared resource.

Amarel includes the following hardware (this list may already be outdated since the cluster is actively growing):

52 CPU-only nodes, each with 28 Xeon e5-2680v4 (Broadwell) cores + 128 GB RAM
20 CPU-only nodes, each with 28 Xeon e5-2680v4 (Broadwell) cores + 256 GB RAM
4 28-core e5-2680v4 nodes each with 2 x Nvidia Pascal P100 GPUs onboard
2 high-memory nodes, each with 56 e7-4830v4 (Broadwell) cores + 1.5 TB RAM

Amarel Ownership

Note: ownership of Amarel resources is certainly not required because all Rutgers community members have **FREE** access

However, benefits of buying-in:



- Dedicated, immediate access to purchased resources via private job queue (partition)
- No limit on the # of jobs running
- Free 1 TB of project storage + access to additional storage purchases
- No indirect costs, this is an equipment purchase / 4-year ownership plan
- <https://oarc.rutgers.edu/services/condo-model>



Learning to Use Applications

Bala Desinghu, PhD
OARC, Rutgers

<https://libguides.rutgers.edu/graduatespecialist/workshops>

The screenshot shows the Rutgers University Libraries website at <https://libguides.rutgers.edu/graduatespecialist/workshops>. The page title is "Graduate Specialist Program (New Brunswick Libraries): Workshops". The main content area lists several workshop categories:

- Python workshops
- Statistical software (SAS and R)
- NVivo workshops
- Digital humanities workshops

Below these, there are sections for Python Workshops, NVivo Workshops, and Digital Humanities Workshops, each with specific details and links.

Python Workshops
Workshops are offered in either Alexander Library or LSM (with identical content). Participants in LSM-based workshops must bring their own laptops. At Alexander, you can either bring your own laptop, or use the desktops in the lab.

Python Basics and Data Exploration

- Friday, February 1 – 12:30-2:30 pm, Alexander Library Room 413 (Instructor, Hang Miao)
- Wednesday, February 13 – 12:00-2:00 pm, LSM Conference Room (Instructor, Aditya Vas)

This workshop will be an accelerated introduction to fundamental concepts such as variable assignment, data types, basic calculations, working with strings and lists, control structures (e.g. for-loops), functions. We will also start working with pandas, a popular data science library in Python, to explore a dataset on foodborne outbreaks reported to the CDC.

Data Manipulation and Analysis with Python

- Friday, February 8 – 12:30-2:30 pm, Alexander Library Room 413 (Instructor, Hang Miao)
- Monday, February 18 – 3:15-5:15 pm, LSM Conference Room (Instructor, Aditya Vas)

NVivo Workshops

Introduction to NVivo

- Tuesday, April 3 – 2:00-3:30 pm, Alexander Library Room 413 (Instructor, Cathy Wang)

This workshop is intended as a basic introduction to using NVivo, a software that supports qualitative and mixed methods research. The workshop focuses on introducing key mechanisms of the software that may be applied as required by different analytical approaches.

Advanced NVivo: Data Visualizations Using NVivo 12

- Tuesday, April 17 – 2:00-3:30 pm, Alexander Library Room 413 (Instructor, Cathy Wang)

This workshop focuses on the introduction of a suite of visualizations that help you gain deeper insights from your data by exploring and unearthing patterns, trends and connections.

Digital Humanities Workshops



Documentation and Use

Bala Desinghu, PhD
OARC, Rutgers

Amarel User's Guide

<https://rutgers-oarc.github.io/training>

Final Notes:

- Do not run compute-intensive or memory-intensive applications on the log-in nodes (amarel1, amarel2, etc.)
- For best performance, work in /scratch, then move files when finished
- Need help?
 - E-mail help@oarc.rutgers.edu
 - Please remember to do the following in your emails:
 - Specify the cluster in question (Amarel, Perceval, Didact, etc.)
 - Specify your Job ID (if you have one)
 - Paste complete error messages in the email (if you have one)
 - Use an official Rutgers e-mail only (others will be disregarded)



Thank You