MAPRONANO-ACE –IDI ARTIFICIAL INTELLIGENCE, MACHINE LEARNING & BIOINFORMATICS SHORT COURSE

Bioinformatics, AI Workflow—Computational infrastructure, HPC data acquisition
-- A practical demonstration

Rodgers Kimera

Snr. Systems Engineer at

African Centers of Excellence in Bioinformatics $ Data Intensive Science

10/5/2021

ACE AFRICAN CENTERS OF EXCELLENCE IN BIOINFORMATICS

Overview

- AI Vs. HPC overview
- Bioinformatics overview
- Relating AI, Bioinformatics to HPC
- Introduction to HPC
- SLURM Job directives
- HPC Software stack
- Anaconda Package Manager

Objectives

- Understand why HPC is needed in the fields of AI and Bioinformatics
- Understand how HPC works
- Get started on using HPC
- Understand job scheduling
- Write basic SLURM job scripts
- Understand the HPC software stack
- Creating customized user environments

AI Vs. HPC

- AI - leverages computers and machines to mimic the problem-solving and decision-making capabilities of the human mind.
- It is the science and engineering of making intelligent machines, especially intelligent computer programs.
- HPC refers to any computer system that has increased capabilities far beyond what ordinary computers can manage.
- HPC is hardware, and AI is software.

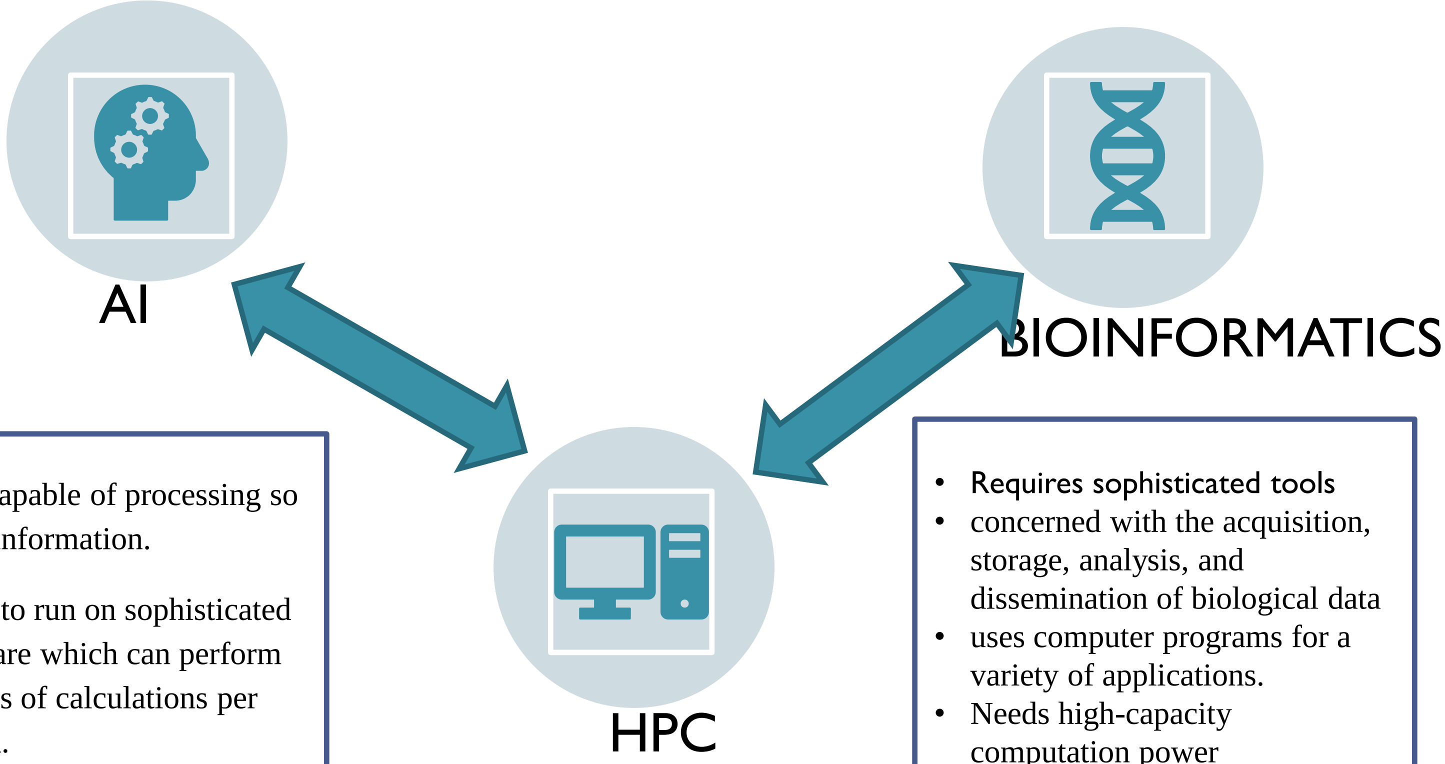
AI Vs. HPC cont…..

- AI is capable of processing so much information hence it needs to run on sophisticated hardware which can perform trillions of calculations per second, or more.
- This is where HPC and AI intersect since HPC uses dense computer clusters – in sync with one another to perform the necessary calculations at blistering speeds and run the most advanced AI.

Bioinformatics

- Concerned with the acquisition, storage, analysis, and dissemination of biological data, most often DNA and amino acid sequences.
- Uses computer programs for a variety of applications such as GALAXY, SPAdes, BioJava etc. in determining gene and protein functions, establishing evolutionary r/ships among others

Relating AI, Bioinformatics to HPC



- AI is capable of processing so much information.
- Needs to run on sophisticated hardware which can perform trillions of calculations per second.

- Requires sophisticated tools
- concerned with the acquisition, storage, analysis, and dissemination of biological data
- uses computer programs for a variety of applications.
- Needs high-capacity computation power

Introduction to HPC

- HPC refers to any computer system that has increased capabilities far beyond what ordinary computers can manage.
- High-Performance Computers contain all the usual components of a computer – CPU, RAM, Storage, Cooling, etc.
- HPC was developed to meet the increasing demands for processing speed. As it brings together different technologies like computer architecture, algorithms, system software, programs, and electronics under one canopy
- HPC can solve complex problems quickly and efficiently.

Introduction to HPC cont.…

- **What is HPC?**
  - Working on HPC
  - HPC access and usage
- **Getting started with HPC**
  - Linux basic command
  - Bash scripting
- **Running jobs**
  - Use the batch system
  - Execute parallel programs

HPC contains thousands of compute nodes that work together to complete multiple tasks.

ACE AFRICAN CENTERS OF EXCELLENCE IN BIOINFORMATICS

Introduction to HPC cont.…

User Experience

- Multiuser system
- Unix OS
- Optimized software

Compute power

- Many CPUs system
- Specialized Hardware
- Low-latency/High bandwidth Connections

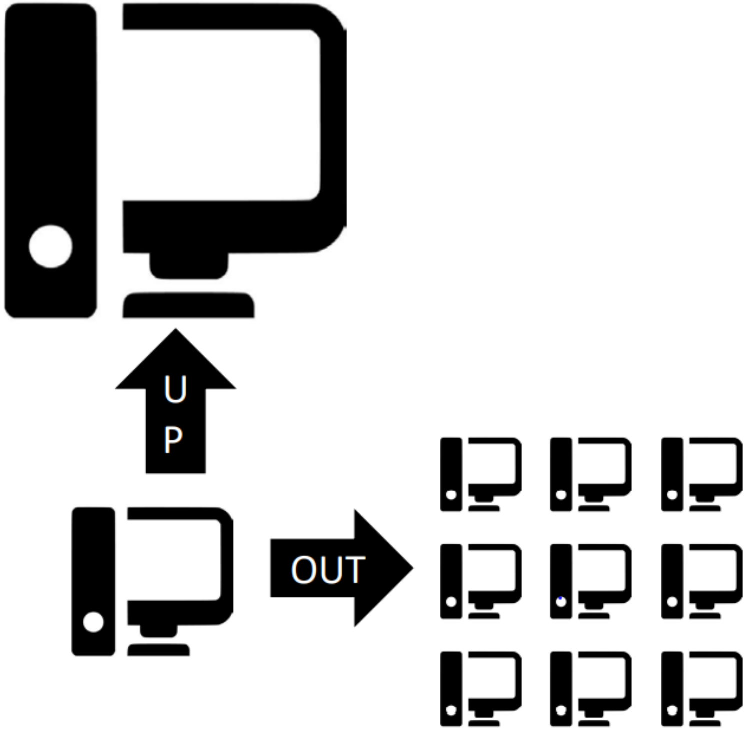
Storage

- Efficient I/O
- Large Memories

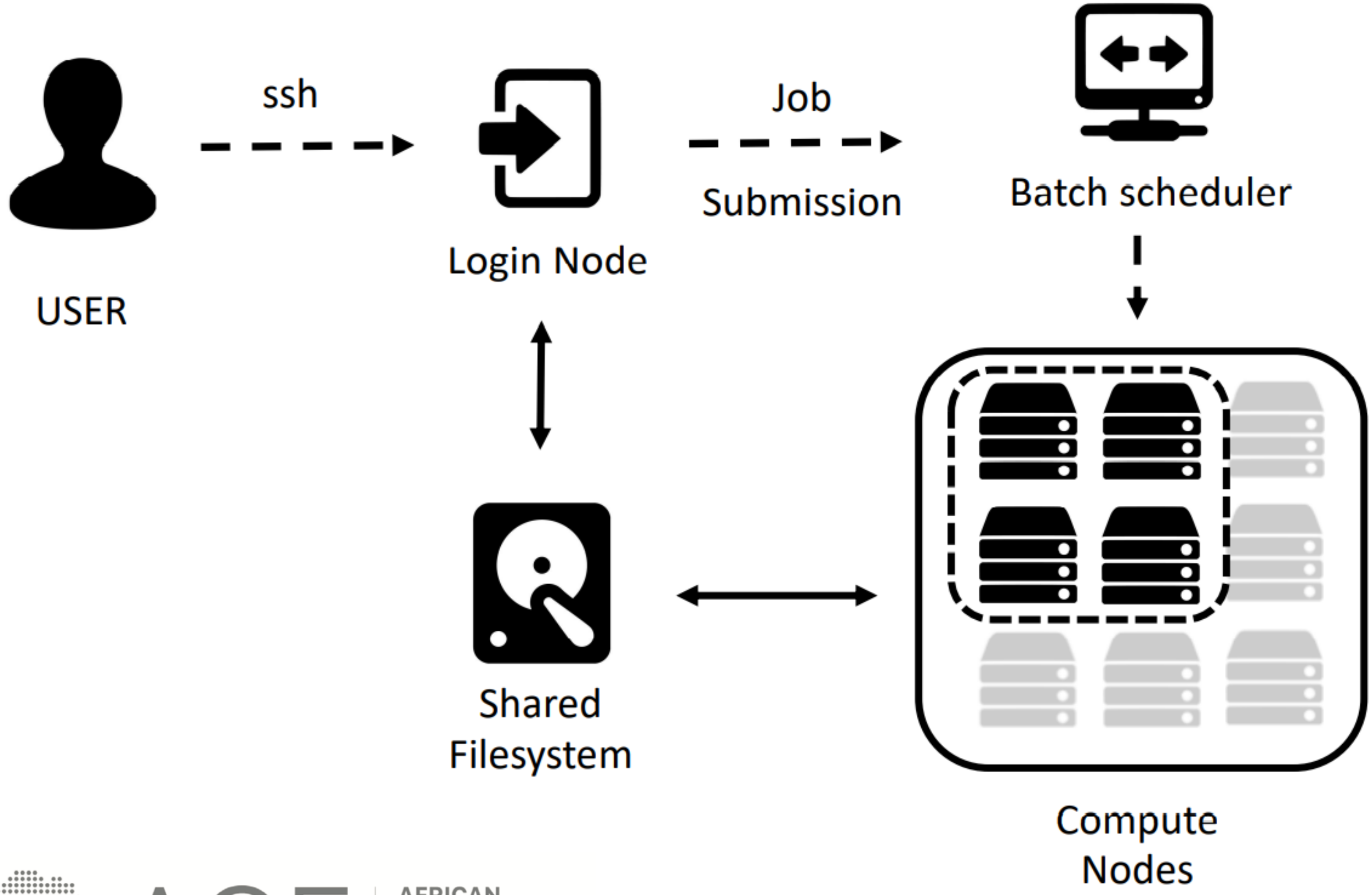
Introduction to HPC cont..

Why, or more, when you need HPC?

- **Scale up**
  - Faster CPUs
  - Large memories
  - Specialized Hardware/Software
- **Scale out**
  - Large parallel application
  - Many small- to medium-size jobs



Working with HPC



AFRICAN CENTERS OF EXCELLENCE IN BIOINFORMATICS

Working with HPC

Login node(s)

- Editing and transferring files
- Compile programs
- Prepare simulations

Compute nodes

- Multicore nodes
- Large memories
- High-speed Interconnections

Batch scheduler

- Resource allocation
- Job queueing
- Accounting

File system

- Parallel FS
- Efficient I/O
- Node local disks

Working with HPC

- Login and transfer files to the remote machine
  - ssh, scp/ftp
  - Command line, GUI
- Prepare your job(s)
  - Input preparation
  - Job submission script
  - Software preparation
- Submit your job and retrieve output
  - Submit job to the batch system
  - Monitor job
  - Retrieve outputs / Remote visualization

Login to an HPC system

Install UNIX tools on your local machine

- **Windows**
  - Putty
  - MobaXterm (http://mobaxterm.mobatek.net)
- **Mac OSX**
  - Terminal (pre-installed)
  - XQuartz (http://www.xquartz.org)
- **Linux**
  - You are already well equipped

Running Jobs on HPC

Batch System

- HPC use batch systems to distribute computational tasks over the available nodes.
- Instead of executing commands interactively, you prepare a job script
  - Script containing the commands to execute
  - Resource characteristics (specific)
- The batch system is responsible for allocating cores, processors or nodes to a job.

Advs of Batch Systems

- It allows to run MANY jobs at the same time.
- Multi-users, queue system
- System load balance

slurm
workload manager

PBS

SLURM Job directives

- A job script must contain directives to inform the batch system about the characteristics of the job.This directive appear as comments (#SBATCH) in the job script and must conform with the sbatch syntax.

#SBATCH --nodes=<num>	request for <num> compute node
#SBATCH --ntasks=<num>	the number of processes to start
#SBATCH --time=DD-HH:MM:SS	total wall clock time of the job
#SBATCH --qos=<queue>	requesting a specific queue
#SBATCH --task=<queue>	requesting a specific queue
#SBATCH --output=<file>	name of the file where std out is printed
#SBATCH --tasks-per-node=<num>	
#SBATCH --cpus-per-task=<num>	

Always type: sbatch --help to find support on what most directives mean and when they should be used.

Slurm Workload Manager - sbatch (schedmd.com)

Batch jobs

batch/slurm1.sh

```
#!/bin/bash
#SBATCH --job-name="test_script"
#SBATCH --ntasks=1
#SBATCH --time=00:02:00
#SBATCH --chdir=.
#SBATCH --output=test_%j.out
#SBATCH --error=test_%j.err
echo "Who am I?"
whoami
echo "Where ?"
srun hostname
sleep 600
```

batch/slurm2.sh

```
#!/bin/bash
#SBATCH --job-name="test_multinode"
#SBATCH --nodes=2
#SBATCH --tasks-per-node=3
#SBATCH --time=00:02:00
#SBATCH --chdir=.
#SBATCH --output=multinode_%j.out
#SBATCH --error=multinode_%j.err
echo "Who am I?"
whoami
echo "Where ?"
srun hostname
sleep 600
```

Submitting job with SLURM

- The method for submitting jobs is to use the SLURM **sbatch** directives directly.
- submits a "job script" to the queue system:

```
sbatch <job script>
```

- **Some SLURM Common Commands**

sbatch [job script]	Submit batch job
sinfo	Queue status
scontrol show job [job id]	Job Status
scontrol show node [node id]	Node Status
scancel [job id]	Delete job/ remove the job from the queue system
squeue	shows all the submitted jobs and their status

Software stack

- Some software packages require certain settings in your user environment, like paths and environment variables.
- Environment Modules;
  - mechanism by which much of the software is made available to the users of the clusters.
  - Provide lots of useful software packages in many different versions
- Most HPC clusters use Package managers to better manage packages for users such as SPACK and Anaconda

Software stack cont.….

- As HPC systems are being used by many users with different requirements,
- Users usually have multiple versions of frequently used software packages installed.
- As it is not easy to install and use many versions of a package at the same time, this system uses environment modules that allow users to configure the software environment with the specific version required.

Software stack cont.….

- **Useful module commands**

module avail	list modules
module avail R	list all installed versions of R
module load R	load the default R version
module load R/3.6.3	load a specific version of R
module unload R	unload R
module list	list currently loaded modules
module purge	unload all modules
module spider	list all modules, even those not available to load
module save	save the currently loaded collection of modules
module reset	load the system default collection of modules
module restore	load your personal default collection of modules

module --help	provide you with all options that need that you can use

Anaconda Package Manager

- One of the commonly used package managers
- Use environment module to load anaconda

```
module load anaconda3-2019.10-gcc-9.3.0-7gh72na
```

- First initialize anaconda to activate the base environment

```
eval "$(conda shell.bash hook)“
```

- See list of all conda packages installed

```
conda list
```

- Then you can proceed to use any conda package per your preference

Anaconda environments in you HPC account

- Requires no root privileges, and will let you customize your Python and other packages to your needs

  ```
  conda create --name environment-name
  ```

- Create an Environment Using a Specific Version of Python

  ```
  conda create -n <environment-name> python=2.7.0
  ```

- Remove an Environment

  ```
  conda env remove --name <environment-name>
  ```
  or
  ```
  conda env remove -n <environment-name>
  ```

- Find list of your environments

  ```
  conda env list
  ```

Practice

- Log into the HPC
- Load a system-wide anaconda installation
- conda create --name py37 python=3.7
- `conda activate py37`
- `conda env list`

Thank you