Introduction

- Distributed and Cluster Computing (CPSC 3620)
- Offered twice per academic year
- Average class size: 40-45 students
- Required junior-level class (typically taken at senior year)
- Contents:
  - Infrastructure/System-oriented
  - Performance/Efficiency
  - High Performance Computing
    - MPI
  - Big Data Computing
    - Hadoop MapReduce
    - Apache Spark
    - HPCCSystems

Computing Resources

- Palmetto Supercomputer
  - 2000+ nodes, open to all faculty/students
  - No administrative access
  - Cannot share nodes among students to support group assignment
  - Preemption from node owners
- CloudLab
  - Limited resources for large-scale study
  - Administrative access
  - Ease of collaboration
  - No preemption

Tutorial

- Set up environments for distributed computing on CloudLab
- MPI
  - Two-node cluster
  - OpenMPI
- Hadoop
  - Three-node cluster
  - Hortonwork Distribution

Learning outcomes through Palmetto

- Basic understanding of parallel application development
- Impacts of scaling and efficiency on larger systems
Both nodes should have the same configuration, no network connection is needed (due to public IP).

### On each node:

- `sudo apt-get update`
- `sudo apt-get install libibnetdisc-dev`
- `sudo nano /etc/environment`
- `PATH=’/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/home/mpiuser/.openmpi/bin’`
- `LD_LIBRARY_PATH=’/lib:/usr/lib:/usr/local/lib:/home/mpiuser/.openmpi/lib’`
- `sudo adduser mpiuser`

`mpicc gethostname.c –o gethostname`

`scp gethostname mpiuser@<the other node>:/home/mpiuser`

`scd nodelist mpiuser@<the other node>:/home/mpiuser`

`mpirun –np 2 –machinefile nodelist ./gethostname`

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/utsname.h>
#include <mpi.h>

int main(int argc, char *argv[]){

  int rank, size;
  MPI_Status status;

  MPI_Init(&argc,&argv);
  MPI_Comm_size(MPI_COMM_WORLD,&size);
  MPI_Comm_rank(MPI_COMM_WORLD,&rank);

  struct utsname uts;
  uname(&uts);
  printf("%d at %s
",rank,uts.nodename);

  MPI_Finalize();

  return 0;
}
```
Assignment Ideas

- Develop a work queue using various allocation strategies:
  - Normal
  - Cyclic
  - Dynamic

- Setup MPI cluster with nodes on separate sites, reduce network connection, and evaluate performance on different allocation strategies

Hadoop on CloudLab

- Enterprise Hadoop
- Hortonworks

On each node

- SSH onto the node from Palmetto
- Change to root: sudo su -
- Execute the following commands:
  - chkconfig --list ntpd
  - chkconfig ntdp on
  - service ntpd start
  - chkconfig iptables off
  - /etc/init.d/iptables stop
  - setenforce 0

On namenode

- Set up ambari server:
  - ambari-server setup

- Select default for all questions

- Select 1 for JDK version

- When all done, start ambari server
  - ambari-server start

On each node

- Setup Ambari download server
  - wget -nv http://public-repo-1.hortonworks.com/ambari/centos6/2.x/updates/2.1.2/ambari.repo -O /etc/yum.repos.d/ambari.repo

- On namenode
  - yum -y install ambari-server
  - yum -y install ambari-agent

- On datanode
  - yum -y install ambari-agent
On each node

- Using vim to edit `/etc/ambari-agent/conf/ambari-agent.ini`
- Change: `hostname=<hostname of namenode as shown in list view of CloudLab>`
- Start Ambari Agent
  `ambari-agent start`
Assuming you had ambari agents up and running ...

Ambari Server (admin/admin)

- HDFS
- YARN+MapReduce2
- Tez
- ZooKeeper
- Ambari Metrics

Ambari Server (admin/admin)

Ambari Server (admin/admin)

Ambari Server (admin/admin)

Edit configuration as you see fit

Deploy ...
Warning due to lack of space and failed checks (ignore)

HDFS

YARN

Tutorial

sudo su hdfs
hdfs dfs -mkdir /user/<username>
hdfs dfs -chown <username>:<username> /user/<username>
exit to <username>
hdfs dfs -ls /user/
git clone https://github.com/clemsoncoe/Introduction-to-Hadoop-data.git
cd Introduction-to-Hadoop-data
hdfs dfs -put gutenberg-shakespeare.txt /user/<username>/
yarn jar /usr/hdp/current/hadoop-mapreduce-client/hadoop-mapreduce-examples-2.7.1.2.3.6.0-3796.jar wordcount
gutenberg-Shakespeare.txt output/
hdfs dfs -ls output
hdfs dfs -cat output/part-r-00000

Assignment Ideas

- Deploy a Hadoop cluster and upload a large data set (Airline on-time performance data: http://stat-computing.org/dataexpo/2009/the-data.html)
- Examine and investigate performance of Hadoop MapReduce as data nodes are killed/added to the cluster
- Examine performance of Hadoop MapReduce as data nodes are located on different sites