

# Distributed Cloud Computing with StarCluster (DC2S)

CHRISTIAN ANGELES

JOHN ZAVALA



# Project Description

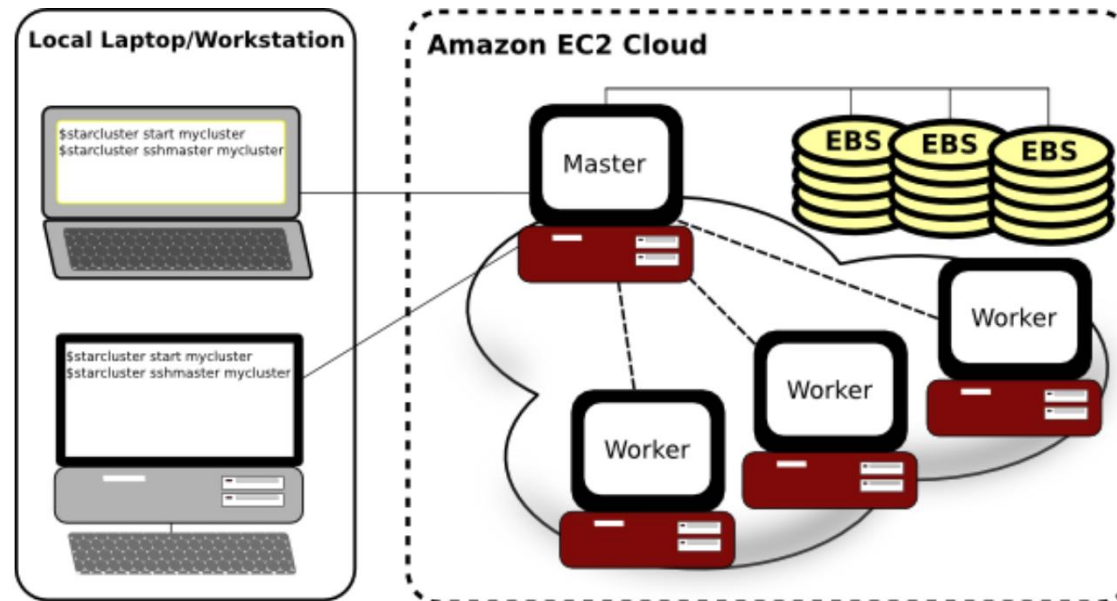
- ▶ Our goal is to simulate computational offloading of a compute-intensive task on an android application to a cloud server.
  - ▶ Server will complete the task via distributed computation.
  - ▶ Data will be an array of at least 1,000,000 elements conceptualizing a large data set--data that is too large to store on an android device.
  - ▶ Program on the cloud server will be performing an exhaustive search and comparison to simulate the computational offloading.
  - ▶ In order to create a compute-intensive environment, the algorithm is going to search and compare the 1,000,000-element-sized array with itself. Forcing the algorithm to perform with a time complexity of  $O(n^2)$ . A slower time run will allow us to see a difference in computational performance.

# Project Environment

- ▶ Amazon Web Services(AWS) EC2
  - ▶ Starcluster AMIs
- ▶ Android OS
- ▶ Linux OS
- ▶ StarCluster
- ▶ Message passing interface(MPI)
  - ▶ mpi4py for Python

# StarCluster

- StarCluster- Is a tool kit that manages computer clusters hosted on Amazon's EC2 Cloud. It is designed to automate and simplify the process of building, configuring, and managing clusters of virtual machines.

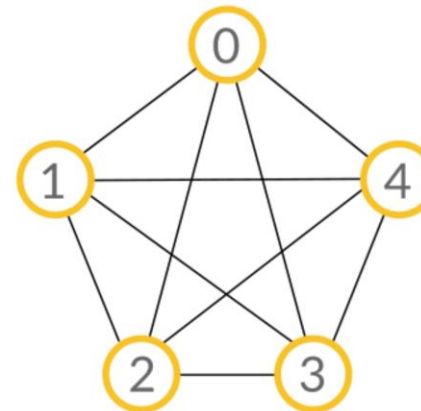


# StarCluster

- ▶ Security Groups- The starcluster configures a security group for our cluster allowing us to control network access to the cluster.
- ▶ Password-less SSH: The star cluster configures the cluster in order for the SSH to be used from any node within the cluster without having to login using a password.
- ▶ Network File System (NFS)- able to attach Elastic Block Storage (EBS) volumes on the cluster in order to obtain persistent storage.
- ▶ StarCluster can dynamically resize clusters in order to make it scalable.
- ▶ Combination of OpenMPI and Sun Grid Engine (SGE)

# Message Passing Interface(MPI)

- ▶ MPI- Standardized message passing library interface that can be used in different languages (ex. C, C++, python).
- ▶ MPI is good for high performance computing and parallelism because it handles the passing of messages between different processes.
- ▶ Distributed computing- Takes a single task and distributes it among computers in order to complete a task.
- ▶ Point to point communications- MPI uses point to point communications to communicate from one process to another.
  - ▶ Blocking/Non-blocking communication
  - ▶ Race conditions



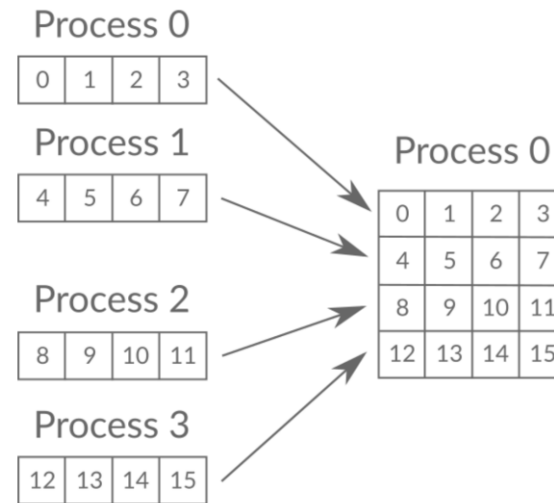
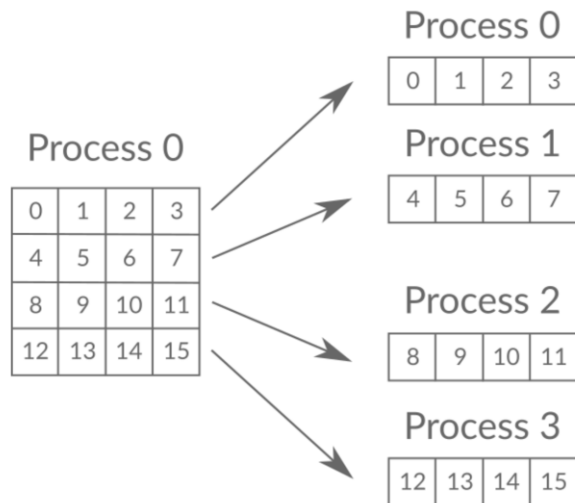
# Message Passing Interface(MPI)

- ▶ Collective communication- There are two types: Broadcasting and Scattering/Gathering.
  - ▶ Broadcasting: One process broadcasts the same information to every process.



# Message Passing Interface(MPI)

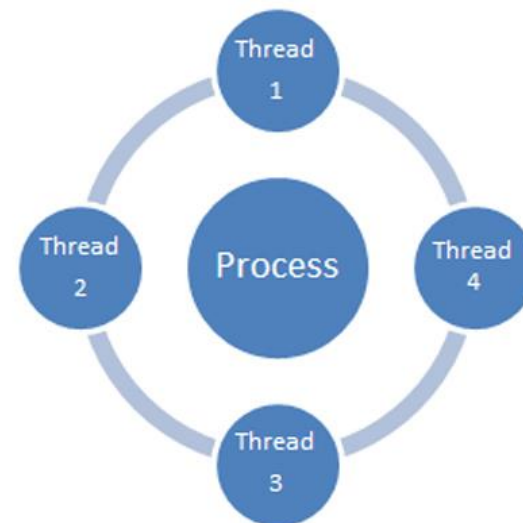
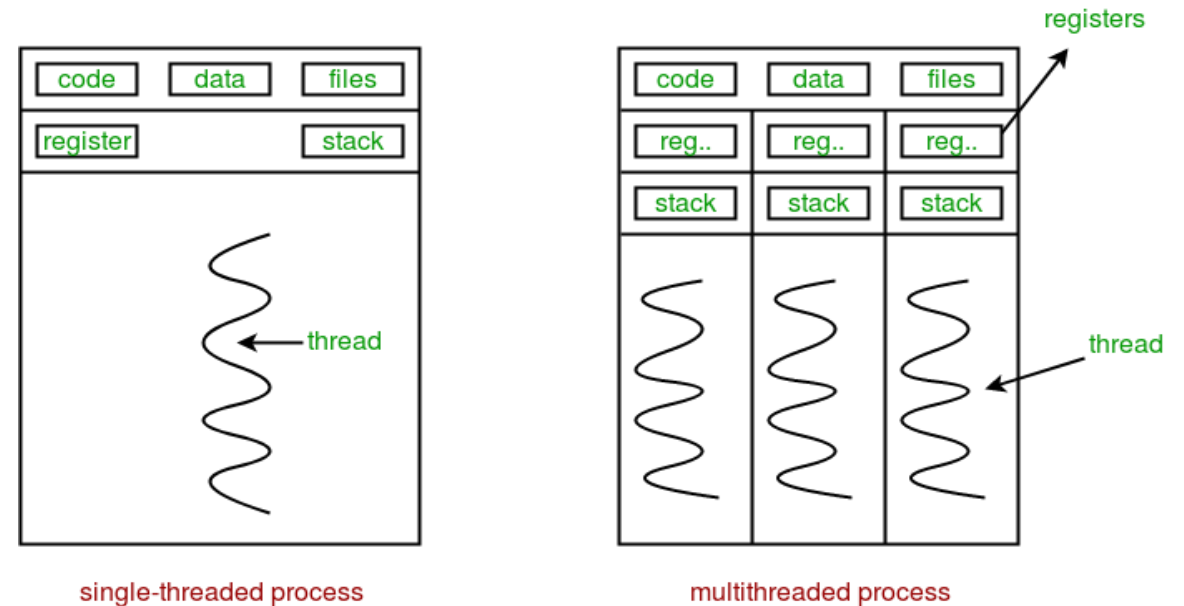
- ▶ Collective communication- There are two types: Broadcasting and Scattering/Gathering.
  - ▶ Scattering/ Gathering: Are used in order to distribute and gather data.





# Multithreading

- Multithreading is an execution model that allows multiple threads to be created within a process such that the threads execute independently but share the same process resources.
- Every thread contains a unique register set and local variables. (Local variables are stored in the stack).
- All the threads within the process share the same global variables and code. (The global variables are stored in the heap).
- Main reason we are using multithreading rather than multiprocessing is because of less overhead. It's a more efficient way of performing the task.



# Cloud Server

- ▶ Written in python for easy implementation
  - ▶ MPI support with mpi4py
  - ▶ Socket programming
  - ▶ Multiprocessing/Multithreading libraries
- ▶ Main process
  - ▶ Listens for client connection
    - ▶ Creates a thread for each client connection
  - ▶ Launches separate process for distributed computing
- ▶ MPI process
  - ▶ Processes created for each node
  - ▶ Divides array into segments
    - ▶ Creates multiple threads for each segment

# Android Application

- ▶ Application development in Android Studio.
- ▶ Designed a simple user interface
  - ▶ Single button to start
- ▶ Contains implementation of socket programming in order to establish a connection to the cloud
  - ▶ Creates a separate thread for connection to the cloud server
    - ▶ AsyncTask



# Demonstration

# Lessons learned

- ▶ AWS Elastic Cloud Computing (EC2)
  - ▶ Creating instances
  - ▶ Configuring network security group
    - ▶ Allows android application to connect
- ▶ StarCluster
  - ▶ Parallel Computing
    - ▶ MPI(Message Passing Interface)
  - ▶ SGE( Sun Grid Engine)
- ▶ Multiprocessing/Multithreading
  - ▶ Android
  - ▶ Python



# References

- ▶ StarCluster
  - ▶ <http://star.mit.edu/cluster>
- ▶ Distributed computing with MPI
  - ▶ <https://www.codingame.com/playgrounds/349>
- ▶ Open MPI
  - ▶ <https://www.open-mpi.org>
- ▶ mpi4py
  - ▶ <https://mpi4py.readthedocs.io>

**Thank you!**

**Questions?**