Hadoop Cluster Administration

Bu eğitim sunumları İstanbul Kalkınma Ajansı'nın 2016 yılı Yenilikçi ve Yaratıcı İstanbul Mali Destek Programı kapsamında yürütülmekte olan TR10/16/YNY/0036 no'lu İstanbul Big Data Eğitim ve Araştırma Merkezi Projesi dahilinde gerçekleştirilmiştir. İçerik ile ilgili tek sorumluluk Bahçeşehir Üniversitesi'ne ait olup İSTKA veya Kalkınma Bakanlığı'nın görüşlerini yansıtmamaktadır.

Outline

- Choosing hardware / platform
- Getting a single node up and running
- Managing a running cluster
  - Caches, Buffers, and Backups
  - Scheduling Policies
- Adding nodes

Caveats and Context

- Why do you need to know this?
- Even if you never have to worry about it, it helps you understand the underlying process

What Machines to Buy

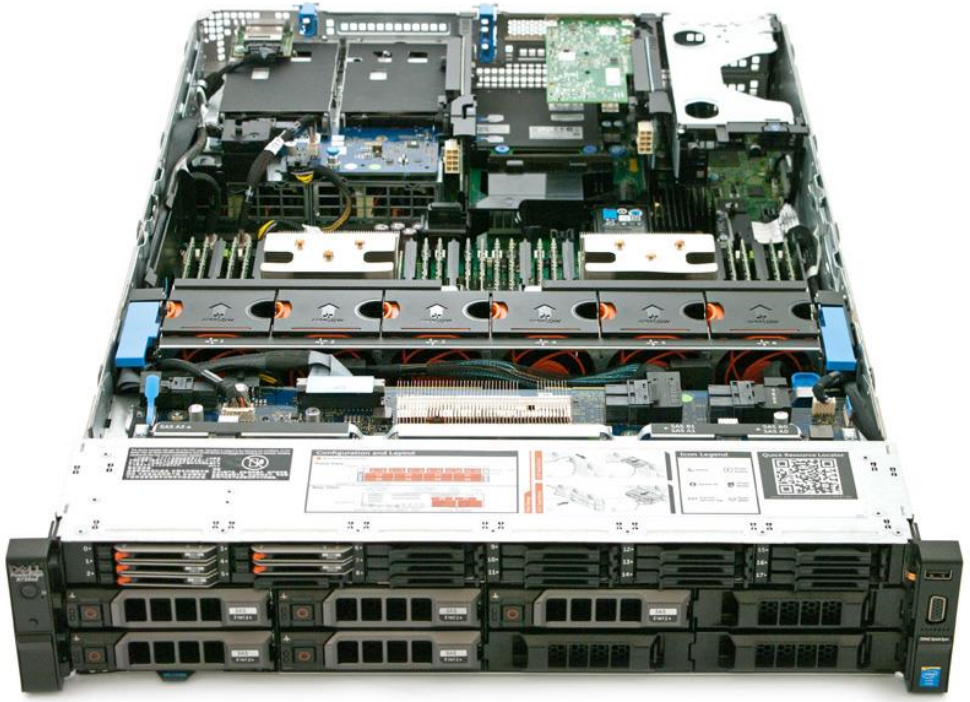
- Depends on your budget
- Get good consumer-grade machines
- Get components that you can replace for the next 4-8 years If you want homogenous hardware, buy expensive now, and have costs descend as you scale out over time

- BAU Big Data cluster:

- **Name nodes**: PowerEdge R730 with 12 core 2 - 2.9 Ghz CPU, 128 GB DDR3 RAM, Gb Ethernet NIC, 16TB SFF Nearline/MDL SAS 7200 RPM hard disk)

- **Data nodes**: PowerEdge R730 with 12 core 2 - 2.6 Ghz CPU, 96GB DDR3 RAM, Gb Ethernet NIC, 24TB SFF Nearline/MDL SAS 7200 RPM hard disk)

What Machines to Buy



Do you even want to buy machines?

- Amazon Elastic Compute Cloud (Amazon EC2) Part of Amazon Web Services (AWS)

- Rent machines for $0.10 / machine hour to $2 / machine hour (depending on CPU / memory)

- Pros
  - Don't pay for electricity
  - Seamless"upgrades"
- Cost
  - Not as cost-effective as running your own cluster
    - 24/7 Who does?
  - Less control

Creating and Using a Hadoop Cluster on EC2

- Install Hadoop on a local machine
- Edit *hadoop/src/contrib/ec2/bin/hadoop-ec2-env.sh*D
  - Add AWS account, key
  - Size of machines
  - Architecture
- Hadoop installation provides a script to create cluster
  - bin/hadoop-ec2 launch-cluster test-cluster 2
  - Starts running a TaskTracker, command returns IP
- Can then either log in
- Or run remotely (just like we're doing)
  - Caution, IO is metered (cent per minute)

Do you even want to bother with virtual machines?

- Amazon offers "Elastic Map Reduce"

Elastic MapReduce

- Uses S3 for Input and Output
- Very little configuration (web-based)
- Can use most of the techniques discussed in class
  - Streaming
  - Custom jar files
  - Chaining jobs
- Cannot use
  - Local data
  - Hadoop pipes
- API or CLI for automation of creating environments / jobs

Complications of Using AWS

- There are outages (beyond your control)
  - E.g. today (April 21, 2011), Reddit, Foursquare, and Quora were down
- While there are SLAs, it's only a refund of what you've paid
- What's the answer?
  - As before, it's almost always redundancy
- Amazon offers four zones
  - US-East (Norcal), US-West (Virginia), Europe (Ireland), Asia (Singapore)
  - Hardware relatively independent across zones
  - Multiple instances increase probability continuity, cost
  - What about software?

How to put together a new cluster

- Installing software
- Letting computers talk to each other
- Configuring the network
- Setting up storage
- Changing options

Installing Software

- Do it yourself
  - Java
  - Hadoop
  - Anything else you need ...

- Using Cloudera or other commercial distributions
  - Maintains internally consistent packages
  - Play well together
  - Provides
    - Packages
      - Different
        for namenode, datanode, secondarynamenode
    - Whirr (image + setup) for use on EC2
    - Virtual Machine Images

SSH Key Distribution

- NameNode must be able to connect to all slave machines (e.g. to start up processes when the cluster starts)
- SSH works on private and public keys
  - Keep private key
  - Distribute public key to the systems you connect to
- Typically done with a script on NameNode that copies public key to many computers
- Do this with "hadoop" user

Specifying Network Topology

- Default configuration puts nodes on the same rack
- For small clusters, this is fine
- Large clusters have more complicated topology
  - Throughput much larger within a rack
  - Tasks will complete faster if jobs are localized to racks
- Goes beyond racks
  - switch, data unit, building, datacenter

Configuring Topology

- The parameter **topology.script.file.name** should point to a script that takes IP addresses or host names and returns the rack location

Setting up HDFS

- NameNode - Hold metadata for the blocks of data on cluster
- Secondary NameNode - Merges EditList with FsImage
  - Identical memory requirement as NameNode
  - Reconciles edits

- NameNode often is the weakest link
- Good idea to have separate machine, less strain on NameNode

- Default
  - Nodes are identical
  - EditList is reconciled only on initialization

Using a Secondary NameNode

- Adding it to the network
  - Add its entry to the *masters*
  - Update **dfs.http.address** so it knows where to get edits

- What if the NameNode fails?
  - Change the IP address of secondary NameNode to that of old NameNode
    - Cannot just be host, as DNS is cached
  - Remove its entry from masters, add new secondary
  - Start the NameNode on what was the secondary

Getting Ready to Run

- Create a hadoop user that own appropriate directories
  - E.g. temporary processing files
  - DataNode blocks
- Distribute configuration files
- Decide which nodes are going to take on which roles
  - masters - list of secondary name nodes
  - slaves - data nodes
- Run start-dfs.sh on the NameNode (SSH keys)
  - Starts all of the data nodes
  - Starts the SecondaryNameNode

- Run start-YARN.sh
  - Starts Node Manager on all of the slave odes
  - Starts Resource Manager on master node

Cluster's Running ... Now What?

- Addressing common problems
- Improving scheduling
- Monitoring performance
- Adding new nodes

fsck and rebalance

- Like the Linux command, checks health of file system
  - Unlike the Linux command, doesn't fix them
- Reports replications
- Can also list where blocks are located for a file
- What to do when unbalanced?
- Wait and let things sort themselves out
- Run bin/start-balancer.sh
- Restart HDFS

Adding New Nodes

- Explicitly specify hosts in dfs.hosts.
  - hosts located on NameNode
- Is your cluster now good to go?

Removing Nodes

- Could just unplug ...
- Add the node to to *dfs.hosts.exclude* and *mapred.hosts.exclude*
- Jobs will not run
- Blocks will not count toward replication
- Run

  bin/hadoop dfsadmin -refreshNodes

- Will begin to move data off nodes

Ongoing Activities

- Monitor health of cluster (e.g. Ganglia)
- Set up alerts to warn of impending issues
- Regularly benchmark important applications
- Adjust parameters as average use cases emerge
- Create infrastructure for changing and deploying new configurations

Summary

- Details of a real installation
  - Data storage
  - Network connectivity
  - Scheduling
  - Adding and removing nodes
  - and many messy details..