Hadoop paper-exercise draft

Distribution or version selection is based on required features and stability. Hadoop is available in both source and binary formats / distributed as tarballs containing both.

NIST stability requirements = ?. Production grade feature set? If not, <https://www.safaribooksonline.com/library/view/hadoop-operations/9781449327279/ch04.html>

**Operating System**

CentOS. **Puppet** to manage the OS preparation and config across the cluster; still high effort.

**Hardware**.

Discuss balancing the ratio of CPU to memory to disk. NIST intended workloads = ?

Workload Type Bound By Reference

|  |  |  |
| --- | --- | --- |
| Classification | CPU | 2 |
| Bayesian classification | I/O; except for first map tasks | 4 |
| Clustering, text mining, NLP, feature extraction | CPU | 2 |
| k-means clustering | CPU in iteration but I/O bound in clustering | 4 |
| Sorting, simple comparisons (read / write) | I/O | 2 |
| Terasort | CPU in map but I/O bound in reduce | 4 |
| Indexing, grouping, movement and transformation | I/O | 2 |
| Nutch indexing | CPU in map but I/O bound in reduce | 4 |

Table 1: example workload restrictions

Distributed computing employs two classes of hardware; masters and workers. Masters processes are generally RAM hungry but low on disk consumption (except for log storage); are critical, and cost more. OS [raid] does not consume much space even contending with logfiles. Workloads with a lot of ingest will require more drives; LFF capacity.

Four aspects of hardware to consider: processing, memory, networking, and storage. Edge nodes, gateways between outside networks and Hadoop clusters are recommended for running administration tools (and Hive) and preserving NameNode resources for critical purposes. Transfer utilities such as Sqoop should almost always be placed on an edge node.

Baselines for typical implementations vary slightly. See tables 2, 3, 4 and 5 for comparisons of recommendations on processors, memory, networking and storage, respectively.

|  |  |  |
| --- | --- | --- |
| Dual socket server with ivy bridges, clocked 2 to 2.5 ghz. | Source | [1](http://www.dummies.com/how-to/content/edge-nodes-in-hadoop-clusters.html) |
| Light processing: 2 hex core CPUs, clocked 2 to 2.5. | Source | [3](https://blog.cloudera.com/blog/2013/08/how-to-select-the-right-hardware-for-your-new-hadoop-cluster/) |
| Dual quad core 2.6 ghz CPU. | Source | 5 |
| 2 hex core 3ghz CPUs with 16gb cache. | Source | 6 |

Table 2: processor recommendations

|  |  |  |
| --- | --- | --- |
| Light processing: 24 to 64GB | Source | 3 |
| 24GB DDR3 RAM | Source | 5 |
| 24, 48 or 64GB DDR 31600 ECC | Source | 6 |

Table 3: memory recommendations

|  |  |  |
| --- | --- | --- |
| Client apps and admin tools: two pair bonded 1gb Ethernet. | Source | 3 |
| High data transfer: two pair bonded 10gb Ethernet. | Source | 3 |
| Dual 1gb Ethernet NICs | Source | 5 |
| 2 x 1gb Ethernet | Source | 6 |

Table 4: networking recommendations

|  |  |  |
| --- | --- | --- |
| Client apps and admin tools: four 900gb drives, raid controller (Cassandra). | Source | 3 |
| Datanode; TaskTracker: 12 to many TB hard disks in a JBOD (for HBase) | Source | 3 |
| NameNode and JobTracker: 4 to 6TB in JBOD\* | Source | 3 |
| \*(1 for zookeeper, 1 for the OS, 2 for FS image, 1 for journal) | Source |  |
| Light processing: eight 1 or 2TB disk drives | Source | 3 |
| 2 or more SATA II disk drives in a JBOD in addition to the host OS | Source | 5 |
| 12 3TB LFF SATA II 7200 RPM disks with a 6GB per second SAS disk controller. | Source | 6 |

Table 5: storage recommendations

Namenode. Requires RAM and dedicated disk. All metadata must fit into physical memory. Metadata contains the filename, permissions, owner and group data, list of blocks that make up each file, and current known location of each replica of each block.

Small files problem: Each file is made up of one or more blocks and has associated metadata. The more files the namenode needs to track, the more metadata it maintains, and the more memory it requires. As a base rule of thumb, the namenode consumes roughly 1 GB for every 1 million blocks.

Provision / device management: write to raid [master machines], should write a copy to and NFS. Storage configuration dictated by hardware purchased to support master daemons. Secondary namenode hardware normally identical to the namenode. Jobtracker. Also memory hungry, difficult to predict. Raid on jobtracker and slave machines not required; redundancy is orchestrated across all the slaves, jobtracker saves to HDFS.

Worker hardware: responsible for both storage and computation. Balance CPU to memory to disk. Each machine also needs additional disk capacity to store temporary data during processing with MapReduce. Rule of thumb for estimating machine raw disk capacity is 25% be reserved for temporary data. Each worker node in the cluster executes a predetermined number of map and reduce tasks simultaneously.

“A cluster administrator configures the number of these slots, and Hadoop’s task scheduler, a function of the jobtracker, assigns tasks that need to execute to available slots. Each one of these slots can be thought of as a compute unit consuming some amount of CPU, memory, and disk I/O resources, depending on the task being performed. A number of cluster-wide default settings dictate how much memory, for instance, each slot is allowed to consume. Since Hadoop forks a separate JVM for each task, the overhead of the JVM itself needs to be considered as well. This means each machine must be able to tolerate the sum total resources of all slots being occupied by tasks at once.”

Baseline configuration: 2x6 core 3 Ghz with 15 to 16GB cache CPU. 64 GB DDR 31600 ECC memory. SAS 6 GB/s disk controller. 12x3 TB LFF SATA II 7200 RPM disks. 2x1 GB Ethernet. (Or 24-48 GB memory: <http://docs.hortonworks.com/HDP2Alpha/index.htm#Hardware_Recommendations_for_Hadoop.htm> )

**Distributions.**

CDH is 100% open source. CDH compatible components include HBase, Hive, Pig, Sqoop, Flume, Zookeeper, Oozie, Mahout, Maven and Hue. Fully stable. CDH is available as tarballs, Redhat RPMs, Suse RPMs, and or a Debian Deb package. CDH packages have dependency on the Oracle RPM.

Long details on configuring YARN on CDH: <http://www.cloudera.com/content/www/en-us/documentation/enterprise/latest/topics/cdh_ig_yarn_cluster_deploy.html>

HDP offers virtual sandbox; and an option to manually install using RPMs. HDP (HMC) integrates with Ambari. Other required software includes YUM, SCP, Curl, Wget, and Pdsh on each host, and Firefox v.12+.

Summary of steps:

**Architecture**

Consider key factors that dictate implementation options.

Determine how software is to be configured on each server.

Consider data and size requirements.

**Preliminary setup**

**HDP deployment**

Via Ambari: YUM installs the HMC bits on CentOS; and HMC can be started. A deployment wizard accessed through browser will present a JDK download as the next step. IPtables must be stopped before deployment. <http://docs.hortonworks.com/HDP2Alpha/index.htm#Deploying_Hortonworks_Data_Platform/Using_HMC/Getting_Ready_to_Install/Getting_Ready_to_Install.htm>

Step one creates a cluster name. Step 2 adds nodes. Dependent on a private key, setup in SSH; as well as a hostdetail.txt file. 3 selects services. Yarn is mandatory and Nagios and ganglia are deployed automatically by the wizard. Other dependencies such as zookeeper for HBase are automatic. 4 assigns hosts. 5 selects mount points. 6 configures custom parameters for eight groups. Settings for each group are described in Hortonworks doc. The monitoring dashboard is accesses through the cluster management application. HBase requires append support.

Alternate flow: Deploying manually: an FQDN is required. Hosts are configured for dns, ntp is enabled, selinux is disabled. Additional steps if deploying behind a firewall / not on the internet. JDK gets manually installed. Hive and HCatalog require a MySQL instance. Desired services users (HDFS, Yarn, MapReduce, Hive, Pig, HCatalog, HBase, zookeeper and Oozie) need to be defined and have accounts created in the system. Manual RPM helper files can be downloaded. See dload companion files. Define directories for core Hadoop and ecosystem components.

**HDP confirmation**

Validate core install: Start HDFS, smoke test HDFS, start yarn, start MapReduce and smoke test. Install and validate services / ecosystem components. Configure ports.

**Cluster tuning**

**Install HUE**

**Configure HA**

**Manage cluster**

**Monitor cluster**

Higher level systems and tools must be wire and API compatible.

See workbench appendix for context on the following three implementation groups:

Group 1

Version control. Repository to manage code artifacts. TFS\*, Subversion, git, Mercurial. \*Basic TFS has no reporting, or Sharepoint exposure.

Build command. MSBuild, Powershell, Ant, Gradle, Maven

Automated code unit testing. Junit, MS Test (unit); Moq, Fluent Assertions (advanced)

Continuous integration. Jenkins, Hudson, TFS, Teamcity. <http://www.slideshare.net/excellaco/agile-engineering-best-practices-richard-cheng> early warnings of problems. CI server handles auto nightly deployments in sandbox; push button deployments in test environments; managed deployments in production environments.

Group 2

Static code analysis. Visual studio code a, findbugs (Java), PMD, cobertura, sonar, checkstyle

Dependency management. Gradle, nuget

Group 3

Automated integration testing. Higher effort. DBUnit, NDBUnit. Tests interaction between multiple components.

Automated acceptance testing. High effort. Specflow, Cucumber (dev lengua), Fitness. Browser testing: Selenium, Watin. Tests complete segments of a system.

Automated deployment. High effort. Fluentmigrator, **Puppet**, Octopus

“The fundamental idea of YARN is to split up the two major responsibilities of the MapReduce - JobTracker i.e. resource management, and job scheduling/monitoring, into separate daemons: a global ResourceManager and per-application ApplicationMaster (AM).”

Analysis: Mllib is Spark’s distributed.