UNIT IV – HADOOP ENVIRONMENT

QUESTION BANK

PART A (2 Marks each)

1. What is a Hadoop cluster? **V.Imp**
2. What are the advantages of Hadoop cluster?
3. What is Hadoop in the cloud?

PART B (5 Marks each)

4. Explain the specifications of a cluster.
5. How to setup and install a cluster? Explain its steps.
6. Explain Hadoop configuration.
7. Explain security in Hadoop. **V.Imp**
8. Explain Hadoop benchmarks.

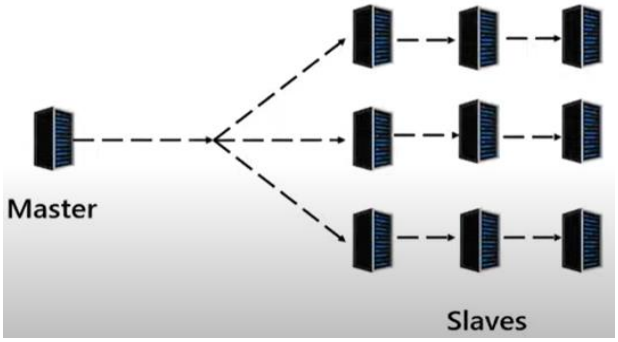
PART C (15 Marks each)

9. Explain the following:
    a) Administering Hadoop – HDFS **V.Imp**
    b) Hadoop Monitoring
    c) Hadoop Maintenance

NOTES

WHAT IS A HADOOP CLUSTER?

- A Hadoop cluster is a **collection of connected commodity computers**.
- They **worked together** and can be **viewed as a single system**.
- Hadoop clusters have **each node set to perform the same task,**
- **controlled** and **scheduled by Master Node.**
- Ex: **Facebook Hadoop cluster** is the biggest Hadoop cluster launched in 2004.
- The **purpose of Master nodes** is to **guide the slave nodes** in a single Hadoop cluster.
- We design Hadoop clusters for **storing, analysing, understanding, and for finding the facts that are hidden behind the data or datasets** which contain some **crucial information**.
- The Hadoop cluster **stores** different **types of data** and processes them.
    - Structured-Data
    - Semi-Structured Data
    - Unstructured Data



ADVANTAGES OF HADOOP CLUSTER

- Scalable
- Cost effective.
- Flexible
- Fast
- Resilient to failure.

CLUSTER SPECIFICATION

- Hadoop is **designed** to **run on commodity hardware**.
- It means that you are not tied to expensive, rather, you can choose standardized, commonly available hardware to build your cluster.
- "**Commodity**" does **not** mean "**low-end**."
- Low-end machines often have cheap components, which have higher failure rates.
- When you are operating tens, hundreds, or thousands of machines, cheap components turn out to be a false economy, as the **higher failure rate** incurs a **greater maintenance cost**.
- Hardware specifications rapidly become obsolete, but for the sake of illustration, a typical choice of machine for running a Hadoop datanode and tasktracker in mid-2010 would have the following specifications:
  - **Processor**
    2 quad-core 2-2.5GHz CPUs
  - **Memory**
    16-24 GB ECC RAM
  - **Storage**
    4 × 1TB SATA disks
  - **Network**
    Gigabit Ethernet
- **Hadoop is designed to use multiple cores and disks**, so it will be able to take full advantage of more powerful hardware.
- The bulk of Hadoop is written in **Java** and can therefore run on any platform with a **JVM**.
- **Windows operating systems** are not supported production platforms.
- For a **small cluster** (on the order of 10 nodes), it is usually acceptable to **run the name- node and the jobtracker on a single master machine**.
- As the **cluster and the number of files stored in HDFS grow**, the namenode needs more memory, so the **namenode and jobtracker should be moved onto separate machines**.
- The **secondary namenode** can be run on the same machine as the namenode, but again for reasons of memory usage (the secondary has the same memory requirements as the primary), it is best to **run** it **on a separate piece of hardware**, especially **for larger clusters.**

CLUSTER SETUP & INSTALLATION

- If have hardware, the next steps are to get it racked up (setup) and install the software needed to run Hadoop.

- The following are describe the customizations that are needed to run Hadoop:
  1) **Installing Java**:
     - **Java 6** or later is required to run Hadoop.
     - The latest stable **Sun JDK** is the preferred option.
  2) **Creating a Hadoop User**:
     - It is good practice to create a dedicated **Hadoop user account** to separate the Hadoop installation from other services running on the same machine.
  3) **Installing Hadoop**:
     - Download Hadoop from the Apache Hadoop releases page (http://hadoop.apache.org/core/releases.html), and unpack the contents of the distribution in a sensible location such as /usr/local.
  4) **Testing the Installation**:
     - Once you have created the installation file, you are ready to test it by installing it on the machines in your cluster.
     - This will probably take a few iterations as you discover kinks in the install.
     - When it is working, you can proceed to configure Hadoop and give it a test run.

HADOOP CONFIGURATION

- There are some important files for controlling the configuration of a Hadoop installation are listed in below table.

Filename	Format	Description
hadoop-env.sh	Bash script	Environment variables that are used in the scripts to run Hadoop.
core-site.xml	Hadoop configuration XML	Configuration settings for Hadoop Core, such as I/O settings that are common to HDFS and MapReduce.
hdfs-site.xml	Hadoop configuration XML	Configuration settings for HDFS the namenode, the sec-ondary namenode, and the datanodes.
mapred-site.xml	Hadoop configuration XML	Configuration settings for MapReduce the jobtracker, and the tasktrackers.
masters	Plain text	A list of machines (one per line) that each run a secondary namenode.
slaves	Plain text	A list of machines (one per line) that each run a datanode and a tasktracker.
hadoop-metrics.properties	Java Properties	Properties for controlling how metrics are published in Hadoop
log4j.properties	Java Properties	Properties for system logfiles, the namenode audit log, and the task log for the tasktracker child process

- These files are all found in the conf directory of the Hadoop distribution.

SECURITY IN HADOOP

- Hadoop assumed that HDFS and MapReduce clusters would be used by a group of cooperating users within a secure environment.
- The measures for restricting access were designed to prevent accidental data loss, rather than to prevent unauthorized access to data.
- For example, the file permissions system in HDFS prevents one user from accidentally wiping out the whole filesystem from a bug in a program.
- Secure authentication mechanism to assure Hadoop that the user seeking to perform an operation on the cluster.
- HDFS file permissions provide only a mechanism for authorization, which controls what a particular user can do to a particular file.
- For example, a file may only be readable by a group of users, so anyone not in that group is not authorized to read it.
- However, authorization is not enough by itself, since the system is still open to abuse via spoofing by a malicious user who can gain network access to the cluster.
- It's common to restrict access to data that contains personally identifiable information (such as an end user's full name or IP address) to a small set of users (of the cluster) within the organization, who are authorized to access such information.
- Less sensitive data may be made available to a larger set of users.
- However, to meet regulatory requirements for data protection, secure authentication must be in place for shared clusters.
- This is the situation that Yahoo! faced in 2009, which led a team of engineers there to implement secure authentication for Hadoop.
- In their design, Hadoop itself does not manage user credentials, since it relies on Kerberos, a mature open-source network authentication protocol, to authenticate the user.
- In turn, Kerberos does not manage permissions.
- Security has been tightened throughout HDFS and MapReduce to protect against un-authorized access to resources.
- The shuffle is secure, preventing a malicious user from requesting another user's map outputs.
- Users can view and modify only their own jobs, not others.

HADOOP BENCHMARKS

- Benchmarks make good tests; you can compare with other clusters as a sanity check on whether your new cluster is performing roughly as expected.
- And you can tune a cluster using benchmark results to squeeze the best performance out of it.
- To get the best results, you should run benchmarks on a cluster that is not being used by others.
- In practice, this is just before it is put into service and users start relying on it.
- Hadoop comes with several benchmarks that you can run very easily with minimal setup cost.
- Benchmarks are packaged in the test JAR file, and you can get a list of them, with descriptions, by invoking the JAR file with no arguments.

- **TestDFSIO** tests the I/O performance of HDFS.
- It is very useful for benchmarking the whole MapReduce system, as the full input dataset is transferred through the shuffle.
- The three steps are: generate some random data, perform the **sort**, then validate the results.
- **MRBench** runs a small job several times. It acts as a good counterpoint to sort, as it checks whether small job runs are responsive.
- **NNBench** is useful for load testing namenode hardware.
- **Gridmix** is a suite of benchmarks designed to model a realistic cluster workload.

HADOOP IN THE CLOUD

- Run Hadoop in the cloud on rented hardware or as a service.
- For instance, **Cloudera** offers tools for running Hadoop in a public or private cloud, and **Amazon** has a Hadoop cloud service called **Elastic MapReduce**.

Hadoop on Amazon EC2

- **Amazon Elastic Compute Cloud (EC2)** is a computing service that allows customers to rent computers (instances) on which they can run their own applications.
- A customer can launch and terminate instances on demand, paying by the hour for active instances.
- Which is a great way to try out your own Hadoop cluster on a low commitment.

ADMINISTERING HADOOP- HDFS

- It means that look at the procedures to keep a cluster running smoothly.
- As an administrator, it is important to have a basic understanding of how the components of HDFS—the namenode, the secondary namenode, and the datanodes organize their persistent data on disk.
- Knowing which files are which can help you diagnose problems or spot that something is awry.
- **Safe mode** is used to ensure that namenode is in safe.
- An administrator has the ability to make the namenode enter or leave safe mode at any time.
- It is sometimes necessary to do this when carrying out maintenance on the cluster or after upgrading a cluster to confirm that data is still readable.
- To enter safe mode, use the following command:

```
% hadoop dfsadmin -safemode enter
Safe mode is ON
```

MONITORING

- Monitoring is an important part of system administration.
- The purpose of monitoring is to detect when the cluster is not providing the expected level of service.

- The master nodes are the most important to monitor: the namenodes (primary and secondary) and the JobTracker.
- Failure of datanodes and TaskTrackers is to be expected, particularly on larger clusters, so should provide extra capacity so that the cluster can tolerate having a small percentage of dead nodes at any time.
- Some administrators run test jobs on a periodic basis as a test of the cluster's health.
- For example, **Chukwa** is a data collection and monitoring system built on HDFS and MapReduce.
- **Logging**: All Hadoop daemons produce logfiles that can be very useful for finding out what is happening in the system.
- **Metrics**: The HDFS and MapReduce collect information about events and measurements that are collectively known as **metrics**. For example, datanodes collect the following metrics: the number of bytes written, the number of blocks replicated, and the number of read requests from clients etc.

MAINTENANCE

- **Metadata backups**:
  - If the namenode's persistent metadata is lost or damaged, the entire filesystem is rendered unusable, so it is critical that backups are made of these files.
  - You should keep multiple copies of different ages (one hour, one day, one week, and one month, say) to protect against corruption.
- **Data backups**:
  - Although HDFS is designed to store data reliably, data loss can occur, just like in any storage system, and thus a backup strategy is essential.
  - With the large data volumes that Hadoop can store, deciding what data to back up and where to store it is a challenge.
  - The key here is to prioritize your data.
  - The highest priority is the data that is critical to the business; however, data that is straightforward to regenerate, or essentially disposable because it is of limited business value, is the lowest priority, and you may choose not to make backups of this category of data.
- **Filesystem check (fsck)**
  - It is advisable to run HDFS's fsck tool regularly (example, daily) on the whole file system to proactively look for missing or corrupt blocks.
- **Filesystem balancer**:
  - Run the balancer tool regularly to keep the filesystem datanodes evenly balanced.