Big Data Analysis

Hadoop Architecture and HDFS

Homework -2

1. What are HDFS and YARN?

Hadoop Distributed File System

HDFS is a Java-based file system that provides scalable and reliable data storage, and it was designed to span large clusters of commodity servers. HDFS has demonstrated production scalability of up to 200 PB of storage and a single cluster of 4500 servers, supporting close to a billion files and blocks. When that quantity and quality of enterprise data is available in HDFS, and YARN enables multiple data access applications to process it, Hadoop users can confidently answer questions that eluded previous data platforms.

YARN (Yet Another Resource Negotiator)

YARN is essentially a system for managing distributed applications. It consists of a central Resource manager (RM), which arbitrates all available cluster resources, and a per-node Node Manager (NM), which takes direction from the Resource manager. The Node manager is responsible for managing available resources on a single node.

2. What are the various Hadoop daemons and their roles in a Hadoop cluster?

Daemons mean Process. Hadoop Daemons are a set of processes that run on Hadoop. Hadoop is a framework written in Java, so all these processes are Java Processes.

Apache Hadoop 2 consists of the following Daemons:

- NameNode
- DataNode
- Secondary Name Node
- Resource Manager
- Node Manager

Namenode, Secondary NameNode, and Resource Manager work on a Master System while the Node Manager and DataNode work on the Slave machine**.**

3. **Why does one remove or add nodes in a Hadoop cluster frequently?**

Basically, in a Hadoop cluster a Manager node will be deployed on a reliable hardware with high configurations, the Slave node's will be deployed on commodity hardware. So chance's of data node crashing is more. So more frequently you will see admin's remove and add new data node's in a cluster.

Also adding of more data node's in the cluster is done to handle large volume of data.

4. **. What happens when two clients try to access the same file in the HDFS?**

HDFS provides support only for exclusive writes so when one client is already writing the file, the other client cannot open the file in write mode. When the client requests the NameNode to open the file for writing, NameNode provides lease to the client for writing to the file. So, if another client requests for lease on the same it will be rejected.

5. **How does NameNode tackle DataNode failures?**

Hadoop file system is a master/slave file system in which Namenode works as the master and Datanode work as a slave. Namenode is so critical term to Hadoop file system because it acts as a central component of HDFS. If Namenode gets down then the whole Hadoop cluster is inaccessible and considered dead. Datanode stores actual data and works as instructed by Namenode. A Hadoop file system can have multiple data nodes but only one active Namenode.

6. What will you do when NameNode is down?

The NameNode recovery process involves the following steps to make the Hadoop cluster up and running:

1. Use the file system metadata replica (FsImage) to start a new NameNode.

2. Then, configure the DataNodes and clients so that they can acknowledge this new NameNode, that is started.

3. Now the new NameNode will start serving the client after it has completed loading the last checkpoint FsImage (for metadata information) and received enough block reports from the DataNodes.

Whereas, on large Hadoop clusters this NameNode recovery process may consume a lot of time and this becomes even a greater challenge in the case of the routine maintenance. Therefore, we have HDFS High Availability Architecture which is covered in the HA architecture blog.

7. How is HDFS fault tolerant?

The HDFS is highly fault-tolerant that if any machine fails, the other machine containing the copy of that data automatically become active. Distributed data storage - This is one of the most important features of HDFS that makes Hadoop very powerful. Here, data is divided into multiple blocks and stored into nodes.

8. Why do we use HDFS for applications having large data sets and not when there are a lot of small files?

HDFS is more efficient for a large number of data sets, maintained in a single file as compared to the small chunks of data stored in multiple files. As the NameNode performs storage of metadata for the file system in RAM, the amount of memory limits the number of files in HDFS file system.

9. How do you define "block" in HDFS? What is the default block size in Hadoop 1 and in Hadoop 2? Can it be changed?

Blocks are the smallest continuous location on your hard drive where data is stored. HDFS stores each file as blocks, and distribute it across the Hadoop cluster. The default size of a block in HDFS is 128 MB (Hadoop 2.x) and 64 MB (Hadoop 1.x) which is much larger as compared to the Linux system where the block size is 4KB.

Yes, I can change the block size of HDFS files by changing the default size parameter present in hdfs-site.xml. But, I will have to restart the cluster for this property change to take effect.