**Assignment 2 – Hoang Duy Vu - 986170**

1. **For NameNode, why it’s not necessary to store block locations persistently?**

NameNode does not store block locations persistently, because this information is reconstructed from datanodes when the system starts

1. **Why is it important to make the NameNode resilient to failures?**

Without the namenode, the filesystem cannot be used

1. **What details are there in the FsImage file?**

Fsimage store list of files, list of blocks for each file, list of DataNodes for each block, file permissions, modification, access times, replication factor, etc.

1. **What is the purpose of the secondary name-node?**

Creating a new FsImage is an I/O- and CPU intensive operation, sometimes taking minutes to perform. So, rather than pausing the active NameNode to perform a checkpoint, HDFS defers it to the SecondaryNameNode.

1. **Does the NameNode stay in the safe mode until all under-replicated files are fully replicated? Why or why not?**

No, in safe mode, only filesystem operations that access the filesystem metadata (such as producing a directory listing) are guaranteed to work. Reading a file will work only when the blocks are available on the current set of datanodes in the cluster, and file modifications (writes, deletes, or renames) will always fail. After a configurable percentage of safely replicated blocks check in with the NameNode, NameNode exits Safemode then it makes the list of blocks that need to be replicated.

1. **What are the core changes in Hadoop 2.x compared to Hadoop 1.x? In other words, state the major differences between Hadoop 1 and Hadoop 2.** **High Availability – Taking care of NameNode SPOF problem**

Core change in Hadoop 2.x

YARN - Taking care of JobTracker SPOF and added support for non-mapreduce type of processing (multitenancy) making MapReduce as a user library, or one of the applications residing in Hadoop.

HDFS Federation - Added support for multiple namespaces with multiple NNs.

High Cluster Utilization - Use of variable-sized Containers instead of fixed-size Slots mechanism

Improved Scalability - Hadoop 2.x supports more than 10,000 nodes per cluster.

MRv2 (simply MRv1 rewritten to run on top of YARN) – no need to rewrite existing MapReduce jobs

1. **What is the difference between MR1 in Hadoop 1.0 and MR2 in Hadoop2.0?**

MRv2 (simply MRv1 rewritten to run on top of YARN) – no need to rewrite existing MapReduce jobs

In MapReduce 1, there are two types of daemon that control the job execution process: a jobtracker and one or more tasktrackers while MR2 run on YARN. It is the familiar MapReduce 1 execution underneath, except that each job now controls its own destiny via its own ApplicationMaster taking care of execution flow (such as scheduling tasks, handling speculative execution and failures, etc.)

1. **What is HDFS Federation? What advantage does it provide?**

HDFS Federation improves the existing HDFS architecture through a clear separation of namespace and storage, enabling generic block storage layer. It enables support for multiple namespaces in the cluster to improve scalability and isolation. Federation also opens up the architecture, expanding the applicability of HDFS cluster to new implementations and use cases.

Key benefit:

* Support for multiple namenodes horizontally scales the file system namespace. It separates namespace volumes for users and categories of applications and improves isolation.
* Block pool abstraction opens up the architecture for future innovation
* It is significantly simpler to design and implement. Namenodes and namespaces are independent of each other and require very little change to the existing namenodes

1. **What is NameNode High Availability and how is it achieved in Hadoop 2?**

HA in Hadoop 2 is HDFS now has automated failover with a hot standby, with full stack resiliency meaning:

The Active NameNode is responsible for all client operations in the cluster. The Standby NameNode maintains enough state to provide a fast failover. In order for the Standby node to keep its state synchronized with the Active node, both nodes communicate through a group of separate daemons called JournalNodes. The file system journal logged by the Active NameNode at the JournalNodes is consumed by the Standby NameNode to keep it’s file system namespace in sync with the Active.

In order to provide a fast failover, it is also necessary that the Standby node have up-to-date information of the location of blocks in your cluster. DataNodes are configured with the location of both the NameNodes and send block location information and heartbeats to both NameNode machines

1. **What is the role of Application Master in YARN application execution?**

An ApplicationMaster for executing shell commands on a set of launched containers using the YARN framework. The ApplicationMaster is started on a container by the ResourceManager's launcher. The first thing that the ApplicationMaster needs to do is to connect and register itself with the ResourceManager. The registration sets up information within the ResourceManager regarding what host:port the ApplicationMaster is listening on to provide any form of functionality to a client as well as a tracking url that a client can use to keep track of status/job history if needed