Assignment- 09/08/2022

* **what is FSImage?**

>> FsImage is file which stores complete snapshot of metadata, all the information about the blocks belongs to file or filesystem property.

FsImage is a file stored on the OS filesystem that contains the complete directory structure (namespace) of the HDFS with details about the location of the data on the Data Blocks and which blocks are stored on which node. This file is used by the Namenode when it is started.

* **What is Rack Awareness?**

>> A rack is nothing but a collection of Data nodes. These Data nodes in a rack are connected to the Namenode through traditional network design via a network switch.

The process of making Hadoop aware of what Data node is part of which rack and how these racks are connected to each other within the Hadoop cluster is what defines rack awareness.

Namenode keeps the rack ids of all the Data nodes. Namenode select the closest Data node when storing the data blocks using the rack information.

Having the knowledge of how different data nodes are distributed across the racks in Hadoop cluster is called rack awareness.

Rack awareness is important as it ensures data reliability and helps to recover data in case of a rack failure.

Hadoop keeps multiple copies for all data that is present in blocks. Each copy of data can be kept in a different rack. By doing this, in case an entire rack suffers a failure for some reason, the data can be retrieved from a different rack is known as replica of rack.

Replication of data blocks in multiple racks via rack awareness is called Replica Replacement Policy.

The policy states that “No more than one replica is placed on one node. And no more than 2 replicas are placed on the same rack.”

* **How 6 is maintained and what is under/over replicated blocks?**

>> Under Replicated Blocks: These are the blocks that do not meet their target replication for the files they belong to. HDFS will automatically create new replicas of under-replicated blocks until they meet the target replication.

Over Replicated Blocks: These are the blocks that exceed their target replication for the files they belong to. Usually, over-replication is not a problem, and HDFS will automatically delete excess replicas.

Data replication must reduce data file transfer time, bandwidth consumption and maintain the consistency between the data and replica nodes. The centralized replication that reduces the total data file access delay and caching algorithm is used for any replica server to join and to leave from the main server. An Integrated File Replication and Consistency Maintenance mechanism algorithm is used to achieve high efficiency in data replication and consistency maintenance at a low cost. Each replica server determines data replication and update polling by dynamically adapting to time varying file query and file update rates. Poll reduction process is to avoid unnecessary updates.

* **What is identity mapper and chain mapper?**

>> The Identity mapper is one of the pre-defined mapper classes that can be used with any key/value pairs of data. It is a generic class and also the default mapper class provided by the Hadoop. When no mapper class is specified in the MR Driver class, the Identity Mapper class is invoked automatically when a Map-Reduce job is assigned.

The Chain Mapper is also one of the pre-defined mapper classes that allows using multiple mapper class within a single Map task. All the mappers are run in a chain fashioned, that is the output of the first mapper becomes the input of the second mapper and so on. The output of the last mapper class is written to the intermediate files.

* **Difference between external and internal tables in hive**

|  |  |
| --- | --- |
| External Table | Internal Table |
| 1) Hive does not manage the data of the External table. | 1) Hive owns the data for the internal tables |
| 2) We create an external table for external use as when we want to use the data outside the Hive. | 2) It is the default table in Hive. When the user creates a table in Hive without specifying it as external, then by default, an internal table gets created in a specific location in HDFS. |
| 3) External tables are stored outside the warehouse directory. They can access data stored in sources such as remote HDFS locations or Azure Storage Volumes. | 3) an internal table will be created in a folder path like /user/hive/warehouse directory of HDFS. We can override the default location by the location property during table creation. |
| 4) We can create the external table by specifying the EXTERNAL keyword in the Hive create table statement. | 4) If we drop the managed table or partition, the table data and the metadata associated with that table will be deleted from the HDFS. |
| 5) Dropping the external table will delete only table metadata. The table content remains untouched. | 5) Dropping the internal table will delete the table data, as well as the metadata associated with the table. |
| 6) With EXTERNAL keyword, Hive knows that it isn’t managing the table data, so it doesn’t move data to its warehouse directory. Hive doesn’t even check whether the external location at the time it is defined exists or not. | 6) When we load data into an internal table, then Hive moves data into the warehouse directory. |
| 7) The TRUNCATE command doesn’t work for the external table. | 7) The TRUNCATE command only works for the internal table. |
| 8) ACID/transactional not works for the external table. | 8) ACID/transactional works only for the internal table. |

* **How does Yarn allocate resources to an application?**

>>Resource Manager: It is the master daemon of YARN and is responsible for resource assignment and management among all the applications. Whenever it receives a processing request, it forwards it to the corresponding node manager and allocates resources for the completion of the request accordingly. It has two major components:

Scheduler: It performs scheduling based on the allocated application and available resources. It is a pure scheduler, means it does not perform other tasks such as monitoring or tracking and does not guarantee a restart if a task fails. The YARN scheduler supports plugins such as Capacity Scheduler and Fair Scheduler to partition the cluster resources.

Application manager: It is responsible for accepting the application and negotiating the first container from the resource manager. It also restarts the Application Master container if a task fails.

* **Which command is used to check status of any application ran by yarn**

>> Call "yarn application -list": Shows all applications in states "SUBMITTED", "ACCEPTED" and "RUNNING"

* **What happens if a Resource Manager fails while executing an application in a high availability cluster?**

>> When a Resource Manager terminated and is restarted or fails over to another Resource Manager in the case of a high availability cluster, the newly active Resource Manager instructs running Application Masters to abort. This uses up an application attempt.

Also, if the Resource Manager is down for some time and the Application Master is unable to connect, it will timeout and abort. That uses up an application attempt too.

When a new Resource Manager becomes active, it can recover applications with failed attempts that have not exceeded their max-attempts.

* **How to calculate the number of executors given to a spark submit job?**

Number of available executors = (total cores/num-cores-per-executor)

* **How to calculate total processing capacity of a cluster?**

Storage Considerations: The number of data nodes you need is determined by the size of the data, how it will be analyzed, and the number of replicas you will have. By default, Apache Hadoop has 3 copies.

In this case, if we want to store X GB of data, we need X\*3 GB of storage for the period.

Processing Considerations: In addition to having enough space to store your data, you will need room for data processing, computing, and miscellaneous other tasks.

We can assume that, on an average day, only 10% of data is being processed, and a data process creates three times temporary data. Therefore, you need to account for around 30% of your total storage as extra space.

Number of Data Nodes Required: The final calculation for the number of data nodes required for your system will be dependent on your JBOD (“just a bunch of disks”) capacity.

For example: Let’s say that you need 500GB of space. If you have a JBOD of 12 disks, and each disk can store 6TB of data, then the data node capacity, or the maximum amount of data that each node can store, will be 72 TB. Data nodes can be added as the data grows, so to start with its better to select the lowest number of data nodes required.