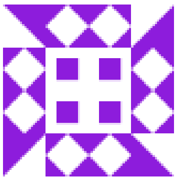
Hadoop Cluster Capacity Planning of Data Nodes for Batch and In-Memory Processes

Let's talk about capacity planning for data nodes. We'll start with gathering the cluster requirements and end by learning about RAM requirements.



by
Mamta Chawla
·

Jul. 14, 17 · Big Data Zone · Tutorial

Here, I am sharing my experience setting up a Hadoop cluster for processing approximately 100 TB data in a year. The cluster was set up for 30% realtime and 70% batch processing, though there were nodes set up for NiFi, Kafka, Spark, and MapReduce. In this blog, I mention capacity planning for data nodes only. In next blog, I will explain capacity planning for name node and Yarn. Here is how we started by gathering the cluster requirements.

Prerequisites

While setting up the cluster, we need to know the below parameters:

- What is the volume of data for which the cluster is being set? (For example, 100 TB.)
- The retention policy of the data. (For example, 2 years.)
- The kinds of workloads you have — CPU intensive, i.e. query; I/O intensive, i.e. ingestion, memory intensive, i.e. Spark processing. (For example, 30% jobs memory and CPU intensive, 70% I/O and medium CPU intensive.)
- The storage mechanism for the data — plain Text/AVRO/Parque/Jason/ORC/etc. or compresses GZIP, Snappy. (For example, 30% container storage 70% compressed.)

Data Nodes Requirements

With the above parameters in hand, we can plan for commodity machines required for the cluster. (These might not be exactly what is required, but after installation, we can fine tune the environment by scaling up/down the cluster.) The nodes that will be required depends on data to be stored/analyzed.

By default, the Hadoop ecosystem creates three replicas of data. So if we go with a default value of 3, we need storage of *100TB *3=300 TB* for storing data of one year. We have a retention policy of two years, therefore, the storage required will be *1 year data* retaention period=300*2=600 TB*.

Assume 30% of data is in container storage and 70% of data is in a Snappy compressed Parque format. From various studies, we found that Parquet Snappy compresses data to 70-80%.

We have taken it 70%. Here is the storage requirement calculation:

total storage required for data =total storage % in container storage + total storage * %in compressed format*expected compression600*.30+600*.70*(1-.70)=180+420*.30=180+420*.30=306 TB.*

In addition to the data, we need space for processing/computation the data plus for some other tasks. We need to decide how much should go to the extra space. We also assume that on an average day, only 10% of data is being processed and a data process creates three times temporary data. So, we need around 30% of total storage as extra storage.

Hence, the total storage required for data and other activities is *306+306*.30=397.8 TB*.

As for the data node, JBOD is recommended. We need to allocate 20% of data storage to the JBOD file system. Therefore, the data storage requirement will go up by 20%. Now, the final figure we arrive at is *397.8(1+.20)=477.36 ~ 478 TB*.

Let's say *DS=478 TB*.

Number of Data Nodes Required

Now, we need to calculate the number of data nodes required for 478 TB storage. Suppose we have a JBOD of 12 disks, each disk worth of 4 TB. Data node capacity will be 48 TB.

The number of required data nodes is *478/48 ~ 10*.

In general, the number of data nodes required is *Node= DS/(no. of disks in JBOD*diskspace per disk)*.

Note: We do not need to set up the whole cluster on the first day. We can scale up the cluster as data grows from small to big. We can start with 25% of total nodes to 100% as data grows.

Now, let's discuss data nodes for batch processing (Hive, MapReduce, Pig, etc.) and for in-memory processing.

As per our assumption, 70% of data needs to be processed in batch mode with Hive, MapReduce, etc.

10.70=7* nodes are assigned for batch processing and the other 3 nodes are for in-memory processing with Spark, Storm, etc.

CPU Cores and Tasks per Node

For batch processing, a *2*6*-core processor (hyper-threaded) was chosen, and for in-memory processing, a *2*8* cores processor was chosen. For batch processing nodes, while one core is counted for CPU-heavy processes, .7 core can be assumed for medium-CPU intensive processes. As we have assumption, 30% heavy processing jobs and 70% medium processing jobs, Batch processing nodes can handle [(no. of cores* %heavy processing jobs/cores required to process heavy job)+ (no. of cores* %medium processing jobs/cores required to process medium job)]. Therefore tasks performed by data nodes will be;

12.30/1+12*.70*/.7=3.6+12=15.6 ~15* tasks per node.

As hyperthreading is enabled, if the task includes two threads, we can assume 15*2~30 tasks per node.

For in-memory processing nodes, we have the assumption that *spark.task.cpus=2* and *spark.core.max=8*2=16*. With this assumption, we can concurrently execute *16/2=8* Spark jobs.

Again, as hyperthreading is enabled, the number of concurrent jobs can be calculated as *total concurrent jobs=no. of threads*8*.

RAM Requirement for a Data Node
Now, let's calculate RAM required per data node. RAM requirements depend on the below parameters.

*RAM Required=DataNode process memory+DataNode TaskTracker memory+OS memory+CPU's core number *Memory per CPU core*
At the starting stage, we have allocated four GB memory for each parameter, which can be scaled up as required. Therefore, RAM required will be *RAM=4+4+4+12*4=60* GB RAM for batch data nodes and *RAM=4+4+4+16*4=76* GB for in-memory processing data nodes.

The steps defined above give us a fair understanding of resources required for setting up data nodes in Hadoop clusters, which can be further fine-tuned. In next blog, I will focus on capacity planning for name node and Yarn configuration.