



Big Data Capacity Planning: Achieving the Right Size of the Hadoop Cluster

Author: Nitin Jain



Abstract

As the data analytics field is maturing, the amount of data generated is growing rapidly and so is its use by businesses. This increase in data helps improve data analytics and the result is a continuous circle of data and information generation. In order to handle these new volumes of data, IT organizations must right-size their Hadoop clusters to balance the OPEX and CAPEX. This paper details key dimensioning techniques and principles that help achieve an optimized size of a Hadoop cluster.

Understanding the Big Data Application

Big data applications running on a Hadoop cluster can consume billions of records a day from multiple sensors or locations. Applications process terabytes of data which can generate valuable insights to be consumed in real-time or periodically. Real-time consumption requires a more stringent query SLA and higher memory footprint as compared to periodic updates which require lower memory footprint, but higher disk volumes.

Role of Infrastructure in Sizing

With the advance of computing frameworks such as Hadoop, Spark, MapReduce and Storm, there are many variations of infrastructure to support big data applications. Now, these applications can be deployed on physical machines or virtual machines on-premise, in a private cloud or on the public cloud. The performance of an application varies drastically depending on the choice made for the infrastructure.

Key Considerations and Recommendations

Key Considerations	Recommendation
Input Volume Rate	For real time and hourly insights, peak data rates should be considered. For daily insights, median rates should be considered. For weekly insights, average data rates should be considered.
RAID Configuration	Replication factor is often mistakenly considered as replacement for RAID. Replication factor ensures higher data locality, but RAID ensures data safety at a physical level. Use both Replication Factor and RAID for highly precious data.
Data Purging	Different stages have different SLAs and each stage requires data cleanup which requires an extra 'write operation' on disk. This should be added while calculating disk IOPS.
Infrastructure	For the same CPU, RAM and disk family, the performance is best on a physical deployment; it is about 20-30% lower on virtual machines or private clouds; and is about 60-70% lower on a public cloud. Most of the public clouds offer only 1 CPU thread.
Data Growth	Day-by-day data is increasing. Since big data applications are a long-term investment, growth factor should be considered in defining the size of cluster. Ideally, it should be QoQ growth, but YoY growth can be considered for ease of procurement.
Resources Per Process	When an application gets deployed it runs a number of smaller services like Ingestion, Fusion, Analysis, Publication, etc. For each one of these services, RAM, CPU, IOPS and disk storage must be used.



Formulae

1. HDFS Storage Calculation

Let's say that Application A runs 1 service in the background and it requires X CPU, Y amount of memory for Z data rate/sec. 1 record is of size B, so the 1-day storage will be $S_a = R * B * 86400 / 10^9$ GB. Now consider the replication factor of HDFS and multiply it further. This number should also be updated based on the RAID configuration: if RAID is 0 then use overload factor of 1; if the RAID is 5 then use the overload factor of 1.5 and in RAID 10, use the overload factor of 2.

2. Cluster Size as per Environment

Let's say from #1 the size of storage is Shdfs, memory is Mhdfs and CPU is Chdfs then storage should remain constant irrespective of environment, but RAM and CPU should be increased to cater to environment overheads. In the Virtualization environment, whether its in a private or public cloud, the biggest hit comes in the form of network and disk throughput.

3. Growth Factor

Depending on growth factor of the data (say G), multiply all the resources with G

$$S_{final} = Shdfs * (1 + G)$$

$$M_{final} = Mhdfs * (1 + G)$$

$$C_{final} = Chdfs * (1 + G)$$

$$IOPS_{final} = IOPShdfs * (1 + G)$$



Conclusion

The technology revolution including the availability of billions of data records, software advances, new frameworks, and the availability of powerful hardware, has made big data processing possible. However, optimal sizing of the cluster is equally important for an application to continue to generate valuable insight. Hardware CAPEX represents a significant investment upfront and requires recurring OPEX, hence a balance between the use of an application and the sizing should be done for a given data rate. Sizing applications for monthly insights at peak data rates may not be the right decision to make, unless your use case demands this. You will want to work closely with the people that will be running and using the application on a regular basis to understand their business requirements and size accordingly.

References

Big data applications built in Guavus