IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 3, Ver. I (May-Jun. 2016), PP 30-34
www.iosrjournals.org

Performance Analysis of Hadoop Application For Heterogeneous Systems

Naresh E[1], Poornesha B D[2], Vijaya Kumar B P[3]
[1] *Research Scholar, CSE, Jain University, Bangalore, Karnataka, India*
[1, 2, 3] *Department of ISE, MSRIT, MSR Nagar, Bangalore, Karnataka, 560054, India*

Abstract: *Apache Hadoop is open source software that allows for the processing of large data set sin nodes. This work involves non-functional testing of Hadoop application in order to get the observable performance of an application. This work also includes the study and observations of performance analysis of Hadoop application for various machine architectures like Intel Core 2 Duo, and Intel i3.*

Keywords: *BigData, Split files, Fuunctional testing, Hadoop throughput and performance, Mappers, Reducers.*

I. Introduction

As the technology evolves huge amount of data is getting generated in day to day life by Sensors, Smartphone, Mobile devices and by web applications, there are three dimensions involved in generating the data they are Volume, Variety and Velocity(3V's) [1] [4]. Volume defines the amount of data generated from multiple systems, which needs to be processed and analyzed. Variety defines type of data getting generated like structured data, semi-structured data and unstructured data, analyzing of these types of data is a tedious jobs so some scripting techniques can be adopted to analyze these data and involves lot of effort. Velocity defines the rate at which the data is generated due to digitalization, by real-time systems and mission critical application, this leads to the emergence of new domain-Big Data in organizations. Testing of this Big Data is crucial task and organizations are worried in handling of this data because of lack of knowledge on what to test and how to test [5]. Organizations are facing many challenges in forming test strategies for structured and unstructured data, setting up on environment and functions of components in Big Data model results in poor quality of data in production hit the response times very badly [7].

In order to manage accuracy of this data there are many testing types like functional and non-functional testing are essential along with error free test data and environment. Functional testing involves testing of Hadoop map reduce process and input data validation to ensure both input and output are in good quality and validating a extracted data before storing in downstream system [1]. Non functional testing involves testing of performance of the system plays a key role in accepting the model.

II. Functional Testing

Hadoop distributed file system (HDFS) makes input data available to multiple nodes. These input data can be extracted from multiple sources in structured or unstructured manner and loaded into HDFS by splitting files in multiple numbers and process, these multiple files using map and reduce operations finally extract the data generated from HDFS and store into downstream systems.

As we are working with huge data and processing at multiple nodes, there are chances of missing quality data and inject of unrelated data leads to quality issues at each stage of the process. Data functional testing involves the testing of the data at three phases, they are validation of Hadoop pre-processing, validation of Hadoop map-reduces process and validation of Extracted data and store in downstream systems shown in the below Hadoop Map-Reduce Architecture [2] [5].

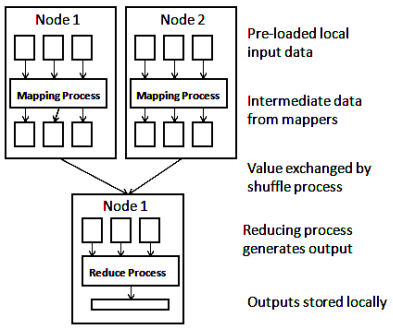

Figure 1: Hadoop Map-Reduce Architecture

Figure 1 shows the Hadoop framework for distributed processing of large datasets using map and reduce in any node of a system, initially pre-loaded input data is loaded into the HDFS for processing, map process split the data and store the intermediate key-value pair's data from mappers. These key-value pairs are shuffled based on the key before sending into reduce phase, reduce phase performs the reduce operation and stores the generated output in local file system [9].

2.1 Validation of Hadoop Pre-processing

Data can be extracted from various sources and load into HDFS for processing, there are many issues involved during this process like data not in format, incorrect data captured from source systems, incorrect split of data files and incorrect storage of data. Validation of Hadoop pre-processing includes:

- Comparing the extracted input file format with the source data format and ensure data extracted in correct format.
- Data captured in input files should match the source data.
- Proper splits can be made and its compatible with the HDFS distributed file system.
- Input files should be bigger in size, Hadoop reads the blocks each of size may be in multiple of 64MB, it's good to have the file size in multiple of thousands of 64MB [3] [12] [13].

2.2 Validation of Hadoop map reduces process

Once the files are loaded into HDFS, now its Hadoop map-reduce job to process the input files from different sources, there are many issues involved in these process includes jobs run correctly in single node but not in multiple nodes, incorrect aggregation of data in reduce phase, incorrect output data format, coding error in map-reduce phase, mapper spill size. Validation of Hadoop map-reduce process includes:

- Tune number of map and reduce tasks appropriately [11].
- Validation of data processing and output files generated, output file format should be as per the requirement specified.
- Compare the output file with the input source file formats.
- Verify and validate the business logic configuration in single node against the multiple nodes [6].
- Consolidate and validate the data after the reduce phase.
- Validate the key value pair generated in map and reduce phase and verify the shuffling activity thoroughly [8] [12].
- Size of the mapper output is sensitive to disk IO, network IO and memory sensitive on shuffle phase, use of minimal map output key-map output value and compressing of mapper output and minimizing the mapper output will give the observable performance, this can be done by filter out records on mapper side instead of reducer side [3].

Compress the intermediate output of mappers before transmitting to reducers, which will reduce the less disk I/O and network traffic from reading and moving files around [13].

2.3 Validation of extracted data and store in downstream systems

After completion of data processing, output files are generated and moved to downstream systems for loading. There are few issues involved in this phase like incorrect rule transformation is applied, incorrect loading of files to downstream databases, output files format not compatible with the storage format.

Validation of post processing includes:

- Validation of transaction rules before extracting the data.
- Validation of data corruption in output files and validation of data integrity in a target system.
- Comparing storage database format and output file format, setting of constraints rules in the target system [6].

III. Non Functional Testing

This type of testing involves Performance testing, Failover testing and Load testing to identify any bottleneck involved in the process.

3.1 Performance Testing

Any big data project deals with processing of huge volumes of data in a less time using multiple nodes, this data may be structured, unstructured. Performance will be hit badly because of poor design of architecture, low level coding standards and poor configuration of parameters so such systems are not meeting the user expectations for huge and complex data. Apart from the above, performance will hit badly due to improper input files, most of the map processing jobs are running at reduce phase such as aggregation of data, redundant

sorts and shuffle tasks[5], Communication delay between map to reduce phase. These performance issues can be resolved by:

- Maintaining a huge file size of input files.
- Carefully designing the system architecture by considering the design constructs and performing performance test to identify bottleneck.
- Performance test can be carried out by setting up data in huge volume and infrastructure similar to production
- Using Hadoop performance monitoring tool capture performance metrics such as job completion time, caching percentage, throughput, CPU utilization, memory utilization, top number of process consuming CPU etc.

3.2 Failover Testing

Hadoop architecture consists of HDFS components, job and task tracker, name node and many data nodes. There are many chances like components of Hadoop become non functional due to name or data node failure, job or task tracker failures etc. Architecture can be designed such a way that it should automatically rectify or recover from the problem and start processing data.

Failover testing is a important in Hadoop implementation and few validations need to be performed are frequent checking of edit logs, back up of name node which holds the metadata information, no data loss during the data node failures, operation should not halt if any of the data node fails or during replacement of data nodes, data replication during the new data nodes, replication is initiated whenever data node fails or data become corrupted. There should be constant schedule for Recover Point Objective (RPO) and Recovery Time Objective (RTO) in a system and metrics can be captured for RPO and RTO [6].

3.3 Load Testing

Its necessary to measure the Hadoop system behavior under both normal and peak load conditions to identify maximum operational capacity of an application, any bottlenecks in the application and which element causing the performance degradation. For the big data analysis Hadoop system should have capable enough to process vast amount of data in the form of files from different sources. Load testing can be carried out in three stages Physical testing, Dynamic testing and Cyclical testing [15].

Physical testing involves providing the constant input file load for a specified amount of time. Dynamic testing involves providing a variable input file load to map-reduce operations. Cyclical testing consists of repeated unloading and loading of input files for specified durations and conditions [10]. For each of this load testing measure the performance of the system using the below captured metrics:

- Calculating the average and peak response time of the map-reduce phase, this can be calculated by time taken to process the input files and store the computed data in downstream systems.
- Capturing the CPU and memory utilization.
- Behavior of map and reduce phases and measuring the response time for each.
- Throughput that is number of input files (size) processed per second, Error rate and how many concurrent users can work on the system.

IV. Analysis And Results

A We have conducted a experiments with the 4GHZ Core 2 Duo processor, RAM of 4GB and Hard Disk of 100GB in Ubuntu 12.04 LTS and Apple Intel Core i3 64-bit single node system in order to calculate the performance of the single node system in terms of number of input file size processed per unit of time i.e throughput of the Hadoop application [14], it can be calculated by size of the input file processed per total amount of CPU time taken in Milli Seconds.

Input to the Hadoop application consists of the variable file sizes, at the beginning of the experiments, we have split the 1GB file size into 40 files, 20 files, 10 files of equal size and next we proceed with the single file of bigger sizes to process the data in a file, The experiments is carried with the Hadoop configured with the two mapers and one reducer in Core 2 duo and Intel Core i3 Systems as shown in the below Table 1.

Sl.No	Files	Total File Size	Core 2 Duo Throughput	Intel Core i3 Throughput
1	40	1GB	3452.5	4098.3
2	20	1GB	3664.6	5168.7
3	10	1GB	3579.1	5113.1
4	1	1GB	3890.3	5237.7
5	1	2GB	4391.6	5355.3
6	1	5GB	5384.9	5835.6
7	1	10GB	5540.5	5754.2
8	1	20GB	5371.4	5904.5
9	1	30GB	5716.5	6333.5

Table 1: Throughput for variable file sizes

From the above table 1, we can observe that the throughput of the application is very low means 1GB file split into multiple files of equal size (first three experiments) and throughput is varies as the files of small size and throughput is good if we concatenate the same multiple files (40 files or 20 files or 10 files) and input to the application as a single file (1GB file). From the above analysis we observed that Throughput increase as the size of the file increases as shown in the below graph: Figure 2.

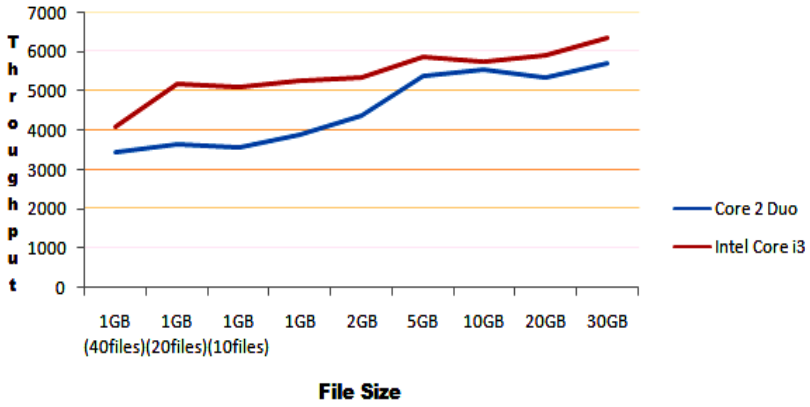

Figure 2: Variable File Size versus Throughput

We have further analyzed the rate processing of records defined as percentage of processing of records by total number of records processed by machines in a unit amount of time, means rate of processing of Intel Core i3 is calculated by percentage of total bytes of processed by Intel Core i3 in unit time divided by total records processed by Core 2 duo and Intel Core i3 as shown in table 2.

Sl.No	Files	Total File Size	Core 2 Duo	Intel Core i3
1	40	1GB	45.7	54.3
2	20	1GB	40.5	55.4
3	10	1GB	41.2	58.5
4	1	1GB	42.6	57.4
5	1	2GB	45.6	54.9
6	1	5GB	47.9	52.1
7	1	10GB	49.5	53.5
8	1	20GB	47.8	54.4
9	1	30GB	47.4	52.6

Table 2: Rate of Processing

We have conducted a experiment with Hadoop in core 2 duo and IntelCore i3 systems, to processing of records in same files of variable size and found that processing of records in Intel Core i3 is faster than Core 2 duo as shown in figure 3.

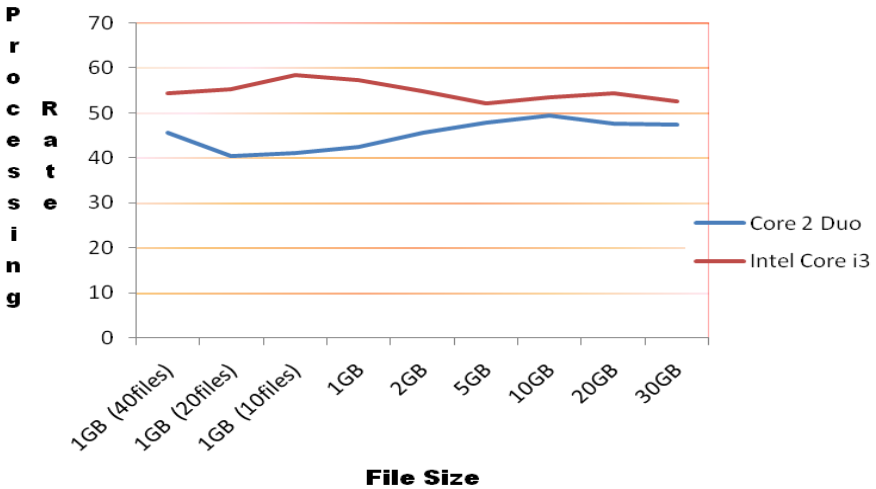

Figure 3: Rate of Processing of different machines

V. Conclusions

Data quality challenges can be resolved by deploying a well planned testing approach for both functional and non-functional requirements. Applying a right test strategies will improve the performance quality during processing, pre-processing and post processing. As we discussed in above section we can achieve good observable performance from huge single files than multiple files of variable size and necessary parameter configuration of mapper and reducer also helps in achieving a good performance of Hadoop application. Big Data testing is a specialized stream, tester should built with good data analysis skill set for identifying quality issues in data, it will help in identifying defects early and reduce the cost of implementation.

Acknowledgements

We indebted to management of MSRIT, Bangalore for excellent support in completing this work at right time, and also for providing the academic and research atmosphere at the institute. A special thanks to the authors mentioned in the references.

References

[1] Paul C. Zikopoulos, Chris Eaton, Dirk deRoos, George Lapis, Thomas Deutsch,Understanding Big Data McGraw-Hill Companies 2012.
[2] Lam,Chuck,Hadoop in Action, Manning Publications Co. 2010, Pages 21-25.
[3] Dryman,Carpediem, Hadoop Performance Tuning Best Practices, 241-242, ACM,2014.
[4] Zikopoulos, Paul and Eaton, Chris and others, Understanding big data: Analytics for enterprise class hadoop and streaming data, McGraw-Hill Osborne Media 2011.
[5] Vinaykumar Chandrashekar, VinyasShrinivasShetty, Testing in a Big DataWorld, Manning EuroSTAR 2013.
[6] Mahesh Gudipati, ShanthiRao, Naju D. Mohan and Naveen Kumar Gajja, Big Data: Testing Approach to Overcome Quality Challenges, Infosys, 2013.
[7] Almeida, Fernando, Calistru, Catalin, The main challenges and issues of big data management, April 2013.
[8] Xie, Jiong and Yin, Shu and Ruan, Xiaojun and Ding, Zhiyang and Tian,Yun and Majors, James and Manzanares, Adam and Qin, Xiao, Improving map reduce performance through data placement in heterogeneous hadoopclusters,IEEE, 2010.
[9] Michael Kopp, About the Performance of Map Reduce Jobs. http://apmblog.compuware.com/2012/01/25/about-the-performance-of-map-reduce-jobs/ 2012.
[10] Menasc, Daniel, Load testing of web sites Internet Computing, IEEE, vol=6, num=4, pages=70-74, IEEE, 2002.
[11] Shrinivas B Joshi, Apache hadoop performance tuning methodologies and best practices ICPE 12 Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering, Pages 241-242.
[12] White,Tom, Hadoop: The definitive guide O'Reilly Media, Inc., 2012.
[13] Todd Lipcon 7 Tips for Improving MapReduce Performance, Apache Hadoop for the Enterprise, Cloudera, http://www.cloudera.com/blog/2009/12/7-tips-for-improving-mapreduce- performance, 2009.
[14] Performance measurement of a Hadoop Cluster, https://www.amax.com/enterprise/pdfs/AMAX%20Emulex%20Hadoop%20Whitepaper.pdf, February 232012.
[15] What Are Performance Metrics Used in Load Testing [online], Available: http://loadstorm.com/load-testing-metrics/ 2015.