I.J. Information Technology and Computer Science, **2015, 11, 55-61**
Published Online October 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2015.11.07

Performance Analysis in Bigdata

Pankaj Deep Kaur
Guru nanak Dev University Reg Campus,Jalandhar/CSE, India
E-mail: pankajdeepkaur@gmail.com

Amneet Kaur and Sandeep Kaur
Guru Nanak Dev University Reg. Campus Jalandhar/CSE, India
E-mail: {arora.amneet@gmail.com, kaursandeep116@gmal.com}

***Abstract*—**Big data technologies like Hadoop, NoSQL, Messaging Queues etc. helps in BigData analytics, drive business growth and to take right decisions in time. These Big Data environments are very dynamic and complex; they require performance validation, root cause analysis, and tuning to ensure success. In this paper we talk about how we can analyse and test the performance of these systems. We present the important factors in a big data that are primary candidates for performance testing like data ingestion capacity and throughput, data processing capacity, simulation of expected usage, map reduce jobs and so on and suggest measures to improve performance of bigdata

***Index Terms*—**NOSQL, YCSB, HDFS.

I. Introduction

Big data (Petabytes or Exabyte) describes large amount of both structured and unstructured data that is difficult to process using traditional database and software techniques. The data is loosely structured and incomplete that is to be minded for information. It includes information gathered from social media, internet-based devices that is Smartphone and tablets, video and voice recordings, and logging of structured and unstructured data.

Big Data is characterized by 3Vs (Fig-1) as high volume, velocity and variety information assets that require cost-effective and innovative forms of information processing for decision making. The technologies associated with big data analytics include NoSQL databases, Hadoop and Map Reduce [10].

A. Importance of Bigdata

- Big data enables organizations to accomplish several objectives.
- It helps to support real-time decisions, anytime and anywhere.
- It includes various types of information that can be used in decision making.
- helps to explore and analyze information
- Improve business outcomes and manage risk, now and in the future

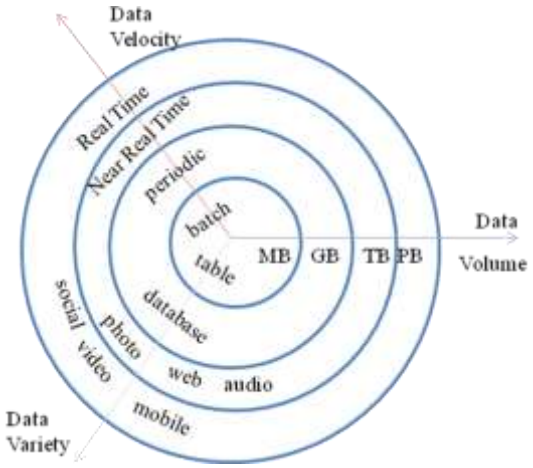

Fig.1. Diagrammatic view of bigdata

B. Bigdata Processing

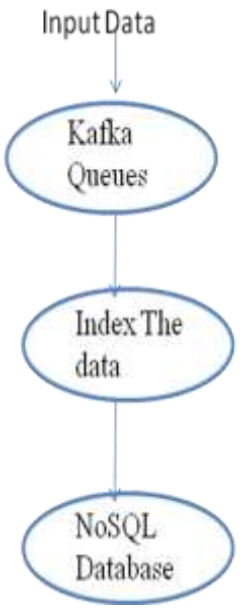

Fig.2. Bigdata Processing

Bigdata processing (Fig.2) is achieved through kafka queues. Multiple data from various sources goes through the queue and is moved to either a NoSQL data store or

HDFS. Depending on the data store we can write NoSQL queries or map reduce programs to extract the data and create reports for enabling business decisions. Performance measurement is important to support decision making and action taking in organizations. It should be as dynamic as possible to keep pace with changes that happen in organizations. Therefore, it must be aligned to the organizations strategies and should be reviewed periodically.

Bigdata is not only large amount of data but there are various other features which create big difference between large amount of data and massive data. There are various definations regarding bigdata about how bigdata is viewed-

- Attributive
- Comparative
- Architectural

1. Attributive

As report given by IDC in 2011, bigdata extract the value from large amount of data by capturing high velocity data which further led to change of definition in bigdta which includes velocity, volume, variety and value. Further META group analyst noted bigdata as only three dimensional, velocity, volume and variety

2. Comparative

Mckinsey's report in 2011 defined bigdata as large data which makes it difficult to capture, store and analyze the data. This definition did not define bigdata properly. However it gave evolutionary aspect regarding bigdata

3. Architectural

As suggested by national institute of standard and technology (NIST), large velocity, volume and variety of data coined as bigdata limits its ability to use relational database system due to such features and require new technological means and horizontal scaling for processing and storing data. Bigdata is further divided into two views as bigdata science and bigdata framework. Bigdata science covers techniques for evaluation of bigdata and bigdata framework covers algorithms and software libraries that help in distributed processing of bigdata across various clusters.

II. BIGDATA PERFORMANCE CHALLENGES

A. Volume

Performance challenges faced due to size of BigData.

- **Scalability**-The mean time between failures (MTBF) falls as the scale of data increases. The failure of a particular node in a cluster affects the calculation work that is required for processing bigdata that lead to processing delays.
- **Power Constraints**- leads to lowering of clock speed
- **Impact on Networking**- large amount of data pushed on a network takes longer time to finish a job and even can lead to failure of data node.
- **Impact on Cloud Services**- large amount of data lead to degradation in wan transfer speed over long distances.

B. Velocity

Performance challenges due to increased speed of flow of data

- **Access latencies:** speed with data is accessed in memory and access time for hard-disk, create performance problems.
- **Response time:** critical response-time results are observed.
- **Impact of security-** increase in velocity leads to security relevant data.

C. Variety

Large amount of structured, semi-structured and unstructured data lead to performance implications

- **Data types-** most crucial challenge arise out of variety of data types.
- **Tuning-** tuning the infrastructure of data storage in case large amount of data.
- **Veracity-** data collected from different sources leads to error and unreliable data.

Performance Evaluation Need

- **Meeting the Need for Speed:** Data can be gathered and analyzed easily to generate instant values.
- **Addressing Data Quality:** Data should be accurate and ensure data quality.
- **Displaying Meaningful Results-** improving performance will produce more accurate results.
- **Timely Data-** data should reach without wasting any time.
- **Business Decisions-** to support business decision and action taking in organization.
- **High response-time: t**o obtain timely response
- **Ensure security:** to ensure security of data
- **Avoid access latencies:** to avoid any time delays
- **To Avoid Errors:** To generate error free data

Performance Evaluation Factors

There are different performance factors (Fig-3)

- **Data Ingestion and Throughput**:Primarily, data ingestion is done through messaging queues like Kafka, Rabbit MQ, and Zero MQ and so queues must perform optimally to achieve maximum throughput.
- **Data Processing**: Data processing refers to the speed with which the queries are executed to

generate the output results which are further used for generating business reports and analysis.

- **Data Persistence**: Performance evaluations of different databases are conducted and one is chosen which suits are application requirements.
- **Data fetching**- analysis fetched from aggregated data. Quite often this is achieved via web interface and can be easily tested for performance.[1]

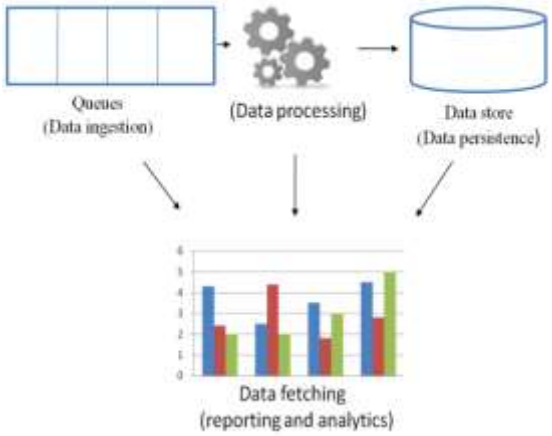

Fig.3. Performance Factors

Performance Evaluation Areas

- **Data Storage**: How data is stored across different nodes. What is the replication factor?
- Commit logs: setting the value for commit log before it grows.
- **Concurrency**: how many threads can perform read and write operations: concurrent reads and concurrent writes.
- **Caching**: caching uses large amount of memory so we should tune them properly.
- **Timeouts***:* Values for connection time out, query timeout etc.
- **JVM parameters**: GC collection algorithms, heap size etc.
- **Map Reduce performance**: Sorts, merge etc.
- **Message queue**: Message rate, size etc.

III. Performance Evaluaton

A. Big Data Benchmarks

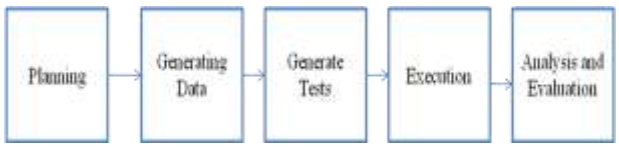

Fig.4. Bigdata Benchmark

Big data benchmarks (Fig.4) are developed to evaluate and test the performance of big data systems and architectures. Firstly, at planning stage, evaluation metrices are determined, and then data and test used in the evaluation are determined. Next, the Benchmark test is carried out at execution step. Finally results are obtained and Performance is evaluated.

Efficient benchmarking produce accurate and error free results. It promotes BigData technology and develops new innovative theories and algorithms. It tunes the system validate deployment strategies and carries other efforts to improve systems. For example, benchmarking results can identify the performance bottlenecks in big data systems, thus optimizing system configuration and resource allocation.

B. Performance Testing

1. Performance testing challenges

Organizations are facing challenges in defining the strategies for validating the performance of individual sub components, creating an appropriate test environment, working with NoSQL and other systems.

- **Diverse set of technologies**: Each component in a big data belongs to a different technology. So we need to test each component individually.
- **Unavailability of specific tools**: No single tool is available for each of the technology. For e.g. database testing tools for NoSQL might not fit for message queues.
- **Test scripting**: There are no record and playback mechanisms for such systems. Scripting is required to design test cases and scenarios.
- **Test environment**: It might not always be feasible to create a performance testing environment because of the cost and scale. So we need to have a scaled down version to predict performance of all the components
- **Monitoring solutions**: Since every segment has an alternate method for uncovering execution measurements constrained arrangements exists that can screen the whole environment for execution irregularities and identify issues.
- **Diagnostic solutions**: Custom solutions need to develop to further predict the performance areas.

2. Performance testing needs

- **Increasing need for live integration of information**: As data is obtained from multiple sources, so it becomes important to integrate the information which gives us clean and reliable data.
- **Instant Data Collection and Deployment:** Decisive Actions and predictive analytics have led enterprises to adopt instant data collection solutions. These decisions bring in significant business impact.
- **Real-time scalability needs:** Big Data Applications are designed such it matches the level of scalability and monumental data processing that is involved in a given scenario. Critical errors in the design of Big Data Applications can lead to critical situations. Hardcore testing involves better performance.

3. Performance testing approach

Any big data project involves in processing huge volumes of structured and unstructured data and is processed across multiple nodes to complete the job in less amount of time. At times because of poor design and architecture performance is degraded. Some of the areas where performance issues can occur are imbalance in input slits, redundant shuffle and sorts, moving most of the aggregation computations to reduce process and so on. Performance testing is conducted by setting up huge volume of data in an environment close to production. Utilities like Nagios, Zabbix, Hadoop, MangoDB, Casandra monitoring etc. can be used to capture performance metrics and identify the bottlenecks. Performance metrics like memory, throughput, job completion time etc. are critical for monitoring and analysis [2].

The process (Fig.5) starts with the setting up of the BigData cluster which is to be tested for performance. Depending on the typical usage of the components, we need to identify and create corresponding workloads [2]. As a next step, custom scripts are created Further, tests are executed to simulate realistic usage and results are identified. Based on the results, we can tune the cluster and re-execute the tests till the maximum performance is achieved.

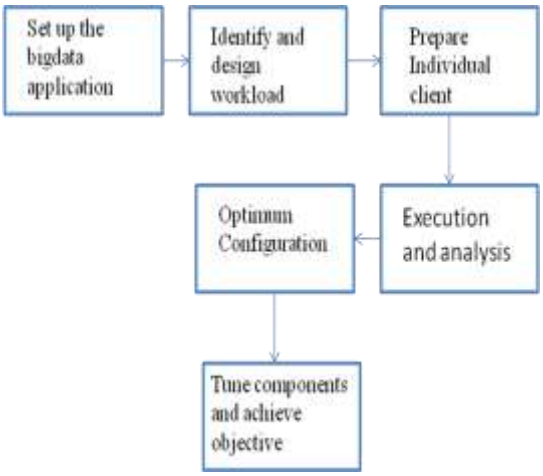

Fig.5. Steps for Performance Testing

4. Performance Testing Tools

- **YCSB**: YCSB is a yahoo cloud service benchmark testing tool that performs reads, writes and updates according to workloads and measure throughput in operation and record the latency.
- **Sandstorm**: Sandstorm is an automated performance testing tool that performs performance testing and provides a scripting interface to stress test any big data application.[2]
- **JMeter**: JMeter provides plugins which apply load to Cassandra. These plugins can send requests over Thrift. The plugin is fully configurable.
- **Independent Custom Utilities**: Cassandra stress test etc.
- **HPCC systems**: HPCC means 'high performance computing cluster' and provides 'higher performance'.
- **Apache Dril**l: It is part of the Apache Incubator and it offers a distributed system to perform analyses of large datasets that are based on Dremel

5. Performance Testing Techniques

Various testing techniques are required such as functional and non-functional in order to obtain good quality and error free data. [13]

Functional testing

It includes validating structured-unstructured data, validating map-reduces process and data storage validation.

Hadoop is used for distributed data processing of large sets of data across cluster of computer. It uses map/reduce technique in which data is divided into small fragments and further executed on any node in cluster.

Certain steps are executed in Hadoop (Fig. 6).

a. Load the data in HDFS
b. Execute map-reduce operation.
c. Obtain the output result from HDFS

Fig.6. Steps in HDFS.

a. Load data in HDFS

Data is obtained from various sources and input into the HDFS and divided into multiple nodes.

- Verify the data to avoid any data corruption.
- Check the validity of data which is to be loaded in HDFS.

b. Execute map-reduce operations

Firstly, data is processed using map-reduce operation and desired result is obtained.

- To ensure correctness of key-value pair generated we need to validate the map-reduce process.
- Aggregation of data is check after reduce operation is performed.
- Check and compare the output data with input data to ensure correctness of data.

c. Obtain the Output Results

In this step output results are obtained from HDFS and further send to database system for storage which act as repository for data to generate the reports and used for further processing.

Non-Functional Testing

Performance Testing- Increase the performance of bigdata applications by increasing response time, data capacity size and processing capacity and also system is tested under load to improve performance. Performance monitoring tool is used which identify issues and capture the performance.[13]

Failover Testing- Hadoop architecture consists of various nodes hosted on server machine, so there may be a case of node failure. Failover testing detects such failures and recovers data to proceed with processing. Certain validations are performed during failover testing such as checkpoints and recovery of edit logs. Recovery time objective and recovery Point Objective matrices are captured during this testing [13]

IV. Performance Improvement Strategies

- **Data quality**- It is must to obtain quality of data from different sources such as from social media, sensors etc.
- **Data Sampling**-Tester's must recognize suitable sampling technique that includes right test data set.
- **Automation**-automate the test suites. Bigdata test suites are use number of times as dta in database is updated periodically. This saves lot of time when validating bigdata
- **Database sharding**- Dividing the data and running in parallel.
- In case bigdata is carried on networks, switches and routers should be choosen with queuing and buffering strategies.
- In memory computing technique is used which enables faster computation and analysis.
- A new version of Hadoop which implements CRC32C by using hardware support to improve performance.
- Security mechanisms are applied such that it doesnot increase acess latencies.
- Constantly review the data changes and identify the ways in which data objects of different types are represented.

V. Related Work

The MapReduce model, developed by Dean and Ghemawat [4], introduced a programming model and the associated implementation for distributed processing of large volumes of unstructured data using commodity hardware. The MapReduce model implemented on Google's cluster by Dean and Ghemawat, had demonstrated good performance for sorting, and pattern searching (Grep) on unstructured data.

In 2009, Pavlo et al [8] discussed an approach to comparing MapReduce model to Parallel DBMS. As part of their experiments, they compared Hadoop, Vertica, and DBMS-X. The authors used benchmarks consisting of a collection of tasks that were running on the three platforms. For each task, they measured each system's performance for various degrees of parallelism on a cluster of 100 nodes. They used Grep, Aggregate, Join, and Selection tasks.

In 2011 Fadika *et al* [9] presented a performance evaluation study to compare MapReduce platforms under a wide range of use cases. They compared the performance of MapReduce, Apache Hadoop, Twister, and LEMO. The authors designed the performance design test under the following seven categories: data intensive, CPU intensive, memory intensive, load-balancing, iterative application, fault-tolerance and cluster heterogeneity.

In 2009, Dinh *et al* [2] conducted a performance study of Hadoop Distributed File system for reading and writing data. They used the standard benchmark program TestDFSIO.java that is available with the Hadoop distribution. Their study discussed the implementation, design, and analysis of reading and writing performance.

In 2013 Elif Dede, Bedri Sendir[3] evaluated the reading and writing performance of Cassandra and Hadoop under different scenarios, including the YCSB (yahoo cloud service benchmark) Benchmark's Workload C. various experiments were performed on the Gridand Cloud Computing Research Lab Cluster at BinghamtonUniversity.8 Nodes in a cluster, each of which has two 2.6Ghz IntelXeon CPUs, 8 GB of RAM, 8 cores, and run a 64-bitversion of Linux 2.6.15.

Manoj V in 2014 evaluated the performance of Cassandra in comparison with RDBMS and its read/write performance was calculated on basis of number of threads. Write performance of Cassandra is faster than RDBMS whereas its read performance is slower when tested under 1000 concurrent threads.

Abramova v in 2013 evaluated Cassandra and mangoDB. Execution time was tested according to data size and workload. With increase in data size and running different workloads Cassandra came out to be faster than MangoDB.

Abramovain 2014 tested Cassandra performance based on certain factors such as data size, number of nodes, number of threads and workload characteristics, and analyzed whether desirable speedup and scalability properties were met. Scaling the number of nodes and

datasets does not guarantee performance improvement. But Cassandra deals well with concurrent request threads and scales well with concurrent threads.

A. Gandini1 in 2013 evaluated the performance of various No-sql databases as H-Base, MangoDB and Cassandra on the basis of number of nodes, number of cores and replications. The final results were obtained in the form of throughput and latency.

VI. Conclusion

We conclude that organizations have to choose the best solutions according to their needs, to solve their performance testing challenges. It is desirable to get all the possible options like testing and monitoring available under the same hood, as it will help in reducing the complications that arise when dealing with multiple alternatives to achieve a common goal. When it comes to Big Data, we can use any of the above tools to run performance and stress test directly on the database to identify and resolve any problems.

Acknowledgment

We would like to express our sincere gratitude to our advisor Dr. Pankaj Deep Kaur for the continuous support, patience, motivation, enthusiasm, and immense knowledge. Her guidance helped us in all the time of research and writing of this term paper. We could not have imagined having a better advisor than her.

References

[1] http://www.cmgindia.org/wpcontent/uploads/2014/01/PerformanceTestingofNoSQLApplications.pdf.

[2] http://www.qaiglobalservices.com/conference/stc2013/PDFs/Mustufa_Batterywala.pdf.

[3] Chaudhary U and Singh H Mapreduce performance evaluation through benchmarking and stress testing on multi-node Hadoop cluster. *International Journal of Computational Engineering Research (IJCER)* 4:2250-3005, 2014.

[4] Dean J and Ghemawat S MapReduce simplified data processing on large clusters. *Communications of the ACM* 51:107-113, 2008.

[5] Dede E, Sendir B, Kuzlu P, Hartog J and Govindaraju M (2013) An Evaluation of Cassandra for Hadoop. In *Cloud Computing (CLOUD) 2013 IEEE Sixth International Conference* (pp. 494-501). IEEE.

[6] Dokeroglu T, Ozal S, Bayir M A, Cinar M S and Cosar A Improving the performance of Hadoop Hive by sharing scan and computation tasks. *Journal of Cloud Computing* 3:1-11, 2014.

[7] Jewell D, Barros R D, Diederichs S, Duijvestijn L M, Hammersley M, Hazra A, and Zolotow C *Performance and Capacity Implications for Big Data*. IBM Redbooks, 2014.

[8] Pavlo A, Paulson E, Rasin A and Abadi D J, DeWitt D J, Madden S, and Stonebraker M A comparison of approaches to large-scale data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data* (pp 165-178) *ACM, 2009.*

[9] Fadika Z, Dede E, Govindaraju M, & Ramakrishnan L Benchmarking mapreduce implementations for application usage scenarios. *In Grid Computing (GRID) 2011 12th IEEE ACM International Conference on (pp 90-97) IEEE, 2011, September*.

[10] http://searchcloudcomputing.techtarget.com/definition/big-data-Big-Data.

[11] http://www.cigniti.com/big-data-testing.

[12] http://www.bigdata-startups.com/open-source-tools/data-analysis-platforms.

[13] Nagdive A S, Tugnayat R M & Tembhurkar M P. Overview on Performance Testing Approach in Big Data. *International Journal of Advanced Research in Computer Science*, *5*(8).

[14] Gudipadi M, Rao S, Mohan D N &Gajja N K. Bigdata: Testing approach to overcome quality challenges in Infosys lab Briefings 11(1).

[15] Abramova V, and Bernardino J "NoSQL databases: MongoDB vs cassandra" In Proceedings of the International C* Conference on Computer Science and Software Engineering pp. 14-22 ACM.

[16] http://en.wikipedia.org/wiki/big_data.

[17] Manoj V "Comparative Study of NoSQL Document, Column Store Databases and Evaluation of Cassandra" 2014 International Journal of Database Management Systems (IJDMS) Vol, 6.

[18] Abramova V, Bernardino J and Furtado P "Testing Cloud Benchmark Scalability with Cassandra" In Services (SERVICES), 2014 IEEE World Congress on pp. 434-441 IEEE.

[19] Gandini A, Gribaudo M, Knottenbelt W. J, Osman R and Piazzolla P, 'Performance evaluation of NoSQL databases" In Computer Performance Engineering, pp 16-29.

Authors' Profiles

Dr. Pankaj Deep Kaur is working as an Assistant Professor in the Department of Computer Science and Engineering, Guru Nanak Dev University, RC, Jalandhar, India. She received her Bachelor's Degree in Computer Applications (2000) and Master's Degree in Information Technology (2003) from Guru Nanak Dev University, Amritsar, India. She completed her Ph.D. in Resource Scheduling in Cloud Computing from Thapar University, Patiala (2014) and has over ten years of teaching and research experience. She has been a university position holder in her graduation studies and received Gold Medal for her excellent performance in her Post Graduation studies. She is a recipient of Junior Research Fellowship from Ministry of Human Resource and Development, Govt. of India. Her research interests include Cloud Computing and Big data.

Amneet Kaur is currently pursuing her post graduation from Guru Nanak Dev University Reg Campus Jalandhar. She received her Bachelor's Degree in computer science (2014) from Guru Nanak Dev Engineering College, Ludhiana.

Sandeep Kaur is currently pursuing her post graduation from Guru nanak Dev University Reg Campus Jalandhar. She received her Bachelor's Degree in Information Technology (2014) from Baba Banda Singh Bahadur Engineering College, Fategarh Sahib.

How to cite this paper: Pankaj Deep Kaur, Amneet Kaur, Sandeep Kaur,"Performance Analysis in Bigdata", International Journal of Information Technology and Computer Science(IJITCS), vol.7, no.11, pp.55-61, 2015. DOI: 10.5815/ijitcs.2015.11.07

