A TESTING APPROACH TOWARDS BIG DATA

13th Annual International software testing conference, India
## Contents

Abstract ........................................................................................................................................................................ 2

1. Introduction .................................................................................................................................................................. 3
   1.1 Big data and its needs .......................................................................................................................................... 4
   1.2 What is big data testing ....................................................................................................................................... 4

2. Big data testing challenges ........................................................................................................................................ 5
   2.1 Test Data ............................................................................................................................................................ 5
   2.2 Environment ....................................................................................................................................................... 5
   2.3 Performance ....................................................................................................................................................... 5

3. Test Data for Big Data Testing ................................................................................................................................ 6
   3.1 Test Data - Requirement & Importance of its Quality .......................................................................................... 6
   3.2 Test Data Generation .......................................................................................................................................... 6
   3.3 Data storage ........................................................................................................................................................ 6

4. Test environment setup and its significance ............................................................................................................. 7
   4.1 Test environment needs ........................................................................................................................................ 7
   4.2 Importance of monitoring of test environment .................................................................................................. 7

5. Performance testing in Big Data ............................................................................................................................... 8
   5.1 Performance bottlenecks of Big Data ................................................................................................................ 8
      5.1.1 Linux tuning for best performance ................................................................................................................. 8
      5.1.2 Hadoop tuning for best performance ........................................................................................................... 8
      5.1.2.1 Base configuration (in mapred-site.xml file) .......................................................................................... 9
      5.1.2.2 Compression (in mapred-site.xml) .......................................................................................................... 9
      5.1.2.3 JVM re-use policy (In mapred-site.xml) .................................................................................................. 10
      5.1.2.4 HDFS block size (in hdfs-site.xml) .......................................................................................................... 10
      5.1.2.5 Map-side spills (in mapred-site.xml) .................................................................................................. 10
      5.1.2.6 Reduce side spills (in mapred-site.xml) ................................................................................................. 11
   5.2 Benchmarking of Hadoop .................................................................................................................................. 11

6. Conclusion ............................................................................................................................................................... 11
Abstract

Quality is a most important aspect in any software product, and data plays a significant role in defining quality of a software product. Growing number of users and information available across the different industry and platform creates a new opportunity for the traditional application. They start moving towards NoSQL database technologies to handle data of large size and diverse formats. But what, if data becomes a challenge.

Challenge in terms of data storage, data processing and its maintenance. These three are the challenges but at the same time it a requirement of an enterprise which hold a huge cost. Exponential growth of enterprise data demands a new approach to test software products where data is collected from single/multiple sources for processing.

Traditionally, most of the software testing processes focuses on- environment, some set of pre-defined steps and expected results. But in big data, it needs to be re-defined. Here we have two major player, data and environment. Also Performance is the major bottleneck in big data. This paper highlights the strategic areas of functional testing and explores new business perspectives for testing services in big data domain. It also uncovers different fundamental component involved in any big data application and there testing needs. Along with architectural performance challenges, it also covers recommended enhancements based on actual parameters & data.
1. Introduction

Big Data is the next big tradeoff in the technology universe. We can describe Big Data as something which is growing exponentially with time, and carry raw but very valuable information inside that can change the future of any enterprise. It is a collection, which represents large dataset, may be collected from multiple sources or stored in an organization.

Earlier, we were only dealing with well-structured data, hosted in large data warehouses and investing a cost in maintaining those data warehouse and hiring expert processional to maintain and secure information hosted in that data warehouse. Data was structured and can be queried anything as per the needs. But now, this exponential growth of data generates a new vision for data science along with some major challenges.

It refers to our ability to collect or analyze huge amount of data in a minute or two, and extract valuable information out of it to add value to our existing business or services to customers. Let’s understand a real time example, Major companies like Wal-Mart are leveraging the benefit of big data by collecting data from customer’s buying patterns at their stores, their internal stock information, and their inventory demand supply relations and analyze all, in seconds even in real time to add value to its customer experience.

So, extracting information from large dataset somewhat calls a concept of Data mining which is an analytic process originally designed to explore large datasets. Ultimate goal of data mining is to search for consistency in a pattern or systematic relationship between variables, which helps in predicting the next pattern or behavior.

Now, if we take concepts of data mining forward along with large data set, at some extent it become a blocker for our existing approach, because big data may contain structured or unstructured data even it may contain data in multiple formats also.

Hence storing such a huge data without any defined structure is next impossible with our traditional approach of data warehouse or data marts. And here come a concept called NoSQL, which is “Not only SQL”. Technologies evolved around this concept and now we have number of framework to work and to deal with big data.
1.1 Big data and its needs

If we analyze the growth of data in last 10 years, then we'll find a huge tradeoff in the growth curve. Exponential growth is identified in the growth of data. Volume of data increases with tremendous velocity, and in Varity of formats. High-volume, high-velocity and high-variety information can become assets for any organization if maintained accurately. The cost of errors in any big data application can vary from nothing to large amounts of money. Hence there arises a need for quality Big Data solution, which gives confidence to take key decisions on the basis of analytic reports. As data is the key central ingredient in big data, hence generating and maintaining data is very important.

Growth of data is not a real issue; major concern is what to do with big data? What useful information can be captured? At the time, storing those large data sets cost like anything. Organization need to invest a lot to store and manage their large dataset generated by their ERP systems or any application which they are using or running in the market. If we take an example of organizations like Facebook or twitter, they are continuously generating millions of records every day, which need terabytes of memory to store. But the question is- why it needs to store or invest in storing & maintaining such a large data. Even after storing the data in the commodity cluster of hardware, it demands for maintenance which is an addition to the cost of storing the data.

Now, here comes an opportunity to process such a huge data and get something valuable out of it. But processing of large dataset is a challenge if we keep using traditional approach or technologies. Processing of big data needs a framework which is based on the concept of ‘NoSQL’ i.e. Not only SQL. Data captured from different sources may be in a defined structure or may be unstructured. But there processing is as important as your valuable assets. Unstructured data also carries information which can be extracted for key decisions of an organization. So this explains the need to big data and its value. Data has its power and it just need to be utilized in a right dimension.

1.2 What is big data testing

Testing is an art of achieving quality level in your software product, in terms of perfection of functionality, performance, user experience or usability. But when it comes to big data testing, you need to keep your focus more on functional and performance aspect of an application. Performance is the key parameter in any big data application which is meant to process terabytes of data. Successful processing of terabytes of data using commodity cluster with number of other supportive components needs to be verified. Processing should be faster and accurate which demands high level of testing.

Processing may be of three types – Batch, real time or interactive. And based on which, we need to integrate different component along with NoSQL data store as per the needs. For example, in case of batch processing, Hadoop can be used along with hive or hbase which works as a wrapper on Hadoop to make it easy to process any amount of data on it. Testing of Hadoop along with its integration with other components (like- hive, hbase etc.) is required to ensure the accurate and optimized environment setup for big data application. Now after environment, application deployment needs to be verified. Deployment includes compatibility of application with the cluster created and tested. Scalability is primary demand in any big data application. And increasing the size of cluster should not affect working work application of cluster.
2. **Big data testing challenges**

Functional testing is the major focus in big data application. Any application with Hadoop as file system & some other supportive components like hive or zookeeper or flume, majorly demands for perfection of functionality. Testing approach to be adopted, should be competent enough to cover all the gaps between application requirement and its implementation. The origin of those gaps are nothing but the challenges that are associated with its final outcome. In big data testing, certain challenges are involved which needs to be addressed by the big data testing approach.

2.1 **Test Data**

Exponential growth had been observed in the growth of data from last couple of years. Huge amount of data are being generated daily and stored in large data centers or data marts. So, it start demanding for its efficient storage and a way to process it in an optimized way. If we consider telecom industry, it generates large number of call logs daily and they need to be processed for better customer experience and compete in the market. Same goes with the test data, test data should be similar to production data and should contain all the logically acceptable fields in it.

This becomes a challenge for testing big data application, generating test data similar to production data is a real challenge. For a moment, it can be achieved using any small data generation utility or proprietary code can be used but at the same time, it should be logically correct, which give raise to another challenge. Logical correctness of data is a real challenge and again it needs a tool to do so. Test data should also be large enough of verify proper working big data application.

2.2 **Environment**

Processing of data highly depends on environment and its performance. Optimized environment setup give high performance and fast data processing results. Distributed computing is used for processing of big data which has data hosted in a distributed environment. So, when it comes to testing environment, it need similar environment, where test engineer can perform or execute different test scenarios. It should have multiple number of node and data should be distributed over the nodes. At the same time, it also need to monitor those node, to ensure highest performance with minimum CPU and memory utilization. Nodes should monitored and there should have graphical presentation of node performance. So, test environment has two aspects – distributed nodes and their monitoring, which should be covered in testing approach.

2.3 **Performance**

Performance is the key major requirement of any big data application, and of course because of which enterprises are moving towards NoSQL technologies, technologies which can handle their big data and process in minimum time frame. Large dataset should be processed in minimum considerable time frame. In big data testing, performance testing is a challenges, it require monitoring of cluster nodes during execution and also time taken for every iteration of execution.
3. Test Data for Big Data Testing

Data plays a vital role in testing of big data application. Application is meant to process data and provide expected output based on implemented logic. The logic needs to be verified before moving to production, as the implementation of logic is completely based on business requirement and data.

3.1 Test Data- Requirement & Importance of its Quality

Good quality test data is as important as the test environment. Applications are developed over the data and some business logics that needs to be applied to get valuable information out. So, we have two important aspects, data and application logic. Data and its quality drive the whole application functionality. It define the quality of application and ensure handling of raw or invalid data that might be pushed for processing.

In big data world, data can have any format or size, it may be in the form of document, xml, JSON, PDF etc., at the same time data size may goes up to terabytes of petabytes. Hence, test data should also have multiple format and size should be large enough to ensure handling of large data processing. In big data testing, it needs data with logical values as per the application requirement and format which is supported by the application.

Along with it, data quality is another aspect of big data testing. Ensuring quality of data before processing through application ensures the accuracy of final output. Data quality testing itself is a huge domain and cover lot of best practices which includes – data completeness, conformity, accuracy, validity, duplication and consistency etc. It should be included in the big data testing and this ensure the level of accuracy application is supposed to provide.

3.2 Test Data Generation

Data is important and hold a lot of value which help enterprise to take key business decisions. Generation of test data is again a challenging job, there are multiple parameter, which has to be taken care while generating test data. It needs a tool, which can help to generate data and should have functions or logic can also be applied over it. Keeping the requirement under consideration and feasibility, Talend (an open studio) is a best candidate to fulfill the requirements of data generation. It is an open studio which provides number of function to control output and option to create schema based on business requiremeants. It can be used to generate any amount of data and that data can further be loaded on the Hadoop or any other file system used in application.

3.3 Data Storage

After generation of test data along with quality, it needs to host on a file system. For testing of big data application, data should be stored in the system similar to production environment. As we are working in big data space, there should have different number of node and data must be in the distributed environment. As mentioned earlier, talend open studio is a good candidate for generating data, it can also help to load those generated file on Hadoop or any other file system. Talend provides component for loading data on Hadoop file system directly from local directories on which it is running. During the data load process, it need high bandwidth connectivity to make it fast, but if in case of low bandwidth it allow you to create an archive of the same and then load on Hadoop.

In we consider an example of Hadoop, then we have Datanodes and a single Namenode which is taking care of all the operation performed on the cluster. It just need to put data on the HDFS and it automatically distribute data on HDFS and maintain replica of the same.
So, as explained, data is a crucial part of big data testing approach. Quality of data is as important as its generation. Data storage is also an important aspect. After completion of these parts/stage, next important stage will be test environment, which is challenges for big data domain.

4. Test environment setup and its significance

After generation of test data and achieving its quality, it become ready for processing through application. Application is developed on certain technology, like in case of Hadoop, big data application there should have map reduce jobs written to process data. In Big data testing, test environment should be efficient enough to process large amount of data as done in case of production environment. Real time production environment cluster generally have 30-40 node of cluster and data is distributed on the cluster nodes. There must have some minimum configuration for each node used in the cluster. Cluster may have two modes, in-premise or cloud. For testing in big data, it need the same kind of environment with some minimum configuration of node.

For setting up local cluster i.e. in-premise cluster, all the node are in the local area network and connected via LAN. And in cases of cloud, cluster can be deployed over the cloud machine and used till is being used by the application. After which cloud node can released and destroyed.

Also, scalability is also desired to be there in the test environment of big data testing, it help to study performance of application with increase of number of resources. That data can be used to define SLA (service level agreement) for that particular application.

Our Big data testing approach, highlight another option to setup test cluster. Virtual machines can used for setting up test environment. Virtual managers provide excellent feature of manage number of virtual machine in a single base machine. User can configure RAM or HDD required, and at the same time, the allocation can also be dynamic depending on needs. It give us facility to modify number of processor need to get maximum performance. Some automated tools might also be used to setup cluster environment and that should also scalability feature, i.e. user should be able to add new node any time in the existing cluster.

4.1 Test environment needs

Testing environment needs raised from the application requirement, application should have certain requirement on which it is supposed to be run. And for testing that application, it need similar environment, test environment should have cluster with distributed nodes and data. It should have enough space to store and process large amount of data. During processing of data, it increases load on the machine and utilize CPU and memory. In big data testing, it should be taken care that CPU utilization should be minimum, and hence in case of virtual machine we can configure it to optimize the processing. This gives a flexibility to maintain and manage test environment as per the needs of application or test scenario.

4.2 Importance of monitoring of test environment

Monitoring of node used in test environment, is also important to keep the performance high without compromising with the CPU and memory utilization or we can say without putting extra load on machines. Otherwise, we may have to face other disasters of over utilization, nodes may got crashed or goes down during processing of data which affects the end results.
Ganglia is an good option to be used for monitoring of node of a cluster, this give graphical view of CPU and memory used during execution of test case. The only which might be faced is the installation of ganglia on each nodes which can also be automated used one open source tool available. Other tools can also be used for the same like Monit. With the help of monitoring of nodes, user can optimize cluster for best possible performance.

In big data testing, test environment should be monitored continuously for every iteration of execution, whose results can be used to optimize Linux or Hadoop performance.

5. Performance testing in Big Data

Big data application are meant to process large amount of data, and it is expected that it should take minimum time to process maximum data. Along with it, application jobs should consume considerable amount of memory and CPU. In big data testing, performance parameter plays an important role and helps to define SLA’s. It covers performance of base machine and cluster. Also, for example, In case of Hadoop, map reduce jobs should be written with proper coding guidelines, so to perform better in production environment. Profiling can also be done on map reduce jobs before integration, to ensure their optimized execution. Similarly, in case of real time analysis, if application is developed on storm architecture, then spout and bolt should be written in a way to give maximum performance.

5.1 Performance bottlenecks of Big Data

Collective performance of big data application is associated with performance of base machine, cluster and the application. If we address the performance bottlenecks of big data, i.e. configuration and efficiency of computing nodes and cluster running on it. Application code also contribute in the performance of data processing. So to overcome those bottleneck, big data testing approach recommend certain tuning options to optimize and raise end to end performance results. Performance monitoring is done at test environment, along with which we can tune clusters and base machine for optimizing the performance.

5.1.1 Linux tuning for best performance

Linux system can be optimized for best performance, we can change some parameter in Linux which help to achieve maximum performance, like-

- Maximize number of open file descriptors (FD) ulimit – 32768 or unlimited
- To prevent fetch failure net.core.somaxconn Linux kernel parameter for high rate of incoming connection requests (default is 128 and we can configure it to 1024 for test cluster)

5.1.2 Hadoop tuning for best performance

If take an example of Hadoop file system, Hadoop can have different configuration parameter which is offered by Hadoop and those can used for optimizing the performance of test cluster. We have mentioned parameter list with its default value and description. Also recommended values are mentioned which can be used for test environment for big data testing.
### 5.1.2.1 Base configuration (in mapred-site.xml file)

<table>
<thead>
<tr>
<th>Hadoop Parameter</th>
<th>Default</th>
<th>Description</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>mapred.map.tasks</td>
<td>2</td>
<td>The default number of map tasks per job.</td>
<td>4</td>
</tr>
<tr>
<td>mapred.tasktracker.map.tasks.maximum</td>
<td>2</td>
<td>The maximum number of map tasks that will be run simultaneously by a task tracker.</td>
<td>4</td>
</tr>
<tr>
<td>mapred.reduce.tasks</td>
<td>1</td>
<td>The default number of reduce tasks per job.</td>
<td>4</td>
</tr>
<tr>
<td>mapred.tasktracker.reduce.tasks.maximum</td>
<td>2</td>
<td>The maximum number of reduce tasks that will be run simultaneously by a task tracker.</td>
<td>4</td>
</tr>
<tr>
<td>mapred.map.child.java.opts</td>
<td>-</td>
<td>Java heap sizes for Map</td>
<td>1024</td>
</tr>
<tr>
<td>mapred.reduce.child.java.opts</td>
<td>-</td>
<td>Java heap sizes for Reduce</td>
<td>1024</td>
</tr>
</tbody>
</table>

### 5.1.2.2 Compression (in mapred-site.xml)

<table>
<thead>
<tr>
<th>Hadoop Parameter</th>
<th>Default</th>
<th>Description</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>mapred.map.output.compression.codec</td>
<td>org.apache.hadoop.io.compression.DefaultCodec</td>
<td>If the map outputs are compressed</td>
<td></td>
</tr>
<tr>
<td>mapred.output.compress</td>
<td>False</td>
<td>Should the job outputs be compressed?</td>
<td>true</td>
</tr>
<tr>
<td>mapred.output.compression.type</td>
<td>RECORD</td>
<td>If the job outputs are to be compressed as SequenceFiles, how should they be compressed? Should be one of NONE, RECORD or BLOCK.</td>
<td></td>
</tr>
<tr>
<td>mapred.output.compression.codec</td>
<td>org.apache.hadoop.io.compression.DefaultCodec</td>
<td>If the job outputs are compressed, how should they be compressed?</td>
<td></td>
</tr>
</tbody>
</table>
5.1.2.3 JVM re-use policy (In mapred-site.xml)

Enabling JVM reuse policy reduces the overhead of JVM startup and teardown. It also improves performance since the JVM spends less time interpreting Java bytecode.

<table>
<thead>
<tr>
<th>Hadoop Parameter</th>
<th>Default</th>
<th>Description</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>mapred.job.reuse.jvm.num.tasks</td>
<td>1</td>
<td>How many tasks to run per jvm.</td>
<td>-1 (unlimited)</td>
</tr>
</tbody>
</table>

Setting this value to -1 indicates that an unlimited number of tasks can be scheduled on a particular JVM instance.

5.1.2.4 HDFS block size (in hdfs-site.xml)

If the Hadoop job is spawning a large number of Map tasks experiment with larger HDFS block sizes. Reducing the number of Map tasks this way can decrease the overhead of starting up and tearing down of Map JVMs. It can also reduce the cost involved in merging map output segments during the Reduce phase. Larger block sizes also help increase the execution time taken by each Map task. It is better to run small number of longer running Map tasks rather than large number of very short running Map tasks.

<table>
<thead>
<tr>
<th>Hadoop Parameter</th>
<th>Default</th>
<th>Description</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>dfs.block.size</td>
<td>134217728</td>
<td>The default block size for new files, in bytes.</td>
<td>268435456 (256 Mb)</td>
</tr>
</tbody>
</table>

5.1.2.5 Map-side spills (in mapred-site.xml)

While Map tasks are running the generated intermediate output is stored into a buffer. This buffer is a chunk of reserved memory that is part of Map JVM heap space.

<table>
<thead>
<tr>
<th>Hadoop Parameter</th>
<th>Default</th>
<th>Description</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>io.sort.mb</td>
<td>100</td>
<td>The total amount of buffer memory to use while sorting files, in megabytes</td>
<td>316</td>
</tr>
<tr>
<td>io.sort.record.percent</td>
<td>0.05</td>
<td>The percentage of io.sort.mb dedicated to tracking record boundaries.</td>
<td>0.162 (16 % of 316 Mb)</td>
</tr>
<tr>
<td>io.sort.spill.percent</td>
<td>0.80</td>
<td>The soft limit in either the buffer or record collection buffers. Once reached, a thread will begin to spill the contents to disk in the background.</td>
<td>0.99</td>
</tr>
</tbody>
</table>
5.1.2.6 Reduce side spills (in mapred-site.xml)

<table>
<thead>
<tr>
<th>Hadoop Parameter</th>
<th>Default</th>
<th>Description</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>mapred.job.shuffle.input.buffer.percent</td>
<td>0.70</td>
<td>The percentage of memory to be allocated from the maximum heap size to storing map outputs during the shuffle.</td>
<td>0.90</td>
</tr>
<tr>
<td>mapred.job.reduce.input.buffer.percent</td>
<td>0.0</td>
<td>The percentage of memory relative to the maximum heap size to retain map outputs during reduce phase.</td>
<td>0.8</td>
</tr>
</tbody>
</table>

5.2 Benchmarking of Hadoop

Hadoop comes with some benchmarking too, like TESTDFSIO, filebench, mrbench, nnbench, testrpc, loadgen, etc. These benchmarking can be used to check the performance of Hadoop cluster and optimized accordingly.

6. Conclusion

Looking back, we have started with a problem called big data and its challenges, keeping the challenges in mind we have defined approach of testing of big data applications. The approach started with the roles of test data in big data testing and its different aspects like data quality and storage. After which we move to test environment where monitoring and performance is the major bottleneck. The testing approach covers and address different options to overcome those bottleneck. Performance testing is another part of big data testing, and is the key non-function aspect any big data application. Our testing approach recommends performance enhancement at Linux and cluster level, and we have highlighted certain parameter and their value which can be modified to get maximum output in case of Hadoop file system. The parameters are highlighted along with their default values and a little description. Even after performance tuning, it need to benchmark the performance, for which we have highlighted certain benchmarking options that are available with Hadoop distribution only. Concluding the paper and highlighting the benefits, this approach of big data testing will make it easy for test engineer to verify and certify the business requirement implementations and for stack holders, it saves a huge amount of cost, which has to be invested to get the expected business returns.