Big Data Testing - Challenges, Processes And Best Practices

Brijesh Gopalakrishna, Consultant
Ashish Mishra, Consultant

Deloitte Consulting India Pvt Ltd.
Abstract

Big Data Testing is a trending topic in Software Industry nowadays. We have access to large amount of data also referred to as Big Data. At the click of a button we generate megabytes of data. Effectively managing, maintaining and using this data for Testing is a challenge.

Many organizations want to derive value from this data, the first step for which is testing this data. Testing for these complex datasets would be a challenge in the near future.

With this paper we would study about what is Big Data, its characteristics, its importance, processes involved, the tools which can be used Big Data Testing and the best practices.
Big Data - What is it?

A collection of data sets which are large and complex and are difficult to process; and does not fit well into tables and that generally responds poorly to manipulation by Structured Query Language (SQL) are known as Big Data. The most important feature of Big Data is its structure, with different classes of Big Data having very different structures.

Due to enormous volumes of data of various types that are coming in, the demand for more insights from more data at reasonable cost is growing really fast. In 2012, 2.8 Zettabytes (ZB) of data was created, of which only 0.5% of that data was used for analysis. The volume of data is expected to grow to 40ZB by 2020.

Big Data Overview

Suppose we have a 100 MB document which is difficult to send, or a 100 MB image which is difficult to view, or a 100 TB video which is difficult to edit. In any of these instances, we have a Big Data problem. Or suppose company ‘A’ is able to process a video of 300TB while company ‘B’ cannot. We would say that company ‘B’ has a Big Data problem. Thus, as you can see, Big Data can be system-specific or organization-specific.

Big Data is not only about the size of the data. It is related to the velocity and variety as well. They are known as the 4 V’s of Big Data – Volume, Velocity, Variety and Veracity. In addition, another V is often added to the Big Data dimensions; Veracity.

Velocity and Volume focus on the speed and amount of data. Variety and Veracity refers to the category and trustworthiness of data, respectively

- **Volume** (how much): Capturing and processing large quantities of data. Large amount of data from billions of smartphones, sensors, customer survey results, market trend results etc. As a result of this there is tremendous increase in volume of data generated.

- **Velocity** (changing frequency and real time processing): Processing data in real time. This involves how quickly the data is generated and stored. In many businesses this data is stored real time. For example, we always see recommendations on a shopping site. These are captured by storing the clickstream history of people who access the application.

- **Variety** (category of data): How many different types of data can be processed, e.g. email, video, audio, text, log files, and various transaction data. Data from various sources includes raw, structured and semi-structured data. We have different kinds of data coming from web pages, email, error logs etc. Type of Data has evolved significantly over the years.

- **Veracity** (truthfulness and confidence): How much you can trust the data for its legitimacy when it is pouring in from various sources.
Layers of Big Data

To gain actionable insight from data, it has to pass through different stages. There are four layers in a Big Data platform which everyone must be aware of.

Data Sources Layer: This is the first stage where the organization accumulates its data from various sources, i.e. social networking sites, emails, transaction records, and data residing in the existing database. It is best to perform a detailed assessment of the problem you are going to address, how it helps the business, and measure it against the data you currently have. You may need to go to new sources of data.

Data Storage Layer: This is where data gets stored once it is collected from various sources. As computing capacity and storage capacity has increased over the decades, data storage has attained prime importance in Big Data. While considering a file system for storing data, keep in mind that it should be easily accessible, free from cyber threats, and easily implementable and manageable. Google had come up with such a file system — GFS (Google File System) — over a decade ago and they had not made it as open source. Later, Yahoo did a lot of work in this area and came up with Hadoop Distributed File System [HDFS] and made it as open source under Apache Software Foundation. HDFS has the capability to run on commodity hardware and handles large scale data with the help of MapReduce (a component of Hadoop which helps in data processing).

Data processing or Analysis Layer: This phase will analyze the data collected in the above phase to derive the insights from it. MapReduce is the common tool used in this analysis. It is a programming model and an associated implementation for processing and generating large data sets with parallel, distributed algorithm on a cluster. The analytic phase will result in trends and patterns of a particular business.

Data Output Layer: In this phase, insights gained from the analysis phase will be transferred to those who are supposed to act on it. These outputs can be in the form of charts, figures, and key recommendations. Clear and concise communication is a must here as the decision makers may not have a thorough knowledge in statistics.
Big Data - Investment by the industry

Whether Big Data is perceived as a problem or an opportunity depends on how you or your organization approaches it. In the beginning, it is certainly a problem for all as we do not have the right infrastructure and resources to process it. Also, we are unsure about the insights and value which will be derived from it. Once the data is processed with the right tools, it can lead to the breakthrough insights which will help in strategic decision making and thereby the growth of the organization. In a survey conducted by supply chain LLC, 76% of customers considered Big Data an opportunity rather than a problem. A recent CNBC quote supports this notion; “Data is the new oil - in its raw form oil has little value; once processed it helps power the world”. Meanwhile, a study conducted by IBM in 2012 found that nearly half of the respondents considered customer satisfaction as the top priority. They think Big Data as a key to understanding the customer and predicting their behavior.

Here are two common use cases that are driving awareness and investment across industries:

1. Demand-driven, customer relationship-oriented initiatives. Business cases driving demand-driven initiatives center on increasing customer service levels, better identifying and acting on upsell and cross-sell opportunities, and reducing customer churn. Early adopters typically include telcos, retailers, banks, insurance firms, and citizen-oriented eGovernment initiatives.

2. Supply-driven, efficiency-oriented initiatives. The amount of data is increasing exponentially, but organizations’ ability to access and interpret it is not. Examples include smart grid technology and the expanded use of sensors to create more intelligent systems. The industries most affected include oil and gas, utilities, healthcare, and financial services and are moving to improve their efficiency and effectiveness i.e. driving increased interest in big data.

Big Data Testing

For big data testing strategy to be effective, the “4Vs” of big data — volume (scale of data), variety (different forms of data), velocity (analysis of streaming data in microseconds) and veracity (certainty of data) — must be continuously monitored and validated.
Before moving on to how testing is performed in Big Data systems, let’s take a look at the basic aspects of Big Data processing on the basis of which further testing procedure can be determined.

Aspects of Big Data testing:

1. Validation of structured and unstructured data
2. Optimal test environment
3. Dealing with non-relational databases
4. Performing non-functional testing

Failure in the above-mentioned things may result in the production of poor quality of data, delays in testing and increased cost of testing.

Big Data Testing can be performed in two ways i.e. functional and nonfunctional testing. A very strong test data and test environment management are required to ensure error-free processing of data.

**A. Functional Testing**

Functional Testing is performed in three stages namely,

1. Pre-Hadoop Process Testing
2. MapReduce Process Validation
1. **Pre-Hadoop Process Testing:** HDFS lets you store huge amounts of data on a cloud of machines. When the data is extracted from various sources such as web logs, social media, RDBMS, etc., and uploaded into HDFS (Hadoop Distributed File System), an initial stage of testing is carried out as mentioned below.

- Verification of the data acquired from the original source to check if it is corrupted or not
- Validation of data files if they were uploaded into correct HDFS location
- Checking the file partition and then copying them to different data units
- Determination of a complete set of data to be checked
- Verification of synchronicity of the source data with that of the data uploaded into HDFS

2. **MapReduce Process Validation:** MapReduce Processing is a data processing concept used to compress the massive amount of data into practical aggregated compact data packets.

- Testing of business logic first on a single node then on a set of nodes or multiple nodes
- Validation of the MapReduce process to ensure the correct generation of the “key-value” pair
- After the “reduce” operation, validation of aggregation and consolidation of data
- Comparison of the output generated data with the input files to make sure the generated output file meets all the requirements

3. **ETL Process Validation and Report Testing:** ETL stands for Extraction, Transformation, and Load testing approach. This is the last stage of testing in the queue where data generated by the previous stage is first unloaded and then loaded into the downstream repository system i.e. Enterprise Data Warehouse (EDW) where reports are generated or a transactional system analysis is done for further processing.

- Check the correct application of transformation rules
- Inspection of data aggregation to ensure there is no distortion of data and it is loaded into the target system
- Ensure there is no data corruption by comparing with the HDFS file system data
- Validation of reports that include the required data and all indicators are displayed correctly

**B. Non-Functional Testing**

Since Hadoop processes large chunks of data of varying variety and speed, it becomes imperative to perform architectural testing of the Big Data systems to ensure the success of your project. This non-functional testing is performed in two ways, i.e. Performance Testing and Failover Testing.

1. **Performance Testing:** Performance Testing performs the testing of job completion time, memory utilization and data throughput of the Big Data system. The main objective of performance testing is not restricted to only an acknowledgment of application performance but to improve the performance of the Big Data system as a whole too. It is performed as follows:

- Obtain the metrics of performance of Big Data systems i.e. response time, maximum data processing capacity, speed of data consumption etc.
- Determine conditions which cause performance problems i.e. assessing performance limiting conditions
- Verification of speed with which MapReduce processing (sorts, merges) is executed
- Verification of storage of data at different nodes
- Test JVM Parameters such as heap size, GC Collection Algorithms, etc.
- Test the values for connection timeout, query timeout, etc.
2. Failover Testing: Failover testing is done to verify seamless processing of data in case of failure of data nodes. It validates the recovery process and the processing of data when switched to other data nodes. Two types of metrics are observed during this testing i.e.

- Recovery Time Objective
- Recovery Point Objective

To ensure all is working well, the data extracted and processed is undistorted and in sync with the original data, above-mentioned testing procedures are performed. Big Data processing could be batch, real-time or interactive hence when dealing with such huge amount of data, Big Data testing becomes imperative as well as inevitable.

**Big Data Testing - Challenges**

“75% of businesses are wasting 14% of revenue due to poor data quality”

– Experian Data Quality

“Data quality costs (companies) an estimated $14.2 million annually”

– Gartner

The McKinsey Company report emphasizes the role of next generation data integration platforms in ensuring only relevant data is identified as well as the importance of a quality analysis mechanism to convert it into an asset for the company

At present, testers process clean and structured data. However, they also need to handle semi-structured and unstructured data. Key issues that require relatively more attention in big data testing include:

- Data security.
- Performance issues and the workload on the system due to heightened data volumes.
- Scalability of the data storage media.

Data warehouse testing, performance testing, and test data management are the fundamental components of big data testing. Addressing these challenges is tantamount to verifying the entire big data testing continuum.

Here are 10 signs that clearly indicate if one needs help with Big Data and Analytics testing:

1. **High amounts of down-time:** During the deployment of Big Data applications revolving around predictive analytics, organizations might face a multitude of problems. It is almost certain that issues have gone un-checked during data collection in such cases. This is easily tackled by testing instantly during data collection and deployment, thereby reducing total down time.

2. **Issues with scalability:** Usually, the development cycle starts with smaller data sets and gradually progresses to handling larger sizes. If the initial runs of the application work as designed, but if results tend to deviate, issues with scalability become quite evident. One can avoid this entirely by using smart data samples to test the framework of the application at key moments.

3. **Poor efficiency:** Big data applications extract information from data sets across many channels in real time to perform data analysis. Frequently, the data obtained is extremely complex and is prone to be full of inaccuracies. The quality of data needs to be tested from the source to its last destination to ensure its reliability, thereby increasing the overall efficiency throughout the process.

4. **Bad optimization:** A manufacturer should ideally be able to create new business process with the help gained from Big Data and predictive analytics. Inability to handle data in an appropriate fashion over an extended period of time,
visibly indicates improper optimization of existing processes to deliver the best results. With the right mode of testing, this can be avoided.

5. **Inadequate quality**: While using Big Data, various characteristics associated with data need to be checked, some of them being: validity, accuracy, conformity, consistency, duplicity, etc. If one or more aspects are ignored, the quality of data takes a massive hit. An organization should invest in thoroughly checking the data to ensure proper functionality.

6. **Lapses in Security**: Issues with security while dealing with Big Data can be catastrophic for any organization. Data sets containing confidential information need to be protected to maintain client’s trust. Testing should be carried out at various layers of the application using different testing methods to avoid becoming a victim of hacking.

7. **Sub-par Performance**: Since Big Data applications interact with live data for real time analytics, performance is key. Performance testing, when run alongside other types of testing, such as scalability and live integration, testing allows one to stay competitive.

8. **Issues with the digitization of information**: Even today, a substantial amount of information exists in the non-digital forms [paper documents] and hence is not available at the click of a button. As organizations convert those to digital forms, it is important to adequately test the data to ensure information isn’t lost or corrupted.

9. **Inconsistent functionality**: Access to a lot of different data sets today is what makes Big Data lucrative. An enterprise can generate limitless possibilities with the right kind of knowledge. But if the results acquired over time with Big Data applications and predictive analytics turn out to be inconsistent, it becomes a case of hit or miss for the organization. Appropriate testing allows them to determine variability accurately and eliminates uncertainty.

10. **Ensuring competitive advantage**: An organization can continue to stay relevant in today’s highly competitive and dynamic market while using by employing the use of the right testing tools available to them to get the best results.

---

**Big Data Testing – Best Practices**

Testing teams and the testing process itself must be mature enough to understand and analyze challenges, surpass several layers of data transfers, pass quality checks, and facilitate transformation changes—with proper validation and filtering.

Therefore, testing professionals must look beyond mere test case execution. Working in a collaborative mode with developers, they must plan each iteration for aggregate and incremental data loads across various testing phases such as data preparation, integration, and validation.

Such validation should happen early in the cycle, during work model initiation, so that data model changes can be dealt with precision and minimal impact. In short, testers must become data scientists, capable of handling data structure layouts, processes, and data loads.

Some key points to keep in mind for the tester turned data scientist –

- Avoid the sampling approach. It may appear easy and scientific, but is risky. It’s better to plan load coverage at the outset, and consider deployment of automation tools to ingress data across various layers.
- Use predictive analytics to determine customer interests and needs. Derive patterns and learning mechanisms from drill-down charts and aggregate data.
- Ensure correct behavior of the data model by incorporating alerts and analytics with predefined result data sets.
- Identify KPIs and a set of validation dashboards after careful analysis of algorithms and computational logic.
- Ensure right-time incorporation of changes in reporting standards and requirements. This calls for continuous collaboration and discussions with stakeholders. Associate the changes with the metadata model as well.
- Deploy adequate configuration management tools and processes.
Case Studies

There is also a need for best-in-class testing platforms and customizable testing frameworks, and a changed management mindset.

- For instance, a leading US bank was able to successfully improve their portfolio management and customer retention with a comprehensive data testing strategy. Insights drawn from the customer transaction data had revealed a relatively less positive association with the bank, and the need for improved targeted marketing through campaigns and promotional offers.
  
  In an unprecedented move at this organization, testers in their roles as data scientists filtered inefficiencies in business governance, and thus the QA function laid the foundation for a strong data warehouse structure.
  
  QA and testing best practices and effective score handling techniques enabled the bank to initiate proactive measures to limit customer churn, improve customer retention and increase the customer base while delivering personalized interactions.

- In another interesting case, at a telecom company, data testing practices helped improve decision making, based on patterns, insights, and key performance indicators from massive sets of aggregated data. With use cases focused on user journeys – from activation to product purchase, QA interventions enriched the aggregated warehouse data with proper insights, fusing it with improved dimensions and metrics. The improved governance and channel experience resulted in enhanced customer retention.

- Big data is still emerging and there is a lot of onus on testers to identify innovative ideas to test the implementation. Testers can create small utility tools using excel macros for data comparison which can help in deriving a dynamic structure from the various data sources during the pre-Hadoop processing stage.

  For instance, for one of the large retail customers’ big data implementation in the BDT (Behavior Driven Testing) a test automation framework model using Cucumber and Ruby was designed. The framework helped in performing count and data validation during the data processing stage by comparing the record count between the Hive and SQL tables and confirmed that the data is properly loaded without any truncation by verifying the data between Hive and SQL tables.

Similarly when it comes to validation on the map-reduce process stage, it definitely helps if the tester has good experience on programming languages. The reason is unlike SQL where queries can be constructed to work through the data MapReduce framework transforms a list of key-value pairs into a list of values. A good unit testing framework like Junit or PyUnit can help validate the individual parts of the MapReduce job but they do not test them as a whole.

Building a test automation framework using a programming language like Java can help here. The automation framework can focus on the bigger picture pertaining to MapReduce jobs while encompassing the unit tests as well.

Setting up the automation framework to a continuous integration server like Jenkins can be even more helpful. However, building the right framework for big data applications relies on how the test environment is setup as the processing happens in a distributed manner here. There could be a cluster of machines on the QA server where testing of MapReduce jobs should happen.
Challenges of Big Data

In its simplest form there are four major challenges that every organization will have to face when implementing a Big Data platform.

1. **Ownership**: Since Big Data is heavily business-oriented, top management of the organization will have to play a major role. They should be the leaders of Big Data projects. Big Data is helping organizations of all sizes to make better business decisions, save costs, improve customer service, deliver better user experience, and identify security risks. The insights gleaned from Big Data and the corresponding organization changes have to be managed very carefully. That's why top management has to play a vital role in this.

2. **Data**: Identifying the correct and most relevant data is another challenge as there are various sources of it. Only the relevant data will help in producing meaningful insight to guide the management to take critical decisions. For example, if the organization wants to analyze the customer experience on their website, it would be good to collect the failure login attempts and other related error logs in the site rather than collecting the log of only successful connections.

3. **People**: For a successful Big Data project, the team should be a mixture of Data Scientists, Technology experts, and Business owners. Data Scientists will use their skills and expertise to correlate data sets, identify patterns, and generate the final insights. Technology experts form the core of the Big Data initiative by playing a role in identifying the right set of software and hardware tools required for the platform. Business Owners define the outcome and work with Data Scientists and Technology experts to achieve the outcome at the right time.

4. **Technology**: This is the backbone of the Big Data platform in any organization. Hardware infrastructure and software tools are the two technology components that need to be in place according to the requirement of the organization. Cloud computing can be one of the options from a hardware infrastructure point of view. Tools like Hadoop, NoSQL, and MongoDB should be identified and selected for collecting, processing, storing, and analyzing data sets.

All of the challenges above need to be addressed and managed in a balanced way so that the Big Data project will succeed. Neglecting any one of these will create great problems for Big Data projects.

Common Myths about Big Data

Those starting their Big Data journey should be aware of some common myths so that the project will not be a waste of time or manpower.

1. **Big is simple**: We know that Apache Hadoop can store and process tons of data and it provides an inbuilt fault tolerance like in-cluster replication to improve cluster availability. However, HDFS doesn’t natively provide a solution for advanced data protection or disaster recovery. For such functionality, enhanced Hadoop distributions like that from MapR would be required.

2. **Fast analytics using Hadoop**: A common misconception about Hadoop is that it's fast. It is only designed for high throughput batch-style processing to reduce the impact due to common hardware failures in systems. However, these days there are a number of enhancements to address the issue of performance, among them integrating traditional database, streaming data, and in-memory processing products.

3. **Store everything**: Big Data hype has created an impression that Big Data can store forever all the data that an enterprise can have. It could be true, but the ultimate purpose of having a data analytic solution will not be utilized. The truth is you can expect a faster, more efficient, and cost-effective solution when you store less needed data on the framework.
4. **Start it as others are doing it:** This is the wrong approach, at least to Big Data, as it can waste money and effort. Putting tons of data on a scalable cluster and expecting the Data Scientist to pull out the insights for you will never happen. As with any other project, success will mostly depend on having a thought-out plan and strategy in place to drive the whole framework of tools and other resources, including Data Scientists.

**Conclusion**

Big Data is a buzzword heard just about everywhere. Why is Big Data getting this much attention? Because it has the potential to profoundly affect the way we do business. In the past, we used to look at small data and make our decisions. Now, with the Internet of Things and technology advances, we are moving huge sets of data (instead of moving technology towards data) to computing. This has become what is known as Big Data Analytics.

Big Data is about analytics, not storage. Start with questions, not data. All problems are not Big Data problems. We will have to audit our data, classify problems, and set an approach. Invest in up-skilling the resources, build the parts, and plan the whole. In the coming years, Big Data is going to transform how we live, how we work, and how we think. Using the positive sides of Big Data Analytics and the corresponding insights holds the promise of changing the world. Hence, Big Data is a big deal!

To ensure all is working well, the data extracted and processed is undistorted and in sync with the original data, above-mentioned testing procedures are performed. Big Data processing could be batch, real-time or interactive hence when dealing with such huge amount of data, Big Data testing becomes imperative as well as inevitable.

Misconception, must be cleared – if one thinks Big Data testing is just to gain better insights and intelligence. Big Data testing is also about improving customer’s experience with business, brand, and products. Start small, and then scale big.
References & Appendix

1. Generating Value from Big Data by Babu Analakkat EMC © 2015
2. Big Data Analytics: Turning Big Data into Big Money by Frank Ohlhorst John Wiley & Sons © 2013
3. Understanding Big Data by Chris Eaton, Dirk Deroos, Tom Deutsch, George Lapis and Paul Zikopoulos
5. http://blog.xebia.in
Author Biography

Brijesh Gopalakrishna

Brijesh has around 6 years of experience working with cross functional team on large scale business transformation projects for the LSHC, Telecom clients. Has over 5 years of experience in Functional testing and Performance testing of custom applications. Proficient in Requirement gathering, Test plans, Test design, Test execution and preparing Test summary reports. Brijesh holds a Post-Graduation in Engineering.

Ashish Mishra

Ashish has a total of more than 7 years of industry experience. He has worked in Banking, Insurance and Healthcare domains. He has extensive experience in software testing. He is proficient in various aspects of testing and quality assurance. He is an engineering graduate, and holds multiple certifications like ISTQB, Epic, ALM etc.
THANK YOU!