DATA ANALYTICS QA STRATEGY & APPROACH

Article Particulars: Received: 08.03.2018 Accepted: 23.03.2018 Published: 28.04.2018

RAMKUMAR SOUNDARAPANDIAN
Senior Manager – Ecommerce, Marketing Automation & Cloud Technology
Capgemini America Inc, United States of America

Abstract

Big Data is a term encompassing the use of techniques to capture, process, analyses and visualize potentially large datasets in a reasonable timeframe not accessible to standard IT technologies. By extension, the platform, tools and software used for this purpose are collectively called "Big Data technologies.

The current technology enables us to efficiently store and query large datasets, the focus is now on techniques that make use of the complete data set, instead of sampling. This has tremendous implications in areas like machine learning, pattern recognition and classification, to name a few.

With the coming of Big data technologies like Hadoop, NoSQL, Messaging Queues etc. organization have got the tools to dive deep into the large amounts of data and come up with analytics and intelligence that can help them drive business growth and take right decisions in time. But, testing Big data is one of the biggest challenges that the organizations face because of the lack of knowledge on what to test and how to test. They have been facing challenges in defining appropriate test strategies, tools, working with NoSQL, setting up optimal test environments and so on.

Different testing types like functional and non-functional testing are required to ensure that the data from varied sources is processed error free and is of good quality to perform analysis. Functional testing activities like validation of map reduce process, structured and unstructured data validation, data storage validation are important to ensure that the data is correct and is of good quality. Apart from functional validations other non-functional testing like performance and failover testing plays a key role to ensure the whole process is scalable and is happening within specified SLA.

Big Data Testing Approach

In the early days, a large sample was over 100 records. In today's Big Data era, papers routinely report many thousands, and even millions of records. Large samples provide a powerful tool for testing. We focus on regression models that are popular with researchers, but the approach generalizes to inference with other statistical models. The super-power approach encompasses the different modelling steps from framing and study design to data visualization, model building, validation and inference. The super-power approach enables testing more pointed and more complex, including more control variables, quantifying more subtle and rare relationships, improving robustness checking, strengthening model validity and generalizability, developing insights through analysis of subsamples, and making inferences even in the presence of some violated model assumptions.

Small-Sample Testing Approach

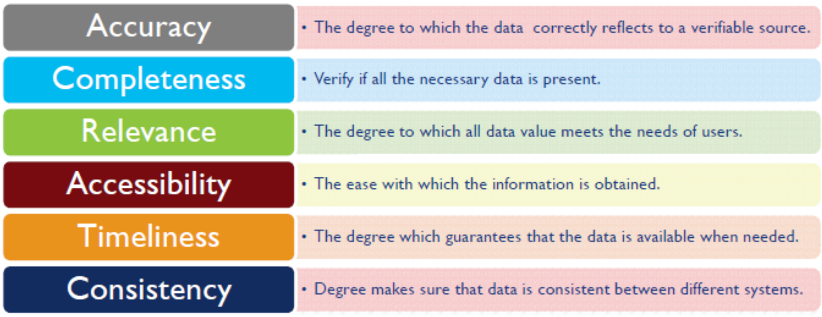
The small-sample approach assumes that data are scarce and hence the amount of information that they can contain is limited due to limited statistical power and increased sampling error. Simple hypotheses and models are chosen in order to allow proper estimation and inference from the small dataset.

Super-Power Test Approach

Large samples are advantageous for testing hypotheses due to high statistical power and reduced sampling error (smaller type I and II error rates). In particular, large samples allow testing more pointed and more complex hypotheses, including many control variables, and quantifying more subtle and rare relationships; they offer improved robustness checking, model validity assessment (internal and external), and predictive power evaluation, and allow inferences even in the presence of some violated model assumptions. The super-power approach encompasses the entire data analysis process, from framing and study design, through data exploration, modeling and deriving conclusions. This is illustrative of the need for different tools and approaches for testing hypotheses with Big Data, and the ability of appropriate tools to discover more intricate effects.

Data Quality and its Dimensions

- Data quality is a perception of data to be fit to serve its purpose. (Fitness for use)
- Various dimension of Data quality are



Accuracy	• The degree to which the data correctly reflects to a verifiable source.
Completeness	• Verify if all the necessary data is present.
Relevance	• The degree to which all data value meets the needs of users.
Accessibility	• The ease with which the information is obtained.
Timeliness	• The degree which guarantees that the data is available when needed.
Consistency	• Degree makes sure that data is consistent between different systems.

Data Quality Challenges in Big Data

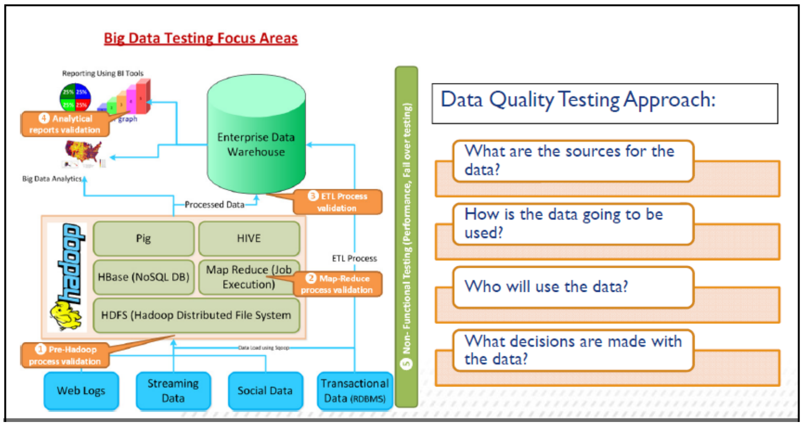
- In Big Data, the data is received from unstructured sources
- There is no control on the quality of the created data from unstructured sources
- Reduced time for analysis and action
- Analytical results from the unstructured data need to be integrated into the existing DW/BI architecture
- Validation of messages from unstructured data sources should be ascertained to derive the exact meaning in the context.

Data Quality Testing Approach

- Possible text issues from unstructured sources
- Consider checking for misspelled words
- Manage synonym list (lvm – left voice message or left a message)
- For social media's instant message abbreviations
- Terminology related to banking or relevant industry
- Check for invalid data from the sensors (RFID Tags, Manufacturing Sensors etc.)

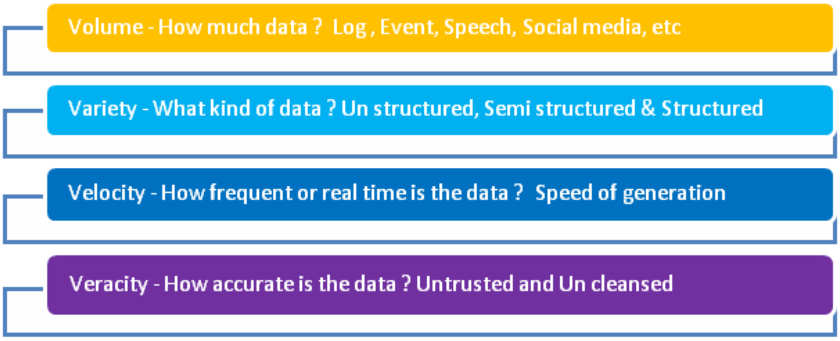
Testing Big data is one of the biggest challenges faced by organizations because of lack of knowledge on what to test and how much data to test. Organizations have been facing challenges in defining the test strategies for structured and unstructured data validation, setting up an optimal test environment, working with non-relational databases and performing non-functional testing. These challenges are causing in poor quality of data in production and delayed implementation and increase in cost. Robust testing approach need to be defined for validating structured and unstructured data and start testing early to identify possible defects early in the implementation life cycle and to reduce the overall cost and time to market.

Different testing types like functional and non-functional testing are required along with strong test data and test environment management to ensure that the data from varied sources is processed error free and is of good quality to perform analysis. Functional testing activities like validation of map reduce process, structured and unstructured data validation, data storage validation are important to ensure that the data is correct and is of good quality. Apart from functional validations other non-functional testing like performance and failover testing plays a key role to ensure the whole process is scalable and is happening within specified SLA. Big data implementation deals with writing complex Pig, Hive programs and running these jobs using Hadoop map reduce framework on huge volumes of data across different nodes. Hadoop is a framework that allows for the distributed processing of large data sets across clusters of computers. Hadoop uses Map/Reduce, where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. Hadoop utilizes its own distributed file system, HDFS; which makes data available to multiple computing nodes.


Big Data Testing Focus Areas
Reporting Using BI Tools
25% 25%
25% 25%
graph
4 Analytical reports validation
Big Data Analytics
Enterprise Data Warehouse
Processed Data
3 ETL Process validation
Pig
HIVE
HBase (NoSQL DB)
Map Reduce (Job Execution)
HDFS (Hadoop Distributed File System
hadoop
ETL Process
2 Map-Reduce process validation
1 Pre-Hadoop process validation
Data Load using Sqoop
Web Logs
Streaming Data
Social Data
Transactional Data (RDBMS)
5 Non- Functional Testing (Performance, Fail over testing)
Data Quality Testing Approach:
What are the sources for the data?
How is the data going to be used?
Who will use the data?
What decisions are made with the data?

Volume, Variety, Velocity and Veracity: How to Test?

During these phases of Big data processing, the four dimensions or characteristics of Big data i.e. volume, variety, veracity and velocity are validated to ensure there are no data quality defects and no performance issues.



Volume - How much data ? Log , Event, Speech, Social media, etc

Variety - What kind of data ? Un structured, Semi structured & Structured

Velocity - How frequent or real time is the data ? Speed of generation

Veracity - How accurate is the data ? Untrusted and Un cleansed

Volume

The amount of data created both inside corporations and outside the corporations via the web, mobile devices, IT infrastructure, and other sources is increasing exponentially each year. Huge volume of data flows from multiple systems which need to be processed and analyzed. When it comes to validation it is a big challenge to ensure that whole data setup processed is correct. Manually validating the whole data is a tedious task. We should use compare scripts to validate the data. As data is stored in HDFS is in file format scripts can be written to compare two files and extract the differences using compare tools. To reduce the time for execution we can either run all the comparison scripts in parallel on multiple nodes just like how data is processed using Hadoop map-reduce process. This approach will reduce further regression testing cycle time. When we don't have time to validate complete data, sampling can be done for validation.

Variety

The variety of data types is increasing, namely unstructured text-based data and semi-structured data like social media data, location- based data, and log-file data.

Structured Data is data which is in defined format which is coming from different RDBMS tables or from structured files. The data that is of transactional nature can be handled in files or tables for validation purpose. Semi structured data does not have any defined format but structure can be derived based on the multiple patterns of the data.

Velocity

The speed at which new data is being created – and the need for real-time analytics to derive business value from it -- is increasing thanks to digitization of transactions, mobile computing and the sheer number of internet and mobile device users. Data speed needs to be considered when implementing any Big data appliance to overcome performance problems.

Veracity

Veracity refers to the quality, accuracy, integrity, and credibility of data. The data gathered could have missing pieces, be inaccurate, or fail to provide real, valuable insights. Veracity, in essence, reflects the level of trust placed in the collected data.

Data can sometimes become messy and challenging to use. A large volume of data can cause more confusion than insights if it's incomplete. For example, in the medical field, incomplete data about a patient's medication could endanger their life.

Both value and veracity play crucial roles in defining the quality and insights derived from data. Thresholds for the truth of data often exist -- and should exist -- in an organization at the executive level to determine its suitability for high-level decision-making.

Functional Testing

As we are dealing with huge data and executing on multiple nodes there are high chances of having bad data and data quality issues at each stage of the process. Data functional testing is performed to identify these data issues because of coding errors or node configuration errors. Testing should be performed at each of three phases of Big data processing to ensure that data is getting processes without any errors.

Validations of Pre-Hadoop Processing

Data from various sources like weblogs, social network sites, call logs, transactional data etc., is extracted based on the requirements and loaded into HDFS before processing it further.

Issues

Some of the issues we face during the phase of moving data from source systems to Hadoop include incorrect data captured from source systems, improper storage of data, and incomplete or incorrect replication. We compare the input data file against the data in source systems to ensure accurate extraction.

Validations

- Some high level scenarios that need to be validated during this phase include:
- Validating that data processing is completed and output file is generated.

- Validating the business logic on standalone node and then validating after running against multiple nodes
- Validating the aggregation and consolidation of data after reduce process.
- Validating the output data against the source files and ensuring the data processing is completed correctly.

Validation of Data Extract and Load into EDW

Once map reduce process is completed and data output files are generated, this processes data is moved to enterprise data warehouse or any other transactional systems depending on the requirement. Some issues that we face during this phase include incorrectly applied transformation rules

Issues

Incorrect load of HDFS files into EDW and incomplete data extract from Hadoop HDFS.

Validations

Some high leave scenarios that need to be validated during this phase include:

- Validating that transformation rules are applied correctly
- Validating the load in target system
- Validating the aggregation of data
- Validating the data integrity in the target system

Validation of Reports

Analytical reports are generated using reporting tools by fetching the data from EDW or running queries in Hive.

Issues

Some of the issues faces while generating reports are report definition not set as per the requirement, report data issues, layout and format issues.

Validations

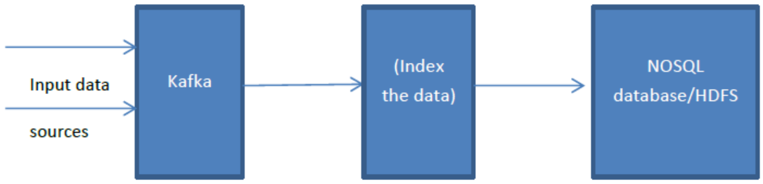
Some high level validations performed during this phase include:

- Cube Testing
- Dashboard Testing

Non-functional Testing

In the earlier sections we have seen how functional testing is performed at each phase of Big data processing, these tests are performed to identify functional coding issues, requirements issues. Performance testing and failover testing need to be performed to identify performance bottlenecks and to validate the non-functional requirements.

Performance Test Focus Area



The above diagram is a very high level representation of a big data analytics application. As a first step, multiple input streams are used to input data in the system via Kafka queues. The input data goes through the queue and it 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.

Unlike the traditional web applications that are performance tested from the end user perspective, these systems present an altogether different performance testing areas that we need to focus:

- Data ingestion and throughout
- Data processing
- Subcomponent performance

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, and Hadoop monitoring etc. can be used to capture performance metrics and identify the bottlenecks. Performance metrics like memory, throughput, job completion time etc.

Performance Testing Challenges

Performance testing Big Data is one of the challenges faced by the organizations because of lack of knowledge on what to test, how to test and how much data to test. Organizations have been 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. These challenges are responsible for poor quality in production, delayed implementation and increase in cost. Let’s look at some of these challenges in a bit more details:

- Diverse set of technologies
- Unavailability of specific tools
- Test scripting
- Test environment
- Monitoring solutions
- Diagnostic solutions

Critical Performance Areas

As a general rule, it’s important to note that simply adding nodes to a cluster will not improve performance on its own. You need to replicate the data appropriately, and then send traffic to all the nodes from your clients. If you aren’t distributing client requests, the new nodes could just stand by somewhat idle.

The following is the list of important areas that should be looked at and monitored to achieve optimum performance from the Big data cluster.

Data Storage: How data is stored across different nodes. What is the replication factor?

Commit logs: You can set the value for how large the commit log is allowed to grow before it stops appending new writes to a file and creates a new one.

- Concurrency
- Caching
- **Timeouts**: Values for connection time out, query timeout etc.

- JVM parameters: GC collection algorithms, heap size etc.
- Map reduces performance: Sorts, merge etc.
- Message queue: Message rate, size etc.

Failover Testing

Hadoop architecture consists of a name node and hundreds of data notes hosted on several server machines and each of them are connected. There are chances of node failure and some of the HDFS components become non-functional. Some of the failures can be name node failure data node failure and network failure. HDFS architecture is designed to detect these failures and automatically recover to proceed with the processing.

Failover testing is an important focus area in Big data implementations with the objective of validating the recovery process and to ensure the data processing happens seamlessly when switched to other data nodes.

Some validations that need to be performed during failover testing are validating that checkpoints of edit logs and FsImage of name node are happening at a defined intervals, recovery of edit logs and FsImage files of name node, no data corruption because of the name node failure, data recovery when data node fails and validating that replication is initiated when one of data node fails or data become corrupted. Recovery Time Objective (RTO) and Recovery Point Objective (RPO) metrics are captured during failover testing.

Best Practices

Data Quality

It is very important to establish the data quality requirements for different forms of data like traditional data sources, data from social media, and data from sensors, etc. If the data quality is ascertained, the transformation logic alone can be tested, by executing tests against all possible data sets.

Data Sampling

Data sampling gains significance in Big data implementation and it becomes the testers' job to identify suitable sampling techniques that includes all critical business scenarios and the right test data set.

Automation

Automate the test suites as much as possible. The Big data regression test suite will be used multiple times as the database will be periodically updated. Hence an automated regression test suite should be built to use it after reach release. This will save a lot of time during Big data validations.

The most effective approach to regression testing is based on developing a library of tests made up of a standard set of test cases that can be run every time you build a new version of the program. The most difficult aspect involved in building a library of test cases is determining which test cases to include. The most common suggestion from authorities in the field of software testing is to avoid spending excessive amounts of time trying to decide and err on the side of caution. Automated tests, as well as test cases involving boundary conditions and timing almost definitely belong in your library.

Periodically review the regression test library to eliminate redundant or unnecessary tests. Do this about every third testing cycle. Duplication is quite common when more than one person is writing test code. An example that causes this problem is the concentration of tests that often develop when a bug or variants of it are particularly persistent and are present across many cycles of testing. Numerous tests might be written and added to the regression test library. These multiple tests are useful for fixing the bug, but when all traces of the bug and its variants are eliminated from the program, select the best of the tests associated with the bug and remove the rest from the library.

The Regression test set (RTS) is about our system. If the system changes, then if the RTS doesn't also change to reflect the changes to the system, it will eventually drift out of step with the current system and will eventually become useless and misleading. Care should be taken in identifying the appropriate regression test cases that fit for purpose.

A regression test suite is the set of test scenarios that are designed to ensure that software is accurate and correct after undergoing corrections or changes. Each level of testing (i.e., unit testing, system testing, and acceptance testing) should have its own regression test suite. For example, unit testing would have a set of white box tests used to originally test the code or designed to test the changed area, system testing would have a set of black box tests, and acceptance testing would have a separate set of black box tests.

Conclusion

Data quality challenges can be effectively addressed by deploying a structured testing approach for both functional and non-functional requirements. Applying the right test strategies and following best practices will enhance testing quality, helping identify defects early and reducing the overall cost of implementation. It is essential for organizations to invest in building skill sets in both development and testing.

Big data testing is a specialized stream, and a testing team should be equipped with a diverse skill set, including coding, white box testing skills, and data analysis skills. This diverse skill set enables them to perform a better job in identifying quality issues in data.

References

1. Big data overview, Wikipedia.org at http://en.wikipedia.org/wiki/Big_data.
2. Kelly, J. (2012), Big data: Hadoop, Business Analytics and Beyond, A Big data Manifesto from the Wikibon Community. Available at http:// wikibon.org/wiki/v/Big_Data:_ Hadoop,_Business_Analytics_and_ Beyond, Mar 2012.
3. Analytics in 2012 Backs Big Data, Cloud Trends. Justin Kern, Information Management, http://www.information-management.com/news/analytics-predictive-big-data-cloud-IIA-Davenport-10021670-1.html
4. James Kobielus, Forrester Hadoop: What Is It Good For? Absolutely . . . Something, http://blogs.forrester.com/james_kobielus/11-06-06-hadoop_what_is_it_good_for_absolutely_something