Strengthening the Quality of Big Data Implementations

Open-source technologies are helping organizations across industries gain strategic insights from the torrents of data that now flow through IT systems. A robust big-data validation framework can significantly improve high-volume, big-data testing — helping to fortify quality assurance, refine analytical thinking and enhance the overall customer experience.

Executive Summary

Today’s enterprises rely on big-data analytics to inform everything from new product development and customer service to strategic decision making across the business landscape. Yet transforming these insights into serviceable and competitive products challenges IT departments to more effectively make meaning from the vast pools of structured, unstructured and semi-structured data they must manage. According to a recent report from McKinsey & Company,\(^1\) decoding big data requires next-generation data-integration platforms to ensure that all relevant data is gathered and prepared for further analysis — a task that involves enhancing quality assurance (QA) mechanisms to confirm data integrity.

Considering the increasing demand for QA, it is essential that IT organizations deploy customizable testing platforms. To keep pace with these requirements, industry experts are building quality-assurance frameworks around open-source platforms like Hadoop.

Since big data is a specialized domain, it requires specialized testing — a job that is traditionally performed manually. Automating this task remains a work in progress. Nonetheless, given the exponential growth of big data, testers need to stay abreast of ever-evolving technologies, and leverage the latest and best methods for validating these assets. This compels the testing community to develop innovative accelerators and automated solutions for assuring the quality of big data.

In this white paper, we touch upon the additional skill sets data warehouse testers will need to achieve success in validating big data. We also examine ways to test “the 3 Vs” of big data: volume (scale of data), variety (different forms of data) and velocity (analysis of streaming data in microseconds). Finally, we illustrate the focal points for testing and validating the end-to-end architecture of Hadoop and its integrated systems.
The Need for a Big-Data Validation Framework

Leading enterprises are scrambling to get a big slice of the big data pie — putting pressure on big-data experts to actively formulate new and improved testing frameworks. Considering that technologies change often before they find their roots, it is imperative for enterprises to develop a ready-to-use product within a short time frame — making automated testing a critical part of the big-data equation.

Best-of-breed testing frameworks empower respective businesses and their IT teams to rapidly develop, test and deploy applications — justifying investments in sophisticated testing tools and accelerators. Big-data validation frameworks should be comprehensive yet easy-to-use — designed to test high-volume, data-centric big data implementations.

These frameworks can enable testing and QA communities to:

• Reuse testing components.
• Detect defects earlier — preventing data anomalies in the first stages of the software development life cycle (SDLC).
• Dramatically improve requirements traceability.
• Automate test planning, test designing and test execution at every stage in the big-data testing lifecycle.
• Empower non-technical business users to create automated test processes and test cases without any prior scripting knowledge.
• Gain deeper insights to support key business decisions and strategic planning.

Experts are utilizing open-source platforms such as Hadoop to transform testing methodologies. Considering the wide spectrum of Hadoop components that are built to accommodate diverse technologies, IT organizations can take full advantage of this comprehensive, one-stop solution/framework for big-data testing.

Dive Deep into Hadoop

Traditional data-sampling techniques are no longer viable options for big-data testing. Today, testers must develop automated scripts that help achieve better test coverage and higher quality in less time. This requires big-data testers to

---

The House of Hadoop

![Figure 1](image-url)
assemble tailored short programs, scripts and accelerators that can be packaged as a one-stop testing framework.

The Hadoop architecture (see Figure 1) comprises two core components – HDFS and MapReduce – with the flexibility to integrate additional elements (Hive, HBase, Pig, Sqoop, Flume). NoSQL databases Mongo, Cassandra, Marklogic, traditional databases (Oracle, MySQL, SQL Server) and business intelligence tools such as Tableau, Datameer and Splunk can also be integrated for analytics and reporting.

Understanding the Hadoop Testing Spectrum

Big-data testing is integral to translating business insights harvested from big data and producing high-quality products. Testers will be better equipped to utilize superior testing mechanisms by understanding the big-data spectrum (see case study, above).

Hadoop Testing at a Glance

Testing in Hadoop can be categorized along the following pillars:

- Core components testing (HDFS, MapReduce)
- Essential components testing (Hive, HBase, NoSQL)
- Supporting components testing (Flume, Sqoop)
- Easy connectivity and validation of traditional RDBMS, NoSQL databases and Hadoop components.

Our big-data validation framework helped the client to:

- Reduce testing efforts by 60% – enabling faster time-to-market.
- Achieve ROI in six straight releases.
- Deliver comprehensive test coverage with complex data.

Helping a Telecom Assure the Quality of its Big-Data Implementation

A Fortune 500 media and telecommunications service provider migrated a huge data set (in volume and variety) captured from the Internet, set-top boxes – as well as mobile, home-security and control systems – to a big-data platform consisting of the following Hadoop components: HDFS, Hive, Pig, Sqoop and Flume. Cognizant worked with the client to implement a big-data validation framework that included:

- A customized tool that automates the process of capturing and validating data.
- A custom-built result reporting engine, which highlights discrepancies and presents results on a user-friendly dashboard.
- Hadoop cluster setup testing. Hadoop supports hundreds of nodes that sustain distributed processing and high availability. Multiple NameNodes and DataNodes, along with TaskTrackers and JobTrackers, run simultaneously. To ensure that the setup is ready prior to loading data into Hadoop, testers should perform simple smoke tests, such as HDFS format checks, health status checks of NameNode, DataNodes and Secondary NameNodes, as well as HDFS file copy and listing checks. Additionally, Hadoop is equipped with pre-built snippet programs such as Terasort, MRBench, NN Bench and MiniDF-SCluster/MiniMRCluster for testing the infrastructure setup.

Core Hadoop Components Testing

At its core, Hadoop is an open-source MapReduce implementation. Testing Hadoop can be a challenge, since the distributed processing of large data sets happens across clusters of computers. However, testers can help ensure efficient test coverage of the Hadoop ecosystem with systematic test-planning. To achieve the desired results, Hadoop components should be categorized and validation points finalized for each component to ensure successful end-to-end data flow.

- Hadoop cluster setup testing. Hadoop supports hundreds of nodes that sustain distributed processing and high availability. Multiple NameNodes and DataNodes, along with TaskTrackers and JobTrackers, run simultaneously. To ensure that the setup is ready prior to loading data into Hadoop, testers should perform simple smoke tests, such as HDFS format checks, health status checks of NameNode, DataNodes and Secondary NameNodes, as well as HDFS file copy and listing checks. Additionally, Hadoop is equipped with pre-built snippet programs such as Terasort, MRBench, NN Bench and MiniDF-SCluster/MiniMRCluster for testing the infrastructure setup.
- **HDFS testing.** HDFS, a distributed file system designed to run on commodity hardware, uses the master-slave architecture. To examine the highly complex architecture of HDFS, QA teams need to verify that the file storage is in accordance with the defined block size; perform replication checks to ensure availability of NameNode and DataNode; check for file load and input file split per the defined block size; and execute HDFS file upload/download checks to ensure data integrity.

- **Validation of MapReduce jobs.** Programming Hadoop at the MapReduce level means working with the Java APIs and manually loading data files into HDFS. Testing MapReduce requires testers who are skilled in white-box testing. QA teams need to validate whether transformation and aggregation are handled correctly by the MapReduce code. Testers need to begin thinking as developers. MapReduce core business logic can be validated with JUnit. However, Mapper and Reducer code testing is accomplished by introducing MRUnit tests, which are based on Junit, an extension for testing MapReduce code.

### Essential Hadoop Components

- **HBase testing.** HBase, a non-relational (NoSQL) database that runs on top of HDFS, provides fault-tolerant storage and quick access to large quantities of sparse data. Its purpose is to host very large tables with billions of rows and millions of columns. HBase testing involves validation of RowKey (PK) usage for all the access attempts to HBase tables; verification of version settings in the query output; verification that the latency of individual reads and writes is per the threshold; and checks on the Zookeeper to ensure the coordination activities of the region servers.

- **Hive testing.** Hive enables Hadoop to operate as a data warehouse. It superimposes structure on data in HDFS, then permits queries over the data using a familiar SQL-like syntax. Hive’s core capabilities are extensible by UDFs (user defined functions). Since Hive is recommended for analysis of terabytes of data, the volume and velocity of big data are extensively covered in Hive testing. (Hive testing can be classified into Hive functional testing and Hive UDF testing).

  From a functional standpoint, Hive testing incorporates validation of successful setup of the Hive meta-store database; data integrity between HDFS vs. Hive and Hive vs. MySQL (meta-store); correctness of the query and data transformation logic; checks related to number of MapReduce jobs triggered for each business logic; export/import of data from/to Hive; data integrity, and redundancy checks when MapReduce jobs fail.

  Hive core capabilities are made extensible by writing custom UDFs. The UDFs must be tested for correctness and completeness, which can be achieved using tools such as JUnit. Testers should also test the UDF in Hive CLI directly, especially if there are uncertainties in functions that deal with the right data types, such as struct, map and array.

- **NoSQL Testing.** NoSQL (NotOnlySQL) are next-generation databases that are non-relational, distributed, open-source and horizontally scalable. Some of the widely used NoSQL databases are HBase, Cassandra, Mongo, CouchDB, Riak and Marklogic. A data warehouse tester who has hands-on experience with any one of the traditional databases should be able to work with other traditional RDBMS, with minimal effort. However, a NoSQL database tester will need to acquire supplementary skills for each of the NoSQL databases in order to perform quality testing.

  Testing NoSQL databases can validate big data’s “variety” aspect. Because each NoSQL database has its own language, testers need to be well versed in the programming languages that are used to develop the NoSQL database. Also, NoSQL databases are independent of the specific application or schema, and can operate on a variety of platforms and operating systems.

  QA areas for these databases include data type checks; count checks of the documents in a collection; CRUD operations checks; master-slave integrity checks; timestamp and its format checks; checks related to cluster failure handling; and data integrity and redundancy checks on task failure.

  A manual testing approach around Hadoop components can be constructed into a robust QA framework. Moreover, there is a cascading scope for customizing the framework to a spe-
specific domain. In turn, this customized framework can facilitate key business decisions and strategic planning.

Supporting Components

Flume and Sqoop improve interoperability with the rest of the data world. Sqoop is a tool designed to import data from relational databases into Hadoop and vice versa. Flume directly imports streaming data into HDFS.

- **Flume Sqoop testing.** In a data warehouse, the database supports the import/export functions of data. Big data is equipped with data ingestion tools such as Flume and Sqoop, which can be used to move data into and out of Hadoop. Instead of writing a stand-alone application to move data to HDFS, it’s worth considering existing tools for ingesting data, since they offer most of the common functions. General QA checkpoints include successfully generating streaming data from Web sources using Flume, checks over data propagation from conventional data storages into Hive and HBase, and vice versa.

Looking Forward: The Only Constant is Change

Technology, more often than not, follows nature’s patterns. Business insights are harvested from abundant, untapped big data. Considering the increasing demand for big-data QA, it is essential that IT organizations deploy customizable testing platforms and frameworks. To keep pace, experts must build a robust QA framework for ensuring accuracy and consistency — enabling businesses and their IT departments to:

- Optimize resources to perform testing with respect to hardware, people and time.
- Employ domain-specific, built-in libraries to enable non-technical business users to perform testing.
- Address early automation in every phase of the big data software testing life cycle.

Footnotes


About the Author

Sushmitha Geddam is a Project Manager in the R&D team of Cognizant’s Data Testing Center of Excellence and leads the company’s big-data QA initiatives. During her 11-year diversified career, she has delivered multifaceted enterprise business software projects in a wide array of domains in specialized testing areas (big data, DW/ETL, data migration). She is responsible for providing delivery oversight, thought leadership and best practices; developing internal talent in technology CoEs, and defining the quality and testing processes and strategizing tools for big-data implementations. She is PMP-certified, a Cloudera-certified developer for Apache Hadoop and a Datameer-certified analyst. She can be reached at Sushmitha.Geddam@cognizant.com.

About Cognizant

Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services, dedicated to helping the world’s leading companies build stronger businesses. Headquartered in Teaneck, New Jersey (U.S.), Cognizant combines a passion for client satisfaction, technology innovation, deep industry and business process expertise, and a global, collaborative workforce that embodies the future of work. With over 75 development and delivery centers worldwide and approximately 199,700 employees as of September 30, 2014, Cognizant is a member of the NASDAQ-100, the S&P 500, the Forbes Global 2000, and the Fortune 500 and is ranked among the top performing and fastest growing companies in the world. Visit us online at www.cognizant.com or follow us on Twitter: Cognizant.