Journal of Engineering and Applied Sciences 13 (Special Issue 4): 3830-3836, 2018
ISSN: 1816-949X


# An Efficient System for Validation of Big Data Techniques

[1]Naveen Garg, [2]Sanjay Singla and [3]Surender Jangra
[1]Department of Computer Science Engineering, I.K. Gujral Punjab Technical University, Jalandhar, India
[2]GGS College of Modern Technology, Kharar, Punjab, India
[3]Guru Teg Bahadur College, Bhawanigarh, Sangrur, Punjab, India

**Abstract:** Big data is one of the major process nowadays. It is collection of huge amount of data which is increasing at an exceptional rate every day. Analysis of big data helps in taking decision in any business. Accuracy in analysis of big data plays an important role for any business in market. Various tools are being used to test big data. Therefore, there is need for testing of big data analytical techniques. This study proposed an efficient approach eBDTV (Efficient Big Data Testing Validation) which can be helpful in testing these big data techniques. This approach will help in testing of 4 V's, i.e., Volume, Variety, Velocity and Veracity along with their performance. Validation of requirements and data is also required which is necessary to take decision for any organization. This study discusses the challenges of testing and their solution in order to overcome these challenges.

**Key words:** eBDTV, big data, testing, challenges, organization, performance, solution, overcome

## INTRODUCTION

Big data can be described as the data which is difficult to process with the help of traditional databases and software. Big data consists of unstructured, structured and semi structured data. For any business, accurate processing of data is must. With the help of results obtained from big data processing, decision is taken for providing business solution. Data is received from different sources and every source stores the data in different format. So, it becomes difficult to process this unstructured data using traditional tools. On the other hand the speed at which data is received is very high therefore, it becomes tedious to process data in timely manner (Zikopoulos *et al*., 2011). The data can have undesirable values such as noise which have the tendency to change the results. With these reasons, big data analytics comes into picture. Using big data analytical tools like Hadoop, pig, hive, spark, etc., big data can be processed and analyzed efficiently. Now, the question comes into mind is that what is accuracy rate of this processed data? Will it be helpful in taking correct decisions for our business? Processed data is as per the needs of business or not? For answering these questions, testing plays a very vital role, i.e., testing of these big data techniques (Garg *et al*., 2016; Hewitt, 2010).

Big data consists of 4 v's, I.e., Volume, Velocity, Variety and Veracity. Figure 1 depicts the data coming from different sources in different structure along with its volume, velocity and veracity.

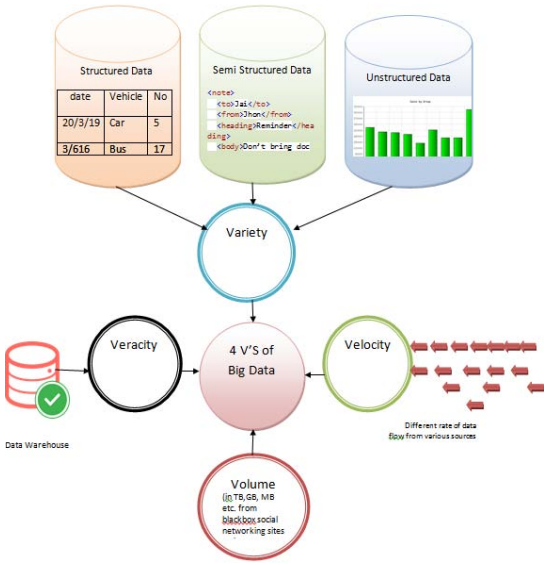


Fig 1: Representation of data corresponding to 4 V's

**Problems with big data testing:** One major issue of big data is that the data is highly unpredictable which is more often unstructured. Data is generated from various sources such as web logs, social websites, media and GPS sytems etc. Therefore, any conventional testing approach is of no use in case of big data testing. Accurate results are expected for taking decisions in business from huge data and veracity of data is also unpredictable. To go

**Correspondimg Author:** Naveen Garg, Amity University, Uttar Pradesh, India

through and access detailed information needed from this huge volume of data that too at a very high speed with increase in the degree of granularity makes this challenge worse. Some techniques are proposed for big data testing but every business have different expectations. So, validation is much more required which will help in validating the business need with results obtained from processing of data (Garg *et al*., 2016; Alexandrov *et al*., 2013).

## MATERIALS AND METHODS

**Proposed system:** Our research focuses on validating the requirements and data, testing the 4 V's (Volume, Variety, Velocity and Veracity), validating business processing logic followed by incremental and performance testing. After this, practical implementation is carried out which is explained in the coming topics.

**Steps to be followed for big data validation:**

- Validation of requirements
- Validation of data
- Testing of 4 V's
- Validation of business processing logic
- Performance testing
- Validation and transformation of data

**Validation of requirements:** Analysis of big data is helpful for taking decisions for any business which are further based on deep analysis of system. If our system is not processed with exact requirements of business, then it became waste to do analysis. Therefore, it is important to validate the requirements of business process.

**Validation of data:** After validation of requirements, it is important to gather proper data for processing. Big data systems of any organization should be capable of collecting required data from different sources. Big data analysis framework should b e capable of collecting and validating the data in any format. Data may be in any form, i.e., structured, semi structured or unstructured format. Structured data may in form of spread sheets, text files, etc. Unstructured data may be any image, video or data from web or other sources. Data may be generated from any source and it may be useful for us. So, it is not any standard form of collection of data. Data must be validated conformed to some requirements of system. If collected data is totally different from our required domain, then it will slow down processing of our system.

**Testing of 4 V's**

**Variety:** The diversified data is important along with the quantity. The variety in data can be regarding the gadgets or the sources from where the data can be generated. Such as cell phones, tablets, social media and so forth. In terms of sources, web-based social networking is the runaway pioneer. Facebook and Twitter alone are creating a larger number of information every day than some other specialized instruments. This shows that data is present in various formats and this is the problem with large amount of data that needs to be conquered. This large data is referred to as big data (Maheshwari and Chaturvedi, 2012).

**Velocity:** Big data velocity manages the pace at which information streams in from sources like business forms, machines, systems and human connection with things like web-based social networking destinations, cell phones, and so forth. The stream of data is enormous and consistent. This real time data can enable analysts and organizations to settle on profitable choices that gives advantages, profits and ROI in the event that you can deal with the speed.

**Volume:** It is evaluated that, on a normal, 2.3 trillion gigabytes of information is produced each day. It is unreasonable to collect this huge amount of data. Most organizations in the US have no <100,000 gigabytes of information stored and every one of them will disclose to you that they aren't gathering enough information. The correct approach is to battle the inclination of making your organization's server an information dump. Endeavors must be made to utilize the correct programming to channel the relevant information. Testing of this enormous amount of data is a challenge (Fig. 2).

**Veracity:** Veracity deals with the data which is not good. The fact is unfiltered information will probably be terrible. In spite of the fact that information quality and ease of use depends generally on the source, you can never rule out garbage. This lack of quality of data makes numerous businesses to depend on data examination. That is the wrong approach. There is no tool of business that is error free. Therefore, organizations should work harder on executing the correct innovation and individuals for its management in order to deal with unreliable data.

**Validation of business processing logic:** MapReduce plays a very important role. It is the role of tester to

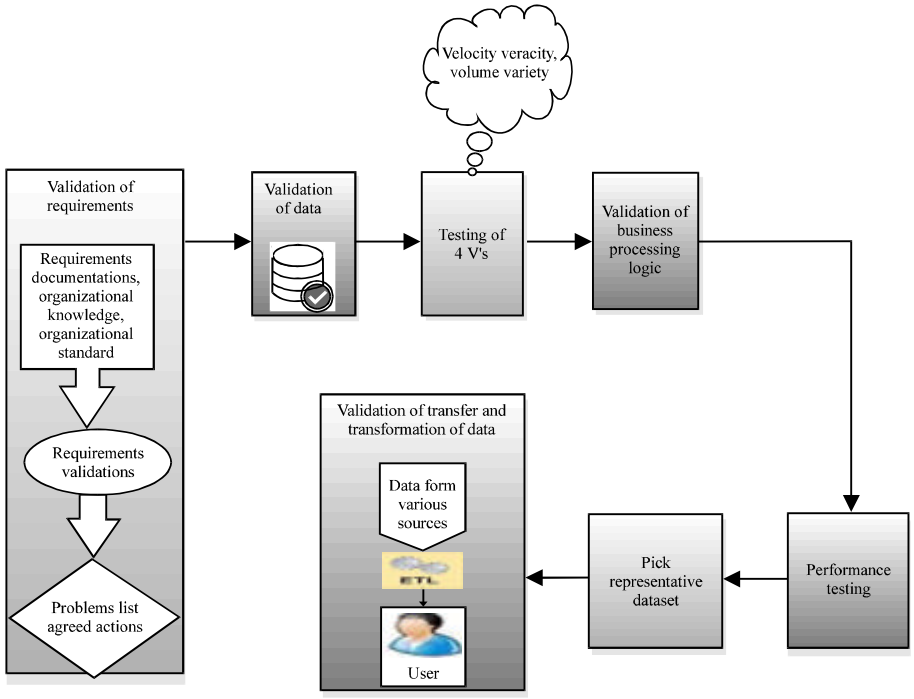


Fig. 2: eBDTV-proposed system for validation of big data techniques

examine the business logic validation for each and every node and thereafter validating by running across numerous nodes in order to ensure (Das and Pradhani, 2015):

- Guide accuracy in mapreduce process
- Data aggregation or isolation rules are maintained
- Creation of key esteem sets
- Handling the data after mapreduce step

**Performance testing**

**Challenges:** It is a challenge for organizations characterize strategies and ratify sub segments, making working environment suitable working with various frameworks such as NoSQL.

**Diverse arrangement of innovations:** Each segment in a major information has a place with an alternate innovation. So, we have to test every segment independentl (Alexandrov *et al*., 2013).

**Unavailability of particular tools:** There is no single tool which can be used for each segment.

**Test scripting:** No record or description is provided for such frameworks. In order to design test cases and scheme, scripting is required.

**Test condition:** Due to cost and scalability problem, it may not generally be possible to develop test conditions for performance testing. So, there is a down form to forecast performance of all segments.

**Monitoring arrangements:** Since, each portion has a substitute strategy for revealing execution estimations that can screen the whole condition for execution abnormalities and distinguish issues.

**Diagnostic arrangements:** In order to create to additionally anticipate the execution zones, custom arrangements need to be created.

**Testing needs:** Expanding requirement for live incorporation of data: As information is acquired from various sources, so, it winds up noticeably essential to coordinate the data which provide us with perfect and steady information.

**Instant data collection and deployment:** Determined actions and prescient examination need to be driven to embrace moment information gathering arrangements. These choices acquire huge business affect.

**Real-time adaptability needs:** In order to meet the conditions of versatility and data handling, big data

applications needs to be planned accordingly to provide suitable situations. Basic mistakes in the outline of big data applications can prompt basic circumstances. No-nonsense testing includes better execution.

**Testing approach:** Any huge information venture includes in preparing enormous volumes of organized and unstructured information and is handled over various hubs to finish the employment in less measure of time. Now and again in light of poor outline and design execution is corrupted. Huge data is collected and processed in performance testing. Many utilities including Zabbix, Hadoop, Casandra observing and so forth can be utilized to catch execution measurements and recognize the bottlenecks. Execution measurements like memory, throughput, work culmination time and so forth are basic for observing and examination.

As the procedure begins with clustering big data that needs to be tested against performance. The workload is created by identifying usage of each segment. Creation of custom contents is followed by simulation of real-time usage and identification of results. The clusters can be re-executed till the till the most extreme execution is accomplished.

**Testing tools**

**YCSB:** A benchmark tool for testing, a yahoo product which is used for reading, writing and updating to calculate throughput (Anonymous, 2016; Li *et al*., 2015).

**Sandstorm:** Another tool for performance testing which provide interface for scripting for big data stess test.

**JMeter:** Plugins are provided in JMeter which sends call over thrift. Complete configuration is provided for plugins.

**Apache drill:** Apache incubator segment offering performance analysis on large amount of data depend on Dremel in a distributed manner (Kaur *et al*., 2015).

## RESULTS AND DISCUSSION

**Pick representative dataset:** File in the format .csv from a local system is loaded in HDFS Table 1. Once file is loaded in HDFS, it is run in Pig as shown in Fig. 3. The data was tested and analyzed with the parameters of time and memory (Fig. 4 and 5).

Validation was done on the dataset of 480000 records. The memory signifies the total memory used by the dataset at the time of processing of mapper and reducer. The different memory parameters are validated such as total physical memory used, heap usage.

Table 1: Memory assessment of mapper and reducer

| Support | Total physical memory used (MB) | Total JVM heap usage (MB) | Total virtual memory used (MB) | Transaction count |
|---|---|---|---|---|
| 0.70 | 479.00 | 444.8 | 1110.7 | 480000 |
| 0.65 | 531.12 | 548.0 | 1509.0 | 480000 |
| 0.60 | 721.80 | 650.0 | 2190.7 | 480000 |
| 0.50 | 723.50 | 650.0 | 2191.7 | 480000 |
| 0.50 | 720.20 | 647.8 | 2187.7 | 480000 |
| 0.45 | 640.10 | 610.5 | 2110.1 | 480000 |
| 0.40 | 724.90 | 652.8 | 2190.5 | 480000 |
| 0.35 | 735.30 | 649.5 | 2190.6 | 480000 |
| 0.30 | 629.40 | 637.8 | 1930.6 | 480000 |
| 0.25 | 717.80 | 642.9 | 2187.5 | 480000 |

Table 2: Time assessment of mapper and reducer

| Support percentage | Total time (sec) | Elapsed time (sec) | CPU spend time (sec) | Transaction count |
|---|---|---|---|---|
| 0.70 | 71.0 | 65 | 30.19 | 480000 |
| 0.65 | 70.5 | 67 | 37.70 | 480000 |
| 0.60 | 71.9 | 68 | 49.20 | 480000 |
| 0.50 | 72.1 | 66 | 50.80 | 480000 |
| 0.50 | 71.7 | 65 | 50.90 | 480000 |
| 0.45 | 71.7 | 66 | 35.10 | 480000 |
| 0.40 | 71.7 | 66 | 47.80 | 480000 |
| 0.35 | 71.7 | 65 | 41.30 | 480000 |
| 0.30 | 71.6 | 66 | 35.40 | 480000 |
| 0.25 | 71.8 | 66 | 41.90 | 480000 |

Time signifies the total time required for the execution of mappers and reducers. Time was validated against different values and finally values were validated (Table 2). Figure 6 graph between Total time, elapsed time and CPU time vs support percentage.

Figure 6 and 7 signifies that as support value increases, the corresponding time and memory increases. The time includes total time required for the execution of the algorithm. Elapsed time is the time required for the execution of the processing of the mapper and reducer. The CPU time spend is the usage of cpu for the processing of the algorithm. These all are equally important for the analysis of the time and validation of the algorithm.

Therefore, it has been concluded that the memory and time required to process and store the data increases as we lower the support values.

**Validation of transfer and transformation of data:** In the midst of ETL shapes, we need to support data trade and change to ensure that data uprightness is not bartered. Favoring data trade is by and large fundamental since we know expected that regards are equal would special regards. If the data is traded from a database to another,

| | playerID | yearID | stint | teamID | lgID | G | G_batting | AB | R | H | 2B | 3B | HR | RBI | SB | CS | BB | SO | IB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | aardsda01 | 2004 | 1 | SFN | NL | 11 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 3 | aardsda01 | 2006 | 1 | CHN | NL | 45 | 43 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 4 | aardsda01 | 2007 | 1 | CHA | AL | 25 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 5 | aardsda01 | 2008 | 1 | BOS | AL | 47 | 5 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | |
| 6 | aardsda01 | 2009 | 1 | SEA | AL | 73 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 7 | aardsda01 | 2010 | 1 | SEA | AL | 53 | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 8 | aaronha01 | 1954 | 1 | ML1 | NL | 122 | 122 | 468 | 58 | 131 | 27 | 6 | 13 | 69 | 2 | 2 | 28 | 39 | |
| 9 | aaronha01 | 1955 | 1 | ML1 | NL | 153 | 153 | 602 | 105 | 189 | 37 | 9 | 27 | 106 | 3 | 1 | 49 | 61 | |
| 10 | aaronha01 | 1956 | 1 | ML1 | NL | 153 | 153 | 609 | 106 | 200 | 34 | 14 | 26 | 92 | 2 | 4 | 37 | 54 | |
| 11 | aaronha01 | 1957 | 1 | ML1 | NL | 151 | 151 | 615 | 118 | 198 | 27 | 6 | 44 | 132 | 1 | 1 | 57 | 58 | |
| 12 | aaronha01 | 1958 | 1 | ML1 | NL | 153 | 153 | 601 | 109 | 196 | 34 | 4 | 30 | 95 | 4 | 1 | 59 | 49 | |
| 13 | aaronha01 | 1959 | 1 | ML1 | NL | 154 | 154 | 629 | 116 | 223 | 46 | 7 | 39 | 123 | 8 | 0 | 51 | 54 | |
| 14 | aaronha01 | 1960 | 1 | ML1 | NL | 153 | 153 | 590 | 102 | 172 | 20 | 11 | 40 | 126 | 16 | 7 | 60 | 63 | |
| 15 | aaronha01 | 1961 | 1 | ML1 | NL | 155 | 155 | 603 | 115 | 197 | 39 | 10 | 34 | 120 | 21 | 9 | 56 | 64 | |
| 16 | aaronha01 | 1962 | 1 | ML1 | NL | 156 | 156 | 592 | 127 | 191 | 28 | 6 | 45 | 128 | 15 | 7 | 66 | 73 | |
| 17 | aaronha01 | 1963 | 1 | ML1 | NL | 161 | 161 | 631 | 121 | 201 | 29 | 4 | 44 | 130 | 31 | 5 | 78 | 94 | |
| 18 | aaronha01 | 1964 | 1 | ML1 | NL | 145 | 145 | 570 | 103 | 187 | 30 | 2 | 24 | 95 | 22 | 4 | 62 | 46 | |
| 19 | aaronha01 | 1965 | 1 | ML1 | NL | 150 | 150 | 570 | 109 | 181 | 40 | 1 | 32 | 89 | 24 | 4 | 60 | 81 | |
| 20 | aaronha01 | 1966 | 1 | ATL | NL | 158 | 158 | 603 | 117 | 168 | 23 | 1 | 44 | 127 | 21 | 3 | 76 | 96 | |
| 21 | aaronha01 | 1967 | 1 | ATL | NL | 155 | 155 | 600 | 113 | 184 | 37 | 3 | 39 | 109 | 17 | 6 | 63 | 97 | |
| 22 | aaronha01 | 1968 | 1 | ATL | NL | 160 | 160 | 606 | 84 | 174 | 33 | 4 | 29 | 86 | 28 | 5 | 64 | 62 | |
| 23 | aaronha01 | 1969 | 1 | ATL | NL | 147 | 147 | 547 | 100 | 164 | 30 | 3 | 44 | 97 | 9 | 10 | 87 | 47 | |
| 24 | aaronha01 | 1970 | 1 | ATL | NL | 150 | 150 | 516 | 103 | 154 | 26 | 1 | 38 | 118 | 9 | 0 | 74 | 63 | |

Fig. 3: Data sample in tabular form

```
me-2.5.jar to DistributedCache through /tmp/temp-763925313/tmp17906577/joda-time
-2.5.jar
2017-09-19 17:34:29,588 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.JobControlCompiler - Setting up single store job
2017-09-19 17:34:29,711 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - 1 map-reduce job(s) waiting for submission
.
2017-09-19 17:34:29,718 [JobControl] INFO  org.apache.hadoop.metrics.jvm.JvmMetr
ics - Cannot initialize JVM Metrics with processName=JobTracker, sessionId= - al
ready initialized
2017-09-19 17:34:29,793 [JobControl] WARN  org.apache.hadoop.mapreduce.JobResour
ceUploader - No job jar file set.  User classes may not be found. See Job or Job
#setJar(String).
2017-09-19 17:34:29,984 [JobControl] INFO  org.apache.hadoop.mapreduce.lib.input
.FileInputFormat - Total input paths to process : 1
2017-09-19 17:34:29,985 [JobControl] INFO  org.apache.pig.backend.hadoop.executi
onengine.util.MapRedUtil - Total input paths to process : 1
2017-09-19 17:34:29,995 [JobControl] INFO  org.apache.pig.backend.hadoop.executi
onengine.util.MapRedUtil - Total input paths (combined) to process : 1
2017-09-19 17:34:30,042 [JobControl] INFO  org.apache.hadoop.mapreduce.JobSubmit
ter - number of splits:1
2017-09-19 17:34:30,371 [JobControl] INFO  org.apache.hadoop.mapreduce.JobSubm
ter - Submitting tokens for job: job_local700166821_0003
```

Fig. 4: Screenshot of data processing using pig

we can endorse the source and target data quickly by checking the amount of segments, lines and the section's names and data sorts. In the occasion that time allows, every datum cell should be evaluated. The endorsement can be automated when source and target data are given. Favoring data change is more convoluted. For instance,

```
(zuvelpa01,1982,1,ATL,NL,2,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2)
(zuvelpa01,1983,1,ATL,NL,3,3,5,0,0,0,0,0,0,0,0,2,1,0,1,0,0,0,3)
(zuvelpa01,1984,1,ATL,NL,11,11,25,2,5,1,0,0,1,0,0,2,3,0,0,0,0,0,11)
(zuvelpa01,1985,1,ATL,NL,81,81,190,16,48,8,1,0,4,2,0,16,14,1,0,4,0,3,81)
(zuvelpa01,1986,1,NYA,AL,21,21,48,2,4,1,0,0,2,0,0,5,4,0,0,4,0,1,21)
(zuvelpa01,1987,1,NYA,AL,14,14,34,2,6,0,0,0,0,0,0,0,4,0,0,2,0,1,14)
(zuvelpa01,1988,1,CLE,AL,51,51,130,9,30,5,1,0,7,0,0,8,13,0,0,8,0,3,51)
(zuvelpa01,1989,1,CLE,AL,24,24,58,10,16,2,0,2,6,0,0,1,11,0,1,0,0,0,24)
(zuvelpa01,1991,1,KCA,AL,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2)
(zuverge01,1951,1,CLE,AL,16,16,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,16)
(zuverge01,1952,1,CLE,AL,2,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2)
(zuverge01,1954,1,CIN,NL,2,2,2,1,1,0,0,0,0,0,0,0,1,,0,0,0,0,2)
(zuverge01,1954,2,DET,AL,35,35,64,1,8,1,0,0,3,0,1,1,14,,0,9,0,2,35)
(zuverge01,1955,1,DET,AL,14,14,4,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,14)
(zuverge01,1955,2,BAL,AL,28,28,23,1,5,1,0,0,0,0,0,1,5,0,0,1,0,1,28)
(zuverge01,1956,1,BAL,AL,62,62,17,0,2,0,0,0,2,0,0,1,7,0,0,4,0,0,62)
(zuverge01,1957,1,BAL,AL,56,56,23,1,3,0,0,0,0,0,0,3,8,0,0,2,0,0,56)
(zuverge01,1958,1,BAL,AL,45,45,9,0,2,0,1,0,2,0,0,1,2,0,0,0,0,0,45)
(zuverge01,1959,1,BAL,AL,6,6,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,6)
(zwilldu01,1910,1,CHA,AL,27,27,87,7,16,5,0,0,5,1,,11,,,1,1,,,27)
(zwilldu01,1914,1,CHF,FL,154,154,592,91,185,38,8,16,95,21,,46,68,,1,10,,,154)
(zwilldu01,1915,1,CHF,FL,150,150,548,65,157,32,7,13,94,24,,67,65,,2,18,,,150)
(zwilldu01,1916,1,CHN,NL,35,35,53,4,6,1,0,1,8,0,,4,6,,0,2,,,35)
```

Fig. 5: Screenshot of processed data

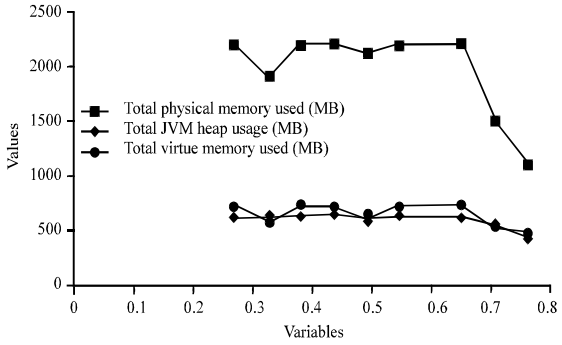


Fig. 6: Graph between total memory, heap usage and virual memory vs. support percentage

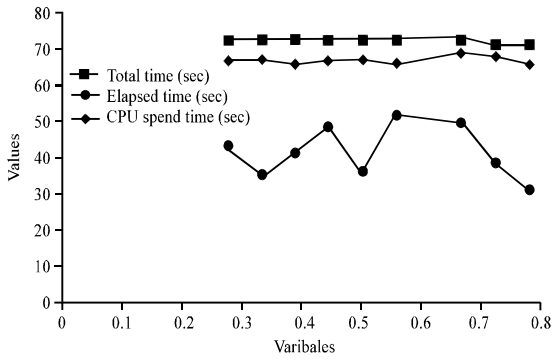


Fig. 7: Graph between total time, elapsed time and CPU time vs. support percentage

we may add up to data from ten source tables into one target table. A bit of the portions of the target table use comparable data sorts as extraordinary while distinctive fragments may use differing data sorts. We propose two plans at different endorsement granularity levels. In any case, we endorse whether the target data has cure data sorts and regard ranges at an irregular state. The test structure decides data sorts and regards ranges from necessities by then makes test to favor the due date. The second game plan decides point by guide speciffications toward affirm each change run the show. The test structure differentiates the source data and the target data to survey whether the target data was changed viably. The two plans anticipate that analyzers will create the change speciffication in an association that the test structure can read. By then the framework normally separates the speciffication and makes tests to favor the change. We simply favor the source and target data, not the advance concentrations in the inside. These are checked for frustration conclusion (researching) (Li *et al*., 2015).

## CONCLUSION

The proposed technique provides the solution for testing big data techniques in terms of business logic, validation of data and requirements as well as performance of processing techniques. In the present study, practical approach for analyzing the performance of given data is successfully carried out depicting the performance of the system. By this system, we can conclude that different organizations require different solutions as per their needs in order to solve specific challenges. All the affective factors have to be considered while coming to the take decision in any organization. As big data itself is huge, handling data is quite complex but once handled, performance can be improved and decision making can be properly managed.

## REFERENCES

Alexandrov, A., C. Brucke and V. Markl, 2013. Issues in big data testing and benchmarking. Proceedings of the 6th International Workshop on Testing Database Systems (DBTest'13), June 24, 2013, ACM, New York, USA., ISBN:978-1-4503-2151-8, pp: 1-5.

Anonymous, 2016. Testing big data using hadoop system. Atos, Bezons, France.

Das, D., D. Rout and P. Pradhani, 2015. Intelligent analytics assurance for big data systems: A framework for real time big data testing. Tata Consultancy Services, Mumbai, India.

Garg, N., S. Singla and S. Jangra, 2016. Challenges and techniques for testing of big data. Procedia Comput. Sci., 85: 940-948.

Hewitt, E., 2010. Cassandra: The Definitive Guide. O'Reilly Media, Inc., Sebastopol, California, USA., ISBN:978-1-449-39041-9, Pages: 305.

Kaur, P.D., A. Kaur and S. Kaur, 2015. Performance analysis in bigdata. Intl. J. Inf. Technol. Comput. Sci., 7: 55-61.

Li, N., A. Escalona, Y. Guo and J. Offutt, 2015. A scalable big data test framework. Proceedings of the 2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST'15), April 13-17, 2015, IEEE, Graz, Austria, ISBN:978-1-4799-7125-1, pp: 1-2.

Maheshwari, N. and P. Chaturvedi, 2012. Testing the giant: Testing big data. Infosys, Hyderabad, India.http://conference.qaiglobalservices.com/stc2013/PDFs/Prateek_Chaturvedi.pdf

Zikopoulos, P.C., C. Eaton, D. DeRoos, T. Deutsch and G. Lapis, 2011. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. McGraw-Hill Education, New York, USA., ISBN:978-0-07-179054-3, Pages: 142.