IMPROVING BIGDATA INTEGRITY BY DEFINING ACCURATE BASELINES

MRS. K PADMA PRIYA
Lecturer in Computer Science
Bhavan's Vivekananda College
Sainikpuri, Secunderabad,
Telangana, India.

MRS. N. SHARON ROSY
Lecturer in Computer Science
Bhavan's Vivekananda College
Sainikpuri, Secunderabad,
Telangana, India.

Abstract: Big data does not just mean a lot of information. It also refers to so-called unstructured data – sensor data, social media outpourings, video and images - that do not fit neatly into the rows and columns of most databases. In the near-future, Big Data could significantly improve government policymaking, social-welfare programs and scholarship. But having more data is no substitute for having high-quality data. One of the most fundamental challenges in the process of data integration is setting realistic expectations. The term data integration conjures a perfect coordination of diversified databases, software, equipment, and personnel into a smoothly functioning alliance, free of the persistent headaches that mark less comprehensive systems of information management.

In measuring what works and what doesn't often a lot of time goes into planning and collecting data, but inaccurate or incomplete data can mask the success of initiatives.

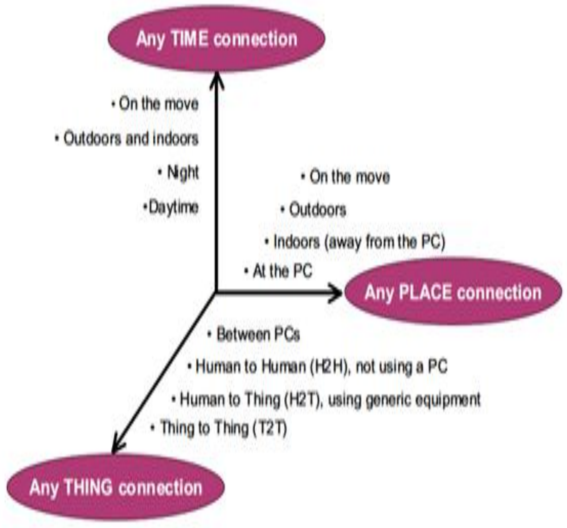
Only if baselines are accurate do they allow us to judge the success of whatever action we implement. This paper suggests the specific baselines that have to be defined for the data accuracy and integrity. The requirements analysis stage offers one of the best opportunities in the process to recognize and digest the full scope of complexity of the data integration task. Thorough attention to this analysis is possibly the most important ingredient in creating a system that will live to see adoption and maximum use.

Objective: By defining specific baselines to improve the Big Data Integrity and Accuracy.

Keywords: Data Integration, Data Accuracy, Requirements Analysis, Specific Baselines.

I INTRODUCTION

Extension of the current Internet and providing connection, communication, and inter-networking between devices and physical objects, or "Things," is a growing trend is referred to as the "Internet *of Things* ".



Data means raw facts. It includes textual content
, multimedia content like images, video and audio, on a variety of platforms such as enterprise, social media, and sensors.

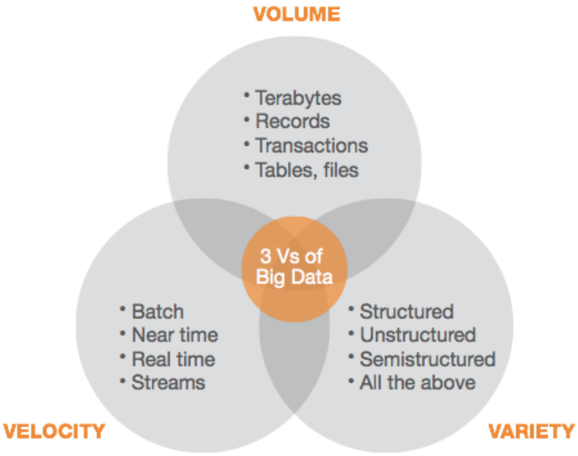
Textual Content can be categorized as follows:

Structured Data: Data which resides in the form of rows and columns is called structured data.
Eg: Relational databases and Spreadsheets.
Unstructured Data: Data which does not have a pre-defined structure is called Unstructured Data..
Eg: Free text on web, audio, videos, pdf file, text document etc.
Semi-structured Data: It is a form of structured data that does not confine with the formal structure of data models associated with relational databases or other forms of data tables.. Eg: XML
Big Data [1] refers to massive volume of unstructured data like sensor data, social media outpourings, video and images which do not fit neatly into the rows and columns of most databases.

Characteristics of Big Data:

Variety:

A single data quality metric[2] will not be applicable for the entire data and you would need to separately define data quality metrics for each data type because data is taken from different sources. Moreover, assessing and improving the data quality of unstructured or semi-structured data is way more tricky and complex than that of structured data. As data is taken from different sources they often have serious semantic differences. For example, "profit" can have widely varied definitions across the business units of an organization or external agencies. Thus, the fields with identical names may not mean the same thing which is made even worse by the lack of adequate and consistent meta-data.



Volume:

Big data implies enormous volumes of data. It used to be employees created data. Now that data is generated by machines, networks and human interaction on systems like social media the volume of data to be analyzed is massive.

Velocity:

Velocity is the measure of how fast the data is coming in from different sources. For example, Facebook has to handle a tsunami of photographs every day. It has to ingest it all, process it, file it, and somehow, later, be able to retrieve it.Because the flow of data is huge and continuous, this real-time data can help researchers and businesses to make valuable decisions which can provide strategic competitive analysis if you are able to handle the velocity.

II. IMPORTANCE OF INTEGRITY

Integrity:

Data governance must be a priority as there's no getting away from the fact that data flows from all across these days.[5] Neither will this information be arriving from all directions; it exists in various formats: everything from numbers and formulas to individual words and pieces of text. Traditionally, just as many tools would be used to deal with it as well. There might be some staff who would be relying on their own spreadsheets and word documents while others might depend more on data-competent team members who put their faith in advanced data visualization tools. This sort of discrepancy causes its own set of unique problems – and makes data useless.[6]

The danger of inaccurate data:

Accuracy of data can drive efficiency, profitability and growth. Inaccurate data, can cause real detriment to a business – and its bottom line. This can be from as simple and not very harmless circumstances like one person using data discovery while a second focuses instead on Excel or standard reports. The impact of slightly unrelated terminology which can make data out of sync, could be very high. The biggest consideration here is that these losses are totally avoidable, by ensuring that the data accuracy remains a main consideration.

Before developing a Big Data, a Database administrator should consider data movement and data retrieval requirements. For them, the big data environment means that there is no time (and usually insufficient disk or magnetic tape) for database reorganizing or database backups.

This means that the databases themselves must be defined with these considerations in mind:

- *) Bulk Insert or bulk table upload from a pre-sorted data.
- *) As the data is load-only with no update, Minimal free space for tables and indexes.
- *) Standard data extract processes and data transform processes with their scripts.
- *) Object naming standards should be good along with having a good metadata or a system catalog definition.

As per the definition of Big Data, it emphasizes not only on storing huge amount of data, but also on its retrieval process.[3]

III. FACTORS IMPROVING BIG DATA CONSISTENCY AND INTEGRITY

Here are the eight factors which might be useful for improving the consistency of Big Data:

1. **Current Data set size:**
   Current Data Set size is the set of data a system needs to address during normal operation. A complex system will have many distinct working sets, among which one or two of them usually dominate. The working set can be much smaller than the total set, in stream-like applications such as email or a news feed. People rarely access messages more than a few weeks old; they might as well be considered a different system. can focus on the rough size of the working set, for the initial analysis you as opposed to the detailed characteristics.
2. **Average transaction size:**
   It can be thought of as the Current Data set of a single transaction done by the system. How much data does the system have to access in order to complete a transaction? Downloading a photo and running a web search involves similar-sized answers .However, the amounts of data accessed in the background are very different.
3. **Request rate:**
   In a search engine you have 5 to 10 queries per each user over a period of minutes. A game may require multiple transactions per second per user. How many transactions are expected per hour / minute / second? Is there a peak hour, or is demand steady? In short, it is the combination of throughput and transaction size which governs most of the total data flow of the system.
4. **Update rate:**
   This is a measure of how frequent data is added, deleted, and edited. An email system has a high add rate, a low deletion rate, and an almost-zero edit rate. A useful way to gauge how much to worry about the update rate is to compare it to the read throughput. The growth rate of the data also ties into the working set size or retention policy. A 0.1% growth rate implies a three-year retention (365 * 3 is about 1,000), and vice-versa. A 1% rate implies 100 days.
5. **Consistency:**
   How swiftly does an update have to spread through the system? Consider, A Stock trading systems which has to

reconcile in milliseconds. A comments system is generally expected to declare new comments within a second or two, with frenetic work backstage to provide the illusion of immediate updation to the commenter. Consistency is a critical factor if the update rate is a significant portion of the request rate and also if propagating updates is especially important to the business, e.g. signing up into an online account or price and inventory changes.

6. **Locality:**
How much is the portion of the working set one request needs access to? How is that portion defined? On one extreme you have search engines: a user might want to query bits from anywhere in the system. In an email application, the user is guaranteed to access only their inbox which is a tiny well-defined part of the whole. Say for another instance, you may have a reduplicated storage for email attachments, leaving it victim to hot spots.

7. **Computation:**
What operations do you need to perform on the data? Can it be pre-computed? Are you doing intersections of large data set? How is the computation being done- Is the computation brought to the data, or the other way around? Why?

8. **Latency:**
How quickly are these transactions supposed to return success or failure? Users seem to be alright with a flight search or a credit card transaction taking several seconds. A web search has to return within a few hundred milliseconds. An API that outside systems depend on should return in 100 milliseconds or less. It’s equally important to think about the variance between the two. It is perhaps worse to answer 90% of queries in 0.1 seconds and the rest in 2 seconds, rather than all requests in 0.2 seconds.

IV BIG DATA RESOURCE CONSTRAINTS

A Database Administrator addresses the resource constraints in the big data environment with the help of balance of techniques. Storage constraints are addressed by reducing free space, eliminating database reorganization and reducing or eliminating backups. Extended run times for bulk loads are reduced with the help of intelligent data partitioning. Traditional tools which were developed especially for regular or small data and their older architecture cannot process Big Data efficiently. No SQL data sources are not so easy to handle and thus have no solution.[7]

When one of the new technologies like Hadoop is involved, the validation process becomes more complicated as mentioned below. Data Integrity will be maintained by implementing a pre-Hadoop validation, Hadoop Map-Reduce validation and a Post-Hadoop validation.

Big Analytics:
The value of analytics (also termed as business intelligence, or BI) implies the need to create and maintain a big and huge data environment.[4] Imagine a huge data store with simultaneous mass loading and querying of the data. In those cases, the DBA should be aware of the data availability requirements and its performance tuning.

Long-running extract-transform-load (ETL) jobs can lock important data during implementation. To make data more available, the DBA can use an active and an inactive table technique. Each critical structure is defined by two tables, or two sections of the same table. One of them is designated as active, while the other becomes inactive. Querying is directed towards the active section of the table while data loading is executed against the inactive partition. After the load is complete, the table definitions are switched.

V TOOLS USED FOR IMPROVING DATA PERFORMANCE IN BIG DATA

i) HADOOP:

Hadoop is a Java-based programming; open source framework that supports the processing and storage of extremely large data sets in a distributed computing environment. It was developed as a part of the Apache project sponsored by the Apache Software Foundation.

Hadoop advantages and disadvantages:-

For humungous data sets:
Not in all the cases is Hadoop utilized for every organization which feels their data is huge. Along with huge data handling capacities, Hadoop also comprises limitations on the programming of applications and to the fasten the results are obtained. Therefore, organizations having comparatively less data(in MBs or GBs) are recommended to use Excel, SQL for getting faster and better results. Whereas, when data gets bigger such as Terabytes or even Petabytes, then Hadoop is the most efficient technology to be applied as its enormous scalability will save time & cost.

Data Mixing:

Hadoop is best to be applied when an organization is having data diversity to be processed. The most significant advantage HDFS (Hadoop Distributed File System) has is that it is very flexible regarding data types. Hadoop can handle it in the best way possible no matter the raw data is structured as in ERP system, semi-structured as in XML or completely unstructured i.e. videos, audios.

Specialized programming skills: Hadoop is been driven to be converted into a general purpose computing framework, however, if programmers have mastered the skill of Java coding then it is best to utilize it as all the Hadoop applications are developed in Java as on date. Therefore, This is also the reason that if a professional is having skills of Java coding along with science of handling data, he/she will be high in demand by organizations.

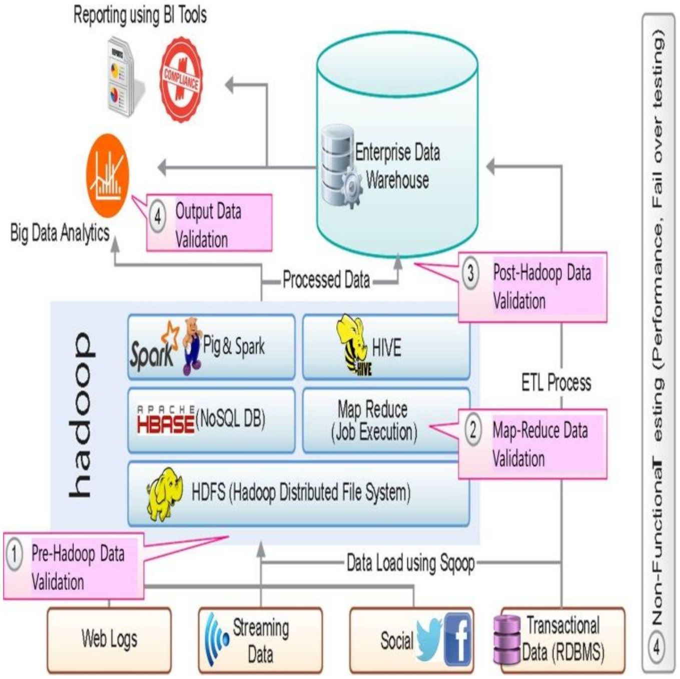
Future vision of Hadoop utilization: Even though an organization does not have a data huge enough to utilize Hadoop it will have to visualize itself using it in the near future. It will be beneficial for it to start experimenting with Hadoop and prepare working with the IT professionals to work with it comfortably.

Optimum data utilization: In some cases, there are chances that some potential data has to be thrown aside as it might be fatal to archive it. To retain this data and utilize it in the best possible way Hadoop can be used as it can handle data as huge as in Petabytes.

Disadvantages of Hadoop

Trade-off for Time: Hadoop is no doubt the best to be utilized to handle huge data(as large as Petabytes). Only improvement needed to this tool is the time taken to produce the result. It is therefore recommended to utilize Excel or SQL or any other tool to process smaller data up to Gigabytes.

Intense optimization for queries: To get the best out of Hadoop, a considerable investment is required, in order to optimize the queries. In order for better implementation of Hadoop, it needs to embed the feature of optimizing the queries. If we carry out the same process with software-based optimizers in combination with conventional data warehouse platforms, better & economical results can be obtained.


Reporting using BI Tools
Enterprise Data Warehouse
Big Data Analytics
4 Output Data Validation
Processed Data
3 Post-Hadoop Data Validation
hadoop
Spark Pig & Spark
HIVE
ETL Process
APACHE HBASE (NoSQL DB)
Map Reduce (Job Execution)
2 Map-Reduce Data Validation
HDFS (Hadoop Distributed File System)
1 Pre-Hadoop Data Validation
Data Load using Sqoop
Web Logs
Streaming Data
Social
Transactional Data (RDBMS)
4 Non-Functional Testing (Performance, Fail over testing)

Inability to interactive access to random data: . One of the most significant cons of Hadoop is that it has limitations which restrict it to access & serve interactive queries for random data in its batch-oriented MapReduce.

Crucial data storage: Another notable limitation of Hadoop is that it is not efficient in storing sensitive and crucial data. Hadoop comprises of basic data and access security, hence, there is a risk of accidentally losing crucially identifiable information

Data warehouse replacement: There has been a notion building in the industry that Hadoop can totally replace the current traditional data warehouse platforms. This is not the complete truth as Hadoop can complement data warehouse platforms but cannot replace it.

ii) DATABUCK

DataBuck: It is an autonomous, self-learning Data Matching tool which is a Big Data Quality validation tool. Machine Learning is used to simplify the elaborate and complex validations.

Advantages of Databuck:

- Traditional data-sampling techniques are no longer viable options for Big Data Integrity Validation.
- DataBuck can be used for a faster “**Data Health Check**” for constantly monitoring data discrepancies between different IT systems.
- Easy connectivity and validation of traditional RDBMS, NoSQL databases, Hadoop and Cloud components.

iii) APACHE SPARK

Spark was developed to improve the performance of data processing. It doesn’t just help in a little more performance, rather boosts up to a 10x-100x improvement in performance. In general, Spark can improve performance across the world for most workloads, while radically improving it for a subset of workloads. Spark’s in-memory approach is especially advantageous for machine learning algorithms that are iterative in nature. Apache Spark consists of a Spark core and a set of libraries similar to those available for Hadoop. The core is the distributed execution engine and a set of languages (Java, Scala, Python and R) which are supported for distributed application development. Spark offers an unprecedented ease of development and ability to combine all of the library seamlessly into the same application. Unlike Hadoop, Spark can be run in a variety of modes on a variety of platforms, each offering a different method for managing the data cluster and its resources such as standalone, Apache Mesos, Hadoop YARN, and in the cloud.

COMPARISON BETWEEN HADOOP AND APACHE SPARK:

How do they work?

Both Hadoop and Apache Spark are the frameworks of big-data, however they don't really serve the same purposes.

Hadoop is a distributed data infrastructure which distributes massive data collected across multiple nodes within a cluster of commodity servers. This ultimately means you don't need to buy and maintain expensive custom hardware. It also indexes and keeps track of the data, enabling big-data processing and analytics far more effectively than which was possible previously. Spark, on the other hand, is a data-processing tool that operates on those distributed data collections; however, it doesn't do distributed storage.

How difficult are the two tools to work with?

As Spark uses tons of high level operators, it is very easy to work with. Using MapReduce the developers need to hand code each and every operation which makes it difficult to work with.

Can I use one without the other?

Along with having a storage component known as the Hadoop Distributed File System, Hadoop also has a processing component called MapReduce, which doesn't need Spark to get the processing done. Conversely, Spark can also be used without Hadoop. Although, Spark does not have its own file management system, it needs to be integrated with one of them, if not HDFS, then any other cloud-based data platform.

Latency?

Spark provides a low latency computing whereas Hadoop(Map Reduce) is a high latency computing framework.

Which of them is speedier?

Spark is generally a lot faster than MapReduce (Hadoop) because of the way it processes data. While MapReduce operates in steps, Spark processes on the whole set of data in one swoop.

Can failure be recovered?

Hadoop was developed to be naturally resilient to system faults or failures because data are written to disk after every operation. Although, Spark has similar built-in resiliency by virtue of the fact that its data objects are stored in something called resilient distributed datasets which are distributed across the data cluster. These data objects can be stored in memory or on disks.[8]

CONCLUSION:

This paper proposes eight factors which can help in improving the consistency of Big Data. It also gives an overview of the various Data Analytics, Validation and Performance tools which are currently being used in Big Data. These might help in selecting the appropriate tool for retrieving different types of data from various resources.

REFERENCES:

1) Advancing Discovery in Science and Engineering. Computing Community Consortium. Spring 2011. http://csjournals.com/IJCSC/PDF7-2/13.%20JP.pdf

2) Advancing Personalized Education. Computing Community Consortium. Spring 2011.

3) Following the Breadcrumbs to Big Data Gold. Yuki Noguchi. National Public Radio, Nov. 29, 2011. http://www.npr.org/2011/11/29/142521910/the-digital-breadcrumbs-that-lead-to-bigdata

4) The Search for Analysts to Make Sense of Big Data. Yuki Noguchi. National Public Radio, Nov. 30, 2011. http://www.npr.org/2011/11/30/142893065/the-search-for-analysts-to-make-sense-of-big-dataThe Age of Big Data. Steve Lohr. New York Times, Feb 11, 2012. http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html

5) Maintaining Data Integrity in the Analytics 'Wild West', CIO JOURNAL, http://deloitte.wsj.com/cio/2014/10/14/maintaining-data-integrity-in-the-analytics-wild-west/

6) Propelling the Future of Big Data , IBM BIG DATA HUB http://www.ibmbigdatahub.com/blog/propelling-future-big-data-and-data-science

7) Comparison of MapReduce and Spark programming Frameworks for Big Data Analytics by Jai Prakash Verma, Atul Patel.