Big Data Analysis and its Comparison with RDBMS

Ankita Bhalla
M.Tech (CSE)
GNDU Amritsar

Richa Arora
M.Tech (CSE)
GNDU Amritsar

Abstract

Today the term big data draws a lot of attention. Big Data is a new frontier in IT where data sets are becoming enormous that they are almost impossible to manage using traditional database management tools. Data types and content are getting more complicated; volume is going up and serious. Big Data includes structured and unstructured data coming from tweets, social networking sites etc. Traditional systems, and the data management techniques associated withthem, have failed to scale to Big Data. NoSQL act as a paradigm shift for Big Data. Various characteristics of big data and Lambda Architecture for handling the big data are discussed.

Keywords: Big Data, Lambda Architecture, RDBMS

1. Introduction

Today's organizations are facinghuge challenges related to data. The rapid growth of data continues, due to the increasing use of new devices and sensors, and rapidly declining hardware cost. Huge amount of data is required in every sector. More organizations now store terabytes and even petabytes of data. Data complexity is also increasing because of structured data in relational format and unstructured data such as Word or PDF files, images, videos and geo-spatial data. It is estimated that 80% of data captured is in unstructured format. Customers are also challenged by the velocity and value of data.

2. Big Data

Data is extracted from data sets and stored in data warehouses for analysis. This collection of large and complex data sets is called BIG DATA. As of today data is in yottabytes i.e. in 10^{24} bytes. Big data is very popular term for unstructured and structured information and used to describe the availability and exponential growth. Big Data is key basis of competition and growth.

Big Data is a new frontier in IT where data sets are becoming enormous that they are almost impossible to manage using traditional database management tools. Data types and content are getting more complicated; volume is going upand serious.

Depending upon the capabilities of the organizations and the various applications that require large data sets to process and analyze the information of various domains, big data varies accordingly. The size of large data sets is going beyond limits i.e. in yottabytes so the currently used software tools are unable to capture process and manage the data within tolerable elapsed time. Because of this difficulty various big data tools are being developed to handle various aspects of large amount of data. The various tools of big data includes Big Science, Web Logs, RFID, Social networks and Social data.

Defination: Big data is where the data volume, acquisition velocity, ordata representation limits the ability to perform effectiveanalysis using traditional relational approaches or requiresthe use of significant horizontal scaling for efficientprocessing.

Examples of big data

- RFID (radio frequency ID) systems generate up to 1,000 times the data of conventional bar code systems.
- In every second, around 10,000 payment card transactions are made around the world.
- More than 1 million customer transactions are handled by Walmart in an hour.
- 340 million tweets are sent per day on twitter.
- More than 901 million active users generate social interaction data on facebook.

Need of Big Data

- Data explosion, driven by declining hardware cost and new data sources
- Greater variety of data - customers need to analyze both relational and non-relational data
- Over 80% of data captured is unstructured
- Increased velocity of data requiring organizations to respond quickly to rapidly changing data
- The need to explore data interactively with few preconceived questions.[1]

In 2009 it is 800,000 petabytes. But it is estimated that size of data will become 35 zettabytes in 2020. It means data and content has been increased 44 times over the decade. Majority of data growth is being driven by unstructured data and billions of large objects. The data which is coming from mobility devices, social networking sites, entertainment constitutes 80% of the world's unstructured data [2]

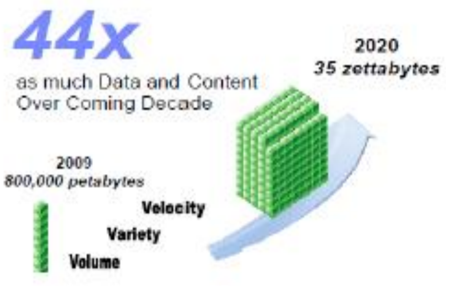

Figure 1. Growth of data[2]

3. Challenges of Big Data

There are four key challenges that define big data which are described as follows:

- **Velocity**
  Velocity defines the speed at which data is coming. For example: Even at 140 characters per tweet, the high velocity (or frequency) of Twitter data ensures large volumes (over 8 TB per day).
- **Variety**
  Variety means the different types of data like audio, textual, video etc. For example: non-traditional data formats exhibit a dizzying rate of change. As new services are added, new sensors deployed, or new marketing campaigns executed, new data types are needed to capture the resultant information.
- **Complexity**
  As data is coming from various sources so it is very complex to manage it.Various sources can be databases, internet, journals, archives, reports etc.
- **Volume**
  Volume is related to the amount of data. For example: With more than 25,000 airline flights per day, the daily volume of just this single data source runs into the Petabytes.

These all are increasing at very fast rate due to social computing, context aware computing, social networking etc. It is estimated that in every year around 1700 TB of data is generated by various personal location services like Navigation Devices, Navigation Apps on Phones, Geo targeted apps and others.

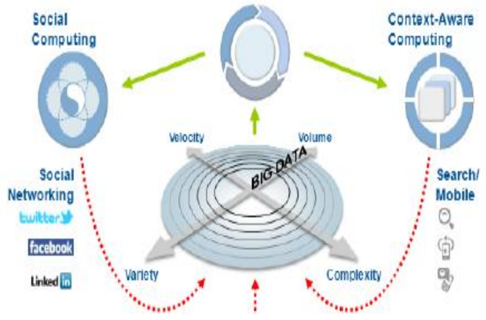

Figure 2. Challenges of big data [2]

4. Importance of Big Data

Smart phones and other GPS devices offers advertisers an opportunity to target consumers when they are in close proximity to a store, a coffee shop or a restaurant. This opens up new revenue for service providers and offers many businesses a chance to target new customers.

Retailers usually know who buys their products. Use of social media and web log files from their ecommerce sites can help them understand who didn’t buy and why they chose not to, information not available to them today. This can enable much more effective micro customer segmentation and targeted marketing campaigns, as well as improve supply chain efficiencies.

Finally, social media sites like facebook and LinkedIn simply wouldn’t exist without big data. [3] Their business model requires a personalized experience on the web, which can only be delivered by capturing and using all the available data about a user or member.

5. Problems with Traditional Architecture

In order to manage large amount of data using traditional architecture i.e. RDBMS various problems and complexities arises that are discussed as below:

- **Fault-tolerance:** In traditional architecture fault tolerance is hard to handle. It is very complex to keep the applications working in the failure conditions. It is managed manually such as creating the replicas of data.
- **Lack of human fault tolerance:** As system gets more complex, the probability of encountering the errors also increases. In case of big data, human fault tolerance is not optional because big data adds so many complexities for building the applications.
- **Maintenance:** Maintenance is enormous amount of work in traditional architecture. Horizontal and vertical portioning of database is time consuming and error prone. Database is not self aware of its distributed nature and does not manage the partitioning itself.

6. Lambda Architecture

The Lambda Architecture solves the problem of computing arbitrary functions on arbitrary data in real time by decomposing the problem into three layers: the batch layer, the serving layer, and the speed layer.

Everything starts from the "query = function (all data)" equation. Run your query functions on the fly on the complete dataset to get the results. Instead of computing the query on the fly, read the results from the precomputed view using precomputed query function “the batch view”.

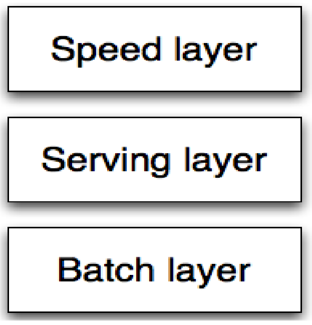

Figure 3. Lambda Architecture [4]

6.1 Batch Layer

The portion of the Lambda Architecture that precomputes the batch views is called the "batch layer". This layer is most complex layer in lambda architecture. The batch layer stores the master copy of the dataset and precomputes batch views on that master dataset. The master dataset is a very large list of records. The batch layer requires two things to do its job: store an immutable, constantly growing master dataset, and compute arbitrary functions on that dataset. [4]

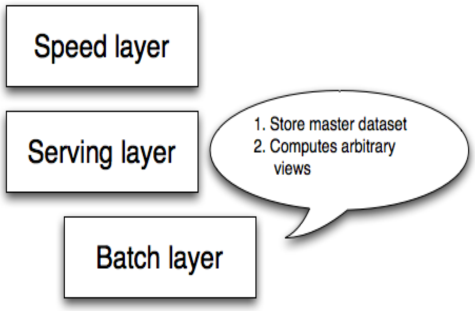

Figure4. Batch layer [4]

6.2 Serving Layer

The batch layer emits batch views as the result of its functions. The serving layer loads the batch views that can be queried. The serving layer indexes the batch view and loads it up so it can be efficientlyqueries. The serving layer is a specializeddistributed database that loads in a batch views, makes them queryable, andcontinuously swaps in new versions of a batch view as they're computed by thebatch layer. Since the batch layer usually takes at least a few hours to do an update,the serving layer is updated every few hours.

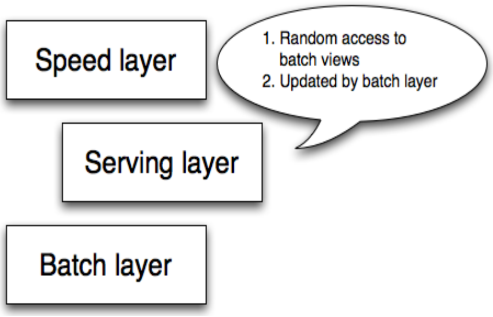

Figure5. Serving layer [4]

A serving layer database only requires batch updates and random reads. Most notably, it does not need to support random writes that is why it is less complex. This simplicity makes them robust, easy to configure, and easy to operate. [4]

6.3 Speed Layer

Speed layer deals with fully real time data systems. In this layer, arbitrary functions are computed on arbitrary data in real time. Speed layer looks similar to batch layer as both produces views but there are some key differences like in order to achieve the fastest latencies possible, the speed layerdoesn't look at all the new data at once. Instead, it updates the realtime view as itreceives new data instead of recomputing them like the batch layer does. This iscalled "incremental updates" as opposed to "recomputation updates". Another bigdifference is that the speed layer only produces views on recent data, whereas thebatch layer produces views on the entire dataset. [4]

The speed layer requires databases that support random reads and randomwrites. Therefore it is more complex than batch and serving layer in terms of implementations and operations.

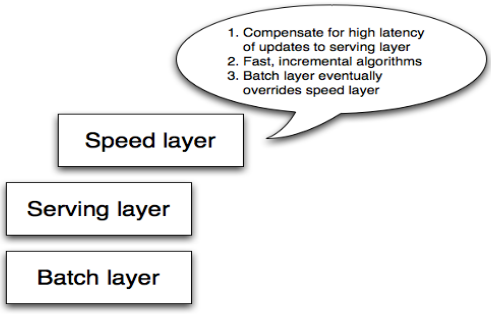
1. Compensate for high latency of updates to serving layer
2. Fast, incremental algorithms
3. Batch layer eventually overrides speed layer
Speed layer
Serving layer
Batch layer

Figure 6. Speed Layer [4]

7. Comparison of RDBMS and Big Data

Characteristics	Traditional RDBMS	Big Data
Data Size	Data size is in gigabytes.	Data size is in yottabytes
Latency	Has low latency	Has high latency
Language	Uses SQL	Uses NoSQL, UQL
Type of data	Structured like Transactional, Corporate	Structured, Semi Structured and Unstructured like Internet, Derivative
Schema/Structure	Has static schema	Has dynamic schema
Integrity	Has high integrity	Has Low integrity
Scalability	Low	High
Fault Tolerance	Fault tolerance is hard. Lack of human fault-tolerance	Fault tolerance for hardware/software failures. It is human fault tolerant.
Extensibility	Less extensible.	Highly extensible. It allows functionality to be added with a minimal development cost.
Maintenance	Maintenance is an enormous amount of work	Requires Minimal Maintenance.
Robustness	Not robust	Robust
Opportunity	Very small growth	Massive growth
Impact	Analyze Existing Businesses	Create New Businesses
Mode	Data collection	Data analysis

8. Properties of Big Data System

Low latency reads and updates

Majority of the applications requires reads and updates to be satisfied with low latency typically between a few milliseconds to a few hundred milliseconds without compromising the robustness of the system.

Scalable

Scalability is the ability to maintain performance even if data and/or load by adding resources to the system increasing sharply. The Lambda Architecture is horizontally scalable across all layers.

General

A general system can support a wide range of applications. The Lambda Architecture generalizes various applications such as financial management systems, social media analytics, scientific applications, and social networking.

Extensible

Extensible systems allow functionality to be added with a minimal development cost. In order to implement a new feature or change to an existing feature requires a migration of old data into a new format.

Minimal maintenance

An important part of minimizing maintenance is choosing components that have as small an implementation complexity as possible. In the Lambda Architecture complexity is pushed out of the core components into the components which give the temporary results.

Robust and fault-tolerant

Big Data makes the system robust by avoiding various complexities in the system. Big Data are self-aware of their distributed nature. So things like partitioning and replication are handled by itself. Data will automatically rebalance onto that new machine. Human fault tolerance can't be ignored so there is a much stronger human fault-tolerance guarantee in Big Data than in a traditional system. [4]

9. Conclusions

Nowadays we are dealing with large amount of data which comes from various resources like social networking sites, journals, blog posts, tweets, archives etc. It is very difficult and complex to manage such large amount of data. Traditional RDBMS approach is not relevant for such a huge amount of data. Big data is like super set of data warehouse. To derive real business value from big data, there is a need of right tools to capture and organize a wide variety of data types from different sources. In this paper we also discussed about Lambda Architecture which gives less latency than Hadoop Architecture.

10. References

[1] Microsoft Corporation: Microsoft Big Data, 2011.

[2] Anil Vasudeva,IMEX Research: NextGen Infrastructure for Big Data, 2012.

[3] Oracle: Big Data for Enterprise, 2012.

[4] Nathan Marz, James Warren: Manning Publications, Big Data Principles and best practices of scalable realtime data systems, Version 7, 2012