Classification of data, Characteristics, Evolution and definition of Big data, What is Big data, Why Big data, Traditional Business Intelligence Vs Big Data,Typical data warehouse and Hadoop environment. Big Data Analytics: What is Big data Analytics, Classification of Analytics, Importance of Big Data Analytics, Technologies used in Big data Environments, Few Top Analytical Tools , NoSQL, Hadoop.

What is DATA?

Data is a raw fact and is defined as individual facts, such as numbers, words, measurements, observations or just descriptions of things.

For example, data might include individual prices, weights, addresses, ages, names, temperatures, dates, or distances.

There are two main types of data:

1. **Quantitative data** is provided in numerical form, like the weight, volume, or cost of an item.
2. **Qualitative data** is descriptive, but non-numerical, like the name, gender, or eye color of a person.

Characteristics of Data

The key characteristics of the data are:

1. **Composition:** The composition of data deals with the structure of data, that is, the sources of data, the granularity, the types, and the nature of data as to whether it is static or real-time streaming.
2. **Condition:** The condition of data deals with the state of data, that is, "Can one use this data as is for analysis?" or "Does it require cleansing for further enhancement and enrichment?“

3. **Context:** The context of data deals with "Where has this data been generated?" "Why was this data generated?" "How sensitive is this data?" "What are the events associated with this data?" and so on.

The other following are six other characteristics of data which are discussed below:

1. Accuracy
2. Validity
3. Reliability
4. Timeliness
5. Relevance
6. Completeness

1. Accuracy

Data should be sufficiently accurate for the intended use and should be captured only once, although it may have multiple uses. Data should be captured at the point of activity .

2. Validity

Data should be recorded and used in compliance with relevant requirements, including the correct application of any rules or definitions. This will ensure consistency between periods and with similar organizations, measuring what is intended to be measured.

3. Reliability

Data should reflect stable and consistent data collection processes across collection points and over time. Progress toward performance targets should reflect real changes rather than variations in data collection approaches or methods. Source data is clearly identified and readily available from manual, automated, or other systems and records.

4. Timeliness

Data should be captured as quickly as possible after the event or activity and must be available for the intended use within a reasonable time period. Data must be available quickly and frequently enough to support information needs and to influence service or management decisions.

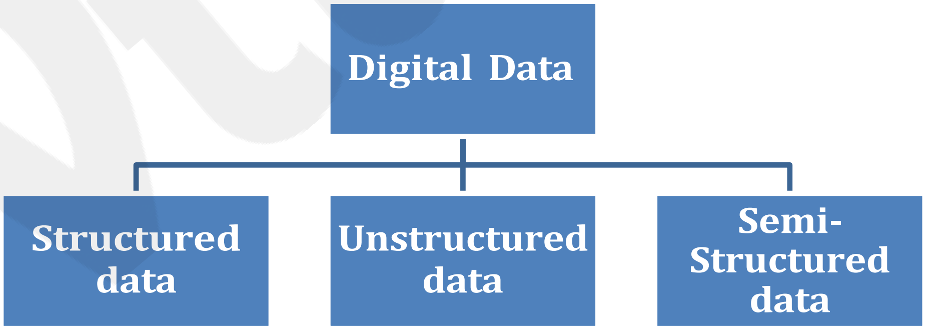
5. Relevance

Data captured should be relevant to the purposes for which it is to be used. This will require a periodic review of requirements to reflect changing needs.

6. Completeness

Data requirements should be clearly specified based on the information needs of the organization and data collection processes matched to these requirements.

Classification of data



1. Structured Data:

- Structured data refers to any data that resides in a fixed field within a record or file.
- Having a particular Data Model.
- Meaningful data.
- Data arranged in a row and column.
- Structured data has the advantage of being easily entered, stored, queried and analysed.
- E.g.: Relational Data Base, Spread sheets.
- Structured data is often managed using Structured Query Language (SQL)

Sources of Structured Data:

- SQL Databases
- Spreadsheets such as Excel
- OLTP Systems
- Online forms
- Sensors such as GPS or RFID tags
- Network and Web server logs
- Medical devices

Ease of working with structured data:

1. **Insert/update/delete:** The Data Manipulation Language (DML) operations provide the required ease with data input, storage, access, process, analysis, etc.

2. **Security:** Information security is ensured through strong encryption and tokenization, protecting data throughout its lifecycle. Access control measures ensure only authorized individuals can decrypt and view sensitive information, maintaining compliance.

3. **Indexing:** An index is a data structure that speeds up the data retrieval operations (primarily the SELECT DML statement) at the cost of additional writes and storage space, but the benefits that ensue in search operation are worth the additional writes and storage space.

4. **Scalability:** The storage and processing capabilities of the traditional RDBMS can be easily scaled up by increasing the horsepower of the database server (increasing the primary and secondary or peripheral storage capacity, processing capacity of the processor, etc.).

5. **Transaction processing:** RDBMS has support for Atomicity, Consistency, Isolation, and Durability (ACID) properties of transaction.

Atomicity: A transaction is atomic, means that either it happens in its entirety or none of it at all.

Consistency: The database moves from one consistent state to another consistent state. In other words, if the same piece of information is stored at two or more places, they are in complete agreement.

Isolation: The resource allocation to the transaction happens such that the transaction gets the impression that it is the only transaction happening in isolation.

Durability: All changes made to the database during a transaction are permanent and that accounts for the durability of the transaction.

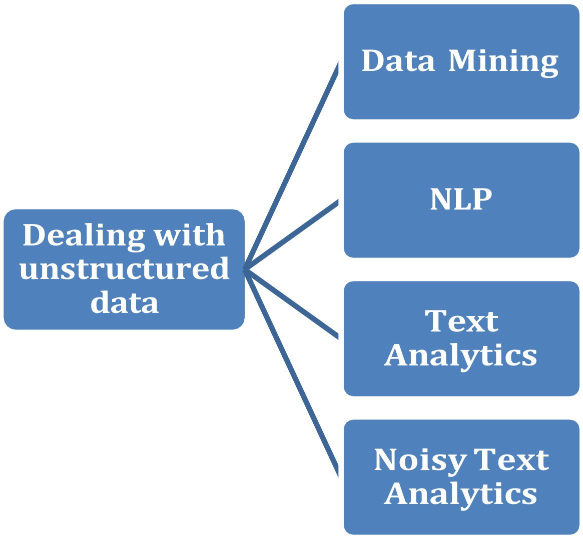
2. Unstructured data:

- Unstructured data can not readily classify and fit into a neat box
- Also called unclassified data.
- Which does not confirm to any data model.
- Business rules are not applied.
- Indexing is not required.
- E.g.: photos and graphic images, videos, streaming instrument data, webpages, Pdf files, PowerPoint presentations, emails, blog entries, wikis and word processing documents

Sources of Unstructured Data:

- Web pages
- Images (JPEG, GIF, PNG, etc.)
- Videos
- Memos
- Reports
- Word documents and PowerPoint presentations
- Surveys

How to deal with unstructured data?



1. **Data mining:** First, we deal with large data sets. Second, we use methods at the intersection of artificial intelligence, machine learning, statistics, and database systems to unearth consistent patterns in large data sets and/or systematic relationships between variables. It is the analysis step of the "knowledge discovery in databases" process.

Few popular data mining algorithms are as follows:

- **Association rule mining:** It is also called "market basket analysis" or "affinity analysis". It is used to determine "What goes with what?" It is about when you buy a product, what is the other product that you are likely to purchase with it. For example, if you pick up bread from the grocery, are you likely to pick eggs or cheese to go with it.

- **Regression analysis**: It helps to predict the relationship between two variables. The variable whose value needs to be predicted is called the dependent variable and the variables which are used to predict the value are referred to as the independent variables.

- **Collaborative filtering:** It is about predicting a user's preference or preferences based on the preferences of a group of users.

2. Text analytics or text mining: Compared to the structured data stored in relational databases, text is largely unstructured, amorphous, and difficult to deal with algorithmically. Text mining is the process of extracting high quality and meaningful information (through devising of patterns and trends by means of statistical pattern learning) from text. It includes tasks such as text categorization, text clustering, sentiment analysis, concept/entity extraction, etc.

3. Natural language processing (NLP): It is related to the area of human computer interaction. It is about enabling computers to understand human or natural language input.

4. Noisy text analytics:

It is the process of extracting structured or semi-structured information from noisy unstructured data such as chats, blogs, wikis, emails, message boards, text messages, etc. The noisy unstructured data usually comprises one or more of the following: Spelling mistakes, abbreviations, acronyms, non standard words, missing punctuation, missing letter case, filler words such as "uh", "um", etc.

3. Semi-structured data:

- Self-describing data.
- Metadata (Data about data).
- Also called quiz data: data in between structured and semi structured.
- It is a type of structured data but not followed data model.
- Data which does not have rigid structure.
- E.g.: E-mails, word processing software.

- XML and other markup language are often used to manage semi structured data.

Sources of semi-structured Data:

- E-mails
- XML and other markup languages
- Binary executables
- TCP/IP packets
- Zipped files
- Integration of data from different sources
- Web pages

It has the following features:

1. It does not conform to the data models that one typically associates with relational databases or any other form of data tables.

2. It uses tags to segregate semantic elements.

3. Tags are also used to enforce hierarchies of records and fields within data.

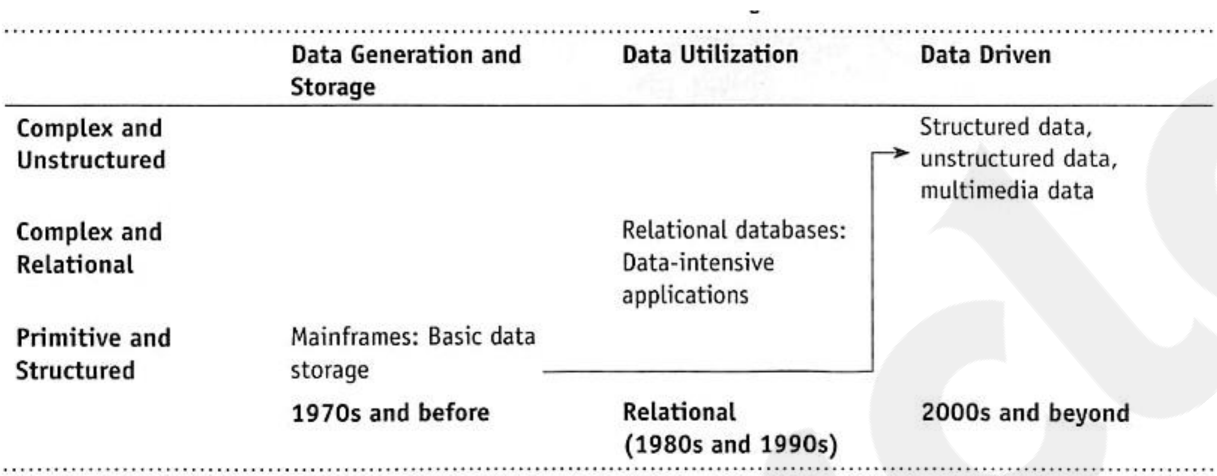
4. There is no separation between the data and the schema. The amount of structure used is dictated by the purpose at hand.

5. In semi-structured data, entities belonging to the same class and also grouped together need not necessarily have the same set of attributes. And if at all, they have the same set of attributes, the order of attributes may not be similar and for all practical purposes it is not important as well.

Evolution of Big Data:

1970s and before was the era of mainframes. The data was essentially primitive and structured. Relational databases evolved in 1980s and 1990s. The era was of data intensive applications. The World Wide Web (WWW) and the Internet of

Things (IoT) have led to an onslaught of structured, unstructured, and multimedia data.



	Data Generation and Storage	Data Utilization	Data Driven
Complex and Unstructured			Structured data, unstructured data, multimedia data
Complex and Relational		Relational databases: Data-intensive applications	
Primitive and Structured	Mainframes: Basic data storage		
	1970s and before	Relational (1980s and 1990s)	2000s and beyond

Definition of Big data

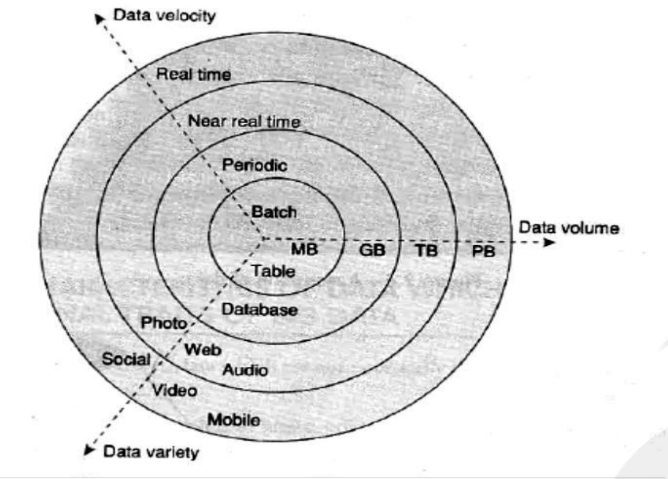
Big data is high-volume, high-velocity, and high-variety information assets that demand cost effective, innovative forms of information processing for enhanced insight and decision making.

or

Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size.

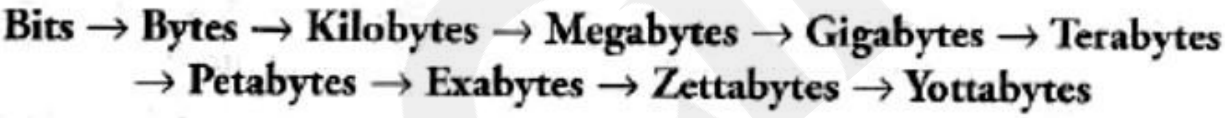
What is Big data?

Big data is data that is big in volume, velocity and variety.



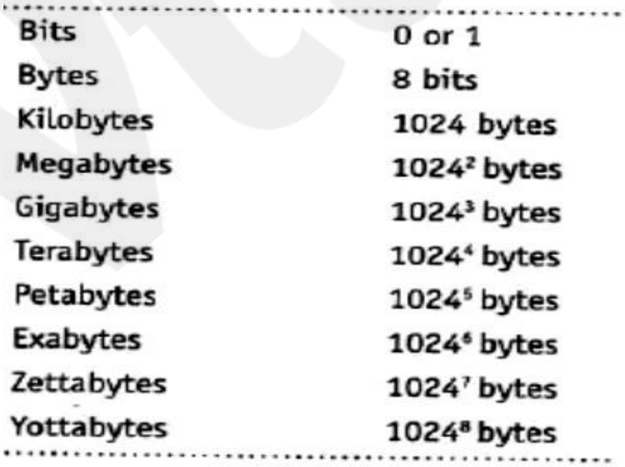
1. Volume:

We have seen the growth of data from bits to bytes to petabytes and exabytes.

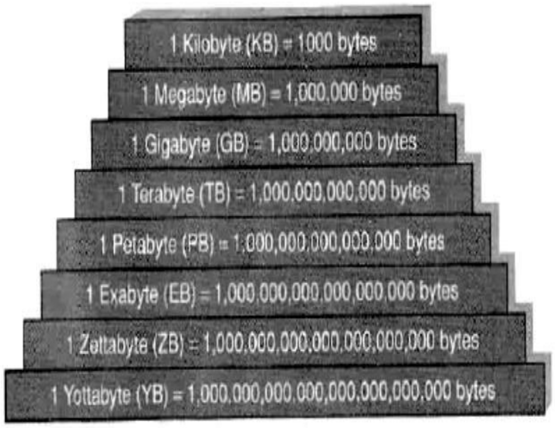


Bits → Bytes → Kilobytes → Megabytes → Gigabytes → Terabytes → Petabytes → Exabytes → Zettabytes → Yottabytes

The name Big Data itself is related to an enormous size. Big Data is a vast 'volume' of data generated from many sources daily, such as business processes, machines, social media platforms, networks, human interactions, and many more.

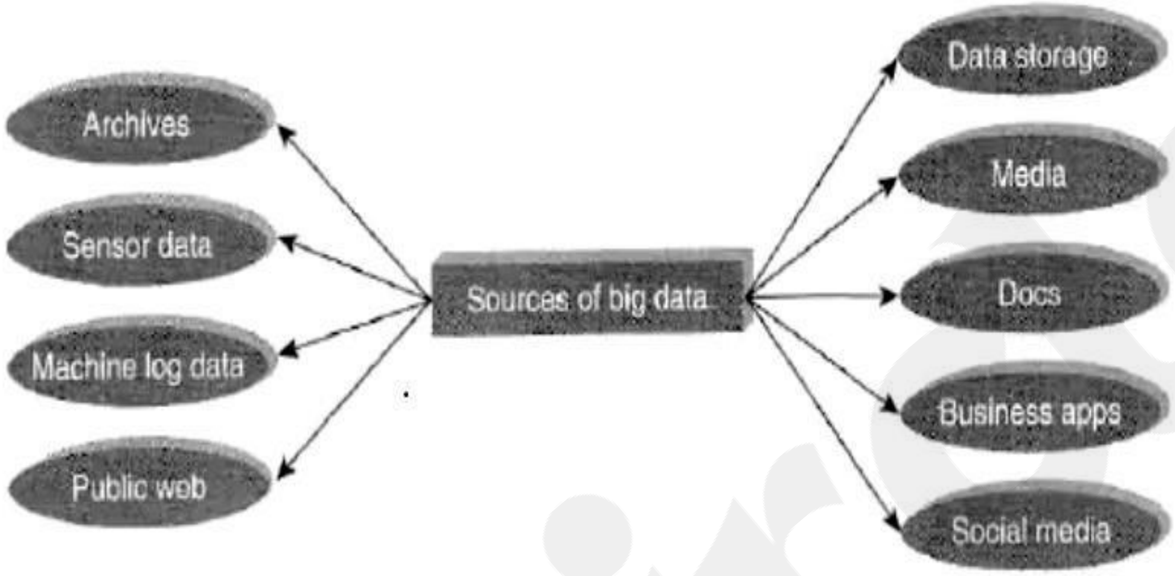


Bits	0 or 1
Bytes	8 bits
Kilobytes	1024 bytes
Megabytes	1024^2 bytes
Gigabytes	1024^3 bytes
Terabytes	1024^4 bytes
Petabytes	1024^5 bytes
Exabytes	1024^6 bytes
Zettabytes	1024^7 bytes
Yottabytes	1024^8 bytes



1 Kilobyte (KB) = 1000 bytes
1 Megabyte (MB) = 1,000,000 bytes
1 Gigabyte (GB) = 1,000,000,000 bytes
1 Terabyte (TB) = 1,000,000,000,000 bytes
1 Petabyte (PB) = 1,000,000,000,000,000 bytes
1 Exabyte (EB) = 1,000,000,000,000,000,000 bytes
1 Zettabyte (ZB) = 1,000,000,000,000,000,000,000 bytes
1 Yottabyte (YB) = 1,000,000,000,000,000,000,000,000 bytes

Where does the data get generated?



1. Typical internal data sources:

Data present within an organization's firewall. It is as follows:

Data storage: File systems, SQL (RDBMSS - Oracle, MS SQL Server, DB2, MySQL, PostgreSQL, etc.), NoSQL (MongoDB, Cassandra, etc.), and so on.

Archives: Archives of scanned documents, paper archives, customer correspondence records, patients' health records, students' admission records, students' assessment records, and so on.

2. External data sources: Data residing outside an organization's firewall like Public Web: Wikipedia, weather, regulatory, compliance, census, etc.

3. Both (internal + external data sources)

- Sensor data: Car sensors, smart electric meters, office buildings, air conditioning units, refrigerators, and so on.

- Machine log data: Event logs, application logs, Business process logs, audit logs, click stream data, etc.

- Social media: Twitter, blogs, Facebook, LinkedIn, YouTube, Instagram, etc.

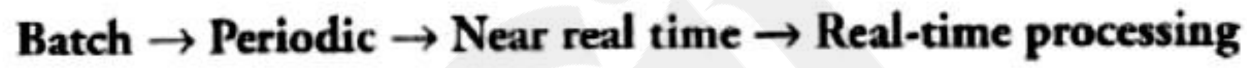
- Business apps: ERP, CRM, HR, Google Docs, and so on.

- Media: Audio, Video, Image, Podcast, etc.

- Docs: Comma separated value (CSV), Word Documents, PDF, XLS, PPT, and so on.

2. Velocity:

We have moved from batch processing to real time processing.



Batch → Periodic → Near real time → Real-time processing

3. Variety:

Variety deals with a wide range of data types and sources of data like Structured data, semi-structured data and unstructured data.

Other characteristics of data

1. Veracity and Validity:

Veracity refers to biases, noise and abnormality in data.

Validity refers to accuracy and correctness of the data.

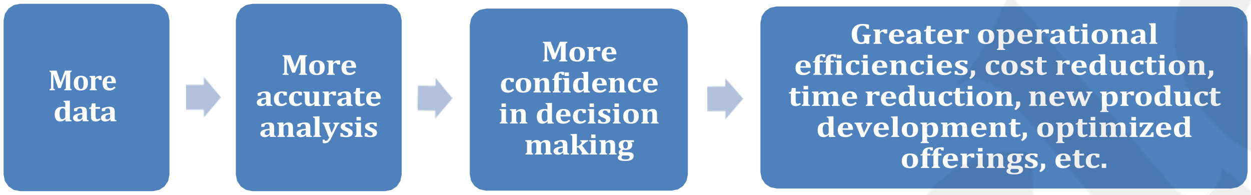
2. Volatility:

Volatility of data deals with, how long is the data valid and how long should it be stored. There is some data that is required for long-term decisions and remains valid for longer periods of time. However, there are also pieces of data that quickly become obsolete minutes after their generation.

3. Variability:

Data flows can be highly inconsistent with periodic peaks.

Why Big data?



More data → More accurate analysis → More confidence in decision making → Greater operational efficiencies, cost reduction, time reduction, new product development, optimized offerings, etc.

It helps organizations to:

- To understand Where, When and Why their customers buy
- Protect the company's client base with improved loyalty programs
- Seizing cross-selling and up selling opportunities
- Provide targeted promotional information
- Optimize Workforce planning and operations
- Improve inefficiencies in the company's supply chain
- Predict market trends
- Predict future needs
- Make companies more innovative and competitive
- It helps companies to discover new sources of revenue

Importance of Big data

TechVidvan
Importance of Big data
Cost Saving
Time Saving
Social media Listening
Customer acquisition
Marketing insights
Innovation

1. Cost Savings

Big Data tools like Apache Hadoop, Spark, etc. bring cost-saving benefits to businesses when they have to store large amounts of data. These tools help organizations in identifying more effective ways of doing business.

2. Time-Saving

Real-time in-memory analytics helps companies to collect data from various sources. Tools like Hadoop help them to analyze data immediately thus helping in making quick decisions based on the learnings.

3. Understand the market conditions

Big Data analysis helps businesses to get a better understanding of market situations. For example, analysis of customer purchasing behavior helps companies to identify the products sold most and thus produces those products accordingly. This helps companies to get ahead of their competitors.

4. Social Media Listening

Companies can perform sentiment analysis using Big Data tools. These enable them to get feedback about their company, that is, who is saying what about the company. Companies can use big data tools to improve their online presence.

5. Boost Customer Acquisition and Retention

Understanding customer needs is crucial for business success, as neglecting them can lead to loss of clientele and hinder growth. Big data analytics helps identify customer trends and behaviors, enabling businesses to make informed decisions and stay competitive.

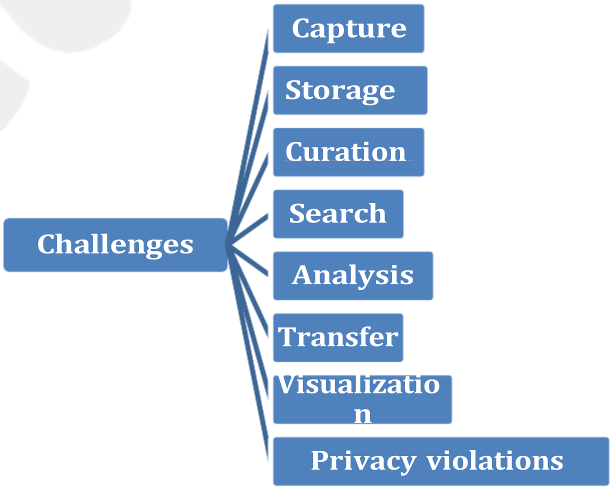
6. Solve Advertisers Problem and Offer Marketing Insights

Big data analytics shapes all business operations. It enables companies to fulfill customer expectations. Big data analytics helps in changing the company's product line. It ensures powerful marketing campaigns.

7. The driver of Innovations and Product Development

Big data makes companies capable to innovate and redevelop their products.

Challenges with Big data



1. Managing massive amounts of data

It's in the name—big data is big. Most companies are increasing the amount of data they collect daily. Eventually, the storage capacity a traditional data center can provide will be inadequate, which worries many business leaders. Forty-three percent of IT decision-makers in the technology sector worry about this data influx overwhelming their infrastructure. To handle this challenge, companies are migrating their IT infrastructure to the cloud. Cloud storage solutions can scale dynamically as more storage is needed. Big data software is designed to store large volumes of data that can be accessed and queried quickly.

2. Integrating data from multiple sources

The data itself presents another challenge to businesses. There is a lot, but it is also diverse because it can come from a variety of different sources. A business could have analytics data from multiple websites, sharing data from social media, user information from CRM software, email data, and more. None of this data is structured the same but may have to be integrated and reconciled to gather necessary insights and create reports. To deal with this challenge, businesses use data integration software, ETL software, and business intelligence software to map disparate data sources into a common structure and combine them so they can generate accurate reports.

3. Ensuring data quality

Analytics and machine learning processes that depend on big data to run also depend on clean, accurate data to generate valid insights and predictions. If the data is corrupted or incomplete, the results may not be what you expect. But as the sources, types, and quantity of data increase, it can be hard to determine if the data has the quality you need for accurate insights. Fortunately, there are solutions for this. Data governance applications will help organize, manage, and secure the data you use in your big data projects while also validating data sources against what you expect them to be and cleaning up corrupted and incomplete data sets. Data quality software can also be used specifically for the task of validating and cleaning your data before it is processed.

4. Keeping data secure

Many companies handle data that is sensitive, such as:

- Company data that competitors could use to take a bigger market share of the industry
- Financial data that could give hackers access to accounts
- Personal user information of customers that could be used for identity theft

If a business handles sensitive data, it will become a target of hackers. To protect this data from attack, businesses often hire cybersecurity professionals who keep up to date on security best practices and techniques to secure their systems. Whether you hire a consultant or keep it in-house, you need to ensure that data is encrypted, so the data is useless without an encryption key. Add identity and access authorization control to all resources so only the intended users can access it. Implement endpoint protection software so malware can't infect the system and real-time monitoring to stop threats immediately if they are detected.

5. Selecting the right big data tools

Fortunately, when a business decides to start working with data, there is no shortage of tools to help them do it. At the same time, the wealth of options is also a challenge. Big data software comes in many varieties, and their capabilities often overlap. How do you make sure you are choosing the right big data tools? Often, the best option is to hire a consultant who can determine which tools will fit best with what your business wants to do with big data. A big data professional can look at your current and future needs and choose an enterprise data streaming or ETL solution that will collect data from all your data sources and aggregate it. They can configure your cloud services and scale dynamically based on workloads. Once your system is set up with big data tools that fit your needs, the system will run seamlessly with very little maintenance. Thinking about hiring a data analytics company to help your business implement a big data strategy? Browse our list of top data analytics companies, and learn more about their services in our hiring guide.

6. Scaling systems and costs efficiently

If you start building a big data solution without a well-thought-out plan, you can spend a lot of money storing and processing data that is either useless or not exactly what your business needs. Big data is big, but it doesn't mean you have to process all of your data. When your business begins a data project, start with goals in mind and strategies for how you will use the data you have available to reach those goals. The team involved in implementing a solution needs to plan the type of data they need and the schemas they will use before they start building the system so the project doesn't go in the wrong direction. They also need to create policies for purging old data from the system once it is no longer useful.

7. Lack of skilled data professionals

One of the big data problems that many companies run into is that their current staff have never worked with big data before, and this is not the type of skill set you build overnight. Working with untrained personnel can result in dead ends, disruptions of workflow, and errors in processing. There are a few ways to solve this problem. One is to hire a big data specialist and have that specialist manage and train your data team until they are up to speed. The specialist can either be hired on as a full-time employee or as a consultant who trains your team and moves on, depending on your budget. Another option, if you have time to prepare ahead, is to offer training to your current team members so they will have the skills once your big data project is in motion. A third option is to choose one of the self-service analytics or business intelligence solutions that are designed to be used by professionals who don't have a data science background.

8. Organizational resistance

Another way people can be a challenge to a data project is when they resist change. The bigger an organization is, the more resistant it is to change. Leaders may not see the value in big data, analytics, or machine learning. Or they may simply not want to spend the time and money on a new project. This can be a hard challenge to tackle, but it can be done. You can start with a smaller project and a small team and let the results of that project prove the value of big data to other leaders and gradually become a data-driven business. Another option is placing big data experts in leadership roles so they can guide your business towards transformation.

Traditional Business Intelligence Vs Big Data

- BI(Business Intelligence) is a set of processes, architectures, and technologies that convert raw data into meaningful information that drives profitable business actions. It is a suite of software and services to transform data into actionable intelligence and knowledge.

- Traditional BI methodology is based on the principle of grouping all business data into a central server. Typically, this data is analyzed in offline mode, after storing the information in an environment called Data Warehouse. The data is structured in a conventional relational database with an additional set of indexes and forms of access to the tables (multidimensional cubes).

- In traditional BI environment, all the enterprise's data is housed in a central server whereas in a big data environment data resides in a distributed file system. The distributed file system scales by scaling in or out horizontally as compared to typical database server that scales vertically.
- In traditional BI, data is generally analyzed in an offline mode whereas in big data, it is analyzed in both real time as well as in offline mode.
- Traditional BI is about structured data and it is here that data is taken to processing functions (move data to code) whereas big data is about variety: Structured, semi-structured, and unstructured data and here the processing functions are taken to the data (move code to data).
- Data processed by Big Data solutions can be historical or come from real-time sources. Thus, companies can make decisions that affect their business in an agile and efficient way.
- Big Data technology uses parallel mass processing (MPP) concepts, which improves the speed of analysis. With MPP many instructions are executed simultaneously, and since the various jobs are divided into several parallel execution parts, at the end the overall results are reunited and presented. This allows you to analyze large volumes of information quickly.

Typical data warehouse

- Big Data has become the reality of doing business for organizations today.
- There is a boom in the amount of structured as well as raw data that floods every organization daily.

- If this data is managed well, it can lead to powerful insights and quality decision making.
- Big data analytics is the process of examining large data sets containing a variety of data types to discover some knowledge in databases, to identify interesting patterns and establish relationships to solve problems, market trends, customer preferences, and other useful information.
- Companies and businesses that implement Big Data Analytics often reap several business benefits.
- Companies implement Big Data Analytics because they want to make more informed business decisions.
- A data warehouse (DW) is a collection of corporate information and data derived from operational systems and external data sources.
- A data warehouse is designed to support business decisions by allowing data consolidation, analysis and reporting at different aggregate levels.
- Data is populated into the Data Warehouse through the processes of extraction, transformation and loading (ETL tools).
- Data analysis tools, such as business intelligence software, access the data within the warehouse.

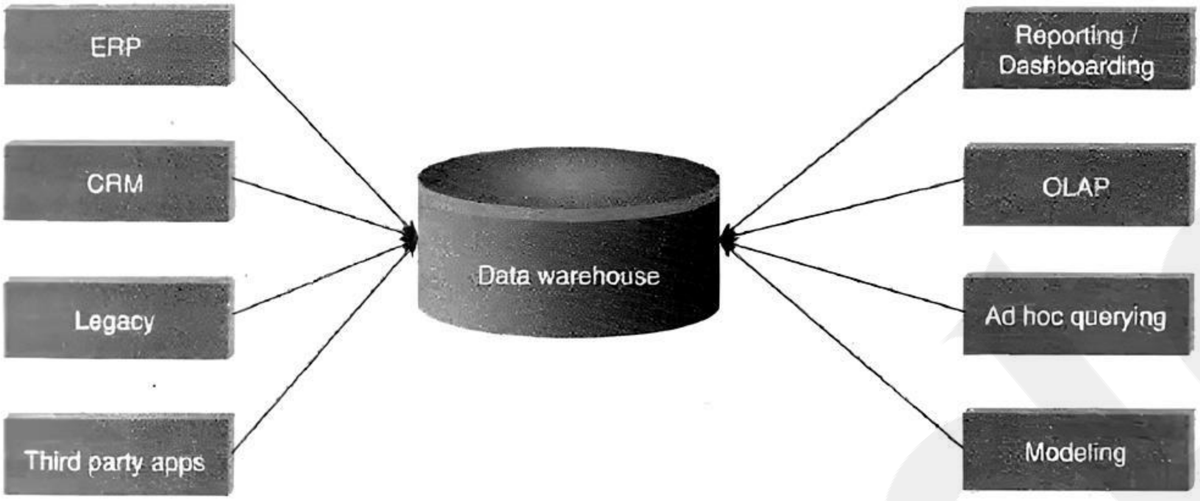


Fig: Typical Data Warehouse

Typical Hadoop Environment

- Hadoop is changing the perception of handling Big Data especially the unstructured data.
- Apache Hadoop enables surplus data to be streamlined for any distributed processing system across clusters of computers using simple programming models.
- It truly is made to scale up from single servers to a large number of machines, each and every offering local computation, and storage space.
- Instead of depending on hardware to provide high-availability, the library itself is built to detect and handle breakdowns at the application layer, so providing an extremely available service along with a cluster of computers, as both versions might be vulnerable to failures.

Hadoop Community Package Consists of

- File system and OS level abstractions
- A MapReduce engine (either MapReduce or YARN)

- The Hadoop Distributed File System (HDFS)
- Java ARchive (JAR) files
- Scripts needed to start Hadoop
- Source code, documentation and a contribution section

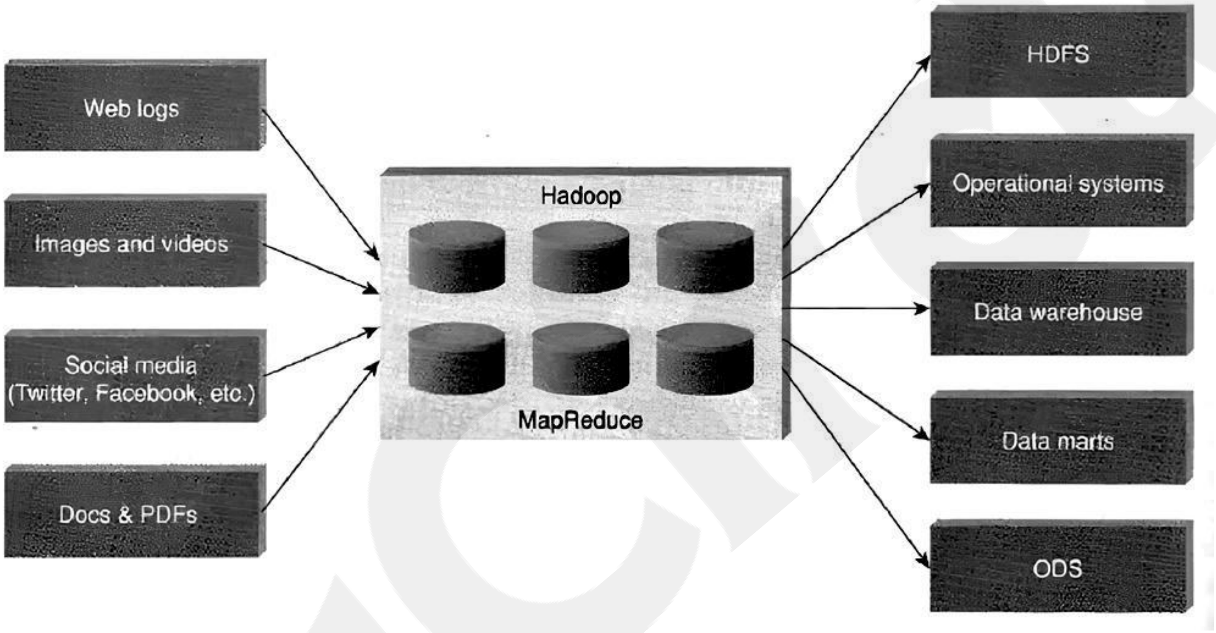
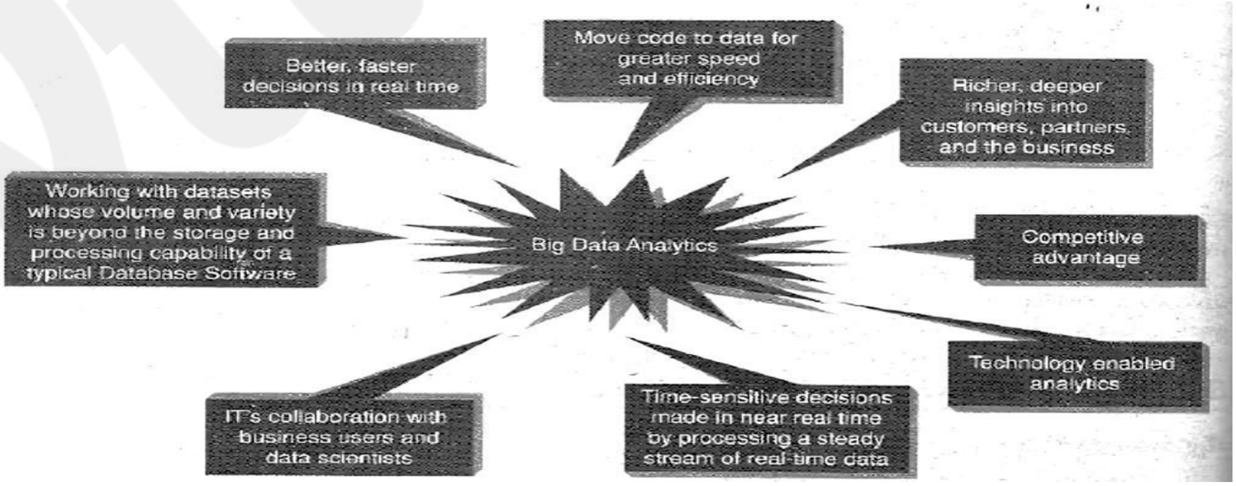


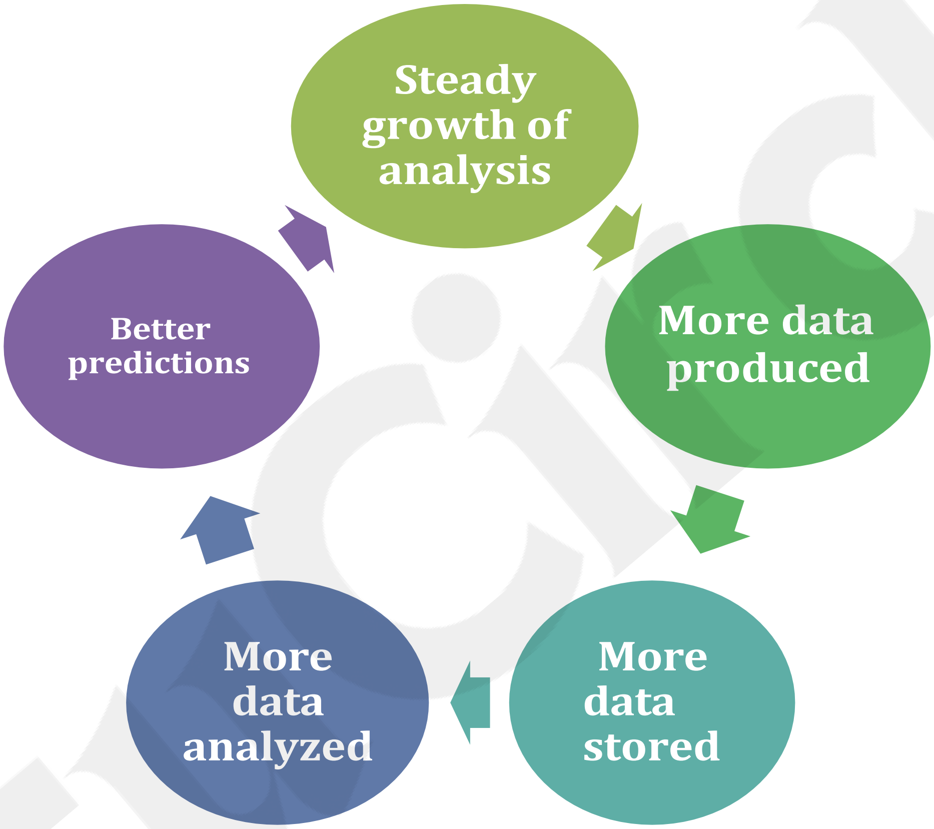
Fig: Typical Hadoop Environment

What is Big data Analytics?


Better, faster decisions in real time
Move code to data for greater speed and efficiency
Richer, deeper insights into customers, partners, and the business
Working with datasets whose volume and variety is beyond the storage and processing capability of a typical Database Software
Big Data Analytics
Competitive advantage
Technology enabled analytics
IT's collaboration with business users and data scientists
Time-sensitive decisions made in near real time by processing a steady stream of real-time data

Big data analytics is the process of examining large data sets containing a variety of data types to discover some knowledge in databases, to identify interesting patterns and establish relationships to solve problems, market trends, customer preferences, and other useful information.

What Big data Analytics Entails?



Classification of Analytics

1. Classify analytics into basic, operationalzed, advanced and monetized
2. Classify analytics into Analytics 1.0, Analytics 2.0 and Analytics 3.0

Classify analytics into basic, operationalzed, advanced and monetized:

1. Basic analytics: This primarily is slicing and dicing of data to help with basic business insights. This is about reporting on historical data, basic visualization, etc.

2. Operationalized analytics: It is operationalized analytics if it gets woven into the enterprise's business processes.

3. Advanced analytics: This largely is about forecasting for the future by way of predictive and prescriptive modeling.

4. Monetized analytics: This is analytics in use to derive direct business revenue.

Classify analytics into Analytics 1.0, Analytics 2.0 and Analytics 3.0

Analytics 1.0:

- Era: mid 1950s to 2009
- Descriptive statistics (report on events, occurrences, etc. of the past)
- Key questions asked: What happened? Why did it happen?
- Data from legacy systems, ERP, CRM, and 3rd party applications.
- Small and structured data sources. Data stored in enterprise data warehouses or data marts.
- Data was internally sourced.
- Relational databases

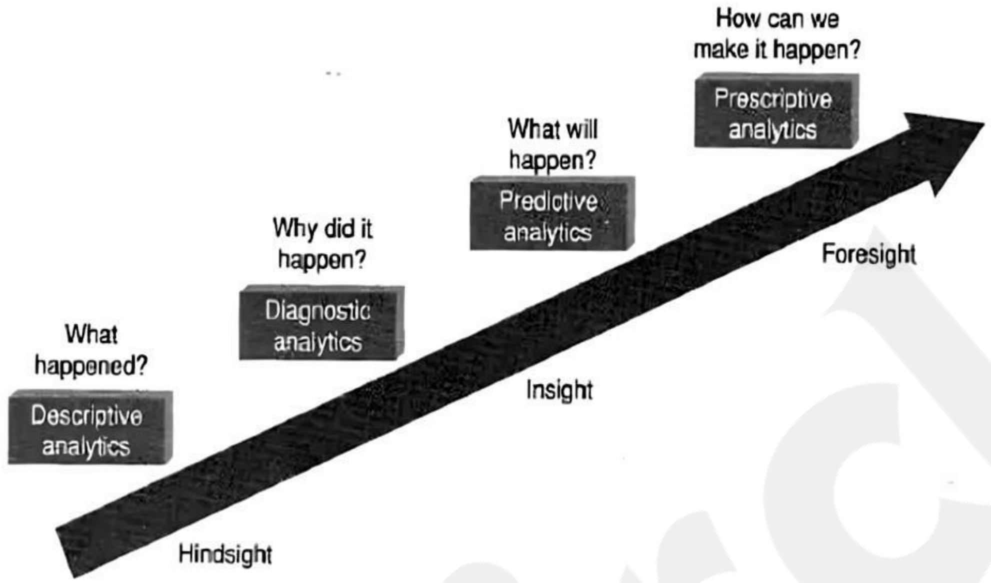
Analytics 2.0:

- Era: 2005 to 2012

- Descriptive statistics + predictive statistics (use data from the past to make predictions for the future)
- Key questions asked: What will happen? Why will it happen?
- Big data is being taken up seriously. Data is mainly unstructured, arriving at a much higher pace. This fast flow of data entailed that the influx of big volume data had to be stored and processed rapidly, often on massive parallel servers running Hadoop.
- Data was often externally sourced.Database appliances, Hadoop clusters, SQL to Hadoop environments, etc.

Analytics 3.0:

- Era: 2012 to present
- Descriptive + predictive + prescriptive statistics (use data from the past to make prophecies for the future and at the same time make recommendations to leverage the situation to one's advantage)
- Key questions asked: What will happen?When will it happen?Why will it happen? What should be the action taken to take advantage of what will happen?
- A blend of big data and data from legacy systems, ERP, CRM, and 3rd party applications. A blend of big data and traditional analytics to yield insights and offerings with speed and impact.
- Data is both being internally and externally sourced.
- In memory analytics, in database processing, agile analytical methods, machine learning techniques, etc.



Descriptive analytics

Descriptive analytics is a statistical method that is used to search and summarize historical data in order to identify patterns or meaning. Data aggregation and data mining are two techniques used in descriptive analytics to discover historical data. Data is first gathered and sorted by data aggregation in order to make the datasets more manageable by analysts. Data mining describes the next step of the analysis and involves a search of the data to identify patterns and meaning. Identified patterns are analyzed to discover the specific ways that learners interacted with the learning content and within the learning environment.

Advantages:

- Quickly and easily report on the Return on Investment (ROI) by showing how performance achieved business or target goals.

- Identify gaps and performance issues early - before they become problems.

- Identify specific learners who require additional support, regardless of how many students or employees there are.

- Identify successful learners in order to offer positive feedback or additional resources.

- Analyze the value and impact of course design and learning resources.

Predictive analytics

Predictive Analytics is a statistical method that utilizes algorithms and machine learning to identify trends in data and predict future behaviors The software for predictive analytics has moved beyond the realm of statisticians and is becoming more affordable and accessible for different markets and industries, including the field of learning & development.

For online learning specifically, predictive analytics is often found incorporated in the Learning Management System (LMS), but can also be purchased separately as specialized software. For the learner, predictive forecasting could be as simple as a dashboard located on the main screen after logging in to access a course. Analyzing data from past and current progress, visual indicators in the dashboard could be provided to signal whether the employee was on track with training requirements.

Advantages:

- Personalize the training needs of employees by identifying their gaps, strengths, and weaknesses; specific learning resources and training can be offered to support individual needs.

- Retain Talent by tracking and understanding employee career progression and forecasting what skills and learning resources would best benefit their career paths. Knowing what skills employees need also benefits the design of future training.

- Support employees who may be falling behind or not reaching their potential by offering intervention support before their performance puts them at risk.

- Simplified reporting and visuals that keep everyone updated when predictive forecasting is required.

Prescriptive analytics

Prescriptive analytics is a statistical method used to generate recommendations and make decisions based on the computational findings of algorithmic models. Generating automated decisions or recommendations requires specific and unique algorithmic models and clear direction from those utilizing the analytical technique. A recommendation cannot be generated without knowing what to look for or what problem is desired to be solved. In this way, prescriptive analytics begins with a problem.

Example

A Training Manager uses predictive analysis to discover that most learners without a particular skill will not complete the newly launched course. What could be done? Now prescriptive analytics can be of assistance on the matter and help determine options for action. Perhaps an algorithm can detect the learners who require that new course, but lack that particular skill, and send an automated recommendation that they take an additional training resource to acquire the missing skill. The accuracy of a generated decision or recommendation, however, is only as good as the quality of data and the algorithmic models developed. What may work for one company's training needs may not make sense when put into practice in another company's training department. Models are generally recommended to be tailored for each unique situation and need.

Descriptive vs Predictive vs Prescriptive Analytics

Descriptive Analytics is focused solely on historical data. You can think of Predictive Analytics as then using this historical data to develop statistical models that will then forecast about future possibilities. Prescriptive Analytics takes Predictive Analytics a step further and takes the possible forecasted outcomes and predicts consequences for these outcomes.

Why Big data Analytics is important?

1. Reactive - Business Intelligence: Business Intelligence (BI) allows the businesses to make faster and better decisions by providing the right information to the right person at the right time in the right format. It is about analysis of the past or historical data and then displaying the findings of the analysis or reports in the form of enterprise dashboards, alerts, notifications, etc. It has support for both pre-specified reports as well as ad hoc querying.

2. Reactive Big Data Analytics: Here the analysis is done on huge datasets but the approach is still reactive as it is still based on static data.

3. Proactive Analytics: This is to support futuristic decision making by the use of data mining, predictive modeling, text mining, and statistical analysis. This analysis is not on big data as it still uses the traditional database management practices on big data and therefore has severe limitations on the storage capacity and the processing capability.

4. Proactive - Big Data Analytics: This is sieving through terabytes, petabytes, exabytes of information to filter out the relevant data to analyze. This also includes high performance analytics to gain rapid insights from big data and the ability to solve complex problems using more data.

Technologies used in Big data Environments

1. In-Memory Analytics

Data access from non-volatile storage such as hard disk is a slow process. The more the data is required to be fetched from hard disk or secondary storage, the slower the process gets. One way to combat this challenge is to pre-process and store data (cubes, aggregate tables, query sets, etc.) so that the CPU has to fetch a small subset of records. But this requires thinking in advance as to what data will be required for analysis. If there is a need for different or more data, it is back to the initial process of pre-computing and storing data or fetching it from secondary storage.

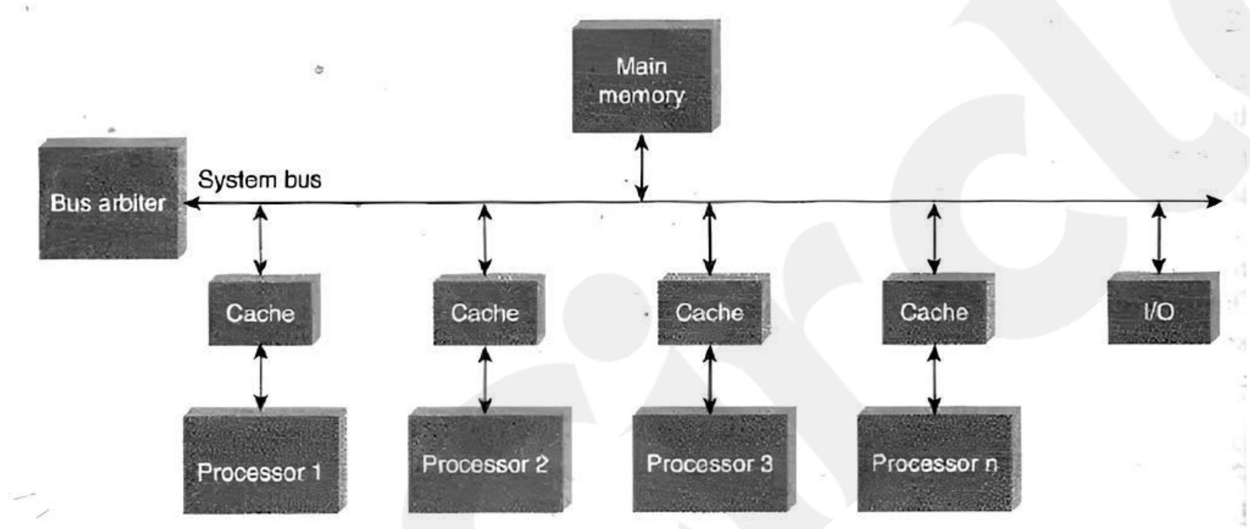
This problem has been addressed using in-memory analytics. Here all the relevant data is stored in Random Access Memory (RAM) or primary storage thus eliminating the need to access the data from hard disk. The advantage is faster access, rapid deployment, better insights, and minimal IT involvement.

2 In-Database Processing

In-database processing is also called as in-database analytics. It works by fusing data warehouses with analytical systems. Typically the data from various enterprise On Line Transaction Processing (OLTP) systems after cleaning up (de-duplication, scrubbing, etc.) through the process of ETL is stored in the Enterprise Data Warehouse (EDW) or data marts. The huge datasets are then exported to analytical programs for complex and extensive computations. With in-database processing, the database program itself can run the computations eliminating the need for export and thereby saving on time. Leading database vendors are offering this feature to large businesses.

3. Symmetric Multiprocessor System (SMP)

In SMP, there is a single common main memory that is shared by two or more identical processors. The processors have full access to all I/O devices and are controlled by a single operating system instance. SMP are tightly coupled multiprocessor systems. Each processor has its own high-speed memory, called cache memory and are connected using a system bus.

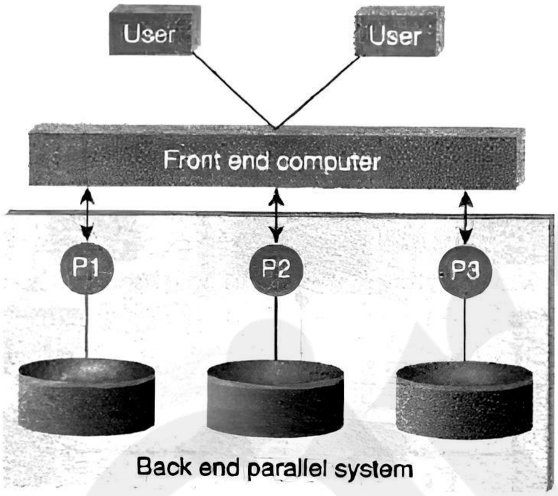


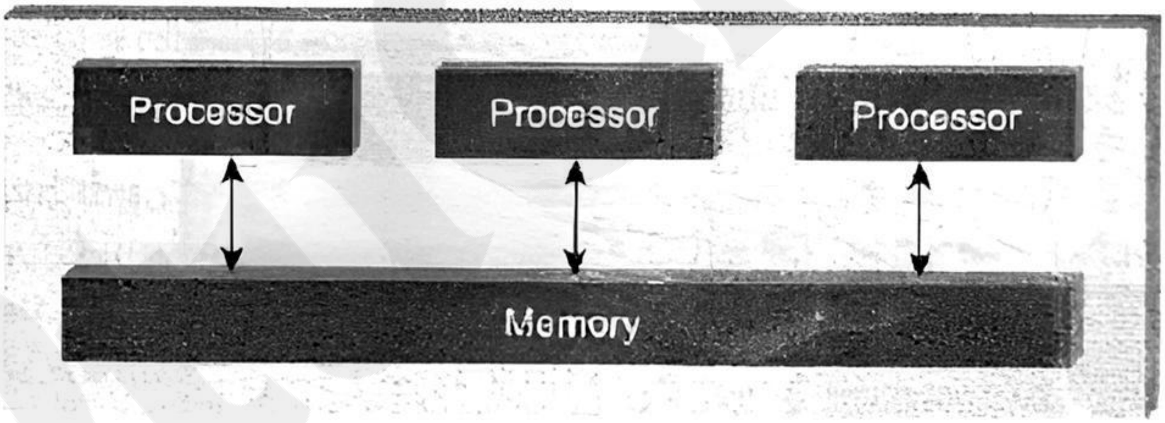
4. Massively Parallel Processing

Massive Parallel Processing (MPP) refers to the coordinated processing of programs by a number of processors working parallel. The processors, each have their own operating systems and dedicated memory. They work on different parts of the same program. The MPP processors communicate using some sort of messaging interface. The MPP systems are more difficult to program as the application must be divided in such a way that all the executing segments can communicate with each other. MPP is different from Symmetrically Multiprocessing (SMP) in that SMP works with the processors sharing the same operating system and same memory. SMP is also referred to as tightly-coupled multiprocessing.

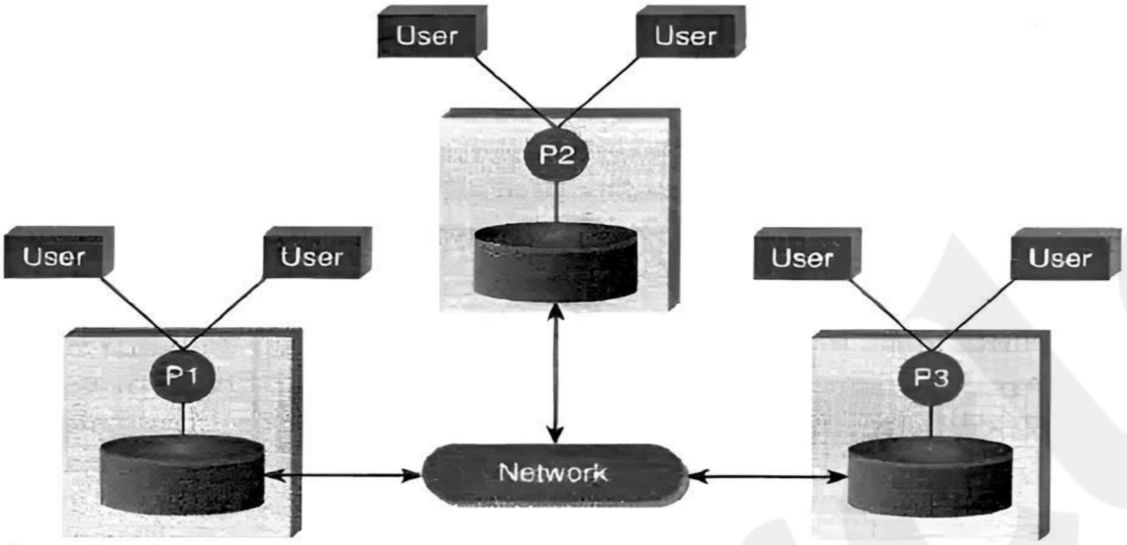
5. Difference between Parallel and Distributed systems:

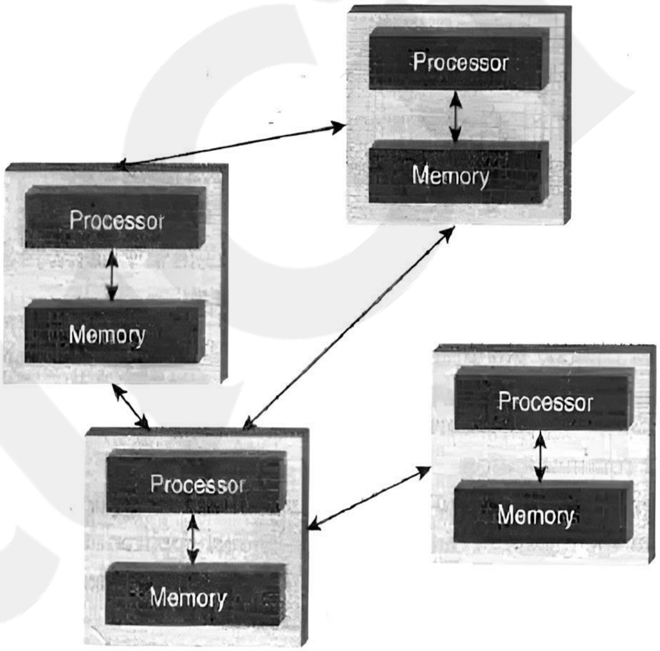
The parallel database system is a tightly coupled system. The processors cooperate for query processing. Either the processors have access to a common memory as shown in figure below or make use of message passing for communication.





Distributed database systems are known to be loosely coupled and are composed by individual machines. Each of the machines can run their individual application and serve their own respective user. The data is usually distributed across several machines, thereby necessitating quite a number of machines to be accessed to answer a user query.





6. Shared Nothing Architecture

The three most common types of architecture for multiprocessor high transaction rate systems are:

1. Shared Memory (SM).

2. Shared Disk (SD).

3. Shared Nothing (SN).

- In shared memory architecture, **a common central memory** is shared by multiple processors.
- In shared disk architecture, **multiple processors share a common collection of disks** while having their own private memory.
- In shared nothing architecture, **neither memory nor disk is shared among multiple processors**.

Advantages of a "Shared Nothing Architecture"

1. Fault Isolation: In a Shared Nothing Architecture, each node operates independently without shared memory or disk, ensuring faults are isolated to the failing node. Failures affect only message exchanges, preventing system-wide disruptions.

2. Scalability: Assume that the disk is a shared resource. It implies that the controller and the disk bandwidth are also shared. Synchronization will have to be implemented to maintain a consistent shared state. This would mean that different nodes will have to take turns to access the critical data. This imposes a limit on how many nodes can be added to the distributed shared disk system, thus compromising on scalability.

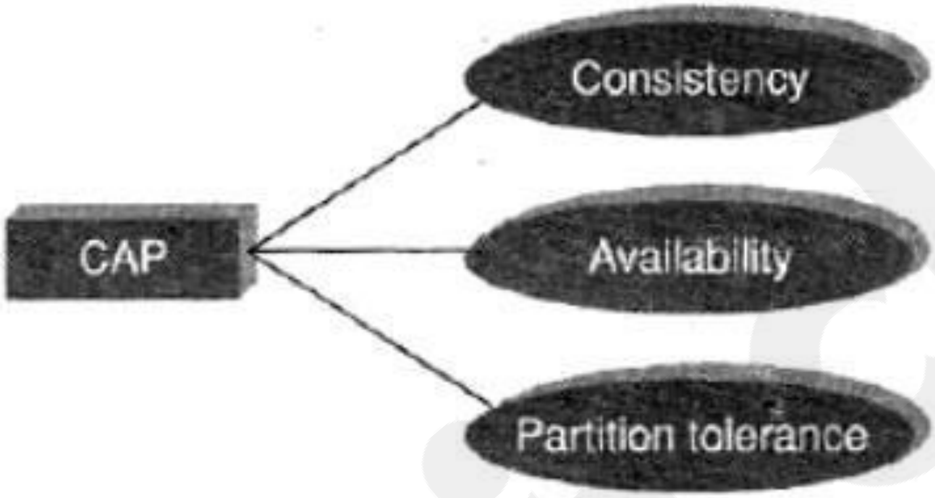
7. CAP Theorem

The CAP theorem is also called the Brewer's Theorem. It states that in a distributed computing environment (a collection of interconnected nodes that share data), it is impossible to provide the following guarantees. At best you can have two of the following three - one must be sacrificed.

1. Consistency

2. Availability

3. Partition tolerance



CAP Theorem

1. Consistency implies that every read fetches the last write.

2. Availability implies that reads and writes always succeed. In other words, each non-failing node will return a response in a reasonable amount of time.

3. Partition tolerance implies that the system will continue to function when network partition occurs.

When to choose consistency over availability and vice-versa...

1. Choose availability over consistency when your business requirements allow some flexibility around when the data in the system synchronizes.

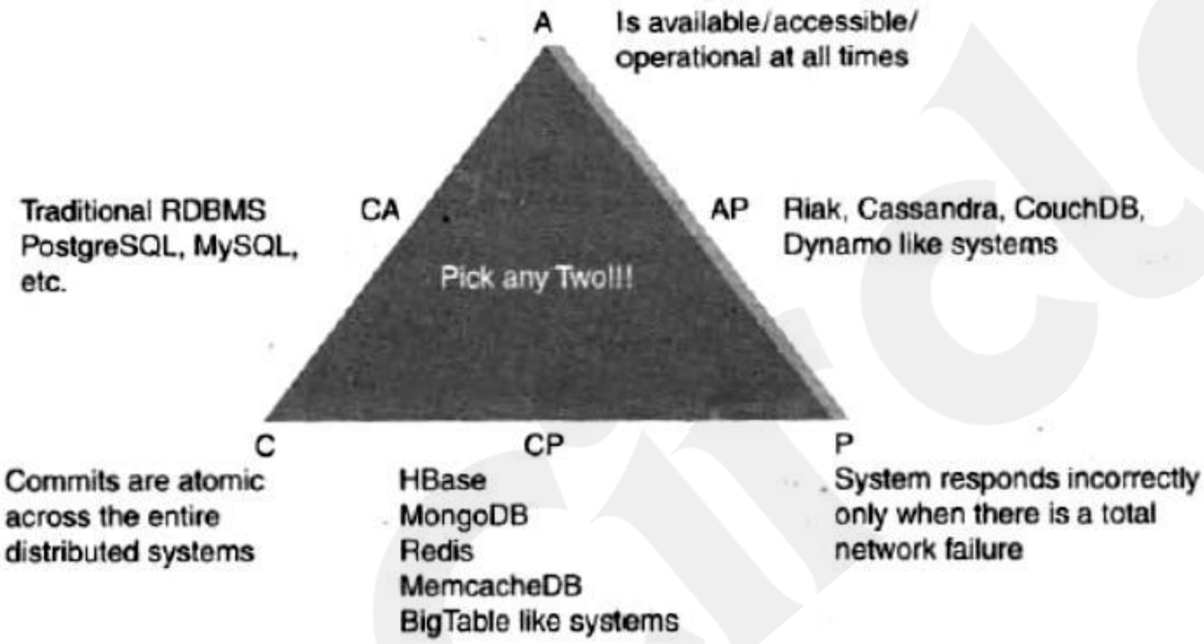
2. Choose consistency over availability when your business requirements dead atomic reads and

Examples of databases that follow one of the possible three combinations

1. Availability and Partition Tolerance (AP)

2. Consistency and Partition Tolerance (CP)

3. Consistency and Availability (CA)


A
Is available/accessible/ operational at all times
CA
Traditional RDBMS PostgreSQL, MySQL, etc.
AP
Riak, Cassandra, CouchDB, Dynamo like systems
Pick any Two!!!
C
Commits are atomic across the entire distributed systems
CP
HBase
MongoDB
Redis
MemcacheDB
BigTable like systems
P
System responds incorrectly only when there is a total network failure

Few top Analytical tools

There is no dearth of analytical tools in the market.

1. MS Excel
2. Statistical Analysis System(SAS)
3. IBM Statistical Package for the Social Sciences(SPSS) Modeler
4. Statistica
5. Salford systems
6. World Programming System(WPS)

Open Source Analytics Tools

1. R analytics
2. Weka

NoSQL (NOT ONLY SQL)

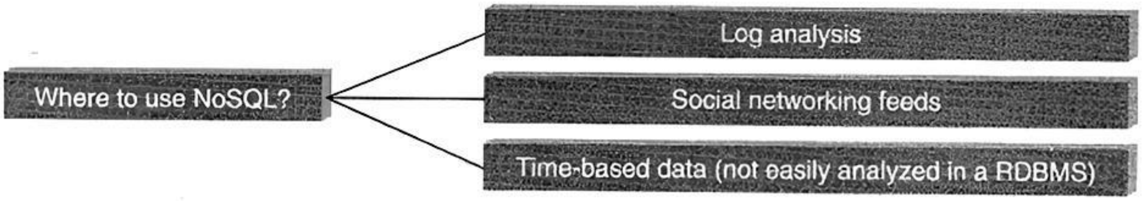
The term NoSQL was first coined by Carlo Strozzi in 1998 to name his lightweight, open-source, relational database that did not expose the standard SQL interface.

Few features of NoSQL databases are as follows: .

1. They are open sources
2. They are nonrelational
3. They are distributed
4. They are schema less
5. They are cluster friendly
6. They are born out of 21st century web applications.

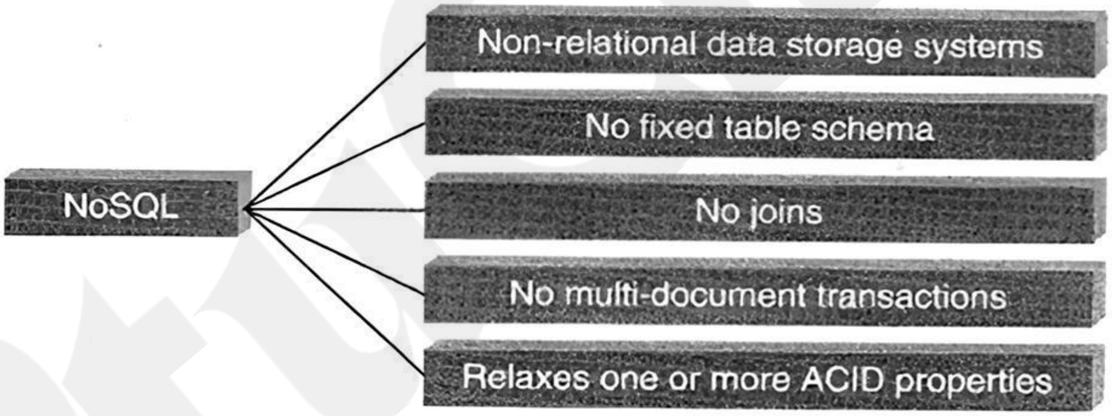
Where is it Used?

- NoSQL databases are widely used in big data and other real-time web applications.
- NoSQL. databases is used to stock log data which can then be pulled for analysis.
- It is used to store social media data and all such data which cannot be stored and analyzed comfortably in RDBMS.



What is it?

- NoSQL stands for Not Only SQL. These are non-relational, open source, distributed databases. They are hugely popular today owing to their ability to scale out or scale horizontally and the adeptness at dealing with a rich variety of data: structured, semi-structured and unstructured data.


NoSQL
Non-relational data storage systems
No fixed table schema
No joins
No multi-document transactions
Relaxes one or more ACID properties

1. **Are non-relational:** They do not adhere to relational data model, In fact, they are either key-value pairs or document-oriented or column-oriented or graph-based databases.
2. **Are distributed:** They are distributed meaning the data is distributed across several nodes in a cluster constituted of low-cost commodity hardware.

3. **Offer no support for ACID properties (Atomicity, Consistency, Isolation, and Durability)**: They do not offer support for ACID properties of transactions. On the contrary, they have adherence to Brewer's CAP (Consistency, Availability, and Partition tolerance) theorem and are often seen compromising on consistency in favor of availability and partition tolerance.

4. **Provide no fixed table schema**: NoSQL databases are becoming increasing popular owing to their support for flexibility to the schema. They do not mandate for the daa to strictly adhere to any schema structure at the time of storage.

Types of NoSQL Databases:

Document databases: These databases store data as semi-structured documents, such as JSON or XML, and can be queried using document-oriented query languages. Eg. MongoDB

Key-value stores: These databases store data as key-value pairs, and are optimized for simple and fast read/write operations.Eg. Redis, Coherence

Column-family stores: These databases store data as column families, which are sets of columns that are treated as a single entity. They are optimized for fast and efficient querying of large amounts of data. Eg., Big Table

Graph databases: These databases store data as nodes and edges, and are designed to handle complex relationships between data. Eg., Amazon Neptune

Why NoSQL?

1. It has scale out architecture instead of the monolithic architecture of relational databases.

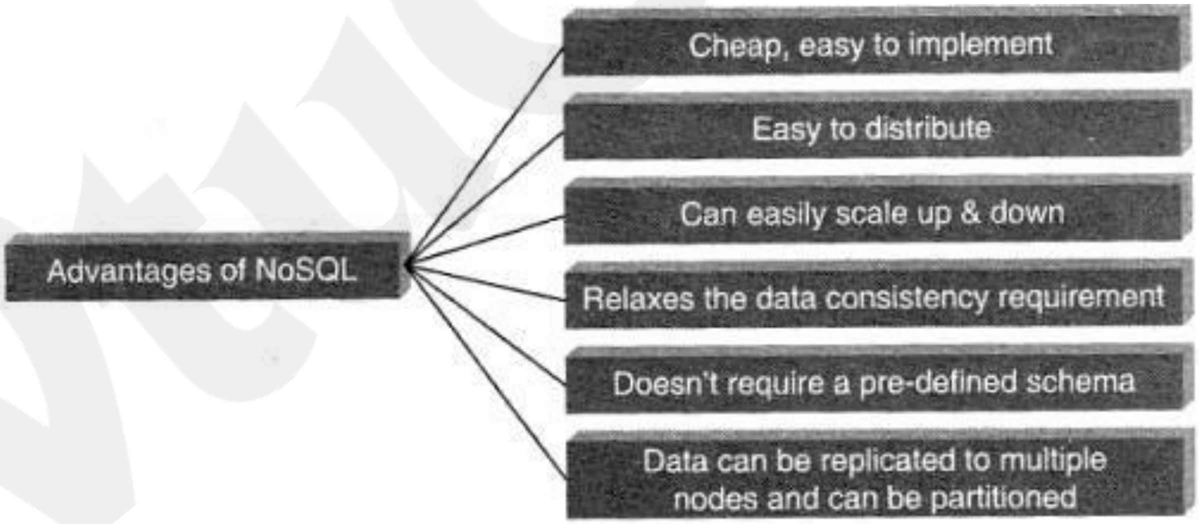
2. It can house large volumes of structured, semi-structured, and unstructured data.

3. **Dynamic schema:** NoSQL database allows insertion of data without a pre-defined schema. In other words, it facilitates application changes in real time, which thus supports faster development, easy code integration, and requires less database administration.

4. **Auto-sharding:** It automatically spreads data across an arbitrary number of servers. The application in question is more often not even aware of the composition of the server pool. It balances the load of data and query on the available servers; and if and when a server goes down, it is quickly replaced without any major activity disruptions.

5. **Replication:** It offers good support for replication which in turn guarantees high availability, fault tolerance, and disaster recovery.

Advantages of NoSQL (NOT ONLY SQL)


Advantages of NoSQL
Cheap, easy to implement
Easy to distribute
Can easily scale up & down
Relaxes the data consistency requirement
Doesn't require a pre-defined schema
Data can be replicated to multiple nodes and can be partitioned

1. Can easily scale up and down: NoSQL database supports scaling rapidly and elastically and even allows to scale to the cloud.

(a) Cluster scale: It allows distribution of database across 100+ nodes often in multiple data centers.

(b) Performance scale: It sustains over 100,000+ database reads and writes per second.

(c) Data scale: It supports housing of 1 billion+ documents in the database.

2. Doesn't require a pre-defined schema: NoSQL does not require any adherence to pre-defined schema. It is pretty flexible. For example, if we look at MongoDB, the documents (equivalent of records in RDBMS) in a collection (equivalent of table in RDBMS) can have different sets of key-value pairs.

3. Cheap, easy to implement: Deploying NoSQL properly allows for all of the benefits of scale, high availability, fault tolerance, etc. while also lowering operational costs.

4. Relaxes the data consistency requirement: NoSQL databases have adherence to CAP theorem (Consistency, Availability, and Partition tolerance). Most of the NoSQL databases compromise on consistency in favor of availability and partition tolerance. However, they do go for eventual consistency.

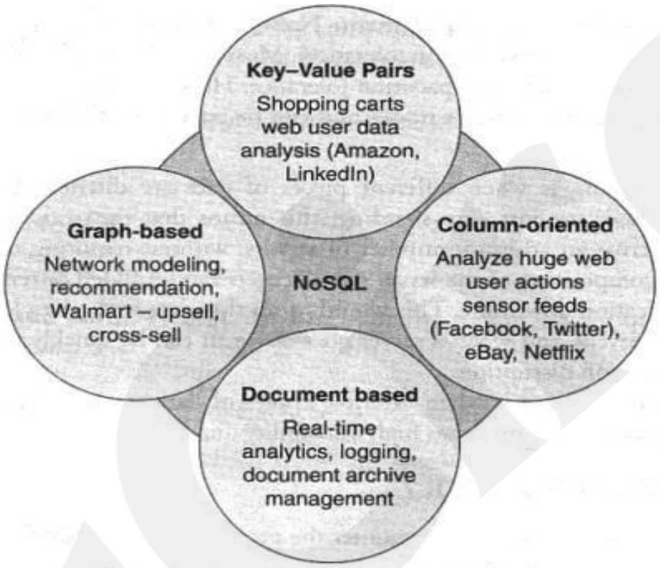
5. Data can be replicated to multiple nodes and can be partitioned:

There are two terms that we will discuss here:

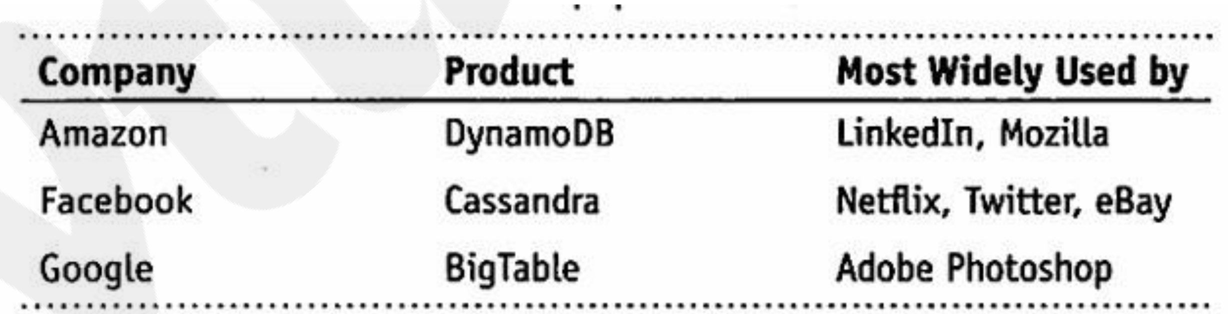
(a) **Sharding:** Sharding is when different pieces of data are distributed across multiple servers. NoSQL databases support auto-sharding; this means that they can natively and automatically spread data across an arbitrary number of servers, without requiring the application to even be aware of the composition of the server pool. Servers can be added or removed from the data layer without application downtime. This would mean that data and query load are automatically balanced across servers, and when a server goes down, it can be quickly and transparently replaced with no application disruption.

(b) **Replication:** Replication is when multiple copies of data are stored across the cluster and even across data centers. This promises high availability and fault tolerance.

Use of NoSQL in industry

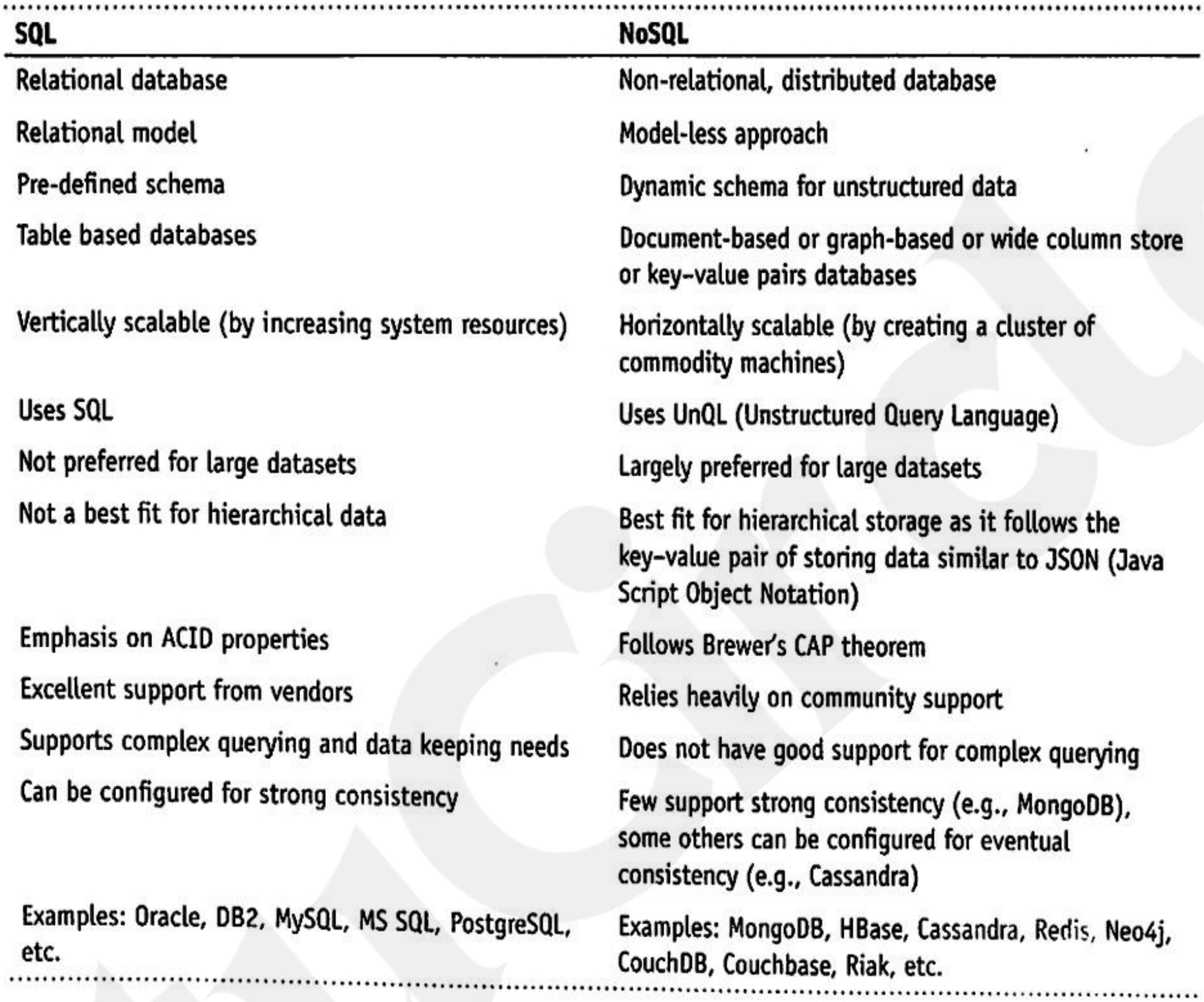

Key–Value Pairs
Shopping carts web user data analysis (Amazon, LinkedIn)
Graph-based
Network modeling, recommendation, Walmart – upsell, cross-sell
NoSQL
Column-oriented
Analyze huge web user actions sensor feeds (Facebook, Twitter), eBay, Netflix
Document based
Real-time analytics, logging, document archive management

NoSQL vendors



Company	Product	Most Widely Used by
Amazon	DynamoDB	LinkedIn, Mozilla
Facebook	Cassandra	Netflix, Twitter, eBay
Google	BigTable	Adobe Photoshop

Difference between SQL and NoSQL



SQL	NoSQL
Relational database	Non-relational, distributed database
Relational model	Model-less approach
Pre-defined schema	Dynamic schema for unstructured data
Table based databases	Document-based or graph-based or wide column store or key–value pairs databases
Vertically scalable (by increasing system resources)	Horizontally scalable (by creating a cluster of commodity machines)
Uses SQL	Uses UnQL (Unstructured Query Language)
Not preferred for large datasets	Largely preferred for large datasets
Not a best fit for hierarchical data	Best fit for hierarchical storage as it follows the key–value pair of storing data similar to JSON (Java Script Object Notation)
Emphasis on ACID properties	Follows Brewer's CAP theorem
Excellent support from vendors	Relies heavily on community support
Supports complex querying and data keeping needs	Does not have good support for complex querying
Can be configured for strong consistency	Few support strong consistency (e.g., MongoDB), some others can be configured for eventual consistency (e.g., Cassandra)
Examples: Oracle, DB2, MySQL, MS SQL, PostgreSQL, etc.	Examples: MongoDB, HBase, Cassandra, Redis, Neo4j, CouchDB, Couchbase, Riak, etc.

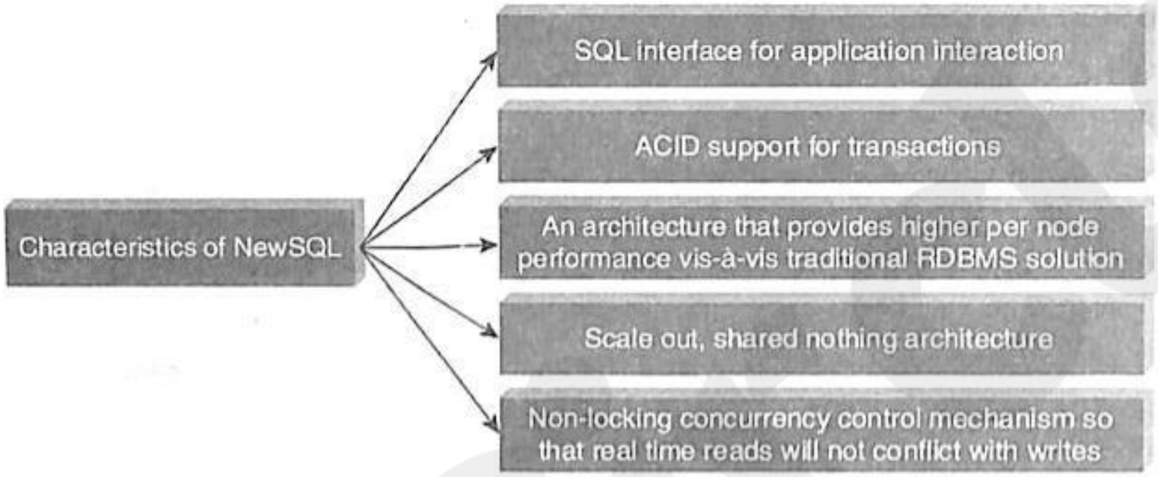
NewSQL

There is yet another new term doing the rounds - "NewSQL". So, what is NewSQL and how is it different from SQL and NoSQL?

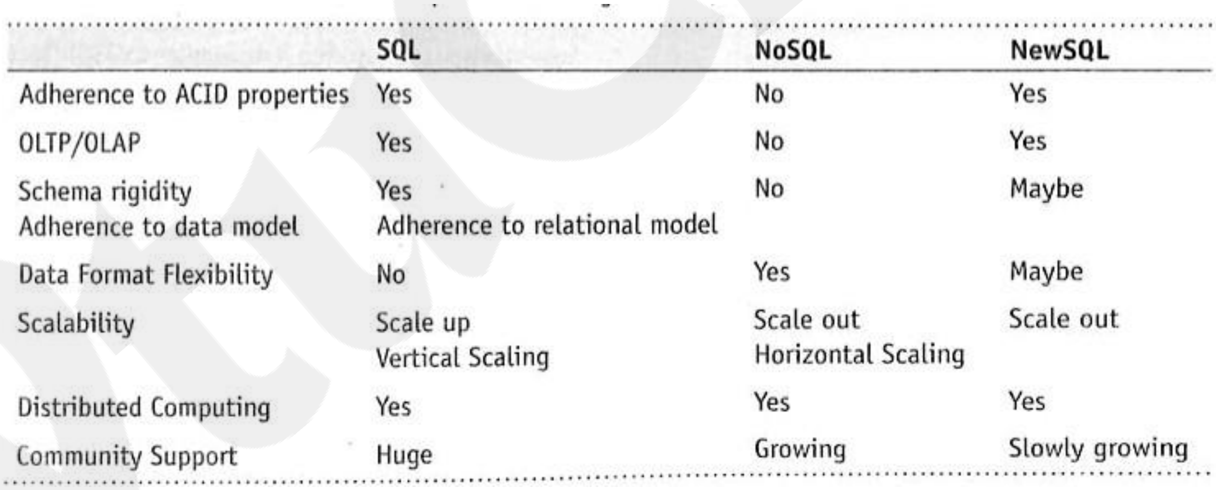
We need a database that has the same scalable performance of NoSQL systems for On Line Transaction Processing (OLTP) while still maintaining the ACID guarantees of a traditional database. This new modern RDBMS is called NewSQL. It supports relational data model and uses SQL as their primary interface.

Characteristics of NewSQL

NewSQL is based on the shared nothing architecture with a SQL interface for application interaction.


Characteristics of NewSQL
SQL interface for application interaction
ACID support for transactions
An architecture that provides higher per node performance vis-à-vis traditional RDBMS solution
Scale out, shared nothing architecture
Non-locking concurrency control mechanism so that real time reads will not conflict with writes

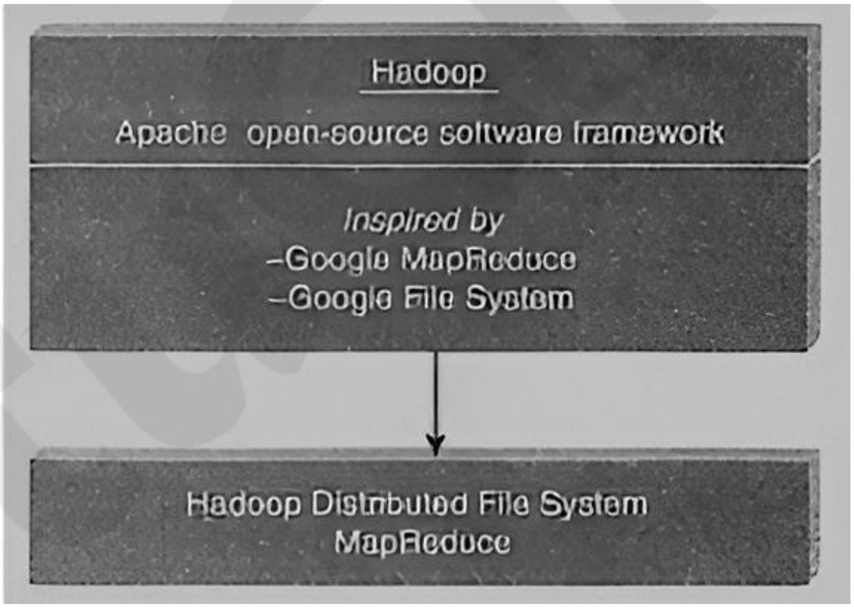
Comparison of SQL, NoSQL, and NewSQL



	SQL	NoSQL	NewSQL
Adherence to ACID properties	Yes	No	Yes
OLTP/OLAP	Yes	No	Yes
Schema rigidity	Yes	No	Maybe
Adherence to data model	Adherence to relational model		
Data Format Flexibility	No	Yes	Maybe
Scalability	Scale up Vertical Scaling	Scale out Horizontal Scaling	Scale out
Distributed Computing	Yes	Yes	Yes
Community Support	Huge	Growing	Slowly growing

Hadoop

- Hadoop is an open-source project of the Apache foundation.
- It is a framework written in Java, originally developed by Doug Cutting in 2005 who named it after his son's toy elephant who was working with Yahoo then.
- It was created to support distribution for "Nutch", the text search engine.
- Hadoop uses Google's MapReduce and Google File System technologies as its foundation.
- Hadoop is now a core part of the computing infrastructure for companies such as Yahoo, Facebook, LinkedIn, Twitter, etc.


Hadoop
Apache open-source software framework
Inspired by
–Google MapReduce
–Google File System
Hadoop Distributed File System
MapReduce

Features of Hadoop

1. It is optimized to handle massive quantities of structured, semi-structured, and unstructured data, using commodity hardware, that is, relatively inexpensive computers.

2. Hadoop has a shared nothing architecture.

3. It replicates its data across multiple computers so that if one goes down, the data can still be processed from another machine that stores its replica.

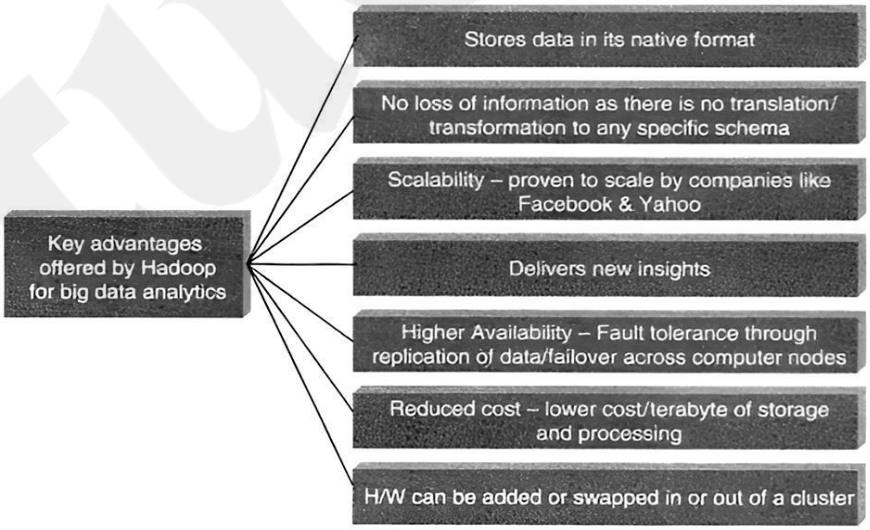
4. Hadoop is for high throughput rather than low latency. It is a batch operation handling massive quantities of data; therefore the response time is not immediate.

5. It complements On-Line Transaction Processing (OLTP) and On-Line Analytical Processing (OLAP). However, it is not a replacement for a relational database management system.

6. It is NOT good when work cannot be parallelized or when there are dependencies within the data.

7. It is NOT good for processing small files. It works best with huge data files and datasets.

Key advantages of Hadoop


Key advantages offered by Hadoop for big data analytics
Stores data in its native format
No loss of information as there is no translation/ transformation to any specific schema
Scalability – proven to scale by companies like Facebook & Yahoo
Delivers new insights
Higher Availability – Fault tolerance through replication of data/failover across computer nodes
Reduced cost – lower cost/terabyte of storage and processing
H/W can be added or swapped in or out of a cluster

1. Stores data in its native format: Hadoop's data storage framework (HDFS - Hadoop Distributed File System) can store data in its native format. There is no structure that is imposed while keying in data or storing data. HDFS is pretty much schema-less. It is only later when the data needs to be processed that structure is imposed on the raw data.

2. Scalable: Hadoop can store and distribute very large datasets (involving thousands of terabytes of data) across hundreds of inexpensive servers that operate in parallel.

3. Cost-effective: Owing to its scale-out architecture, Hadoop has a much reduced cost/terabyte of storage and processing.

4. Resilient to failure: Hadoop is fault-tolerant. It practices replication of data diligently which means whenever data is sent to any node, the same data also gets replicated to other nodes in the cluster, there- by ensuring that in the event of a node failure, there will always be another copy of data available for use.

5. Flexibility: One of the key advantages of Hadoop is its ability to work with all kinds of data: structured, semi-structured, and unstructured data. It can help derive meaningful business insights from email conversations, social media data, click-stream data, etc. It can be put to several purposes such as log analysis, data mining, recommendation systems, market campaign analysis, etc.

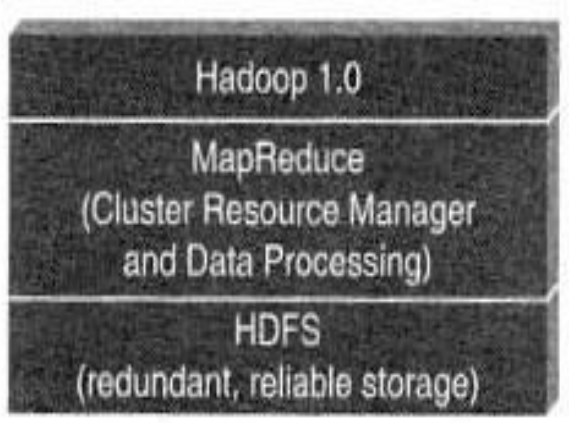
6. Fast: Processing is extremely fast in Hadoop as compared to other conventional systems owing to the "move code to data" paradigm.

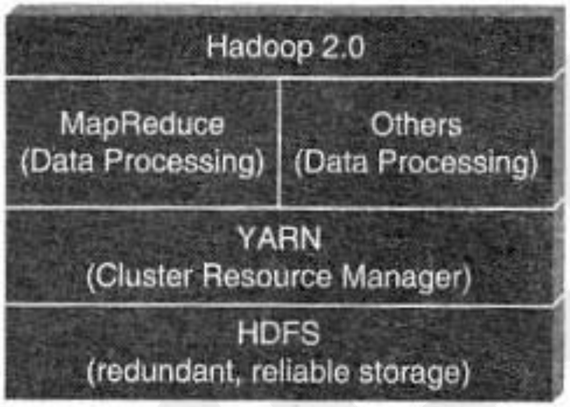
Hadoop has a shared-nothing architecture.

Versions of Hadoop

There are two versions of Hadoop available:

1. Hadoop 1.0
2. Hadoop 2.0





1. **Hadoop 1.0:**

It has 2 main parts:

1. **Data storage framework:** It is a general-purpose file system called Hadoop Distributed File System (HDFS). HDFS is schema-less. It simply stores data files. These data files can be in just about any format. The idea is to store files as close to their original form as possible. This in turn provides the business units and the organization the much needed flexibility and agility without being overly worried by what it can implement.
2. **Data processing framework:** This is a simple functional programming model initially popularized by Google as MapReduce. It essentially uses two functions: the MAP and the REDUCE functions to process data. The "Mappers" take in a set of key-value pairs and generate intermediate data (which is another list of key-value pairs). The "Reducers" then act on this input to produce the output data. The two functions seemingly work in isolation from one another, thus

enabling the processing to be highly distributed in a highly-parallel, fault-tolerant, and scalable way.

There were, however, **a few limitations of Hadoop 1.0**. They are as follows:

1. The first limitation was the requirement for MapReduce programming expertise along with proficiency required in other programming languages, notably Java.

2. It supported only batch processing which although is suitable for tasks such as log analysis, large-scale data mining projects but pretty much unsuitable for other kinds of projects.

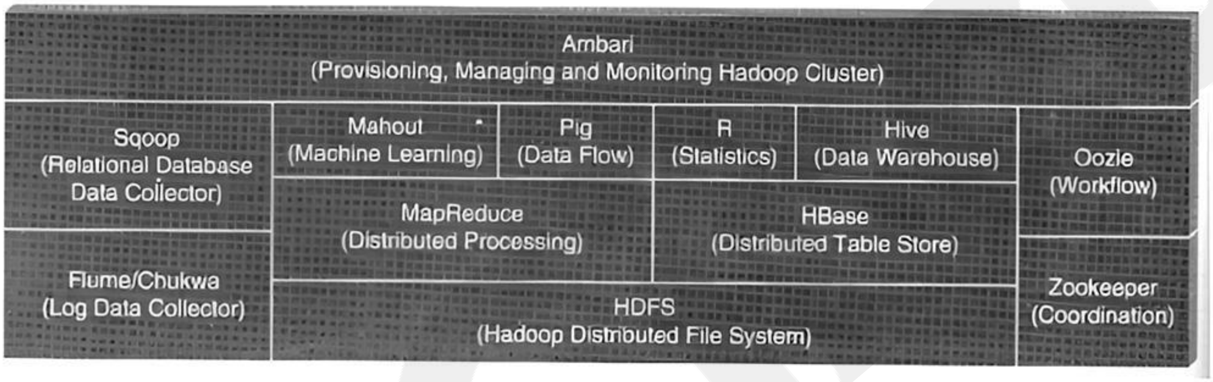
3. One major limitation was that Hadoop 1.0 was tightly computationally coupled with MapReduce, which meant that the established data management vendors were left with two options: Either rewrite their functionality in MapReduce so that it could be executed in Hadoop or extract the data from HDFS and process it outside of Hadoop. None of the options were viable as it led to process inefficiencies caused by the data being moved in and out of the Hadoop cluster.

2. Hadoop 2.0:

In Hadoop 2.0, HDFS continues to be the data storage framework. However, a new and separate resource management framework called **Y**et **A**nother **R**esource **N**egotiator (YARN) has been added. Any application capable of dividing itself into parallel tasks is supported by YARN. YARN coordinates the allocation of subtasks of the submitted application, thereby further enhancing the flexibility, scalability, and efficiency of the applications. It works by having an Application Master in place of the erstwhile JobTracker, running applications on resources governed by a new NodeManager (in place of the erstwhile TaskTracker). Application Master is able to run any application and not just MapReduce.

This, in other words, means that the MapReduce Programming expertise is no longer required. Furthermore, it not only supports batch processing but also real-time processing. MapReduce is no longer the only data processing option; other alternative data processing functions such as data standardization, master data management can now be performed natively in HDFS.

Overview of Hadoop Ecosystems



The components of the Hadoop ecosystem are shown in Figure.

There are components available in the Hadoop ecosystem for data ingestion, processing, and analysis.

Data Ingestion → Data Processing → Data Analysis

Components that help with Data Ingestion are:

1. Sqoop
2. Flume

Components that help with Data Processing are:

1. MapReduce
2. Spark

Components that help with Data Analysis are:

1. Pig
2. Hive
3. Impala

HDFS

It is the distributed storage unit of Hadoop. It provides streaming access to file system data as well as file permissions and authentication. It is based on GFS (Google File System). It is used to scale a single cluster node to hundreds and thousands of nodes. It handles large datasets running on commodity hardware. HDFS is highly fault-tolerant. It stores files across multiple machines. These files are stored in redundant fashion to allow for data recovery in case of failure.

HBase

It stores data in HDFS. It is the first non-batch component of the Hadoop Ecosystem. It is a database on top of HDFS. It provides a quick random access to the stored data. It has very low latency compared to HDFS. It is a NoSQL database, is non-relational and is a column-oriented database. A table can have thousands of columns. A table can have multiple rows. Each row can have several column families. Each column family can have several columns. Each column can have several key values. It is based on Google BigTable. This is widely used by Facebook, Twitter, Yahoo, etc.

Difference between HBase and Hadoop/HDFS

1. HDFS is the file system whereas HBase is a Hadoop database.
2. HDFS is WORM (Write once and read multiple times or many times). Latest versions support appending of data but this feature is rarely used. However, HBase supports real-time random read and write.
3. HDFS is based on Google File System (GFS) whereas HBase is based on Google Big Table.
4. HDFS supports only full table scan or partition table scan. Hbase supports random small range scan or table scan.
5. Performance of Hive on HDFS is relatively very good but for HBase it becomes 4-5 times slower.
6. The access to data is via MapReduce job only in HDFS whereas in HBase the access is via Java APIs, Rest, Avro, Thrift APIs.

7. HDFS does not support dynamic storage owing to its rigid structure whereas HBase supports dynamic storage.

8. HDFS has high latency operations whereas HBase has low latency operations.

9. HDFS is most suitable for batch analytics whereas HBase is for real-time analytics.

Hadoop Ecosystem Components for Data Ingestion

1. Sqoop: Sqoop stands for SQL to Hadoop. Its main functions are:

a) Importing data from RDBMS such as MySQL, Oracle, DB2, etc. to Hadoop file system (HDFS, HBase, Hive).

b) Exporting data from Hadoop File system (HDFS, HBase, Hive) to RDBMS (MySQL, Oracle, DB2).

Uses of Sqoop

a) It has a connector-based architecture to allow plug-ins to connect to external systems such as MySQL, Oracle, DB2, etc.

b) It can provision the data from external system on to HDFS and populate tables in Hive and HBase.

c) It integrates with Oozie allowing you to schedule and automate import and export tasks.

2. Flume: Flume is an important log aggregator (aggregates logs from different machines and places them in HDFS) component in the Hadoop ecosystem. Flume has been developed by Cloudera. It is designed for high volume ingestion of event-based data into Hadoop. The default destination in Flume (called as sink in flume parlance) is HDFS. However it can also write to HBase or Solr.

Hadoop Ecosystem Components for Data Processing

1. MapReduce: It is a programing paradigm that allows distributed and parallel processing of huge datasets. It is based on Google MapReduce. Google released a paper on MapReduce programming paradigm in 2004 and that became the genesis of Hadoop processing model. The MapReduce framework gets the input data from HDFS. There are two main phases: Map phase and the

Reduce phase. The map phase converts the input data into another set of data (key-value pairs). This new intermediate dataset then serves as the input to the reduce phase. The reduce phase acts on the datasets to combine (aggregate and consolidate) and reduce them to a smaller set of tuples. The result is then stored back in HDFS.

2. **Spark:** It is both a programming model as well as a computing model. It is an open-source big data processing framework. It was originally developed in 2009 at UC Berkeley's AmpLab and became an open-source project in 2010. It is written in Scala. It provides in-memory computing for Hadoop. In Spark, workloads execute in memory rather than on disk owing to which it is much faster (10 to 100 times) than when the workload is executed on disk. However, if the datasets are too large to fit into the available system memory, it can perform conventional disk-based processing. It serves as a potentially faster and more flexible alternative to MapReduce. It accesses data from HDFS (Spark does not have its own distributed file system) but bypasses the MapReduce processing.

Spark can be used with Hadoop coexisting smoothly with MapReduce (sitting on top of Hadoop YARN) or used independently of Hadoop (standalone). As a programming model, it works well with Scala, Python (it has API connectors for using it with Java or Python) or R programming language. The following are the Spark libraries:

a) Spark SQL: Spark also has support for SQL. Spark SQL uses SQL to help query data stored in disparate applications.

b) Spark streaming: It helps to analyze and present data in real time.

c) MLib: It supports machine learning such as applying advanced statistical operations on data in Spark Cluster.

d) GraphX: It helps in graph parallel computation.

Spark and Hadoop are usually used together by several companies. Hadoop was primarily designed to house unstructured data and run batch processing operations on it. Spark is used extensively for its high speed in memory computing and ability to run advanced real-time analytics where the two together have been giving good results.

Hadoop Ecosystem Components for Data Analysis

1. **Pig:** It is a high-level scripting language used with Hadoop. It serves as an alternative to MapReduce. It has two parts:

(a) **Pig Latin:** It is SQL-like scripting language. Pig Latin scripts are translated into MapReduce jobs which can then run on YARN and process data in the HDFS cluster. It was initially developed by Yahoo. It is immensely popular with developers who are not comfortable with MapReduce. However, SQL developers may have a preference for Hive.

How it works? There is a “Load" command available to load the data from “HDFS" into Pig. Then one can perform functions such as grouping, filtering, sorting, joining etc. The processed or computed data can then be either displayed on screen or placed back into HDFS.

It gives you a platform for building data flow for ETL (Extract, Transform and Load), processing and analyzing huge data sets.

(b) **Pig runtime:** It is the runtime environment.

2. **Hive:**

Hive is a data warehouse software project built on top of Hadoop. Three main tasks performed by Hive are summarization, querying and analysis. It supports queries written in a language called HQL or HiveQL which is a declarative SQL-like language. It converts the SQL-style queries into MapReduce jobs which are then executed on the Hadoop platform.

Difference between Hive and RDBMS

Both Hive and traditional databases such as MySQL, MS SQL Server, PostgreSQL support SQL interface. However, Hive is better known as a dataware house (D/W) rather than a database.

Let us look at the difference between Hive and traditional databases as regards the schema.

1. Hive enforces schema on Read Time whereas RDBMS enforces schema on Write Time. In RDBMS, at the time of loading/inserting data, the table's schema is enforced. If the data being loaded does not conform to the schema then it is rejected. Thus, the schema is enforced on write (loading the data into the database). Schema on write takes longer to load the data into the database; however it makes up for it during data retrieval with a good query time performance. However, Hive does not enforce the schema when the data is being loaded into the D/W. It is enforced only when the data is being read/retrieved. This is called schema on read. It definitely makes for fast initial load as the data load or insertion operation is just a file copy or move.

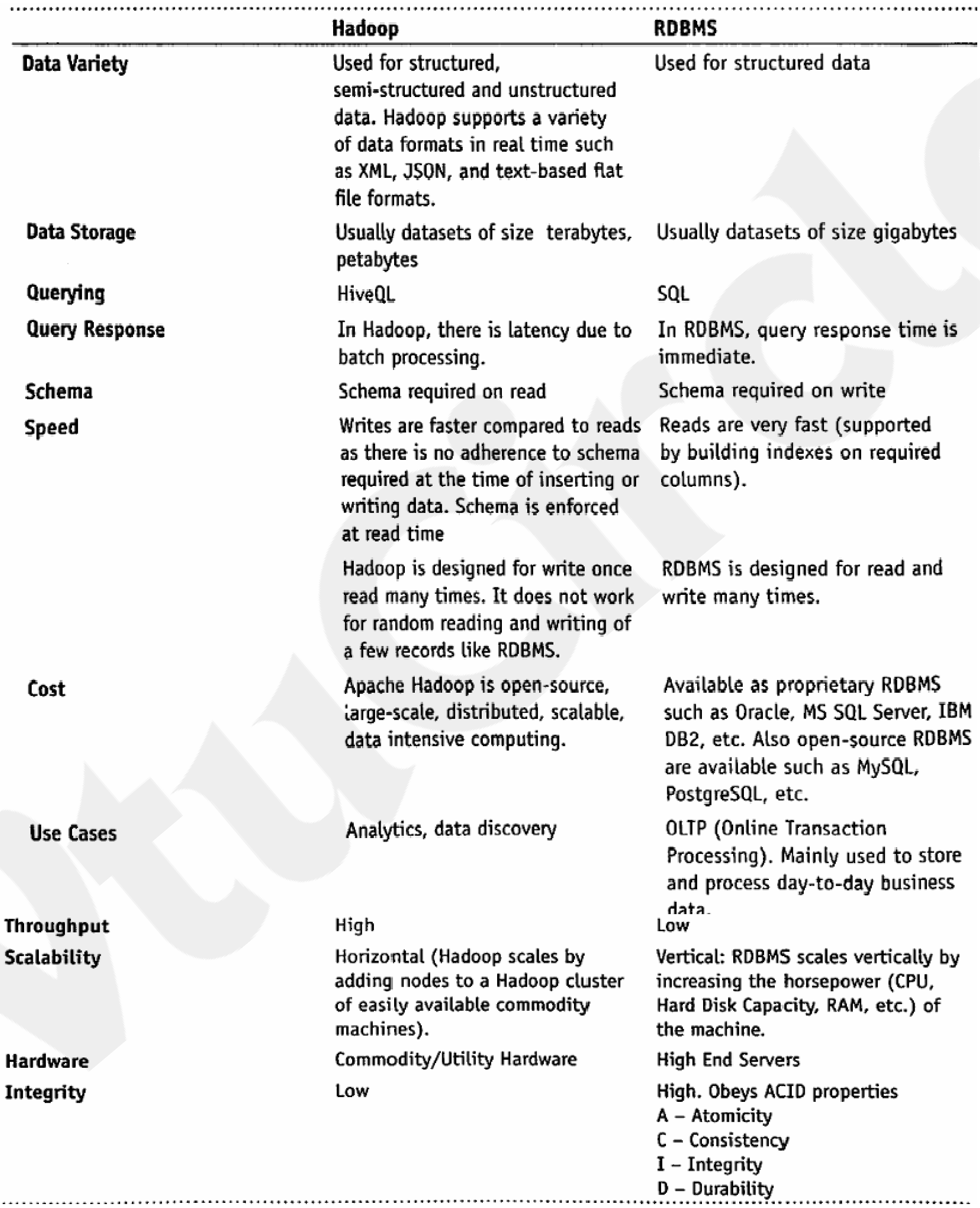
2. Hive is based on the notion of write once and read many times whereas the RDBMS is designed for read and write many times.

3. Hadoop is a batch-oriented system. Hive, therefore, is not suitable for OLTP (Online Transaction Processing) but, although not ideal, seems closer to OLAP (Online Analytical Processing). The reason being that there is quite a latency between issuing a query and receiving a reply as the query written in HiveQL will be converted to MapReduce jobs which are then executed on the Hadoop cluster. RDBMS is suitable for housing day-to-day transaction data and supports all OLTP operations with frequent insertions, modifications (updates), deletions of the data.

4. Hive handles static data analysis which is non-real-time data. Hive is the data warehouse of Hadoop. There are no frequent updates to the data and the query response time is not fast. RDBMS is suited for handling dynamic data which is real time.

5. Hive can be easily scaled at a very low cost when compared to RDMS. Hive uses HDFS to store data, thus it cannot be considered as the owner of the data, while on the other hand RDBMS is the owner of the data responsible for storing, managing and manipulating it in the database.

6. Hive uses the concept of parallel computing, whereas RDBMS uses serial computing.



	Hadoop	RDBMS
Data Variety	Used for structured, semi-structured and unstructured data. Hadoop supports a variety of data formats in real time such as XML, JSON, and text-based flat file formats.	Used for structured data
Data Storage	Usually datasets of size terabytes, petabytes	Usually datasets of size gigabytes
Querying	HiveQL	SQL
Query Response	In Hadoop, there is latency due to batch processing.	In RDBMS, query response time is immediate.
Schema	Schema required on read	Schema required on write
Speed	Writes are faster compared to reads as there is no adherence to schema required at the time of inserting or writing data. Schema is enforced at read time	Reads are very fast (supported by building indexes on required columns).
	Hadoop is designed for write once read many times. It does not work for random reading and writing of a few records like RDBMS.	RDBMS is designed for read and write many times.
Cost	Apache Hadoop is open-source, large-scale, distributed, scalable, data intensive computing.	Available as proprietary RDBMS such as Oracle, MS SQL Server, IBM DB2, etc. Also open-source RDBMS are available such as MySQL, PostgreSQL, etc.
Use Cases	Analytics, data discovery	OLTP (Online Transaction Processing). Mainly used to store and process day-to-day business data.
Throughput	High	Low
Scalability	Horizontal (Hadoop scales by adding nodes to a Hadoop cluster of easily available commodity machines).	Vertical: RDBMS scales vertically by increasing the horsepower (CPU, Hard Disk Capacity, RAM, etc.) of the machine.
Hardware	Commodity/Utility Hardware	High End Servers
Integrity	Low	High. Obeys ACID properties A – Atomicity C – Consistency I – Integrity D – Durability

Difference between Hive and HBase

1. Hive is a MapReduce-based SQL engine that runs on top of Hadoop. HBase is a key-value NoSQL database that runs on top of HDFS.

2. Hive is for batch processing of big data. HBase is for real-time data streaming.

Impala

It is a high-performance SQL engine that runs on Hadoop cluster. It is ideal for interactive analysis. It has very low latency measured in milliseconds. It supports a dialect of SQL called Impala SQL.

ZooKeeper

It is a coordination service for distributed applications.

Oozie

It is a workflow scheduler system to manage Apache Hadoop jobs.

Mahout

It is a scalable machine learning and data mining library.

Chukwa

It is a data collection system for managing large distributed systems.

Ambari

It is a web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters.

Hadoop Distributions

Hadoop is an open-source Apache project. Anyone can freely download the core aspects of Hadoop. The core aspects of Hadoop include the following:

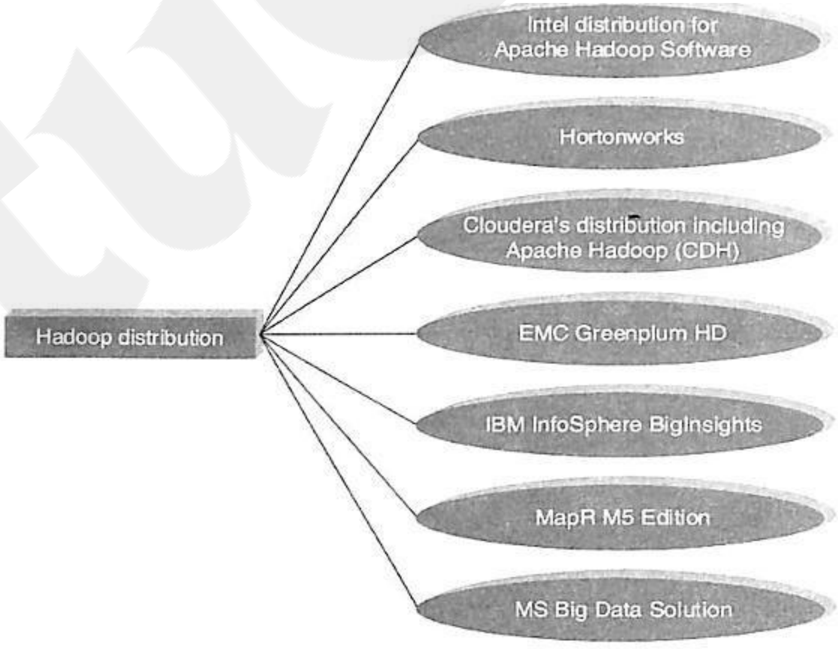
1. Hadoop Common

2. Hadoop Distributed File System (HDFS)

3. Hadoop YARN (Yet Another Resource Negotiator)

4. Hadoop MapReduce

There are few companies such as IBM, Amazon Web Services, Microsoft, Teradata, Hortonworks, Cloudera, etc. that have packaged Hadoop into a more easily consumable distributions or services. Although each of these companies have a slightly different strategy, the key essence remains its ability to distribute data and workloads across potentially thousands of servers thus making big data manageable data. A few Hadoop distributions are given in Figure below.


Hadoop distribution
Intel distribution for Apache Hadoop Software
Hortonworks
Cloudera's distribution including Apache Hadoop (CDH)
EMC Greenplum HD
IBM InfoSphere BigInsights
MapR M5 Edition
MS Big Data Solution

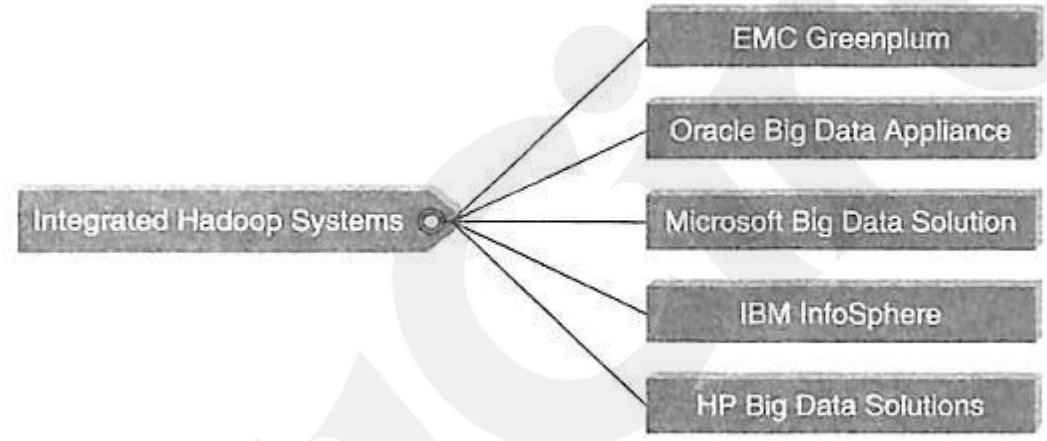
Difference between Hadoop v/s SQL

Feature	Hadoop	SQL
Technology	Modern	Traditional
Volume	Usually in PetaBytes	Usually in GigaBytes
Operations	Storage, processing, retrieval and pattern extraction from data	Storage, processing, retrieval and pattern mining of data
Fault Tolerance	Hadoop is highly fault tolerant	SQL has good fault tolerance
Storage	Stores data in the form of key-value pairs, tables, hash map etc in distributed systems.	Stores structured data in tabular format with fixed schema in cloud
Scaling	Linear	Non linear
Providers	Cloudera, Horton work, AWS etc. provides Hadoop systems.	Well-known industry leaders in SQL systems are Microsoft, SAP, Oracle etc.
Data Access	Batch oriented data access	Interactive and batch oriented data access
Cost	It is open source and systems can be cost effectively scaled	It is licensed and costs a fortune to buy a SQL server, moreover if system

Feature	Hadoop	SQL
		runs out of storage additional charges also emerge
Time	Statements are executed very quickly	SQL syntax is slow when executed in millions of rows
Optimization	It stores data in HDFS and process though Map Reduce with huge optimization techniques.	It does not have any advanced optimization techniques
Structure	Dynamic schema, capable of storing and processing log data, real-time data, images, videos, sensor data etc.(both structured and unstructured)	Static Schema, capable of storing data(fixed schema) in tabular format only(structured)
Data Update	Write data once, read data multiple times	Read and Write data multiple times
Integrity	Low	High
Interaction	Hadoop uses JDBC(Java Database Connectivity) to communicate with SQL systems to send and receive data	SQL systems can read and write data to Hadoop systems

Feature	Hadoop	SQL
Hardware	Uses commodity hardware	Uses propriety hardware
Training	Learning Hadoop for entry-level as well as seasoned profession is moderately hard	Learning SQL is easy for even entry-level professionals

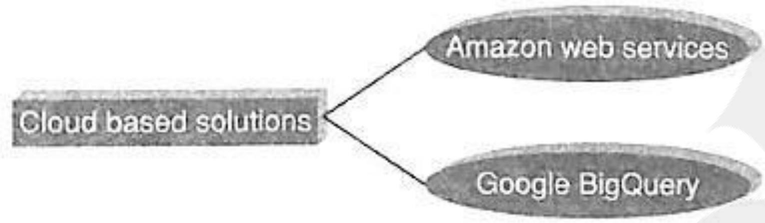
Integrated Hadoop Systems Offered by Leading Market Vendors


Integrated Hadoop Systems
EMC Greenplum
Oracle Big Data Appliance
Microsoft Big Data Solution
IBM InfoSphere
HP Big Data Solutions

Cloud-Based Hadoop Solutions

Amazon Web Services holds out a comprehensive, end-to-end portfolio of cloud computing services to help manage big data. The aim is to achieve this and more along with retaining the emphasis on reducing costs, scaling to meet demand, and accelerating the speed of innovation.

The Google Cloud Storage connector for Hadoop empowers one to perform MapReduce jobs directly on data in Google Cloud Storage, without the need to copy it to local disk and running it in the Hadoop Distributed File System (HDFS). The connector simplifies Hadoop deployment, and at the same time reduces cost and provides performance comparable to HDFS, all this while increasing reliability by elimi- nating the single point of failure of the name node.


Cloud based solutions
Amazon web services
Google BigQuery