UNIT VI

CHAPTER 6

NoSQL Database

Syllabus

- Introduction to NoSQL Database
- Types and examples of NoSQL Database- Key value store, document store, graph, Performance
- Structured verses unstructured data
- Distributed Database Model
- CAP theorem and BASE Properties
- Comparative study of SQL and NoSQL
- NoSQL Data Models
- Case Study - unstructured data from social media
- Introduction to Big Data, Hadoop : HDFS, MapReduce

Syllabus Topic : Introduction to NoSQL Database

6.1 Introduction to NoSQL Database

The relational databases are widely used in software industry. The design of relational databases is not such that which can cope with the scale and agility challenges that face modern real time applications, nor were they built to obtain benefit of the commodity storage and processing power available now a day. Now a days the data management becomes very difficult because of the tremendously increase in the size of data in various emerging fields like social networking, e-commerce etc.

- NoSQL is also known as "non SQL" or "non relational" or "Not Only SQL". NoSQL is database which provides a complete different mechanism for storing and retrieval of data which is modelled in means other than the tabular relations used in relational database management systems.
- Sometimes the term NOSQL seems to be confusing as handling data without SQL is out of imagination for some people. But actually the meaning of SQL is Not Only SQL.
- NoSQL challenges the dominance of relational databases. NoSQL databases are increasingly used in big data and real-time web applications.
- In NoSQL the data structures used like key-value, column, graph or document are different from those used in traditional relational database system which makes some operations faster in NoSQL.
- Usually the data structures used by NoSQL databases are also more flexible than relational database system.
- NoSQL allows the insertion of data without a predefined schema (design). In this database, it is possible to make significant application changes in the system without having worry about service interruptions. Because of it, the speed of development increases, the integration of code becomes more reliable and time of database administration decreases.
- To enforce data quality controls, developer has to add application-side code. NoSQL supports validation rules to be applied on the database which helps user to control the data while maintaining the advantage of a dynamic schema.
- NoSQL databases are more concentrated on availability, partition tolerance, and speed for which they may compromise the consistency. The

basic reason of no wide adoption of NoSQL is use of low level query language rather than SQL.

History of NoSQL

- Such database have existed since year 1960, but not known as "NoSQL". Need of this database comes in picture with the rise of web related applications like Google, Facebook, Amazon etc.
- In 1998 Carlo Strozzi first introduced a lightweight, open source relational database system which did not expose the standard Structured Query Language (SQL) interface. Carlo Strozzi gives the name as "NoSQL" to it. He suggested that as the NoSQL is differ from the traditional relational model, it should be called as "NoREAL" means "No Relational".
- Ohan Oskarsson, then a developer at Last.fm, reintroduced the term *NoSQL* in early 2009 when he organized an event to discuss "open source distributed, non relational databases". The name attempted to label the emergence of an increasing number of non-relational, distributed data stores, including open source clones of Google's BigTable/ MapReduce and Amazon's Dynamo.
- Most of the early NoSQL systems did not attempt to provide atomicity, consistency, isolation and durability guarantees, contrary to the prevailing practice among relational database systems.
- Based on 2014 revenue, the NoSQL market leaders are MarkLogic, MongoDB, and Datastax. Based on 2015 popularity rankings, the most popular NoSQL databases are MongoDB, Apache Cassandra, and Redis.

Use NoSQL when needs are like

1. Decentralized applications (e.g. Web and mobile)
2. Continuous availability; no downtime
3. High velocity data (devices, sensors, etc.)
4. Data coming in from many locations
5. Structured data is available with some semi/unstructured data.
6. To maintain high data volumes; retain forever.

What Is NoSQL ?

- NoSQL systems are also called as "Not only SQL" which indicates that they may support query languages like SQL.
- NoSQL is not depending on column, rows or schema for structure, it is no-relational database management systems. The data models of NoSQL are more flexible.
- NoSQL provides scalability, availability and fault tolerance and emerges as a alternative for relational database.
- The speciality of NoSQL is that, it may not require fixed table schemas avoids join operations, and also scale horizontally.
- Developers are working with applications that create massive volumes of new, rapidly changing data types - structured, semi-structured, unstructured and polymorphic data. NoSQL is useful for data which is growing far more rapidly or unstructured data or data which does not store in the relational schemas of RDBMS. There are common types of unstructured data: user and session data; chat, messaging, and log data; time series data such as IoT and device data; and large objects such as video and images.

Features of NoSQL

- Design simplicity.
- Simpler "horizontal" scaling to clusters of machines. This was a problem in relational databases.
- More control over data availability.

Observations regarding NoSQL

- It does not use the relational model.
- It runs well on clusters.
- NoSQL is Mostly open-source.
- It is Schema-less.

Why NoSQL ?

- All IT professionals and industry database experts come to know that NoSQL is here to stay.
- A recent study performed on NoSQL market growth forecasts a very strong compound annual growth rate of 21 percent for NoSQL technology from 2013-2018. It shows bright future for NoSQL.

- NoSQL databases has been much more clearly articulated today. NoSQL database refers to groups of databases that are not based on relational database model.
- The data storage model used by NoSQL database is not some fixed data model, but the common features among the NoSQL database is that the relational and tabular database model of SQL based database is not used.

Advantages of NoSQL

We cannot consider that NoSQL databases are straight substitution for relational database management system (RDBMS). But for many issues regarding data, NoSQL seem to be better.

1. **Data storage :** NoSQL databases supports storing data "in the form of Key value pair which give ability to store simple data structures. The document NoSQL database provides the ability to handle a range of flat or nested structures.
2. **Support for unstructured text :** NoSQL databases can handle unstructured text easily. This ability increases information effectively and can help organizations make better decisions.
3. **Ability to handle change over time :** NoSQL databases are capable of managing changes because of the systematic storage system.
4. **No reliance on SQL magic :** SQL(Structured Query Language) is the predominant language which is used to write queries in relational database management systems. Even if several NoSQL databases provide support for SQL access, they do so for compatibility with existing applications like business intelligence (BI) tools. NoSQL is not dependent on SQL for processing. NoSQL databases support their own query languages that can support data processing.
5. **Ability to scale horizontally on commodity hardware :** NoSQL databases supports distribution of a database across several servers. Hence if there is requirement of more data storage, then number of servers can be increased and connect them to database cluster (horizontal scaling) making them work as a single data service.
6. **Breadth of functionality :** Near about all the relational databases support the same characteristics but in a slightly different way, so they are all similar. In contrast, the NoSQL databases come in different core types: key-value, document store and graph. Out of these types, the one select to suit our requirements is not hard.
7. **Support for multiple data structures :** There is requirement of simple as well as complex data structures. NoSQL databases provide support for a range of data structures. Key-value stores can handle Simple binary values, lists, maps, and strings. Document databases can manage highly complex parent-child hierarchal structures Graph stores can describe the web of interrelated information.
8. **Big data applications :** In some systems the data grows rapidly. Such big volume data can be easily handled by NoSQL databases.
9. **Database administration :** The NoSQL has data distribution and auto repair capabilities, simplified data models and fewer tuning and administration requirements. This leads to less requirement of hands-on management.
10. **Economy :** These databases are designed to be used with low-cost commodity hardware.

It is difficult for application developer to find match between the relational data structures and the in-memory data structures. Using NoSQL databases they can develop the system without having to convert in-memory structures to relational structures.

Disadvantages of NoSQL

1. No standard schema.
2. Less use of SQL.

Companies using NoSQL

Now a day's many companies using NoSQL. Some of them are :

- Google
- Facebook
- Adobe
- Foursquare
- Digg
- Vermont public radio
- LinkedIn
- Mozilla

Syllabus Topic : Types and Examples of NoSQL Database

6.2 Types and Examples of NoSQL Database

There have been various approaches to classify NoSQL databases, each with different categories and subcategories.

Following are some types of NoSQL :

1. Key Value Store
2. Document Store
3. Column Store
4. Graph

Syllabus Topic : Key Value Store

6.2.1 Key Value Store

SPPU - Oct. 16

University Question

Q. Explain key value store NOSQL data model. **(Oct. 2016(In sem), 5 Marks)**

- The Key-value (KV) store system uses the concept of associative array, as their fundamental data model. In this model, data is represented as a collection of key-value pairs. Every single item in the database is stored as an attribute name (or 'key'), together with its value.
- In NoSQL database, a table exists with two columns : one is the Key and the other is Value.
- The key in the key-value pair should not be repeated because it the unique identifier that helps to access the value associated with that key uniquely.
- The Key value stores allow the developer of application to store schema-less data.
- Key-value databases are the simplest form of the NoSQL databases. Other advanced models are mostly extensions to this key-value model.
- **Examples** include Memcached, Riak, ArangoDB, InfinityDB, Oracle NoSQL Database, Redis and dbm.
- All key-value databases are not similar; there are major differences between these databases. For example: Data in **MemcacheDB** is not persistent while in **Riak** it is persistant. Such features are useful when implementing some solutions. It is important to not only choose a key-value database based on your requirements, it is also important to choose which key-value database.
- Examples of key-value NoSQL database applications :
  - Dynamo
  - Redis
  - FairCom
  - MemcacheDB
  - Riak
- Four main core operations perform on key-value store :
  1. **Get(key) :** It return the single value of given key.
  2. **Put(key,value) :** It assigns value to key.
  3. **Multi-get(key1,key2,key3,..,keyn):** It returns multiple values of given multiple keys.
  4. **Delete(key) :** It deletes both key and value present in it.

Example

This is a simple example for key-value store. Here keys are the names of employees and values are their contact numbers.

Key	Value
Sam	(234) 567-8901
jack	(134)526-6845
Ron	(245)452-4584
kenny	(356)584-1458

Where key value store is used ?

Key-value databases can be used in many scenarios. Such as,

- **General Web / Computers**
  - User profiles
  - Article/blog comments
  - Emails
- **E-commerce**
  - Shopping cart contents
  - Product categories
  - Product details

Advantages of Key Value Store

- Key value store is the simplest type of NoSQL.
- Supports simple queries very efficiently.

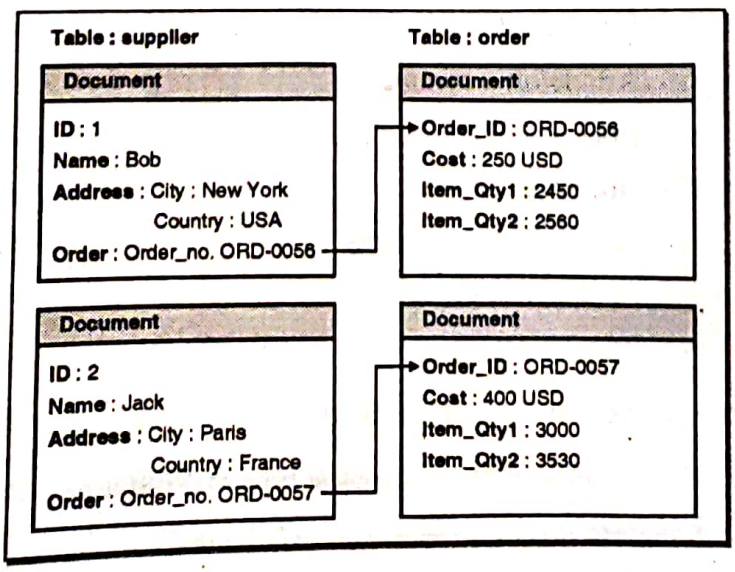
- Extended form of key-value stores is able to sort the keys.
- It is specially designed for storing data as a schema free data.
- Very simple data-modeling pattern should be understandable by anyone.
- With little or no maintained indexes, the key-value stores are designed to be more scalable and extremely fast.
- Suitable for system where data is not highly related.

Disadvantages of Key Value Store

- The indexing and scanning capabilities are absent. It does not help if we want to perform more operation as per user requirement than the basic CRUD (Create, Read, Update, Delete) operations.
- Only one row simple queries can be executed efficiently.
- Difficult to perform SQL operations like JOINS, GROUP BY etc.
- Selecting appropriate data type for value is difficult.
- Difficult to use constraints like FOREIGN KEY or NOT NULL.
- More application code is required to reassemble collections of key-value pairs into objects.

Syllabus Topic : Document Store

6.2.2 Document Store



- These databases store records as "documents" where a document can generally be thought of as a grouping of key-value pairs.
- The documents are identified by the unique keys which represents them. One defining characteristics of a document-oriented database is that in addition to the key lookup performed by a key-value store, the database offers an API or query language that retrieves documents based on their contents.
- Document databases are extension to key-value store. Addition to query capabilities of key-value databases, they provide indexing and the ability to filter documents based on attributes in the document.
- Examples of Document Store NoSQL database applications :
  - MongoDB
  - Couchbase

Advantages of Document Store

- The performance is good and the distribution across various servers becomes a lot easier.
- There is no need of translation between object in SQL and application. The object can directly be converted into document.
- They have strong indexing features and can rapidly execute different queries.

6.2.3 Column Store

- Column store is column oriented NoSQL database. Data is stored in cells grouped in columns of data rather than as rows of data. The logical grouping of columns is created in column families. There is no limitation on number of columns in a column family.
- These columns can be created at runtime. The operations like read write are performed using columns rather than rows.
- The column store structure gives the advantage of fast search / access and data aggregation over the row format data storage of relational databases.
- Relational databases store a single row as a continuous disk entry. Different rows are stored in different places on disk while column store database store all the cells related to a column as a

continuous disk entry which makes the search/access faster.

- **For example :** Displaying titles from a bunch of a million articles will be a tedious and time wasting task while using relational databases as it will go over each location to get item titles. While in column store, title of all the items can be obtained with just one disk access.
- Examples of column store NoSQL database applications :
  - HBase
  - BgTable
  - HyperTable

Advantages of Column Store

- Efficient storage and data compression.
- Fast data loads.
- Simple configuration.

Disadvantages of Column Store

- Queries with table joins can reduce high performance.
- Transactions are to be avoided or just not supported.

Syllabus Topic : Graph

6.2.4 Graph Store

- The Graph Store NoSQL database technology is designed to handle very large sets of data which may be structured, semi-structured or unstructured.
- In a Graph Base NoSQL Database, rigid format of SQL or the tables and columns representation does not exists. Rather a flexible graphical representation is used which is perfect to address scalability concerns. The graph structure contains edges, nodes and properties.
- The graph store is helpful in transferring the data from one model to other very easily.
- Graph store stores the entities and relationships between these entities. Entities are also known as nodes, which have properties. Nodes represent entities such as people, businesses, accounts, or any other item to be tracked.
- They are roughly the equivalent of the record, relation, or row in a relational database, or the document in a document database.
- Relations are known as edges that can have properties.

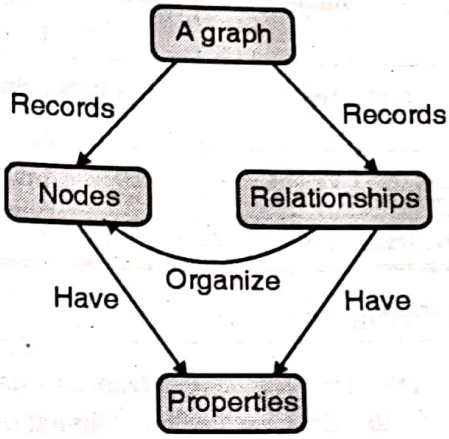


Fig. 6.2.1

- Edges are denoting directions. Nodes are generally organized with the help of relationships. The data is stored once by the organization of graph. This data can be interpreted in different ways depending upon the relationships between nodes.
- Key points related to graph store.
- Edges and nodes are used by these databases to represent and store data.
- The nodes are organised with the help of some relationships in between them, which is represented by edges between the nodes.
- Both the nodes and the relationships have some definite properties.
- Examples of Graph store NoSQL database applications :
  - Neo4j
  - Polyglot

Advantage of Graph Store

- Performance
- Flexibility
- Agility(ability to move quickly or easily)

Comparison of all the four NoSQL Databases

Database model	Performance	Scalability	Flexibility
Key value store database	High	High	High
Column store database	High	High	Moderate
Document store database	High	Variable(high)	High
Graph database	Variable	Variable	High

Syllabus Topic : Performance

6.3 Performance

- NoSQL is basically an advanced database developed to deal with the limitations of traditional relational databases in meeting Big Data demands. NoSQL represent various technologies which consider the three V's of big data.
  1. The exponentially growing volume of data.
  2. The velocity in which this data needs to be processed.
  3. The great variety of data being generated by today's applications.
- These demands require a high performance database.
- The data models of NoSQL and RDBMS are different. As we have seen NoSQL systems can be divided into four distinct groups :
  1. Key-value stores
  2. Document-oriented stores
  3. Column family stores
  4. Graph Store
- The database of NoSQL is distributed on different machines. The data models of key-value stores, document stores, and column family stores are key oriented. Hence there are two partition (distribution) strategies which are based on keys.
  1. In the first strategy, the datasets are distributed by the range of their keys. The keyset is split into blocks by the routing server. These blocks are allocated to different nodes. Then one node is assigned to handle the specific key range. Client contacts the routing server to find certain key to get the partition table.
  2. The second strategy is used to provide more availability and simpler cluster architecture. The performance of NoSQL increases because of load balancing. Because of replication, the failing nodes can be easily replaced which provider better availability and durability.
- Performance is an important factor. To evaluate performance of data storage solutions, specific scenarios are used. The general operations performed by applications that use the data store are simulated by these scenarios.
- There are certain advantages of NoSQL which effectively enhance its performance.
- Expressive query language which allows developers to build powerful features with data.
- Secondary indexes offer powerful indexes for quick data navigation along with the benefits of NoSQL.
- Scalability across inexpensive commodity hardware so one can easily grow the application to meet rising demands.
- Flexible data model that allows to quickly adapt the changing needs of your business.

Syllabus Topic : Structured verses Unstructured Data

6.4 Structured verses Unstructured Data

Now a day, the Structured and unstructured data are both used effectively in big data analysis. Few years ago processing capability was limited, memory was inadequate, and cost of data-storage was high. Hence, structured data was the only option to manage data effectively. In recent times, the use of unstructured data increases due to increased availability of storage and the sheer number of complex data sources.

6.4.1 Structured Data

- It is considered that about twenty percent of data in business transactions is in structural form. This structured data is easy to store and manage in

different databases. The structured data can be easily ordered and processed by data mining tools.

- Structured data is the traditional data which consist of very well-organized information. It concerns all data which can be stored in database in the format of tables with rows and columns.
- Sometimes the structured data can be machine generated. For example data regarding heart rate, blood pressure that comes from medical devices, data like rotation per minute, temperature which come from manufacturing sensors or web server logs (number of times a page is visited).
- The Structured data can also be generated by human. For example personal information of an employee line id, name salary etc.
- The data has the advantage of being easily entered, stored, queried and analyzed. The structured data is always easy to compare For Example - Salary of Manager is more than the Clerk. Kunal got more marks than Rahul.
- Some more examples of structured data :
  - Customer data
  - Sales data

6.4.2 Unstructured Data

- The unstructured data is the form of information which does not have any predefined data model. Simply the unstructured data means the data which is not well defined. This data is not suitable for traditional relational model. It is usually text-heavy, but also contains data like as dates, numbers etc. Such data is difficult to understand and manage using traditional programs because it generates irregularities and ambiguities. In social media maximum of data is in unstructured form. The most big data sources like WhatApp, Facebook, twitter have unstructured data. It is difficult to run any analytics on such data. To analyze such data we need to convert it into structured form.
- In 1998, Merrill Lynch cited a rule of thumb that somewhere around 80-90% of all potentially usable business information may originate in unstructured form. This rule of thumb is not based on primary or any quantitative research, but nonetheless is accepted.
- Data with some form of structure may still be characterized as unstructured if its structure is not helpful for the processing task easily.
- For unstructured data RDBMS is not suitable or it is not fit in it.
- The example of textual unstructured data is email messages, PowerPoint presentations, Word documents, chat data etc. Non-textual unstructured data is in the form like JPEG images, MP3 audio files and Flash video files.

6.4.3 Comparison between Structured and Unstructured Data

Sr. No.	Structure Data	Unstructured Data
1	Structured data contain well defined content.	Unstructured data not containing well defined data.
2	Structured data is easily understood.	Unstructured data has to be processed to understand.
3	Data is stored in RDBMS.	RDBMS is not good fit for data.
4	In structured data, content is typically in text format.	In unstructured data, content is in the form of Text, Images, Audio, Video, Documents.
5	Representation is in discrete row and columns form.	Representation is less defined.
6	Data is stored in file format.	Data is in unmanaged file structure.
7	Example : Customer data Sales data	Example : Images Video

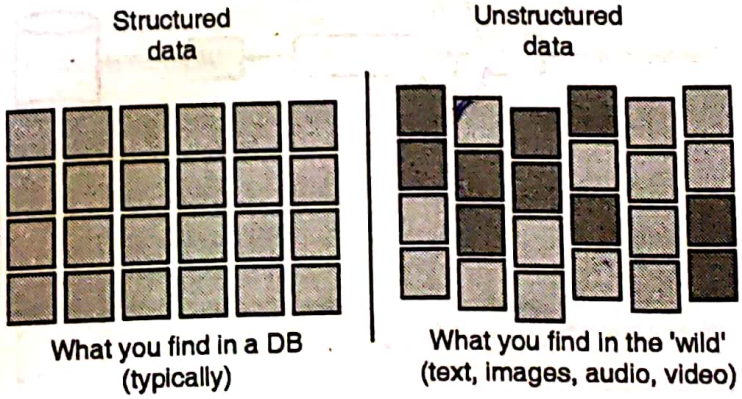


Fig. 6.4.1 : Structured and unstructured data

Syllabus Topic : Distributed Database Model

6.5 Distributed Database Model

A database system can be viewed as consisting of three software modules: a transaction manager (TM), a data manager (DM), and a scheduler.

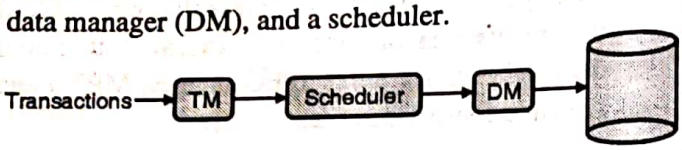


Fig. 6.5.1

- The transaction manager examines the execution of a transaction. It intercepts and executes all the transactions which have been submitted. Thus, the Transaction Manager is the mediator between users and the database system.
- Concurrency control is enforced by the scheduler. It grants or releases locks on data objects as per the requests of a transaction.
- The database is managed by the data manager. It performs the read-write requests issued by the transaction manager on behalf of a transaction by operating them on the database. The failure recovery is responsibility of data manager. Thus, the DM is the interface between the scheduler and the database.
- The concurrency control model of a distributed database system is shown in Fig. 6.5.2.

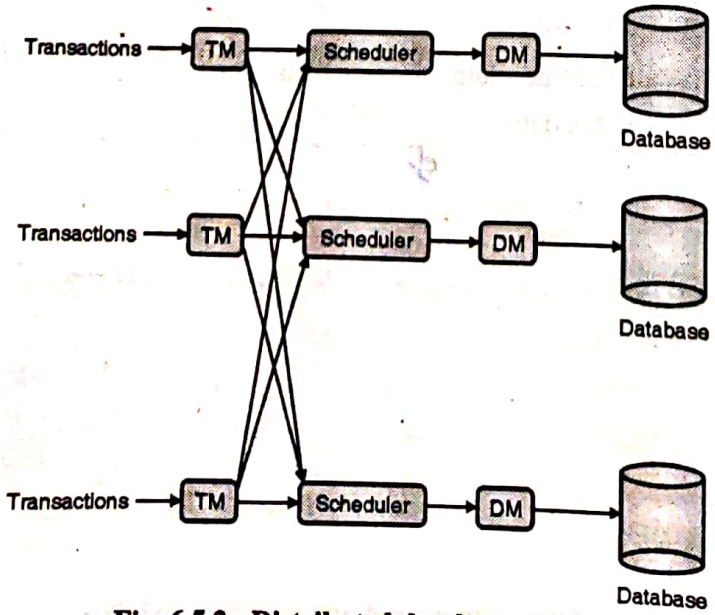


Fig. 6.5.2 : Distributed database system

Syllabus Topic : CAP Theorem and BASE Properties

6.6 CAP Theorem and BASE Properties

6.6.1 CAP Theorem

CAP stands for Consistency, Availability and Partitioning tolerance. In 2001 the CAP theorem was given by Eric Brewer, a professor at the University of California, Berkeley and one of the founders of Google, in the keynote of Principles of Distributed Computing.

Statement

- In general it is expected that every system should have Consistency, High-Availability and Partition-tolerance. But it is really hard for any system to achieve all these three properties at the same time.

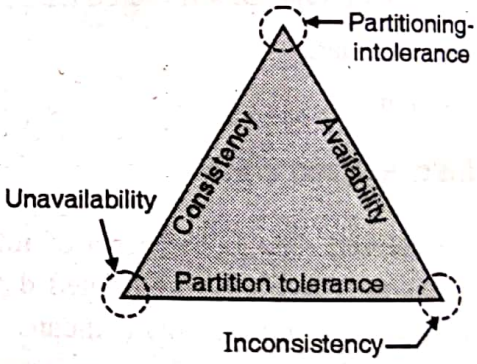


Fig. 6.6.1

- Maximum of two properties out of three can be achieved by the system. First we will see these three properties.

Consistency

- The system should follow the rule of sequence of updates which is present across all replicas in a cluster.
- Regardless of the location the data should be consistent. For example if client1 writes data in sequence A, B then another clients should be able to read the data in same order written by client1. It is also known as strong consistency.

Availability

- A service should be available to operate fully. It should be guaranteed that every request receives a

response about the status (successful or failed). It is also possible that a system which is not available may be consistent. It is hard to achieve *consistency* and *availability* at the same time.

- Both are contradictory to each other, i.e. if consistency is relaxed then system is highly available under the partitioning conditions (see next definition) and *strong consistency* means that in some specific conditions the system will not be available.

Partition Tolerance

- If failure may occur in any part of the system, the entire system does not get shut down.
- For example, consider cluster of n replicated nodes and network is unavailable among some number of nodes because of some reasons like network cable got chopped. Because of this synchronization of data is not possible.
- Here some part of system is in working condition while the other is not. If there is partition in the network, then there is possibility to lose consistency as we allow updates to both sides of the partition. Or we lose availability as we may shut down the system because of error until condition is resolved.
- A simple meaning of this theorem is "It is impossible for a protocol to guarantee both consistency and availability in a partition prone distributed system". This was mentioned above in example.

Adjustment between Requirements

1. **Available and Partition-Tolerant :** You have two nodes and the link between the two is severed. Since both nodes are up, the system can be designed to accept requests on each of the nodes. This insures the availability of system even if the network is partitioned. However independent results are issued by each node. Here high availability and partition tolerance is provided by compromising consistency.
2. **Consistent and Partition-Tolerant :** You have three nodes and one node loses its link with the other two. A rule cab made that a result will be returned only when a majority of nodes agree. So, even if the partition is there, the system will return a consistent result. Although the separated node is up, it won't be available since it is not able to reach consensus.
3. Finally, a system can be both *consistent and available*, but it possible that it may have to block on a partition. Most of the NoSQL database system architectures favour one factor over the other.

6.6.2 BASE Properties SPPU - May 16

University Question

Q. Explain BASE properties of NOSQL database with suitable example. **(May 2016, 5 Marks)**

- In the Chapter 4, we have seen the ACID properties of DBMS. ACID properties are an important concept for databases.
- Let's recall in brief what ACID means in traditional RDBMS community before moving to the BASE Properties.

The four properties are as follows.

- **Atomicity :** This property states that, either all operations contained by a transaction are done successfully or none of them complete at all.
- **Consistency :** The consistency property ensures that the transaction executed on the database system will bring the database from one valid state to another.
- **Isolation :** In case of concurrent transactions, the isolation property ensures that the system state should be same that would be obtained if transactions were executed sequentially.
- **Durability :** The durability property assures that after transaction committed successfully the updates made should remain permanent in the database even in the event of power loss, crashes, or errors.

ACID provides *strong consistency* (synchronous transactions) for partitioned databases and thus provides less availability.

Consistency is always preferable, but it is no always available.

- Base Property means Basically Available, Sof state, Eventual consistency.
- Consistency is always preferable, but it is no always available. In the NoSQL world, ACI transactions are less suitable as some DBMS d not have requirements for strict consistency, dat freshness and accuracy in order to gain othe benefits, like scale and resilience. By considerin this, the BASE Consistency model is developed.

- The BASE properties are as follows :
  1. **Basic Availability :** Most of the time the database appears to work.
  2. **Soft-state :** It is not necessary that the stores should be write-consistent. Also the different replicas have to be mutually consistent all the time.
  3. **Eventual consistency :** Stores may show the consistency at some later point. Eventual consistency which is normally asynchronous in nature is a form of a weaker consistency which improves speed and availability.
- The BASE (Basically Available, Soft state, Eventual consistency) is the opposite of ACID.
- ACID is pessimistic i.e. consistency is required at the end of every operation. BASE is optimistic, i.e. it accepts that there is uncertainty in consistency.
- A BASE Model focus on availability as it is important for scale, but it doesn't offer guaranteed consistency of replicated data at write time.
- At the end we can say the BASE model of consistency provides a less strict assurance than **ACID :** Data will be consistent in the future, either at read time or may be always consistent for some points.

Syllabus Topic : Comparative Study of SQL and NoSQL

6.7 Comparative Study of SQL and NoSQL

For managing the database SQL has been most widely used programming language over the last few decades. It is a Relational Database Management System. But in today's era NoSQL has arises for an option to the SQL. There is very high difference between SQL and NoSQL. In this topic we will see the comparative study of SQL and NoSQL.

The conceptual difference is :

A framework of a relational database which is setup with the defined categories used by SQL. The tables of SQL are idle for storing data which is structured. The structured data like name, address fits into the SQL format perfectly.

But when data is unstructured it needed another format which is not dependant on the relationships of the data. When this situation occurs that time we can use NoSQL. NoSQL allows for the storage of an unstructured data without categorizing the data into fixed tables. NoSQL database scales horizontally. NoSQL is also known as Non-relational database or distributed database.

Factual difference is

- In SQL databases are structured in the form of tables, but in NoSQL databases are structured in the form of documents, graphs, or key-value pairs.
- In SQL Database there is a standard definition of schema which must be worked with the structured data. While In NoSQL there is no standard definition for schema which must be worked with the structured data.
- SQL database have predefined schema for structured data while NoSQL database have dynamic schema for unstructured data.
- SQL databases has feature of vertical scaling while NoSQL databases has feature of horizontal scaling.
- SQL database are designed and managed with SQL (Structured Query language) while NoSQL database are designed and managed with the UnQL (Unstructured Query Language).
- The syntax of SQL does not vary with database, while the syntax of UnQL varies with database.
- SQL is preferable for handling complex query while NoSQL cannot handle complex query. SQL queries are more powerful than NoSQL queries.
- Examples of SQL databases are : MySQL, Oracle, MS-SQL while examples of NoSQL are : MongoDB, BigTable, Redis, Hbase, Neo4i and CouchDb.
- SQL cannot manage big data which is stored in hierarchical manner while NoSQL handles hierarchical data better than SQL. Hence, NoSQL is preferable to SQL when managing big data.

Comparision between SQL and NoSQL

Sr. No.	SQL	NoSQL
1.	SQL means structured query language.	NoSQL means NOT only SQL language.
2.	Data is in structured and organized form.	Data is in unstructured form.
3.	It is used for small to medium scale data set effectively.	It is used for large set of data.
4.	SQL is schema based.	NoSQL is schema less.
5.	Data is stored as row and column in table where each column is of specific type.	The data model is depending upon the database type.
6.	Relational database is table based.	Key value pair, storage, column, document store, graph database.
7.	Example : SQL server Oracle	Example : MongoDB HBase

Syllabus Topic : NoSQL Data Models

6.8 NoSQL Data Models

Data Model

Data model is the mode which organizes the data. It is a representation which is used to understand and manipulate the data. The data model helps to :

- Represent the data elements in analytical view.
- Maintain relationship of these elements.
- Describe the way by which we interact with the database.

In RDBMS the relational model was used to represent the data. The data is represented in the form of tables (combination of rows and columns). Each row represents some entity while columns represent relationships in the same table or with another table. The relational data modeling has been a well-defined discipline for many years. However now a days with the emergence of NoSQL databases, need of a new data model arises.

NoSQL Data Model

The NoSQL data model is different from traditional relational data model. There are different data models :

- Key-value
- Document
- Column - family
- Graph

The three model Key-value, document and column- family share common features of Aggregate Orientation.

NoSQL Agreegate Model

- In this model it considered that we have to manage more complex data compared to relational model. The complex data structure are in the form of map, list etc. The aggregate model uses this complex structure. Aggregate is a collection of data objects which are treated as a single unit to manage and manipulate. Atomic operations are expected to update aggregates. Using aggregate it is easy to work on cluster which is unit of machines. Aggregates helps application developer by solving the mismatch problem which usually occur in relational model.
- When the aggregate model runs on a cluster, it gives several advantages on computation power and data distribution. While gathering data, it requires minimizing the number of nodes. The aggregate gives an important view that which data should be stored together.
- The **Key-Value** and **Document** databases are strongly aggregate oriented. These databases contain number of aggregates with a key to get the data. In Key-Value we can store any type of object.
- The Column-Family has two level aggregate structure. The first key is the row identifier. The second level values are defined to as columns.

Important point related to NoSQL Data Aggregate Model

- All these models use aggregated index by a key.
- The key is used for searching the data.
- The Aggregate acts as a atomic unit for modification.

- In the document model, the document is treated as single unit of storage. This model makes document transparent for querying.
- In Column-Family model, the columns are divided into column families and treated as single units.
- All models improve the accessibility of data.

Syllabus Topic : Case Study - Unstructured Data from Social Media

Case study is in detail study of a phenomenon, like a person, group, or situation. The phenomenon is studied in detail, cases are analyzed and solutions or interpretations are presented.

6.9 Case Study - Unstructured Data from Social Media

In this case study we are going through following steps.

1. Background
2. Problem
3. Proposed Solution
4. Implementation
5. Results

1. Background

- The unstructured data is the form of information which does not have any predefined data model. This data is not suitable for traditional relational model. It is usually text-heavy, but also contains data like as dates, numbers etc.
- Such data is difficult to understand and manage using traditional programs because it generates irregularities and ambiguities.
- In social media maximum of data is in unstructured form. The most big data sources like WhatApp, Facebook, twitter have unstructured data. It is difficult to run any analytics on such data. To analyze such data we need to convert it into structured form.

2. Problem

For social media sites, it is really hard to organize such data. There are various problems to handle the data which lead to find out some solution. Social media contains different types of data some of which is important and some not.

- Chat History
- Posts of users
- Free advertisements posted by users
- Twits
- Comments

There is no any predefined or standard format of this data. Tweets, Facebook postings and other social comments have to be analyzed to determine the sentiment of the population. The business strategies are decided on this data.

3. Proposed Solution

To handle the unstructured data properly it should be converted into structured or semi structured format. But if volume of the data is very big then it is difficult to convert it. For such big data we can use the database framework like NoSQL and Software system like Hadoop.

There are two solutions for handling unstructured data :

A. Convert it into structured format.

B. Use database framework like NoSQL and Software system like Hadoop.

4. Implementation

A. Converting unstructured data into structured format

The unstructured data can be converted into semi-structured or structured data by converting the text into words and phrases which can fit into relational tables. Then it can be categorized. The analysis techniques can then be used to conclude and arrive at results. It is important that the type of data, the source and its general understanding has to be re-vamped. So does this process of converting unstructured to structured data may be manual or machine driven through algorithms. Algorithms reduce accuracy but increase scale.

B. Use database framework like NoSQL and Software system like Hadoop.

We have already seen the database framework NoSQL which can deal with such unstructured data. The database tools like Hadoop, MongoDB also are useful in dealing with unstructured data.

5. Results

Both above options are 100 percent effective in dealing with the vast social media unstructured data. Both options give a perfect solution to manage this data. Data management involves access and manipulation of data.

Syllabus Topic : Introduction to Big Data

6.10 Introduction to Big Data

Now a day the amount of data created by various advanced technologies like Social networking sites, E-commerce etc. is very large. It is really difficult to store such huge data by using the traditional data storage facilities.

Until 2003, the size of data produced was 5 billion gigabytes. If this data is stored in the form of disks it may fill an entire football field. In 2011, the same amount of data was created in every two days and in 2013 it was created in every ten minutes. This is really tremendous rate.

In this topic, we will discuss about big data on a fundamental level and define common concepts related to big data. We will also see in deep about some of the processes and technologies currently being used in this field.

Big Data

Big data means huge amount of data, it is a collection of large datasets that cannot be processed using traditional computing techniques. Big Data is complex and difficult to store, maintain or access in regular file system. Big Data becomes a complete subject, which involves different techniques, tools, and frameworks.

There are various sources of big data. Now a days in number of fields such huge data get created. Following are the some of fields.

- **Stock Exchange :** The data in the share market regarding information about prices and status details of shares of thousands of companies is very huge.
- **Social Media Data :** The data of social networking sites contains information about all the account holders, their posts, chat history, advertisements etc. On topmost sites like facebook and whatsapp, there are literally billions of users.
- **Video sharing portals :** Video sharing portals like youtube, Vimeo etc. contains millions of videos each of which require lot much of memory to store.
- **Search Engine Data :** The search engines like Google and Yahoo holds lot much of metadata regarding various sites.
- **Transport Data :** Transport data contains information about model, capacity, distance and availability of various vehicles.
- **Banking Data :** The big giants in banking domain like SBI or ICICI hold large amount of data regarding huge transactions of account holders.
- The data can be categorized in three types.
  1. **Structured Data :** This type of data is stored in relations(tables) in Relational Database Management System.
  2. **Semi-structured Data :** This type of data is neither raw data nor typed data in a conventional database system. A lot of data found on the web can be described as semi-structured data. This type of data do not have any standard formal model. This data is stored using various formats like XML and JSON.
  3. **Unstructured Data :** This is data do not have any pre-defined data model. The data of video, audio, Image, text, web logs, system logs etc. comes under this category.
- In general there are some important issues regarding data in traditional file storage system.
  1. **Volume :** Now a days the volume of data regarding different fields is high and potentially increasing day by day. Organizations collect data from a variety of sources, including business transactions, social media and information etc.
  2. **Velocity :** The configuration of system single processor, limited RAM and limited storage capacity system cannot store and manage high volume of data.
  3. **Variety :** The form of data from different sources is different.
  4. **Variability :** The flow of data coming from sources like social media is inconsistent

because of daily emerging new trends. It can show sudden increase in size of data which is difficult to manage.

5. **Complexity :** As the data is coming from various sources, it is difficult to link, match and transform such data across systems. It is necessary to connect and correlate relationships, hierarchies and multiple data linkages of the data.

- All these issues are solved by the new advanced **Big Data Technology.**

Big Data Technologies

- Big data technologies are based on the accuracy of analyzing the data. This leads to more operational efficiencies, reductions in cost, reduction in failure and data loss.
- To implement the Big Data Technology, there is requirement of strong infrastructure which can manage and process the big data. This data may be structured or unstructured. The infrastructure must be able to protect data privacy and security.

Characteristics of big data

- Big data stores and manages huge amount of data in time and cost effective manner.
- It can analyze data of any form like unstructured, structured or streaming.
- It can maintain multiple copies of the important data across clusters.
- The System Failure Mechanism of this technology is very strong which avoid the data loss.
- Data can be stored in blocks on different machines which can be merged any times as per requirement.
- It captures data from live events in real time.
- There are various technologies in the market from different vendors including Amazon, IBM, Microsoft, etc., to handle big data. There are two classes of Big Data technology which can handle big data.
  - Operational Big Data
  - Analytical Big Data

Operational Big Data

- It was developed to address the shortcoming of traditional database and it is faster and can deal with large quantity of data spread over multiple servers. We are also using cloud computing architectures to allow massive computation to run effectively as well as it is cost efficient. This has made Big Data workload easier to manage, faster to implement as well as cheaper. Here in addition to interaction with user it also provides artificial intelligence about the active data. For example in games the moves of user are studies and next course of actions are suggested.
- Systems like MongoDB and NoSQL comes under this category.
  - **MongoDB :** This system provides operational capabilities for interactive, real-time workloads where data is primarily captured and stored.
  - **NoSQL :** The new advanced cloud technology is used to implement the huge computations to be run efficiently and inexpensively. It helps to manage the data easily and fast. This Cloud Technology is used in NoSQL big data system.

Analytical Big Data

- These systems provides analytical capabilities for complex analysis which considers most or all of the data.
- For example Massively Parallel Processing (MPP) database systems and MapReduce.
- Analytical Big Data is addressed by MPP database systems and MapReduce. These technologies has evolved as a result of shortcoming in traditional database which deals with one servers only. On the other hand MapReduce provides new method of analyzing data which is beyond the scope of SQL.

Challenges regarding Big Data

There are various challenges relate to big data :

- Getting the source data
- Making correction in this data
- Storing the data
- Searching specific data depending upon some criteria
- Sharing data between machines and users
- Transferring data
- Analyzing the data

Syllabus Topic : HADOOP - HDFS, MapReduce

6.11 Hadoop

SPPU - Dec. 15

University Question

Q. What is Hadoop? (Dec. 2015, 2 Marks)

- **Hadoop** is an open source, Java-based programming framework which supports the processing and storage of extremely large sets of data in a distributed computing environment using simple programming models.
- Hadoop has very strong processing power and the ability to handle virtually unlimited parallel tasks.
- With the help of Hadoop, applications can be run on systems with thousands of commodity hardware nodes. It can handle thousands of terabytes of data. Hadoop has distributed file system which facilitates rapid data transfer rates among nodes. This allows the system to proceed even in case one or more nodes get failed. This approach avoids the unexpected data loss.
- Hadoop has quickly emerged as a foundation for big data processing tasks like scientific analytics of data, planning of business and sales, and processing enormous volumes of data including social media data.

History of Hadoop

- Computer scientists Doug Cutting and Mike Cafarella created Hadoop in 2006 to support distribution for the Nutch (search engine). The main aim is to increase the speed of search results by the distribution of data and implement calculations on different computers by multitasking.
- Later on Cutting joined Yahoo but him still works on the Nutch project with the ideas based on Google's early work with automating distributed data storage and processing.
- The Nutch Prject divided into two parts –
  - Nutch - Web crawler portion
  - Hadoop - Distributed computing and processing portion
- In 2008, Yahoo released Hadoop as an open-source project. Now Apache Software Foundation (ASF) manages the Hadoop's framework and ecosystem of technologies.

6.11.1 Modules (Components) of Hadoop

SPPU - Dec. 14, Dec. 15, May 16, Dec. 16

University Questions

Q. Explain different components of HADOOP. (Dec. 2014, Dec. 2015 7 Marks)

Q. Explain in brief different building blocks of HADOOP. (May 2016, Dec. 2016, 7 Marks)

- **HDFS : HDFS** stands for Hadoop Distributed File System. It states that the files will be broken into blocks and stored in nodes over the distributed architecture. It provides high-throughput access to application data.

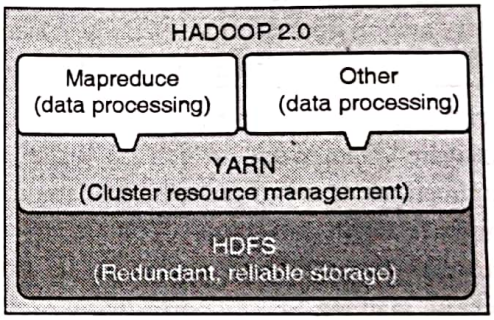


Fig. 6.11.1

- **Yarn :** Yarn stands for "Yet another Resource Negotiator". It is used for job scheduling and managing the cluster (multiple nodes).
- **Map Reduce :** This is YARN-based system for parallel processing of large data set using key value pair. The Map task takes input data and converts it into a data set which can be computed in Key value pair.
- **Hadoop Common :** These Java libraries and utilities are used to start Hadoop. These are used by other Hadoop modules. These libraries provide file system and OS level abstractions.

Advantages of Hadoop

1. Huge amounts of any kind of data can be stored and processed quickly.
2. **Computing power :** Hadoop's distributed computing model processes big data fast.
3. **Fault Tolerance :** In case of failure of any node the tasks are automatically redirected to other nodes.

4. **Flexibility :** Any kind of unstructured data like text, images and videos can be stored.
5. **Low cost :** This open-source framework is free.
6. **Scalability :** New nodes can be easily added to handle big tasks.

Disadvantage of Hadoop

1. Hadoop is rough in manner because the software is under active development.
2. Programming model is very restrictive.
3. Joins of multiple datasets are tricky and slow.
4. Cluster management is hard : In the cluster, operations like debugging, distributing. software, collection logs etc are too hard.
5. Requires care and may limit scaling.

6.11.1.1 MapReduce SPPU - May 15, May 16

University Question
Q. Write a short note on : MapReduce in Hadoop.
(May 2015, May 2016, 5 Marks)

- MapReduce is an important part of Hadoop. It is a software framework which is used to write applications easily to process huge amount of data (multi-terabyte data-sets) simultaneously on large clusters (thousands of nodes) in reliable, fault-tolerant manner.
- MapReduce basically referes to two tasks performed by the Hadoop programs. One is map and another is reduce.

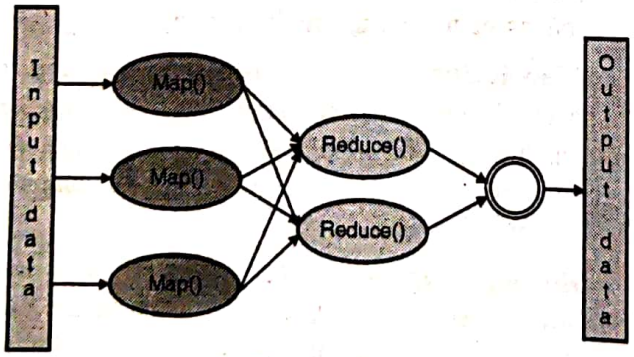


Fig. 6.11.2

- Hadoop programs perform following two tasks on **MapReduce** :
  - **The Map Task :** This is the first task, which takes a set of data and converts it into another set of data in which individual elements are broken down into tuples (key/value pairs).
  - **The reduce :** This job takes the output of previously executed map task as input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job.
- In MapReduce framework, the data of input and output is stored in a file system. The framework handles the scheduling of all the tasks, monitoring these tasks and if fails, re-executes them.
- The main advantage of MapReduce is that it is simple to scale data processing over multiple computing nodes. The data processing primitives are known as mappers and reducers in the MapReduce model.
- The MapReduce framework consists of single master JobTracker and one slave TaskTracker per cluster-node.
- **Master JobTracker :** The tasks under master are -
  - Managing the resources.
  - Tracking consumption and availability of resources.
  - Scheduling the jobs component tasks on the slaves.
  - Monitoring the tasks and re-executing the failed tasks.
- **The slaves TaskTracker :** It execute the tasks as per the directions of the master and provide task-status information to the master periodically.
- The JobTracker is very important in Hadoop MapReduce service. If JobTracker goes down, all running tasks get halted.

Advantages of MapReduce

1. Scalable.
2. Fault tolerant.
3. Simple coding model.
4. Supports unstructured data.

6.11.1.2 Hadoop Distributed File System (HDFS)

- The HDFS is the primary storage system used by Hadoop applications. HDFS is a distributed file system and a framework provided by Hadoop for the analysis and transformation of huge data sets which uses the MapReduce paradigm. The HDFS is based on Google File System (GFS). It provides

high-performance access to data across Hadoop clusters (thousands of computers), HDFS has become a key tool for managing pools of big data and supporting big data analytics applications.

HDFS is usually deployed on commodity hardware of low-cost where the possibility of server failures is common. The file system is designed to be highly fault-tolerant. The HDFS facilitates the rapid transfer of data between different computer nodes and enables Hadoop systems to proceed its execution even if one or more nodes get failed. That decreases the risk of catastrophic failure, even in the event that numerous nodes fail.

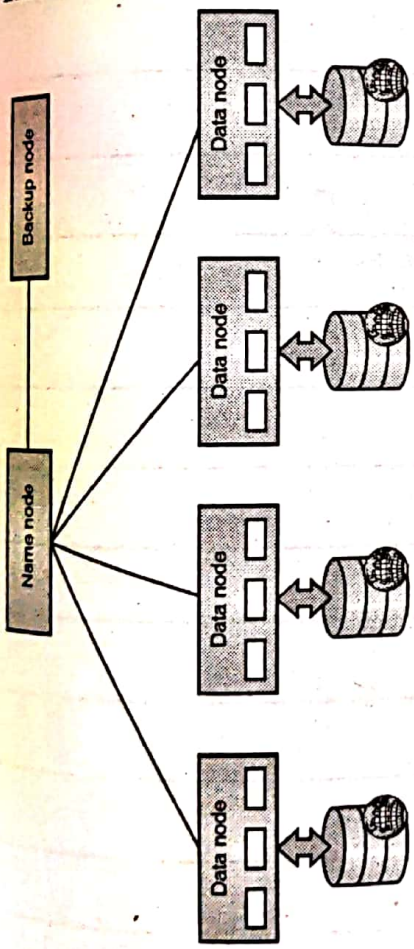


Fig. 6.11.3

- The architecture used by HDFS is known as master/slave architecture.
- NameNode which manages the metadata of file system and DataNode which stores the actual data.
- The HDFS namespace is a hierarchy of files and directories. Inodes are used to represent these file and directories. Inodes are used to record attributes such as permissions, modification and access times etc. The file content is split into large blocks and each block of the file is independently replicated at multiple DataNodes.
- The tree structure of namespace is maintained by the NameNode. It maps the blocks to DataNodes. In a cluster there may be hundreds of DataNodes and thousands of HDFS clients per cluster, as number of application tasks can be executed by each DataNode simultaneously.

Advantages of HDFS

1. High scalability.
2. Low limitation.
3. Open source.
4. Low cost.

Disadvantages of HDFS

1. Still rough - means software under active development.
2. Programming model is very restrictive.
3. Cluster management is high.

□□□