**Assignment 9.3**

Ques -1) Nosql Databases

A NoSQL database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases Motivations for this approach include: simplicity of design, simpler ["horizontal" scaling](https://en.wikipedia.org/wiki/Horizontal_scaling#Horizontal_and_vertical_scaling) to [clusters](https://en.wikipedia.org/wiki/Cluster_computing) of machines (which is a problem for relational databases),[[2]](https://en.wikipedia.org/wiki/NoSQL#cite_note-leavitt-2) and finer control over availability. The data structures used by NoSQL databases (e.g. key-value, wide column, graph, or document) are different from those used by default in relational databases, making some operations faster in NoSQL. The particular suitability of a given NoSQL database depends on the problem it must solve. Sometimes the data structures used by NoSQL databases are also viewed as "more flexible" than relational database tables.

Ques-2) Types of Nosql Databases

**Types of NoSQL databases-**

There are 4 basic types of NoSQL databases:

1. **Key-Value Store** – It has a Big Hash Table of keys & values {Example- Riak, Amazon S3 (Dynamo)}
2. **Document-based** **Store- It**stores documents made up of tagged elements. {Example- CouchDB}
3. **Column-based Store-**Each storage block contains data from only one column, {Example- HBase, Cassandra}
4. **Graph-based**-A network database that uses edges and nodes to represent and store data. {Example- Neo4J}

**Example of key**-value store

|  |  |
| --- | --- |
| **Key** | **Value** |
| “India” | {“B-25, Sector-58, Noida, India – 201301” |
| “Romania” | {“IMPS Moara Business Center, Buftea No. 1, Cluj-Napoca, 400606″,City Business Center, Coriolan Brediceanu No. 10, Building B, Timisoara, 300011”} |
| “US” | {“3975 Fair Ridge Drive. Suite 200 South, Fairfax, VA 22033 |

Example of Document based store

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | {officeName:”3Pillar Noida”,  {Street: “B-25, City:”Noida”, State:”UP”, Pincode:”201301”}  }  {officeName:”3Pillar Timisoara”,  {Boulevard:”Coriolan Brediceanu No. 10”, Block:”B, Ist Floor”, City: “Timisoara”, Pincode: 300011”}  }  {officeName:”3Pillar Cluj”,  {Latitude:”40.748328”, Longitude:”-73.985560”}  } |

Exampleof column based store

2-dimensional table comprising of rows and columns is part of the relational database system.

|  |  |  |  |
| --- | --- | --- | --- |
| **City** | **Pincode** | **Strength** | **Project** |
| Noida | 201301 | 250 | 20 |
| Cluj | 400606 | 200 | 15 |
| Timisoara | 300011 | 150 | 10 |
| Fairfax | VA 22033 | 100 | 5 |

For above RDBMS table a BigTable map can be visualized as shown below.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | {  3PillarNoida: {  city: Noida  pincode: 201301  },  details: {  strength: 250  projects: 20  }  }  {  3PillarCluj: {  address: {  city: Cluj  pincode: 400606  },  details: {  strength: 200  projects: 15  }  },  {  3PillarTimisoara: {  address: {  city: Timisoara  pincode: 300011  },  details: {  strength: 150  projects: 10  }  }  {  3PillarFairfax : {  address: {  city: Fairfax  pincode: VA 22033  },  details: {  strength: 100  projects: 5  }  } |

Ques) CAP Theorem

No distributed system is safe from network failures, thus network partitioning generally has to be tolerated. In the presence of a partition, one is then left with two options: consistency or availability. When choosing consistency over availability, the system will return an error or a time-out if particular information cannot be guaranteed to be up to date due to network partitioning. When choosing availability over consistency, the system will always process the query and try to return the most recent available version of the information, even if it cannot guarantee it is up to date due to network partitioning.[4]

In the absence of network failure – that is, when the distributed system is running normally – both availability and consistency can be satisfied.

CAP is frequently misunderstood as if one had to choose to abandon one of the three guarantees at all times. In fact, the choice is really between consistency and availability only when a network partition or failure happens ; at all other times, no trade-off has to be made.[5]

Database systems designed with traditional ACID guarantees in mind such as RDBMS choose consistency over availability, whereas systems designed around the BASE philosophy, common in the NoSQL movement for example, choose availability over consistency.[6]

Ques HBase Architecture

Physically, HBase is composed of three types of servers in a master slave type of architecture. Region servers serve data for reads and writes. When accessing data, clients communicate with HBase RegionServers directly. Region assignment, DDL (create, delete tables) operations are handled by the HBase Master process. Zookeeper, which is part of HDFS, maintains a live cluster state.

The Hadoop DataNode stores the data that the Region Server is managing. All HBase data is stored in HDFS files. Region Servers are collocated with the HDFS DataNodes, which enable data locality (putting the data close to where it is needed) for the data served by the RegionServers. HBase data is local when it is written, but when a region is moved, it is not local until compaction.

The NameNode maintains metadata information for all the physical data blocks that comprise the files.

Ques- HBase vs RDBMS

| RDBMS | HBase |
| --- | --- |
| This supports scale up. In other words, when more disk and memory processing power is needed, we need to upgrade it to a more powerful server. | This supports scale out. In other words, when more disk and memory processing power is needed, we need not upgrade the server. However, we need to add new servers to the cluster. |
| This uses SQL queries for reading records from tables. | This uses APIs and MapReduce for accessing data from HBase tables. |
| This is row oriented, that is, each row is a contiguous unit of page. | This is column oriented, that is, each column is a contiguous unit of page. |
| The amount of data depends on configuration of server. | The amount of data does not depend on the particular machine but the number of machines. |
| It's Schema is more restrictive. | Its schema is flexible |