Abstract: Big Data is a collection of datasets which are large and complex. Datasets can be structured or unstructured which gets input from various sources. KV-Store is a database in which data will be stored in key and value where each data have different key and value. With the help of KV(key-value) database we can support multiple CRUD operations at a time.

Key-words: Big Data, CRUD, CAP, BASE.

I. INTRODUCTION
Information technology gives utmost importance to processing data. Some terabytes of data is not sufficient for storing large amount of data. The amount of produced data is growing rapidly every year. The data which is stored in a database KV(Key-value) which have not any structure. In the scene the data will be stored in database in Key-value manner. Each data have a unique key and value.

A. Overview of Big Data:
The continuous increase in generation of large volume of data from blogs, tweets, facebook, Youtube, social networking sites, business processes opens a wide range of opportunities for organization to store, process, and analyze the data. Data is used by demands which help in formulating better making strategies and gain competitive advantage in the market.

B. Definition of Big Data:
Big Data refers to the large dataset that are complex and diverse in nature which to store, process, and analyze by using traditional relational database technologies.

C. Key Characteristics of Big Data:
Volume: volume describe to refer the scale of data which is generated from various sources which may be not only
Variety: variety describe various formats such as structured or unstructured data.
Velocity: velocity describe to the speed of data production which refers to rate of change of data generation from gigabyte per day to terabyte per day.
Veracity: Veracity refers to the data generation sources, data that is generally should be noise free.
Value: Value is most important characteristics for companies order to discover huge hidden value from large dataset.

D. Sources of Bigdata:
Data is not only limited to text format and may be form of videos, music, and large high quality images on social media more sources that produces data added on continuous basis for companies that existed earlier all of the data was generated exactly instead it was done by employee.

E. Uses of Bigdata:
In banking system, Education sector, Government sector, health sector, manufacturing and retail sector, etc

II. OVERVIEW OF KV-STORE
A. Introduction:
KV-value Database is also called KV-Store Database. It is one type of NoSQL database. In KV-store there is no Schema required for store data. values are identified and access via key and store value which can be number, string, image or videos etc. It is most flexible model of NoSQL because the application has complete control over what is stored in value.

B. Example of KV-Store database:
KV pair is a well establish concept in programming language. which is typically refers to a key-value as associative array structure.
Below example is of a student information database. It is simple but to provide an idea of how can Key-value works. In the given figure there are two columns representing key and a value. Here key is unique and representing their values or attributes corresponding to it and data is represented in the form of ring and the partitioning of data is done on the basis of their alphabets (in sorted order) and data is also replicated in the form of ring.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
</table>
| 1   | Name:Meera  
     | Semester:4  
     | Department:CE  
     | Spi:7.9 |
| 2   | Name:Keshav  
     | Semester:4  
     | Department:CE  
     | Spi:8.0 |

Figure-1 Keyvalue(KV)database

C. How types of data can be stored in a key-value database:
The key-value part refers to the fact that the database stores data as a collection of key/value pairs. This is a simple method of storing data, and it is known to scale well.

- **Key:** The key in a key-value pair must (or at least, should) be unique. This is the unique identifier that allows you to access the value associated with that key.
- **What is key-component:** A key component is a Java String. Issues of comparison can be answered by examining how Java Strings are compared using your preferred encoding. Because it is a String, a key component can be anything you want it to be.
- **Key major:** email address.
- **Key minor:** various properties, such as the user's street address, phone number, photograph, and name.
- **Value:** The combination of major and minor key components. So, for example, the value for an email address plus a street address might be multiple fields related to street number, street name, city, and so forth. If we can write a records components as. If file system path delimited by a ("<>") For example if you used multiple major components to identify a record, and one such record using the following major components: "Lenovo", and "HP." Another record might use "Lenovo" and "Dell". And a third might use "Acer", and "Toshiba". Then the major components for those records could be written as:

```plaintext
<Lenovo, HP>  
<Lenovo, Dell>  
<Acer, Toshiba>
```

D. Key Value(KV)database uses:
- It handles vast amount of data.
- Create a collection and optionally define a list of field data types using configuration files.
- perform create-read-update-delete(CRUD)operation using search lookup commands.
- Manage collection using the REST API.
- Replication of data is done using database in the form of ring. The data which is replicated it is stored in a ring. It is also useful to manage alphabetic order. figure2 describes ring partition and replication of data.

Figure-2 Ring Partition and replication of data
III. BIG DATA WITH KV-STORE

A. ACID Transaction:
ACID(Atomic, consistent, Isolated, Durable) which are database transaction.

- Atomic: Everything in a transaction succeeds or the entire transaction is rolled back.
- Consistent: A transaction cannot leave the database in an inconsistent state.
- Isolated: Transactions cannot interfere with each other.
- Durable: Completed transactions persist, even when servers restart

These qualities seem indispensable, and yet they are incompatible with availability and performance in very large systems.

B. CAP Theorem:
CAP theorem applied for distributed system. The theorem states that networked shared-data systems can only guarantee/strongly support two of the following three properties

- **Consistency**: A guarantee that every node in a distributed cluster returns the same, most recent, successful write. Consistency refers to every client having the same view of the data. There are various types of consistency models. Consistency in CAP (used to prove the theorem) refers to linearizability or sequential consistency, a very strong form of consistency.
- **Availability**: Every non-failing node returns a response for all read and write requests in a reasonable amount of time. The key word here is every. To be available, every node on (either side of a network partition) must be able to respond in a reasonable amount of time.
- **Partition Tolerant**: The system continues to function and upholds its consistency guarantees in spite of network partitions. Network partitions are a fact of life. Distributed systems guaranteeing partition tolerance can gracefully recover from partitions once the partition.

C. BASE Theorem:
BASE (Basic Availability, Soft-state, Eventual consistency) it is enough for the database to eventually be in a consistent state. Accounting systems do this all the time. It’s called closing out the books. It’s OK to use stale data, and it’s OK to give approximate answers. It’s harder to develop software in the fault-tolerant BASE world compared to the fastidious ACID world, but Brewer’s CAP theorem says you have no choice if you want to scale up. However, as Brewer points out in this presentation, there is a continuum between ACID and BASE.

IV. CONCLUSION AND FUTURE WORK
The main aim of this paper is to give overview of Bigdata and KV(Key-Value) database. It describes about Bigdata concept and its uses. Also describe about KV(Key-value)database, how data will be stored in database, key components, key value. It describes about data will be stored in KV(key-value) with the help of CAPE, BASE Theorem. The future work for this to handle kV databases on an oracle cloud using hadoop technology.

REFERENCES