TYBSC IT

Next Generation Technology (NGT)

SEM 5 (NOV-2019) Q.P.Code:53707

1.Attempt any three of the following:

a. What is Big Data? List the different uses of Big Data. (5)

Ans: Big data is a term used to describe data that has massive volume, comes in a variety of structures, and is generated at high velocity. This kind of data poses challenges to the traditional RDBMS systems used for storing and processing data. Big data is paving way for newer approaches of processing and storing data.

The different usage of big data are:

1. **Visibility**
   a. Accessibility to data in a timely fashion to relevant stakeholders generates a tremendous amount of value.
   b. Consider a manufacturing company that has R&D, engineering, and manufacturing departments dispersed geographically.
   c. If the data is accessible across all these departments and can be readily integrated, it can not only reduce the search and processing time but will also help in improving the product quality according to the present needs.
2. **Discover and Analyze Information**
   a. Most of the value of big data comes from when the data collected from outside sources can be merged with the organization's internal data.
   b. Organizations are capturing detailed data on inventories, employees, and customers.
   c. Using all of this data, they can discover and analyze new information and patterns; as a result, this information and knowledge can be used to improve processes and performance.
3. **Segmentation and Customizations**
   a. Big data enables organizations to create tailor-made products and services to meet specific segment needs.
   b. This can also be used in the social sector to accurately segment populations and target benefit schemes for specific needs..
4. **Aiding Decision Making**
   a. Big data can substantially minimize risks, improve decision making , and uncover valuable insights.
   b. Automated fraud alert systems in credit card processing and automatic fine-tuning of inventory are examples of systems that aid or automate decision-making based on big data analytics.
5. **Innovation**
   a. Big data enables innovation of new ideas in the form of products and services.
   b. It enables innovation in the existing ones in order to reach out to large segments of people.
   c. Using data gathered for actual products, the manufacturers can not only innovate to create the next generation product but they can also innovate sales offerings.

b. Briefly explain how is MongoDB different from SQL. **(5)**

Ans The following are the ways in which MongoDB is different from SQL.

1. MongoDB uses documents for storing its data, which offer a flexible schema (documents in same collection can have different fields). This enables the users to store nested or multi-value fields such as arrays, hashes, etc. In contrast, RDBMS systems offer a fixed schema where a column's value should have a similar data type. Also, it's not possible to store arrays or nested values in a cell.

2. MongoDB doesn't provide support for JOIN operations, like in SQL. However, it enables the user to store all relevant data together in a single document, avoiding at the periphery the usage of JOINs. It has a workaround to overcome this issue.

3. MongoDB doesn't provide support for transactions in the same way as SQL. However, it guarantees atomicity at the document level. Also, it uses an isolation operator to isolate write operations that affect multiple documents, but it does not provide "all-or-nothing" atomicity for multi-document write operations.

---

c. What is MongoDB? Explain the features of MongoDB. **(5)**

Ans. MongoDB is one of the leading NoSQL document store databases. It enables organizations to handle and gain meaningful insights from Big Data.

The features of MongoDB are as follows:

Schema-less Database:

- It is the great feature provided by the MongoDB. A Schema-less database means one collection can hold different types of documents in it.
- It is not necessary that the one document is similar to another document like in the relational databases. Due to this feature, MongoDB provides great flexibility to databases.

Document Oriented:

- In MongoDB, all the data stored in the documents.
- the data is stored in fields(key-value pair) instead of rows and columns which make the data much more flexible in comparison to RDBMS. And each document contains its unique object id.

Indexing:

- Every field in the documents is indexed with primary and secondary indices this makes easier and takes less time to get or search data from the pool of the data.
- If the data is not indexed, then database search each document with the specified query which takes lots of time and not so efficient.

Replication:

- MongoDB provides high availability and redundancy with the help of replication, it creates multiple copies of the data and sends these copies to a different server so that if one server fails, then the data is retrieved from another server.

Aggregation:

- It allows to perform operations on the grouped data and get a single result or computed result. It is similar to the SQL GROUPBY clause.

- It provides three different aggregations i.e, aggregation pipeline, map-reduce function, and single-purpose aggregation methods.

High Performance:

- The performance of MongoDB is very high and data persistence as compared to another database due to its features like scalability, indexing, replication, etc.

---

d. Explain how Volume, Velocity and Variety are important components of big data. **(5)**
Ans.

Volume:

a. Volume in big data means the size of the data.
b. as businesses are becoming more transaction-oriented, we see ever increasing numbers of transactions; more devices are getting connected to the Internet, which is adding to the volume
c. This huge volume of data is the biggest challenge for big data technologies.
d. The storage and processing power needed to store, process, and make accessible the data in a timely and cost effective manner is massive.

Variety:

a. The data generated from various devices and sources follows no fixed format or structure.
b. Variety of big data are text files, log files, streaming videos, photos, meter readings, stock ticker data, PDFs, audio, and various other unstructured formats.
c. There is no control over the structure of the data these days.
d. New sources and structures of data are being created at a rapid pace.
e. Example: to provide alternate routes for commuters, a traffic analysis application needs data feeds from millions of smartphones and sensors to provide accurate analytics on traffic conditions and alternate routes.

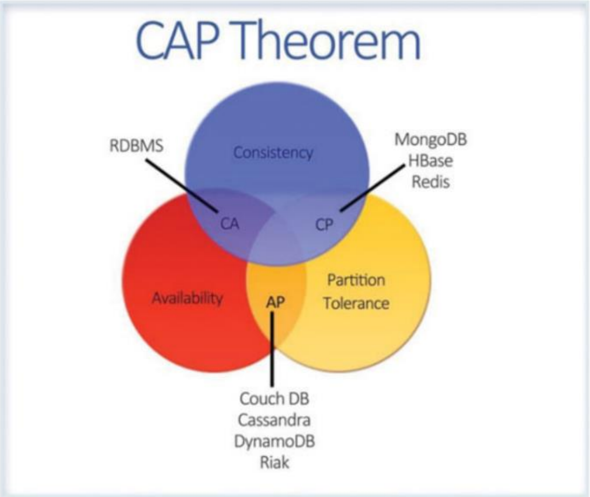
Velocity:

a. Velocity in big data is the speed at which data is created and the speed at which it is required to be processed.
b. If data cannot be processed at the required speed, it loses its significance.
c. Due to data streaming in from social media sites, sensors, tickers, metering, and monitoring, it is important for the organizations to speedily process data both when it is on move and when it is static.
d. Reacting and processing quickly enough to deal with the velocity of data is one more challenge for big data technology.

---

e. Write a short note on Cap theorem. **(5)**

Ans.



- CAP theorem also called as brewers theorem.
- Eric brewer outline the cap theorem in 2000
- The theorem states that when designing an application in a distributed environment namely consistent, availability and partition tolerance.

a. **Consistency** means that the data remains consistent after any operation is performed that changes the data, and that all users or clients accessing the application see the same updated data.

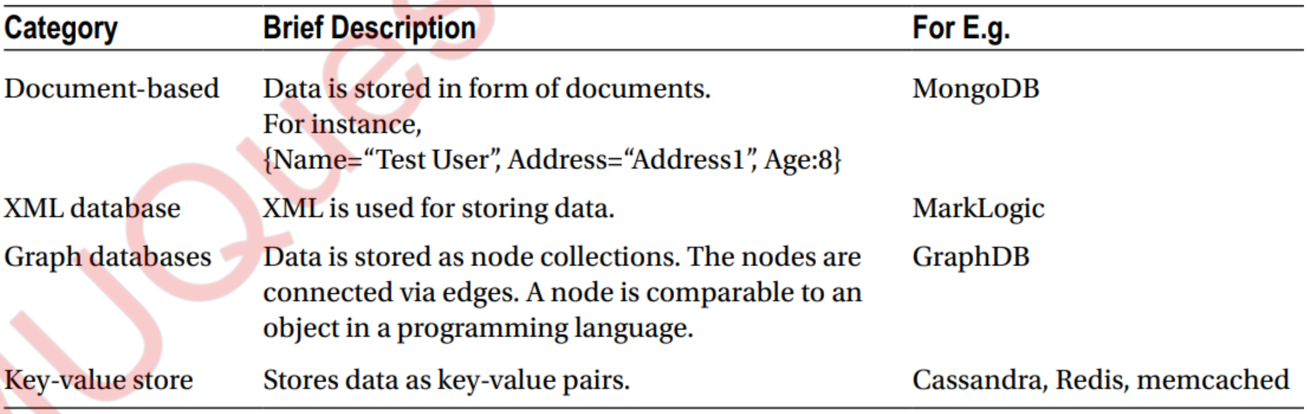
b. **Availability** means that the system is always available.

c. **Partition Tolerance** means that the system will continue to function even if it is partitioned into groups of servers that are not able to communicate with one another.

---

f. Discuss the various categories of NoSQL Databases. **(5)**

Ans.



Category	Brief Description	For E.g.
Document-based	Data is stored in form of documents. For instance, {Name="Test User", Address="Address1", Age:8}	MongoDB
XML database	XML is used for storing data.	MarkLogic
Graph databases	Data is stored as node collections. The nodes are connected via edges. A node is comparable to an object in a programming language.	GraphDB
Key-value store	Stores data as key-value pairs.	Cassandra, Redis, memcached

---

2.Attempt any three of the following:

a. Explain Binary JSON(BSON). **(5)**

Ans. BSON:

MongoDB stores the JSON document in a binary-encoded format termed as BSON. This is termed as BSON.

The BSON data model is an extended form of the JSON data model.

MongoDB's implementation of a BSON document is fast, highly traversable, and lightweight.

It supports embedding of arrays and objects within other arrays, and also enables MongoDB to reach inside the objects to build indexes and match objects.

The Identifier(_id):

MongoDB stores data in documents. Documents are made up of key-value pairs.

Although a document can be compared to a row in RDBMS, unlike a row, documents have flexible schema.

A key is used for querying data from documents.

A key that uniquely identifies each document within a collection. This is referred to as _id in MongoDB.

If you have not explicitly specified any value for a key, a unique value will be automatically generated and assigned to it by MongoDB.

This key value is immutable and can be of any data type except arrays.

---

b. Explain with an example the process of deleting documents in a Collection. **(5)**

Ans. To delete documents in a collection, use the remove () method .

- If you specify a selection criterion, only the documents meeting the criteria will be deleted. If no criteria is specified, all of the documents will be deleted.
- The following command will delete the documents where Gender = 'M' :

  ```
  > db.users.remove({"Gender":"M"})
  ```

- The same can be verified by issuing the find() command on Users:

  ```
  > db.users.find({"Gender":"M"})
  ```

- No documents are returned. The following command will delete all documents:

  ```
  > db.users.remove({})
  > db.users.find()
  ```

- Finally, if you want to drop the collection, the following command will drop the collection:

  ```
  > db.users.drop()
    True
  ```

- Finally, if you want to drop the collection, the following command will drop the collection:

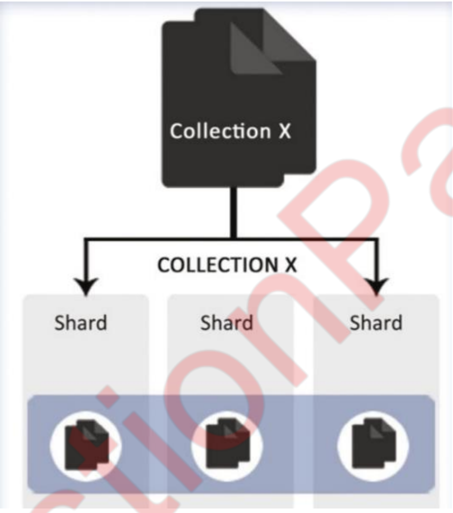
  ```
  > db.users.drop()
    True
  ```

As you can see, the collection name is not displayed, confirming that the collection has been removed from the database.

c. Explain the concept of sharding in detail. **(5)**

Ans.

- In MongoDB, the scaling is handled by scaling out the data horizontally (i.e. partitioning the data across multiple commodity servers), which is also called sharding (horizontal scaling).
- Sharding addresses the challenges of scaling to support large data sets and high throughput by horizontally dividing the datasets across servers.
- where each server is responsible for handling its part of data and no one server is burdened. These servers are also called shards.
- Sharding reduces the operations count handled by each shard.
- example, when data is inserted, only the shards responsible for storing those records need to be accessed.



Use sharding in the following instances:

- The size of the dataset is huge and it has started challenging the capacity of a single system.
- Since memory is used by MongoDB for quickly fetching data, it becomes important to scale out when the active work set limits are set to reach.
- If the application is write-intensive, sharding can be used to spread the writes across multiple servers.

The following are the components of a sharded cluster:

- Shards : The shard is the component where the actual data is stored.
- mongos : The mongos act as the routers.
- Config servers: Config servers are special

d. Discuss the various tools in MongoDB. **(5)**

Ans. There are various tools that are available as part of the MongoDB installation:

1. mongodump :

This utility is used as part of an effective backup strategy. It creates a binary export of the database contents.

2. mongorestore :
   The binary database dump created by the mongodump utility is imported to a new or an existing database using the mongorestore utility.
3. bsondump :
   This utility converts the BSON files into human-readable formats such as JSON and CSV. For example, this utility can be used to read the output file generated by mongodump.
4. mongoimport , mongoexport :
   mongoimport provides a method for taking data in JSON, CSV, or TSV formats and importing it into a mongod instance. mongoexport provides a method to export data from a mongod instance into JSON, CSV, or TSV formats.
5. mongostat , mongotop , mongosniff:
   These utilities provide diagnostic information related to the current operation of a mongod instance.

---

e. Differentiate between Single Key and Compound Index. **(5)**

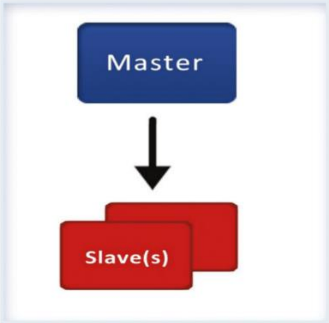
Ans.

Single Field Indexes	Compound Indexes
MongoDB supports user-defined indexes like single field index.	MongoDB supports a user-defined index on multiple fields as well called compound index.
A single field index is used to create an index on the single field of a document.	There sequential order of fields for a compound index.
With single field index, MongoDB can traverse in ascending and descending order. That’s why the index key does not matter in this case.	If a compound index consists of {“name”:1,”city”:1}), then the index will sort first the name and then the city.
MongoDB provides complete support for indexes on any field in a collection of documents.	MongoDB supports compound indexes, where a single index structure holds references to multiple fields within a collection’s documents.

---

f. Write a short note on Mater/Slave replication of MongoDB. **(5)**

Ans.



- In MongoDB, the traditional master/slave replication is available but it is recommended only for more than 50 node replications.
- In this type of replication, there is one master and a number of slaves that replicate the data from the master.
- The only advantage with this type of replication is that there's no restriction on the number of slaves within a cluster.
- However, thousands of slaves will overburden the master node, so in practical scenarios it's better to have less than dozen slaves.
- This type of replication doesn't automate failover and provides less redundancy.
- The master node maintains a capped collection (oplog) that stores an ordered history of logical writes to the database.
- The slaves replicate the data using this oplog collection.

There are two main reasons behind a slave becoming out of sync:

a. The slave shuts down or stops and restarts later. During this time, the oplog may have deleted the log of operations required to be applied on the slave.

b. The slave is slow in executing the updates that are available from the master.

---

3. Attempt any three of the following:

a. Discuss the field used for sharding. **(5)**

Ans. The field used for sharding are:

Time field :

– The data will be distributed evenly among the shards, neither the inserts nor the reads will be balanced.

– the time field is in an upward direction, so all the inserts will end up going to a single shard and the write throughput will end up being same as in a standalone instance.

Hashes :

– A hash of the _id field can be considered the shard key.

– Although this will cater to the write situation of the above (that is, the writes will be distributed), it will affect querying. In this case, the queries must be broadcasted to all the shards and will not be routable.

Use the key, which is evenly distributed, such as Host .

– The biggest potential drawback is that all data collected for a single host must go to the same chunk since all the documents in it have the same shard key.

– This will not be a problem if the data is getting collected across all the hosts, but if the monitoring collects a disproportionate amount of data for one host, you can end up with a large chunk that will be completely unsplittable, causing an unbalanced load on one shard.

Combining the best of options 2 and 3, you can have a compound shard key, such as {host:1, ssk: 1} where host is the host field of the document and ssk is _id field's hash value.

– In this case, the data is distributed largely by the host field making queries, accessing the host field local to either one shard or group of shards.

– At the same time, using ssk ensures that data is distributed evenly across the cluster. In most of the cases, such keys not only ensure ideal distribution of writes across the cluster but also ensure that queries access only the number of shards that are specific to them.

b. List and explain the limitation of Indexes. **(5)**

Ans. The limitation of indexes are as follows**:**

- **Index size :**
  Indexed items cannot be greater than 1024 bytes.
- **Number of indexes per collection :**
  At the most 64 indexes are allowed per collection.
- **Index name length :**
  By default the index name is made up of the field names and the index directions. The index name including the namespace (which is the database and the collection name) cannot be greater than 128 bytes. If the default index name is becoming too long, you can explicitly specify an index name to the ensure Index() helper.
- **Unique indexes in sharded collections** :
  Only when the full shard key is contained as a prefix of the unique index is it supported across shards; otherwise, the unique index is not supported across shards. In this case, the uniqueness is enforced across the full key and not a single field.
- **Number of indexed fields in a compound index :**
  This can't be more than 31 fields.

c. Explain the MongoDB limitations from security perspective. **(5)**

Ans. The MongoDB limitations from security perspective are as follows:

- **No Authentication by Default :**
  Although authentication is not enabled by default, it's fully supported and can be enabled easily.

- **Traffic to and from MongoDB Isn't Encrypted :**
  By default the connections to and from MongoDB are not encrypted. When running on a public network, consider encrypting the communication; otherwise it can pose a threat to your data. Communications on a public network can be encrypted using the SSL-supported build of MongoDB, which is available in the 64-bit version only.

d. Write a short note on deployment. **(5)**

Ans. While deciding on the deployment strategy , keep the following tips in mind so that the hardware sizing is done appropriately. These tips will also help you decide whether to use sharding and replication .

1. **Data set size :**
   a. The most important thing is to determine the current and anticipated data set size.
   b. This not only lets you choose resources for individual physical nodes, but it also helps when planning your sharding plans.

2. **Data importance :**
   a. To determine how important the data is and how tolerant you can be to any data loss or data lagging.

3. **Memory sizing :**
    a. MongoDB works best when the entire data set can reside in memory, thereby avoiding any kind of disk I/O.
    b. If possible, you should always select a platform that has memory greater than your working set size.
    c. If the size exceeds the single node's memory, you should consider using sharding so that the amount of available memory can be increased. This maximizes the overall deployment's performance.

4. **Disk Type :**
    a. It's very important to select a proper disk type. IOPS (input/output operations per second) is the key for selecting a disk type the higher the IOPS, the better the MongoDB performance.
    b. If possible, local disks should be used because network storage can cause poor performance and high latency.

5. **CPU** :
    a. Clock speed can also have a major impact on the overall performance when you are running a mongod with the majority of data in memory.
    b. In circumstances where you want to maximize the operations per second, you must consider including a CPU with a high clock/bus speed in your deployment strategy.

6. **Replication** :
    a. Replication is used if high availability is one of the requirements. In any MongoDB deployment it should be standard to set up a replica set with at least three nodes.
    b. Where there are two nodes in one data center and a backup node in a secondary data center.

---

e. Define Monitoring. Explain the factors to be considered while using Monitoring Services. (5)

Ans.

1. MongoDB system should be proactively monitored to detect unusual behaviours so that necessary actions can be taken to resolve issues.
2. Several tools are available for monitoring the MongoDB deployment.
3. A free hosted monitoring service named MongoDB Cloud Manager is provided by MongoDB developers.

The factors to be considered while using Monitoring Services are:

• **Op counters :**
  – Includes inserts, delete, reads, updates and cursor usage.

• **Resident memory :**
  – An eye should always be kept on the allocated memory.
  – This counter value should always be lower than the physical memory.
  – If you run out of memory, you will experience slowness in the performance due to page faults and index misses.

• **Working set size :**

- The active working set should fit into memory for a good performance, so a close eye needs to be kept on the working set.
- You can either optimize the queries so that the working set fits inside the memory or increase the memory when the working set is expected to increase.

• **Queues :**

- Queue metrics need to be monitored along with the lock percentage.
- If the queues and the lock percentage are trending upwards, that implies that you have contention within the database.

• Whenever there's a hiccup in the application, the CRUD behaviour, indexing patterns, and indexes can help you better understand the application's flow.

• It's recommended to run the entire performance test against a full-size database, such as the production database copy, because performance characteristic are often highlighted when dealing with the actual data.

---

f. What is Data Storage Engine. Differentiate between MMAP and Wired storage engine. (5)

Ans.

1. The data storage engine is the component of the database that is responsible for managing how data is stored, both in memory and on disk.
2. MongoDB supports multiple storage engines, as different engines perform better for specific workloads.
3. MongoDB uses MMAP as its default storage engine.
4. This engine works with memory-mapped files .
5. Memory-mapped files are data files.

Difference between MMAP and Wired storage engine:

MMAP:

i. Memory Mapping(MMAP) has been the only storage engine.
ii. MMAP uses Collection-level locking.
iii. if a client acquires a lock on a document to modify its content,then no other client can access the collection.
iv. MMAP uses journaling to recover from failure.
v. Data compression facilities is not present in MMAP storage engine.
vi. MMAP, increasing the size of the RAM decreases the number of page faults which in turn increases the performance.

WiredTiger:

i. But in version 3.0 and more, another storage engine, WiredTiger has been introduced which has its own benefits.
ii. WiredTiger uses Document-Level Locking.
iii. Multiple clients can access the same collection, since the lock is acquired for that particular document.
iv. In WiredTiger,a consistent view of the data is given by means of checkpoints. so that in case of failure it can rollback to the previous checkpoint.
v. In WiredTiger Storage Engine. Data compression is achieved using two methods

a. 1.Snappy compression
b. 2.Zlib

vi. Wired Tiger can make use of multithreading and hence multi core CPU's can be made use of it.

---

4. Attempt any three of the following:

a. Define In-Memory Database. What are the techniques used in In-Memory Database to ensure that data is not lost? (5)

Ans.

- The solid state disk may have had a transformative impact on database performance, but it has resulted in only incremental changes for most database architectures.
- A more paradigm-shifting trend has been the increasing practicality of storing complete databases in main memory.
- The cost of memory and the amount of memory that can be stored on a server have both been moving exponentially since the earliest days of computing.
- Many other databases of more modest size can now comfortably be stored within the memory of a single server.
- There are some database operations that simply must write to a persistent medium.
- Advantage of large memory system is that allows for the advantages of high-speed access without losing data in the event of a power failure.
- A large memory system requires an architecture that is aware the database is completely memory resident.

➢ In-memory databases generally use some combination of techniques to ensure they don't lose data. These include:

• Replicating data to other members of a cluster.
• Writing complete database images (called snapshots or checkpoints) to disk files.
• Writing out transaction/operation records to an append-only disk file (called a transaction log or journal).

---

b. Explain how does Redis uses disk files for persistence. (5)

Ans.

– Redis is an in-memory key-value store.
– Redis stands for Remote Dictionary Server.
– It was created by Salvatore Sanfilippo in 2009.
– Redis follows a familiar key-value store architecture in which keys point to objects.
– In Redis, objects consist mainly of strings and various types of collections of strings (lists, sorted lists, hash maps, etc.).

Redis uses disk files for persistence:

• **The Snapshot** files store copies of the entire Redis system at a point in time. Snapshots can be created on demand or can be configured to occur at scheduled intervals or after a threshold of writes has been reached. A snapshot also occurs when the server is shut down.

• **The Append Only File (AOF)** keeps a journal of changes that can be used to "roll forward" the database from a snapshot in the event of a failure. Configuration options allow the user to configure writes to the AOF after every operation, at one-second intervals, or based on operating-system-determined flush intervals.

- In addition, Redis supports asynchronous master/slave replication.
- Redis is popular among developers as a simple, high-performance key-value store that performs well without expensive hardware.

---

c. What is Berkeley Analytics Data Stack? Explain its components. (5)
Ans.

- In 2011, the AMPlab (Algorithms, Machines, and People) was established at University of California, Berkeley, to attack the emerging challenges of advanced analytics and machine learning on Big Data.
- The resulting Berkeley Data Analysis Stack (BDAS)—and in particular the Spark processing engine—has shown rapid uptake.

The components of BDAS are as follows:

1) **Spark**
   - Spark is an in-memory, distributed, fault-tolerant processing framework. Implemented in the Java virtual-machine-compatible programming language Scala, it provides higher-level abstractions than MapReduce and thus improves developer productivity.
   - As an in-memory solution, Spark excels at tasks that cause bottlenecks on disk IO in MapReduce. In particular, tasks that iterate repeatedly over a dataset—typical of many machine-learning workloads—show significant improvements.

2) **Mesos**
   - Mesos is a cluster management layer somewhat analogous to Hadoop's YARN.
   - However, Mesos is specifically intended to allow multiple frameworks, including BDAS and Hadoop, to share a cluster.

3) **Tachyon**
   - Tachyon is a fault-tolerant, Hadoop-compatible, memory-centric distributed file system.
   - The file system allows for disk storage of large datasets, but promotes aggressive caching to provide memory-level response times for frequently accessed data.

---

d. What is an Event? State the different types of Events in jQuery. (5)
Ans. All the different visitors' actions that a web page can respond to are called events.
An event represents the precise moment when something happens.

Examples:

- moving a mouse over an element
- selecting a radio button
- clicking on an element

The term **"fires/fired"** is often used with events. Example: "The keypress event is fired, the moment you press a key".

The different type of events are:

Commonly Used jQuery Event Methods:

Mouse Events	**Keyboard Events**	**Form Events**	**Document/ Window Events**
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

- **click()**:The function is executed when the user clicks on the HTML element.

– **Example :** $("p").click(function(){

```
                $(this).hide();
                 });
```

- **dblclick()** : The function is executed when the user double-clicks on the HTML element.

– **Example :** $("p").dblclick(function(){

```
                $(this).hide();
                  });
```

- **mouseenter()** : The function is executed when the mouse pointer enters the HTML element.

– **Example :** $("#p1").mouseenter(function(){

```
                alert("You entered p1!");
                 });
```

- **mouseleave()** : The function is executed when the mouse pointer leaves the HTML element.

– **Example :** $("#p1").mouseleave(function(){

```
                alert("Bye! You now leave p1!");
                 });
```

- **mousedown()** : The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element.

– **Example :** $("#p1").mousedown(function(){

```
                 alert("Mouse down over p1!");
                  });
```

e. Write a short note on jQuery CSS method. **(5)**

Ans. The css() method sets or returns one or more style properties for the selected elements.

When used to return properties:

This method returns the specified CSS property value of the FIRST matched element. However, shorthand CSS properties (like "background" and "border") are not fully supported and may give different results in different browsers.

When used to set properties:

This method sets the specified CSS property for ALL matched elements.

Syntax :

- Return the CSS property value:
  $(*selector*).css(*property*)

- Set the CSS property and value:
  $(*selector*).css(*property,value*)

- Set CSS property and value using a function:
  $(*selector*).css(*property,*function*(index,currentvalue)*)

- Set multiple properties and values:
  $(*selector*).css({*property:value*, *property:value*, ...})

Parameter	**Description**
property	Specifies the CSS property name, like "color", "font-weight", etc.
value	Specifies the value of the CSS property, like "red", "bold", etc.
function(*index,currentvalue*)	Specifies a function that returns the new value for the CSS property • *index* - Returns the index position of the element in the set • *currentvalue* - Returns the current value of the CSS property

---

f. State the features of jQuery. **(5)**

Ans. The features of jQuery are as follows:

i. **DOM manipulation :**
The jQuery made it easy to select DOM elements, negotiate them and modifying their content by using cross-browser open source selector engine called Sizzle.

ii. **Event handling :**
The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.

iii. **AJAX Support :**
The jQuery helps you a lot to develop a responsive and feature rich site using AJAX technology.

iv. **Animations :**
The jQuery comes with plenty of built-in animation effects which you can use in your websites.

v. **Lightweight :**
The jQuery is very lightweight library - about 19KB in size (Minified and gzipped).

vi. **Cross Browser Support :**
The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+

vii. **Latest Technology** :
The jQuery supports CSS3 selectors and basic XPath syntax.

---

5. Attempt any three of the following:

a. Explain the JSON Grammar. **(5)**

Ans. JSON syntax is basically considered as a subset of JavaScript syntax; it includes the following:

- Data is represented in name/value pairs.
- Curly braces hold objects and each name is followed by ':'(colon), the name/value pairs are separated by , (comma).
- Square brackets hold arrays and values are separated by ,(comma).

```
{
        "book":
          [
         {
           "id": "01",
           "language": "Java",
           "edition": "third",
           "author": "Herbert Schildt"
         },

         {
           "id": "07",
           "language": "C++",
           "edition": "second",
           "author": "E.Balagurusamy"
         }
          ]
}
```

JSON supports the following two data structures –

- **Collection of name/value pairs** – This Data Structure is supported by different programming languages.
- **Ordered list of values** – It includes array, list, vector or sequence etc.

– The implementations of the two structures are represented in the forms of the object and array.
– Crockford outlines the two structural representations of JSON through a series of syntax diagrams.

b. Differentiate between XML and JSON. **(5)**

Ans.

JSON	XML
It is JavaScript Object Notation	It is Extensible markup language
It is based on JavaScript language.	It is derived from SGML.
It is a way of representing objects.	It is a markup language and uses tag structure to represent data items.
It does not provides any support for namespaces.	It supports namespaces.
It supports array.	It doesn't supports array.
Its files are very easy to read as compared to XML.	Its documents are comparatively difficult to read and interpret.
It doesn't use end tag.	It has start and end tags.
It is less secured.	It is more secured than JSON.
It doesn't supports comments.	It supports comments.
It supports only UTF-8 encoding.	It supports various encoding.

c. Explain Request Headers. **(5)**

Ans.

- The second category of headers is that of the request headers.
- These headers can be supplied with the request to provide the server with preferential information that will assist in the request.

- They outline the configurations of the client making the request.
- Such headers may reveal information about the user-agent making the request or the preferred data type that the response should provide.
- the request headers are the most commonly configured headers.

- One very useful header is the Accept header.
- It can be used to inform the server as to what MIME type or data type the client can properly handle.
- This can often be set to a particular MIME type, such as application/json, or text/plain.
- It can even be set to */*, which informs the server that the client can accept all MIME types.
- The response provided by the server is expected to reflect one of the MIME types the client can handle.

- The following are request headers:
  - Accept
  - Accept-Charset
  - Accept-Encoding
  - Accept-Language
  - Authorization
  - Expect
  - From
  - Host
  - If-Match
  - If-Modified-Since
  - If-None-Match
  - If-Range
  - If-Unmodified-Since
  - Max-Forwards
  - Proxy-Authorization
  - Range
  - Referer
  - TE
  - User-Agent

---

d. Write a short note on JSON Parsing. **(5)**

Ans.

- Parsing is the process of analyzing a string of symbols, either in natural language or in computer languages, according to the rules of a formal grammar.
- As the grammar of JSON is a subset of JavaScript, the analysis of its tokens by the parser occurs indifferently from how the Engine parses source code. Because of this, the data produced from the analysis of the JSON grammar will be that of objects, arrays, strings, and numbers.
- Additionally, the three literals true, false, and null are produced as well.

JSON.parse

- In a nutshell, JSON.parse converts serialized JSON into usable JavaScript values.
  - Syntax of the JSON.parse() Method :

    JSON.parse(text [, reviver]);

- JSON.parse can accept two parameters, text and reviver.

- The name of the parameter text is indicative of the value it expects to receive.
- The parameter reviver is used similarly to the replacer parameter of stringify, in that it offers the ability for custom logic to be supplied for necessary parsing that would otherwise not be possible by default.

○ The parameter text implies the JavaScript value, which should be supplied.

○ This is a rather important aspect, because any invalid argument will automatically result in a parse error.

Eg : Invalid JSON Grammar Throws a Syntax Error :

```
var str = JSON.parse( "abc123" );  //SyntaxError: JSON.parse: unexpectedcharacter
```

- Throws an error because it was provided a string literal and not serialized JSON.

○ Therefore, "abc123" must be escaped and wrapped with an additional set of quotation marks.

- Valid JSON Grammer Is Successfully Parsed

```
var str = JSON.parse( "\"abc123\"" ); //valid JSON string value
console.log(str) //abc123;
console.log(typeof str) //string;
```

---

e. Explain the stringify object for JSON Object. **(5)**

Ans. 1. A common use of JSON is to exchange data to/from a web server. When sending data to a web server, the data has to be a string.

2. We can Convert a JavaScript object into a string with JSON.stringify(). Stringify a JavaScript Object. Imagine we have this object in JavaScript:

```
var obj = { name: "John", age: 30, city: "New York" };
```

Use the JavaScript function JSON.stringify() to convert it into a string.

```
var myJSON = JSON.stringify(obj);
```

The result will be a string following the JSON notation.

3. Syntax of the JSON stringify Method

```
JSON.stringify(value[, replacer [, space]]);
```

4. The value parameter of the stringify method is the only required parameter of the three outlined by the signature.The argument supplied to the method represents the JavaScript value intended to be serialized.This can be that of any object, primitive, or even a composite of the two.

5. The optional replacer parameter is either a function that alters the way objects and arrays are stringified or an array of strings and numbers that acts as a white list for selecting the object properties that will be stringified.

6. The third parameter, space, is also optional and allows you to specify the amount ofpadding that separates each value from one another within the produced JSON text. This padding provides an added layer of readability to the produced string.

7. Code:

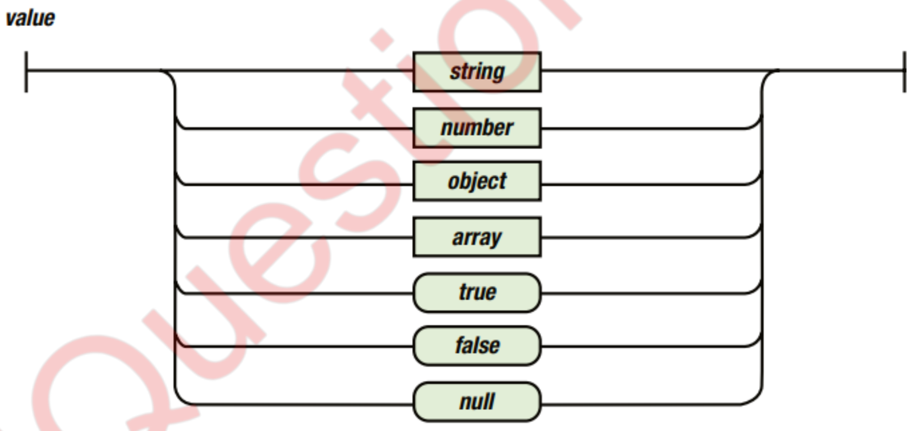
```
 <html>
<head>
<title>JSON programs </title>
```

```
</head>
<body>
<script type="text/javascript">
var data={
"Bname":"JSON",
"Publisher": "TataMcgraw",
"author": "Smith",
"price":250,
"ISBN":"1256897912345"
};
document.writeln(JSON.stringify(data));
document.writeln(JSON.stringify(data,["Bname","author","price"]));
document.write(JSON.stringify(data,["Bname","author","price"],5));
</script>
</body>
</html>
```

f. Discuss the JSON values. **(5)**

Ans.

- JSON is a subset of JavaScript and does not add anything that the JavaScript language does not possess.
- The values that can be utilized within our JSON structures are represented by types.
- JSON makes use of four primitive types and two structured types.
- A JSON value can only be a representative of string, number, object, array, true, false, and null.



Syntax diagram illustrating the possible values in JSON.

- JSON string must always begin and end with the use of double quotes.
- The numbers grammar does not begin or end with any particular symbolic representation.

- Examples of JSON Text Containing a Variety of Valid JSON Values:

  - // JSON text of an array with primitives.
    [null, true, 8]

  - // JSON text of an object with two members.

{ "first": "Ben", "last": "Smith", }

- // JSON text of an array with nested composites.
  [{ "abc": "123" }, ["0", 1, 2, 3, 4, 100]]

- //JSON text of an object with nested composites.
  { "object": { "array": [true] } }