Sonia Kargutkar – i590 Final Exam

Redis:

Redis stands for REmote DIctionary Server and is an open-source in-memory NoSQL database project implementing a distributed, in-memory key-value store with optional durability. The in-memory here refers to Redis mainly relying on the main memory to store data. This makes it more useful for applications where response time is critical like running telecommunications network equipment, mobile advertising networks etc.

Some of the main features of Redis are:

- An exceptionally fast pace of Read/Write operations
- Improved durability and Scalability due to use of Master Slave Replication to create multiple instances.
- Relatively cheaper creation of Redis instances in terms of memory and CPU requirements but depends totally upon the memory available.
- Direct data access since no intermediate data structures required before data storage like JSON in MongoDB or Database Schema in any SQL database.
- Redis also allows direct Algorithmic interaction with your data making it easier to manipulate data.

Data is stored in Redis in the form of Key-Value pairs. Various commands are used to manipulate these keys and the most commonly used ones are SET, GET, DEL, MOVE, RENAME, TYPE, PATTERN and EXISTS.

Redis has 5 types of data structures:

1. Strings: Most basic data structure in Redis
2. Hashes: Redis Hashes are maps between string fields and string value and are the perfect data structure to represent objects.
3. Lists: Redis Lists are simple lists of strings, sorted by insertion order.
4. Sets: Redis Sets are an unordered collection of Strings and have the desirable property of not allowing repeated members.
5. SortedSets: Redis Sorted Sets are non repeating collections of Strings similar to Redis Sets.

Redis implements the Publish/Subscribe messaging paradigm where senders (publishers) are not programmed to send their messages to specific receivers (subscribers). Rather, published messages are characterized into channels, without knowledge of what subscribers may be available. Subscribers express interest in one or more channels, and only receive messages that are of interest, without knowledge of which publishers exist. This can allow for greater scalability and a more dynamic network topology.

Redis transactions allow execution of a group of commands in a single step. All the commands in a transaction are serialized and executed sequentially. This guarantees that the commands are executed as a single isolated operation.

Cassandra

Apache Cassandra is open source, linearly scalable, distributed and fault-tolerant NoSQL database. Designed to handle a huge amount of data. Its distribution design is based on Amazon's Dynamo and its data model on Google's Bigtable. Cassandra implements a Dynamo-style replication model with no single point of failure, but adds a more powerful "column family" data model. Cassandra is being used by some of the biggest companies such as Facebook, Twitter, Cisco, Rackspace, ebay, Twitter, Netflix, and more.

Some of the main features of Cassandra are:

- Massively Scalable Architecture where all nodes are at the same level which provides operational simplicity and easy scale out.
- Masterless Architecture where data can be written and read on any node.
- Linear Scale Performance because as more nodes are added, the performance of Cassandra increases.
- No Single point of failure since Cassandra replicates data on different nodes that ensures no single point of failure.
- Fault Detection and Recovery since failed nodes can easily be restored and recovered.
- Flexible and Dynamic Data Model which supports datatypes with fast writes and reads.
- Data is protected with commit log design and build in security like backup and restore mechanisms.
- Tunable Data Consistency with support for strong data consistency across distributed architecture.
- Multi Data Center Replication is a feature to replicate data across multiple data center.
- Data Compression where Cassandra can compress up to 80% data without any overhead.
- Cassandra Query language that is similar like SQL language. It makes it very easy for relational database developers moving from relational database to Cassandra.

There are following components in Cassandra :

- Node: Node is the place where data is stored. It is the basic component of Cassandra.
- Data Center: A collection of nodes are called data center. Many nodes are categorized as a data center.
- Cluster: The cluster is the collection of many data centers.
- Commit Log: Every write operation is written to Commit Log. Commit log is used for crash recovery.
- Mem-table: After data written in Commit log, data is written in Mem-table. Data is written in Mem-table temporarily.
- SSTable: When Mem-table reaches a certain threshold, data is flushed to an SSTable disk file.
- Bloom Filter and Index: They contain information about location of data in SSTable. Each SSTable has one bloom filter and one index associated with it.

Sonia Kargutkar – i590 Final Exam

HBase

HBase is a distributed column-family oriented database built on top of the HDFS (Hadoop Distributed File System). It is horizontally scalable and is a data model that is similar to Google's Bigtable designed to provide quick random access to huge amounts of structured data. It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System. One can store the data in HDFS either directly or through HBase. Data consumer reads/accesses the data in HDFS randomly using HBase. HBase sits on top of the Hadoop File System and provides read and write access.

HBase is a column-oriented database and the tables in it are sorted by row. The table schema defines only column families, which are the key value pairs. A table can have multiple column families and each column family can have any number of columns. Subsequent column values are stored contiguously on the disk. Each cell value of the table has a timestamp.

Some of the main features of HBase are:

- HBase is linearly scalable.
- It has automatic failure support.
- It provides consistent read and writes.
- It integrates with Hadoop, both as a source and a destination.
- It has easy Java API for client.
- It provides data replication across clusters.

HBase has two type of nodes – Master and RegionServer. Tables are split into regions and are served by the region servers. Regions are vertically divided by column families into "Stores". Stores are saved as files in HDFS. Region servers can be added or removed as per requirement.

HBase contains a shell using which you can communicate with HBase.

Some of the General commands used in HBase are:

- status: This command returns the status of the system including the details of the servers running on the system.
- version**:** This command returns the version of HBase used in your system.
- table_help: This command guides you what and how to use table-referenced commnads.
- whoami**:** This command returns the user details of HBase.

HBase is written in java, therefore it provides java API to communicate with HBase. Java API is the fastest way to communicate with HBase.

Sonia Kargutkar – i590 Final Exam

MongoDB

MongoDB is an open-source and cross-platform document database that provides high performance, high availability, and automatic scaling. A record in MongoDB is a document, which is a data structure composed of field and value pairs. The values of fields may include other documents, arrays, and arrays of documents.

Documents (i.e. objects) correspond to native data types in many programming languages. Embedded documents and arrays reduce need for expensive joins. MongoDB stores BSON documents, i.e. data records, in collections. In MongoDB, databases hold collections of documents. MongoDB stores documents in collections. Collections are analogous to tables in relational databases.

Neo4J

Neo4j is a highly scalable, native graph database purpose-built to leverage not only data but also its relationships. Graphs – i.e., networks – are the most efficient and intuitive way of working with data, mimicking the interconnectedness of ideas in the human mind. Neo4j is built from the ground up to harness the power of graphs for real-time, bottom-line insights. It follows an intuitive adaptable data modeling technique. It uses Cypher, the graph query language. Cypher is a declarative query language: it describes what you are interested in, not how it is acquired. Cypher is meant to be very readable and expressive

Nodes are often used to represent entities, but depending on the domain relationships may be used for that purpose as well. The simplest possible graph is a single Node. A Node can have zero or more named values referred to as properties. A relationship connects two nodes, and is guaranteed to have valid start and end nodes. Just like nodes, relationships can have properties. Relationships allow a graph to resemble a list, a tree, a map, or a compound entity. Properties are named values where the name is a string. The supported property values are: Numeric values, String values, Boolean values, Lists of any other type of value

The use of XML and related technology

HTML has no user-defined tags, describes only data format, no content, a lack of compatibility with other popular browsers. This led to the need for XML.
XML stands for eXtensible Markup Language. It allows for user-defined tags and has its own grammar. It can be used to describe structured and unstructured data.

Some XML Syntax requirements:

- All XML elements must have a closing tag
- XML tags are case sensitive
- XML elements must be properly nested
- XML documents must have one and only one root element
  XML attribute values must be quoted (single quotation mark or double qutoation mark)