Gurjus Singh

MSDS 432 Foundations of Data Engineering

August 16th, 2020

Module 8– Reading Comprehension

1. **Describe the publish/subscribe model as it pertains to events, messaging systems, producers, consumers, and message brokers. (2 pt)**

   The publish/subscribe model is a common approach for notifying consumers about new events. The way it works is that there is a producer who generates a message that contains an event, and this is sent out to consumers who are subscribed to this message. The connection type that could be used for this model is called a Unix pipe or TCP. For this to work though, there needs to be several producer nodes and several consumer nodes as the Unix Pipe and TCP system can only send from one sender to one recipient.

   This publish/subscribe model can not only involve this type of system but different types of systems. For example, some systems involve a way to control how fast a producer can send a message. These messages can be buffered in a queue to send at a specific time or some systems can also apply back pressure to blocking more than a limited number if messages that are in a queue. Also, these systems should be able to be durable and be able to survive node crashes through replication in case messages are lost. Although these messages may not be important if not a huge amount is lost as it depends on the application and purpose of the system. Some systems also use direct network communication as well. This is a way for to pass messages which do not involve nodes in the middle of a source to destination path. Types of direct connections are UDP multicast, Zero MQ which uses TCP or IP, HTTP or RPC.

2. **What are the two main patterns of messaging when using multiple consumers? (2 pt)**

   The two main patterns of messaging when using multiple consumers are Load Balancing and Fan Out. Load Balancing is when each message is delivered to a single consumer. This consumer can share the work of processing the message from the producer. There are also instances in Load Balancing where the message may arbitrarily be passed to consumers. There is no set decision on where to send the message. The idea behind Load Balancing is when a message is expensive to process, who can use consumers to process the message which can be used to parallelize the process.

   Fan-out is another pattern which is when messages are delivered to all the consumers. This allows consumers to each tune-in to the same broadcast that is sending the message, so the consumers do not affect their peers. This is equivalent to batch jobs, that are received by the

same destination. These two patterns can be combined such as there are two separate group of nodes that each receive the message while in each group a single consumer node receives the message.

3. **Explain the concepts of change data capture and event sourcing, and how they differ. (2 pt)**

The concept of change data capture is a mechanism for ensuring that all changes made to the system of record are also reflected in the derived data systems, for the goal that the data is copied accurately on to the derived system. A derived data system takes some result from some other system and transforms and processes it in another way. The goal of change data structure is to have a way of observing data changes written to a database and extracting these changes so the copies can be made on other systems.

The way Change Data Capture works is it makes one database and makes it the leader. It then turn the other databases into followers. The way the changes can be transmitted is through a log-based message broker. Change data capture is usually asynchronous in that the leader does not wait for the change to be applied to the consumers before committing the change.

Event sourcing is a technique that was developed in the domain driven design community. It works by storing changes to an application in a log of change events. The way it differs from CDC is that the events are not mutable. The event is stored by append only and updating and deleting is discouraged even often prohibited. It only is used to reflect changes at the higher level for example in an application.

Event sourcing is powerful because user's events are immutable in that it will not affect the the database. It also is advantageous because it makes it easier for an application to develop, and allows better debugging while also guarding against bugs.

4. **What are Complex Event Processing and Stream Analytics, and how do they differ? (2 pt)**

   Complex event processing is an approach that was first made in the 1990's. It is used in analyzing event streams and developed for applications that need to search for certain kind of patterns in events. It is similar to regular expression, which is used to search for patterns in strings, but it allows you to filter by events. The language used for CEP is usually SQL, or a GUI interface to help input requests to type of events.

   Stream Analytics is another approach for analyzing streams. This differs from Complex Event Processing because it is less interested in finding specific event sequences and is more focused over aggregating the data and statistics over large number of events. For example, some work it can do is measure rates of how often events occur, calculating rolling averages of some value, and comparing statistics between time intervals. This approach also involves probability algorithms as well.

5. **What are some machine learning algorithms that use greedy algorithms? (2 pt)**

Greedy algorithms is an algorithm concept that always tries to choose the best possible outcome for a particular problem. The thing about greedy algorithms is you have to analyze if the solution is correct. There are several algorithms that are considered greedy and it is just a concept and is not a particular algorithm. Some algorithms in machine learning that are used are Dijkstra's Algorithm and Scheduling Jobs, so each server gets equal load.

Machine learning algorithms are algorithms that improve through experience. These algorithms can be improved if automated which most database firms do. Dijkstra's is used by Google and has improved through experience. When one drives and changes routes, the algorithm is automatically updated due to traffic conditions, velocity and direction. The way the algorithm works is edges has weights, and each route is analyzed between nodes to and compared to see which route has the least cost/weights.

In the Scheduling Jobs example, there are n servers and incoming requests are divided in such a way that they are equally spread through these servers. There is an array that tells you the number of jobs currently being processed by a server and an array that tells you how many jobs are incoming to a particular server. In order to reallocate jobs to servers you need to see sum up all the jobs and divide by the number of severs to figure equal number of jobs to put on each server. Even though this type of job does not improve through experience it could if one takes into account load on each server, if it has issues in the past such as crashes overtime and other statistics.