Programing For Big Data

**Q1: Why Apache Spark is replacing Z00 Keeper with KIP-500**

Apache uses ZOO Keeper for storing metadata externally. Concept of External Metadata management isn’t very efficient. We run at least three additional Java processes, and sometimes more. Worse still, storing metadata externally limits Kafka’s scalability. As the amount of metadata grows, so does the length of this loading process.

On the other hand, KIP-500 outlines a better way of handling metadata in Kafka. You can think of this as “Kafka on Kafka,” since it involves storing Kafka’s metadata in Kafka itself rather than in an external system such as ZooKeeper. In the post-KIP-500 world, metadata will be stored in a partition inside Kafka rather than in ZooKeeper. The controller will be the leader of this partition. There will be no external metadata system to configure and manage, just Kafka itself. which makes Kafka even more simple to operate and deploy

**Q2: What is lazy operation in Spark?**

“LAZY” the word itself indicates its meaning ‘not at the same time’.

That means, it evaluates something only when we require it. In accordance with a spark, it does not execute each operation right away, that means it does not start until we trigger any action. Once an action is called all the transformations will execute in one go.

This process is more appealing than executing one operation at the same time. Execution at the same time may lead to delay the process. Overcoming to the downtime, it gives birth to**Lazy Evaluation**.

Using lazy evaluation, It figured out that all transformations can be combined together into a single transformation and executed together. Simply, it says while performing any operation it will not execute until we trigger an action.

**Q3: Why Spark uses off-heap memory?**

Off-heap memory is used in Apache Spark for the storage and for the execution data.

Spark uses off-heap memory for two purposes:

* A part of off-heap memory is used by Java internally for purposes like String interning and JVM overheads.
* Off-Heap memory can also be used by Spark explicitly for storing its data.

**Q4: Why Transformations and actions are used in RDD?**

Tansformations create a new dataset from an existing one. It takes RDD as input and produces one or more RDD as output. Each time it creates new RDD when we apply any transformation. Thus, the so input RDDs, cannot be changed since RDD are immutable in nature.

Actions return a value to the driver program after running a computation on the dataset. An action is one of the ways of sending data from *Executer* to the *driver.*

**Q5: What is difference between narrow and wide transformations?**

|  |  |
| --- | --- |
| Narrow Transformations | Wide Transformations |
| These types of transformations convert each input partition to only one output partition | This type of transformation will have input partitions contributing to many output partitions. |
| This kind of transformations is basically fast. | Slow as compare to narrow dependencies. |
| Does not require any data shuffling over the cluster network or no data movement. | Might Require data shuffling over the cluster network or no data movement. |
| Operation of map()and filter() belongs to this transformations | Functions such as groupByKey(), join(), Repartition(), aggregate()  Are some examples. |

**Q6: Why Is Mesos Used?**

Mesos is a cluster manager that provides efficient resource isolation and sharing across distributed applications or frameworks. Mesos is a open source software originally developed at the University of California at Berkeley. It sits between the application layer and the operating system and makes it easier to deploy and manage applications in large-scale clustered environments more efficiently. It can run many applications on a dynamically shared pool of nodes. Prominent users of Mesos include

**Q7: What is horizontal and vertical scalability?**

Horizontal scaling means scaling by adding more machines to your pool of resources (also described as “scaling out”), whereas vertical scaling refers to scaling by adding more power (e.g. CPU, RAM) to an existing machine (also described as “scaling up”).

**Q8: Can publisher generates messages on multiple topics?**

Yes, A Publisher can generate messages on multiple topics.

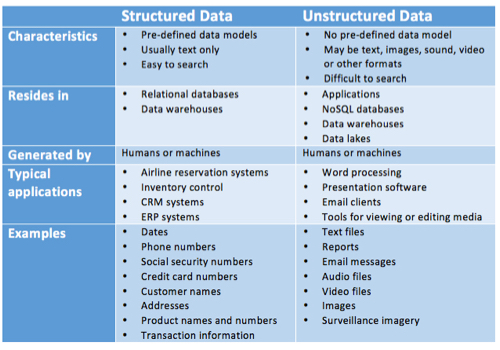
**Q9: What is leader Election algorithm?**

Election algorithms are designed to choose a coordinator.

Election algorithms choose a process from group of processors to act as a coordinator. If the coordinator process crashes due to some reasons, then a new coordinator is elected on other processor. Election algorithm basically determines where a new copy of coordinator should be restarted.

Election algorithm assumes that every active process in the system has a unique priority number. The process with highest priority will be chosen as a new coordinator. Hence, when a coordinator fails, this algorithm elects that active process which has highest priority number.Then this number is send to every active process in the distributed system.

**Q10: What is structured and unstructured data?**



**Q11: Is data gathering in apache flume event based or scheduling?**

It is event based.