What are the design patterns in Scala, all the types of design patterns?

<https://www.lihaoyi.com/post/OldDesignPatternsinScala.html>

Similarities in pattern matching and object oriented design patterns scala?

what can pattern matching be used for in Scala?

What is implicit design pattern in Scala?

<https://www.lihaoyi.com/post/ImplicitDesignPatternsinScala.html>

what does implicit class resemble in the world

What are design patterns in Scala?

<https://pavelfatin.com/design-patterns-in-scala/>

What are pure functions in Scala?

<https://www.geeksforgeeks.org/pure-function-in-scala/>

How do you use pure functions and take that to improve performance by running in parallelism?

* Parallel collections in Java

<https://docs.scala-lang.org/overviews/parallel-collections/overview.html>

* Talk about parallelism in Scala

What is lazy evaluation in Scala?

<https://www.geeksforgeeks.org/scala-lazy-evaluation/>

What is memoization in Scala?

What is a Currying function in Scala?

<https://www.geeksforgeeks.org/currying-functions-in-scala-with-examples/#:~:text=Currying%20in%20Scala%20is%20simply,widely%20in%20multiple%20functional%20languages>.

What are higher order functions?

How would you compose futures in Scala?

<https://danielasfregola.com/2015/04/01/how-to-compose-scala-futures/>

What are syntactic sugar in scala?

What are the different types of For comprehensions in Scala?

<https://www.baeldung.com/scala/for-comprehension>

what is responsibility or unapply method in scala?

<https://docs.scala-lang.org/tour/extractor-objects.html>

what are Scala separator?

<https://www.knowledgehut.com/interview-questions/scala>

<https://www.guru99.com/scala-interview-questions.html>

<https://hackr.io/blog/scala-interview-questions>

<https://www.edureka.co/blog/interview-questions/scala-interview-questions/>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

What is a managed table vs external table in Hive?

* Internal or manage table: When we create a table in hive, it by default manages the data, meaning that hive moves the data into it’s warehouse directory.
* By default tables created in hive are managed.
* Data Load: When we load data into managed table, then Hive moves data into Hive warehouse directory which is hdfs://user/hive/warehouse
* Drop Table: The metadata information along with the table data is deleted from the Hive warehouse if we drop the manage table.
* External Table: External tables are not created by default, we have to use External Keyword during table creation along with the data location.
* CREATE EXTERNAL TABLE <table\_name>(dummy STRING)
* LOCATION ‘/USER/<directory\_name>/<table\_name>’;
* LOAD DATA INPATH ‘user/<directory\_name>/data.txt INTO TABLE <table\_name>’;
* External tables refer to the data that is defined location outside the warehouse directory. The location of the external data is specified at the table creation time.
* Drop Table: Hive just deletes the metadata information regarding the table, but it leaves the table data present in HDFS untouched.

Security:

● Managed tables - hive solely controls the managed table security. Within hive, security needs to be managed; probably at the schema level (depends on organization policies).

● External Tables: these tables’ files are accessible to anyone who has access to HDFS file structure. So, it needs to manage security at the HDFS file/folder lever.

Difference between Avro and Parquet?

* AVRO is a row-based storage format whereas PARQUET is a columnar based storage format. PARQUET is much better for analytical querying i.e. reads and querying are much more efficient than writing. Write operations in AVRO are better than in PARQUET. AVRO is much matured than PARQUET when it comes to schema evolution.

What are Avro use cases?

* Avro supports direct mapping to JSON as well as a compact binary format. It is a very fast serialization format. Avro is widely used in the Hadoop ecosystem. Avro supports polyglot bindings to many programming languages and a code generation for static languages.

What is the use case for a sequence file?

* Sequence files allows you to solve this problem of small files. As discussed sequence file are the files containing key-value pairs. So, you can use it to hold multiple key-value pairs where the key can be unique file metadata, like filename+timestamp and value is the content of the ingested file.

What workflow scheduler did you use for Hadoop jobs?

* Apache Oozie is a workflow scheduler for Hadoop. It is a system which runs the workflow of dependent jobs. Here, users are permitted to create Directed Acyclic Graphs of workflows, which can be run in parallel and sequentially in Hadoop.

How did you schedule Spark jobs?

* Standalone mode: By default, applications submitted to the standalone mode cluster will run in FIFO (first-in-first-out) order, and each application will try to use all available nodes. You can limit the number of nodes an application uses by setting the spark.cores.max configuration property in it, or change the default for applications that don’t set this setting through spark.deploy.defaultCores. Finally, in addition to controlling cores, each application’s spark.executor.memory setting controls its memory use.
* Mesos: To use static partitioning on Mesos, set the spark.mesos.coarse configuration property to true, and optionally set spark.cores.max to limit each application’s resource share as in the standalone mode. You should also set spark.executor.memory to control the executor memory.
* YARN: The --num-executors option to the Spark YARN client controls how many executors it will allocate on the cluster (spark.executor.instances as configuration property), while --executor-memory (spark.executor.memory configuration property) and --executor-cores (spark.executor.cores configuration property) control the resources per executor. For more information, see the YARN Spark Properties.

Why will there be in file issues with executors in spark?

* Failure of worker node – The node which runs the application code on the Spark cluster is Spark worker node. ... Any of the worker nodes running executor can fail, thus resulting in loss of in-memory If any receivers were running on failed nodes, then their buffer data will be lost

What is effective final in Java?

* A non-final local variable or method parameter whose value is never changed after initialization is known as effectively final. A variable which is not declared as final but whose value is never changed after initialization is effectively final. ... Java 8 compiler can detect that the variable counter remains unchanged and we can use a non-final local variable inside a lambda expression.

What is a functional interface in Java?

* A functional interface is an interface that contains only one abstract method. They can have only one functionality to exhibit. From Java 8 onwards, lambda expressions can be used to represent the instance of a functional interface. A functional interface can have any number of default methods.

What frameworks did you use to develop api in java?

* Spring Boot is a Java framework that is built using the venerable Spring framework. You use the framework for creating stand-alone microservices. Spring Boot helps ease that development by minimizing configuration details. It consists of an embedded Tomcat or Jetty application server and other dependencies depending upon project needs. These pre-packaged bundles are called “starter projects” and bundle all needed dependencies into one package. The framework also offers an online project starter called the “Spring Initializr,” available at the start.spring.io website which allows you to configure a new project and then download your customized Spring Boot package.

Write pseudocode in Spark to count words in file

* https://dzone.com/articles/wordcount-with-spark-and-scala

What is a dataframe vs dataset?

DataFrames

* DataFrames gives a schema view of data basically, it is an abstraction. In dataframes, view of data is organized as columns with column name and types info. In addition, we can say data in dataframe is as same as the table in relational database. As similar as RDD, execution in dataframe too is lazy triggered. Moreover, to allow efficient processing datasets is structure as a distributed collection of data. Spark also uses catalyst optimizer along with dataframes.

DataSets

* In Spark, datasets are an extension of dataframes. Basically, it earns two different APIs characteristics, such as strongly typed and untyped. Datasets are by default a collection of strongly typed JVM objects, unlike dataframes. Moreover, it uses Spark’s Catalyst optimizer. For exposing expressions & data field to a query planner.

What is flat map vs map?

* A map is a transformation operation in Apache Spark. It applies to each element of RDD and it returns the result as new RDD. In the Map, operation developer can define his own custom business logic. The same logic will be applied to all the elements of RDD.
* A flatMap is a transformation operation. It applies to each element of RDD and it returns the result as new RDD. It is similar to Map, but FlatMap allows returning 0, 1 or more elements from map function. In the FlatMap operation, a developer can define his own custom business logic. The same logic will be applied to all the elements of the RDD.

How do you connect SQL with Kafka?

* Apache Kafka supports connecting with Microsoft SQL Server and numerous other databases/data warehouses with the help of various in-built connectors. These connectors help bring in data from a source of your choice to Apache Kafka and then stream it to the destination of your choice from Kafka Topics.
* https://hevodata.com/learn/connect-apache-kafka-to-sql-server/

How do you dump into events into spark?

What is your Spark environment?

How do you ingest data in apache nifi?

What packages did you install in Linux?

What is udf in hive?

* User Defined Functions, also known as UDF, allow you to create custom functions to process records or groups of records. Hive comes with a comprehensive library of functions. There are however some omissions, and some specific cases for which UDFs are the solution.

What is Big O notation of reading Hashmap?

* A particular feature of a HashMap is that unlike, say, balanced trees, its behavior is probabilistic. In these cases its usually most helpful to talk about complexity in terms of the probability of a worst-case event occurring would be. For a hash map, that of course is the case of a collision with respect to how full the map happens to be. A collision is pretty easy to estimate.

Use cases in apache nifi?

https://capgemini.github.io/development/introduction-nifi-best-practices/

### [What are workers, executors, cores in Spark](https://stackoverflow.com/questions/32621990/what-are-workers-executors-cores-in-spark-standalone-cluster" \l ":~:text=Executors%20are%20worker%20nodes'%20processes,the%20results%20to%20the%20driver.)

* Executors are worker nodes' processes in charge of running individual tasks in a given Spark job. They are launched at the beginning of a Spark application and typically run for the entire lifetime of an application. Once they have run the task they send the results to the driver. They also provide in-memory storage for RDDs that are cached by user programs through Block Manager.

How is the version specified in the CI/CD in docker?

* https://circleci.com/blog/build-cicd-piplines-using-docker/

How do you keep track of Docker images?

What automated batch processing tools did you use?

What is batch processing intervals?

BDD vs TDD?

Frameworks in testing spark?

CI/CD process in spark?

What is your release process in pyspark?

# [Spark cache vs broadcast](https://stackoverflow.com/questions/38056774/spark-cache-vs-broadcast)

# RDDs are divided into partitions. These partitions themselves act as an immutable subset of the entire RDD. When Spark executes each stage of the graph, each partition gets sent to a worker which operates on the subset of the data. In turn, each worker can cache the data if the RDD needs to be re-iterated.

# Broadcast variables are used to send some immutable state once to each worker. You use them when you want a local copy of a variable.

# These two operations are quite different from each other, and each one represents a solution to a different problem.

# shuffling in spark

# A shuffle occurs when data is rearranged between partitions. This is required when a transformation requires information from other partitions, such as summing all the values in a column. Spark will gather the required data from each partition and combine it into a new partition, likely on a different executor.

\*Kafka batch processing

* At its core, it allow systems that generate data (called Producers) to send their data in real-time in a kafka topic, which has partitions. ... Any topic can then be read by any number of systems who need that data in real-time (called Consumers).

\*spark batch vs spark streaming

* Although people use the word in different ways, Hadoop refers to an ecosystem of projects, most of which are not processing systems at all. Spark is a batch processing system at heart too. ... Spark Streaming is a stream processing system.

\* Linux shell commands

* https://docs.cs.cf.ac.uk/notes/linux-shell-commands/