Interview

1. Tell me about yourself
2. Difference between conceptual, logical and physical data model.
3. What is OLTP system
4. Normalization example
5. What is OLAP system
6. What is Data vault
7. What is SCD
8. What is OPS
9. What is Data lake
10. Different type of Databases
11. Python
    Shell
    Hive
    Pyspark
12. Data Architecture

PySpark

1. Introduction to Spark
2. Why Spark
3. memory management Spark
4. Context in Spark
5. initating Spark Job and Submit Job
6. Parameter to be submitted to Spark Jo
7. Spark Architecture
8. Spark Features
9. Spark support core, Sql, ml, Streami
10. RDD DataFrame Sql Dataset
11. Types OF RDD Action / Transformation
    - Select
    - filter
    - Groupy
    - windowing
    - writing
    - Sampling
    - repartitioning
    - coalescing
    - Count
    - distinct
    - Ordering
    - Top record
    - Joins
    - Set Operation
12. ways to create RDD
    - parallelize
    - from other RDD
    - Python list
13. Partition And repartition.
14. Extraction to HDFS / Save

(15) Performance

RDD Persistence
Caching
Spark Sql architecture
Broadcast Accumultor variable
Catalyst optimizer
Tungsten optimizer execution engine
Prior 2.0 volcano iterator model
After 2.0 whole stage code generation
Tungsten engine

Debugging

to Debug String
explain.

① Introduction to Spark / Why Spark

- Distributed parallel processing
- Horizontally scalable
- Unified technology can be used for SQL, ML, graph, Streaming
- Support multiple languages
- In memory processing
- Map reduce only supported map reduce so in real world other support filter / count so it was stateless, Spark support the same.
- Spark separate storage and processing and hence can run on different cluster manager

	Map reduce	Spark
1	Stateless	support filter
2	Support Java	support all in memory processing
3	Lot of ios	Unified cluster management
4	Different tool make maintainance difficult as they have their own cluster management	
5	Support Batch processing	Support Batch plus Streaming data.

③ memory management spark

Spark support 5 cluster manger

- Kubernetes
- Mesos
- ECS
- Standalone
- Yarn

Spark core is main component of spark and on top of it it has RDD Api to support

- Spark SQL
- Spark ML
- Spark Graph
- Spark Streaming

④ Spark session / Spark context

Spark 1

spark one has dependent context

- SparkContext
- SqlContext
- hiveContext

when we start pyspark it gives sc as SparkContext

Spark 2

spark has spark Session Object which has all three context

- sparkContext
- SqlContext
- hiveContext

when we start pyspark2 it gives spark as spark session object.

to get sparkContext you do sparkSession.sparkContext.

⑤/⑥ Spark-submit job is used to submit spark job to spark cluster for execution

Below are important command line parameter of spark-submit job

- master
- deploy-mode (client / cluster)
- conf (configuration file)
- properties-file (to submit our own conf file)
- driver-memory
- executor-memory
- num-executors
- jars (connector jars ODBC)
- packages (Avro Package)
- py-files (passing Python file to copy on cluster)

8/9 Resilient Distributed Dataset (RDD)

RDD has below features

Resilient
Distributed
Immutable
Lazy evaluation
Linage
fault tolerant
compile time safety

When to Use RDD ?

Performance is not an issue
Need control of processing
For Semistructured data
when schema of data is not known

RDD can't optimize by Spark because RDD data + operation is opaque to Spark and hence sql and dataframe are designed on top of RDD.

⑦ Spark Architecture

Scenario
200mB file in hdfs is to be loaded
map the data
filter the data
group the data
Store the data

- Client machine jvm Submit the job via spark-submit command
- depending on deploy-mode and cluster manager assuming cluster and yarn the application will be processed
- Yarn Resource manager consist of Application manager and Resource scheduler.
- Application manager launch the container (Data node) seeking spare equal to Driver memory and launch Driver Program inside Application master which is launched inside container.
- The Driver program consist of SparkContext, dvm and Spark Application

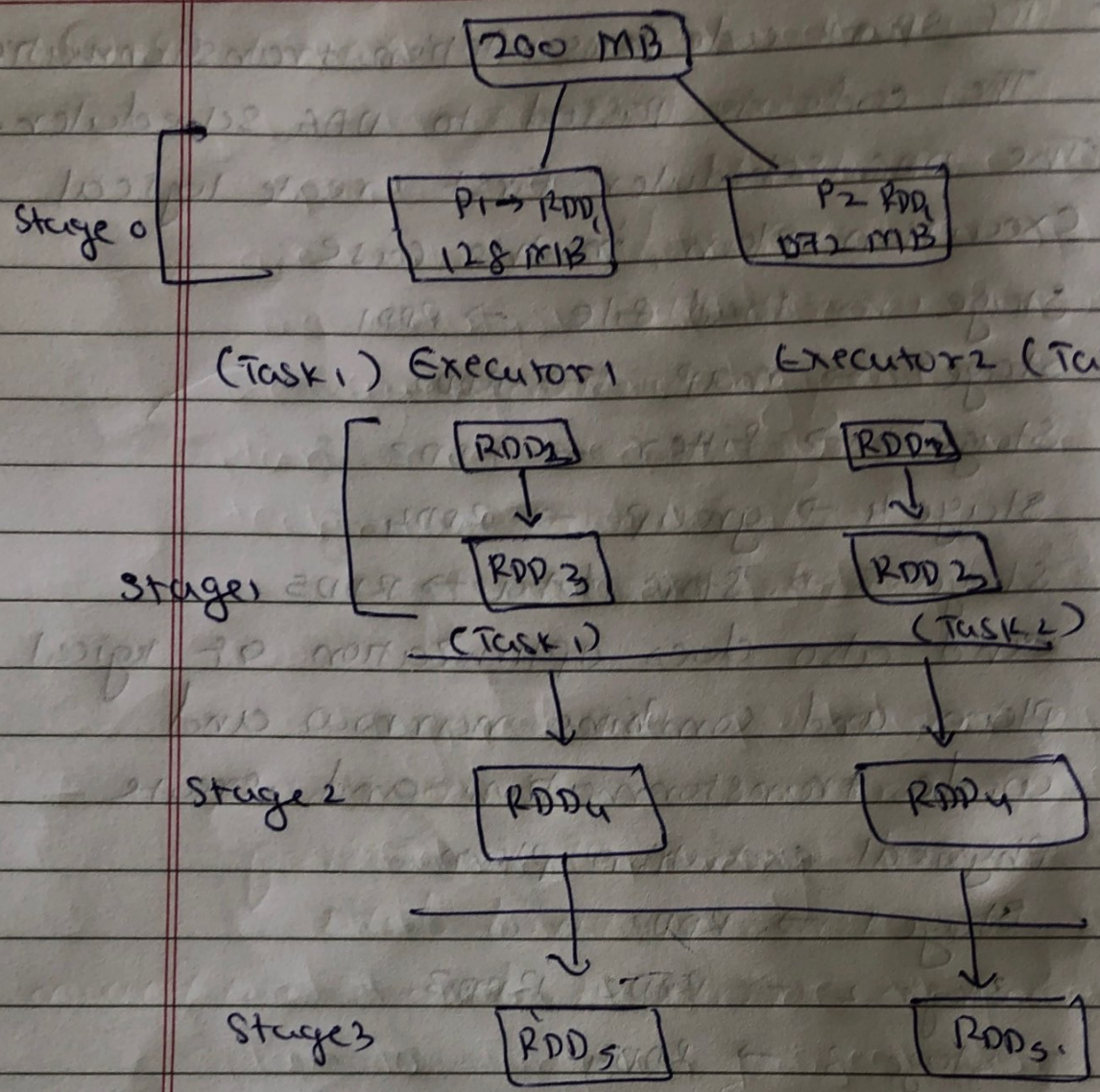
- If spark code has Action transformation The code is passed to DAG scheduler. The DAG scheduler first create logical execution plan, in this case

Stage 1 → load file → RDD1
Stage 2 → map → RDD2
Stage 3 → filter → RDD3
Stage 4 → group → RDD4
Stage 5 → store data → RDD5

and also does optimization of logical plan and combine narrow and wide transformation and create Physical execution plan.

Stage 1 → RDD1
Stage 2 → RDD2, RDD3
Stage 3 → RDD4
Stage 4 → RDD5

- Driver program need two partition to be created as file is of 200MB so it ask for 2 executor to Application master which negotiate with Application manger and two executor Program are created on worker node which status are passed by Node manager and Also the code is passed to these executor along with data



- Once the code is executed and result is outputed the individual node manager i.e executor passed output to driver program i.e Application master and it passed it to client
- The Application master deregister itself from Resource manager.

(11) RDD :- How to Create RDD
GRAC -> Aggregate key

(13) Shuffle Combiner Concept
Partition Coalesce
Use Coalesce to decrease partition
Use repartition to increase partition
repartition tyr to create equal size
partition and hence lot of shuffle happen.

(14) Hadoop only load text, csv and
Sequence file.

(15) Spark need to define storage level
and then persist once storage level
is set and then it cant be changed.
Unpersist method is used to unpersist.

(15) Broadcast variable as variable, constant
which can be shared across
executor once for node. It need
not be changed once broadcasted.

(16) Accumulator variable are used for
commutative or Additive or Associative
operation. This is used for counter.
Accumulator variable value are send to
Driver Program.
Broadcast and Accumulator variable
work only on Action.

(16) Spark Sql Architecture

Spark sql or dataframe queries are submitted to catalyst optimizer, The catalyst optimizer convert the query into logical execution plan and then it uses transform techniques such as

constant folding (literal identification)

Predicate pushdown → push filter

Column pruning → Remove unused column from scan

and it then pass the physical plan to Tungsten project for execution

Tungsten execution engine execute query in parallel.

Prior 2.0 Tungsten execution engine was not used in the volcano iterator model used to execute in parallel as it was working independently and also no dependency on step 1 and 2, it has disadvantage of extensive memory use hence whole stage code generation Tungsten engine was used which uses

Cost based optimizer: and it optimize all steges of execution and consider dependency.

DataFrame

Dataset + Schema = DataFrame

1. Introduction
2. How to Create dataFrame
3. Selecting data in dataFrame
   - Sql
   - table
4. Reading data from sources into df
5. UDF
6. Catalog
7. Datatypes and Row, Column functions
8. Practice
9. Built in Functions

① DataFrame = Dataset + Schema

② Command to create dataframe

Version

range

CreateDataFrame

- Python list
- Dict
- Rdd
- Row Object + Parallelize
- Pandas dataframe + 2D list

③ Spark sql from tempview and global tempview

Global tempview is accessible to all session

RDD + Dataframe + tempview + Spark sql.

③-⑨ Table Function

④ Reading data into dataframe

read.load is used to load different file format

csv orc avro

text Parquet

hive : since hive is integrated so Spark SQL

Jdbc if not then we need to pass Connector.

JDBC data read into spark we can Partition using Parameter

lower bound upperbound and partition
(upperbound - lowerbound) / Partition = Stride
part = lowerbound + stride

⑤ UDF in DF
UDF are user defined function defined in code.
@ Annotation is required to use Python function in dataframe
However to use Python function in spark.sql it need to be register first
UDF are not shared accross session
new-session is used to initiate new session in spark

⑥ Catalog gives details about metadata of dataframe
list databases tables columns
is cached cachedTable uncacheTable
clearcache
recoverpartition refreshtable
refreshByPath
dropGlobalTempView
dropTempView
listFunctions
registerFunctions will register Function

(7) Datatypes in Spark

Integer - Int Float Double

String - Char varchar2 String

Boolean

Date - Date Timestamp

complex - Array, map, struct

All are Types in dataFrame and has alias in Spark Sql

Row - Row class : count, index, toDict()

column - select
orderby
where
null check - null function

Column function apply on Right side of the column.

→

⑧ Practice till now

Select
where / filter
orderby
sortwithpartition
union / unionall
intersect / intersectall
unionbyname
exceptAll (similar to minus)
inner join / join
full Outer join
Left join
Right join
cross join
left anti
left semi
self join
multi condition join
multi dataframe join

Aggregation function
summary
avg
min
max
sum
sumdistinct
count

countdistinct

first

last

collect_set

collect_list

Group by Api

avg mean count min max sum agg

pivot apply

filter Similar to having

no unpivot function so we need to use stack and expr function in Spark SQL

use selectexpr in dataFrame

window function

Ranking

Row_number

Dense

Denserank

percent_rank

rank

ntile

Cume_dist

Analytical
- lag
- lead

Aggregate
avg, sum, min, max, count, first, last

Other
- range between
- rowsbetween
  - unbounded following
  - unbounded preceeding
  - currentrow

first, last, greatest, least, collect_list

Sampling
- Sample
- SampleBy

(8) Built in Functions
- monotically_increasing_id()
- lit() - literal static column
- expr() -
- spark_partition_id()
- rand() / randn()

String Functions

Split
length
lower, upper, initcap
ltrim, rtrim, trim
lpad, rpad
reverse
repeat
hex
concat
concat_ws
Substring
Substring_index
instr
- locate
translate
- overlay 3.0 Spark only

Regexp Function
regexp_extract
regexp_replace
rlike()

Date
Page

Date Functions

current_date
current_timestamp
next_day
last_day - of month
dayofweek
dayofmonth
dayofyear
weekofyear
second
minute
hour
month
quarter
year
months_between
date_add( )
date_sub( )
add_months( )
datediff( )
date_trun( )
date_format( )
unix_timestamp
to_timestamp
from_unixtime
from_utc_timestamp
to_date

Null Fnctions

isNull

isNan

nanvl

Coalesce

Collection Functions

Array map struct :

size - map, Array

element_at - map, Array

struct : creat struct

array : create array

array_max

array_min

array_distinct

array_repeat

flatten

slice

array_position

array_remove

array_sort

Sort_array - we can pass what to be nuy

array_contains

array_union

array_except

array_intersect

array_join

arrays_zip
arrays_overlap
shuffle

create_map
map_from_entries
map_from_arrays
map_keys
map_values
map_concat
sequence

Null Functions

drop : drops null values
df.na.drop()
fill : fill null values.
replace

maths Function

abs
exp
factorial
sqrt
cbrt
pow - Power

floor
ceil
round
trunc
signum

more

explode
explode_outer
posexplode
posexplode_outer
flatten

Formatting

format_number
format_string

Json Function

from_json
to_json
Json_tuple
schema_of_json
get_json_object

Partition

repartition
coalesce

Extraction

csv txt parquet json orc hive JDBC