Crack Data Engineering Interviews

Apache Hive

120+ Real-time Interview Questions

THE BRIGHT SIDE

www.youtube.com/@ashok.thebrightside

ashok.thebrightside@gmail.com

1. What is Apache Hive?

- Open source software - reading, writing and managing large data set files in HDFS/Hbase.
- It gives an SQL-like interface to query data
- Impala/Trino/Presto

2. What is the default database of Hive to store its metadata?

- Derby
  - Embeded metastore of Hive.
  - We can configure to use any DB for metastore.

3. What is the default execution engine in Hive?

- Map Reduce

4. How to change the execution engine in Hive?

- SET hive.execution.engine = tez;

5. Which component of Hive connects to Hadoop cluster for query execution?

- Execution engine.

6. Which component of Hive converts SQL query to jar file for execution?

- Execution engine.

7. How do you see the execution plan of a query?

- Use EXPLAIN

```
hive (default)> explain select * from customer;
OK
STAGE DEPENDENCIES:
  Stage-0 is a root stage

STAGE PLANS:
  Stage: Stage-0
    Fetch Operator
      limit: -1
      Processor Tree:
        TableScan
          alias: customer
          Statistics: Num rows: 1 Data size: 258 Basic stats: COMPLETE Column stats: NONE
          Select Operator
            expressions: custid (type: int), firstname (type: string), lastname (type: strin
pe: string), transactamt (type: int)
            outputColumnNames: _col0, _col1, _col2, _col3, _col4, _col5, _col6
            Statistics: Num rows: 1 Data size: 258 Basic stats: COMPLETE Column stats: NONE
            ListSink
```

8. How do you see the database in which you are currently working on?

- Set hive.cli.print.current.db = true

```
hive (custdb)> set hive.cli.print.current.db=false;
hive> set hive.cli.print.current.db=true;
hive (custdb)>
```

- beeline --showDbInPrompt=true

9. What is the default storage location of Hive tables?

- /user/hive/warehouse

10. What do you do to have a table which should exists only for that particular Hive session?

- Create Temporary table.

```
hive> create temporary table t3(col1 int, col2 string);
OK
Time taken: 0.047 seconds
```

- Hive 0.14 onward supports temporary tables. You can use them as a normal table within a user session. The location will be /tmp/hive/<user>/*.

11. What are the limitations of Hive Temporary tables?

- It is available only to that session where it is created.
- Automatically removed when session terminated.
- **Partition columns** option is not supported on temporary tables.
- Creating **indexes** on temporary table is not allowed.
- The user cannot access the permanent table with same name as temporary tables during that session without dropping or renaming the temporary table.

12. What are the different types of tables in Hive?

- Temporary Table
- Managed Table
- External Table

13. Scenario: We need to store one of the feed data into Hive table and expose it for consumption. What are the steps you will take?

- Talk to source team and get schema information of the feed data like data type, data size, volume of data etc.
- Talk to business team who is going to consume the data and find the search criteria they will be using.
- Based on the above two create table with partitioning/bucketing/indexing etc.

14. What are the default terminators in Hive?

- Field Terminator – ctrl + A
- Line Terminator - \n

15. How do you get the schema information of tables in Hive?

- DESCRIBE FORMATTED <Table Name>

16. Can we use DESCRIBE for Database to see the details?

- Yes. It can be used for Tables, databases and views.

17. How do you load data from Linux location to Hive table location using Hive script?

- LOAD DATA LOCAL INPATH '/home/hduser/hive/data/txns' INTO TABLE txnrecords;
- LOAD DATA LOCAL INPATH '/home/hduser/hive/data/txns' OVERWRITE INTO TABLE txnrecords;

It copies the content from Linux to Hive table location

18. How do you load data from HDFS location to Hive table location using Hive script?

- LOAD DATA INPATH '/user/hduser/txns1' INTO TABLE txnrecords;
- LOAD DATA INPATH '/user/hduser/txns1' OVERWRITE INTO TABLE txnrecords;

It moves data from HDFS location to Hive table location.

19. How do you convert row of records into single object in Hive?

- Use Collect_Set() or Collect_List()

Collect_Set() removes duplicates

```
hive (sales_db)> select * from purchase_transaction;
```

Output:

```
John 8 160
John 8 160
Karen 9 100
Peter 10 660
Peter 10 600
Karen 1 100
Peter 2 200
Peter 3 600
John 4 80
Karen 5 120
William 6 170
John 7 140
```

```
hive (sales_db)> select sales_rep_name , collect_set(transaction_amt) from purchase_transaction group by sales_rep_id;
```

Output

```
John [160,80,140]
Karen [100,120]
Peter [660,600,200]
William [170]
```

```
hive (sales_db)> select sales_rep_name, collect_list(transaction_amt) from purchase_transaction group by sales_rep_name;
```

Output

```
John [160,160,80,140]
Karen [100,120]
Peter [660,600,200]
William [170]
```

20. How do you convert array of data into individual rows?

- Explode – can't use with other columns
- Lateral View Explode – can use other columns
- PosExplode - explode with index.

```
hive> select * from std_course_details;
OK
std_course_details.std_id       std_course_details.stud_name    std_course_details.location     std_course_details.course
1       vamshi  hyd     ["hive","hadoop","spark"]
2       chandana        bangalore       ["reactjs","javascript"]
3       Divya   pune    ["python","pyspark","airflow","spark"]
4       srikanth        pune    ["java","springboot"]
5       sreethan        pune    ["c","c++"]
Time taken: 0.203 seconds, Fetched: 5 row(s)
```

```
hive> select explode(course)  from std_course_details;
OK
col
hive
hadoop
spark
reactjs
javascript
python
pyspark
airflow
spark
java
springboot
c
c++
Time taken: 0.206 s
```

```
hive> select std_id,stud_name,location,courses  from std_course_details LATERAL VIEW explode(course) courses_list as courses;
OK
std_id  stud_name       location        courses
1       vamshi  hyd     hive
1       vamshi  hyd     hadoop
1       vamshi  hyd     spark
2       chandana        bangalore       reactjs
2       chandana        bangalore       javascript
3       Divya   pune    python
3       Divya   pune    pyspark
3       Divya   pune    airflow
3       Divya   pune    spark
4       srikanth        pune    java
4       srikanth        pune    springboot
5       sreethan        pune    c
5       sreethan        pune    c++
Time taken: 0.092 seconds, Fetched: 13 row(s)
```

posexplode

```
Time taken: 0.108 seconds, Fetched: 4 row(s)
hive> select id, his_desig from arraytbl lateral view posexplode(desig) p as his_desig where id =5;
FAILED: SemanticException [Error 10083]: The number of aliases supplied in the AS clause does not match the number of columns output by the UDTF expected 2 aliases but got 1
hive> select id,des_id, his_desig from arraytbl lateral view posexplode(desig) p as des_id,his_desig where id =5;
OK
5       0       A
5       1       B
5       0       A
5       1       B
Time taken: 0.047 seconds, Fetched: 4 row(s)
hive> select * from arraytbl;
OK
5       Ashok   100     ["A","B"]       Chn
5       Ashok   100     ["A","B"]       Chn
51      Ashokl  100     ["Al","Bl"]     Chn
1       Arvindh 40000   ["SWE","Analyst","SAnalyst","PL"]       hyd
2       Bala    30000   ["SWE","Analyst","SAnalyst","PM"]       hyd
3       Chandra 50000   ["SWE","Analyst","PL","PM"]     Che
4       Gokul   30000   ["SWE","Analyst","SAnalyst",""] hyd
Time taken: 0.039 seconds, Fetched: 7 row(s)
hive> select id,des_id, his_desig from arraytbl lateral view posexplode(desig) p as des_id,his_desig where id =5;
OK
5       0       A
5       1       B
5       0       A
5       1       B
Time taken: 0.032 seconds, Fetched: 4 row(s)
```

name	phone_numbers	cities
AAA	["365-889-1234", "365-887-2232"]	["Hamilton"]["Burlington"]
BBB	["232-998-3232", "878-998-2232"]	["Toronto", "Stoney Creek"]

Applying a lateral view explode on the above table will expand the both Telephone and Cities and do a cross join, your final table will look like this.

name	phone_numbers	cities
AAA	365-889-1234	Hamilton
AAA	365-887-2232	Hamilton
AAA	365-889-1234	Burlington
AAA	365-887-2232	Burlington
BBB	232-998-3232	Toronto
BBB	878-998-2232	Toronto
BBB	232-998-3232	Stoney Creek
BBB	878-998-2232	Stoney Creek

when to use lateral view posexplode in hive

```
    name,
    phone_number,
    city
from temp.test_laterla_view_posexplode
lateral view posexplode(phone_numbers) pn as pos_phone, phone_number
lateral view posexplode(cities) pn as pos_city, city
where
    pos_phone == pos_city
```

With above query you will get following results, where phone number is mapped with corresponding city.

name	phone_number	city
AAA	365-889-1234	Hamilton
AAA	365-887-2232	Burlington
BBB	232-998-3232	Toronto
BBB	878-998-2232	Stoney Creek

Converting columns to rows (UNPIVOT) in hiveql

I have a table with a structure like this:

```
column1, column2, column3, X1, X2, X3, X4
A1,      A2,      A3,      5,  6,  1,  4
```

I would like to convert this into

```
column1, column2, column3, Key, Value
 A1,      A2,      A3,      X1,  5
 A1,      A2,      A3,      X2,  6
 A1,      A2,      A3,      X3,  1
 A1,      A2,      A3,      X4   4
```

```
select column1, column2, column3, m_key, m_val from
    (select column1, column2, column3, map("X1", X1, "X2", X2, "X3", X3, "X4", X4) as map1
    from table1) as t1
lateral view explode(map1) xyz as m_key, m_val
```

21. What are the different types of indexes available in Hive?

- Compact Index – use in high cardinal column
- Bitmap – use in low cardinal column

22. What are the different types of partitions available in Hive?

- Static – based on some external factors
- Dynamic – based on data
  - set hive.exec.dynamic.partition=true;
  - set hive.exec.dynamic.partition.mode=nonstrict;

#	Static Partitioning	Dynamic Partitioning
Partition Creation	We need to manually create each partition before inserting data into a partition	Partitions will be created dynamically based on input data to the table.
Suitable for	We need to know all partitions in advance. So it is suitable for use cases where partitions are defined well ahead and are small in number	Dynamic partitions are suitable when we have lot of partitions and we can not predict in advance new partitions ahead of time.
Examples	Departments, State Names, etc	Date, city names etc

Static Partition:

load data local inpath '/home/hduser/hive/student_details2' into table student partition(course= "hadoop");

Dynamic partition:

- load data local inpath '/home/hduser/hive/student_details' into table stud_demo;
- insert into student_part partition(course) select id, name, age, institute, course from stud_demo;

23. Where NULL and Empty values will be stored in case of partitions?

- __HIVE_DEFAULT_PARTITIONS dir.

24. Does partition values are case sensitive?

- Yes

25. Can we do dynamic partition using LOAD DATA command?

- NO

26. Whether metadata will be updated if we load partition data using LOAD DATA command?

- Yes
  - load data local inpath '/home/hduser/hive/student_details2' into table student partition(course= "hadoop");

27. Can we do update and delete in Hive table?

- By default, NO.
- Yes, if transaction is enabled and table is in ORC format and bucketed.
  - **TBLPROPERTIES ('transactional'='true')**
  - **set hive.enforce.bucketing = true;**
- **Limitations:**
  - External tables are not supported.
  - Only ORC format is supported.
  - LOAD is not supported on ACID transactional Tables. Hence use INSERT INTO.
  - On Transactional session, all operations are auto commit as BEGIN, COMMIT, and ROLLBACK are not yet supported.

28. How partition helps in managing data?

- We can remove the portion data from table easily by just deleting the partition.
- We can update the portion of data with updated value by just overwriting partition folder data.

29. **Scenario:** How do you load only few fields from file?

- It is a two step process
  - Load all data to managed table – LOAD DATA LOCAL INPATH
  - Do insert select with required fields from managed to External table.

30. **Scenario:** I manually created a partition folder and kept data inside it. But when I query from hive table for that partition value, I am not getting data. What should I do?

- Run msck repair or manually add the partition.
  - It is **Hive's** metastore consistency check

31. **Scenario:** I deleted some of the partition folder manually and then executed msck repair, will it remove the partition from metadata?

- NO, we have to manually DROP the partition.

32. **Scenario:** I created a new partition folder with a name of non partition column and running msck repair. What will happen, it will update the metadata?

- No, it will throw error. (We can add only new value to the partition column.)

33. How do you choose partition type?

- Dynamic partition – to create partitions based on data.
- Static partition – to create partitions based on some external parameters.

34. In what scenario we can use ADD PARTITION than MSCK REPAIR?

- In case of ADD PARTITION we can specify the location of partition value.
- Ex – ALTER TABLE emp ADD PARTITION (dept = 'dev') LOCATION '/user/hive/dept = 'ACCT'

35. Can we perform DELETE and TRUNCATE on Hive Table?

- Truncate is possible (only in Managed table)
- Delete is not possible by default. (transactional,ORC, bucketing)

36. What are the pros and cons of Hive partitioning?

Pros

- Improves search performance
- Manages data efficiently
- Easy to update or delete portion of data

Cons

- It adds overhead on Namenode to maintain partition directory information. If partitions are done based on high cardinal column by mistake then it will add more burden to Namenode.

37. How do you segregate or partition the data if table has no low cardinal column?

- Use Bucketing

38. How many reducers will be called when we insert data into Bucketed table?

- Number of reducers will be equal to number of buckets

39. How Bucketing works?

- It segregates the data by applying Mod operation with bucket number on hash value of bucket column data.

40. In which bucket NULL and empty string will go?

- It will be stored in bucket zero.

41. Why bucket join is faster than partition join?

- In case of bucketing we will have almost equal number of data in each bucket and only respective buckets are picked up and performed join operations.

42. How do you export hive table data to Linux location?

- Use INSERT LOCAL DIRECTORY
  - INSERT LOCAL DIRECTORY '/home/hduser/data' ROW FORMAT DELIMITED FIELDS-TERMINATED-BY ',' SEELCT * FROM TEMP;

43. **Scenario:** I have created more partition folders manually. What is the easiest way to add those partition information in metastore?

- Msck repair.

44. **Scenario:** You have 10 columns in Hive table, how do you export column 1,3,5 alone into Sqoop?

- It is a two step process
  - Create a managed table with required columns and so insert select of required columns from source table.

    (OR)

    INSERT DIRECTORY '/user/hduser/data' ROW FORMAT DELIMITED FIELDS-TERMINATED-BY ',' SEELCT col1, col3, col5 FROM TEMP;
  - Then do Sqoop Export from the directory. --export-dir

45. How do you run Hive script from Linux terminal?

- Hive –f hiveFile.hql
  - $ beeline --hivevar var1=value -f file.sql
- Hive –e “select * from custdb.sample”
  - $ beeline -e "select current_date()"

46. What is the default file format in Hive?

- Text File.

47. What are the file formats supported by Hive?

- Text File.
- SequenceFile.
- RCFile.
- Avro Files.
- ORC Files.
- Parquet.

48. What is SerDe in Hive?

- SerDe is short for Serializer, Deserializer
- Hive uses SerDe for storing in HDFS and retrieving data from HDFS.
- Default is LazySimpleTextSerDe
- Built-in SerDe
  - Avro (Hive 0.9.1 and later)
  - ORC (Hive 0.11 and later)
  - RegEx
  - Thrift
  - Parquet (Hive 0.13 and later)
  - CSV (Hive 0.14 and later)
  - JsonSerDe (Hive 0.12 and later in hcatalog-core)

49. Scenario: How do you process XML data in Hive?

- Use XML SerDe library
  - Specify Fields to be parsed using Xpath
  - Specify Input and Output format
  - In TBL properties, give the nodes to be considered for processing.

```
CREATE [EXTERNAL] TABLE <table_name> (<column_specifications>)
ROW FORMAT SERDE 'com.ibm.spss.hive.serde2.xml.XmlSerDe'
WITH SERDEPROPERTIES (
["xml.processor.class"="<xml_processor_class_name>",]
"column.xpath.<column_name>"="<xpath_query>",
...
["xml.map.specification.<xml_element_name>"="<map_specification>"
...
]
)
STORED AS
INPUTFORMAT 'com.ibm.spss.hive.serde2.xml.XmlInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
[LOCATION "<data_location>"]
TBLPROPERTIES (
"xmlinput.start"="<start_tag ",
"xmlinput.end"="<end_tag>"
);
```

50. What are the complex types in Hive?

- Array – to store elements of same type
- Map – to store elements as key-value pair. If you dont know the # of elements to be stored.
- Struct – to define consistent structure to data
- ~~Union~~

51. How to do Sqoop-Hive integration?

- Generally, we do Sqoop – HDFS – (load data inpath) HIVE
- $ sqoop import --connect jdbc:mysql://localhost/serviceorderdb --username root -P --table productinfo **--hive-import --hive-table** serviceorderdb.productinfo -m 1
  - It first loads data into hdfs dir
  - Then it internally does load data inpath from hdfs dir to hive table

52. **Scenario:** how do you handle incorrect date format in Hive? Ex – You are getting date in the form of dd-MM-YYYY

```
hive> create table employee (id int, joiningdate date);
OK
Time taken: 0.185 seconds
hive> insert into employee values(1,'31-01-2021')
    > ;
Query ID = hduser_20210402220817_f46f40e4-60fd-42e2-9e58-7f9e1528ae81
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1617426321935_0001, Tracking URL = http://CodeMechanics:8088/proxy
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1617426321935_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2021-04-02 22:08:26,599 Stage-1 map = 0%,  reduce = 0%
2021-04-02 22:08:32,953 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.19 sec
MapReduce Total cumulative CPU time: 1 seconds 190 msec
Ended Job = job_1617426321935_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://localhost:54310/user/hive/warehouse/employee/.hive-staging_hi
Loading data to table default.employee
Table default.employee stats: [numFiles=1, numRows=1, totalSize=5, rawDataSize=4]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.19 sec   HDFS Read: 3669 HDFS Write: 77 SU
Total MapReduce CPU Time Spent: 1 seconds 190 msec
OK
Time taken: 16.855 seconds
hive> select * from employee;
OK
1       NULL
Time taken: 0.066 seconds, Fetched: 1 row(s)
```

```
hive> insert into employee values(2,'2021-01-31');
Query ID = hduser_20210402221221_1c6b5246-7cd8-4d23-b5c9-a340734dfedc
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1617426321935_0002, Tracking URL = http://CodeMechanics:8088/proxy/application_161742
Kill Command = /usr/local/hadoop/bin/hadoop job  -kill job_1617426321935_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2021-04-02 22:12:26,802 Stage-1 map = 0%,  reduce = 0%
2021-04-02 22:12:33,146 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.12 sec
MapReduce Total cumulative CPU time: 1 seconds 120 msec
Ended Job = job_1617426321935_0002
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://localhost:54310/user/hive/warehouse/employee/.hive-staging_hive_2021-04-02_22-12
Loading data to table default.employee
Table default.employee stats: [numFiles=2, numRows=2, totalSize=18, rawDataSize=16]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.12 sec   HDFS Read: 3757 HDFS Write: 85 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 120 msec
OK
Time taken: 13.256 seconds
hive> select * from employee;
OK
1       NULL
2       2021-01-31
Time taken: 0.057 seconds, Fetched: 2 row(s)
hive>
```

Cont.

- It is a two step process
  - Insert the date into string column in managed table.
  - Do Substring on string date field and format the date as expected and load to external table.
- select from_unixtime(unix_timestamp('2016/06/01','yyyy/MM/dd'),'yyyy-MM-dd') from table1;

```
hive> select unix_timestamp('2016/06/01','yyyy/MM/dd') from table1;
OK
1464739200
Time taken: 2.469 seconds, Fetched: 1 row(s)
hive> select from_unixtime(unix_timestamp('2016/06/01','yyyy/MM/dd'),'yyyy-MM-dd') from table1;
OK
2016-06-01
Time taken: 1.314 seconds, Fetched: 1 row(s)
hive>
```

53. Which compression technique is good for cold data?

- Gzip – for data which is accessed infrequently, provides more compression.
- Snappy – for data which is accessed frequently, provides less compression and good splittability.

Compression Ratio : GZIP compression uses more CPU resources than Snappy or LZO, but provides a higher compression ratio.

General Usage : GZip is often a good choice for **cold** data, which is accessed infrequently. Snappy or LZO are a better choice for **hot** data, which is accessed frequently.

Snappy often performs better than LZO. It is worth running tests to see if you detect a significant difference.

Splittablity : If you need your compressed data to be splittable, BZip2, LZO, and Snappy formats are splittable, but GZip is not.

GZIP compresses data 30% more as compared to Snappy and 2x more CPU when reading GZIP data compared to one that is consuming Snappy data.

LZO focus on decompression speed at low CPU usage and higher compression at the cost of more CPU.

For ***longer term/static*** storage, the GZip compression is still better.

54. How do you parse the data if you have used –enclosed-by option?

```
sqoop import --connect jdbc:mysql://localhost/custpayments --username root --password root -table customers -m 2 --split-
by customernumber --target-dir /user/hduser/custdata/ --delete-target-dir --enclosed-by '\"';


create external table custmaster (customerNumber int,customername varchar(20),contactlastname varchar(20),contactfirstname
varchar(20))
ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
WITH SERDEPROPERTIES (
 "separatorChar" = ",",
 "quoteChar" = "\"")
LOCATION '/user/hduser/custdata/';
```

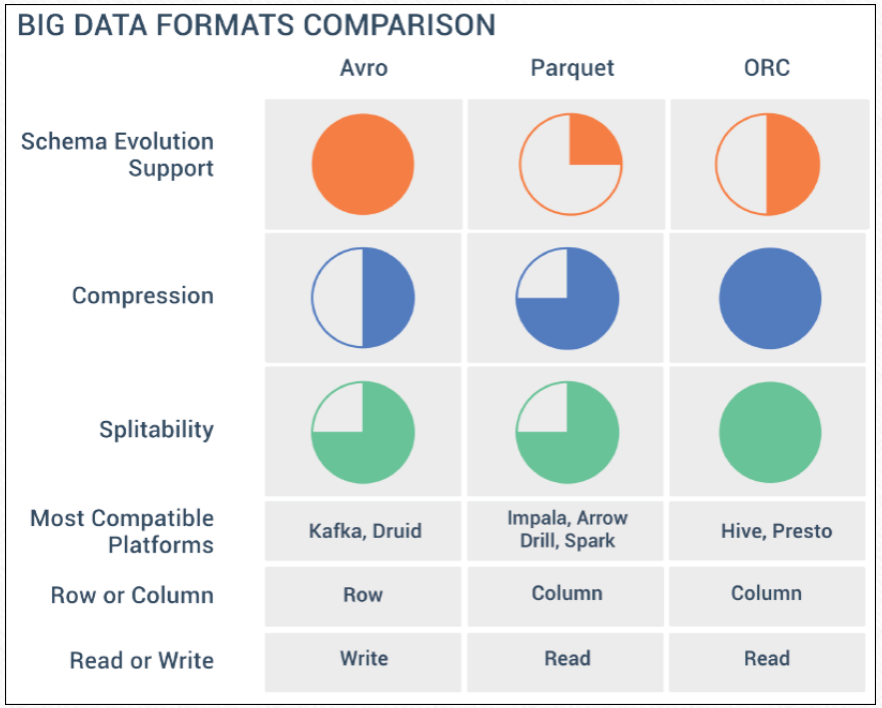
55. Renaming a table will rename directory as well?

- Yes, in case of managed table.

56. How to rename external table directory by renaming table?

- We cant rename the directory by just renaming the external table name.
- Make it to Managed Table, rename the table and then make it back to external table.
  - SET TBLPROPERTIES('EXTERNAL'='TRUE');

57. Data Format Comparison



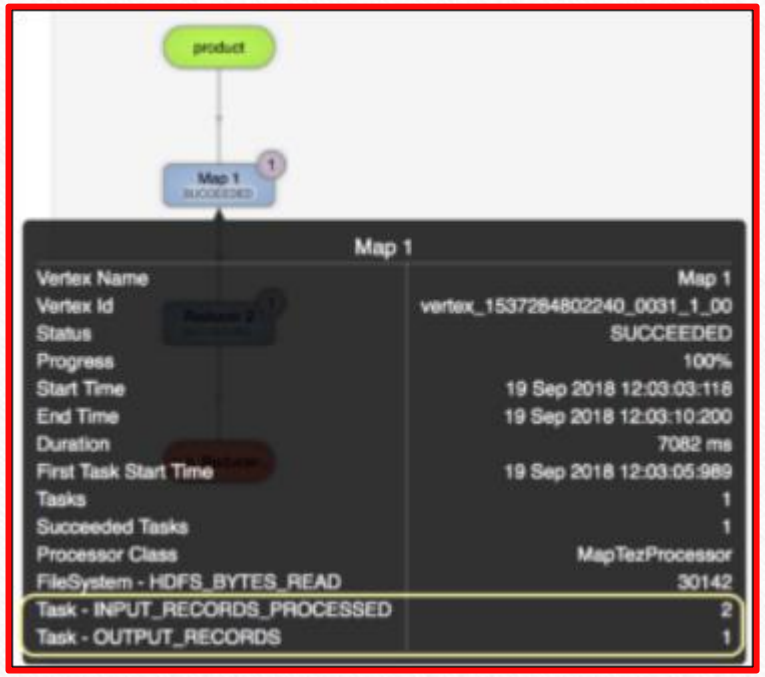
ORC – Hortonworks
Parquet - Cloudera

58. What is Vectorization?

- This enables hive to fetch 1024 rows in a batch rather fetching single record at a time.
- set hive.vectorized.execution.enabled = true;
- Example: Below table has 1560 rows

```
set hive.vectorized.execution.enabled = false;
select count(*) from foodmart.product
where
    product.product_name like "%Washington%"
```

```
set hive.vectorized.execution.enabled = true;
select count(*) from foodmart.product
where
    product.product_name like "%Washington%"
```

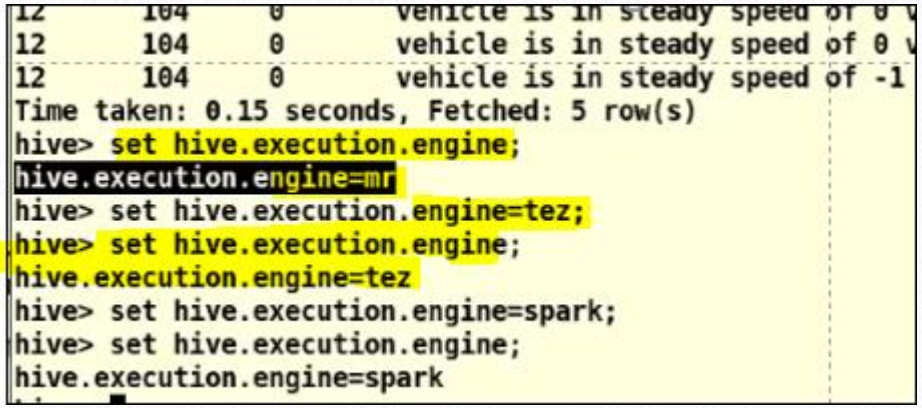


59. What are the advantages of bucketing?

- Bucketed tables allow faster execution of map side joins, as the data is stored in equal-sized buckets/parts.
- Efficient sampling happens for bucketed tables when compared to non – bucketed ones.
- As in partitioning, the Bucketing feature also offers faster query performance.

60. How do you see which execution engine hive currently uses?

- Use SET hive.execution.engine

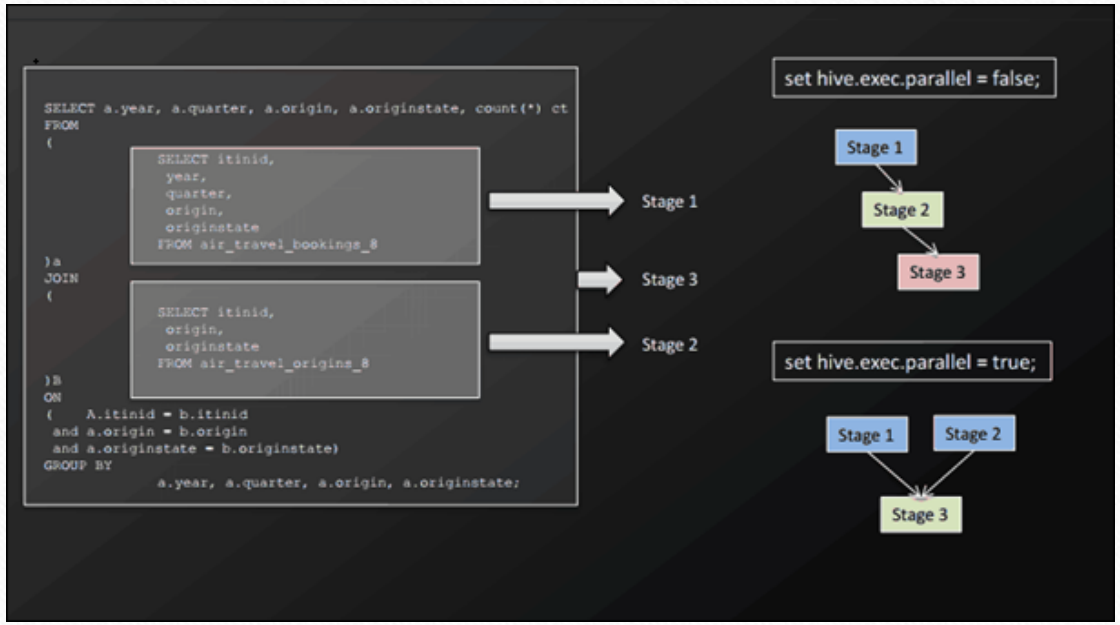


61. What is hive thrift server?

- A service which permits a remote client to submit requests to Hive.
- It is also called HiveServer.
- Since It is built on Apache Thrift, it is called thrift server.

62. How to make queries to execute in parallel?

- It allows independent tasks in a query to run in parallel.
  - SET *hive.exec.parallel = true*



63. If underlying data in HDFS is not meeting the type of HIVE table fields, what will happen?

```
hive> !hadoop fs -cat /user/hive/warehouse/employee/*
    > ;
21/04/03 21:49:35 WARN util.NativeCodeLoader: Unable to load n
1^AN
2^A2021-01-31
hive>
```

No parser error

```
hive> create table emp (id int, id2 int) LOCATION '/user/hive/warehouse/employee';
OK
Time taken: 0.229 seconds
hive> select * from emp;
OK
1       NULL
2       NULL
Time taken: 0.243 seconds, Fetched: 2 row(s)
hive>
```

64. Does Hive is suitable for OLTP?

- No. Arbitrary UPDATE/DELETE is not possible. Hive is suitable for OLAP.

65. How to view partition information of a table?

```
hive> show partitions emp_part;
OK
partition
dept=ACCT
dept=IT
dept=__HIVE_DEFAULT_PARTITION__
Time taken: 0.46 seconds, Fetched: 3 row(s)
hive>
```

66. Can we create partition columns after the table has been created?

- NO

67. Can we use LOAD DATA command for loading data into bucketed table?

- NO. As segregation of data into buckets needs intelligence, will be taken care by MR.

68. How do you run Hadoop commands from Hive CLI?

- Use !
- Hive> !hadoop fs –cat /user/hive/warehouse/employee/*

69. Explain what happens here

sqoop import --connect jdbc:mysql://localhost/custdb --username root --password root --table customers --hive-import --hive-table retail.customer -m 1;

- Sqoop creates dir **customer** at /user/hduser/. If the dir customer already exists at /user/hduser/, it will throw error.
- Sqoop import table customer to /user/hduser/customer
- Create HIVE table based on DDL structure of rdbms, if table doesn't exists. If table exists, this step will not occur.
- Load data inpath from sqoop - step 1 location to Hive table location - warehouse
- Delete the customer folder at /user/hduser/customer which is used for loading data.

70. Partition folder and data is there in HDFS but in HIVE select * you are not able to see the data. Why?

- Metastore is not updated. Do alter table and add partition OR use msck repair tablename

71. Can we use partition column for bucketing. Can we do bucketing and partition on same column.

- No.
  - We cant create bucketing on partition column, because partition column is virtual column

72. How many bucket file will be created and where it will be placed if we create 2 bucket on a table which has two partitioned column?

```
CREATE TABLE cust1(id int,name varchar(10), country varchar(10),state varchar(10))

CREATE TABLE cust2(id int,name varchar(10)) PARTITIONED BY(country string, state string) clustered by (id) into 2 buckets;

insert into cust1 values(1,'ashok','INDIA','TN'),(2,'vino','INDIA','BL'),(3,'sharmi','US','TEXAS'),(4,'Naveen','US','CA')

insert into cust2 partition(country,state) select id, name, country,state  from cust1;
```

```
INDIA
        BL
                0000000_0 - has data as 2(id column) % 2 = 0
                0000001_0 - no data
        TN
                0000000_0 - no data
                0000001_0 - has data
US
        CA
                0000000_0 - has data
                0000001_0 - no data
        TEXAS
                0000000_0 - no data
                0000001_0 - has data
```

73. Is it good to use index in Hive?

- NO
  - Indexing uses separate external table to hold indexes. So, it occupies extra space
  - Building indexes will take time and slow down performance during that time.
- Use ORC instead which has built in Indexes.

74. How to find total managed and external tables in HIVE?

- Select TableType, count(*) from TBLS

75. Can we create External table using CTAS?

- NO

76. How to create table structure from another table without data?

- create table new_table_name LIKE old_table_name

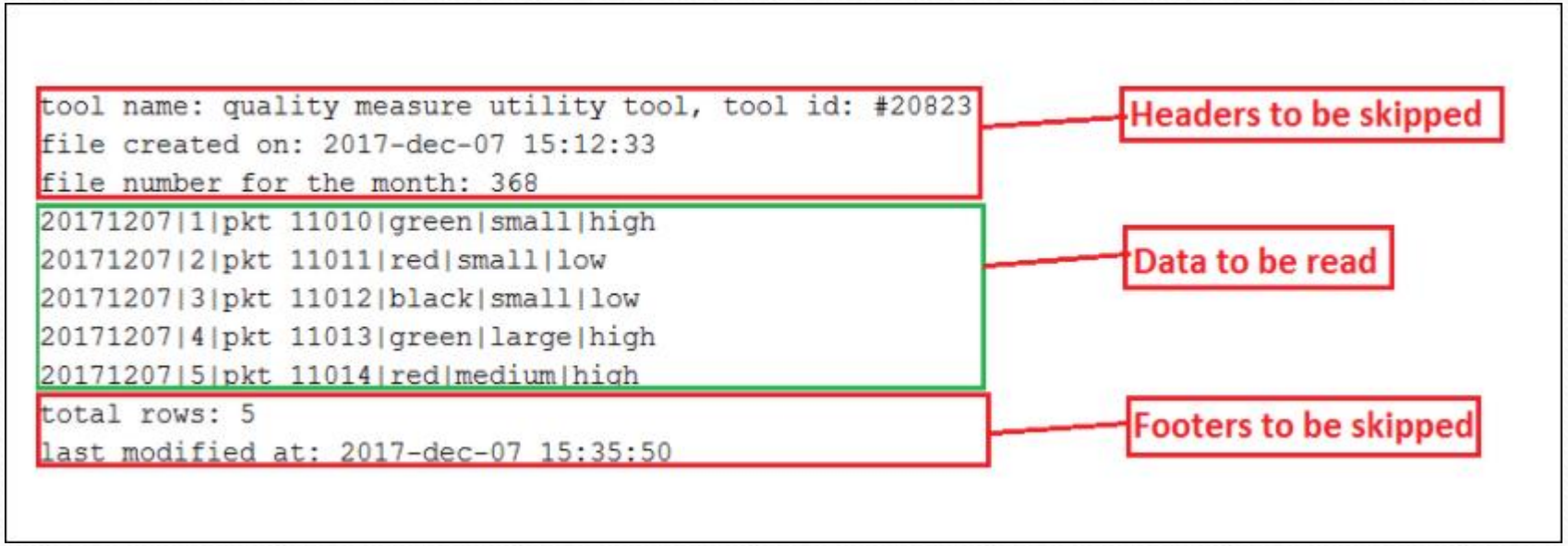
77. How to delete certain records from Hive table which is not Transactional supported?

- insert overwrite table insurance partition(datadt,hr) select * from insurance where issuerid1 is not null;

78. How do you restrict user from inserting data into table after first insert?

- SET TBLPROPERTIES("immutable"="true")
- First insert and overwrite is allowed.

79. How do you skip Header and Footer rows in Hive



Cont.

Use TBLPROPERTIES

skip.header.line.count

skip.footer.line.count

```
[code lang="sql"]
CREATE DATABASE IF NOT EXISTS testdb;
USE testDB;

DROP TABLE IF EXISTS testdb.sample_table;
CREATE TABLE IF NOT EXISTS testdb.sample_table
(
    log_date string,
    row_num int,
    pkt_number string,
    pkt_color string,
    pkt_size string,
    pkt_quality string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '|'
LINES TERMINATED BY '\n' STORED AS TEXTFILE
TBLPROPERTIES('skip.header.line.count'='3', 'skip.footer.line.count'='2');
```

80. How do you handle Java Heap OutOfMemory issue?

- Set the container and heapsize in Hive to resolve this error:
  - set mapreduce.map.memory.mb=4096; (container physical memory)
  - set mapreduce.reduce.memory.mb=4096; (container physical memory)
  - set mapreduce.map.java.opts=-Xmx3686m; (Java heap memory)
  - set mapreduce.reduce.java.opts=-Xmx3686m; (Java heap memory)

The Java heap is the area of memory used to store objects instantiated by applications running on the JVM.

The flag Xmx specifies the maximum memory allocation pool for a Java Virtual Machine (JVM), while Xms specifies the initial memory allocation pool.

81. **Scenario:** If data is present in multiple subdirectories in HDFS how do you retrieve those in Hive table?

```
/data/access/web1/2014/09
/data/access/web1/2014/09/access-20140901.log
[... etc ...]
/data/access/web1/2014/10
/data/access/web1/2014/10/access-20141001.log
[... etc ...]
/data/access/web2/2014/09
/data/access/web2/2014/09/access-20140901.log
[... etc ...]
/data/access/web2/2014/10
/data/access/web2/2014/10/access-20141001.log
[... etc ...]
```

```
set hive.input.dir.recursive=true;
set hive.mapred.supports.subdirectories=true;
set hive.supports.subdirectories=true;
set mapred.input.dir.recursive=true;
```

82. How to enable Fetch Task instead of MapReduce Job for simple query in Hive

- hive.fetch.task.conversion
- hive.fetch.task.aggr
- hive.fetch.task.conversion.threshold

hive.fetch.task.conversion

- This parameter controls which kind of simple query can be converted to a single fetch task.
- Possible values
  - None – to disable
  - Minimal - SELECT *, FILTER on partition columns (WHERE and HAVING clauses), LIMIT only.
  - More - Value "more" means SELECT, FILTER, LIMIT only (including TABLESAMPLE, virtual columns)."more" can take any kind of expressions in the SELECT clause, including UDFs.

Cont. hive.fetch.task.conversion

```
hive> set hive.fetch.task.conversion;
hive.fetch.task.conversion=minimal
hive> explain select * from passwords limit 1;
OK
STAGE DEPENDENCIES:
  Stage-0 is a root stage

STAGE PLANS:
  Stage: Stage-0
    Fetch Operator
      limit: 1
      Processor Tree:
        TableScan
```

```
hive> explain select col0 from passwords limit 1;
OK
STAGE DEPENDENCIES:
  Stage-1 is a root stage
  Stage-0 is a root stage

STAGE PLANS:
  Stage: Stage-1
    Map Reduce
      Map Operator Tree:
          TableScan
```

```
hive> set hive.fetch.task.conversion=more;
hive> explain select col0 from passwords ;
OK
STAGE DEPENDENCIES:
  Stage-0 is a root stage

STAGE PLANS:
  Stage: Stage-0
    Fetch Operator
      limit: -1
      Processor Tree:
        TableScan
          alias: passwords
          Statistics: Num rows: 9963904 Data size:
```

And it also allows "select * " with "filter" to use fetch task, eg:
select * from passwords where col0='root';
So we have to use "more" very carefully and only use it after fully testing. Because fetch can not utilize the parallelism of MapReduce framework.

Cont. hive.fetch.task.aggr

- Aggregation queries with no group-by clause (for example, select count(*) from src) execute final aggregations in a single reduce task. If this parameter is set to true, Hive delegates the final aggregation stage to a fetch task, possibly decreasing the query time.
- It can help when the output of the mappers are very small, by reducing the time for shuffle and launching reduce task.

Cont. hive.fetch.task.aggr

```
hive> set hive.fetch.task.aggr;
hive.fetch.task.aggr=false
hive> explain select count(*) from passwords;
OK
STAGE DEPENDENCIES:
  Stage-1 is a root stage
  Stage-0 is a root stage

STAGE PLANS:
  Stage: Stage-1
    Map Reduce
      Map Operator Tree:
          TableScan
            alias: passwords
            Statistics: Num rows: 9963904 Data size: 477218560 Basi
            Select Operator
              Statistics: Num rows: 9963904 Data size: 477218560 Ba
              Group By Operator
                aggregations: count()
                mode: hash
                outputColumnNames: _col0
                Statistics: Num rows: 1 Data size: 8 Basic stats: C
                Reduce Output Operator
                  sort order:
                  Statistics: Num rows: 1 Data size: 8 Basic stats:
                  value expressions: _col0 (type: bigint)
      Reduce Operator Tree:
        Group By Operator
          aggregations: count(VALUE._col0)
          mode: mergepartial
```

```
hive> set hive.fetch.task.aggr=true;
hive> explain select count(*) from passwords;
OK
STAGE DEPENDENCIES:
  Stage-1 is a root stage
  Stage-0 is a root stage

STAGE PLANS:
  Stage: Stage-1
    Map Reduce
      Map Operator Tree:
          TableScan
            alias: passwords
            Statistics: Num rows: 9963904 Data size: 47721
            Select Operator
              Statistics: Num rows: 9963904 Data size: 477
              Group By Operator
                aggregations: count()
                mode: hash
                outputColumnNames: _col0
                Statistics: Num rows: 1 Data size: 8 Basic
                File Output Operator
                  compressed: false
                  table:
                      input format: org.apache.hadoop.mapr
                      output format: org.apache.hadoop.hiv
                      serde: org.apache.hadoop.hive.serde2

  Stage: Stage-0
    Fetch Operator
```

Cont. hive.fetch.task.aggr

Performance differences

```
hive> set hive.fetch.task.aggr=false;
hive> select count(*) from passwords;
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in by
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201501081639_0044, Tracking URL = htt
Kill Command = /opt/mapr/hadoop/hadoop-0.20.2/bin/../bir
Hadoop job information for Stage-1: number of mappers: 1
2015-01-15 12:10:15,706 Stage-1 map = 0%,  reduce = 0%
2015-01-15 12:10:26,764 Stage-1 map = 47%,  reduce = 0%,
2015-01-15 12:10:27,776 Stage-1 map = 100%,  reduce = 0%
2015-01-15 12:10:32,808 Stage-1 map = 100%,  reduce = 10
MapReduce Total cumulative CPU time: 9 seconds 270 msec
Ended Job = job_201501081639_0044
MapReduce Jobs Launched:
Job 0: Map: 1  Reduce: 1   Cumulative CPU: 9.27 sec   MA
Total MapReduce CPU Time Spent: 9 seconds 270 msec
OK
9963904
Time taken: 19.505 seconds, Fetched: 1 row(s)
```

```
hive> set hive.fetch.task.aggr=true;
hive> select count(*) from passwords;
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no
Starting Job = job_201501081639_0045, Tracking URL
Kill Command = /opt/mapr/hadoop/hadoop-0.20.2/bin/.
Hadoop job information for Stage-1: number of mappe
2015-01-15 12:10:47,231 Stage-1 map = 0%,  reduce =
2015-01-15 12:10:58,301 Stage-1 map = 47%,  reduce
2015-01-15 12:10:59,308 Stage-1 map = 100%,  reduce
MapReduce Total cumulative CPU time: 7 seconds 260
Ended Job = job_201501081639_0045
MapReduce Jobs Launched:
Job 0: Map: 1   Cumulative CPU: 7.26 sec   MAPRFS R
Total MapReduce CPU Time Spent: 7 seconds 260 msec
OK
9963904
Time taken: 13.817 seconds, Fetched: 1 row(s)
```

Note, if the query has "group-by", it can not use this feature. For example:
select count(*) from passwords group by col0;

hive.fetch.task.conversion.threshold

- This parameter controls input threshold (in bytes) for applying hive.fetch.task.conversion.
- Default threshold is 1 GB.
- This parameter calculates or estimates based on the table size, not the result set size

Cont. hive.fetch.task.conversion.threshold

```
[root@n3a passwords]# pwd
/mapr/my.cluster.com/user/hive/warehouse/passwords
[root@n3a passwords]# du -sh  .
465M .
```

If hive.fetch.task.conversion.threshold is less than the table size, it will use MapReduce Job, else Fetch Task.

```
hive> set hive.fetch.task.conversion.threshold=100000000;
hive> select * from passwords limit 1;
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_201501081639_0046, Tracking URL = http://n1a.myclu
Kill Command = /opt/mapr/hadoop/hadoop-0.20.2/bin/../bin/hadoop job
Hadoop job information for Stage-1: number of mappers: 1; number of r
2015-01-15 12:19:06,474 Stage-1 map = 0%,  reduce = 0%
2015-01-15 12:19:11,496 Stage-1 map = 100%,  reduce = 100%, Cumulativ
MapReduce Total cumulative CPU time: 850 msec
Ended Job = job_201501081639_0046
MapReduce Jobs Launched:
Job 0: Map: 1   Cumulative CPU: 0.85 sec   MAPRFS Read: 0 MAPRFS Writ
Total MapReduce CPU Time Spent: 850 msec
OK
root x 0 0 root /root /bin/bash
Time taken: 6.698 seconds, Fetched: 1 row(s)
```

95 MB

```
hive> set hive.fetch.task.conversion.threshold=600000000;
hive> select * from passwords limit 1;
OK
root x 0 0 root /root /bin/bash
Time taken: 0.325 seconds, Fetched: 1 row(s)
```

572 MB

83. Why do we use RDBMS for Hive Metastore?

- It has to be quick.
- HDFS – WORM.
- HDFS – not for small files and upsert.

84. --incremental lastmodified not working with avro

- Sqoop currently do not support merge of avro files (which is required by the incremental import in lastmodified mode). There is already JIRA SQOOP-1094 to add such support.

85. Scenario: Whether this works **insert overwrite table payments(customernumber,contactfirstname) select customernumber,contactfirstname from customer;**

- We can't use overwrite with column fields in target table as overwrite means to entire table, not column specific.

86. How to export HIVE table to RDBMS?

- SQOOP Export
  - With –export-dir
    - sqoop export --connect jdbc:mysql://localhost/custdb --username root –P --table payments --export-dir /user/hadoopuser/payments
  - With Hcatalog
    - sqoop export
      --connect jdbc:mysql://localhost/retail_db
      --username root -P
      --table mysql_test
      --hcatalog-table test

87. Order By, Sort By, Distribute By and Cluster By.

Order by:

Final output will be sorted using single reducer.

If hive.mapred.mode=strict, then limit clause is required followed by Order By, as MNR uses only one reducer even for huge dataset, which may slow down the process.

Sort By:

Hive uses the columns in SORT BY to sort the rows before feeding the rows to a reducer.

Difference between Sort By and Order By

Hive supports SORT BY which sorts the data per reducer. The difference between "order by" and "sort by" is that the former guarantees total order in the output while the latter only guarantees ordering of the rows within a reducer. If there are more than one reducer, "sort by" may give partially ordered final results.

Distribute By:

Hive uses the columns in Distribute By to distribute the rows among reducers. All rows with the same Distribute By columns will go to the same reducer. However, Distribute By does not guarantee clustering or sorting properties on the distributed keys. To further sort the data, we have to use "Sort By" along with "Distribute By".

If distribute and sort fields are different, we can use this.

For example, we are *Distributing By x* on the following 5 rows to 2 reducer:

```
x1
x2
x4
x3
x1
```

Reducer 1 got

```
x1
x2
x1
```

Reducer 2 got

```
x4
x3
```

Note that all rows with the same key x1 is guaranteed to be distributed to the same reducer (reducer 1 in this case), but they are not guaranteed to be clustered in adjacent positions.

Cluster By:

It is a short-cut for both Distribute By and Sort By. If distribute and sort fields are same, we can use this.

In contrast, if we use *Cluster By x*, the two reducers will further sort rows on x:

Reducer 1 got

```
x1
x1
x2
```

Reducer 2 got

```
x3
x4
```

88.How do you determine number of buckets?

- Find Table Size, if it is new table, check the size of SOR table. Say, it is 2560 MB
- HDFS block size = 128 MB
- Bucket calculation
  - 2560/128 = 20 (this can also be chosen for bucket count, but bucket will be filled soon and we need to redefine the table with new bucket count)

    Bucket size will be power of 2.
  - 2^n >= 20, n = log(20,2) = 4.322 = ~ 5
  - Now, 2^5 = 32.
    - 32 is the bucket count for this size of table.
- Whenever the table size changes, bucket count will change. For most of the minor increments in table size, existing bucket count will accommodate that. But when we calculate bucket count, if it comes as 2^6 as per this example, then we need to create a new table with new bucket size and move the data from old table to new table (insert select) and rename the table.

89.If you run a select * query in Hive, Why does it not run MapReduce?

- It is based on **hive.fetch.task.conversion** property value. Possible values are none, minimal and more

90.Can I have multi column sub query in hive?

- How to convert this for Hive: Select * from table1 A where (A.id,A.name,A.Roll_no) in (Select Id,name,roll_no from table2)
- Answer:
  - Select * from table1 A where concat(A.id,A.name,A.Roll_no) in(Select concat(Id,name,roll_no) from table2)

  OR

  - select a.* from table1 a inner join table2 b on (a.id=b.id and a.name=b.name and a.rollno=b.rollno)

91.How to fix vertex error in Hive?

- Vertex error is due to container memory issue.
- set hive.execution.engine=mr instead of TEZ.

92.How to read fixed width data in Hive?

- Using RegexSerde

```
CREATE EXTERNAL TABLE customers (userid STRING, name STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
WITH SERDEPROPERTIES ("input.regex" = "(.{10})(.{10})" )
LOCATION '/user/hduser/cscdata';
```

93.How to Change table from external to managed and vice versa?

- alter table customers SET TBLPROPERTIES('EXTERNAL'='FALSE');

94.How to handle hive variables which will change based on environment?

- Use hiveVar or hiveConf
  - hive -hiveconf CURRENT_DATE='2020-01-10' -f /home/hduser/xyz.hql
- Use file name
  - hive -f /home/hduser/xyz.hql -i /home/hduser/paramfile.txt

95.Data will be erased when you drop Managed & External table?

- Managed Table – Yes
- External Table - No

96.How to manage partitions automatically?

- ALTER TABLE customer SET TBLPROPERTIES ('partition.retention.period'='7d');

97.External table with partition is dropped by mistake and how do you reload?

- Create the table back on top of the particular hdfs folder
- Msck repair on table.

98.If any external tools loaded data into hive partition location. How do you ensure those new partitions available for querying?

- alter table <db_name>.<table_name> add partition(`date`='<date_value>') location '<hdfs_location_of the specific partition>';
- Msck repair table <db_name>.<table_name>
- Automatic discovery
  - set metastore.partition.management.task.frequency=3600; (in seconds)
  - ALTER TABLE csc SET TBLPROPERTIES ('discover.partitions' = 'true');

99.How to handle small file issue in Hive?

- HAR to reduces files in partition directory
  - set hive.archive.enabled=true;
  - set hive.archive.har.parentdir.settable=true;
  - set har.partfile.size=1099511627776;
  - Once the configuration values are set, a partition can be archived with the command:
  - ALTER TABLE table_name ARCHIVE PARTITION (partition_col = partition_col_value, partition_col = partiton_col_value, ...)

Setting split size:

set hive.merge.mapfiles=true;

set hive.merge.mapredfiles=true;

set hive.merge.smallfiles.avgsize=104857600;

set hive.merge.size.per.task=209715200;

set mapred.max.split.size=68157440;

set mapred.min.split.size=68157440;

set hive.exec.mappers.bytes.per.mapper=268435456; (how much size of file each mapper should produce, more the number lesser will be the number of files.)

set hive.exec.reducers.bytes.per.reducer=268435456;

After setting the above configuration, run the below.

INSERT OVERWRITE TABLE table1 SELECT * FROM table1 (small files loaded by nifi or some other streaming sources);

100.If we change the partition location of a hive table using ALTER TABLE with new location then the data for that partition will also be moved?

- No. It has to be moved manually into new location

101.What is the default maximum dynamic partition that can be created? How can you change it?

- By default the number of maximum partition that can be created is 100.
- We can change it by issuing the following command:
  - SET hive.exec.max.dynamic.partitions.pernode = <value>
- We can set the total number of dynamic partitions that can be created by one statement by using below
  - SET hive.exec.max.dynamic.partitions = <value>

102.I dropped a hive managed table, can I get the hdfs data recovered back?

- Yes, from the trash folder.
- If you don't want to have the data to be in Trash folder use PURGE option.
  - DROP TABLE [IF EXISTS] table_name [PURGE];

103.Does “minus”/”except” supported in hive?

- No. Use Left Join

```
SELECT x
FROM abc
LEFT JOIN bcd
ON abc.x = bcd.x
WHERE bcd.x IS NULL
```

104.Subqueries are supported in Hive?

- Hive supports subqueries in the FROM clause (through Hive 0.12).

```
SELECT col
FROM (
  SELECT a+b AS col
  FROM t1
) t2
```

- Subqueries in the WHERE Clause. Has to be uncorrelated query.

```
SELECT *
FROM A
WHERE A.a IN (SELECT foo FROM B)
```

- EXISTS and NOT EXISTS subqueries:

```
SELECT A
FROM T1
WHERE EXISTS (SELECT B FROM T2 WHERE T1.X = T2.Y)
There are a few limitations:

These subqueries are only supported on the right-hand side of an expression.
IN/NOT IN subqueries may only select a single column.
EXISTS/NOT EXISTS must have one or more correlated predicates.
References to the parent query are only supported in the WHERE clause of the subquery.
```

105.Is it possible to load the data to hive without using HDFS in between?

- Use LOAD DATA LOCAL INPATH

106.Does hive can manage data only using hdfs as a storage layer?

- No, hdfs is default, u can use any other storage layers like cloud fs – s3, nosqls, db etc. This can be done using respective storage handler.

107.How to change NULL value to something else?

- Use COALESCE or NVL. ISNULL is not supported in Hive.
  - COALESCE(column, CAST(0 AS BIGINT))
  - nvl(T value, T default_value)

108.Need to load 1 year data in a table, how to distribute equally? Assume each day has 128 MB of data.

- **set mapred.map.tasks=365;**
- **set mapred.reduce.tasks=365;**
- insert into table abc partition(**data_dt**) select * from src distribute by **data_dt**;

This distribute by distributes data equally to 365 mapper/reducers as we set. Data load will be distributed uniformally otherwise if not we set this, lets say, hive creates 50 mappers, here few mapper will load more data and few will load less data.

109. How do you define small tables in Hive joins?

- Small table size is defined by
  - set hive.mapjoin.smalltable.filesize = 30000000;
- During joins all left tables (except last one) should be small enough to keep those in memory, and last table will be streamed while join, to perform map-only join.
- If we are not sure how to place the order of tables in query, we can provide STREAMTABLE hint.
  - SELECT /*+ STREAMTABLE(table1) */ table1.val, table2.val
    FROM table1 JOIN table2 ON (table1.key = table2.key1)

110. Difference between ORC & PARQUET?

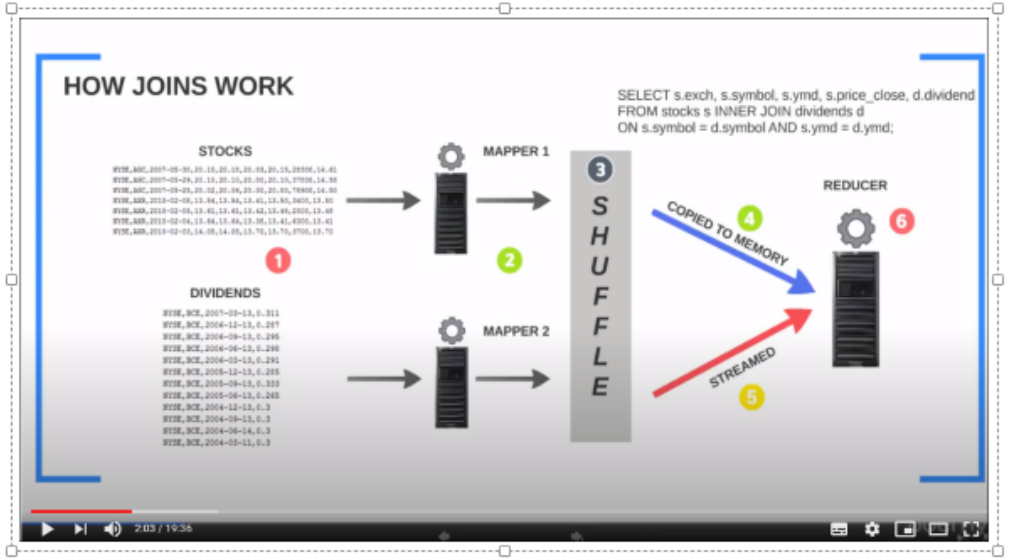
- ORC
  - Vectorization supports only ORC not other file format in Hive
  - Acid table properties supports only in ORC upto hive v 3x
  - ORC doesn’t support complex types
- PARQUET
  - Vectorization supports for Parquet for Spark
  - Parquet supports complex types

111. When do you use what type of engines?

- MR - Batch ETL/ELT load without strict SLA or ETA.
- SPARK - In-memory, to meet faster SLA/ETA and for complex queries.
- TEZ/IMPALA/PRESTO/TRINO – for in-memory, ad-hoc/real-time simple queries.

112. Joins in HIVE

Reduce Side Join



Joins without Optimization

Reduce Side Join

1. Each mapper reads individual table
2. Each mapper emits a key value pair. **Key will be join column**, our case Symbol column and emits entire record as value.
3. Shuffle phase sorts the record by key and each key will be assigned to reducers in case of multiple reducers and records for each key will be sent to appropriate reducer.
4. All table except last table will be loaded into memory in reducer side.
5. Records from the last table in joins will be streamed to the reducer.
6. Reducer will cross the results coming from stocks and dividends and apply the join conditions and derive the result set.

From the above process we came to know that last table is streamed to reducer. Here the big table is STOCKS table which is a good candidate for streaming otherwise keeping big table in memory Will hog the memory space. So, we have a chance of optimization here.

So, change the table position in query. Now the bigger table - STOCK will be streamed.

```
### REGULAR INNER JOIN ###

hive> SELECT s.exch, s.symbol, s.ymd, s.price_close, d.dividend
FROM stocks s INNER JOIN dividends d
ON s.symbol = d.symbol AND s.ymd = d.ymd;

### INNER JOIN WITH BETTER TABLE PLACEMENT FOR OPTIMIZATION ###

--Inner join positioning stocks table to the right
hive> SELECT s.exch, s.symbol, s.ymd, s.price_close, d.dividend
FROM dividends d INNER JOIN stocks s
ON s.symbol = d.symbol AND s.ymd = d.ymd;
```

Optimized – Streamtable hint

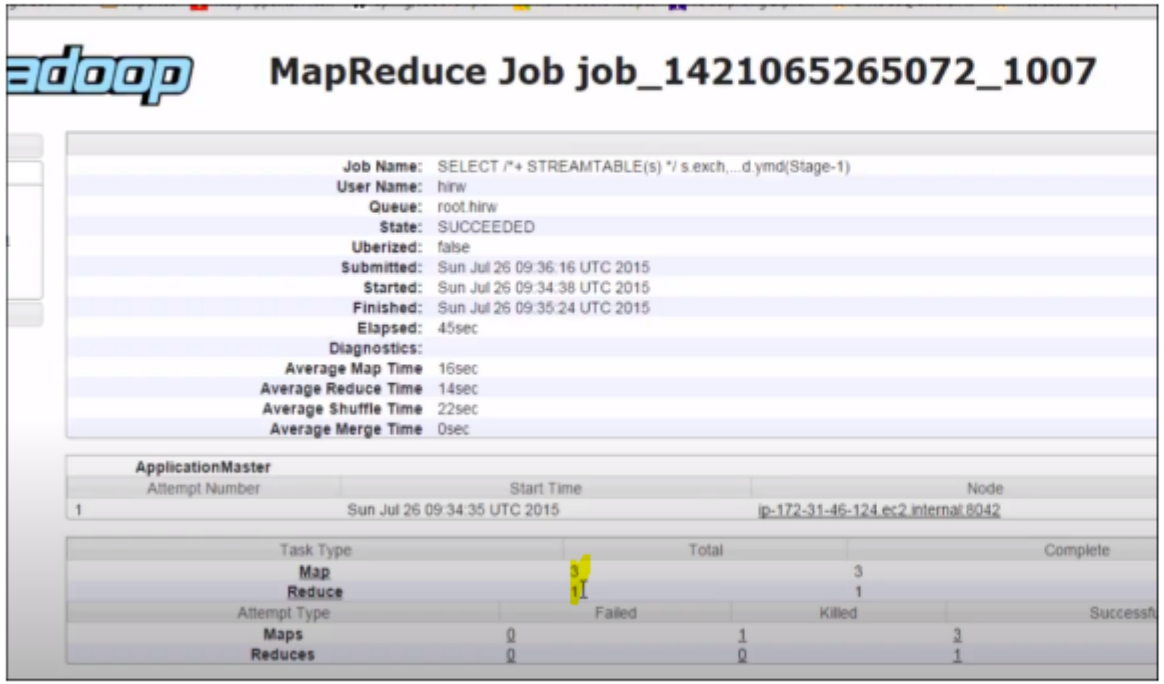
Reduce Side Join

The better and cleaner way to achieve the same without changing the position of the table in query is by using STREAMTABLE hint. It specifies the table that should be streamed to the reducer.

STREAMTABLE

```
### REGULAR INNER JOIN ###

hive> SELECT s.exch, s.symbol, s.ymd, s.price_close, d.dividend
FROM stocks s INNER JOIN dividends d
ON s.symbol = d.symbol AND s.ymd = d.ymd;

### INNER JOIN WITH BETTER TABLE PLACEMENT FOR OPTIMIZATION ###

--Inner join positioning stocks table to the right
hive> SELECT s.exch, s.symbol, s.ymd, s.price_close, d.dividend
FROM dividends d INNER JOIN stocks s
ON s.symbol = d.symbol AND s.ymd = d.ymd;

### STREAM A TABLE TO REDUCER ###

hive> SELECT /*+ STREAMTABLE(s) */ s.exch, s.symbol, s.ymd, s.price_close, d.dividend
FROM stocks s INNER JOIN dividends d
ON s.symbol = d.symbol AND s.ymd = d.ymd;
```

When we execute the query, and checking the MR job. It uses 3 mappers and 1 reducer. The expensive operation in any join process is shuffle and reduce phase - these phases does sorting and merging records. We can optimize the join if we can avoid shuffle and reduce phase and execute the join entirely in Map phase.



MapReduce Job job_1421065265072_1007

Job Name:	SELECT /*+ STREAMTABLE(s) */ s.exch,...d.ymd(Stage-1)
User Name:	hirw
Queue:	root.hirw
State:	SUCCEEDED
Uberized:	false
Submitted:	Sun Jul 26 09:36:16 UTC 2015
Started:	Sun Jul 26 09:34:38 UTC 2015
Finished:	Sun Jul 26 09:35:24 UTC 2015
Elapsed:	45sec
Diagnostics:	
Average Map Time	16sec
Average Reduce Time	14sec
Average Shuffle Time	22sec
Average Merge Time	0sec

ApplicationMaster

Attempt Number	Start Time	Node
1	Sun Jul 26 09:34:35 UTC 2015	ip-172-31-46-124.ec2.internal:8042

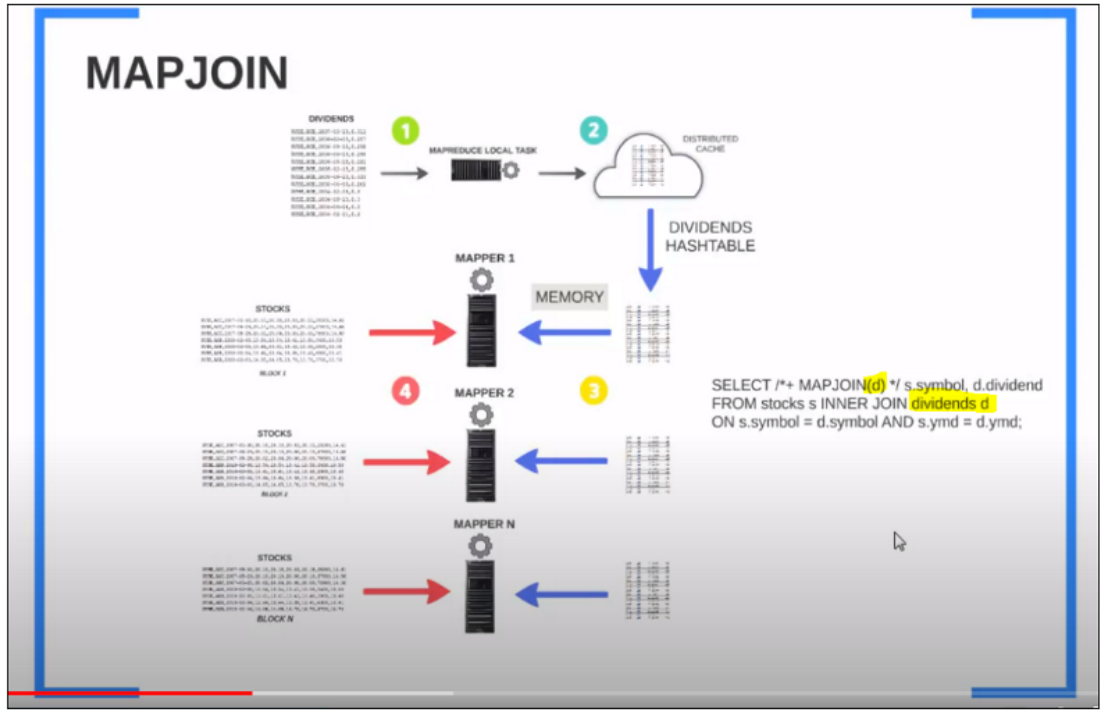
Task Type	Total	Complete
Map	3	3
Reduce	1	1

Attempt Type	Failed	Killed	Successful
Maps	0	1	3
Reduces	0	0	1

Reduce Side Join

To enable Map Join, hive has MapJoin hint. MAPJOIN allow us to execute entire join in Map phase itself.

Map Side Join



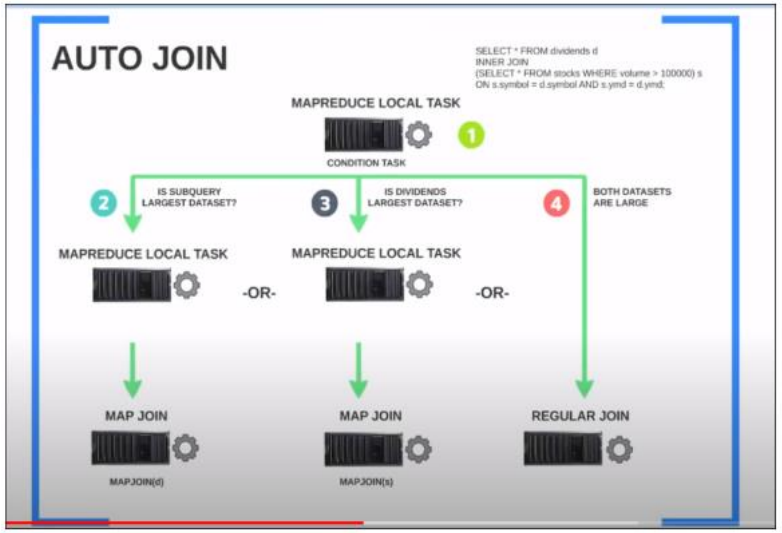
Optimized – MapJoin Hint

Map Side Join

Dividend is the small dataset, we specified in MAPJOIN hint.

1. New MR Local Task is launched before original join MR task.
2. This new task reads and serializes the dividend dataset into a hash table. HT is uploaded into hadoop's distributed cache - it is a facility provided by a MR framework to cache files. Once it Is added into distributed cache, it will be copied to local file system where mapper is going to run actual task on data. Thus divided table will be available to each mapper locally.
3. Here mapper does join with cache data.
   Give small table in MAPJOIN hint. So that it can be passed to each node's cache.

If we have given large table in MAPJOIN hint, it will lead to serious problem. HIVE by default goes to MAP join whenever possible.



To enable auto map join, set hive.auto.convert.join = true; Now u no need to set Map join hint any more.

To do map only join, we need to know the small table to keep it in cache. If we set auto convert, hive needs to find the small table in run time and do the map side join. If your query Has subquery as mentioned in the above screen shot, hive can't determine the size of the result set of sub query. Size of the small table is configued in **hive.mapjoin.smalltable.filesize** , by default it is 30 MB. If dataset is less than 30 MB, it is considered small.

1. Conditional task will be created. Based on the result, step 2 or 3 or 4 will be executed.

```
2015-07-26 09:52:11     Starting to launch local task to process map join;      maximum mem
2015-07-26 09:52:12     Dump the side-table into file: file:/tmp/hirw/hive_2015-07-26_09-52
le01--.hashtable
2015-07-26 09:52:12     Upload 1 File to: file:/tmp/hirw/hive_2015-07-26_09-52-08_056_4099
ble
2015-07-26 09:52:12     End of local task; Time Taken: 1.478 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1421065265072_1009, Tracking URL = http://ec2-54-164-162-158.compute-1.
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1421065265072_1009
Hadoop job information for Stage-3: number of mappers: 2; number of reducers: 0
2015-07-26 09:52:18,952 Stage-3 map = 0%,  reduce = 0%
2015-07-26 09:52:27,212 Stage-3 map = 50%,  reduce = 0%, Cumulative CPU 4.43 sec
```

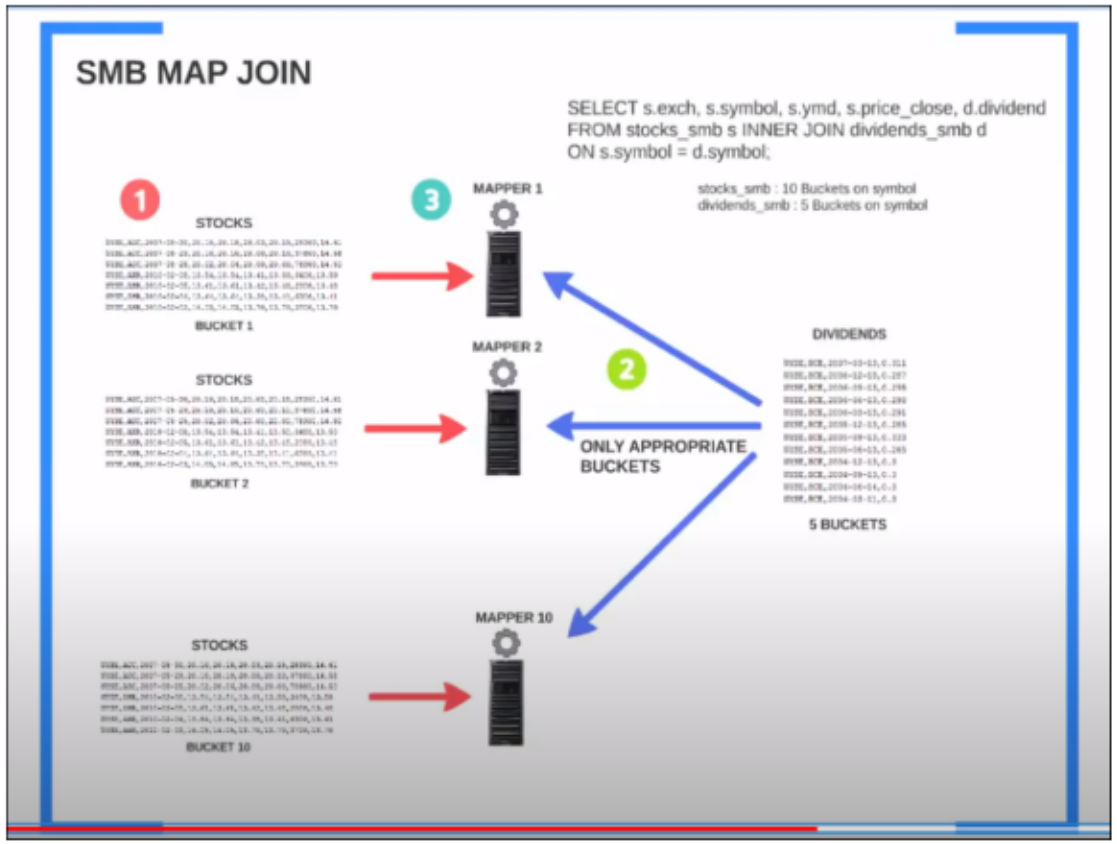
So far we have seen with regular tables. What happens with bucketed tables. When the table is sorted and bucket, hive can perform faster map side joins.

Optimized – MapJoin Hint

Map Side Join

- Enable auto convert join. So that Hive will use Map side join wherever possible.
  - Set hive.auto.convert.join = true
  - Set hive.auto.convert.join.noconditionaltask = true
  - Set hive.mapjoin.smalltable.filesize = 300000
  - Set hive.auto.convert.join.noconditionaltask.size = 600000

SORT MERGE BUCKET JOIN



Criteria:

1. All join table must be bucketized.
2. Both the table must be bucketed on same column and sorted on same column
3. No of buckets of big table must be divisible by buckets of small table.
4. Bucket column and join column should be same
5. Set all the below properties

```
hive> SET hive.auto.convert.sortmerge.join=true;
hive> SET hive.optimize.bucketmapjoin = true;
hive> SET hive.optimize.bucketmapjoin.sortedmerge = true;
hive> SET hive.auto.convert.sortmerge.join.noconditionaltask=true;
```

No of files created in HDFS after inserting records = no of buckets
No of reducers used while inserting records into bucketed table = no of buckets

For bucketing to work, we need to set the below.
Hive.enforce.bucketing = true

Default is Map Reduce side join.
Small table in cache - map side join
Join on same column In bucketed table - bucket join - same bucket will be loaded to memory
If records sorted and bucketed - SMB join - only the join key data alone loaded in to memory at a time not whole bucket. It uses merge-sort.

1. MR job spawn mappers based on big table. Here big table (STOCK) is 10 bucketed, so 10 mappers will be launched for the join operation to perform.
2. Only matching bucket of small table will be replicated on each mapper.
3. Perform join operation on each mapper.
   Since the records are already sorted and only matching buckets are considered for join operation. This join is entirely performed on Map side, enabling much faster join.

Explanation on Parameters

`Map join` is used when one of the join tables is small enough to fit in the memory, so it is very fast. here's the explanation of all parameters:

`hive.auto.convert.join`

When this parameter set to `true`, Hive will automatically check if the smaller table file size is bigger than the value specified by `hive.mapjoin.smalltable.filesize`, if it's larger than this value then query execute through common join. Once auto convert join is enabled, there is no need to provide the map join hints in the query.

`hive.auto.convert.join.noconditionaltask`

When three or more tables are involved in join, and

`hive.auto.convert.join = true` - Hive generates three or more map-side joins with an assumption that all tables are of smaller size.

`hive.auto.convert.join.noconditionaltask = true`, hive will combine three or more map-side joins into a single map-side join if size of n-1 table is less than 10 MB. Here size is defined by `hive.auto.convert.join.noconditionaltask.size`.

`hive.mapjoin.smalltable.filesize`

This setting basically the way to tell optimizer the definition of small table in your system. This value defines what is small table for you and then when query executes based on this value it determines if join is eligible to convert into `map join`.

`hive.auto.convert.join.noconditionaltask.size`

The size configuration enables the user to control what size table can fit in memory. This value represents the sum of the sizes of tables that can be converted to hashmaps that fit in memory.

Thank You

THE BRIGHT SIDE

www.youtube.com/@ashok.thebrightside

ashok.thebrightside@gmail.com