1 what is the block size in cloudera 5.8

hadoop 1.x 64mb

hadoop 2.x 128mb

2 what is the crach directorty in hive

the scratch directory (which is used by Hive to store temporary output and plans) to /tmp/mydir for all subsequent statements:

set hive.exec.scratchdir=/tmp/mydir;

3 how to set map side join in hive when we join queries

Small file = 25 mb by default is store in in memory and stream big file from hard disk and it does join operation ,

1. By specifying the keyword, /\*+ MAPJOIN(b) \*/ in the join statement.

2. By setting the following property to true.

|  |  |
| --- | --- |
| 1 | hive.auto.convert.join=true |

For performing Map-side joins, there should be two files, one is of larger size and the other is of smaller size. You can set the small file size by using the following property:

|  |  |
| --- | --- |
| 1 | hive.mapjoin.smalltable.filesize=(default it will be 25MB) |

Now, let us perform Map-side joins and join the two datasets based on their IDs.

|  |  |
| --- | --- |
| 1 | SELECT /\*+ MAPJOIN(dataset2) \*/ dataset1.first\_name, dataset1.eid,dataset2.eid FROM dataset1 JOIN dataset2 ON dataset1.first\_name = dataset2.first\_name; |

4 what is the spit-by and how it work

--split-by id will split your data **uniformly** on the basis of number of mappers *(default 4)*.

Now boundary query by default is something like this.

--boundary-query "SELECT min(id), max(id) from some\_table"

But if you know id **starts from** val1 and **ends with** val2. Then there is no point to calculate min() and max() operations. This will make sqoop command execution faster.

You can specify any arbitrary query returning val1 and val2.

**Edit:**

Right now (1.4.7) there is no way in sqoop to specify uneven partitions for splitting.

For example, you have data like:

1,2,3,51,52,191,192,193,194,195,196,197,198,199,200

If you defined 4 mappers in the command. It will check min and max which is 1 and 200 in our case.

Then it will split it into 4 parts:

1-50

51-100

101-150

151-200

Yes, in this 3rd mapper(101-150) will get **nothing** from the RDBMS table.

But there is no way to define custom partition like :

1-10

51-60

190-200

5 how you handle incremental append

--Incremental append

--check-column id

--last value 0

6 how you handle null values in rdbms while importing data

The --null-string and --null-non-string arguments are optional.\ If not specified, then the string "null" will be used.

1. If (null string) property is not included during sqoop import, then NULLs are stored as [**blank** for integer columns] and [**blank** for string columns] in HDFS. 2.If the HIVE table on top of HDFS is queried, we would see [**NULL** for integer column] and [**blank** for String columns]
2. If the (--null-string '\N') property is included during sqoop import, then NULLs are stored as [**'\N'** for both integer and string columns].
3. If the HIVE table on top of HDFS is queried, we would see [**NULL** for both integer and string columns not **'\N'**]

7 how add record to existing tables using sqoop

yes using append

8 how you manage the null values using sqoop while importing

9 can we change block size in cloudera cluster

yes we can chane block size in hdfs

type1.

In conf/ folder we can change the value of dfs.block.size in configuration file hdfs-site.xml. In hadoop version 1.0 default size is 64MB and in version 2.0 default size is 128MB.

<property>

<name>dfs.block.size<name>

<value>134217728<value>

<description>Block size<description>

<property>

type2.

Configuration conf = new Configuration() ;

conf.set( "dfs.block.size", 128\*1024\*1024) ;

type3.

hadoop fs -D dfs.block.size=134217728 -put local\_name remote\_location

Bloch change questions

Will this change require restart of the cluster or it will be taken up automatically and all new files will have the default block size of 128MB

A restart of the cluster will be required for this property change to take effect.

What will happen to the existing files which have block size of 64M? Will the change in the configuration apply to existing files automatically?

Existing blocks will not change their block size.

If not automatically done, then how to manually do this block change?

To change the existing files you can use distcp. It will copy over the files with the new block size. However, you will have to manually delete the old files with the older block size. Here's a command that you can use

No, it will not. It will keep the old block size on the old files. In order for it to take the new block change, you need to rewrite the data. You can either do a hadoop fs -cp or a distcp on your data. The new copy will have the new block size and you can delete your old data.

The default size of [**Distributed Cache**](http://data-flair.training/blogs/hadoop-distributed-cache/) files is 10GB.However we can modify the limit by updating parameter local.cache.size in hdfs-sit.xml

Enable Compression in Hive

hive.exec.compress.output TRUE

hive.exec.compress.intermediate TRUE

hive.auto.convert.join

hive.auto.convert.join.noconditionaltask

hive.optimize.bucketmapjoin

avoid order by and try to use sort by

set hive.vectorized.execution.enabled = true;

set hive.vectorized.execution.reduce.enabled = true;

set hive.vectorized.execution.reduce.groupby.enabled = true;

hive.compute.query.using.stats

hive.stats.fetch.partition.stats

hive.stats.fetch.column.stats

hive.stats.autogather

ANALYZE TABLE employee COMPUTE STATISTICS FOR COLUMNS;

ANALYZE TABLE employee COMPUTE STATISTICS FOR COLUMNS id, dept;