



# Few more optimizations

...

## Hive Vectorization & changing hive engine

# IMPORTANT

## **Copyright Infringement and Illegal Content Sharing Notice**

All course content designs, video, audio, text, graphics, logos, images are Copyright© and are protected by India and international copyright laws. All rights reserved.

Permission to download the contents (wherever applicable) for the sole purpose of individual reading and preparing yourself to crack the interview only. Any other use of study materials – including reproduction, modification, distribution, republishing, transmission, display – without the prior written permission of Author is strictly prohibited.

**Trendytech Insights** legal team, along with thousands of our students, actively searches the Internet for copyright infringements. Violators subject to prosecution.



# What is vectorization in Hive

**Vectorized query execution is a Hive feature that greatly reduces the CPU usage for typical query operations.**

**A standard query execution system processes one row at a time. Vectorized query execution streamlines operations by processing a block of 1024 rows at a time. This makes it very efficient.**



# Enabling vectorized execution

**Note:** To use vectorized query execution, you must store your data in ORC format.

**We also need to set the following variable:**

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

**Vectorized execution is off by default.**

```
hive> set hive.vectorized.execution.enabled;  
hive.vectorized.execution.enabled=false  
hive> █
```



# Vectorized Execution

Let us create a table and insert the record.

```
create table vectorizedtable(state string,id int) stored as orc;
```

```
insert into vectorizedtable values('karnataka',1);
```

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

```
explain select count(*) from vectorizedtable;
```

```
hive> create table vectorizedtable(state string,id int) stored as orc;  
OK  
Time taken: 0.114 seconds  
hive> █
```



# Vectorized Execution

```
hive> set hive.vectorized.execution.enabled=true;
hive> set hive.vectorized.execution.enabled;
hive.vectorized.execution.enabled=true
hive> █
```

```
Reduce Output Operator
  sort order:
    Statistics: Num rows: 1 Data size: 8 Basic stats: COMPLETE Column
stats: COMPLETE
  value expressions: _col0 (type: bigint)
Execution mode: vectorized
Reduce Operator Tree:
  Group By Operator
    aggregations: count(VALUE._col0)
    mode: mergepartial
    outputColumnNames: _col0
    Statistics: Num rows: 1 Data size: 8 Basic stats: COMPLETE Column stats:
COMPLETE
  File Output Operator
    compressed: false
    Statistics: Num rows: 1 Data size: 8 Basic stats: COMPLETE Column stats
: COMPLETE
```



# Changing the Hive Engine

**Hive queries can run on three different kinds of execution engines and those are listed below:**

**mr (stands for mapreduce)**

**tez (tez engine)**

**spark (spark engine)**

**Note: By default in cloudera the hive engine is set to mr (mapreduce)**



# Changing the Hive Engine

Let us try to change the engine to spark as shown below

```
0: jdbc:hive2://> set hive.execution.engine;
+-----+
|          set          |
+-----+
| hive.execution.engine=mr |
+-----+
1 row selected (0.063 seconds)
0: jdbc:hive2://> set hive.execution.engine=spark;
No rows affected (0.007 seconds)
0: jdbc:hive2://> set hive.execution.engine;
+-----+
|          set          |
+-----+
| hive.execution.engine=spark |
+-----+
1 row selected (0.004 seconds)
0: jdbc:hive2://> █
```





# Changing the Hive Engine

Let us now try executing the below query:

```
select product_id, sum(amount) from orders group by product_id;
```

```
Status: Running (Hive on Spark job[0])
Job Progress Format
CurrentTime StageId StageAttemptId: SucceededTasksCount(+RunningTasksCount-FailedTa
sksCount)/TotalTasksCount [StageCost]
2020-05-05 04:29:09,890 Stage-0_0: 0/1 Stage-1_0: 0/1
2020-05-05 04:29:10,895 Stage-0_0: 0(+1)/1 Stage-1_0: 0/1
2020-05-05 04:29:13,918 Stage-0_0: 1/1 Finished Stage-1_0: 1/1 Finished
Status: Finished successfully in 7.05 seconds
OK
+-----+-----+
| product_id | _c1 |
+-----+-----+
| phone      | 1200.0 |
| t-shirt    | 66.0   |
| broom      | 30.0   |
| camera     | 5200.0 |
+-----+-----+
4 rows selected (23.3 seconds)
```

**Note: Do not try to evaluate based on time it takes because the resources are less and data is very small.**



**We have learnt a few more hive Optimizations**

**Happy Learning!!!**



**5** Star Google Rated  
Big Data Course

**LEARN FROM THE EXPERT**



**9108179578**

**Call for more details**



# Follow US

**Trainer** Mr. Sumit Mittal

**Phone** 9108179578

**Email** trendytech.sumit@gmail.com

**Website** <https://trendytech.in/courses/big-data-online-training/>

**LinkedIn** <https://www.linkedin.com/in/bigdatabysumit/>

**Twitter** @BigdataBySumit

**Instagram** bigdatabysumit

**Facebook** <https://www.facebook.com/trendytech.in/>

**Youtube** [https://www.youtube.com/channel/UCbTggJVf0NDTfWX-C\\_gUGSg](https://www.youtube.com/channel/UCbTggJVf0NDTfWX-C_gUGSg)