

Hadoop • Hive

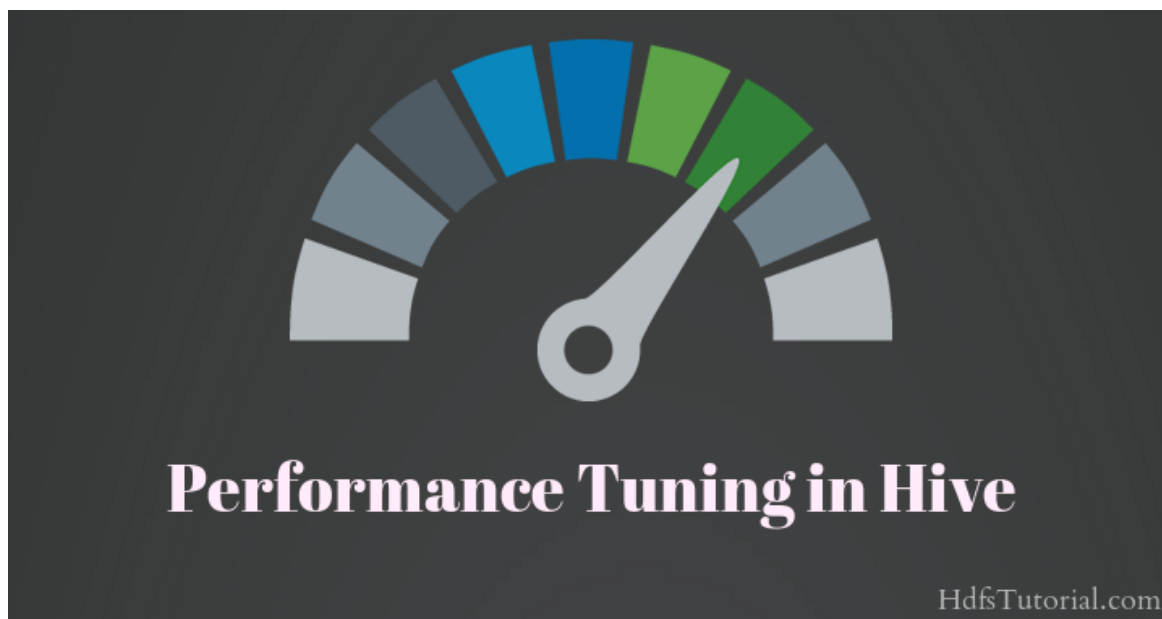
# A Definitive Guide To Hive Performance Tuning- 10 Excellent Tips

by HDFS Tutorial Team • 7 min read



If the Hive code is not written properly, you may face timing in hive query execution. And so hive performance tuning is very important.

When you do Hive query optimization, it helps the query to execute at least by 50%. If your query is not optimized, a simple select statement can take very long to execute.



There are many methods for Hive performance tuning and being a Hadoop developer; you should know these to do well with the queries in a production environment.



### Cloud & DevOps Architect

Get hands-on Cloud & DevOps certification training and have a highly successful career!

 Intellipaat

**Contents** [\[show\]](#)

## Hive Performance Tuning- 10 Best Tips to adopt

You may be knowing some of these hive query optimization techniques like using parallel lines, file formats, optimizing joins, etc. But I will also discuss some advanced hive performance tuning techniques so that you can master the optimization of hive queries.

So let's start with Hive performance tuning techniques!

### 1. Use Tez to Fasten the execution

Apache TEZ is an execution engine used for faster query execution. It fastens the query execution time to around 1x-3x times.

Set hive.execution.engine=tez;

If you are using Cloudera/Hortonworks, then you will find TEZ option in the Hive query editor as well.

## 2. Enable compression in Hive

Compression techniques reduce the amount of data being transferred and so reduces the data transfer between mappers and reducers.

For better result, you need to perform compression at both mapper and reducer side separately. Although gzip is considered as the best compression format but beware that it is not splittable and so should be applied with caution.

Other formats are snappy, lzo, bzip, etc. You can set compression at mapper and reducer side using codes below-



Also, the compressed file should not be more than few hundred MBs else it may impact the jobs.

### 3. Use ORC file format

ORC (optimized record columnar) is great when it comes to hive performance tuning. We can improve the query performance using ORC file format easily. You can check [Hadoop file formats](#) in detail here.

There is no barrier like in which table you can use ORC file and in response, you get faster computation and compressed file size.

It is very easy to create ORC table, and you just need to add STORED AS ORC command as shown below.

Syntax:

```
Create table orctbl (id int, name string, address string) stored as ORC
tblproperties ("orc.compress"= "SNAPPY");
```

```
Insert overwrite table orctbl select * from tbl details;
```

## 4. Optimize your joins

If you are using joins to fetch the results, it's time to revise it. If you have large data in the tables, then it is not advisable to just use normal joins we use in SQL. There are many other joins like Map Join; bucket joins, etc. which can be used to improve Hive query performance.

You can do the following with joins to optimize hive queries-

- **Use Map Join**

Map join is highly beneficial when one table is small so that it can fit into the memory. Hive has a property which can do auto-map join when enabled. Set the below parameter to true to enable auto map join.

Set `hive.auto.convert.join` to true to enable the auto map join. You can either set this from the command line or from the `hive-site.xml` file.

```
<property>
<name>hive.auto.convert.join</name>
<value>true</value>
<description>Whether Hive enables the optimization about converting
common join into mapjoin based on the input file size</description>
</property>
```

- **Use Skew Join**

```
<property>
```

```
<name>hive.optimize.skewjoin</name>
```

```
<value>true</value>
```

```
<description>
```

Whether to enable skew join optimization. The algorithm is as follows: At runtime, detect the keys with a large skew. Instead of processing those keys, store them temporarily in an HDFS directory. In a follow-up map-reduce job, process those skewed keys. The same key need not be skewed for all the tables, and so, the follow-up map-reduce job (for the skewed keys) would be much faster, since it would be a map-join.

```
</description> </property>
```

- **Bucketed Map Join**

If tables are bucketed by a particular column, you can use bucketed map join to improve the hive query performance.

You can set the below two property to enable the bucketed map join in Hive.

```
<property>
```

```
<name>hive.optimize.bucketmapjoin</name>
```

```
<value>true</value>
```

```
<description>Whether to try bucket mapjoin</description>
```

```
</property>
```

```
<property>
```

```
<name>hive.optimize.bucketmapjoin.sortedmerge</name>
```

```
<value>true</value>
```

## 5. Use partition

Partition is a useful concept in Hive. It is used to divide the large table based on certain column so that the whole data can be divided into small chunks. It allows you to store the data under sub-directory inside a table.

Selecting the partition table is always a critical decision, and you need to take care of future data as well as the volume of data as well. For example, if you have data of a particular location then partition based on state can be one of the ideal choices.

Here is the syntax to create partition table-

```
CREATE TABLE countrydata_partition  
(Id int, countryname string, population int, description string)  
PARTITIONED BY (country VARCHAR(64), state VARCHAR(64))  
row format delimited  
fields terminated by '\t'  
stored AS textfile;
```

### There are two types of partition in Hive-

- Static partition
- Dynamic partition

Static partition is the default one. To use dynamic partition in Hive, you need to set the following property-



## 6. Bucketing can also be used

If you have more number of columns on which you want the partitions, bucketing in the hive can be a better option. We use CLUSTERED BY command to divide the tables in the bucket.

Here is the syntax to create bucketed table-

```
CREATE TABLE emp_bucketed_table(  
ID int, name string, address string, salary string )  
COMMENT 'this is a bucketed table'  
PARTITIONED BY (country VARCHAR(64))  
CLUSTERED BY (state) INTO 10 BUCKETS  
STORED AS TEXTFILE;
```

To enable bucketing in Hive, you need to set the following property-

```
SET hive.enforce.bucketing=true;
```

This should be set every time you are writing the data to the bucketed table.

## 7. Parallel execution

As we know, Hive converts the queries into different stages during execution. These stages are usually getting executed one after the other and thus increases the time of execution. Below are some of the normal steps involved-

But the good thing is, you can set some of this independent stage to process parallel. This is a parallel execution in Hive. For this, you need to set the below properties to true-

```
Set hive.exec.parallel = true;
```

## 8. Vectorization

Vectorization improves the query performance of all the operation like scans, aggregations, filters and joins, by performing them in batches of 1024 rows at once instead of single row each time.

Again you will have to set some parameter to enable vectorization-

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

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

## 9. Cost based optimization

Cost based optimization (CBO) is the new feature to Hive. CBO offers better hive query performance regarding cost.

To use CBO, you need to set the following properties-

```
set hive.cbo.enable=true;
```

```
set hive.compute.query.using.stats=true;
```

## 10. Avoid Global sorting

Global sorting in Hive is getting done by the help of the command ORDER BY in the hive. But the issue is, if you're using ORDER BY command, then the number of reducers will be set to one which can be illogical when you have large **Hadoop dataset**.

So when you don't need global sorting, use SORT BY command which sorts the result per reducer.

Even you can also use DISTRIBUTE BY command if you want to control which particular rows will go with which reducer.

These were some of the best Hive performance tuning techniques one can apply to Hive. Use these techniques and improve Hive query performance easily.

Do let me know if you have any other method to improve the hive query performance.





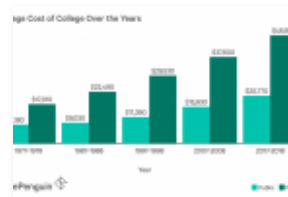
## 5 Best Online Shopping Hacks for Smart Shopping

Scenario Based Hadoop Interview Questions and Answers [Mega List]

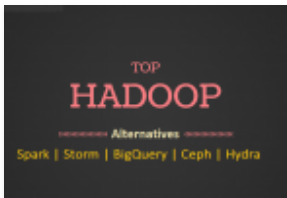
### You may also like



**Hadoop**  
Hadoop for Beginners 101: Where to Start and How



**Hadoop**  
Understanding the Rising Cost of Higher Education



**Big Data • Hadoop • Spark • Technology**  
5 Top Hadoop Alternatives to Consider in 2020



**Hadoop**  
7 Things to Know to Choose a Winning Topic for...

### 5 Comments



**Romeo sloman**

December 31, 2016 at 8:26 am

Great post guys!!

Agree with you that CBO plays an important role in the optimization.

Reply



January 9, 2017 at 11:59 pm

[Home](#)

[Online Training](#) ▾

[Free Course](#) ▾

[Jobs](#)

[Blog](#)

[Books](#) ▾

[Forum](#)



Awesome post sir!!

Need some help on Sqoop...can I post it here or should I email you?

[Reply](#)



**Liva Sett**

January 12, 2017 at 8:33 pm

Great post buddy.

Liked the joining and CBO part and it helps a lot when it comes to timing issue with the query.

Thanks again!

[Reply](#)

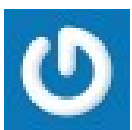


**Mjdmse**

March 25, 2017 at 3:49 pm

Hive Performance Tuning – Optimize Hive Query Perfectly

[Reply](#)



**CharlesBab**

August 6, 2017 at 4:23 am

Thanks so much for these Hive optimization tips.

[Reply](#)










[Home](#)[Online Training ▾](#)[Free Course ▾](#)[Jobs](#)[Blog](#)[Books ▾](#)[Forum](#)[report this ad](#)

## Recent at Hdfs Tutorial

[8 Reasons to Hire a Tutor](#)[5 Benefits of Remote Viewing Cameras for Businesses](#)[5 Big Data Use Cases in Banking and Financial Services](#)[PUBG Mobile: Beginner Tips and Tricks for Survival](#)[10 Best Online Video Games Websites](#)

# HdfsTutorial

<b>Analytics Jobs</b>	<b>Home</b>	Online Training ▾	Free Course ▾	Jobs	Blog	Books ▾	Forum	🔍
-----------------------	-------------	-------------------	---------------	------	------	---------	-------	---

	<b>Data Scientists [3+ yrs Exp]</b> <a href="#">Bangalore</a> ▸ <a href="#">Forecubes</a> Inc
	<b>Sr.Analyst (Fraud Detection) [4-8 yrs]</b> <a href="#">Gurgaon</a> ▸ <a href="#">American</a> Express
	<b>Data Scientist – Tiger Analytics</b> <a href="#">Chennai</a> ▸ <a href="#">Tiger</a> Analytics
	<b>Analytics Manager and 2 Others</b> <b>Profiles</b> <a href="#">India</a> ▸ <a href="#">Think</a> Analytics
	<b>Tableau Developer</b> <a href="#">Pune</a> ▸ M3BI

<b>Like Us On Facebook</b>





[report this ad](#)[report this ad](#)

## About Us

Hdfs Tutorial is a leading data website providing the online training and Free courses on Big Data, Hadoop, Spark, Data Visualization, Data Science, Data Engineering, and Machine Learning. The site has been started by a group of analytics professionals and so far we have a strong community of 10000+ professionals who are either working in the data field or looking to it. You can check more [about us here](#). If you are looking to advertise here, please [check](#)



## Popular Posts

5 Big Data Use  
Cases in Banking  
and Financial  
Services



The Role of Big Data  
in Revolutionizing  
Post-COVID-19  
Real Estate



EXL Data Science  
Interview Questions

## Our Services

We here at Hdfs Tutorial, offer wide ranges of services starting from development to the data consulting. We have served some of the leading firms worldwide. If you are looking for any such services, feel free to check **our service offerings** or you can email us at **hdfstutorial@gmail.com** with more details.

Along with this, we also offer online instructor-led training on all the major data technologies.