TPC-H Benchmarking Hive and Spark

Manasa Bollavaram

June 7, 2017

1 Introduction

The report is aimed at deciding between Hive and Spark according to the need. Accordingly, TPC-H benchmarking, an ad hoc decision support benchmark, of Hive and Spark is conducted. The tag line of Spark, "Lightning-Fast Cluster Computing", provides an apt description of Spark. Especially waiting overnight to get jobs done with Hive and realizing you can get it done in a span of minutes with Spark, I wouldnt hesitate to call Spark as "Lightning-Fast**est** Cluster Computing".

However, Spark has its fair share of issues too, especially memory issues. If the input big data is not partitioned well, there is a high probability of facing out of memory issues and Spark might just shut down, one aspect where Hive sweeps in and takes its trophy.

Scalability analysis of Hive with datasize and clusters is also explored. It is found that, the performance does not increase linearly with the increase in resources and the performance increase is a lot of query and workload specific.

.

2 Evaluated systems

Hive is a data warehouse infrastructure built on top of Hadoop [1]. Hive can be employed to perform data summarization, ad hoc querying, big data analysis etc. Hive uses HiveQL querying language. HiveQL is a SQL like querying language. The SQL like queries are converted to Map Reduce scripts automatically relieving user to write complex Map Reduce Scripts. Hive 2.1.1 with Hadoop 2.7.3 on AWS clusters has been used to benchmark.

Spark is a cluster-computing system for big data. One of the design features, making Spark a fast computing system, is it's ability to store the intermediate data in memory and reuse it. This ability makes Spark fast on machine learning problems where analysis includes multiple iterations on data. Spark 2.1.0 with Hadoop 2.7.3 on AWS clusters has been used to benchmark. Zeppelin 0.7.1 has been used to run Scala code to load data into

Spark and run SQL queries.

AWS Clusters of 3 nodes and 5 nodes have been used to perform benchmarking. In the 3 nodes or 5 nodes, one node acts as a Master Node storing the metadata/job tracker and remaining slave nodes are used to store the data. m3.xlarge instances have been used whose specifications are 4 vCPU cores, 15GB RAM, 2 x 40 (GB) SSD storage, high frequency Intel Xeon E5-2670 v2 (Ivy Bridge) Processor. Elastic Block Storage of 500GB is used for each run to store the data.

3 Benchmark Used for Evaluation

TPC-H Benchmark has been used to evaluate Hive and Spark systems. TPC-H is an ad-hoc decision support benchmark, where ad hoc decisions imply making decisions on problems not anticipated and not recurring. Benchmark consists of 22 ad hoc queries and concurrent data modifications. It provides the data and query generator programs. Dbgen program generates the data with the desired scalefactor. Qgen program generates queries. To query Hive, SQL queries have been slightly modified because HiveQL does not support sub queries. Each query was run 3 times and average is reported in the paper.

4 Data

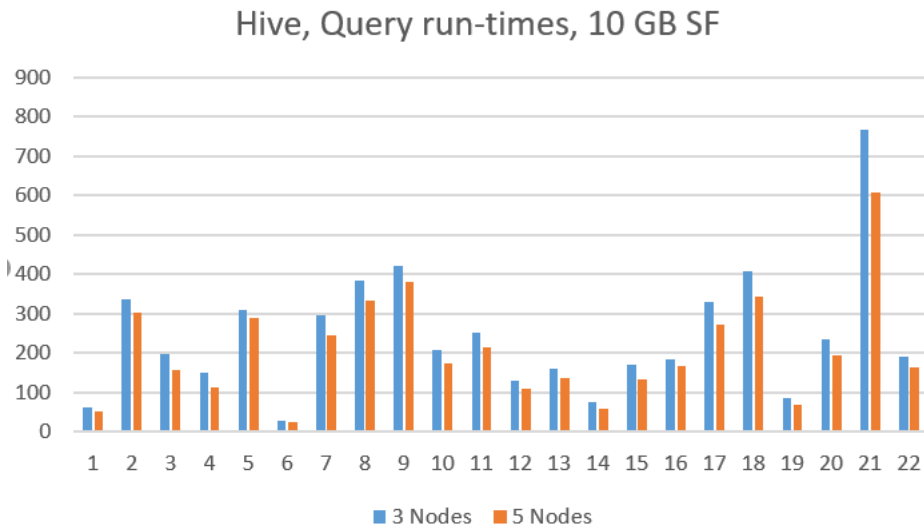
Data of scale factors 10 and 100 have been generated using dbgen program and pre-loaded into HDFS before running queries. In the case of Hive, tables are stored in Optimized Row Columnar (ORC) format and Zlib compressed. In the case of Spark, data is converted to a dataframe , equivalent to a relational table and stored as temporary tables. Data is regenerated for every analysis to ensure that the input data is replicated uniformly across all nodes. Hive and Spark are run separately to ensure exclusive access to clusters resources.

5 Evaluation results

TPC-H Benchmark queries were evaluated for Hive and Spark, with 10GB and 100GB datasets, on 3 Nodes and 5 nodes AWS clusters. Total run-time results are provided in the appendix. Following are the facts reported regarding speed.

- Spark is 3.88x faster compared to Hive on a 10GB dataset, 3 Node cluster
- Spark is 4.50x faster compared to Hive on a 10GB dataset, 5 Node cluster
- Spark is 1.5x faster compared to Hive on a 5 node cluster using 100 GB dataset, only first three queries of TPC-H were run
- Hive is 1.3x faster when 2 nodes scaled up to 4 nodes , on 10 GB dataset

Figure 1: TPC-H Benchmark Results on Hive, with 10GB dataset, on 3 and 5 node clusters from AWS



- Hive is 1.65x faster when 2 nodes scaled up to 4 nodes, on 100 GB dataset
- Hive is 3.26x faster when data is scaled down from 100 GB to 10 GB on a 3 node cluster
- Hive is 2.5x faster when data is scaled down from 100 GB to 10 GB on a 5 node cluster
- Spark is 1.5x faster when 2 nodes scaled up to 4 nodes, on 10 GB dataset

5.1 Scalability of Hive:

From Fig. 1, for most of the queries there is less than a linear increase in speedup for Hive when scaled from 2 to 4 nodes. For queries q5, q22 and q19, the performance almost remains the same. But for a 100 GB dataset as indicated in Fig. 2, the performance of Hive is almost linearly spedup for most of the queries, when scaled from 2 to 4 nodes. Although, for some queries like q11, the performance almost remains the same.

From Fig. 3 and 4, it is clear that performance increase is less than linear with the scaling down of data. For certain queries like q6, performance is not affected much with the data scale.

Some generalized statements can be made considering total run-time such as scaling the data 10 times makes it a lot slower on a 3 node cluster compared to a 5 node cluster. Definitely a nonlinear increase in performance with the availability of resources. But, for a

Figure 2: TPC-H Benchmark Results on Hive, with 100 GB dataset, on 3 and 5 node clusters from AWS

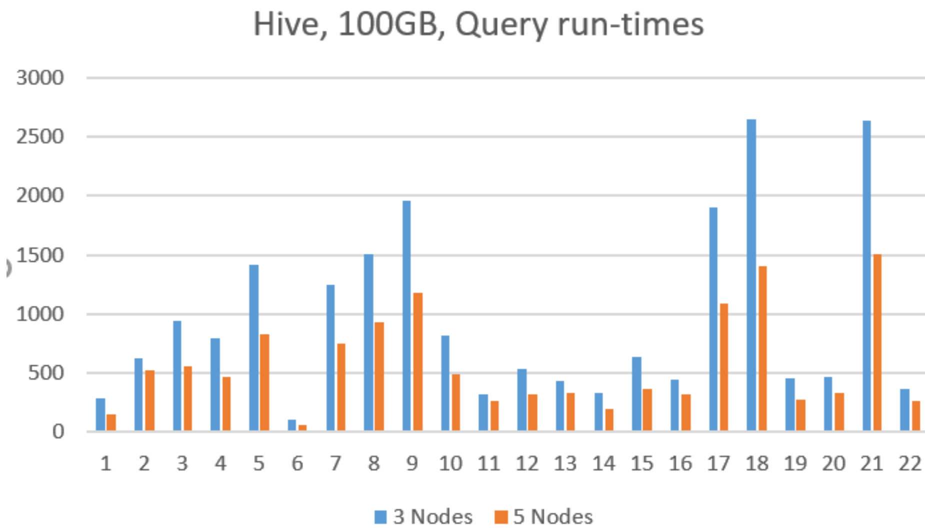


Figure 3: TPC-H Benchmark Results on Hive, with a 3 node cluster, on 10 GB and 100 GB datasets

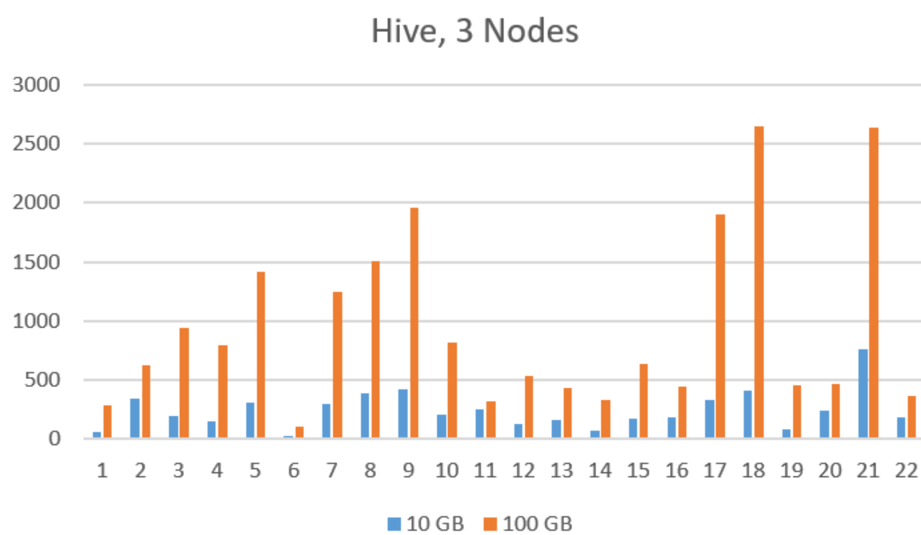


Figure 4: TPC-H Benchmark Results on Hive, with a 5 node cluster, on 10 GB and 100 GB datasets

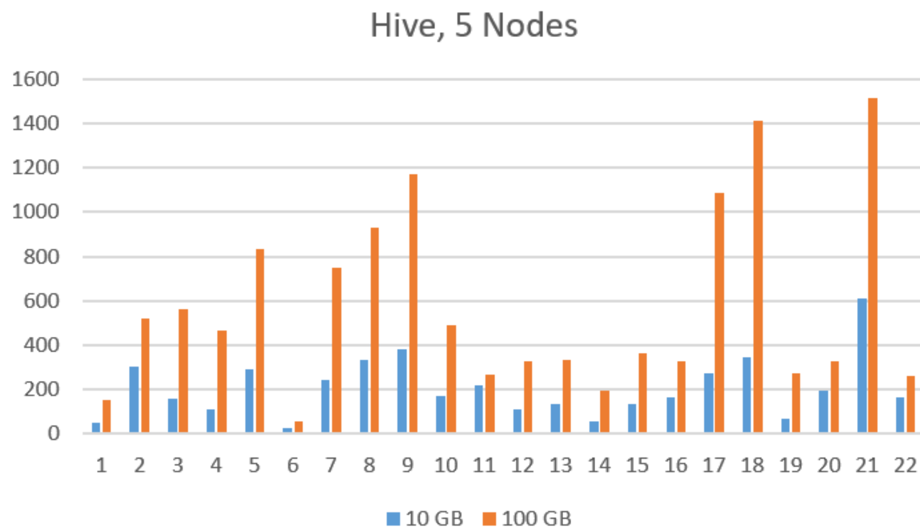


Figure 5: TPC-H Benchmark Results on Hive and Spark, with 10GB dataset on 3 nodes from AWS

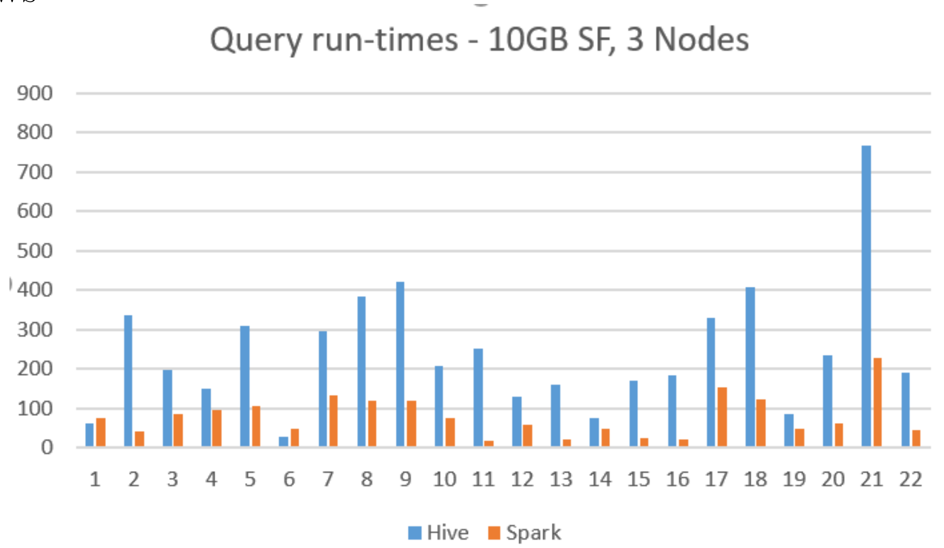


Figure 6: TPC-H Benchmark Results on Hive and Spark, with 10GB dataset on 5 nodes from AWS

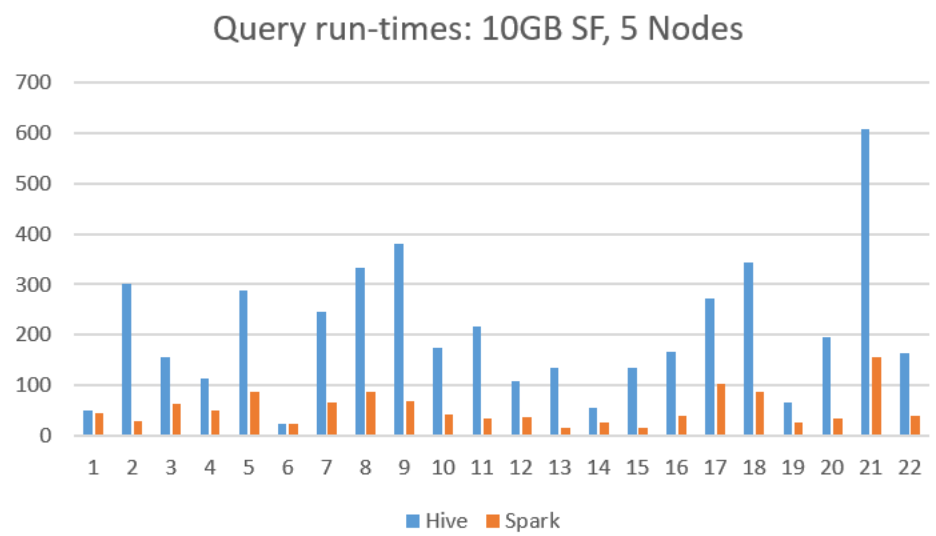


Figure 7: TPC-H Benchmark Results of first three queries on Hive and Spark, with 100 GB dataset, on a 5 node cluster from AWS

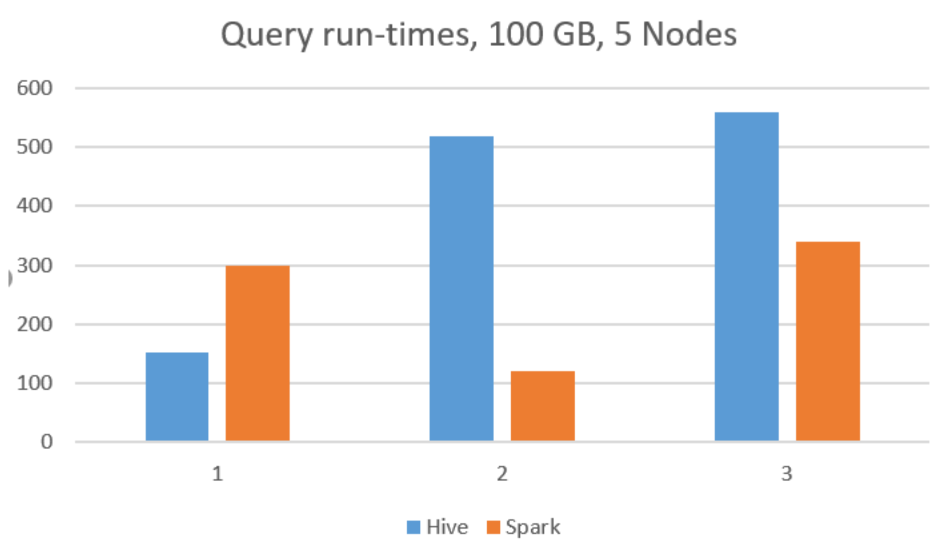
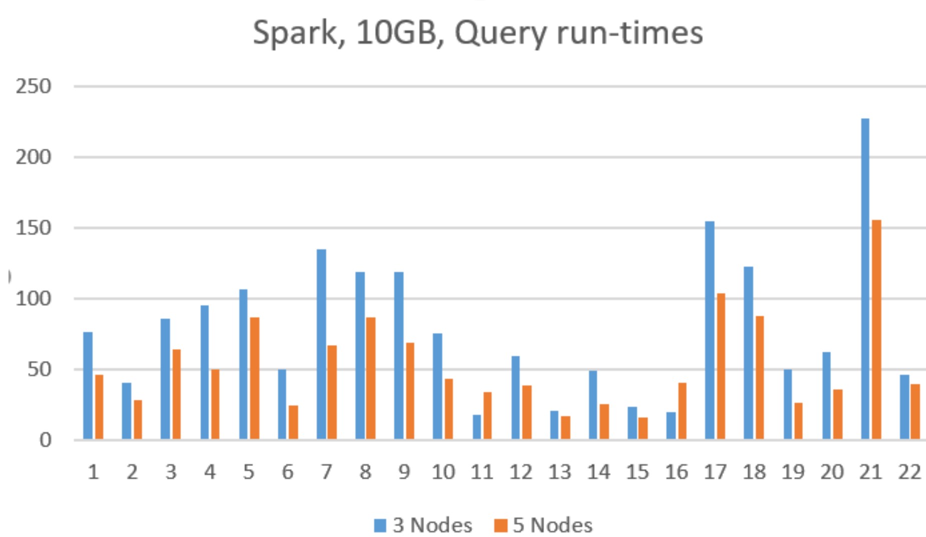


Figure 8: TPC-H Benchmark Results on Spark, with 10 GB dataset, on 3 and 5 node clusters from AWS



lot of cases performance is query specific.

5.2 Hive vs Spark

From Fig. 5 and 6, it is clear that spark is much faster for most of the queries compared to Hive. However, for some queries like q1, q6, Hive seems to be doing equally well or even better than Spark. For queries like q13, q16, q11, Spark provides a more than linear increase in speedup when compared to Hive, for nodes scaled up from 2 to 4.

From Fig. 7, it is clear that Hive performs better compared to Spark for query 1. Query 1 involves aggregates over a large table and Spark seems to not perform better in such cases if the table is not partitioned well. In fact, for query 4 which involves a lot of aggregate tasks on the largest table, spark shuts down giving out of memory issues. It can be handled by partitioning the tables and spark should definitely be performing better in the case of partitioned tables. But, it raises an interesting drawback of Spark where spark is not suitable for multi-users because the memory has to be known before submitting jobs.

The increase in performance of spark to hive is a lot of query and workload specific. But a generalized statement is that Spark is faster compared to hive although the performance may not scale linearly.

5.3 Scalability of Spark

As observed in the case of scalability of Hive, the same applies to Spark. The performance does not scale linearly and is a lot of query specific. From Fig. 8, for queries like q21, q14, q19, q4, Spark shows a linear speedup for nodes scaled up from 2 to 4. For queries like q16, q11, Spark ran slower when nodes are scaled up from 2 to 4. For queries like q17, q18, there is a less than linear increase in speedup.

6 Conclusion

Spark is definitely faster than Hive but comes at an expensive RAM, and a lot of out of memory issues too. If there is no urgency for results, then Hive is better as it is cheap. But for real time streaming data processing, Spark is the best. Also, for multiple iterations on the data, Spark performs better because it performs in-memory analytics making it quicker to use the data compared to hive which converts SQL like queries to map reduce jobs, map reduce jobs write intermediate data to disk.

7 Appendix

Total run-time of 22 queries for 10 GB dataset

- 3 Nodes, 3 trials for each query
  - Hive - 5 hr 50 min
  - Spark 1hr 30min
- 5 Nodes, 3 trials for each query
  - Hive 4 hr 30 min
  - Spark 1hr

Total run-time of 22 queries on Hive for 100 GB dataset, 3 trials for each query

- 3 Nodes 19 hr
- 5 Nodes 11hr 30min

Total run-time of 3 queries on Spark for 100 GB dataset

- 5 Nodes 40 min

8 References

1. Running the TPC-H Benchmark on Hive, Yuntao Jia, August 10, 2009
2. TPC-H Benchmark, http://www.tpc.org/tpch/
3. TPC-H Benchmark Document, http://www.tpc.org/tpch/spec/tpch2.8.0.pdf