

Benchmarking SQL-on-Hadoop Systems: TPC or Not TPC?

Avrilia Floratou¹(✉), Fatma Özcan¹, and Berni Schiefer²

¹ IBM Almaden Research Center, 650 Harry Road, San Jose, CA, USA
{aflorat,fozcan}@us.ibm.com

² IBM Toronto Lab, 8200 Warden Ave, Markham, ON, Canada
schiefer@ca.ibm.com

Abstract. Benchmarks are important tools to evaluate systems, as long as their results are transparent, reproducible and they are conducted with due diligence. Today, many SQL-on-Hadoop vendors use the data generators and the queries of existing TPC benchmarks, but fail to adhere to the rules, producing results that are not transparent. As the SQL-on-Hadoop movement continues to gain more traction, it is important to bring some order to this “wild west” of benchmarking. First, new rules and policies should be defined to satisfy the demands of the new generation SQL systems. The new benchmark evaluation schemes should be inexpensive, effective and open enough to embrace the variety of SQL-on-Hadoop systems and their corresponding vendors. Second, adhering to the new standards requires industry commitment and collaboration. In this paper, we discuss the problems we observe in the current practices of benchmarking, and present our proposal for bringing standardization in the SQL-on-Hadoop space.

Keywords: SQL · Hadoop · Benchmark · TPC · SPEC · STAC · TPC-DS · TPC-H

1 Introduction

Benchmarks are an integral part of software and systems development, as they provide a means with which to evaluate systems performance in an objective way. While the discussion and the work on new big data benchmarks are in progress, many vendors use the Transaction Processing Performance Council (TPC) [22] benchmark schema, data generator, and queries but are very selective about which parts of the specification and disclosure rules they follow. The TPC was formed to help bring order and governance on how performance testing should be done and results published. Without the rules, the results are not comparable, and not even meaningful.

In the relational database world there was a transition from the “lawless” world of Debit-Credit to the much more rigorous and unambiguous world of TPC-A/TPC-B [11].

Central to the concepts pioneered by the TPC include:

1. The notion of a specification that is sufficiently high level to permit multiple vendor participation while simultaneously ensuring a high degree of comparability.
2. The concept of “full disclosure” or disseminating sufficient detail that it should be possible to both understand and potentially duplicate the published results.
3. The requirement to “audit” results to ensure adherence to the specification.

In addition, the TPC Policies describe the manner in which the benchmark results can and can not be compared in a public forum. These “Fair Use” rules set the standard for what is and is not allowed¹. In particular there is a requirement for:

1. **Fidelity:** Adherence to facts; accuracy
2. **Candor:** Above-boardness; needful completeness
3. **Due Diligence:** Care for integrity of results
4. **Legibility:** Readability and clarity

In contrast to this well-regulated relational database benchmarking environment, in the world of SQL-on-Hadoop we are in the “wild west”. Example of this is the (mis)use of the TPC benchmarks by the SQL-on-Hadoop systems. Some example SQL-on-Hadoop systems include IBM Big SQL [3, 12, 15], Hortonworks Hive [2], Cloudera Impala [5], Presto [18], Microsoft Polybase [8], and Pivotal HAWQ [17].

The relatively “free” access and EULA (End User License Agreements) rules of newer systems that are not constrained by the famous “DeWitt” clause² makes it easy for the SQL-on-Hadoop vendors to conduct performance comparisons between their system and competitors and publish the results. In fact, so far in 2014 we have observed a large number of web blogs written by SQL-on-Hadoop vendors that compare the performance of their system against the competitor systems reporting results using components of database benchmarks such as TPC-H [27] and TPC-DS [23]; Two benchmarks that are very popular for testing SQL-based query processing capabilities of relational databases.

A closer look at these performance comparisons reveals that the rules of the benchmarks are typically not followed. As we will discuss in more detail in the following section, it is common for vendors to pick a subset of the queries of the benchmark and perform a comparison using only those. Secondly, the queries are modified because more often they only support a limited subset of the SQL standard. Finally, it is not clear whether and how well the competitor open-source system was tuned.

According to the TPC standards, each database vendor installs, tunes and performs a full run of the database benchmark according to the benchmark specification, on his system only and then produces a report that describes all the details. This report and the performance results are audited by an accredited

¹ http://www.tpc.org/information/about/documentation/tpc_policies_v6.0.htm#_Toc367096059.

² http://en.wikipedia.org/wiki/David_DeWitt.

TPC Auditor and are submitted to the TPC for certification. When a potential customer wants to compare various systems in the same category from different vendors, she can review the high level performance metrics, including price performance, the detailed underlying implementation details, and the auditor’s certification letter indicating compliance with all benchmark requirements.

In this paper, we want to point out the fact that in the world of SQL over Hadoop rigorous scientific benchmarking has been replaced by unscientific comparisons in the name of marketing and we would like to draw attention to this problem. We believe new benchmarks are needed that test not only the traditional structured query processing using SQL, but also emphasize the unique features of the Hadoop ecosystem and emerging big data applications. We also emphasize the need for benchmark specifications and industry commitment in order to bring standardization in the SQL-on-Hadoop space. Finally, we present our proposal towards these objectives.

2 Current Practices for Reporting Performance Results

In this section, we provide some examples of how SQL-on-Hadoop vendors use the TPC benchmarks when they evaluate their systems against the competitor systems.

One commonly used benchmark by all SQL-on-Hadoop vendors is the traditional TPC-DS benchmark [23]. The TPC-DS benchmark consists of 99 queries which access 7 fact tables and multiple dimension tables. TPC-H is also another popular benchmark used by the SQL-on-Hadoop vendors to test the performance of their systems. For those interested, the 22 TPC-H queries for Hive and Impala are available online [28, 29].

Recently, the Cloudera Impala developers have published a comparison between Hive, Impala and a traditional DBMS (called DBMS-Y) using TPC-DS as the workload basis [6, 24] and subsequent comparisons between Impala and other SQL-on-Hadoop systems [25, 26]. In their first comparison [24] they argue that using this workload Impala can be up to 69X faster than Hive and is generally faster than DBMS-Y with speedups up to 5X. In their subsequent comparison they find that Impala is on average 5X faster than the second fastest SQL-on-Hadoop alternative (Shark). The Impala developers have also provided the data definition language statements and queries that they have used in their study [13].

By taking a closer look at these queries, we will observe that only 19 queries out of the 99 TPC-DS queries are used. An additional query which is not part of the TPC-DS benchmark has also been introduced. Moreover, these queries access a single fact table only (out of the 7 fact tables that are part of the TPC-DS dataset). This results in query plans of a similar pattern when the fact table is joined with multiple dimension tables: the small dimension tables are broadcast to the nodes where the fact table resides and a join is performed locally on each of these nodes, without requiring any repartitioning/shuffling of the fact table’s data.

Clearly, picking a subset of the queries that have a common processing pattern (for which a particular system is optimized) and testing the systems on only those queries do not reveal the full strengths and limitations of each system and is against the rules of the TPC-DS benchmark.

Another observation on the study of the queries in [24–26, 28, 29] is that some of these queries were modified in various ways:

- Cloudera’s Impala does not currently support windowing functions and rollup. Thus, whenever a TPC-DS query included these features, these were removed from the query. This is not fair to other systems such as Apache Hive and possibly DBMS-Y which already support these features.
- An extra partitioning predicate on the fact table has been added in the where clause of each query [24–26]. This predicate reduces the amount of the fact table data that need to be accessed during the query processing. It is worth noting that there exist advanced query rewrite techniques that introduce the correct partitioning predicates on the fact table without manual intervention [21].
- According to the TPC-DS specification the values in the query predicates change with the TPC-DS scale factor. However, the queries published in [13] do not use the correct predicate values for the supported scale factor.
- In both the TPC-H and the TPC-DS benchmarks, the queries contain predicates on the DATE attributes of the tables. These predicates typically select a date interval (e.g., within a month from a given date). This interval should be computed by the system under test by using, for example, built-in date functions. However, in the published TPC-H and TPC-DS inspired queries by the open-source vendors in [24–26, 28, 29], the date intervals are already pre-computed by the authors of the queries.
- In some queries, the predicates in the WHERE clauses are transformed into predicates in the ON clauses and are manually pushed down closer to the join operations.
- Some queries are re-written in a specific way that enforces the join ordering during the query execution.

When vendors present performance comparisons between different versions of their own systems, they use a very small set of the queries. This creates a false impression of a general characteristic when only a small biased (hand-picked) number of queries is used to substantiate a claim. For example, in a blog from Hortonworks [14], a single TPC-DS query is used to show the performance benefits using Hive’s ORC file format with predicate pushdown over the previous Hive versions.

In this blog, Hortonworks also claims that the ORC columnar format results in better compression ratios than Impala’s Parquet columnar format for the TPC-DS dataset at 500 scale factor. This claim, was later dismissed by Cloudera [6], which showed that if the same compression technique is used, the Parquet columnar format is more space-efficient than the ORC file format for the same dataset. This incident points out the importance of transparency. It is also worth noting that a 500 scale factor is not a valid scale factor for the TPC-DS workload.

It is quite clear that the rules established by the TPC for the TPC-H and the TPC-DS benchmark specifications are not followed in today’s experimental comparisons for SQL-on-Hadoop systems. This can be quite misleading, since each SQL-on-Hadoop vendor can now modify, add or remove queries when comparing to other SQL-on-Hadoop systems. In fact, it is not even clear how well the competitor systems are tuned for these modified workloads.

Another performance comparison between different SQL-on-Hadoop systems have been published from the UC Berkeley AMPLab [1]. The benchmark used in these comparisons is inspired by the queries used in a 2009 SIGMOD paper [16] and contains only 4 queries. We would like to point out that the queries used in [16] were created in order to compare the vanilla MapReduce framework with parallel databases, and to prove that parallel databases excel at join. Since at that time MapReduce had limited functionality, these queries were very simple (e.g., contained at most one join operation). Today’s SQL-on-Hadoop systems are much more sophisticated than the vanilla MapReduce framework and thus should not be evaluated with such simple benchmarks but with more sophisticated benchmarks that will reveal their full strengths and limitations.

We believe that these few examples clearly demonstrate the chaotic situation that currently exists in the SQL-on-Hadoop world when it comes to benchmarking and performance evaluation.

3 Whither SQL-on-Hadoop Benchmarking

Given the fact that benchmarking in the SQL-on-Hadoop world is in a “wild west” state, a natural question to ask is “What is the solution to this problem?” We believe that the SQL-on-Hadoop vendors should do the following:

1. Build on the decades of RDBMS benchmarking experience and move to a new generation of SQL over Hadoop benchmarking (e.g., [4]).
2. Employ good scientific experimental design and procedures that generate reproducible, comparable and trustworthy results.
3. Adhere to the TPC specifications and policies when using the TPC benchmarks.
4. Create new benchmarks for the SQL-on-Hadoop systems that represent the characteristics and features of the new generation of big data applications (e.g., BigBench [10], TPC-DS Hadoop/Hive Friendly, etc.)
5. Agree on the rules for new benchmarks, and extend existing ones as needed so that all vendors follow them when reporting benchmark results.

Since the benchmarking rules and policies are critical in bringing standardization in the SQL-on-Hadoop space, in the following sections, we discuss the benchmarking methodologies of different industry standard benchmark councils, as well as their relevance to the SQL-on-Hadoop benchmarking space. Finally, we present an outline of our own proposal for bringing standardization in this space.

3.1 TPC

The TPC [22] has been the leading benchmarking council for transaction processing and database benchmarks. However, TPC has been somewhat mired in the traditions of the past, and has been slow to evolve and invent new benchmarks to represent modern workloads and systems. While there has been movement to providing more artifacts to lower the cost of participation, the benchmark kits of the standard TPC benchmarks produced by each vendor remain proprietary. The costs to audit and publish can be prohibitively high and are likely a key reason for low vendor participation. This raises the issue on whether TPC is the organization that will address the problems faced in the SQL-over-Hadoop space.

Recently, the TPC has started re-inventing itself by introducing the TPC **Express** process [9] and by launching TPCx-HS [30], the first benchmark that follows the TPC **Express** process. The TPC **Express** process aims at lowering the cost of participation at TPC and at making the TPC benchmarks accessible to a broad class of practitioners including academia, consumers, analysts and computer hardware and software manufacturers [9]. The TPCx-HS benchmark is the first TPC benchmark focused on big data systems such as Hadoop. As opposed to the previous TPC benchmark, the TPCx-HS benchmark is available via the TPC Web site in the form of a downloadable kit. The existence of a kit, independent of the vendor that runs the benchmark, is a key characteristic of the TPC **Express** process which aims to make the benchmark more readily available. As opposed to other database benchmarks such as TPC-H, whose results must be validated by a certified TPC auditor, the TPCx-HS benchmark results can also be validated using a peer-review process. The members of the peer-review committee are official TPCx-HS members.

Another problem of the TPC benchmarks is that they have historically taken a rather narrow view of the “measure of goodness” while real customers have broader considerations. The former CTO of SAP, Vishal Sikka, published a blog posting that attempted to articulate the characteristics that should be considered in a new benchmark³. While the blog itself was a thinly disguised advertisement for HANA, Vishal does raise some interesting aspects that can and should be considered when constructing a new benchmark. In particular, he points to five core dimensions that should be considered:

1. going deep (the benefit of allowing unrestricted query complexity)
2. going broad (the benefit of allowing unrestricted data volume and variety)
3. in real-time (the benefit of including the most recent data into the analysis)
4. within a given window of opportunity (the benefit of rapid response time)
5. without pre-processing of data (the cost of data preparation)

Most of these are relevant to current and future Hadoop systems and should be taken into account when defining a new benchmark for the SQL-on-Hadoop systems.

³ <http://www.saphana.com/community/blogs/blog/2013/09/16/does-the-world-need-a-new-benchmark>.

3.2 SPEC

The Standard Performance Evaluation Corporation (SPEC) [19] is formed to establish, maintain and endorse a standardized set of relevant benchmarks for high-performance computers. SPEC also reviews and publishes submitted results from the SPEC member organizations and other benchmark licensees. SPEC employs a peer-review scheme of the benchmarking results (instead of an auditing scheme) that has been quite successful. The members of the peer-review committee are official SPEC members. Through this membership, the companies that are willing to accept SPEC’s standards can also participate in the development of the benchmarks.

3.3 STAC

The Securities Technology Analysis Center (STAC) [20] is a relatively new organization. It first formed a Benchmark Council in 2007. It is a large, well-funded organization consisting of over 200 financial institutions and 50 vendor organizations, focused on the securities industry. One unique aspect of this organization is that it is run by companies representing “consumers” of IT products. Vendors, while encouraged to participate, do not control the organization. STAC has only recently become interested in Big Data and formed a Big Data special interest group in 2013. The STAC members have written a white paper that characterizes the major Big Data use cases they envision in the securities and banking industry [7].

STAC has begun working on a Big Data benchmark specification whose details are restricted to its members. It is too early to know how strong a benchmark will emerge, and what value it will have outside the securities industry. We encourage the Big Data community to stay abreast of developments at STAC, and encourage STAC to be more open about the benchmarks and specifications that it generates.

4 Our Proposal

The rate of change, the number of new players, and the industry-wide shift to new communication modes (e.g. blogs, tweets) make it next to impossible to conduct benchmarks using the traditional auditing procedures (such as the TPC auditing process) using database TPC benchmarks. Our belief is that the SQL-on-Hadoop community should build on the experiences of different industry standard benchmark councils to bring standardization in the SQL-on-Hadoop space and to produce fair and meaningful benchmarking results. The necessary steps needed towards this goal are the following:

1. Robust benchmark specifications
2. Flexible, portable, and easy to use benchmarking kits
3. Cost-effective, timely, efficient yet high quality peer-reviewing procedures

We believe that the existence of flexible and downloadable benchmarking kits is a significant step towards the widespread adoption of any benchmark. Proprietary benchmark kits result in high implementation cost, and thus make the benchmarking process expensive to start in the first place. This observation has already been made by TPC, and thus the TPCx-HS benchmark is the first TPC benchmark that is vendor-neutral: It comes with a downloadable benchmarking kit. This is a key characteristic of the TPC **Express** process.

Regarding the reviewing of the benchmarking results, we need a robust scheme that will bring organization, transparency and objectivity in the SQL-on-Hadoop benchmarking space. We believe that the best solution to the SQL-on-Hadoop world's chaotic state is the use of a **peer-review** approach. The goal of this approach is to develop timely, cost-effective but high quality "reviews" to minimize the "marketing" effects. Using this approach, every vendor that publishes or compares a set of SQL-on-Hadoop systems, writes a performance report that includes a description of the hardware and software configuration and the tuning process. This report is peer-reviewed, not only by the vendors of the systems tested but by other independent reviewers in the industrial or academic setting. Once an agreement is reached the results would be published online. As noted in an earlier section, the peer-review scheme has already been used by organizations such as SPEC [19] and has been quite successful.

We believe that all the SQL-on-Hadoop vendors would agree on this approach, since they will be able to argue about the performance of their systems whenever another vendor conducts a performance comparison that incorporates their system. The most challenging part of this process is to ensure that the SQL-on-Hadoop vendors be prevented from blocking the publication of unfavorable results for their systems. To avoid such cases, we propose a "revision" process in which the SQL-on-Hadoop vendors that doubt the validity of the results, will provide concrete feedback that proposes configuration changes that should to be made. A maximum number of revision requests per vendor could also be set (e.g., up to two revision requests). This approach guarantees that: (a) the vendors that have doubts about the validity of the experimental setting/result will be forced to provide accurate and detailed feedback and (b) these vendors will be prevented from indefinitely blocking the publication of the results.

5 Conclusions

The existence of new, more sophisticated benchmarks that can represent the new generation workloads, is certainly a big step for the SQL-on-Hadoop community. However, their formulation is not going to bring any standardization unless these benchmarks are accompanied by rules and policies that will ensure transparency and objectivity. Otherwise, these benchmarks will be abused in the name of marketing, similar to what is happening now with the existing TPC benchmarks. To realize this vision, all the SQL-on-Hadoop vendors should come to an agreement on how to use the benchmarks, and how to report performance results using them. We believe that a **peer-review** approach along with the existence of portable

and easy-to-use benchmarking kits is the only viable solution. Furthermore, templates for summarizing results in a standard way, similar to a TPC Executive Summary, should be created and provided to all. Of course, we do not expect that the existence of published unofficial results that present performance evaluations for different systems will cease to exist. However, if the community agrees upon the standards, publishes results based on the standards, and uses an effective reviewing scheme, then the importance of these web blogs and their impact on the end-user will be significantly reduced. This is exactly the same reason that led to the formation of the TPC and its first benchmark, TPC-A, more than 25 years ago. IBM is eager to join the community in bringing order to the exciting world of SQL-on-Hadoop benchmarking.

References

1. AMPLAB Big Data Benchmark. <https://amplab.cs.berkeley.edu/benchmark/>
2. Apache Hive. <http://hive.apache.org/>
3. IBM InfoSphere BigInsights. http://www-01.ibm.com/support/knowledgecenter/SSPT3X_3.0.0/com.ibm.swg.im.infosphere.biginsights.product.doc/doc/whatsnew.html
4. Chen, Y., Raab, F., Katz, R.: From TPC-C to big data benchmarks: a functional workload model. In: Rabl, T., Poess, M., Baru, C., Jacobsen, H.-A. (eds.) WBDB 2012. LNCS, vol. 8163, pp. 28–43. Springer, Heidelberg (2014)
5. Cloudera Impala. <http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/impala.html>
6. Cloudera Impala Technical Deep Dive. <http://www.slideshare.net/huguk/hug-london2013>
7. Costley, J., Lankford, P.: Big Data Cases in Banking and Securities (2014). <https://stacresearch.com/news/2014/05/30/big-data-use-cases-banking-and-securities>
8. DeWitt, D.J., Nehme, R.V., Shankar, S., Aguilar-Saborit, J., Avanes, A., Flaszka, M., Gramling, J.: Split query processing in polybase. In: ACM SIGMOD, pp. 1255–1266 (2013)
9. TPC Express. http://www.tpc.org/tpctc/tpctc2013/slides_and_papers/004.pdf
10. Ghazal, A., Rabl, T., Hu, M., Raab, F., Poess, M., Crolotte, A., Jacobsen, H.-A.: BigBench: towards an industry standard benchmark for big data analytics. In: ACM SIGMOD, pp. 1197–1208 (2013)
11. Gray, J. (ed.): The Benchmark Handbook for Database and Transaction Systems, 2nd edn. Morgan Kaufmann, San Francisco (1993). <http://research.microsoft.com/en-us/um/people/gray/benchmarkhandbook/chapter2.pdf>
12. Groves, T.: The Big Deal about InfoSphere BigInsights v3.0 is Big SQL. <http://www.ibmbigdatahub.com/blog/big-deal-about-infosphere-biginsights-v30-big-sql>
13. Impala TPC-DS Kit. <https://github.com/cloudera/impala-tpcds-kit>
14. ORCFile in HDP 2.0. <http://hortonworks.com/blog/orcfile-in-hdp-2-better-compression-better-performance/>
15. Ozcan, F., Harris, S.: Blistering Fast SQL Access to Your Hadoop Data. <http://www.ibmbigdatahub.com/blog/blistering-fast-sql-access-your-hadoop-datal>
16. Pavlo, A., Paulson, E., Rasin, A., Abadi, D.J., DeWitt, D.J., Madden, S., Stonebraker, M.: A comparison of approaches to large-scale data analysis. In: ACM SIGMOD. ACM, New York (2009)

17. Pivotal HAWQ. <http://pivotalhd.docs.gopivotal.com/getting-started/hawq.html>
18. Presto. <http://prestodb.io/>
19. SPEC: Standard Performance Evaluation Corporation. <http://www.spec.org/>
20. STAC: Security Technology Analysis Center. <https://stacresearch.com/>
21. Szlichta, J., Godfrey, P., Gryz, J., Ma, W., Pawluk, P., Zuzarte, C.: Queries on dates: fast yet not blind. In: Proceedings of the 14th International Conference on Extending Database Technology, EDBT/ICDT 2011, pp. 497–502. ACM, New York (2011)
22. Transaction Processing Performance Council. <http://www.tpc.org>
23. The TPC-DS Benchmark. <http://www.tpc.org/tpcds/>
24. TPC-DS-like Workload on Impala (part 1). <http://blog.cloudera.com/blog/2014/01/impala-performance-dbms-class-speed/>
25. TPC-DS-like Workload on Impala (part 2). <http://blog.cloudera.com/blog/2014/05/new-sql-choices-in-the-apache-hadoop-ecosystem-why-impala-continues-to-lead/>
26. TPC-DS-like Workload on Impala (part 3). <http://blog.cloudera.com/blog/2014/09/new-benchmarks-for-sql-on-hadoop-impala-1-4-widens-the-performance-gap/>
27. The TPC-H Benchmark. <http://www.tpc.org/tpch/>
28. TPC-H Scripts for Hive. <https://issues.apache.org/jira/browse/HIVE-600>
29. TPC-H Scripts for Impala. <https://github.com/kj-ki/tpc-h-impala>
30. The TPCx-HS Benchmark. <http://www.tpc.org/tpcx-hs/spec/tpcx-hs-specification-v1.1.pdf>