

Big Data Benchmark - Big DS

Jun-Ming Zhao^(✉), Wen-Shuan Wang, Xian Liu, and You-Fu Chen

HP Building, No 112, Jianguo Road Chaoyang District, Beijing 100022, China
{jimmy.zhao, wenshuan.wang, shine.liu,
youfu.chen}@hp.com

Abstract. Performance and scalability in clusters of heterogeneous and complex Big Data Analytic environments are always unpredictable. In this paper, we are trying to address this problem by using a benchmark named “Big DS”. The benchmark adopts many great ideas from some famous industry benchmarks like TPC-H [1], TPC-DS [1], SPECvirt_sc2010 [2] and SPECjbb2005 [2], we also adopt some ideas from non-standard benchmarks liked TeraSort [3], SWIM [4], etc. By defining a configurable workload for different big data analytics environment, Big DS can be used for measuring the performance and scalability of a big data analytics platform or environment for different business.

Keywords: Big data · Analytics platform · Analytics engine · Benchmark · Performance · Scalability · Analytics · TPC [1] · SPEC [2] · TPC-DS [1] · Terasort · SWIM [4] · Hadoop [5] · Hive [6] · Impala [7] · Data warehouse · BI

1 Introduction

Since Google published BigTable [8], a lot of new Big Data technologies and solutions have been innovated from different communities or vendors, with different programming languages and different features. The most famous one of them is Hadoop [5]. You probably can see a new name of Big Table, NoSQL or NewSQL jump into your eyes every 3 months, claiming different brilliant features. But it is hard to tell which technology or solution has better performance and scalability.

At the same time, there are many requirements for a meaningful benchmark. Karl Huppler [9] defined the following set of attributes of a good benchmark:

- Relevant
- Repeatable
- Fair and portable
- Economical
- Verifiable

While designing Big DS, the above requirements were taken into consideration. Specifically they were implemented as follows:

- Relevant
 - Real world environments – The target environment of Big DS is a hybrid and scale-out data mining system/cluster. Just like most of big data environments,

the data will be stored in a cluster and it will also be analyzed in the same system.

- Real world workloads - The benchmark need to be a good sample of the real world workloads. In the big data area, there are so many different use cases. The workload should be able to handle huge volume data, the data will be integrated using the similar way of real world; the workload should be able to handle different data structure with different size.
- Performance and Scalability of the whole cluster – Performance and scalability of target systems are essential for cluster-oriented benchmarks. Benchmark needs to show different capability of target system and their scalability for better predicting future usage of the system.
- Repeatable and verifiable
 - Repeatability and verifiability of benchmark results are always important, no matter what type of workload is measured. In Big DS, we will reuse the successfully used business model from TPC-DS with important updates, which simulate the social and mobile impact to the business model of the real world. The micro-benchmarks will be re-designed carefully, so that their test result can keep consistent in different runs in the same configuration.
- Economical
 - Big data clusters tend to be expensive, even though each node of the cluster might be cheap. So, while designing the benchmark, we are trying to use a thin driver design to reduce the cost of the overall system. And, the benchmark will include performance per dollar and performance per watt metrics, so that benchmark publishers will use more cost conscious system configurations. To be more specific, total cost of the system and total power consumption of the cluster will be included in the benchmark report.

2 Problem Statements

2.1 Gaps Analysis

Here we list the typical problems or requirements for a benchmark while choosing a big data analytics system, from different perspectives of big data customers. We also list the solutions to fill the gaps:

- Buyer of Big Data Analytics System:
 - Problem 1: Complicated and different business models, configurations, execution processes and metrics. It's really hard to compare with each other.
 - Solution: We need a well-defined and trustable business model; a proved fix process to execute; a vendor independent design; and fully disclosed and auditable result.

- Big Data Solution Vendors
 - Problem: How to easily show my own value?
 - Solution: Common metrics and sub metrics. Well-defined specification and guidance but allow customization for different technologies. Fair, repeatable and verifiable.
- Standard Organizations
 - Problem: There’s no one fit-for-all solutions for different customers.
 - Solution: Flexible framework and proven-models of the new environment and trustable benchmark(s).

3 Big Data Benchmark - Big DS

3.1 The New Generation of Data Analytics Process

The evaluation of technology brings us the so called Software Defined Network/Data Center, and infrastructure of big data systems can definitely be considered as the most important one of them. Open source solutions liked Hadoop or traditionally BI technologies are merging their strengths together. Traditional BI vendors liked Teradata or HP Vertica starting to embed Hadoop as their data integration component or as a plugin. We can see more Hadoop or similar technologies vendors support better analytics features in their solutions. Cloudera Impala [7] or Apache Drill [13] are the top players in this area.

Figure 1 below shows one of the changes to the existing BI process defined based on what we observed from Hadoop, Spark and other systems. And we think this change will be a common case for those big data technology vendors.

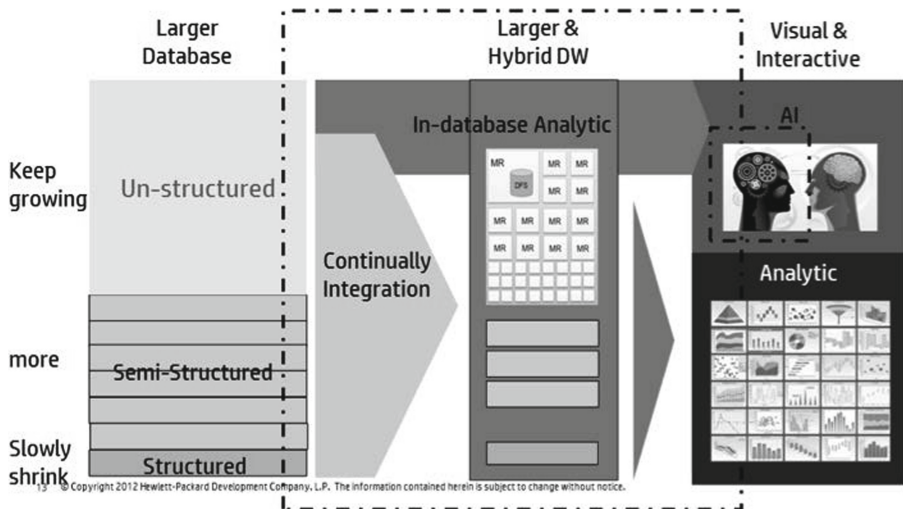


Fig. 1. Model of Modern Hybrid Big Data Platform (The dashed box includes the changes we consider for a big data benchmark.)

Figure 1 shows a model of big data analytics system. The whole environment is a hybrid big data system, where data is stored and analyzed in the same cluster.

On the left are the types of data that will be loaded, extracted, transformed and analyzed in the environment, including unstructured data, semi-structured data and structured data. To better simulate the real world big data analytics environment, the data should be more LIVE data or real time data. New data will be integrated continuously. And visualization tools or other machine learning tools will be used to directly analyze the live data and generate report or tabular data from the system.

On the right side, there is a smaller dashed box, which represents the machine learning will also be supported and executed in the analytics platform.

This is the environment to meet modern big data analytics requirements.

3.2 Starting from TPC-DS

TPC-DS is an evolution edition of TPC-H benchmark. Its predecessor is adopted widely in the database world. The benchmark resembles a decision support system. It covers basic procedures of a generic BI process.

TPC-DS is an existing industry standard benchmark, which is designed for bigger and more complex BI environments. The business model is a good sample of the real world workload. And the data scaling and workloads also match the real world BI process pretty well. From the big data perspective point of view, TPC-DS design meets the 3Vs (volume, velocity and variety) of Big Data. Founding Big DS on TPC-DS can meet most of the modern cooperation's Big Data requirements. This will also increase its acceptance in the industry as a big data benchmark.

The next section will describe how we extend TPC-DS to BigDS, and why we extend them.

3.3 BigDS

BigDS is an extension of TPC-DS for big data. To meet the big data requirements, the extension includes:

- Increase the data volume with modified business model
- Brand new data integration/ETL process
- Cloud benchmark based design for scale out environment
- Configurable benchmark in both data generation and workload execution

Figure 2 shows the modified business model to TPC-DS Business Model:

There are four domains in the new business model. They simulate the new architecture of retail business model with social network and mobile components.

The marketing domain simulates a new sales channel which is generated from social networks. Social networks change a lot to the internet. They changed and are changing customers' ways to use the Internet and how they spend their life in the network world. The most important thing, social networks open the door of big data. Customers start to log their life in the Internet. They share, talk and show huge amount

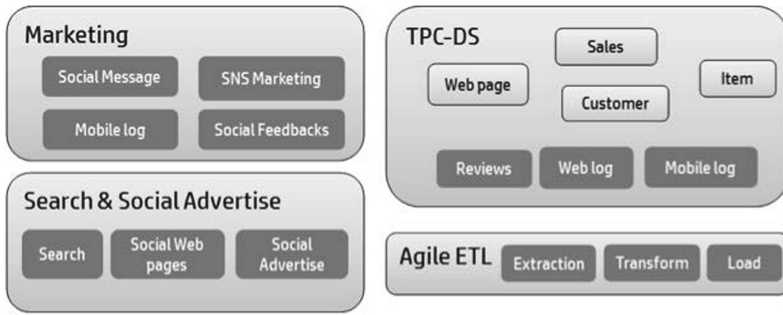


Fig. 2. Big DS Business Model Diagram

of information of themselves and the world around them in social network. This heavily changes the way how we understand our customers. So this is the most important change we make to the original business model. We keep this part as a new channel and it's different with the original web sales channel.

The second new domain is called Search and Social Advertisement domain. In this domain, we try to simulate the way that how we bring our products to our customer. To avoid conflict with the old web sales channel, it will be more focused on the advertisement of social network websites. Company with the social components, we added in the new social marketing and search components. They will simulate different user behaviors and actions which can drive to a final purchase.

The TPC-DS domain is almost the same as the old business model. To be able to know more about our customer and internal business efficiency, we add a few components to generate more data from existing components. The dark boxes are new added component in the diagram. They also represent for different kind of data structure, including unstructured data and semi-structured data.

The fourth domain is what we called Agile ETL part. This part will simulate how the data will be prepared and processed before entering the real analytics process. We will explain more in below section.

3.4 Modified Business Model

In upcoming years more and more companies will put part of their marketing activities on those social networking websites and mobile apps. Huge social networking websites like Facebook or Tencent will produce a huge amount of click stream data on their website, and this also brings a new way for customer to buy things. The original retail system design of TPC-DS has a well-defined data structure. We can easily extend the TPC-DS data schema with social networking and mobile parts. By doing this change, the data schema of BigDS will be largely increased, creating large opportunities for modeling a complex but more realistic big data analytics environment.

The data structure of social networking and mobile data are composite of structured data, unstructured data and semi-structured data, but a large portion of them are semi-structure data and unstructured data.

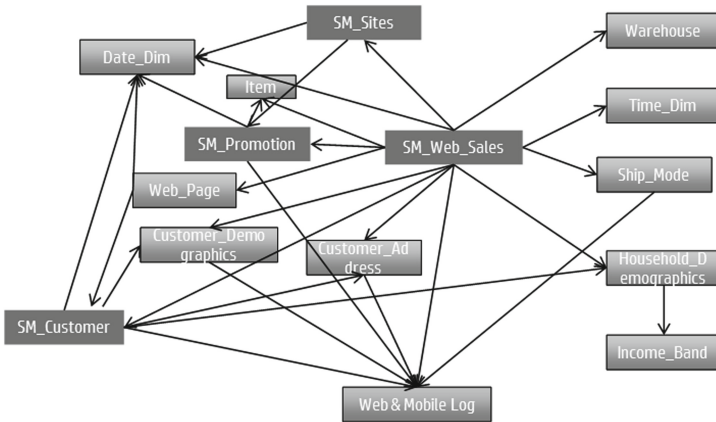


Fig. 3. Relationship of data model with social and mobile data

Figure 3 shows the changes we made to the old table model. The dark boxes are new tables.

They are the base tables of the new social sales channel, all other social and mobile tables will be added based on them. With these tables, we will be able to add more social customer behavior data in the future based on the micro-benchmarks we want to simulate.

The new added tables will also introduce some changes to existing tables, but those are all minor change. It won't change the original behaviors and business transactions of the TPC-DS business model.

3.5 Agile ETL – New Way of Data Integration

As we mentioned, modern big data analytics systems are more focused on live data analytics. So in this benchmark, the data integration process will be largely different compared to what TPC-DS uses. What we call Agile ETL is the way that we inject data into the analytics platform will have finer granularity and will be executed more frequently.

In the real world of big data analytics, analysts will observe and analyze the business, then decide to collect new data, and then develop new analysis component based on those data. New data collectors will be developed in hours, and the report or dashboard might also be developed in hours, or even shorter time. Analytics platform needs to handle the new data and start to analyze them quickly and easily. This is really different from the traditional BI process. A huge amount of data will be collected and loaded quickly. Some of the data extraction and transforming might be done in the same data warehouse in the big data cluster. The efficiency and effect of the new analysis will be known in a short time.

To simulate agile ETL processes well, we will categorize the data generator into different categories. Each category might be executed in different frequency so that they can better simulate the real world workloads:

- Data generator will be categorized based on the final purpose of report. Reports will be defined as real time report, hourly report, daily report, weekly report and monthly report. Different report will be executed in different frequency.
- Data generator will be also categorized based on their data size and data structure. Bigger data might cause more system resource consumption. In real environment, they will be executed at night. In our benchmark, there's also a "night" time designed.

As what TPC-DS does, Agile ETL component will be also considered as part of the workload, and its performance will be counted into the final score of performance metrics.

3.6 Other Considerations

Scalability. In most cases, big data analytics platform are scale-out environments, besides knowing the peak performance of the environment, knowing how well the environment can scale is also an important metric for benchmark user. So BigDS will have multiple metrics, scalability metrics will be one of them.

While designing the scalability metrics, we want to leverage other good benchmark examples. There are many other great benchmarks already available on the world that has many great characters. SPECpower_2008, SPECjbb2005, SPECjbb2013 and SPECvirt_sc2010 are good examples of them. In a big data analytics platform, we think these two metrics can show the scalability of an environment:

- How big the data platform can handle?
- How many jobs the platform can handle?

As a big data benchmark, in most cases, our environment is big can might be easily saturated with small workload. By adopting the idea of benchmarks we mentioned above liked SPECvirt_sc2010, we will use similar idea in BigDS.

We will use a concept called "Copy" in BigDS. While designing BigDS, there is a base micro-benchmark set (the green box in Fig. 4) and a base set of data generators (the blue box in Fig. 4). To saturate the cluster, the base set of data generator and workload can be executed with multiple copies at the same time. How many copies can be executed in the cluster will be the scalability metrics of BigDS. While adding data generators and micro-benchmarks, all of them will not impact each other and can be executed in parallel.

The base copy of micro-benchmarks is composed with three different kind of workloads:

- Ported TPC-DS queries
- SQL liked queries, i.e. Hive [6] Queries
- Analytics jobs, i.e. Map Reduce jobs or other machine learning jobs.

Figure 4 diagram shows one of ways of BigDS execution. In that way, we will generate data first, and then executed the base set of micro-benchmark. To know the

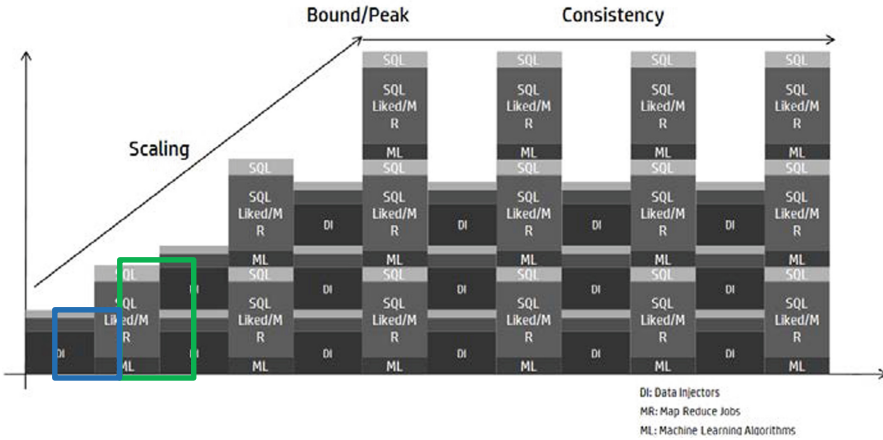


Fig. 4. Workload Scalability Test Pattern (Color figure online)

scalability, we will then increase the copy number of workloads step by step until the copy number we defined.

There’s another way that we will execute the workload – by the frequency of the report. The workload will be executed using the similar way as below but more phases. The exactly phases will be determined after all micro-benchmarks are characterized.

Configurable Benchmark. Customer data might come from different sources. We designed different structured of data in BigDS, and will develop different kind of micro-benchmarks. But those micro-benchmarks might not be fixed for all BigDS users. So in our future design, user will be able to enable or disable a certain micro-benchmark if you only tend to use BigDS as a performance tool to know performance of your environment. If users want to compare the result with others, then it needs to run fully list of all defined micro-benchmarks with required execution scenario.

4 Summary

BigDS is an experiment for evaluating the performance and scalability of big data platform. We define some characteristics of big data analytics platforms so that we can have a more fixed and common configuration from thousands of big data technologies and vendors.

The design leverages some great ideas from the existing benchmark TPC-DS [1], BigBench [11], SPECvirt_2010sc [2], SPECjbb2005 [3], etc., adding many attributes of real big data environment which are essential for most of modern big data business environment. In this paper, we define the data model, workload characters and workload execution process of typical big data analytics platform.

The benchmark is still under development. Ongoing work includes adding more micro-benchmarks and corresponding data generator, and pick up a best way which can really measure the performance and scalability of a big data analytics platform.

References

1. TPC. TPC is a trademark of the Transaction Processing Performance Council. TPC-H and TPC-DS are the decision support benchmarks of TPC organization. <http://www.tpc.org>
2. SPEC. SPEC is a trademark of the Standard Performance Evaluation Corporation 1995–2014. SPECjbb2005 is the server side Java Benchmark of SPEC.org. SPECjbb2013 is the evaluation version of SPECjbb2005. SPECvirt_2010sc is the server consolidation virtualization benchmark of SPEC.org. <http://www.spec.org>
3. TeraSort. Refer to the Apache Terasort benchmark, which is a MapReduce version of Sort benchmark
4. SWIM. SWIM stands for Statistical Workload Injector for MapReduce. The synthesis methodology is adopted in BigDS and it's supporting toolset
5. Apache Hadoop and it's related projects. Apache Hadoop is an open-source software framework for storage and large-scale processing of data-sets on clusters of commodity hardware. Hadoop is an Apache top-level project being built and used by a global community of contributors and users. It is licensed under the Apache License 2.0
6. Apache Hive. Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. [1] While initially developed by Facebook
7. Cloudera Impala. Cloudera Impala is Cloudera's open source massively parallel processing (MPP) SQL query engine for data stored in a computer cluster running Apache Hadoop. <http://www.cloudera.com/content/cloudera/en/products-and-services/cdh/impala.html>
8. Google BigTable. Refer to Google's BigTable paper. <http://research.google.com/archive/bigtable-osdi06.pdf>
9. Huppler, K.: Chairman TPC. The author of "The art of building a good benchmark" (2009). <http://www.tpc.org/tpctc/tpctc2009/tpctc2009-03.pdf>
10. WBDB, Workshop of Big Data Benchmarking, San Jose. <http://clds.ucsd.edu/wbdb2012>
11. Big Bench. Extend TPC-DS specification to include unstructured and semi-structured data; modify the TPC-DS. In: A data model for BigBench was proposed in the First WBDB Workshop by Ghazal (2012)
12. Deep Analytic Pipeline. A Benchmark Proposal by Milind Bhandarkar (Pivotal Chief Scientist), (2013). <http://clds.sdsc.edu/sites/clds.sdsc.edu/files/2013-03-07-DeepAnalyticsPipeline.pdf>
13. Apache Drill Project. Apache Drill is an open-source software framework that supports data-intensive distributed applications for interactive analysis of large-scale datasets. Drill is the open source version of Google's Dremel system which is available as an IaaS service called Google BigQuery. <http://incubator.apache.org/drill/>
14. Google Dremel. <http://research.google.com/pubs/pub36632.html>
15. Google Big Query. <https://developers.google.com/bigquery/>