# The Vision of BigBench 2.0

Tilmann Rabl
University of Toronto
Toronto, Canada
tilmann.rabl@utoronto.ca

Michael Frank, Manuel Danisch
bankmark
Passau, Germany
firstname.lastname@bankmark.de

Hans-Arno Jacobsen
Middleware Systems Research Group
jacobsen@eecg.toronto.edu

Bhaskar Gowda
Intel
Hillsbury, Oregon, USA
bhaskar.d.gowda@intel.com

## ABSTRACT

Data is one of the most important resources for modern enterprises. Better analytics allow for a better understanding of customer requirements and market dynamics. The more data is collected, the more information can be extracted. However, information value extraction is limited by data processing speeds. Due to fast technological advances in big data management there is an abundance of big data systems. This leaves users in the dilemma of choosing a system that features good end-to-end performance for the use case. To get a good understanding of the actual performance of a system, realistic application level workloads are required.

To this end, we have developed BigBench, an application level benchmark focused only on big data analytics. In this paper, we present the vision of BigBench 2.0, a suite of benchmarks for all major aspects of big data processing in common business use cases. Unlike other efforts, BigBench 2.0 will have completely consistent and integrated model and workload, which will allow realistic end-to-end benchmarking of big data systems.

## 1. INTRODUCTION

In recent years, big data has become one of the driving factors of innovation. Many problems that seemed close to unsolvable not too long ago become easy using enough input data and statistical methods. The increasing capabilities in collecting ever larger amounts of data have created a lively ecosystem of all kinds of systems for big data processing.

Due to the lack of standards users of big data systems have a difficult time comparing deployments. This is not unlike the early days of the emergence of relational database systems, when standard consortia like the Transaction Processing Performance Council (TPC) were founded to offer objective comparisons. The TPC was successful over the years to offer a solution to the users' dilemma in form of industry standard benchmarks, which are agreed upon by many vendors. Widely used workloads and benchmarks are two of the driving factors of standardization. Due to the complexity and variety of big data applications and use cases, big data benchmarking is a broad and complex field. Many different kind of benchmarks have been proposed. Examples are micro-benchmarks for individual file system operations and functional benchmarks for sorting and counting (e.g., [9]). However, there is limited work on application level benchmarks, which give insights in the actual performance of a given deployment for real workloads. In [8], we have proposed BigBench, an application level, end-to-end big data *analytics* benchmark. BigBench was fully implemented and is widely used to extensively test big data systems. Based on the discussion with many industry experts, we have identified several key opportunities to improve BigBench [2]. In this paper, we summarize the vision of the next version of BigBench, BigBench 2.0. The main contribution of BigBench 2.0 is the integration of many common big data processing aspects into a consistent and realistic use case. While BigBench 1.0 has achieved this for batch big data analytics on structured, semi-structured, and unstructured data, BigBench 2.0 aims at integrating more data types, data velocities, and processing models.

The rest of this paper is structured as follows, in Section 2, we give a brief introduction to BigBench 1.0. We discuss and motivate all major envisioned parts of BigBench 2.0 in Section 3. In Section 4, we give an outlook on the implementation before discussing major challenges in the proposal in Section 5 and concluding in Section 6.

## 2. BIGBENCH 1.0

The idea of the BigBench 1.0 data model was first presented at the first Workshop on Big Data Benchmarking in 2012[1]. It was based on TPC-DS, the TPC's latest decision support benchmark [14]. Based on the discussions at the workshop, a small group formed that specified the first version of BigBench [8]. BigBench 1.0 inherited the retail business model from TPC-DS and the workload was designed around this. By studying customers' workloads and current market research [11] a big data analytics workload comprising 30 queries was specified. One third of the queries was taken directly from TPC-DS, the rest was created based on the study. The data set was adjusted to include unstructured data in form of product reviews and semi-structured


*DanaC* '15, May 31 - June 04 2015, Melbourne, VIC, Australia

http://dx.doi.org/10.1145/2799562.2799642 ...$15.00.

[1] WBDB2012 - http://clds.sdsc.edu/wbdb2012

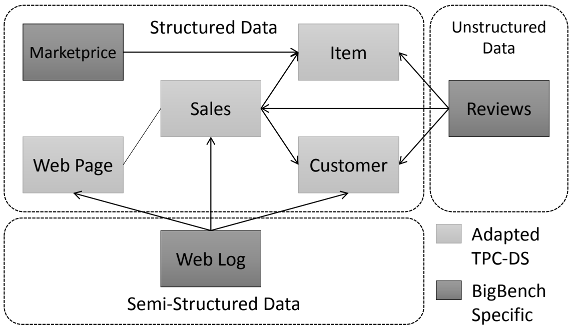


**Figure 1: BigBench 1.0 data set overview**

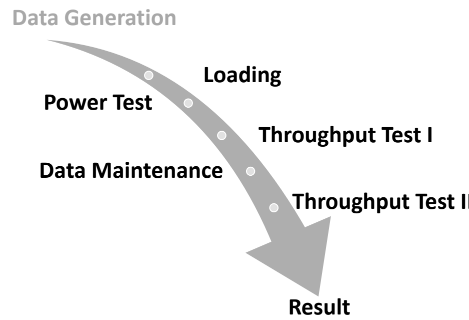


**Figure 2: BigBench 1.0 benchmark process**

data in terms of click streams. An overview of the data set can be seen in Figure 1. BigBench was designed as a system independent benchmark, however, in order to experiment with the workload and data set an initial prototype was implemented in Teradata Aster in 2012 and a reference implementation was completed on top of the Hadoop framework in 2014[2]. Currently, BigBench is evaluated by the TPC as the first application level, end-to-end, industry standard big data benchmark.

Even though BigBench 1.0 captures big data analytics in retail business cases, it does not give a complete picture of big data management. The reason is that big data processing starts before the analytic processing and, therefore, is often done in form of data processing pipelines [3]. In contrast to this, BigBench 1.0 is executed in a batch processing style. This is also reflected in the full benchmark process depicted in Figure 2. After the data generation, which is not measured, data is loaded and then two types of analytic workloads are run: a power test and two throughput tests. The power test executes all 30 queries serially and measures the individual latency of each query. The throughput tests run a user defined number of parallel streams of the 30 queries. Between the first and second throughput test a maintenance is performed, which updates 1% of the data. The final result is the query throughput, which is based on the average latency per query. To capture the full processing pipeline and to address the advances in big data processing, which result in ever new use cases and thus workloads, we are designing BigBench 2.0.

## 3. BIGBENCH 2.0

BigBench 1.0 is a big data *analytics* benchmark. BigBench 2.0 aims at covering the complete big data pipeline in the retail business model. An overview of the vision of BigBench 2.0 can be seen in Figure 3. The BigBench 1.0 part in the extended benchmark can be seen at the right upper side of the figure. Furthermore, the 2.0 proposal includes streaming, key-value processing, graph processing, ETL requirements, and multimedia data types. The analytics part is extended, it includes more machine learning tasks and more procedural tasks. Another extension is visualization workloads. These typically result from the use of big data visualization products that create many queries resulting from data scientists exploring data characteristics. In many data centers, visualization workloads are a major part of the overall workload. In the following, we give details on the individual modules and their connections.

[2]Available at `https://github.com/intel-hadoop/Big-Data-Benchmark-for-Big-Bench`

### 3.1 Stream Processing

The stream processing module will feature several streaming applications that will be performed on click stream data. Examples will include simple aggregations and filtering. More complex workloads could be added if they are identified in the use case. Emerging use cases in this area are streaming machine learning tasks like classification and clustering. The stream processing module is designed to be run as part of the end-to-end workload, but it can also be used as an individual benchmark for stream processing systems or any type of system with comparable processing capabilities.

Like for the initial BigBench proposal, we will consult previous benchmark proposals, but make sure that the workload is system independent and is fully integrated in the business model. An example of a stream benchmark is the Linear Road Benchmark [1]. Similar to our proposed module, the benchmark features relatively simple queries. Another, more recent proposal is StreamBench [10], which also specifies simple queries in form of aggregation and filtering. The results of the stream processing model will (partially) be fed into the key-value processing module.

### 3.2 Key-Value Processing

Like the stream processing module, the key-value store (KVS) module will also be usable as an individual benchmark. The operations will follow the simple Create, Read, Update, Delete (CRUD) structure. This means the workload will not contain complex operations like joins or aggregations. However, the data will be complex and the workload will have operations against multiple tables with different ingest rates and with different access patterns.

As can be seen in Figure 3, the main data source for the KVS is the stream processing module. Consequently, the application for KVS will also include click streams. Other projected applications include user management and shopping baskets.

Stream processing and key-value processing will require a new form of driver, since the workload generation requires substantial hardware resources unlike the long running complex analytics workloads. Therefore, he reference kit will be constructed similarly to the popular YCSB suite [7] and its extensions. The data set will however, be adjusted to the

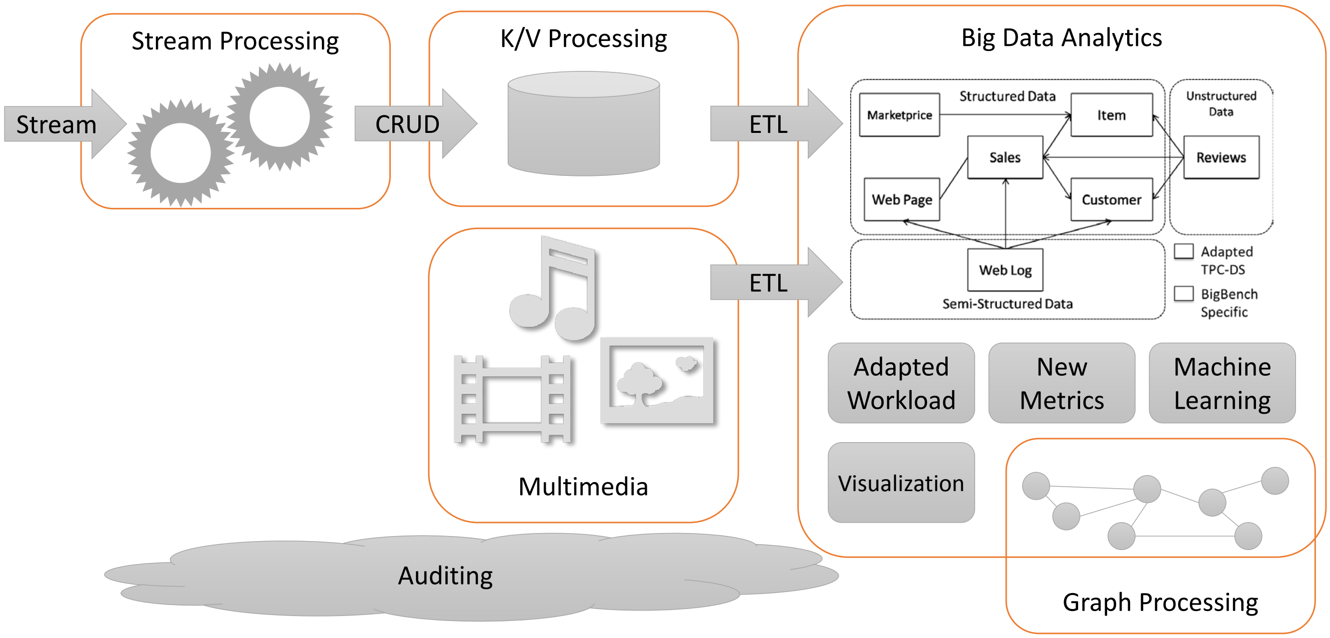


**Figure 3: BigBench 2.0 overview**

rest of the benchmark model and will comprise complex, realistic data types.

## 3.3 Big Data Analytics

The big data analytics part is extended in BigBench 2.0. The module covers multiple workloads, namely machine learning, procedural tasks, and declarative SQL-like workloads. Given that these jobs are typically performed by specialized libraries and systems, the different categories will be available as separate workloads. The analytics part already presents a substantial amount of work for the system under test in BigBench 1.0, therefore, the total number of queries will not be increased drastically.

However, the machine learning part, which in the BigBench 1.0 kit mainly comprises 5 Mahout queries, will be substantially transformed to include larger input data sets and more machine learning tasks. In terms of specification the machine learning tasks will become more abstract and not require a certain algorithm. Rather, they will be specified as general machine learning concepts, e.g., the workload will include clustering tasks, rather than k-means tasks.

The procedural workload, which contains tasks that are typically performed in a map reduce style system or in user defined functions in databases will be slightly extended to reflect real world workloads. Potential workloads include search indexes construction and cross-correlation.

## 3.4 Graph Analytics

The omission of graph analytics was a major point of critique in BigBench 1.0. Therefore, BigBench 2.0 will contain a graph analytics module included in the analytics part. Potential candidates are PageRank workloads as well as social graphs. The graph analytics part will be inspired by the Graph500 benchmarking suite , the Linked Data Benchmarking Council's benchmarks [13, 4], as well as the recent proposal by Capota et al, Graphalytics [5]. Unlike previous proposals, the graph analytics part of BigBench 2.0 will be integrated with the other analytical tasks. We intend to also create tasks that span not only the graph part but also structured and unstructured elements.

## 3.5 Multimedia

A recent trend is the analysis of non textual data like images and audio. This can be face or voice recognition, feature extraction, and image search. In the retail model, for example, filtering of inappropriate user generated content or product counterfeiting are a relevant workloads. The inclusion of multimedia data will also drastically increase the data size and, thus, put more stress on the I/O system of the system under test. Multimedia analysis also requires additional machine learning techniques. Relevant workloads exist in the multimedia community and several benchmarks and challenges are available as example application [12, 16]

## 3.6 Other Features

Besides extension of the workload and data set, BigBench 2.0 will include many additional features. Emphasis will be put on auditing tools. To enable benchmark users to verify their runs, additional procedures will be included in the driver that verify data integrity, compare the result sets, measure thresholds for the accuracy of machine learning tasks. Like the first kit, the driver will be completely modular and enable easy replacement of certain components (e.g., using different subsystems for streaming or for individual analytical tasks).

# 4. IMPLEMENTATION

Although the benchmark is not specific to a certain system, we will provide a complete reference kit as for BigBench 1.0. The implementation choices of the kit will be mostly pragmatic. All processing components of the kit will be open-source software.

The BigBench 1.0 kit was completely built on top of the Apache Hadoop stack [6]. Since the start of the implementation ever new systems have been developed, many of which with a much higher efficiency and better performance than plain Hadoop. In the BigBench 2.0 implementation, we will put some effort in choosing performant components for the kit, since this is also a major concern for the BigBench 1.0 version. Like the first version, the kit will be open to make it easy to replace parts with different software components.

## 5. CHALLENGES

We see several interesting challenges in the new direction of the BigBench proposal. The first one is the complexity problem. Running big data workloads is a time consuming task. In industry standard benchmarking, this is intensified since many runs are required to ensure all systems work properly and efficiently. Users of BigBench 1.0 report run times of multiple days for TB scale data sizes. With richer workloads and more components run times are likely to increase. Therefore, we have to ensure that the overall process stays manageable in terms of time consumption and does not include unnecessary repetition.

A second challenge is balancing. Since we will have vastly different types of workloads the balance between the types of operations needs to be carefully selected. Otherwise one type of workload will completely dominate the others. Furthermore, the different types of processing require different types of client machines. While a single client machine is enough to fully utilize a large scale cluster running BigBench 1.0, key-value stores, for example, can sustain a number of clients that is in the order of the number of server machines.

In terms of implementation, the consistency of different parts of the data set and workload is challenging. Using the Parallel Data Generation Framework [15], we can solve this for the data set, but additional functionality will be required for the workload generation. The workload and data set will also require additional characteristics that can be mined in the machine learning tasks. An interesting requirement in this context is verifiability of the query results. Because we do not want to dictate the machine learning algorithms, we need to set minimum thresholds for the accuracy of a machine learning task. The thresholds have to be automatically adjusted for different data sizes.

## 6. CONCLUSION

In this paper, we have presented the vision of BigBench 2.0. It is designed to fill the gap of a suite of application level big data benchmarks as well as a complete application level, end-to-end big data pipeline benchmark. Unlike previous efforts, the BigBench 2.0 workload and data set will be fully integrated and form a consistent and realistic model of retail big data use cases. This makes the benchmark results meaningful for customers, sales people, testers, and performance engineers. To form this vision, many domain experts from industry and academia have contributed [2].

## 7. REFERENCES


[1] A. Arasu, M. Cherniack, E. Galvez, D. Maier, A. Maskey, E. Ryvkina, M. Stonebraker, and R. Tibbetts. Linear Road: A Stream Data Management Benchmark. In *VLDB*, 2004.

[2] C. Baru, M. Bhandarkar, C. Curino, M. Danisch, M. Frank, B. Gowda, H.-A. Jacobsen, H. Jie, D. Kumar, R. Nambiar, M. Poess, F. Raab, T. Rabl, N. Ravi, K. Sachs, S. Sen, L. Yi, and C. Youn. Discussion of BigBench: A Proposed Industry Standard Performance Benchmark for Big Data. In *TPCTC*, pages 44–63, 2014.

[3] C. Baru, M. Bhandarkar, R. Nambiar, M. Poess, , and T. Rabl. Benchmarking Big Data Systems and the BigData Top100 List. *Big Data*, 1(1):60–64, 2013.

[4] P. Boncz, I. Fundulaki, A. Gubichev, J. Larriba-Pey, and T. Neumann. The Linked Data Benchmark Council Project. `http://oai.cwi.nl/oai/asset/21444/21444B.pdf`, 2013.

[5] M. Capota, T. Hegeman, A. Iosup, A. Prat-Perez, O. Erling, and P. Boncz. Graphalytics: A Big Data Benchmark for Graph-Processing Platforms. In *GRADES*, 2015.

[6] B. Chowdhury, T. Rabl, P. Saadatpanah, J. Du, and H.-A. Jacobsen. A BigBench Implementation in the Hadoop Ecosystem. In *WBDB*, 2013.

[7] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears. Benchmarking Cloud Serving Systems with YCSB. In *SoCC*, pages 143–154, 2010.

[8] A. Ghazal, T. Rabl, M. Hu, F. Raab, M. Poess, A. Crolotte, and H.-A. Jacobsen. BigBench: Towards an Industry Standard Benchmark for Big Data Analytics. In *SIGMOD*, 2013.

[9] J. Gray. GraySort Benchmark. Sort Benchmark Home Page – `http://sortbenchmark.org`.

[10] R. Lu, G. Wu, B. Xie, and J. Hu. Stream Bench: Towards Benchmarking Modern Distributed Stream Computing Frameworks. In *UCC*, pages 69–78, Dec 2014.

[11] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The Next Frontier for Innovation, Competition, and Productivity. Technical report, McKinsey Global Institute, 2011. `http://www.mckinsey.com/insights/mgi/research/technology_and_innovation/big_data_the_next_frontier_for_innovation`.

[12] S. Marchand-Maillet and M. Worring. Benchmarking Image and Video Retrieval: an Overview. In *Proceedings of the 8 $^{th}$ ACM international Workshop on Multimedia information Retrieval*, pages 297–300, Santa Barbara, California, USA, 2006.

[13] R. C. Murphy, K. B. Wheeler, B. W. Barrett, and J. A. Ang. Introducing the Graph 500. *Cray User's Group (CUG)*, 2010.

[14] M. Pöss, R. O. Nambiar, and D. Walrath. Why You Should Run TPC-DS: A Workload Analysis. In *VLDB*, pages 1138–1149, 2007.

[15] T. Rabl, M. Frank, H. M. Sergieh, and H. Kosch. A Data Generator for Cloud-Scale Benchmarking. In *TPCTC*, pages 41–56, 2010.

[16] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVid. In *SIGMM MIR*, pages 321–330, 2006.