Setting the Direction for Big Data Benchmark Standards

Chaitanya Baru[1], Milind Bhandarkar[2], Raghunath Nambiar[3], Meikel Poess[4], and Tilmann Rabl[5]

[1] San Diego Supercomputer Center, UC San Diego, USA
baru@sdsc.edu
[2] Greenplum/EMC, USA
Milind.Bhandarkar@emc.com
[3] Cisco Systems, Inc, USA
rnambiar@cisco.com
[4] Oracle Corporation, USA
meikel.poess@oracle.com
[5] University of Toronto
tilmann.rabl@utoronto.ca

Abstract. The *Workshop on Big Data Benchmarking (WBDB2012),* held on May 8-9, 2012 in San Jose, CA, served as an incubator for several promising approaches to define a big data benchmark standard for industry. Through an open forum for discussions on a number of issues related to big data benchmarking—including definitions of big data terms, benchmark processes and auditing — the attendees were able to extend their own view of big data benchmarking as well as communicate their own ideas, which ultimately led to the formation of small working groups to continue collaborative work in this area. In this paper, we summarize the discussions and outcomes from this first workshop, which was attended by about 60 invitees representing 45 different organizations, including industry and academia. Workshop attendees were selected based on their experience and expertise in the areas of management of big data, database systems, performance benchmarking, and big data applications. There was consensus among participants about both the need and the opportunity for defining benchmarks to capture the end-to-end aspects of big data applications. Following the model of TPC benchmarks, it was felt that big data benchmarks should not only include metrics for performance, but also price/performance, along with a sound foundation for fair comparison through audit mechanisms. Additionally, the benchmarks should consider several costs relevant to big data systems including total cost of acquisition, setup cost, and the total cost of ownership, including energy cost. The second Workshop on Big Data Benchmarking will be held in December 2012 in Pune, India, and the third meeting is being planned for July 2013 in Xi'an, China.

Keywords: Big Data, Benchmarking, Industry Standards.

1 Introduction

The world has been in the midst of an extraordinary information explosion over the past decade, punctuated by the rapid growth in the use of the Internet and in the

R. Nambiar and M. Poess (Eds.): TPCTC 2012, LNCS 7755, pp. 197–208, 2013.

number of connected devices worldwide. The data growth phenomenon is global in nature, with Asia rapidly emerging as a major user base contributing to both consumption as well as generation of data, as indicated in Figure 1. A 2010 IDC study [9] estimates the total amount of enterprise data to grow from about 0.5 zettabyte in 2008 to 35 zettabytes in 2020. Indeed, the rate at which data and information are being generated is faster than at any point throughout history. With the penetration of data-driven computing, web and mobile technologies, and enterprise computing, the emerging markets have the potential for further adding to this already rapid growth in data. Data from all sources—from enterprise applications to machine-generated data—continue to grow exponentially, requiring the development of innovative techniques for data management, data processing, and analytics. This motivates the development of evaluation schemes and benchmark standards encompassing hardware and software technologies and products.

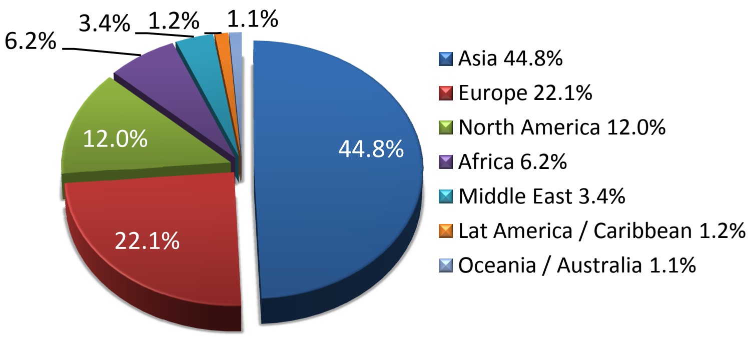

Fig. 1. Internet users in the world distributed by regions 2011 [2]

Evaluating alternative technological approaches and assessing the effectiveness of applications and analytic pipelines designed to tackle the challenges of big data require new approaches to benchmarking, especially since current industry standard benchmarks are not designed to cover the most important aspects of big data. Big data is often distinguished from traditional large databases using the three Vs: *volume*, *variety* and *velocity*. In addition, a big data benchmark may also include a fourth V, namely value.

Big data systems deal with large volumes of data, which are sometimes in the multiple petabyte range. Traditional large-scale industry standard benchmarks, such as TPC-H and TPC-DS, only test systems up to 100 Terabyte. Variety in the context of big data systems refers to the ability to deal with differently organized data, such as unstructured, semi-structured and structured data. Current industry standard benchmarks focus on structured data, mostly relational data. Velocity refers to the ability of a big data system to stay in synchronization with operational systems through periodic refreshes, commonly referred to as extraction, transformation and load (ETL) or data integration (DI). While some of the newer industry standard benchmarks, e.g. TPC-DS, include a periodic refresh process and while their refresh methodology, i.e. concurrent updates, is realistic, they do not implement the same scale and frequency at which data is refreshed in big data applications. Finally, "value" refers to big data processing that creates business value to the customer. The benchmarks should be modeled after real-world processing pipelines that create value to the end user.

A big data benchmark must provide objective measures quantifying performance, scalability, elasticity, and price/performance of any system designed to support big data applications. Such a benchmark would facilitate evaluation of alternative solutions and provide for comparisons among different solution approaches. It would also characterize the new feature sets, enormous data sizes, and shifting loads of big data applications, and the large-scale and evolving system configurations and heterogeneous technologies of big data platforms.

The first *Workshop on Big Data Benchmarking (WBDB2012)* held on May 8-9, 2012 in San Jose, CA [7] served as an important incubator towards the development of an industry standard for big data benchmarking. The objective of *WBDB2012* was to identify key issues and launch an activity around the definition of reference benchmarks to capture the essence of big data application scenarios.

Workshop invitees were drawn from academia and industry, and included practitioners as well as researchers with backgrounds in big data, database systems, benchmarking and system performance, cloud storage and computing, and related areas. Each attendee was required to submit a two-page abstract and provide a five minutes "lightning talk". The workshop website (http://clds.sdsc.edu/wbdb2012) provides copies of papers and presentations. The presentations were classified into four categories: *benchmark properties, benchmark process*, *hardware and software aspects,* and *data generation* for big data workloads.

1.1 Workshop Description

A total of about 60 invited attendees represented about 45 different organizations at the workshop[1]. Each day began with three 15-minute introductory talks, followed by presentations in the morning and discussions in the afternoon. The introductory talks on the first day provided an overview of industry benchmarking efforts and standards and discussed desirable attributes and properties of competitive industry benchmarks. On the second day, the talks focused on big data applications and the different genres of big data, such as genomic and geospatial data, and big data generation. The opening presentations were followed by about twenty "lightning talks" of 5-minutes each by the invited attendees. For the afternoon sessions, the attendees were divided into two equal groups, both groups were asked to discuss the same set of topics and report results at a plenary session at the end of the day.

[1] See http://clds.sdsc.edu/wbdb2012/participants for a list of participants.
See http://clds.sdsc.edu/wbdb2012/organizers for a list of organizers.

The rest of this paper summarizes the discussions and findings from the workshop. Section 2 covers the benchmarking context and topics related to the nature of big data and big data applications, and existing big data benchmark efforts; Section 3 discusses guiding principles for the design of big data benchmarks; Section 4 discusses objectives of big data benchmarking, specifically whether such benchmarks should be targeted to encourage technological innovation or primarily for vendor competition; Section 5 probes some of the details related to big data benchmarks; and Section 6 provides conclusions from the workshop discussions and points to next steps in the process.

2 Benchmark Context

2.1 Application-Level Benchmarking

Workshop attendees were in general agreement that a big data benchmarking activity should begin at the end application level, by attempting to characterize the end-to-end needs and requirements of big data analytic pipelines. While isolating individual steps in such pipelines, e.g. sorting, is indeed of interest, it should be done in the context of the broader application scenario.

2.2 Data Genres and Application Scenarios

A range of data genres should be considered for big data benchmarks including, for example, structured, semi-structured, and unstructured data; graphs (including different types of graphs that might occur in different types of application domains, e.g. social networking versus biological networks); streams; geospatial data; array-based data; and special data types such as genomic data. The core set of operations need to be identified, modeled, and benchmarked for each genre, while also seeking similarities across genres.

It may be feasible to identify relevant application scenarios involving a variety of data genres that require a range of big data processing capabilities. An example discussed at the workshop was data management for an Internet-scale business, for example, an enterprise similar to, say, Facebook or Netflix. A plausible use case for such an application can be constructed requiring big data capabilities for managing data streams (click streams), weblogs, text sorting and indexing, graph construction and traversals, as well as geospatial data processing and structured data processing.

At the same time, the workshop attendees agreed that a single application scenario may not realistically capture the full range of data genres and operations that are broadly relevant across a wide range of big data applications. This may necessitate the development of multiple benchmark definitions based on differing scenarios. These benchmarks together would then capture a comprehensive range of variations.

2.3 Learning from Successful Benchmarks

Fortunately, there are a number of examples of successful benchmarking efforts that we can learn from and leverage. These include benchmarks developed by industry consortia such as the Transaction Processing Council (TPC) and Standard Performance Evaluation Corporation (SPEC); benchmarks from industry-driven efforts such as VMMark (VMWare) and Top500; and, benchmarks like Terasort [10] and Graph500 [13] designed for specific operations and/or data genres. Can a new big data benchmark be defined by simply building upon and extending current benchmark definitions? While this may be possible, a number of issues need to be considered such as whether:

- The existing benchmarks model application scenarios relevant to big data;
- The existing benchmarks can be naturally and easily scaled to the large data volumes necessary for big data benchmarking;
- Such benchmarks can be used more or less “as is”, without requiring significant re-engineering to produce data and queries (operations) with the right set of characteristics for big data applications; and
- The benchmarks have no inherent restrictions or limitations such as, say, requiring all queries to be executed in SQL.

Several existing benchmarking efforts were presented and discussed at the meeting such as the Statistical Workload Injector for MapReduce (SWIM) developed at the University of California, Berkeley [4], GridMix3, developed at Yahoo! [1], YCSB++, developed at the Carnegie Mellon University based on YCSB of Yahoo! [15], and TPC-DS, the latest addition to TPC’s suite of decision support benchmarks [4,16,17].

3 Design Principles for Big Data Benchmarks

As mentioned, several benchmarks have gained acceptance and are commonly used, including the ones from TPC (e.g., TPC-C, TPC-H [16]), SPEC, and Top500. Some of these benchmarks are impressive in their longevity – TPC-C is almost 25 years old and the Top500 list is just celebrating 20 years – and continue to be used. The workshop discussions focused on features that may have contributed to the longevity of the popular benchmarks. For example, in the case of the Top500, the metric is simple to understand: the result is a simple rank ordering according to a single performance number (FLOPS).

TPC-C, which models an on-line transaction processing (OLTP) workload, possesses the characteristics of all TPC benchmarks, (i) it models an application domain; (ii) employs strict rules for disclosure and publication of results; (iii) uses third-party auditing of results; and (iv) publishes performance as well as price/performance metrics. TPC-C requires that as the performance number increases (i.e. the transactions/minute) the size of the database (i.e. the number of warehouses in the reference database) must also increase. TPC-C requires a new warehouse to be introduced for every 12.5 tpmC. Thus, one cannot produce extremely high transactions/minute numbers while keeping the database fixed at some arbitrarily small database size. This "self-scaling" nature of the benchmark, which may well have contributed to the longevity of TPC-C itself, was recognized as a strength and a key desirable feature of any big data benchmark as well.

Other benchmarks, such as TPC-H, specify fixed, discrete "scale factors" at which the benchmark runs are measured. The advantage of that approach is that there are multiple, directly comparable results at a given scale factor. Arguably, one of the characteristics of the *systems under test (SUT)* in TPC benchmarking is that they tend to be "over-specified" in terms of their hardware configuration. Vendors (*aka* benchmark sponsors) are willing to incur a higher total system cost in order to obtain better performance numbers without much of a negative impact on the price/performance numbers. For example, the SUT can employ 10x the amount of disk for a given benchmark database size, whereas real customer installations will only employ 3-4x the amount of disk. In the case of big data benchmarking, there is the distinct possibility that the SUT is actually *smaller* in overall size and configuration than the actual customer installation, given the expense of assembling a big data system for benchmarking and the rate at which enterprise systems are growing. It may, therefore, become necessary to extrapolate (scale up) system performance based on measured results at a smaller scale. There was discussion on whether, and how well, results at one scale factor could be extrapolated to obtain/infer performance at a different scale factor. A concern was that it is typically not possible to extrapolate performance numbers obtained on small systems running small data sets to performance of large systems running large data sets. However, while such extrapolation may be difficult to achieve across a broad range of scale factors, could it be achieved among neighboring scale factor values? Could a result published at a given scale factor be accompanied by information on the "scalability" of the result for data sizes in the neighborhood of that scale factor, specifying the range of data sizes around that scale factor for which this result can be extrapolated? Facilitating such extrapolation of results may require a more extensive set of system performance and system configuration information to be recorded. Thus, while there are arguments for simplicity of the benchmark metrics, e.g. publishing only one or very few numbers as the overall result of the benchmark, more detailed information may indeed be needed to facilitate extrapolation of performance numbers across a range of data sizes.

A strong motivation for extrapolation is the significant costs involved in running big data benchmarks. The very size and nature of the problem requires large installations and significant amounts of preparation and effort. As a pragmatic matter, the benchmark should not be expensive to run, implying that the end-to-end process should be relatively easy and simple. This can be facilitated by the existence of robust, well-tested programs for data generation; robust scripts for running the tests; perhaps, available implementations of the benchmark in alternative technologies, e.g. RDBMS and Hadoop; and an easy method by which to verify the correctness of benchmarks results.

Other key aspects to consider in the benchmarking exercise are *elasticity* and *durability*, viz., the ability to gracefully handle failures. How well does the system perform under dynamic conditions, for example, when the amount of data is increased; when more resources (e.g. nodes) are added to the system; and when some resource are removed from the system as a consequence of a failure, e.g. node or disk failure? While TPC benchmarks require atomicity, consistency, isolation, and durability (ACID) tests to be performed with the SUT, these are performed as standalone tests, outside the window during which performance measurements are made. For big data systems, elasticity and durability need to be intrinsic to the system and, thus, they need to be part of the overall performance test. Elasticity requires that a system be able to utilize and exploit more resources as they become available. Durability ensures that a big data system can continue to function even in the presence of certain types of system failures.

Finally, the benchmark specification should be technology agnostic as much as possible. The applications, reference data, and workload should be specified at a level of abstraction that does not pre-suppose a particular technological approach. There was discussion on the language to be used for specifying the benchmark workload. At one end is an English-based workload specification; while at the other is a specification that is completely encoded by a computer program (e.g. written in Java or C++). If the primary audience of the benchmark were end customers, then the former is preferable: the benchmark should be specified in “lay” terms, in a manner that allows non-technical audiences to grasp the essence of the benchmark and to relate it to their real-world application scenarios. Using English provides the most flexibility and broadest audience, though some parts of the specification could still employ a declarative language like SQL. However, specification in SQL should not imply that the underlying data system is required to “natively” support SQL.

4 Benchmarking for Innovation versus Competition

There was significant discussion at the workshop on the ultimate objective of the benchmarking exercise: whether it served a technical and engineering purpose or a marketing purpose. This choice will obviously influence the nature of the overall exercise. The goals of a technical benchmarking activity are primarily to test alternative technological solutions to a given problem. Such benchmarks focus more on collecting detailed technical information for use in system optimization, re-engineering, and re-design. A competitive benchmark focuses on comparing performance and price/performance (and, perhaps, other costs, such as startup costs and total cost of ownership) among competing products, and may require an audit as part of the benchmark process in order to ensure a fair competition. Furthermore, given that benchmarking can be an expensive activity, it is also important to identify the sponsor of such an activity. The consensus was that this is typically the marketing division, not the engineering division. The workshop discussion made clear that the engineering versus marketing objectives for a benchmark were, indeed, not mutually exclusive. Benchmarks need to be designed initially for competitive purposes—to compare among alternative products/solutions. However, once such benchmarks become successful (such as the TPC benchmarks), there will be an impetus within organizations to use the same benchmarks for innovation as well. Vendors will be interested in developing features that enable their products to perform well on such competitive benchmarks. There are numerous examples in the area of database software where product features and improvement have been motivated, at least in some part, by the desire to perform well in a given benchmark competition. Since a well-designed benchmark suite reflects real-world needs, this means that these product improvements really end up serving the needs of real applications.

In sum, a big data benchmark is useful for both purposes: competition as well as innovation, though the benchmark should establish itself initially as being relevant as a competitive benchmark. The primary audience for the benchmarks are the end customers who need guidance in their decisions on what types of systems to acquire to serve their big data needs. Acceptance of a big data benchmark for that purpose then leads to the use of the same benchmark by vendors for innovation as well. Finally, while competitive benchmarks are useful for marketing purposes, participants from academia are more interested in benchmarking for technical innovation.

5 Benchmark Design

In this section, we summarize the outcome of the discussion sessions on the benchmark design; whether the benchmark should be a component or an End-to-End benchmark; whether the benchmark should be modeled after a specific application; where the benchmark should get its data from, i.e. synthetic vs. real-world data and what metric the benchmark should employ.

5.1 Component vs. End-to-End Benchmark

A key design question is whether the benchmark specification should focus on modeling and benchmarking one or more "end-to-end" big data application scenarios, or on modeling individual steps of an end-to-end application scenario and measuring the performance of those individual components. We refer to the first type as end-to-end benchmarks and the latter as component benchmarks. The system that is being benchmarked may be the software itself, e.g. different software systems running on a given hardware platform, or may include the software and hardware together.

Component benchmarks measure the performance of one (or a few) components of an entire system with respect to a certain workload. They tend to be relatively easier to specify and run, given their focused and limited scope. For components that expose standardized interfaces (APIs), the benchmarks can be specified in a standardized way and run as-is, for example, using a *benchmark kit*. An example of a component benchmark is SPEC's CPU benchmark (latest version is CPU2006 [3]), which is a CPU-intensive benchmark suite that exercises a system's processor, memory subsystem and compiler. Another example of a component benchmark is TeraSort, which has proved to be a very useful benchmark because, (i) sorting is a common component operation in many end-to-end application scenarios, (ii) it is relatively easy to setup and run, and (iii) it has been shown to serve a useful purpose exercising and tuning large-scale systems.

While end-to-end benchmarks can serve to measure the performance of entire systems, they can also be more difficult to specify. Developing a benchmark kit that can be run *as-is* can be difficult due to various system dependencies that may exist and the intrinsic complexity of the benchmark itself. The TPC benchmarks in general are good examples of such end-to-end benchmarks including for OLTP (TPC-C and TPC-E) and decision support (TPC-H and TPC-DS). TPC-DS for instance, measures a system's ability to load a database and serve a variety of requests including *ad hoc* queries, report generation, OLAP and data mining queries, in the presence of continuous data integration activity on a system that includes servers, IO-subsystems and staging areas for data integration.

5.2 Big Data Applications

Big data issues impinge upon a wide range of application domains, covering the range from scientific to commercial applications. Thus, it may be difficult to find a single application that covers all extant flavors of big data processing. Examples of applications that generate very large amounts of data include scientific applications such as in high energy physics (e.g. the Large Hadron Collider, LHC) and astronomy (e.g. the digital sky surveys), and social websites such as Facebook, Twitter, and Linked-in, which are the often-quoted examples of big data. However, the more "traditional" areas such as retail business, e.g. Amazon, Ebay, Walmart, have also reached a situation where they need to deal with big data.

There is also the issue of whether a big data benchmark should attempt to model a concrete application or whether a generic benchmark—using an abstract model based on real applications—would be more desirable. The benefit of a concrete application is that real world examples can be used as a blueprint for modeling the benchmark data and workload. This makes a detailed specification possible, which helps the reader of the specification understand the benchmark and its result. It also helps in relating real world business and their workloads to the benchmark. One approach is to develop a benchmark specification based on retailer model, such as the one used in TPC-DS. This approach has the advantage that it is well understood in the TPC benchmarking community, is well-researched, and accommodates many real world applications scenarios, for example in the area of semantic web data analysis. Another approach is to model the application based on a social website. Large social websites and related services such as Facebook, Twitter, Netflix and others deal with a range of big data genres and a variety of associated processing. In either case, the data will be operated on in several stages, using a data processing pipeline, reflecting the real-world model for such applications. An abstract example of such a pipeline is shown in Figure 2.

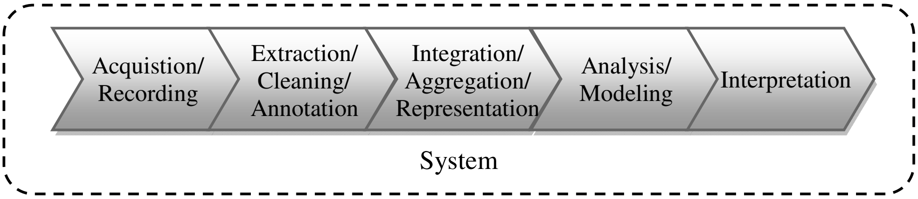

Fig. 2. Example for a big data pipeline [8]

5.3 Data Sources

A key issue is the source of data for the benchmark. Should the benchmark be based on "real" data taken from an actual, real-world application, or use synthetic data? The use of reference datasets is not practical, since that requires downloading and storing extremely large reference datasets from some remote location. Furthermore, real datasets may reflect only certain properties in the data and not others. And, most important, it would be extremely difficult to scale reference data sets to generate data at different scale factors. Thus, the conclusion was to rely on synthetic data designed to capture some of the key real-world characteristics of data. To efficiently generate very large datasets will require the use of parallel data generators [18]. Different genres of big data will require corresponding data generators.

5.4 Metrics

Big data benchmark metrics should include performance metrics as well as cost-based metrics (price/performance). The TPC is the forerunner for setting the rules to specify prices of benchmark configuration. Over the years the TPC has learned "the hard way" how difficult it is to specify rules that govern the way by which hardware and software is priced for benchmark configurations (see [5] for a detailed discussion on this topic). The TPC finalized on a canonical way to measure price of benchmark configurations and defined a pricing specification that all TPC benchmark are required to adhere. While the TPC model can be adopted for pricing, there are also other costs of interest for big data benchmarking. These include *systems setup, or startup cost*, since big data configurations may be very large in scale and setup may be a significant factor in the overall cost, plus some systems may be easier to set up than others; energy cost; and total system cost.

6 Conclusions and Next Steps

The first Workshop on Big Data Benchmarking held on May 8-9, 2012 in San Jose, CA took the first step in identifying and discussing a number of issues related to big data benchmarking, including definitional and process-based issues. The workshop concluded that there was both a need as well as an opportunity for defining benchmarks for big data applications to model end-to-end application scenarios while considering a variety of costs, including setup cost, energy cost, and total system cost. Several next steps are underway.

The workshop served as an incubator for several activities that will bring us closer to an industry standard big data benchmark. The “Big Data Benchmarking Community” has been formed (http://clds.ucsd.edu/bdbc/). It is hosted by the Center for Large-scale Data Systems research (CLDS) at the San Diego Supercomputer Center, UC San Diego. Biweekly phone conferences are being held to keep this group engaged and to share information among members. We are excited to hear that members of this community have started to work on prototypes of end-to-end big data benchmarks.

The second Workshop on Big Data Benchmarking will be held on December 17-18, 2012 in Pune, India, hosted by Persistent Systems. A third workshop is being planned for July 2013 in Xi’an, China, to be hosted by the Shanxi Supercomputing Center.

Acknowledgements. The WBDB2012 workshop was funded by a grant from the National Science Foundation (Grant# IIS-1241838) and sponsorship from Brocade, Greenplum, Mellanox, and Seagate.

References

1. Gridmix3, git://git.apache.org/hadoop-mapreduce.git/src/contrib/gridmix/
2. Internet World Stats – Miniwatts Marketing Group (December 2011), http://www.internetworldstats.com/stats.html
3. SPEC CPU2006: http://www.spec.org/cpu2006/
4. Statistical Workload Injector for MapReduce (SWIM), https://github.com/SWIMProjectUCB/SWIM/wiki
5. TPC: TPC Benchmark DS Specification, http://www.tpc.org/tpcds/spec/tpcds_1.1.0.pdf
6. TPC: TPC-Pricing Specification, http://www.tpc.org/pricing/spec/Price_V1.7.0.pdf
7. Workshop On Big Data Benchmarking (2012), http://clds.ucsd.edu/wbdb2012
8. Agrawal, D., Bernstein, P., Bertino, E., Davidson, S., Dayal, U., Franklin, M., Gehrke, J., Haas, L., Halevy, A., Han, J., Jagadish, H.V., Labrinidis, A., Madden, S., Papakonstantinou, Y., Patel, J., Ramakrishnan, R., Ross, K., Shahabi, C., Suciu, D., Vaithyanathan, S., Widom, J.: Challenges and Opportunities with Big Data. Community white paper (2011)
9. Gantz, J., Reinsel, D.: The Digital Universe Decade – Are You Ready? IDC report (2010), http://www.emc.com/collateral/analyst-reports/idc-digital-universe-are-you-ready.pdf
10. Gray, J.: Sort Benchmark Home Page, http://sortbenchmark.org/
11. Hogan, T.: Overview of TPC Benchmark E: The Next Generation of OLTP Benchmarks. In: Nambiar, R., Poess, M. (eds.) TPCTC 2009. LNCS, vol. 5895, pp. 84–98. Springer, Heidelberg (2009)
12. Huppler, K.: Price and the TPC. In: Nambiar, R., Poess, M. (eds.) TPCTC 2010. LNCS, vol. 6417, pp. 73–84. Springer, Heidelberg (2011)

13. Murphy, R.C., Wheeler, K.B., Barrett, B.W., Ang, J.A.: Introducing the Graph 500. Sandia National Laboratories (2010)
14. Nambiar, R., Poess, M.: The Making of TPC-DS. In: VLDB 2006, pp. 1049-1058, (2006)
15. Patil, S., Polte, M., Ren, K., Tantisiriroj, W., Xiao, L., López, J., Gibson, G., Fuchs, A., Rinaldi, B.: YCSB++: Benchmarking and Performance Debugging Advanced Features in Scalable Table Stores. In: SOCC 2011, pp. 9:1-9:14 (2011)
16. Poess, M., Floyd, C.: New TPC Benchmarks for Decision Support and Web Commerce. SIGMOD Record 29(4), 64–71 (2000)
17. Poess, M., Nambiar, R., Walrath, D.: Why You Should Run TPC-DS: A Workload Analysis. In: VLDB 2007, pp. 1138–1149 (2007)
18. Poess, M., Smith, B., Kollár, L., Larson, P.: TPC-DS, Taking Decision Support Benchmarking to the Next Level. In: SIGMOD 2002, pp. 582–587 (2002)
19. Rabl, T., Frank, M., Sergieh, H.M., Kosch, H.: A Data Generator for Cloud-Scale Benchmarking. In: Nambiar, R., Poess, M. (eds.) TPCTC 2010. LNCS, vol. 6417, pp. 41–56. Springer, Heidelberg (2011)