New Initiatives in the TPC

Meikel Poess(✉)

Oracle Corporation, Redwood Shores, CA 94065, USA
meikel.poess@oracle.com

Abstract. TPC stands for *stricter standards* and *tougher tests* in system benchmarking for over 30 years, during which technology advanced enormously. In order to stay relevant the TPC had to adapt to the advancing technology or become obsolete. Initially, the business use case of computer systems was mainly focused on online transaction processing (OLTP). Hence, the TPC developed a series of OLTP benchmarks, learning valuable lessons along the way (TPC-A, TPC-B, TPC-C and TPC-E). Then the TPC ventured into other use cases such as decision support (TPC-D, TPC-H, TPC-R, TPC-DS), web applications (TPC-App), virtualization (TCP-VMS) and data integration (TPC-DI). When the TPC realized that the old way of developing benchmarks was not sustainable, it added a new way of developing benchmark, express benchmark, which resulted in a flurry of benchmark standards (TPCx-HS, TPCx-V, TPCx-BB, TPCx-HCI, TPCx-IoT and TPCx-AI). With *cloud computing* becoming more and more important, the TPC made sure that new benchmarks would be able to run in the cloud. At the same time the TPC modified its pricing specification to allow for pricing of cloud based benchmarks.

Keywords: Benchmark Standards · Performance Evaluation · Databases

1 Introduction

The Transaction Processing Performance Council (TPC™), founded in 1988, can look back at a very rich history of developing industry standard benchmarks for computer systems. Without doubt these benchmarks shaped the way performance measurements have been conducted for decades.

The foundation of the TPC was mostly driven by the need to overcome a phenomenon, commonly referred to as *benchmarketing*. Using this marketing practice, organizations claim superior performance based on non-standard benchmarks. The aim of running non-standard benchmarks on a computer system is to claim superior performance by focusing on those aspects of the computer systems that are superior to those of competitors. However, these non-standard benchmarks often run workloads without well defined execution rules, no minimum requirements for component redundancy, no rules for fair pricing, no verifiable documentation of the benchmark execution including its overall correctness. This is why TPC benchmarks have been successfully used for decades and are still being used.

R. Nambiar and M. Poess (Eds.): TPCTC 2022, LNCS 13860, pp. 127–148, 2023.
https://doi.org/10.1007/978-3-031-29576-8_10

Technology has rapidly advanced over the years, triggering the TPC to adapt constantly to new circumstances. Before diving into more recent developments in the TPC, I will give a brief history of past TPC benchmarks focusing on how the TPC adapted to changes in technology from the very beginning.

The TPC started by developing on-line transaction processing (OLTP) systems benchmarks. Based on the DebitCredit benchmark that Jim Gray developed in collaboration with 24 others from academia and industry [4], the TPC developed the first OLTP benchmark, TPC-A™ [17]. TPC-A for the first time established a level playing field by mandating production-oriented requirements, mandating the reporting of peak performance numbers that can be sustained for a reasonable amount of time, mandating ACID compliance, and a Full Disclosure Report (FDR) that discloses the complete benchmark implementation and all test output. With technology enhancing in the years following, TPC-A was replaced by TPC-B™ [19] and, eventually, by TPC-C™ [13,20]. TPC-C is still an active benchmark today, serving the benchmark community for 30 years.

1.1 Venturing into New Benchmark Domains

After covering the OLTP benchmark needs the TPC started developing benchmarks in other domains. Two years, after TPC-C was released, the TPC presented its first decision support (DS) benchmark, TPC-D ™ [21]. TPC-D served the benchmark community for five years until 1999 when *Database Management Software* (DBMS) vendors invented materialized views.

The cost for creating materialized views was not adequately represented in TPC-D's performance metric causing results to sky rocket, dwarfing results of vendors without materialized view support. TPC-D was subsequently split into two separate benchmarks, the reporting decision support benchmark, TPC-R™ [12,23], that allowed the use of sophisticated auxiliary data structures, e.g. materialized views, and the ad-hoc decision support workload TPC-H™ [12,22]. While TPC-H flourished, TPC-R was retired in 2005. TPC-H is still an active benchmark with 14 results in 2021, 3 results in 2020 and 8 results in 2019.

After successfully establishing benchmarks for the OLTP and decision support domains, the TPC started developing benchmarks for emerging domains, such as *electronic commerce*, *virtualization*, *hyperconversion*, *big data* and *internet of things*. In parallel the TPC started working on developing updated versions for TPC-C and TPC-H.

The first benchmark in a new domain was TPC-W™ [12], established in 2002. It was, with minor changes, later re-branded as TPC-App™ [18]. It was an application server benchmark simulating the activities of a business-to-business transactional application server. While very popular in academia, TPC-App never generated many benchmark publications and was retired in 2008. For a detailed timeline of TPC benchmarks see Fig. 1.

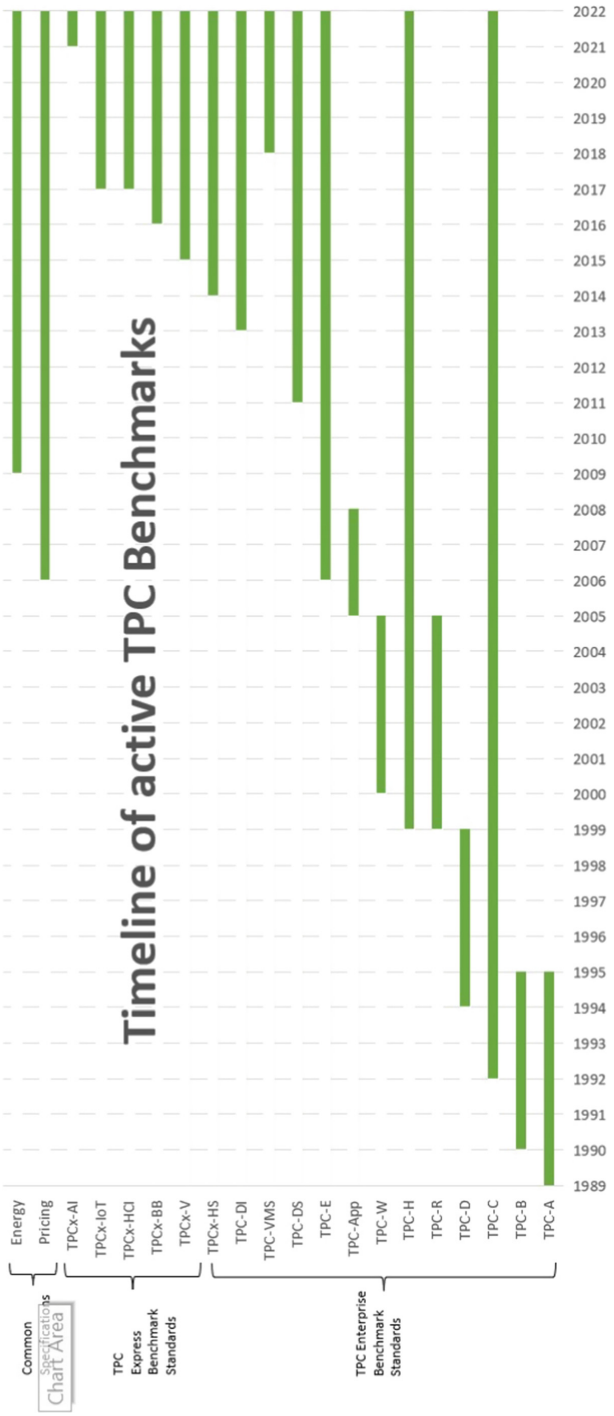


Fig. 1. History of TPC Benchmarks Between the Years 1989 and 2022

1.2 TPC's Benchmark Development Model Until 2013

Until 2013 TPC's development model was to build benchmarks from the ground up, i.e. with new database schemata, new data generators, new transactions, in case of TPC-E™ [6], and new queries, in case of TPC-DS™ [8]. This benchmark development model resulted in benchmarks the TPC categorizes as *enterprise class* benchmarks. Figure 3 shows Clause 11 of the TPC Policies Version 6.18, outlining the development cycle of enterprise class benchmarks. The eight step process outlined in Clause 11 ensures that the resulting benchmarks are technically solid and accepted by the majority of TPC member companies. The following steps outlines the development steps.

- Step 1 - **Benchmark Submittal:** The draft benchmark specification submitted in this early step in the development cycles acts as a document for discussion in the TPC. Any member company can propose a draft benchmark specification. It ensures a clear intent of the benchmark idea.
- Step 2 - **Creation of a Benchmark Subcommittee:** In order to develop the draft specification into a specification that can be proposed in mail ballot (Step 8), a benchmark subcommittee needs to be formed. This subcommittee works under specific rules of the TPC including regular meetings, documentation of current work and operating under Robert's Rules of Order.
- Step 3 - **Status and Direction:** To inform all members of the TPC the subcommittee is required to present a status report at every General Meeting (usually five times a year).
- Step 4 - **Authorization Public Release of Draft Specification:** Once the benchmark specification is complete it is proposed for public release. All discussion and material developed in the TPC is considered TPC confidential and cannot be shared with people outside the TPC (general public). However, to assure a benchmark addresses the needs of the general public the TPC deems it necessary to consider feedback from the general public. This step allows sharing of the draft specification with the public
- Step 5 - **Accepting a Standard for Review:** This step accepts the standard for review. This is a formal step, but has a high bar as it undergoes thorough review by all members to ensure the benchmark specification is sound and solid.
- Step 6 - **Formal Review:** This step includes solicitation of feedback from the general public and within the TPC of the draft benchmark specification. The benchmark subcommittee is encouraged to incorporate the feedback into the draft specification.
- Step 7 - **Approval for Mail Ballot:** This step is a formal step to approve the draft specification for mail ballot. It serves as the final check for the benchmark specification
- Step 8 - **Mail Ballot Approval:** This step is the final step to create a new benchmark. The bar to pass mail ballot is very high. $\frac{2}{3}$ of all member companies must cast their ballot for mail ballot to be valid. To pass the benchmark $\frac{2}{3}$ of those ballots need to be in favor of the benchmark (Fig. 2).

11.2 **TPC-Enterprise Benchmark Development Cycle**

The following outlines the steps for submitting a benchmark proposal and securing approval.

11.2.1 **Step 1: Benchmark Submittal**

Member companies will submit a draft standard specification in a format similar to **TPC Benchmark Standard**s. The proposal is submitted to the **Council** and is forwarded to the **SC** for consideration. The **SC** will review the contents, applicability and potential of the proposal and present a recommendation back to the **Council**, identifying advantages/disadvantages and proposed course of action. The **Council** must then vote to formally accept the proposal for future work.

11.2.2 **Step 2: Creation of a Benchmark Subcommittee**

Given the acceptance of the proposal for future work, the **Council** will then establish and empower a **Benchmark Subcommittee** to develop a formal benchmark **Specification**. To speed-up the benchmark development cycle, the subcommittee is empowered to brief non-members on their benchmark in order to obtain timely feedback.

11.2.3 **Step 3: Status and Direction**

At each **General Meeting,** the **Benchmark Subcommittee** will provide a status update on its work, including a working draft of the **Specification**. During the **General Meeting,** the **Council** may provide direction and feedback to the subcommittee to further their work.

11.2.4 **Step 4: Authorizing Public Release of Draft Specification**

If it deems it advisable, the **Council** may authorize the release of a draft **Specification** to the public. The principal goals of releasing a draft specification are to encourage companies to implement the draft **Specification**, to gather more experimental data, and to speed up the approval of a **TPC-Enterprise Benchmark Standard**.

Within the purpose of the procedure as outlined above, companies are encouraged to run the draft **Specification**, document the results, and discuss the results with **All Members** and customers. Companies may also publish technical articles or make presentations to industry conferences in which they discuss results. However, these articles/presentations are bound by the conditions in **Policies** § 8.1 (Use of TPC Materials) and **Policies** § 8.3 (Fair Use of TPC Specifications).

Comment: Companies are reminded that this draft **Specification** is not a **Benchmark Standard**, and companies must adhere fully to all the provisions and restrictions of the **Fair Use Policy**. Only results published in accordance with a **Benchmark Standard** are considered **TPC Results** and can be publicized as such.

11.2.5 **Step 5: Accepting a Standard for Review**

When the **Benchmark Subcommittee** feels that the **Specification** is of sufficient quality to be considered for formal review and approval, it will submit the **Specification** to the **Council** for approval to advance into formal review. A formal review period of 60 days is customary.

11.2.6 **Step 6: Formal Review**

During this phase, the **Specification** will be made available to **All Members** and the public for formal review. All comments and proposed changes generated from the review will be **posted** to the **Private Web Site** and considered by the **Benchmark Subcommittee** for resolution.

Comment: **Members** and the public are reminded that this **Specification** is not a **Benchmark Standard**, and companies must adhere fully to all the provisions and restrictions of the **Fair Use Policy**. Only results published in accordance with a **Benchmark Standard** are considered **TPC Results** and can be publicized as such.

11.2.7 **Step 7: Approval for Mail Ballot**

The **Benchmark Subcommittee** will propose resolution of comments from the formal review as an updated **Specification** to **All Members** for approval by the **Council**. The **Council** approves the updated **Specification** by voting to send the **Specification** out for **Mail Ballot**.

11.2.8 **Step 8: Mail Ballot Approval**

To become a **Benchmark Standard**, the **Specification** must be approved by a **Mail Ballot** in accordance with **Policies** § 4.5 and **Policies** § 11.3.1.

In the event the **Mail Ballot** is not approved, the benchmark development work will automatically cease. If the benchmark development was the only work of the **Benchmark Subcommittee**, the subcommittee will be disbanded at the conclusion of the next **General Meeting** if the **Council** does not authorize continued work.

Fig. 2. Clause 11 of the TPC Policies Version 6.18: Enterprise Class Benchmark Development Cycle

The TPC realized that the existing approach to developing Enterprise Class benchmarks, e.g. TPC-C, TPC-D and TPC-App, took too much resources in the TPC and their member companies, causing long benchmark development cycles and, thus, late time to market. The main reasons for the long development time was (i) following the stiff development model with its many approval steps involving super majority of the entire TPC membership, (ii) building of benchmarks from the ground-up and (iii) defining technology agnostic benchmarks

The existing development model, which requires many level of approval, gave a lot of opportunity to lobby for or against benchmarks. Especially, the final mail ballot approval step marked a high bar for establishing a new benchmark. For the following example I assume a total TPC membership of 22 companies. In order for a mail ballot to be valid, $\frac{2}{3}$ of all member companies must cast their ballot, e.g. $\lceil \frac{2}{3} * 22 \rceil = 15$. Usually voter participation is less than 100%. Let's assume there is a 91% voter participation, that is, 2 member companies do not plan to cast their votes regardless. If a small group of active member companies, e.g. 4 (Group Of Four), do not wanted an enterprise class benchmark to pass, Group Of Four would only need to convince another 2 active member companies not to cast their ballots for the entire mail ballot to be invalid. Assuming the Group Of Four fails to convince 2 active member companies not to cast their ballot, and, as a result, 20 member companies cast their votes, then mail ballot is valid. However, for the benchmark to pass it requires $\lceil \frac{2}{3} * 20 \rceil = 14$ *yes* votes. At that time, the Group Of Four would only need to convince 2 more companies to vote *abstain*.

Apart from the bureaucratic and political hurdles outlined in the above paragraph, developing an enterprice class benchmark is also very resource intensive. One of the main reasons is the amount of data needed to populate a meaningful database for performance testing purposes. Data generation must be reliable, fast and realistic. Early TPC benchmarks developed their own data generators, such as TPC-C, TPC-E, TPC-H and TPC-DS. However, due to the time it took to develop these generators, the TPC started utilizing data generator frameworks, such as the parallel data generation framework PDGF [14]. PDGF greatly reduced the time to develop a reliable and fast data generator.

Enterprise class benchmarks are technology agnostic. They do not mandate the use of any specific technology/technique to execute the workload specified in the benchmark specification. For instance, TPC-H does not specify any particular Data Definition Language (DDL) for the creation of its eight tables, including data types. Instead, TPC-H defines the minimum requirement for data types.

Figure 3 shows Clause 1.3 of the TPC-H specification. It defines pseudo data types, e.g. *identifier*, and their requirements. A column using the datatype *identifier* must be able to hold any key value generated for the column by the data generator and be able to support at least 2,147,483,647 unique values. A benchmark implementation of TPC-H can use any of the database datatypes as long as the datatype supports the above requirements. However, the implementation datatype chosen by the implementation for a particular datatype definition

1.3 Datatype Definitions

1.3.1 The following datatype definitions apply to the list of columns of each table:

- **Identifier** means that the column must be able to hold any key value generated for that column and be able to support at least 2,147,483,647 unique values;

Comment: A common implementation of this datatype will be an integer. However, for SF greater than 300 some column values will exceed the range of integer values supported by a 4-byte integer. A test sponsor may use some other datatype such as 8-byte integer, decimal or character string to implement the identifier datatype;

- **Integer** means that the column must be able to exactly represent integer values (i.e., values in increments of 1) in the range of at least -2,147,483,646 to 2,147,483,647.
- **Decimal** means that the column must be able to represent values in the range -9,999,999,999.99 to +9,999,999,999.99 in increments of 0.01; the values can be either represented exactly or interpreted to be in this range;
- **Big Decimal** is of the Decimal datatype as defined above, with the additional property that it must be large enough to represent the aggregated values stored in temporary tables created within query variants;
- **Fixed text, size N** means that the column must be able to hold any string of characters of a fixed length of N.

Comment: If the string it holds is shorter than N characters, then trailing spaces must be stored in the database or the database must automatically pad with spaces upon retrieval such that a CHAR_LENGTH() function will return N.

- **Variable text, size N** means that the column must be able to hold any string of characters of a variable length with a maximum length of N. Columns defined as "variable text, size N" may optionally be implemented as "fixed text, size N";
- **Date** is a value whose external representation can be expressed as YYYY-MM-DD, where all characters are numeric. A date must be able to express any day within at least 14 consecutive years. There is no requirement specific to the internal representation of a date.

Comment: The implementation datatype chosen by the test sponsor for a particular datatype definition must be applied consistently to all the instances of that datatype definition in the schema, except for identifier columns, whose datatype may be selected to satisfy database scaling requirements.

1.3.2 The symbol SF is used in this document to represent the scale factor for the database (see Clause 4:).

Fig. 3. Clause 1.5 of TPC-H: Data type Definitions

must be applied consistently to all occurrences of that datatype definition in the schema (see comment in Clause 1.3.1).

There are many advantages, but also crucial disadvantages to this benchmark class. One important advantage of enterprise class benchmarks is that they allow the use of technology that was not anticipated at the time the benchmarks was designed. As a consequence, enterprise class benchmarks encourage the development of technology that may lead to better performance results. On the other hand technology agnostic benchmarks make it much more difficult to define sound benchmark rules that do not open the door for technology that break the benchmark/make it obsolete, as happened to TPC-D.

1.3 Express Benchmarks™, a Benchmark Model for Rapid Benchmark Development

To reduce benchmark development time and approval time for incremental improvements of benchmarks, the TPC established a new benchmark class,

named express benchmarks™ [7] in 2013. Express class benchmarks are following the same rigorous development rules as enterprise class benchmarks. However, instead of requiring a separate benchmark implementation by the test sponsor, they allow a test sponsor to download a benchmark suite, a.k.a. *kit*, that runs the entire benchmark.

A kit based benchmark is self contained and limited to the hardware and software it is designed for. This reduces the benchmark development time as no comprehensive wording to define the benchmark needs to be developed, nor does the code need to be developed for all possible hardware and software combinations. It also opens the door for the development of a TPC benchmark based on open source software.

The bar to run an express class benchmark is much lower compared to that of an enterprise class benchmark. Benchmark sponsors[1] do not have to implement an express class benchmark as the kit provides the necessary implementation. The audit process of an express class benchmark is also much easier. Instead of undergoing an audit with a TPC certified auditor including the scheduling of the audit, waiting for the results from the auditor, paying the auditor, certification of express class benchmarks results can be done by TPC peers in a *pre-publication board* (PPB). It is, however, possible to have an audit of an express class benchmark be certified by a TPC certified auditor.

2 TPC Express Benchmarks™ Becoming Reality

The introduction of the express benchmark class resulted in an enormous increase in new benchmark development. The last Enterprise Class benchmark, TPC-DI™ [11,20], hit the market in 2013 after $5\frac{1}{2}$ years of development, from April 2008 to October 2013.

After that a new era of express class benchmark development began. The first express class benchmark was TPCx-HS™ [1,2,9], a *big data system benchmark*. TPCx-HS, published in 2014, marked a major turning point in benchmark development in the TPC, as it's development is based on the existing open source benchmark, TeraSort. Ironically, Terasort itself was based on a benchmark originally proposed by Jim Gray, *Sort Scan* [4]. Two years later TPCx-BB™ [24] [5] was published, TPC's first *end-to-end big data benchmark*. It is largely based on TPC-DS. In 2017 three Express Benchmarks™ followed, TPCx-V™ [15,27], TPCx-HCI™ [16,25] and TPCx-IoT™ [10,26]. TPCx-V and TPCx-HCI are virtualization benchmarks. TPCx-V is based on TPC-E and TPCx-HCI is based on TPC-V. TPCx-IoT is the first end-to-end benchmark for measuring *internet of things* systems.

3 Big Data Benchmarks

Big data is a term coined to describe processing of very large data datasets. These datasets have three properties: Volume, Velocity, and Variety. Volume reflects the

[1] A benchmark sponsor is a company or multiple companies that run a TPC benchmark and publish its result on the TPC website.

size of the dataset, which is usually measured in Terabytes. Velocity stands for the speed at which data is generated. Variety represents the structure in which data is generated. In traditional database management systems data gathered in real life is transformed into a coherent structured form, i.e. tables, a.k.a. entities. However, recent years have shown that a large percent of data captured in real life cannot be easily transformed into a structured form. These kind of data is considered semi-structured and un-structured. An example for semi-structured data are user clicks from a retailer's website, which are extracted from a web server log. They can vary in format due to their action type. An example of un-structured data is information from product reviews that are commonly used for sentiment analysis. These are usually provided by customers in free form text.

TPCx-HS marks TPC's first big data benchmark and its first express class Benchmark. Based on the TeraSort benchmark, TPCx-HS is considered a system level benchmark measuring hardware, operating system and commercial Apache HDFS API compatible software distributions. It adapted TeraSort's data model, including its data generator, and added TPC's stringent formal benchmark rules for implementation, execution, metric calculation, result verification, result publication and system pricing. TPCx-HS models a continuous system availability of 24 by 7. TPCx-HS defines the usual three primary metrics of any TPC benchmark, *performance metric*, *price-performance* and *system availability*. The performance metric is HSph@SF, the effective sort throughput of the benchmarked configuration:

$$HSph@SF = \frac{SF * 3600}{T}$$

where:

- $SF \in \{1, 3, 10, 30, 100, 300, 1000, 3000, 10000\}$ is the scale factor, indicating the amount of data to be sorted in terabyte, and
- T is the total elapsed time for the run in seconds.

TPCx-HS price-performance metric is defined as:

$$\$/HSph@SF = \frac{P}{HSph@SF}$$

where:

- P is the total cost of ownership of the *system under test* (SUT[2]). P includes the hardware and software components present in the SUT, a communication interface that can support user interface devices, additional operational components configured on the test system, and maintenance on all of the above for a three year period, and

[2] The system under test is essentiall the benchmark configuration, i.e. all hardware components used to run the benchmark and price the particular benchmark run. SUT is a TPC term that is used in all benchmarks.

– HSph@SF is the system performance as defined above.

TPCx-HS' System Availability Date is defined in the TPC pricing specification [3] as the time the SUT is available for general purchase.

In addition to the above main metrics, TPCx-HS also reports the following numerical quantities:

– T_G: Data generation phase completion time with HSGen reported in hh:mm:ss format,
– T_s: Data sort phase completion time with HSSort reported in hh:mm:ss format, and
– T_V: Data validation phase completion time reported in hh:mm:ss format.

The Hadoop ecosystem is moving fast beyond batch processing with MapReduce. In 2016 the TPC introduced TPCx-HS Version 2. Based on TPCx-HS Version 1, Version 2 added support for Apache Spark - a popular platform for in-memory data processing that enables real-time analytics on Apache Hadoop. In addition, TPCx-HS Version 2 supports MapReduce (MR2). Publications on traditional on-premise and clouds deployments. Benchmark publications from TPCx-HS Version 1 and Version 2 are not comparable.

TPCx-BB is an end-to-end benchmark designed to measure the performance of the software and hardware components of data processing systems when executing big data tasks. These tasks uncover hidden data patterns, unknown data correlations, market trends, customer preferences and other useful information that can help organizations make informed business decisions. TPCx-BB models the big data aspect of a large retail company with physical and online presence. The workload is comprised of 30 use-cases, each representing a realistic big data problem in the form of queries and machine learning tasks that execute on large scale datasets, ranging from one terabyte to multiple Petabytes.

Big data problems are especially complex when big data frameworks, such as Apache Hadoop, are used to extract meaningful information from large datasets. Such complexity results mainly from enabling parallel data processing in a distributed fashion across multiple nodes. Currently the TPCx-BB workloads are implemented to be executed on Hadoop-based and Alibaba's MaxCompute platforms. It can be easily extended to other platforms.

TPCx-BB, like any other TPC benchmark, defines three primary metrics, the performance metric BBQpm@SF, reflecting the TPCx-BB Queries per minute throughput, the price/performance metric $/BBQpm@SF and the system availability date.

The performance metric is defined as:

$$BBQpm@SF = \frac{SF * 60 * M}{T_{LD} + \sqrt{T_{PT} * T_{TT}}} \tag{1}$$

The enumerator is adjusted by the data set size, i.e. scale factor (SF) to allow for larger configuration having a larger overall metric. This is solely done

to encourage benchmarking larger scale factors. Without this adjustment configurations running larger scale factors might achieve lower or equal performances than systems running smaller scale factors, although their performance is equal. I will demonstrate this with a simple example. Let's assume a system A, running at scale factor 1 achieves $BBQpm@SF = 100$. Let's further assume a system B with 100 times more power (compute and IO) runs at scale factor 100. Assuming perfect scalability of the workload from SF=1 to SF=100 and from system A to system B, system A would finish the 100 times larger workload in the same time system A finished its workload. With no scale factor adjustment they would both report the same $BBQpm@SF = 100$. To showcase that system B's workload is 100 times larger than that of system A, the metric is adjusted by mulitplying the result by the scale factor.

The denominator is the sum of the adjusted elapsed time of the load test and the geometric mean of the adjusted elapsed times of the power and throughput tests. The elapsed time of the *Load Test* measured in seconds is

$$T_{Load} \tag{2}$$

In real life system are reloaded rarely. Thereby, the TPC decided that only 10% of the elapsed time of the load, T_{Load}, should contribute to the load metric. Hence, the elapsed time of the Load Test T_{Load} is multiplied by the Multiplication Factor (MF).

$$T_{LD} = MF * T_{Load} \tag{3}$$

The number of queries in TPCx-BB is

$$M = 30 \tag{4}$$

The geometric mean of the elapsed time $Q(i)$ $(1 \leq i \leq M)$ in seconds of each of the M queries as measured during the Power Test, multiplied by the number of queries in the benchmark is:

$$T_{PT} = M * \sqrt[M]{\prod_{i=1}^{i=M} Q(i)} \tag{5}$$

The throughput test metric, T_{TT}, is computed as the total elapsed time of the throughput test T_{Tput} divided by the number of streams, n.

$$T_{TT} = \frac{1}{n} * T_{Tput} \tag{6}$$

3.1 Virtualization Benchmarks, TPCx-V and TPCx-HCI

TPCx-V is a Virtualization Benchmark for Database Workloads. It measures the performance of a virtualized server platform under a demanding database workload. It stresses CPU and memory hardware, storage, networking, hypervisor, and the guest operating system. TPCx-V workload is database-centric

and models many properties of cloud services, such as multiple VMs running at different load demand levels, and large fluctuations in the load level of each VM. The execution rules of the benchmark include performing a database load, running the workload, validating the results, and even performing many of the routine audit steps. Another unique characteristic of TPCx-V is an elastic workload that varies the load delivered to each of the VMs by as much as 16x, while maintaining a constant load at the host level.

The Performance Metric reported by TPCx-V measures the number of completed Trade-Result transactions expressed in transactions-per second-V (tpsV). Multiple Transactions are used to simulate the business activity of processing a trade, and each Transaction is subject to a Response Time constraint.

TPCx-HCI is a benchmark for *Hyper-Converged Infrastructure* (HCI). It measures the performance of Hyper-Converged Infrastructure under a demanding database workload. It stresses the virtualized hardware and software of converged storage, networking, and compute resources of the measured HCI platform. The short development time of TPCx-HCI can be attributed to the fact that it leverages many elements of TPCx-V. Results of the two benchmarks are, however, not comparable. The number of virtual machines (VM) is calculated differently for the two benchmarks. The TPCx-HCI workload is database-centric and models many properties of cloud services, such as multiple VMs running at different load demand levels, and large fluctuations in the load level of each VM. The TPCx-HCI benchmarking kit, which was developed from scratch, is a complete end-to-end kit that loads the databases, runs the benchmark, validates the results, and even performs many of the routine audit steps. Two of the main unique characteristics of TPCx-HCI are:

- It has an elastic workload that varies the load delivered to each of the VMs by as much as 16x, while maintaining a constant load at the cluster level. Sustaining optimal throughput for this elastic workload on a multi-node HCI cluster would typically benefit from frequent migrations of VMs to rebalance the load across nodes. This property measures the efficiency of VM migration as well as the uniformity of access to data from all the nodes.
- In the Data Accessibility test of TPCx-HCI, a node is powered down ungracefully, and the benchmark continues to run on the other nodes. The test sponsor has to include a throughput graph for this test, demonstrating the impact on performance, as well as report the recovery time to regain resilience.

3.2 TPCx-IoT

The *internet of things* (IoT) is a term that describes groups of physical devices, a.k.a. "things", that communicate to each other over the *internet*. These devices (sensors) collect data and/or act on data (actuators). The complexity of IoT devices ranges from ordinary household objects, e.g., switches, plugs and home appliances, to sophisticated industrial applications, such as those used in agricultural, manufacturing and energy sectors, e.g.: smart fertilizer systems and

digital control systems. The number of IoT devices currently deployed is estimated to be in the tens of billions. A typical IoT system is designed in a three-tier architecture:

- *Edge Tier*: sensors/actuators send data usually at very high frequencies (usually analog signals) to edge devices. Edge devices convert the analog signals into digital signals and send that data to the gateway systems.
- *Gateway Tier*: gateway database management systems serve as a short-term persistent storage. In many cases they also perform lightweight local analytics by filtering and aggregating data (dashboard-like functionality).
- *Datacenter Tier*: back-end database systems receive data from potentially multiple gateways at low frequencies and store that data long-term to perform complex global analytics.

TPCx-IoT is the first IoT benchmark specifically designed to measure the performance of *IoT gateway systems*. It enables direct comparison of different software and hardware solutions. Using the operational model of a typical electric utility provider with thousands of power substations, TPCx-IoT provides verifiable performance, price-performance and availability metrics for commercially available systems that typically ingest massive amounts of data from large numbers of devices, while running real-time analytic queries. Its flexible design allows TPCx-IoT to be used to assess a broad range of system topologies and implementation methodologies in a technically rigorous and directly comparable manner. TPCx-IoT defines three primary metrics: *performance metric*, *price-performance* and *system availability*. The performance metric, $IoTps$, represents the effective throughput capability of the gateway, where SF is the Scale Factor (amount of data ingested) and T is the ingestion elapsed time in seconds.

$$IoTps = \frac{SF}{T} \tag{7}$$

The Price-Performance metric represents the total cost of ownership of the system over three years for each thousand Transactions. P is the total cost of ownership of the SUT. The first version of TPCx-IoT reported price-performance for each transaction. The reason for reporting the price-performance for each thousand transactions in Version 2 of TPCx-IoT is, because of system prices reducing and performance of systems increasing resulted in very small price-performance numbers with significant digits in the 3rd position after the decimal.

$$\$kIoTps = \frac{1000 * P}{IoTps} \tag{8}$$

3.3 Enterprise and Express Class Publications

Figure 4 shows the number of benchmark publications in both the Enterprise Class and the Express Class categories (Table 1).

Fig. 4. Enterprise and Express Class Benchmark Publications since 2014

Table 1. Comparison Enterprise Class and Express Class Benchmark Specifications.

	Enterprise Class	**Express Class**
Specification Type	Written Specification with some TPC provided code	Complete KIT that runs the benchmark end-to-end
Tuning	Best possible optimization allowed due to custom implementation	Limited tuning based on what is allowed in KIT
Audit	Audit by TPC certified auditor	Peer audit, organized by member companies in Pre Publication Board (PPB)
Member Publication Cost	$1,250	$500
Non-Member Publication Cost	$2,500	$750
Approval Major Revision	$\frac{2}{3}$ of member companies in Council Meeting	$\frac{2}{3}$ of member companies in mail ballot

4 TPC Derived Benchmarks

In recent years the TPC noticed a new trend in the benchmark community: Publication of system performance numbers using benchmarks based on TPC benchmarks. This trend resonates to the situation in the early 80s during which companies engaged in benchmarketing. In the 80s companies modified versions of their competitor's benchmarks, while today companies use, for the most part, benchmarks derived from original TPC benchmarks. In many cases they modify

a TPC benchmark by simplifying the execution rules and workload (transactions/queries). For instance, benchmarks derived from TPC-DS might only perform a Database Load Test and a Power Run Test with a subset of TPC-DS' original 99 queries.

4.1 Initiative to Allow the Use of TPC Benchmark Material in Non-TPC Benchmarks

Instead of battling companies that use benchmarks derived from TPC benchmarks, the TPC started an initiative to allow the use of TPC benchmark material in Non-TPC benchmarks under very specific rules. This initiative amended the TPC policies. The amendments to the TPC policies include a definition of Non-TPC Benchmark Standard, rules on how TPC benchmark names and metrics may be used and a specific disclaimer to easily identify Non-TPC Benchmark Standards.

Figure 5 shows Clause 0.2.36 of the TPC policies that defines Non-TPC Benchmark Standards.

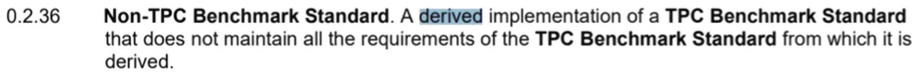
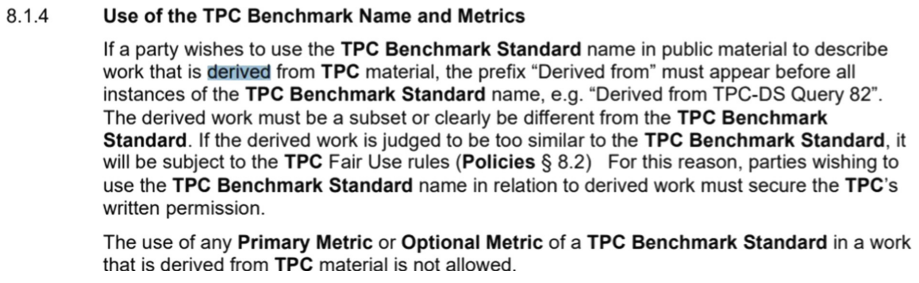

0.2.36 **Non-TPC Benchmark Standard**. A derived implementation of a **TPC Benchmark Standard** that does not maintain all the requirements of the **TPC Benchmark Standard** from which it is derived.

Fig. 5. Definition of a Non-TPC Benchmark Standard

In order to protect TPC's benchmarks from misuse and to clearly differentiate results from Non-TPC Benchmark Standards from results obtained with TPC Benchmark Standards, the TPC mandates that all public material describing work that is derived from TPC material to prefix all instances of Non-TPC Benchmark Standard with "Derived from". Furthermore, if a Non-TPC Benchmark Standard is too similar to the TPC Benchmark Standard, it will be subject to the TPC Fair Use rules (see Fig. 6).

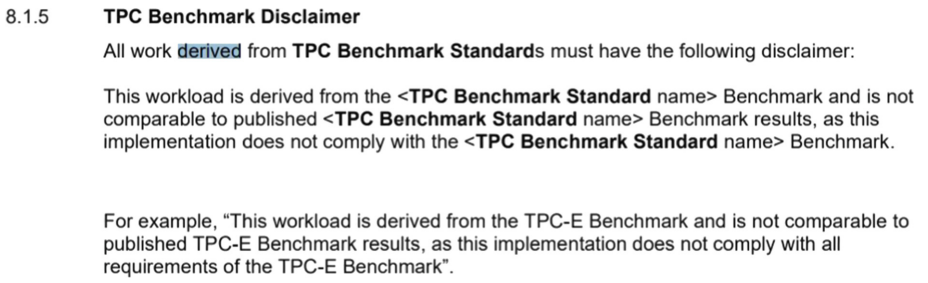

8.1.4 **Use of the TPC Benchmark Name and Metrics**

If a party wishes to use the **TPC Benchmark Standard** name in public material to describe work that is derived from **TPC** material, the prefix "Derived from" must appear before all instances of the **TPC Benchmark Standard** name, e.g. "Derived from TPC-DS Query 82". The derived work must be a subset or clearly be different from the **TPC Benchmark Standard**. If the derived work is judged to be too similar to the **TPC Benchmark Standard**, it will be subject to the **TPC** Fair Use rules (**Policies** § 8.2) For this reason, parties wishing to use the **TPC Benchmark Standard** name in relation to derived work must secure the **TPC**'s written permission.

The use of any **Primary Metric** or **Optional Metric** of a **TPC Benchmark Standard** in a work that is derived from **TPC** material is not allowed.

Fig. 6. Use of the TPC Benchmark Name and Metrics

Furthermore, all work derived from TPC Benchmarks must include a specific disclaimer. Figure 7 shows the exact wording in Clause 8.1.4 of the TPC policies that defines this disclaimer.



8.1.5 **TPC Benchmark Disclaimer**

All work derived from **TPC Benchmark Standard**s must have the following disclaimer:

This workload is derived from the <**TPC Benchmark Standard** name> Benchmark and is not comparable to published <**TPC Benchmark Standard** name> Benchmark results, as this implementation does not comply with the <**TPC Benchmark Standard** name> Benchmark.

For example, "This workload is derived from the TPC-E Benchmark and is not comparable to published TPC-E Benchmark results, as this implementation does not comply with all requirements of the TPC-E Benchmark".

Fig. 7. TPC Benchmark Disclaimer

4.2 TPC's Open Source Initiative

Another initiative, the TPC started to foster the use of Non-TPC Benchmark Standards in a regulated way, is the TPC-OSS^{TM} initiative. It embraces benchmark community driven development and testing, brings the TPC closer to the new generation of benchmark developers that expect access to source code and makes the TPC brand known in the developer community. The TPC-OSS collaborated with HammerDB starting in May 2019. HammerDB is well known in the open source community and has been one of the first open source products to use TPC based benchmarks. To bring the TPC and HammerDB communities together the HammerDB source code was moved to the TPC-Council GitHub repository.

HammerDB is a benchmark Tool Kit that runs benchmarks derived from TPC benchmarks against the following DBMS: Oracle, SQL Server, Db2, Times Ten, MySQL, MariaDB, PostgreSQL, Greenplum, Postgres Plus Advanced Server, Amazon Aurora and Redshift. HammerDB currently implements an OLTP workload based derived from TPC-C, named TPROC-C and an analytical workload derived from TPC-H, named TPROC-H. For each of these TPC derived benchmarks HammerDB creates a database schema, loads the schema with data from TPC provided code and executes the workload. Unlike the original TPC benchmarks, TPC-C and TPC-H, HammerDB provides an end-to-end KIT to execute the workload, making it very easy for anybody to run these complicated workloads.

The first version of HammerDB to be hosted in the TPC GitHub repository was Version 3.3 in November 2019. Since then, the TPC has released 7 follow up releases listed in Table 2.

Table 2. HammerDB Release in the TPC GitHub Repository

Release	Release Date
3.3	11-14-2019
4.0	11-26-2020
4.1	4-2-2021
4.2	7-9-2021
4.3	11-19-21
4.4	3-3-22
4.5	7-22-22
4.6	12-1-22

5 TPCx-AI™, First End-To-End AI benchmark Standard

TPCx-AI, released Sept 7, 2021 is the first end-to-end benchmark standard for measuring the performance of machine learning and data science platform. The benchmark emulates the behavior of representative industry AI solutions that are relevant in current production data centers and cloud environments. TPCx-AI is an express benchmark and, therefore, it can be easily set up in minutes by downloading the benchmark kit that is provided by the TPC (Fig. 8).

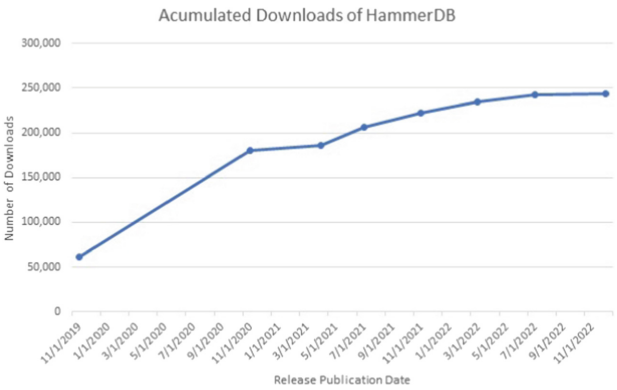


Fig. 8. Accumulated Downloads HammerDB

TPCx-AI addresses the needs of technology providers of both hardware and software companies, companies adopting AI in production environments, companies comparing performance and price performance (TCO), academia researching new AI approaches for large representative data sets (Text, audio, image).

TPX-AI's data generator is an extension of the *Parallel Data Generation Framework* (PDGF). PDGF is a parallel data generator that can produce large amounts of data for arbitrary schemas. The version of PDGF included in the TPCx-AI kit generates the entire TPCx-AI Test data set. PDGF's internal mechanism allow for fast generation of large data because it leverages all CPUs of a systems by running in parallel.

PDGF generates a diverse and representative data set starting at 1GB data set size up to 10TB data set size. It does that by scaling a base data set to larger data sets that resemble the same characteristic as the base data set. The base data set was assembled from different sources and contains different formats.

The data management stages, and data science pipeline modeled by the benchmark address complex business problems with the intent to answer specific business questions. The data management stages like cleansing, exploration and preprocessing mimic modern commercial pipelines that are used in current production environments Specifically, the data management stages employ training, serving and scoring phases.

Because of TPCx-AI being an end-to-end benchmark running a realistic workload on realistic data, the measured price-performance metric and the maintenance requirements for each result can be used to compare the viability of the commercially available solutions.

TPCx-AI includes 10 use cases. Each use cases defines a single problem solved by the *Deep Learning* and *Machine Learning Data Science Pipeline*. Almost all use cases included in TPCx-AI include data generation, data management, training, scoring and serving phases. The use cases describe real customer problems including customer segmentation, customer conversation transcription, sales forecasting, spam detection, price prediction, classification and fraud detection. Each use case is framework and syntax agnostic and can be implemented in many ways to extent the TPCx-AI kit to more frameworks.

The design and flow of the TPCx-AI kit makes it very easy to run the entire benchmark in a standard single-server configuration or in a cluster configuration. Configured as a single-server TPCx-AI can be used to measure performance of scale-up configurations. When configured as a clustered-server TPCx-AI can be used to measure performance of scale-out configurations. Once setup, the user can specify additional parameters that can be tuned as specified by the TPCx-AI specification that can improve the performance of the solution. TPCx-AI's kit also includes a *Test Run*, which consists of a set of tests that can be used for system analysis and tuning.

All TPCx-AI results are audited to certify compliance with the spirit and letter of the TPCx-AI Benchmark Standard by a TPC Certified auditor or a PPB.

Each benchmark run of the kit starts with a *Load Test*, followed by a *Power Training Test*, two *Power Serving tests*, a *Srocing test* and a *Throughput Test*. The Load Test is the process of copying the input data set files to the final location from where they will be accessed to execute each one of the subsequent benchmark phases. The Power Training Test measures the speed that the SUT

processes the training phase of all 10 use cases. The result of the training test are the training model files for each of the use cases. The Power Serving tests measure the speed that the SUT process the serving stages of all 10 use cases. The Scoring Test, a separate serving phase, is executed in sequence for all 10 use cases against a newly generated data (excluding the truth labels) and the resulting labels from separate serving phase are compared with the ground truth labels to determine the accuracy metric or whether an error incurred by any use case. The Throughput Tests measure the ability of the system to process the most serving use cases in the least amount of time with multiple users.

TPCx-AI, as all TPC benchmarks, defines three primary metrics, the performance metric, AIUCpm@SF, the price-performance metric, $/AIUCpm@SF, and the system availability date. Furthermore, TPCx-AI defines the following secondary metrics. The *Computed Load Metric* T_{LD} is the elapsed time of the Load Test in seconds.

$$T_{LD} = T_{Load} \tag{9}$$

The *Computed Power Training Test Metric* T_{PTT} is the geometric mean of the elapsed times UT of each of the use case training times. $UT(i)$ is the elapsed time in seconds of the use case i during the Power Training Test and N is the number of use cases in the benchmark.

$$T_{PTT} = \sqrt[N]{\prod_{i=1}^{i=N} UT(i)} \tag{10}$$

The *Computed Power Serving Test Metric* $TPST$ is the maximum of two power run metrics T_{PST1} and T_{PST2}.

T_{PSTn} is the geometric mean of the elapsed time US in seconds of each of the Use Case Serving times as measured during the Serving Power Test $n \in 1, 2$.

$$T_{PSTn} = \sqrt[N]{\prod_{i=1}^{i=N} UTn(i)} \tag{11}$$

$$T_{PST} = max(T_{PST1}, T_{PST2}) \tag{12}$$

T_{TT} is the throughput test metric computed as the total elapsed time of the throughput test T_{TPUT} divided by the number of streams, S and the number of use case in the Performance Test, N as measured during the throughput test.

$$T_{TT} = \frac{T_{TPUT}}{N * S} \tag{13}$$

Using Eqs. 9 to 13 the primary performance metric is defined as:

$$AIUCpm@SF = \frac{SF * N * 60}{\sqrt[4]{T_{LD} * T_{PTT} * T_{PST} * T_{TT}}} \tag{14}$$

5.1 Allowing Cloud Based Benchmark Publications

When cloud computing became mainstream, it seemed straight forward to say that TPC benchmarks are ready to run in the cloud. However, like in many other situations, the devil is in the detail. One of the largest differences between publishing benchmarks in the cloud compared to on-prem is pricing.

Prior to 2005 each TPC benchmark defined its own pricing rules. In 2005 the TPC released the pricing specification, which consolidated most of the pricing wording into a separate document that each benchmark referred to. The pricing specification ended up defining a new benchmark category, named *common benchmarks*. Another benchmark in this category is the TPC-Energy specification, released in 2012. With the second major release of the pricing specification in November 2014, the TPC introduced wording to support the pricing of TPC benchmarks ran in cloud environments.

6 Conclusion

In this paper I gave an overview of how the TPC has been re-inventing itself over the course of 30 years. Starting with benchmarks for only one business use case (OLTP), the TPC developed benchmarks for other business uses cases, i.e. DSS, web commerce, virtualization, internet of things and artificial intelligence, as computer systems were deployed in those business use cases. The TPC also introduced new benchmarks specifications for aging benchmark specifications, such as TPC-E for TPC-C, TPC-DS for TPC-H. The TPC is constantly looking at how it can better develop benchmarks, which led to the introduction of express class benchmarks and the open source initiative, that is trying to make the TPC more attractive for new developers being used to the open source approach to software and benchmark development. Lastly we have shown how the TPC reacted to new compute environments, such as the cloud.

References

1. https://www.tpc.org/tpcx-hs/default5.asp
2. https://www.tpc.org/TPC_Documents_Current_Versions/pdf/TPCX-HS_v2.0.3.pdf
3. https://www.tpc.org/TPC_Documents_Current_Versions/pdf/TPC-Energy_v1.5.0.pdf
4. Bitton, D., et al.: A measure of transaction processing power. Datamation **31**(7), 112–118 (1985)
5. Ghazal, A., et al.: BigBench: towards an industry standard benchmark for big data analytics. In: Ross, K.A., Srivastava, D., Papadias, D. (eds.) Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, 22–27 June 2013, pp. 1197–1208. ACM (2013). https://doi.org/10.1145/2463676.2463712
6. 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). https://doi.org/10.1007/978-3-642-10424-4_7

7. Huppler, K., Johnson, D.: TPC express – a new path for TPC benchmarks. In: Nambiar, R., Poess, M. (eds.) TPCTC 2013. LNCS, vol. 8391, pp. 48–60. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04936-6_4
8. Nambiar, R.O., Poess, M.: The making of TPC-DS. In: Dayal, U., et al. (eds.) Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, 12–15 September 2006, pp. 1049–1058. ACM (2006). http://dl.acm.org/citation.cfm?id=1164217
9. Nambiar, R., et al.: Introducing TPCx-HS: the first industry standard for benchmarking big data systems. In: Nambiar, R., Poess, M. (eds.) TPCTC 2014. LNCS, vol. 8904, pp. 1–12. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15350-6_1
10. Poess, M., Nambiar, R., Kulkarni, K., Narasimhadevara, C., Rabl, T., Jacobsen, H.: Analysis of TPCx-IoT: the first industry standard benchmark for IoT gateway systems. In: 34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, 16–19 April 2018, pp. 1519–1530. IEEE Computer Society (2018). https://doi.org/10.1109/ICDE.2018.00170
11. Poess, M., Rabl, T., Jacobsen, H., Caufield, B.: TPC-DI: the first industry benchmark for data integration. Proc. VLDB Endow. **7**(13), 1367–1378 (2014). https://doi.org/10.14778/2733004.2733009. http://www.vldb.org/pvldb/vol7/p1367-poess.pdf
12. Pöss, M., Floyd, C.: New TPC benchmarks for decision support and web commerce. SIGMOD Rec. **29**(4), 64–71 (2000). https://doi.org/10.1145/369275.369291
13. Raab, F.: TPC-C - the standard benchmark for online transaction processing (OLTP). In: Gray, J. (ed.) The Benchmark Handbook for Database and Transaction Systems, 2nd edn. Morgan Kaufmann (1993)
14. Rabl, T., Poess, M., Danisch, M., Jacobsen, H.: Rapid development of data generators using meta generators in PDGF. In: Narasayya, V.R., Polyzotis, N. (eds.) Proceedings of the Sixth International Workshop on Testing Database Systems, DBTest 2013, New York, NY, USA, 24 June 2013, pp. 5:1–5:6. ACM (2013). https://doi.org/10.1145/2479440.2479441
15. Sethuraman, P., Reza Taheri, H.: TPC-V: a benchmark for evaluating the performance of database applications in virtual environments. In: Nambiar, R., Poess, M. (eds.) TPCTC 2010. LNCS, vol. 6417, pp. 121–135. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18206-8_10
16. Taheri, H.R., Little, G., Desai, B., Bond, A., Johnson, D., Kopczynski, G.: Characterizing the performance and resilience of HCI clusters with the TPCx-HCI benchmark. In: Nambiar, R., Poess, M. (eds.) TPCTC 2018. LNCS, vol. 11135, pp. 58–70. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11404-6_5
17. TPC, T.P.P.C.: TPC-A benchmark specification version 2.0.0 (1999). https://www.tpc.org/TPC_Documents_Current_Versions/pdf/tpca_v2.0.0.pdf
18. TPC, T.P.P.C.: TPC-App benchmark specification version 1.3.0 (1999). https://www.tpc.org/TPC_Documents_Current_Versions/pdf/tpc-app_v1.3.0.pdf
19. TPC, T.P.P.C.: TPC-B benchmark specification version 2.0.0 (1999). https://www.tpc.org/TPC_Documents_Current_Versions/pdf/tpc-b_v2.0.0.pdf
20. TPC, T.P.P.C.: TPC-C benchmark description (1999). https://www.tpc.org/tpcc/default5.asp
21. TPC, T.P.P.C.: TPC-D benchmark specification version 2.1.0 (1999). https://www.tpc.org/TPC_Documents_Current_Versions/pdf/tpcd_v2.1.0.pdf-09-30
22. TPC, T.P.P.C.: TPC-H benchmark description (1999). https://www.tpc.org/tpch/default5.asp

23. TPC, T.P.P.C.: TPC-R benchmark specification version 2.1.0 (1999). https://www.tpc.org/TPC_Documents_Current_Versions/pdf/tpcr_v2.1.0.pdf
24. TPC, T.P.P.C.: TPCx-BB benchmark description (1999). https://www.tpc.org/tpcx-BB/default5.asp
25. TPC, T.P.P.C.: TPCx-HCI benchmark description (1999). https://www.tpc.org/tpcx-HCI/default5.asp
26. TPC, T.P.P.C.: TPCx-IoT benchmark description (1999). https://www.tpc.org/tpcx-IoT/default5.asp
27. TPC, T.P.P.C.: TPCx-V benchmark description (1999). https://www.tpc.org/tpcx-V/default5.asp