# Position Statement: The Case for a Visualization Performance Benchmark

Leilani Battle* University of Washington
Remco Chang† Tufts University
Jeffrey Heer‡ University of Washington
Michael Stonebraker§ MIT

**ABSTRACT**

In this paper, we discuss the need for a new visualization performance benchmark. Performance benchmarks have been developed by both the database and visualization communities, but for completely different purposes, and as a result these benchmarks fail to address the key factors involved in evaluating visual data systems. We describe a core set of 3 design goals in developing a new benchmark: ease of customization, to support different system designs and architectures; ease of interpretation of benchmark results, to ensure fair and transparent comparisons across visualization systems; and realistic scenarios, to ensure that the benchmark reflects real-world use cases in visual analytics. We then propose methods to develop the dataset, queries, and evaluation metrics for a new data-visualization management benchmark, guided by past benchmarks and our design considerations.

**Index Terms:** K.6.1 [Management of Computing and Information Systems]: Project and People Management—Life Cycle; K.7.m [The Computing Profession]: Miscellaneous—Ethics

## 1 INTRODUCTION

Visualizations are invaluable in the data analysis process, as they enable scientists to explore and interpret billions of datapoints quickly, with just a few rendered images. However, many visualization systems are unable to keep up with the rapid accumulation of data through remote sensors, field sensors, medical and personal devices, social networks, and more. This is due to certain assumptions that many of these systems rely on, such as the assumption that these systems can store entire datasets directly in main memory. With so many massive datasets available, ranging from NASA MODIS satellite imagery [3] to the Internet Movie Database [4] to Twitter streams [1], this assumption no longer matches reality.

In response, new breed of visualization system has become the norm, where the data resides in a database management system (or DBMS) running on a remote server, and a visualization front-end running on a client machine (e.g., a laptop) issues queries to retrieve data from the DBMS (henceforth referred to as **visual data systems**). Recently, visual data systems have focused on enabling interactive exploration of large datasets, where the user observes only low latencies (i.e. 500ms of latency or less) when performing interactions within the visualization front-end. To support this level of interactivity, these systems utilize a variety of strategies, including pre-computation [6, 22, 23, 27], sampling [15, 17, 19, 28], and predictive query execution [5, 6, 9, 20].

However, given the diversity of techniques, domain problems, system architectures, software platforms, etc., it has been difficult for the visualization community to compare these techniques and decide which one is most suited for their needs. For example, while sampling is intuitive and easy to use, it also introduces uncertainty in the resulting visualization. Conversely, pre-computation and prefetching techniques can provide precise answers, but at the cost of high storage requirement and runtime data transfer.

In this paper, we propose a new benchmark that allows for systematic and repeatable measurement of visual data systems. Our benchmark is inspired by the different ways that the database and visualization (and visual analytics) communities evaluate systems. For example, the standard approach to evaluating DBMSs is to run a *benchmark* using each DBMS in question, such as the TPC-H benchmark [13], and compare the results (e.g., the average response time or latency). However, database benchmarks are designed for specific use cases, like data warehousing (i.e., TPC-H), genomics [33], and transactional processing [11, 14], which do not include visual analytics as a high-priority use case. As such, they are a poor approximation for gauging visualization performance, and similar calls have been made for new DBMS benchmarks for visualization [16].

In contrast, visualization benchmarks, like the Visual Analytics Benchmark Repository [30], focus on user perception and productivity (i.e., how well and how thoroughly a user can analyze data with a particular visualization tool). As such, they provide limited support for performance evaluations (i.e., how fast the system runs when a user interacts with it), as well as direct comparisons of analytical operations (i.e., comparing performance for specific queries).

We propose that a new benchmark should combine the best of both worlds by blending methodology from the database and the visualization communities. In particular, we suggest the following:

1. ease of customization of the benchmark, to support a variety of visual data system architectures
2. ease of interpretation of benchmark results, to ensure that evaluations use fair and transparent performance measures
3. Realistic scenarios, to ensure that the benchmark accurately represents how users analyze data through visual data systems.

Given the design considerations, in the rest of the paper we discuss a plan for developing the new benchmark, including data, queries, and comparable measures for evaluating visual data systems.

## 2 CURRENT EVALUATION TECHNIQUES

To better understand how a visualization benchmark is helpful, we examine the limitations of existing evaluation methods. We present the different evaluation methods reported in both the database and the visualization communities, including imMens [23], A-WARE [15], and ForeCache [6]. We summarize the commonalities between these evaluation methods, and propose the creation of a unified visualization benchmark based on the integration of these methods.

### 2.1 Evaluation Setups for Modern Visual Data Systems

Here, analyze how recent visual data systems are evaluated. Specifically, we discuss how datasets and workflows are selected for evaluation, and the measures that are used to compare system performance.

**Datasets**: Most datasets are selected or created based on three features: 1) size, 2) complexity (i.e., number of columns and interesting data distribution properties), and 3) relevance (i.e., whether users of the visual data system already have a vested interest in the dataset).

Real-world datasets are selected mainly for their size, to test how systems scale, as well as for relevance, to show how systems perform in real world use cases. For example, ForeCache is tested using

*e-mail: leilani@cs.washington.edu
†e-mail: remco@cs.tufts.edu
‡jheer@cs.washington.edu
§stonebraker@csail.mit.edu

NASA MODIS data both in terms of scale (e.g., terabytes of data are processed) and applicability to users (e.g., earth scientists are recruited for evaluation) [6]. Synthetic datasets are created primarily to test systems under various distribution-related conditions (e.g., varying distributions within and across data columns [20,21,23,34]).

An ideal dataset merges the best features of real-world and synthetic datasets: it would have direct real-world applications, and interesting size and distribution properties. Unfortunately, finding a single dataset with all of these features is difficult. Instead, it makes more sense to find several real-world datasets that share some of these properties, and create data generators to mimic them. This approach is common for database benchmarks [13, 33], and thus lends itself well to the development of a visualization benchmark.

**Workflows/Workloads**: In addition to evaluating visual data systems based on the size and complexity of the input data, another common criteria is to test its performance across different usage scenarios, or **workflows** or **workloads**. Two methods are used to produce a workflow or workload for evaluation:

1. interaction logs (or DBMS query logs) are collected (e.g., through a user study [5, 6, 20], or retrieved from an existing evaluation [20]) and used to drive performance experiments
2. a user workflow is manually created and translated into a log of interactions (or DBMS queries) for evaluation [10, 15, 23, 34]

The first method is restricted to the specific system and dataset used to generate the logs, and thus may have limited applicability. Furthermore, this technique requires significant effort (i.e., conducting a user study) to produce usable results. However an advantage is that performance gains demonstrated with this evaluation method are strongly supported by real-world use cases.

We have found the second method (manually creating a workload) to be more popular for two reasons. First, any system can be evaluated using any reasonable dataset, providing wide applicability. Second, because a user study does not have to be conducted, visual data systems can be evaluated much faster. However, since real users are missing from the evaluation, it is more challenging to argue for strong performance for real-world applications.

A better approach could be to analyze interaction logs from a study with real users, and then create realistic (but synthetic) workflows from the logs [15], similar to how DBMS benchmarks are developed [12,13,33]. However, the challenge here will be to ensure that the synthetic workflows accurately represent visual analytics tasks *in general*, opposed to tasks specific to a single system.

**Evaluation Measures**: Given a **dataset** and a **workload**, we have found time to be the standard measure used to evaluate performance for visual data systems. However, the consideration of time can applied to different steps of the data analysis process:

1. System response time: similar to measuring query speeds for DMBSs, a system's response time to a user's interaction is often used to gauge the performance of a visual data system.
2. "Cold-start" time: given how fast-paced and varied data analysis tasks can be, it is important to consider how quickly a user can begin to explore a new dataset with a visual data system. This initialization process includes the pre-computation time of the system, or the time required to build supplemental data structures that are needed to drive optimizations (e.g., samples, machine learning models, indexes, and data cube structures).

While the measure of system response time is common place, surprisingly, we found cold-start time to be largely ignored in performance evaluations[1]. For data cube-like structures, pre-computation time could take hours [5, 22, 23, 27], which can have a significant impact on how a user interacts with the system. Even sampling techniques can have a long pre-computation time when executed over massive datasets. As such, we need metrics that represent a holistic (and more realistic) view of the visual analysis process. A visualization benchmark would act as a centralized point for discussion of new and relevant evaluation measures, as well as provide clear and well-documented measures for evaluating visual data systems.

[1]except for Nanocube [22], Hashedcube [27], and Sculpin/ForeCache [5]

### 2.2 Methods for Comparing Visual Data Systems

Visual data systems are typically compared using two different methods. These comparison methods are as follows (for a new system named System A, and an existing competitor named System B):

1. Techniques from System B are re-implemented in System A to make comparisons (e.g., the method used to evaluate ForeCache [6], imMens [23], SeeDB [34], and DICE [20]);
2. Systems A and B are run directly with equivalent experimental settings, and their outputs are compared (e.g., the method used to evaluate A-WARE [15], and standard method for DBMSs).

However, most evaluations favor the first comparison method (i.e., to re-implement system logic) over the second (i.e., directly running other systems). This could be due to the difficulty of acquiring and then running the code for competing systems, as well as issues in acquiring the dataset(s) used to evaluate these systems. Ideally, we should provide tools that others can easily use and build upon. If other database and visualization experts are unable to run these systems, it is even less likely that non-experts will use them.

Furthermore, the systems mentioned above are generally compared to at most two other systems using re-implementation, and it is unlikely that this method will scale. Due to the human-in-the-loop aspect to visual data systems, it is also challenging to replicate the results of these evaluations. Given the growing interest in cross-area collaborations between the database and visualization communities, the number of visual data systems within this space will only increase, and rapidly. Expecting researchers to re-implement every new visual data system that comes out is unrealistic, and becomes more outrageous as the community continues to grow.

Ideally, systems would be compared directly using the same dataset, workload, and performance measures (i.e., using the first comparison method). But given the large number of possible datasets to use, the plethora of use cases supported by different visual data systems, and variability of evaluation methods, selecting a single experimental setup appears to be a daunting task.

However, we have seen this comparison method used frequently in the database community, in particular the TPC benchmarks [26]. Given that the performance evaluations done for visual data systems are similar to those utilized to evaluate DBMSs, a visualization benchmark appears to be a viable alternative to current evaluation methods. By adhering to a widely-accepted benchmark, we can encourage our community to produce easy-to-use systems, in turn supporting wider-spread usage of existing systems and cleaner performance comparisons for new systems that are easier to replicate.

### 2.3 The Need for a Standardized Benchmark

With the variety of methods used to measure performance, it is extremely difficult to objectively compare one visual data system with another, based on reported performance results. We described 3 dataset selection/creation methods, two workload creation methods, and two system comparison methods, resulting in 12 possible evaluations, none of which can be directly compared with another. This problem will only worsen as our community continues to grow. However, we also see that these methods can be merged, and that the resulting hybrid methods share strong similarities with the design of existing database benchmarks. We believe this provides strong evidence not only for the need but also the viability of a new visualization performance benchmark.

A natural starting point is to see if existing benchmarks could be re-purposed as a visualization performance benchmark. In the next section, we discuss the pros and cons of existing benchmarks in the database and visualization communities, and how design decisions

from these benchmarks could be leveraged to develop a visualization performance benchmark.

## 3 Past Benchmarks

Our goal is to develop a unified performance benchmark that enables systematic and repeatable measurement of visual data systems within a realistic environment. As a first step, we review existing methods in the database and visualization communities for developing realistic benchmarks, and identify key properties that can be propagated to the design of a new visualization performance benchmark.

### 3.1 Database Benchmarks

The database community has a long tradition of publishing and utilizing performance benchmarks. For example, the Transaction Processing Performance Council [26] was founded in 1988 to develop benchmarks that provide "objective, verifiable performance data to the industry" [26]. It has since developed benchmarks that are considered the gold standard for evaluating DBMSs for transactional processing (TPC-C [11, 14], TPC-E [11]), online analytical processing (TPC-H [13]), and now virtual environments (TPC-V [31]).

The TPC-H benchmark is of particular interest, which simulates a data warehouse providing decision support for a retail company. TPC-H queries are a mix of analytical queries, for monitoring the warehouse(s), and update queries, for simulating real-time dataset maintenance. This benchmark is a popular evaluation tool for DBMSs supporting Online Analytical Processing (or OLAP). OLAP queries mainly feature aggregation operations to compute statistics, such as computing the count or mean for a given data attribute, and thus share significant overlap in the operations used in visual analytics tasks (e.g., aggregation for bar charts, box plots, and heatmaps).

### 3.2 Visualization Benchmarks

From the visualization community, we focus on the Visual Analytics Benchmark Repository [30], which provides the data, submissions and solutions of past VAST and InfoVis Challenges. What makes this benchmark unique is the availability of ground truth for existing analysis tasks, across several different datasets (i.e., the *solutions* for each Challenge). From this information, one can calculate the accuracy of the answers submitted to the VAST Challenges (which are also part of the benchmark), and by extension evaluate the effectiveness of the visualization tools used to produce these answers.

The majority of VAST/InfoVis challenges, including the last four VAST Challenges, involve analyzing both hand-made and code-generated data. However, a small number of competitions used real-world data instead. For example, the 2006 InfoVis Challenge utilized 2000 Census Data [8] and the InfoVis 2007 Challenge used a subset of the Internet Movie Database [4].

Interestingly, this definition of benchmark in the visualization community (measures accuracy of analyses derived using a visualization tool) deviates from that of the database community (measures DBMS speed and throughput for a known set of queries). However, the data derived for the Visual Analytics Benchmark Repository is still applicable to a performance-driven benchmark.

### 3.3 Comparing the Benchmarks

Here, we discuss the positive and negative aspects of the provided benchmarks with respect to performance, as well as opportunities to bring the benchmarks together in an effort to develop a more effective performance evaluation for visual data systems.

We identify three major limitations to existing benchmarks that make them unsuitable for a visualization performance benchmark:

1. a lack of explicit analysis operations (or queries) for analysis tasks (Visual Analytics Benchmark Repository)
2. a lack of realistic use cases for visual analytics (TPC-H)
3. a lack of flexibility in the benchmark due to partial reliance on hand-made data (Visual Analytics Benchmark Repository)

**Lack of Explicit Analysis Operations**: While the Visual Analytics Benchmark Repository is an interesting candidate for evaluating visual data systems, it falls short because the Visual Analytics Benchmarks were designed to allow flexible analysis workflows. As such, the solutions provided in the Benchmark Repository often only contain vague descriptions of the analysis process, or the answer to expected results (e.g. describing what the "outlier" is and why).

While this flexibility serves the visual analytics community well, as a benchmark for visual data systems, it leaves the exact analysis steps needed to produce the appropriate results up to interpretation. Without a well-specified workflow, it is extremely difficult to compare the performance of two systems. This is akin to comparing the performance of two different DBMSs that are executing different queries. In contrast, the TPC benchmarks have published query sets, so database vendors and researchers know exactly what operations must be supported to run them. This also provides flexibility in how the TPC-H benchmark is utilized. For example, if some operations are not supported, one can still report on the *subset* of TPC-H queries their DBMS can run, which still provides valuable information about the performance of the DBMS, as well as its limitations.

**Lack of Realistic Use Cases**: A major drawback to the TPC-H benchmark for visual analytics is the fact that it simulates a data warehouse. Though this is certainly a valid visual analytics use-case (at least for industry), it is far from representative of the challenges and tasks that the visualization community aims to address. In addition, the TPC-H benchmark schema and queries are not representative of how a visualization tool issues queries to produce interactive visualizations. In comparison, the Visual Analytics Benchmark Repository is designed to simulate real-world visual analytics use cases.

**Lack of Flexibility**: Most benchmarks generate at least some input data through code (e.g., the Threat Stream Data Generator [35] and TPC-H data generator [29]). However, most of the benchmarks in the Visual Analytics Benchmark Repository are small (a few GB, or less), and rely on handmade data, limiting their ability to scale. The TPC-H benchmark in comparison is fully code-generated, and includes a scale factor parameter to increase the dataset size.

### 3.4 Useful Properties

Even though existing benchmarks are unsuitable for evaluating the performance of visual data systems, they have useful properties that could be transferred to a new benchmark, including: 1) dataset customization, 2) an explicit workload (e.g., specific queries to be executed), and 3) realistic visual analytics tasks. We expand on these ideas in the following section with a set of high level design considerations for a future visualization performance benchmark.

## 4 Benchmark Design Considerations

Using our analysis of evaluation methods from the database and visualization communities, we present three high-level design goals for a new benchmark: 1) ease of interpretation, 2) ease of customization, and 3) realistic scenarios. In the rest of this section, we explain each major design consideration. In the rest of the paper, we refer to our proposal as the ***Visualization Performance Benchmark***.

### 4.1 Ease of Interpretation

A major challenge in evaluating the performance of different visual data systems is finding common ground on which to make a direct comparison. A single benchmark helps system designers to focus on a clear set of goals for improving system performance. The designer is also able to gauge the impact of their techniques by calculating how many queries (i.e., analysis operations, workflows) in the benchmark are made faster using the new techniques.

Furthermore, a designer should be able to easily identify explicit strengths and weaknesses in their visualization system within the context of the Visualization Performance Benchmark (e.g., which queries run faster, and which run slower compared to other systems).

After running the benchmark, the performance results should also be straightforward to interpret, which necessitates providing thorough documentation for all queries in the benchmark, including for each query: 1) the user interface interactions that are covered by this particular query (and how this mapping is developed); and 2) the optimization areas that are covered by this query.

### 4.2 Ease of Customization

Each visualization system is designed to support a unique set of dataset types and user interface features. As such, the Visualization Performance Benchmark should be configurable, to suit different system and architecture needs. For example, several systems build data-cube structures to improve performance, but some rely primarily on main memory [22, 27], while others rely primarily on disk storage [5, 6, 23]. There are clear differences in storage utilization and computing needs between these different systems, and the benchmark dataset should be tuned accordingly.

Data distribution factors also play a role in system evaluation, such as dataset skew and correlations between data columns. For example, skewed data can cause a slow-down in DBMS performance if not carefully distributed across multiple machines [32], and can affect the speed of convergence of approximate query processing (or AQP) techniques. AQP has become a popular technique among recent visual data systems [15, 17, 20, 24, 25]. Several visualization systems, including imMens [23] and DICE [20], are also evaluated under a variety distribution conditions. As such, parameterizing these dataset conditions is a critical use case for the benchmark.

### 4.3 Realistic Scenarios

The most important feature of the Visualization Performance Benchmark will be its ability to simulate a broad range of real-world visual analytics use cases. The closer the approximation to real world use cases, the more one can rely on the results from the benchmark as being indicative of a system's performance with real users. These use cases should also encompass a reasonable set of interactions within a user interface, dictated by the set of user interactions featured in existing visual data systems. While a complete set of interactions will be a future point of research, as a starting point, we propose that the benchmark should include the following set of common data interactions: panning, zooming, filtering/selections, changing of axes (i.e., pivoting), and brushing and linking.

A final consideration lies in the structure of an "analysis session". Specifically, users' analyses tend to occur in concentrated bursts, resulting in chains of queries (generated either by the user, or the visualization system) that are often related to one another. User interaction logs are known to be a rich source for understanding and learning behavioral patterns [7, 18], and these patterns can be utilized in optimizations [5, 6]. As such, the Visualization Performance Benchmark will be more powerful if the queries created for the benchmark also follow a session-based structure (i.e., if it incorporates some consistent representation of actual user behavior). Given that there are many ways to analyze a dataset, evidenced by the diversity of submissions to the VAST Challenges, we aim to include a diverse set of workflows, per dataset, into the benchmark.

## 5 Proposed Benchmark Implementation

Given our design considerations, we now describe a plan for developing the Visualization Performance Benchmark. In the remainder of this section, we briefly touch on all aspects of the benchmark (dataset, queries and metrics), but focus on the most challenging aspect to developing the benchmark: creating realistic queries.

### 5.1 Dataset and Evaluation Metrics

Here, we briefly discuss our plans for the other components of the benchmark. We will leverage data generation techniques from existing benchmarks to create a fully code-generated dataset (see Section 3.3 for more details). To ensure that the data is supported by realistic scenarios, we can use data from the previous VAST Challenges as the initial input to our data generator. To establish clear evaluation measures, we will incorporate all standard measures utilized in existing evaluation techniques, as well as under-utilized measures such as cold-start time (see Section 2 for more details).

### 5.2 Creating the Queries

To generate queries, we propose the following implementation steps. Given a specific VAST Challenge dataset from the Visual Analytics Benchmark Repository (e.g., VAST Challenge 2012 [2]), we:

1. Select a set of interactions to be evaluated by the benchmark (see Section 4.3 for an initial set of supported interactions).
2. Collect existing workflows for this dataset, ensuring that the workflows are consistent with the supported interactions.
3. Given a list of consecutive analysis steps in the workflows, translate the analysis steps to queries.

Because the VAST Challenge datasets include multiple submissions, we have access to analyses performed by real users. Thus, we can extract a new workflow for each submission made to the competition. By incorporating multiple submissions as separate workflows, we have the opportunity to showcase a diverse set of analysis strategies within the benchmark. Each workflow represents a unique query set that emphasizes different interactions (e.g., relying more on brushing and linking, less on panning and zooming), enabling system designers to easily customize the benchmark to match the interactions supported by different visual data systems.

System designers have three options for using the benchmark: 1) evaluate across all workflows, 2) choose specific workflows and evaluate only the queries in these workflows, or 3) ignore the workflow structure of the benchmark and only choose queries supporting specific interactions (e.g., only filtering queries). In this way, general-purpose systems (i.e., systems that support all interactions from step one) can be compared against the entire benchmark, and specialized systems can be compared against the subset of the benchmark that they support. Note that when evaluating two specialized systems that do not share complete overlap in supported interactions, only the workflows representing the *intersection* of supported interactions can be directly compared. Similarly, performance results from one workflow cannot be directly compared with the results from another, since each workflow represents a different analysis strategy.

## 6 Discussion

By developing a standard for evaluating visual data systems, we can enable system designers to capture and share systematic and repeatable performance measurements for a variety of systems. With a customizable performance benchmark, we can support a typical set of user tasks, as well as relevant datasets and interfaces, observed in real-world use cases in visual analytics.

However, a single benchmark will fail to cover all performance concerns for all systems, use cases and tasks. As such, we see the Visualization Performance Benchmark as a useful starting point, and encourage the development of complementary benchmarks, similar to what has been done in the database community [13, 33].

Furthermore, the end goal of visual data systems is to improve the *user's* analysis performance. Currently, we see the system's performance as a bottleneck, and thus focus on measures like system response time in the Visualization Performance Benchmark. However, as we improve system performance, we must also consider new opportunities to improve the user's overall analysis performance, and thus new (standardized) evaluation methods for visual data systems.


## Acknowledgments

This work was supported in part by the Moore Foundation Data-Driven Discovery Investigator program, and grants NSF 1452977 (or IIS-1452977) and DARPA FA8750-17-2-0107.

## REFERENCES


[1] Public streams – Twitter Developers. https://dev.twitter.com/streaming/public.

[2] VAST Challenge 2012. http://www.vacommunity.org/VAST+Challenge+2012.

[3] LAADS Web – Search for Data Products. https://ladsweb.nascom.nasa.gov/search/?si=Terra%20MODIS&si=Aqua%20MODIS, Nov. 2016.

[4] IMDb. http://www.imdb.com/interfaces, Aug. 2017.

[5] L. Battle. *Behavior-Driven Optimization Techniques for Scalable Data Exploration*. PhD thesis, Massachusetts Institute of Technology, 2017.

[6] L. Battle, R. Chang, and M. Stonebraker. Dynamic Prefetching of Data Tiles for Interactive Visualization. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pp. 1363–1375. ACM, New York, NY, USA, 2016. doi: 10.1145/2882903.2882919

[7] E. T. Brown, A. Ottley, H. Zhao, Q. Lin, R. Souvenir, A. Endert, and R. Chang. Finding waldo: Learning about users from their interactions. *IEEE Transactions on visualization and computer graphics*, 20(12):1663–1672, 2014.

[8] U. C. Bureau. PUMS Data. https://www.census.gov/programs-surveys/acs/data/pums.html, 2017.

[9] U. Cetintemel, M. Cherniack, J. DeBrabant, Y. Diao, K. Dimitriadou, A. Kalinin, and O. Papaemmanouil. Query Steering for Interactive Data Exploration. In *6th Biennial Conference on Innovative Data Systems Research*. Asilomar, CA, USA, Jan. 2013.

[10] S.-M. Chan, L. Xiao, J. Gerth, and P. Hanrahan. Maintaining interactivity while exploring massive time series. In *IEEE Symposium on Visual Analytics Science and Technology, 2008. VAST '08*, pp. 59–66, Oct. 2008. doi: 10.1109/VAST.2008.4677357

[11] S. Chen, A. Ailamaki, M. Athanassoulis, P. B. Gibbons, R. Johnson, I. Pandis, and R. Stoica. Tpc-e vs. tpc-c: characterizing the new tpc-e benchmark via an i/o comparison study. *ACM SIGMOD Record*, 39(3):5–10, 2011.

[12] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears. Benchmarking cloud serving systems with ycsb. In *Proceedings of the 1st ACM Symposium on Cloud Computing*, SoCC '10, pp. 143–154. ACM, New York, NY, USA, 2010. doi: 10.1145/1807128.1807152

[13] T. P. P. Council. Tpc-h benchmark specification. *Published at http://www.tpc.org/tpch/default.asp*, 21:592–603, 2008.

[14] T. P. P. Council. Tpc-c benchmark specification. *Published at http://www.tpc.org/tpcc/default.asp*, 2010.

[15] A. Crotty, A. Galakatos, E. Zgraggen, C. Binnig, and T. Kraska. The Case for Interactive Data Exploration Accelerators (IDEAs). HILDA '16, pp. 11:1–11:6. ACM, New York, NY, USA, 2016. doi: 10.1145/2939502.2939513

[16] P. Eichmann, E. Zgraggen, Z. Zhao, C. Binnig, and T. Kraska. Towards a benchmark for interactive data exploration. *IEEE Data Eng. Bull.*, 39(4):50–61, 2016.

[17] D. Fisher, I. Popov, S. Drucker, and m. schraefel. Trust Me, I'M Partially Right: Incremental Visualization Lets Analysts Explore Large Datasets Faster. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 1673–1682. ACM, New York, NY, USA, 2012. doi: 10.1145/2207676.2208294

[18] H. Guo, S. R. Gomez, C. Ziemkiewicz, and D. H. Laidlaw. A Case Study Using Visualization Interaction Logs and Insight Metrics to Understand How Analysts Arrive at Insights. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):51–60, Jan. 2016. doi: 10.1109/TVCG.2015.2467613

[19] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online Aggregation. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, SIGMOD '97, pp. 171–182. ACM, New York, NY, USA, 1997. doi: 10.1145/253260.253291

[20] N. Kamat, P. Jayachandran, K. Tunga, and A. Nandi. Distributed and interactive cube exploration. In *2014 IEEE 30th International Conference on Data Engineering (ICDE)*, pp. 472–483, Mar. 2014. doi: 10.1109/ICDE.2014.6816674

[21] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pp. 547–554. ACM, New York, NY, USA, 2012. doi: 10.1145/2254556.2254659

[22] L. Lins, J. T. Klosowski, and C. Scheidegger. Nanocubes for real-time exploration of spatiotemporal datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2456–2465, 2013.

[23] Z. Liu, B. Jiang, and J. Heer. imMens: Real-time Visual Querying of Big Data. In *Proceedings of the 15th Eurographics Conference on Visualization*, EuroVis '13, pp. 421–430. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2013. doi: 10.1111/cgf.12129

[24] D. Moritz and D. Fisher. What users don't expect about exploratory data analysis on approximate query processing systems. In *Proceedings of the 2Nd Workshop on Human-In-the-Loop Data Analytics*, HILDA'17, pp. 9:1–9:4. ACM, New York, NY, USA, 2017. doi: 10.1145/3077257.3077258

[25] D. Moritz, D. Fisher, B. Ding, and C. Wang. Trust, but verify: Optimistic visualizations of approximate queries for exploring big data. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pp. 2904–2915. ACM, New York, NY, USA, 2017. doi: 10.1145/3025453.3025456

[26] R. Nambiar, N. Wakou, F. Carman, and M. Majdalany. Transaction Processing Performance Council (TPC): State of the Council 2010. In *Performance Evaluation, Measurement and Characterization of Complex Systems*, Lecture Notes in Computer Science, pp. 1–9. Springer, Berlin, Heidelberg, Sept. 2010. doi: 10.1007/978-3-642-18206-8_1

[27] C. A. Pahins, S. A. Stephens, C. Scheidegger, and J. L. Comba. Hashedcubes: Simple, Low Memory, Real-Time Visual Exploration of Big Data. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2016. doi: 10.1109/TVCG.2016.2598624

[28] Y. Park, M. Cafarella, and B. Mozafari. Visualization-aware sampling for very large databases. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp. 755–766, May 2016. doi: 10.1109/ICDE.2016.7498287

[29] D. Phillips. tpch-dbgen: TPC-H dbgen, July 2017. original-date: 2012-01-18T19:28:20Z.

[30] C. Plaisant, J. D. Fekete, and G. Grinstein. Promoting Insight-Based Evaluation of Visualizations: From Contest to Benchmark Repository. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):120–134, Jan. 2008. doi: 10.1109/TVCG.2007.70412

[31] P. Sethuraman and H. Reza Taheri. *TPC-V: A Benchmark for Evaluating the Performance of Database Applications in Virtual Environments*, pp. 121–135. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi: 10.1007/978-3-642-18206-8_10

[32] R. Taft, E. Mansour, M. Serafini, J. Duggan, A. J. Elmore, A. Aboulnaga, A. Pavlo, and M. Stonebraker. E-store: Fine-grained elastic partitioning for distributed transaction processing systems. *Proc. VLDB Endow.*, 8(3):245–256, Nov. 2014. doi: 10.14778/2735508.2735514

[33] R. Taft, M. Vartak, N. R. Satish, N. Sundaram, S. Madden, and M. Stonebraker. GenBase: A Complex Analytics Genomics Benchmark. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pp. 177–188. ACM, New York, NY, USA, 2014. doi: 10.1145/2588555.2595633

[34] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. SeeDB: Efficient Data-driven Visualization Recommendations to Support Visual Analytics. *Proc. VLDB Endow.*, 8(13):2182–2193, Sept. 2015. doi: 10.14778/2831360.2831371

[35] M. A. Whiting, W. Cowley, J. Haack, D. Love, S. Tratz, C. Varley, and K. Wiessner. Threat stream data generator: Creating the known unknowns for test and evaluation of visual analytics tools. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, pp. 1–3. ACM, New York, NY, USA, 2006. doi: 10.1145/1168149.1168166