Not just correct, but correct and fast

A look at one of Jim Gray's contributions to database system performance

David J. DeWitt
Microsoft Jim Gray Systems Lab
Madison, Wisconsin
dewitt@microsoft.com

Charles Levine
Microsoft Corporation
Redmond, Washington
charles.levine@microsoft.com

ABSTRACT

This paper examines Jim Gray's role in the specification of the debit/credit benchmark. The publication of this benchmark in a 1985 paper launched a benchmark war among the vendors that resulted in dramatic improvements in database system performance in the years following its publication. It was the genesis of the TPC, an industry consortium which has reshaped the benchmark landscape. Descendents of this benchmark continue to this day to be an important metric of modern transaction processing systems.

1. INTRODUCTION

Jim received the Turing award in 1998 for his fundamental contributions to our understanding of the concept of transactions and the mechanisms for their implementation using write-ahead logging and two-phase locking. These mechanisms have proven to be absolutely critical to the ubiquitous adoption of database system technology for managing information in today's data-centric economy.

However, in addition to teaching us how to make sure that database systems could insure that the data they managed was always "correct", Jim also was instrumental in making sure the systems were "fast". By specifying the key metric for evaluating the performance of database systems in his 1985 paper *A Measure of Transaction Processing Power*, [1] Jim launched a benchmark war that drove the industry forward at a frantic pace for more than 15 years. At the time of the publication of this seminal paper, database systems that could deliver 100 transactions/second were considered state of the art. Obtaining 1,000 transactions/second was viewed as unreachable. Twenty years later, Jim was able to obtain 8,000 transactions/second on his laptop. While advances in hardware certainly played a role, the dramatic improvements achieved were also the result of improvements in the database software.

We begin with some historical background that led Jim to develop the benchmark. After describing the benchmark itself, we explore how the benchmark has impacted the development of database systems as the benchmark was adopted, expanded, and refined. More than 23 years later descendents of this benchmark continue to drive the industry forward.

2. BACKGROUND

Jim's foray into database benchmarking was prompted by our attempts to develop a benchmark for evaluating the performance of relational database systems [2]. The approach we had adopted involved measuring the performance of a set of basic SQL operations (e.g. selections, joins, aggregates, and updates) on a synthetically generated database. Our motivation in designing the Wisconsin benchmark was primarily as a vehicle for evaluating the basic relational operators from which complex queries are composed.

While our benchmarking efforts produced some interesting -- and controversial -- results about the state of the art of the commercial relational products in 1983, Jim was convinced that we had gone about measuring performance in totally the wrong way. Jim's response was to author a paper titled: *A Measure of Transaction Processing Power* [1, 3]. The last line of the paper says it all: "There are lies, damn lies, and then there are performance measures." To this day this statement remains true. Except for IBM, the major database vendors require obtaining their permission before publishing benchmark results and rumors of "benchmark specials" – versions of products tuned for specific benchmarks – are commonplace despite the best efforts of the Transaction Processing Performance Council.

The author list and publication date of this paper say a lot about Jim as a person. First, Jim used "Anon et al." as the author of the paper. Having seen the controversy that our benchmarking paper had generated, Jim needed to protect the names of those co-authors who had supplied results and thought using Anon et al. as the author was funny. Second, he loved to share the credit for his work with others. While he was the one who designed the benchmark and wrote the paper pretty much by himself, he cites "24 computer professionals as contributing including eight academics, two end users, and 14 who worked for various vendors."

Jim carefully selected the publication venue and publication date for the paper. Rather than sending it to an academic conference or journal he wanted the paper seen by a much wider range of readers. So he elected to send the paper to Datamation, which, at the time, was one of the leading publications catering to IT professionals. Today, he would have just posted it on his blog.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

But the last little inside joke was the publication date. We remember distinctly that he called all excited because he had arranged to have the paper published on April 1st (1985). The paper was no April Fool's joke. It changed the entire database industry, driving the field forward for more than 15 years.

3. THE ANON ET AL. BENCHMARK

While many think of the benchmark as defining a single metric, the debit/credit benchmark measured in transactions executed per second, the paper actually defines two additional tests: a sort benchmark and a scan benchmark. Unlike the earlier Wisconsin benchmark that can be viewed as a micro-benchmark designed to measure the performance of both individual SQL operators and a set of simple queries, two of the three tests in the Anon et al. Benchmark were designed to capture the essence of a common database application. The debit/credit test was designed to mimic a typical banking transaction. The scan benchmark was modeled after the process a typical company might use to generate 1/30th of its bills for mailing each night of the month. Since sorting is an important component of any database system, Jim elected to use a sort benchmark to measure the raw performance of the database system being tested.

In addition to these three tests, the benchmark also proposed a way of normalizing the differences in the systems being evaluated. For example, if system A has 10 times the throughput of system B but costs 100 times as much to own and operate, system B is clearly most cost effective.

3.1 The Debit/Credit Benchmark

The debit/credit benchmark was designed to mimic the sequence of actions that occur when a customer makes a withdrawal or deposit at a bank. Its design was motivated by a large bank that wanted to acquire a computer and database system that would enable it to put its 1,000 branches, 10,000 tellers, and 10 million customer accounts on line in the early 1970s. As part of the RFP, the bank specified a performance target for the system of 100 transactions/second (tps) with 95% of the transactions having a response time of less than 1 second and an overall system availability of 99.5%. This RFP was the basis from which Jim formalized the design of what was initially known as the TP1 benchmark[1].

The database for the benchmark is quite simple and is composed of three record types: one to model the account for the branch, one to model the teller's account, and one to model the customer's account. The transaction is equally simple (simplicity is always appealing) and is shown in Figure 1 below.

After updating the account record (to reflect a withdrawal or deposit), an auditing record is appended to a History file, which retains all such records for the most recent 90 days. Then it updates the teller and branch records to reflect that a particular teller processed a transaction for the customer's account.

```
Begin Transaction
  Read message from the teller's terminal
  Read and then update the account record specified
  Append an auditing record to a History file
  Read and then update the appropriate teller record
  Read and then update the appropriate branch record
  Send message to the teller's terminal
Commit Transaction
```

Figure 1: The Debit-Credit Transaction

With a performance target of 100 tps, the benchmark specified that the database should contain 1,000 Branch records, 10,000 Teller records, and 10,000,000 customer accounts. With computers at the time having typically 2-4 MB of main memory, the Branch and Teller tables could be cached in memory, but not the Account table.

Since 100 tps at the time was a difficult goal to achieve, Jim correctly anticipated that some vendors would not be able to meet this target. He also wanted to make sure that vendors did not cheat by shrinking the size of the database so that it fit memory. Thus, he devised a set of scaling rules for the benchmark. For example, if a vendor wanted to assert that its system was capable of 1000 tps, they would have to increase the size of the database by a factor of ten to 10,000 Branches, 100,000 Tellers, and 100M customer accounts. Likewise, a vendor whose product could only achieve 10 tps was allowed to scale the size of the database down by a factor of 10. These scaling rules proved to be critical to the success of the benchmark as they allowed all the vendors to participate, kept them as honest as possible, and kept the target moving as systems got faster and faster.

The Datamation article included the results from running the debit/credit benchmark on a number of commercial products. These results are reproduced in Table 1 below.

SYSTEM	TPS	I/Os	$K/TPS
Lean and Mean	400	6	40
Fast	100	4	60
Good	50	10	80
Common	15	20	150
Funny	1	20	400

Table 1: 1985 Results for the Debit/Credit Benchmark

By promising not to name systems, Jim was able to get vendors to supply results (and be authors). The "Lean and Mean" system provided 4 times the throughput of the "Fast" at only 2.7 times the cost as the "Fast" system and was the most cost effective solution at the time.

Needless to say, the results set off a huge amount of speculation in the community as to which system was which. The vendors, of course, knew which system was theirs. Their customers probably did, too, and the vendors of the Common and Funny

[1] The name "TP1" was an internal IBM code name. It was also sometimes called ET1. Jim explained the origin of the names in the following correspondence with Levine on Jan 23, 1992: *Originally there was TP1-TP7 standing for Transaction Processing benchmark 1-7. TP7 is SCAN, TP1 is DebitCredit. TP1 was coded in DL/1. As IMS evolved, TP1 evolved. Eventually, TP1 was recoded for the new "Eagle" transaction processing system which had a new set of database calls. The resulting transaction profile was called ET1 (Eagle Transaction 1).*

products undoubtedly were put under a lot of pressure from customers to improve their products.

It is interesting that, even in 1985, Jim was aware that customers who repeated the benchmark never obtained the same level of performance as the vendors, a problem that continues to plague the field today. Vendor-supplied results continue to be viewed with suspicion. Whether the problem is that vendors are using "benchmark special" versions of their products or that customers are unable to tune their installations to the same degree as the vendors, it is widely recognized that customers essentially never are able to reproduce results obtained by the vendors. As Jim wrote in 1991 [10] "Put another way, the performance numbers a salesperson quotes are really a guarantee that the product will *never exceed* the quoted performance. Despite these caveats, benchmarks and the concepts that underlie them are important tools in evaluating computer systems' performance and price/performance."

3.2 The Scan Benchmark

The second component of the benchmark was designed by Jim to measure the performance that a typical application-level programmer might obtain from the database system. The benchmark is very simple. It reads and updates every record of a file containing one million, 100 byte records. The benchmark divides the job into 1,000 separate transactions each of which processes 1,000 records. With disks of that period capable of transferring data at 2-3 Mbytes/second, the minimum expected time per transaction was 0.1 seconds. The observed times were much worse, ranging from 1 second to 10 seconds, leaving the vendors a lot of room for improvement. While this benchmark never really received much attention, today we expect scans of tables to run at near-disk speeds.

3.3 The Sort Benchmark

The last component of the benchmark required sorting a file of one million, 100 byte numbers with 10 byte keys. Over time, this benchmark became known as the Datamation Sort benchmark. Jim envisioned this test as a measure of the raw performance of the database system. Sorting has always been an important component of a relational database system both for ordering the results of a query, implementing the distinct operator and as the basis for the sort-merge join algorithm.

The sort benchmark specification was essentially unconstrained with the exception that the input and output files had to be stored sequentially on disk. There were no limitations imposed on the number of CPUs, the amount of memory, or the number of scratch disks employed. With typical computers of the period limited to 2-4MB of memory, the sort required at least two passes. In theory, with a 3MB second/disk, the sort should have required only a minute or two; the observed times ranged from 10 minutes to 10 hours. The results clearly indicated lots of room for improvement.

3.4 Costing Rules

Costing was another factor that Jim introduced in the design of the benchmark. His motivation was to normalize the performance of different systems by somehow incorporating their costs. For example, if system A was twice as fast as system B for a particular benchmark but used four times as much hardware, its cost effectiveness was really 1/2 that of System B. Ideally, the cost of running a benchmark would incorporate the total cost of ownership but Jim struggled with what to include. For example, should personnel costs to run the computer system be included? Or the power consumed? In the end, Jim decided to include the cost of only the hardware and software used, amortized over a five-year lifetime. A benchmark requiring an hour to run is charged a prorated amount of the five-year cost.

Like the rules for scaling the debit/credit transaction benchmark, the costing rules he proposed proved to be an important contribution as it gave vendors wishing to run the benchmark the freedom to pick whatever hardware combination that maximized the overall cost effectiveness of their system. Furthermore, it also reduced the need to use the same hardware platform to obtain comparable results across software vendors. Of course, costing is always ripe for manipulation through discounts on software and hardware.

4. THE AFTERMATH

While most papers take a while to have an impact, publication of the Datamation version of the paper had an immediate impact. Each vendor knew exactly where its product stood compared to its competitors. While the vendor of the "Lean and Mean" system had a lot to crow about, the developers responsible for the "good" and "funny" systems had to be pretty discouraged. The paper launched what was to be a benchmarking war among the major database vendors.

The sort and scan benchmarks were, however, early casualties. It was much easier for the marketing teams to focus on, and sell, a single number. Your system's "tps" rating had a catchy ring to it and could be marketed like mpg ratings for cars.

To keep the process as fair as possible, in 1988 Jim encouraged Omri Serlin and Tom Sawyer to start the Transaction Processing Performance Council (TPC), a coalition of hardware and software vendors [4, 5]. Since the Wisconsin benchmark had caused all the database vendors (except IBM) to add a "no benchmarking allowed" clause (sometimes referred to as the "DeWitt" clause [6]) to their license agreements, results would have to come from the vendors. One of the first actions of the TPC was to agree on a set of benchmarking, costing, auditing, and publication rules that had to be followed to publish a result. An independent audit was "highly recommended" by the TPC, but not required. The TPC made audits by TPC certified auditors mandatory in 1993.

Interestingly, the hardware vendors were also eager to participate in this benchmarking war. Having Oracle run faster on a Sun system than an HP box, drove customers to buy Sun products. To this day, Sun maintains a team of engineers dedicated to tuning Oracle for Sun computers. We expect that every major hardware vendor has a similar effort.

The TPC refined the definition of the original debit/credit benchmark specification, added rules for pricing, ACID, full disclosure, and auditing. The result was dubbed TPC-A and launched in 1989. Jim was personally involved in the TPC-A effort as the Tandem representative for the first year of the TPC. He wrote the ACID clause, a task for which he was uniquely qualified. The ACID clause establishes the rules for transactional semantics in the benchmark. Isolation and Durability were particularly important. Isolation has been the center of some of the biggest battles in the TPC over the years. The durability tests have uncovered countless recovery bugs, even in seemingly mature, well tested products. The ACID clause can be found in every TPC benchmark since TPC-A.

TPC-A was followed a year later by TPC-B, which simplified TPC-A by eliminating the external network and the concept of users. Both TPC-A and TPC-B are direct descendents of Jim's debit/credit benchmark. TPC-A and B brought to an end the wild west of debit/credit benchmarking claims. But rather than dampen the competition, the TPC endowed credibility and legitimacy to the benchmark efforts, which in turn increased the value of winning. Ultimately, the simplicity of TPC-A and B were their undoing [7]. The trivial transaction profile lent itself to benchmark specials.

In 1992, the TPC launched TPC-C. It was the first TPC benchmark without any ties to debit/credit. By 1995, when the TPC-A and TPC-B benchmarks were officially "retired", vendors had reached the 10,000 tpsB level, a simply amazing improvement over a 10-year period.

During this period, progress occurred at an incredible rate. While conceptually simple, it turns out that making the debit-credit benchmark really fast required streamlining all aspects of the DBMS software from the query executor to the I/O system. Sybase's introduction of stored procedures gave a big boost in performance as the entire debit/credit application could be implemented inside the database system, replacing four round-trip messages between the application and the DBMS with a single round trip. As a consequence, Sybase had a significant performance advantage until their competitors added stored procedures to their systems.

Having a single number was not only good for marketing but it was also a great motivational tool to drive engineering teams. If your competitor was twice as fast you had no choice but to try and meet their latest results in your next release. Hardware vendors worked hand-in-hand with the database vendors to ensure that their hardware provided the best platform for running the top products. To this day, vendors use the various TPC benchmarks to verify that changes in a new release of their software have not had an adverse effect on the performance of the system.

Although Jim only directly participated in the TPC for the first year, he remained a big fan. The open competition and full disclosure of *how* results were achieved matched how Jim himself worked in the industry. Further, there were some issues about which Jim was particularly passionate. Transaction isolation was one such issue. In 1993, Jim canceled his other plans at the last moment and flew down to San Diego to attend a TPC meeting where the issue de jure was repeatable read versus read committed isolation. After TPC-C had been released, one database vendor argued strenuously (and repeatedly) that the isolation level be changed from repeatable read to read committed. Jim argued that the lower isolation level allowed incorrect results and compared it to the Intel Pentium floating-point bug which was making headlines at the time. The clarity of Jim's argument and his being the indisputable authority on the subject carried the day.

This benchmarking war did, however, have some negative consequences. The almost total focus on TPC-A and TPC-B results for 10 years allowed the mainstream vendors to mostly neglect the performance of their systems on complex decision support queries. While advances were made during this period in query processing and storage techniques, improvements in query optimization were essentially non-existent. This allowed vendors like Teradata, whose primary focus was decision support and not transactions, to dominate the very large data warehouse market. One wonders what might have been had the vendors not focused solely on the debit/credit part of the Anon et. al. benchmark.

While the commercial vendors ignored both the scan and sort benchmarks, Jim was determined to keep the sort benchmark alive. In 1987 he started a sorting competition that continues to this day [8]. In 1997 the winner of the Datamation sort (sort 1M, 100 byte records with 10 byte keys) was a group at Tandem who won with a time of 980 seconds. The following year Peter Weinberger (who was then at Bell Labs) obliterated this record using a Cray 1 with a time of 28 seconds. Peter's record remained unbeaten until 1993 when Chris Nyberg beat it with a time of 9 seconds using a loaded DEC Alpha and sorting software designed to exploit the use of the Alpha's L1 and L2 caches. Every year the Datamation sort benchmark record dropped until in 2001 a group of students and faculty at Wisconsin lead by Andrea and Remzi Arpaci-Dusseau used a cluster of 32 Linux PCs to perform the sort in less than ½ second. Simply starting a parallel job on 32 clusters in a ½ second is a challenge in itself. At this point, Jim decided that the Datamation sort benchmark had outlived its usefulness and should be retired.

Starting in 1995, Jim expanded the set of sort competitions to include the Minute Sort (how many 100 byte records you can sort in a minute), the Penny Sort (how much can you sort using a penny's worth of hardware), and the Terabyte Sort.

Jim never lost his enthusiasm for this competition as he had a deep appreciation for the software talent required to drive the state of the art in sorting forward. Every year Jim would arrive at the annual SIGMOD conference with trophies in hand to present to the new record holders. We are confident it was one of the highlights of his year.

5. TWENTY YEARS LATER

On April 1st, 2005, the 20th anniversary of the Datamation article, Jim published a paper showing the progress that had been made [9]. Jim decided it would be fun to rerun the original debit/credit benchmark on his laptop to measure first hand the progress that the field had made. Although at the time the TPC-B benchmark had been officially "retired" for 10 years, Jim wanted the experiment to be as similar to the original debit/credit benchmarks as possible. With the help of Charles Levine, he was able to obtain over 8,000 tps using his two-year old laptop and Microsoft SQL Server 2000 (a product that did not even exist in 1985). Of course, he cheated a little bit on the scaling rules he himself had established but it was certainly his prerogative to do so (and fitting given the chosen publication date).

6. SUMMARY

Jim made many contributions to the database field. His theory of transactions and their implementation and the debit/credit benchmark serve as two bookends. Transactions provide the fundamentals that make all electronic commerce possible. His debit/credit benchmark helped drive the database industry forward for 15 years to the point where the cost of a transaction dropped to a small fraction of a penny. Together, these contributions allow electronic commerce to be incredibly low-cost and highly reliable. Our world would be very different today without either one.

7. DEWITT'S PERSONAL REFLECTIONS

While Jim's mentoring was invaluable to my career, his style was not always gentle. Once, when reviewing a paper of mine on database system architectures for client-server environments, Jim scribbled on the review: "DeWitt, What have you been smoking?" I knew it was Jim because he signed the review – something he tended to do with papers he really liked or really hated. Jim's advice wasn't always perfect - one time, when I was an assistant professor, Jim called me up and advised me to give up trying to make parallel database systems work - but he was usually right, and he was certainly right the time he called and told me that our approach to benchmarking was "all wrong".

8. LEVINE'S PERSONAL REFLECTIONS

Early in my career I had the very good fortune to sit across the hallway from Jim when we were both at Tandem. Jim was already a formidable presence in the database world (although I didn't appreciate that at the time) and I was simply a junior software developer a few years out of college. I learned a lot from Jim just by overhearing his conversations. Jim had a marvelous ability to distill complex things to their essential elements and then make connections and see trends. As a mentor, I believe that he was motivated by how much he could help others learn and grow, rather than working with the right people or the right projects. Altruism at its best.

As the TPC-A effort was wrapping up, Jim picked me to take over as Tandem's TPC representative. There were certainly more experienced and knowledgeable people Jim could have chosen, so it was quite a vote of confidence that he picked me. Jim set a high standard for honesty, integrity, and cooperation that I have tried to follow in the TPC and my career.

I have many fond memories of Jim. The last email I got from Jim was three weeks before he disappeared. Replying to an announcement of the birth of my son, Jim wrote "Congratulations. Now the fun begins!" Classic Jim. There's a lot about being a good person I hope to teach my son that I learned from him.

9. REFERENCES

Many of the papers cited below are posted at http://research.microsoft.com/~gray/JimGrayPublications.htm; they are marked with a (*).

[1] "A Measure of Transaction Processing Power," Anon et al., Datamation, April 1, 1985. *

[2] "Benchmarking Database Systems: A Systematic Approach," Bitton, D., DeWitt, D. J., and C. Turbyfil, Proceedings of the 1983 Very Large Database Conference, October 1983.

[3] "A Measure of Transaction Processing Power," Anon et al., Tandem Technical Report, TR 85.2. *

[4] "The History of DebitCredit and the TPC," Omni Serlin, The Benchmark Handbook, Chapter 2, 1993. *

[5] "History and Overview of the TPC," Shanley, Kim, http://www.tpc.org/information/about/history.asp, February 1998.

[6] "DeWitt Clauses: Can We Protect Purchasers Without Hurting Microsoft?" http://www.redorbit.com/news/technology/520809/dewitt_clauses_can_we_protect_purchasers_without_hurting_microsoft/index.html

[7] "The Evolution of TPC Benchmarks: Why TPC-A and TPC-B Are Obsolete," Charles Levine, Jim Gray, Steve Kiss, and Walt Kohler, San Francisco Systems Center Technical Report 93.1, Digital Equipment Corporation, September 1993. *

[8] http://research.microsoft.com/barc/sortbenchmark/

[9] "Thousands of DebitCredit Transactions-Per-Second," Jim Gray, and Charles Levine, Microsoft Research Technical Report, MSR-TR-2005-39, April 1, 2005. *

[10] "Introduction," Jim Gray, The Benchmark Handbook, Chapter 1, 1993. *