

Database and Transaction Processing Performance Handbook

Jim Gray,
Digital Equipment Corp

1. Introduction

This handbook is a compendium of the popular performance and price/performance metrics for database systems and transaction processing systems. Each benchmark tries to answer the question: “What computer should I buy?” Clearly, the answer question is “The system that does the job with the lowest cost-of-ownership”. Cost-of-ownership includes project risks, programming costs, operations costs, hardware costs, and software costs. It’s difficult to quantify project risks, programming costs, and operations costs. In contrast, computer performance *can* be quantified and compared.

This quantitative comparison starts with the definition of a *benchmark* or *workload*. The benchmark is run on several different systems, and the performance and price of each system is measured and recorded. *Performance* is typically a throughput metric (work/second) and *price* is typically a five-year cost-of-ownership metric. Together, they give a price/performance ratio.

For example, the transaction processing benchmarks define a standard transaction workload and a transaction per second (tps) metric. This benchmark can be run on various systems to get a table like the following:

Table 1. Performance and price/performance of several systems on a hypothetical benchmark.

System	throughput	price	price/performance
A	1 tps	10 k\$	10 k\$/tps
B	10 tps	50 k\$	5 k\$/tps
C	50 tps	2 M\$	40 k\$/tps
D	100 tps	1 M\$	10 k\$/tps

If the customer needs less than 1 tps, then system A is the most economic choice. If the customer needs between one and ten tps then system B is the most economic choice. If more than 10 tps are needed, then System D has the best price/performance.

Generic benchmarks are often used in this way as a rough estimate of the relative system performance because the cost of implementing and measuring a specific application on many different systems is usually prohibitive. For example, measuring the data for Table 1 above might

cost several person-years and a million dollars -- considerably more than the price of these systems.

Certain generic benchmarks have become so widely recognized that vendors announce the performance of new products in terms of those benchmarks. For example, DEC, HP, and IBM state the relative performance and price/performance of new machines and system releases by stating their ratings on the Transaction Processing Performance Council's benchmark TPC BM™ A. This practice is becoming increasingly common.

Generic benchmarks give some sense of the relative performance and price/performance of a system. They are analogous to the energy ratings now typical of appliances and automobiles. All such performance ratings are somewhat crude. As the Environmental Protection Agency says of its gasoline mileage ratings for automobiles: "Your actual mileage may vary according to road conditions and driving habits. Use these numbers for comparison purposes only." Put another way, the performance numbers a salesman quotes you 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 system performance and price/performance.

2. The Need for Domain-Specific Benchmarks

No single metric can measure the performance of computer systems on all applications. System performance varies enormously from one application domain to another. Each system is typically designed for a few problem domains and may be incapable of performing other tasks. For example, most supercomputers lack database and transaction processing software and so are inappropriate for most business applications.

The scientific community has evolved benchmarks that measure system performance on numeric computations. These scientific benchmarks do not give much guidance to someone evaluating a database system or a transaction processing system, because database system performance is dominated by the performance of software algorithms rather than by raw hardware speed. Even within the broad category of database systems there is substantial diversity among the performance of systems on different problem domains. One system may be excellent at performing simple update-intensive transactions for an online database; but it may have poor performance on complex queries to that database. Conversely, a system that excels at decision-support queries may not even allow online transactional access to that same data. Several chapters in this book give examples of such systems.

Domain-specific benchmarks are a response to this diversity of computer system use. Each such benchmark specifies a synthetic workload characterizing typical applications in that problem domain. The performance and of this workload on various computer systems then gives a rough estimate of their relative performance on that problem domain. This handbook contains seven domain-specific benchmarks covering database and transaction processing systems.

3. Standards Bodies Defining Benchmarks

Several consortia are defining standard domain-specific benchmarks, standard price metrics, and standard ways of measuring and reporting results. The three most prominent are:

SPEC (System Performance Evaluation Cooperative): A consortium of vendors defining benchmarks for the scientific and workstation domains.

The Perfect Club: A consortium of vendors and universities defining benchmarks for the scientific domain, with particular emphasis on parallel or exotic computer architectures.

TPC (Transaction Processing Performance Council): A consortium of vendors defining benchmarks for transaction processing and database domains.

This handbook presents the currently approved TPC benchmarks, and in addition includes other benchmarks widely used in the database area. The TPC benchmarks are especially interesting because they rigorously define how the tests should be run, how system price should be measured, and how the results should be reported. The TPC began by formalizing the ad hoc DebitCredit and TP1 benchmarks. The resulting two benchmarks are called TPC BM™ A and TPC BM™ B. Vendors now frequently quote the tps (transaction per second) ratings of their systems in terms of these TPC benchmarks. The TPC has now moved on to defining benchmarks that capture complex query, batch, and operational aspects of transaction processing systems.

4. The Key Criteria For a Domain-Specific Benchmark

To be useful, a domain-specific benchmark must meet four important criteria. It must be:

Relevant: It must measure the peak performance and price/performance of systems when performing typical operations within that problem domain.

Portable: It should be easy to implement the benchmark on many different systems and architectures.

Scaleable: The benchmark should apply to small and large computer systems. It should be possible to scale the benchmark up to larger systems, and to parallel computer systems as computer performance and architecture evolve.

Simple: The benchmark must be understandable, otherwise it will lack credibility.

As a case in point, consider the classic mips metric (millions of instructions per second). It is certainly a simple benchmark. But mips is not a portable metric. For example, IBM/370 mips and DEC/VAX mips are not comparable. Mips is not a scaleable metric since its application to multiprocessors is unclear: A multiprocessor consisting of a thousand one-mips processors is not equivalent to a single 1000 mips processor. The main criticism of the mips metric is irrelevance -- it does not measure useful work. For example, different compilers on the same computer can give mips ratings that vary by factors of three. Consequently, software must be compared when comparing processors. Simple measures like mips and dollars/mips, are *too* simple. They do not capture software performance and price/performance, and miss the non-CPU component of hardware performance -- the other 90% of the hardware [PATT90].

The scientific and workstation communities developed specific benchmarks to replace mips as a measure processor the performance. They developed suites of programs typical of scientific computations. The Perfect Club and the System Performance Evaluation Cooperative (SPEC) are successful examples of such benchmark suites [SPEC90], [BERR88]. These benchmarks satisfy the basic requirements of relevance, portability, and simplicity for measuring processor performance on numerical problems. However, they are not scaleable, in the sense that the problem size does not grow with processor power. This may present a problem when measuring the performance of systems consisting of large arrays of processors. Quite recently, the SLALOM scientific benchmark has appeared [GUST90]. It is a numerical benchmark which is portable, scaleable, and simple. In addition, it includes a small amount of input-output, which improves its relevance over the IO-less SPEC and Perfect benchmarks.

5. Domain-specific Benchmarks for Database and Transaction Processing

Each benchmark in this handbook meets the criteria for domain-specific benchmarks: relevance to its domain, simplicity, and scalability. In addition, each has been ported to several computer systems. Performance measurements from several systems are included in most of the benchmark descriptions. A tape containing some of these benchmarks is available from the publisher (see Appendix A).

The benchmarks in this handbook have the virtue of measuring the entire system. They measure the processor, the IO subsystem, the operating system, the compilers, and the database system. TPC BM™ A in addition measures network performance and price/performance. All the benchmarks adopt the pricing scheme pioneered by the Datamation benchmark and refined by the TPC. In that scheme, system price is the five-year price of hardware, software, and vendor maintenance with no special discounts [ANON85], [TPCA89]. Unfortunately, the metrics in this handbook do not measure ease-of-use, ease-of-programming, or ease-of-operations – they are purely performance and price/performance metrics¹.

The benchmarks included here were selected from a large field. First, benchmarks outside the database and transaction processing domain were excluded. Then proprietary benchmarks were excluded (e.g., IBM's RAMPC and ONEK, Stratus' TP1) since their definition is not publicly available. Then one-time benchmarks were excluded (e.g., the California Department of Motor Vehicles [DMV89]), because they are not portable or scaleable. This narrowed the field considerably. The remaining set was evaluated for relevance, portability, scalability, and simplicity. The main criteria was whether the benchmark was widely accepted within its domain-specific community.

6. An Historical Perspective on DB and TP Benchmarks

Several ad hoc benchmarks for data base and transaction processing systems have evolved over the last twenty years creating vague measures such as *transactions per second* and *query processing performance*. Each vendor has a suite of benchmarks inherited from competitive bids. In addition, each has implemented the standard ad hoc benchmarks so that they can measure their performance against the competition. For example, most SQL vendors have implemented the Wisconsin query set (See Chapter 3) and use it to test the performance of each new release, and each new machine. Many customers have repeated this effort to evaluate and compare various relational products. Similar activity has surrounded the Datamation query set (DebitCredit, Sort, Scan), especially the DebitCredit transaction also known as ET1 and TP1 [ANON85].

This ad hoc activity has occasionally been published or leaked to the press, but generally has been treated as company confidential. The vendors had little incentive to publish their performance because it was often embarrassing. When a vendor did publish numbers, they were generally

¹ Recently, PC Week had a “shootout” among several PC vendors in which each had to implement a complete application within four days [PCWE90]. This tests the functionality of the various systems, their ease-of-use, and also tests the talents of the implementors. Such events point in the right direction, but the implementors should be typical customers rather than typical gurus.

treated with skepticism. When comparative numbers were published by third parties or competitors, the losers generally cried foul and tried to discredit the benchmark. Such events often caused *benchmark wars*. Benchmark wars start if someone loses an important or visible benchmark evaluation. The loser reruns it using regional specialists and gets new and winning numbers. Then the opponent reruns it using his regional specialists, and of course gets even better numbers. The loser then reruns it using some one-star gurus. This progression can continue all the way to five-star gurus. At a certain point, a special version of the system is employed, with promises that the enhanced performance features will be included in the next regular release.

Benchmarking is a variation of benchmark wars. For each system, there is some benchmark that rates that system the best. Typically, a company marketing organization defines such a domain-specific benchmark, highlighting the strengths of the product and hiding its weaknesses. The marketing organization then promotes this benchmark as a “standard”, often without disclosing the details of the benchmark. Benchmarking leads to a proliferation of benchmarks and creates confusion. Ultimately benchmarking leads to universal skepticism of all benchmark results.

The claims and counter-claims surrounding benchmarking and benchmark wars caused enormous confusion in the database and transaction processing community. In response, a consortium of 34 software and hardware vendors, led by Omri Serlin, founded the Transaction Processing Performance Council (TPC) in 1988. The TPC’s goal is to define domain-specific benchmarks for transaction processing and database systems. Incident to this goal, they also define a way to calculate system price, and a way to report performance results.

So far the TPC has defined two benchmarks, TPC BM™ A, and TPC BM™ B (see Chapter 2). These two benchmarks have dramatically reduced the benchmark wars. Customers are beginning to request TPC ratings from vendors. The TPC metrics capture peak performance and price/performance of transaction processing and database systems running simple update-intensive transactions. In addition, the TPC reporting procedures, while not air-tight, have a “full disclosure” mechanism that makes it difficult to stretch the truth too much. The benchmark must be done on standard hardware and released software. Any special tuning or parameter setting must be disclosed. In addition, the TPC highly recommends that an independent organization audit the benchmark tests.

7. An Overview of the Benchmarks in this Handbook

The problem domains represented in this handbook and their associated benchmarks are:

Online Transaction Processing including a LAN or WAN network: TPC BM™ A

Online Transaction Processing with no network: TPC BM™ B

Relational Queries: Wisconsin

Mixed Workload of Transactions, Relational Queries, and Utility Functions: AS³AP

Complex Queries and Reporting: Set Query Benchmark

Engineering Workstation-Server: Engineering Database Benchmark

In addition, there are some commercial benchmarks sold by for-profit consulting organizations. The two most prominent of these are Neal Nelson Associates, and AIM Technology [AIM]. They each have proprietary Unix-based benchmarks that they tailor to a customer's needs. The companies provide customers and vendors with competitive and comparative performance evaluation advice. A brief description of Neal Nelson's Business Benchmark is included here as a representative of such benchmarks and services.

Each benchmark in this handbook is presented by someone intimately familiar with it. Omri Serlin was the founder of the TPC consortium. He augments the two standards documents with a historical perspective on the TPC itself, an overview of the two benchmarks, and representative performance figures. These benchmarks are the standard definitions of transaction per second (tps) and \$/tps. They are now widely used to evaluate the approximate performance and price/performance of computer systems for online transaction processing workloads.

The two differ in that TPC BM™ A also measures the performance and price of a computer network, its software, and its terminals. These two benchmarks are the main reason for this handbook. They reflect the field's maturity. The remaining benchmarks are included because they are widely used in their particular domain. Each has proved its usefulness in measuring performance, but they do not have the rigor or force of the TPC benchmarks. Over time, we expect that they will be either be adopted and refined by the TPC or will be replaced by comparable benchmarks from the TPC. But that may be many years away -- the TPC is defining new metrics at a rate of one benchmark per year.

The benchmarks presented in Chapters 3-5 test query processing performance. They consist of (1) a set of pure relational queries, (2) a mixed workload of queries, system administration tasks, along with some simple update jobs, and (3) a suite of complex queries typical of decision support applications. Interestingly, systems that perform well on one of these three benchmarks are not necessarily good at the other two.

The first of these query benchmarks is called the *Wisconsin benchmark*. This benchmark is widely used to test the performance of relational query systems. Dave DeWitt, who with Carolyn

Turbyfill and Dina Bitton invented the benchmark at the University of Wisconsin, restates the benchmark and database generator in modern (SQL) terms. He explains how to port the benchmark, how it scales, and discusses its use to measure the performance of parallel database machines. These benchmarks are a litmus test for the performance of relational systems on simple relational operators

Dina Bitton, Cyril Orji, and Carolyn Turbyfill then present the AS³AP benchmark they developed as a more complete test of relational database systems. The traditional Wisconsin benchmark does not test utility functions, does not mix batch and interactive queries, and does not emphasize multi-user tests. AS³AP provides a more complete metric by including such features. In addition, it adopts the novel approach of setting a time-limit on the benchmark and then measuring the largest database that the system can process within this time limit². This gives each database system an *equivalent database size* rating. AS³AP gives a more balanced and realistic evaluation of the overall performance of a relational system used for database queries.

The third query benchmark in this series presents a much more complex query set. The *Set Query Benchmark* is used to evaluate the ability of systems to process complex queries typically found in decision-support applications and data-mining applications. The Wisconsin and AS³AP query sets map to simple relational queries. Pat O'Neil points out that decision support applications involve complex queries requiring efficient set processing. The Wisconsin and AS³AP queries do not capture this requirement, and most commercial systems do not implement the necessary techniques (notable exceptions are Model 204 and Adabase). O'Neil presents a scaleable database and a Set Query Benchmark that measures this capability. This query set is based on extensive experience with decision support customers in areas ranging from information retrieval to telemarketing. O'Neil then contrasts the performance of Model 204 with IBM's DB2 on this benchmark.

The final benchmark presented here represents an emerging use of database systems to store engineering or object-oriented databases accessed from high-performance workstations in a client-server role. Rick Cattell characterizes such databases as having relatively rich inter-relationships, and the transactions as navigating from one record to its many relatives. The performance of such operations is absolutely critical. Engineers need to be able to access complex objects within a few seconds. Similar problems arise with hyper-text database and image-processing applications. Cattell argues that, at present, this performance requirement precludes the use of general purpose

² This fixed-time approach is also adopted by the SLALOM benchmark which limits the run to 1 minute. AS³AP runs are limited to 12 hours.

database systems because their underlying system architecture is not suited to a workstation-server design, and because the performance penalty of moving data between the application and a classical database system is too high. He presents a scaleable database and the *Engineering Database Benchmark* that captures these demands. By example, he shows that database systems with the correct architecture outperform conventional systems by a factor of ten.

The next chapter presents the *Business Benchmark* used by Neal Nelson Associates to evaluate Unix-based computer systems for specific customers. Neal Nelson presents the design rationale for the benchmark, and describes how it is used by giving three case studies. He makes the argument that customers need vendor-independent “Underwriters Laboratory” to evaluate database systems. His group provides such a laboratory.

The closing chapter presents an auditor’s view of running and reporting benchmarks. Tom Sawyer, who has audited many commercial benchmarks, comments on the tasks and pitfalls of benchmarking.

8. How to Use This Handbook

This handbook has two purposes -- it is a tutorial for the novice and a reference for the professional. For those unfamiliar with the various benchmarks, it presents a tutorial as well as a detailed specification of each. The tutorials explain the rationale behind each benchmark design, and give some sense of the problem domains they intend to measure. In addition, the tutorial sections give a sense of the performance and price/performance of current systems on the benchmarks. Overall, the handbook gives a sense of benchmark design methodology, and of the state of the art of benchmarking transaction processing systems and database systems.

Readers may find the text of the TPC benchmarks especially useful. The careful definitions of performance metrics, pricing, and reporting provide a template for computer system requests-for-proposal and for bids. Along those lines, note that the TPC standards material is *not* copyrighted and so may be freely copied and used in bids, and proposals.

For the performance professional, this handbook is the standard reference for the benchmark definitions. It provides a convenient package of the TPC benchmarks, the query benchmarks and the engineering database benchmark. The professional may find some interesting insights in the tutorials, and in the article by Tom Sawyer on how to conduct, report, and audit a benchmark.

Most importantly, this handbook brings together the main fragments of the database and transaction processing performance field. Much of the information here has not appeared in book

form. In preparing this handbook, each of us discovered many standard benchmarks that we had never before heard of, primarily because they were outside the problem domains we usually encounter.

9. How to Use These Benchmarks

These benchmarks can be used in many ways. Their main uses fall into one of the following four categories:

- **Compare Different Software and Hardware Systems:** The benchmarks can be used to evaluate the relative performance of different systems on different hardware running the same application (e.g., CICS-DB2 on an IBM 3090/200 vs. ACMS-Rdb on a DEC VAX 9000). This is the classic competitive situation between two hardware vendors.
- **Compare Different Software Products on One Machine:** The benchmarks can be used to evaluate the relative performance of two different software products running on the same hardware. O'Neil gives the example of Model 204 and DB2 both running the Set Query benchmark on an IBM AS 9000/50 processor with all other software held the same. He measures the system performance as queries per minute (qpm), and the price/performance as dollars per qpm. This is the classic competitive situation between two software vendors. His results are:

<u>System</u>	<u>qpm</u>	<u>\$/qpm</u>
Model 204	2.7	215k\$
DB2	.63	803k\$

- **Compare Different Machines in a Compatible Family:** The benchmarks can be used to rate the relative performance of computer systems within a computer family. For example the current ratings within the HP family are:

<u>System</u>	<u>tps</u>	<u>A-Local</u>	<u>k\$/tps</u>
HP 960	38.2		36.5
HP 949	32.2		29.3
HP 932 ³	13.6		33.3
HP 922	7.7		35.5
HP 920 ³	4.9		33.0

- **Compare Different Releases of a Product on One Machine:** The benchmarks can also be used to evaluate the performance improvement of one software release of a product over its predecessor. Ideally the new software will be a little bit faster, but often new features slow down old features. So these benchmarks provide performance regression

³ Unaudited results.

tests for vendors and customers alike when doing performance assurance of a new release. For example, the DEC ratings for the two most recent releases of Rdb/ACMS on the VAX 4000-300 are:

	Release	tpsA-Local	k\$/tps
Rdb release 3	VAX 4000-300	21.7	31.9
Rdb release 4	VAX 4000-300	21.6	32.1

Of course, in each of these cases, the benchmarks must be chosen to reflect the problem-specific domain you are interested in.

10. Further Reading

The field of computer performance evaluation is undergoing a transformation, partly due to improved understanding and sophistication, and partly due to the shift of performance issues from the hardware domain to the broader perspective of hardware and software systems architecture. The recent book *Computer Architecture, a Quantitative Approach* by Dave Patterson and John Hennessy gives an excellent treatment of hardware performance issues, and contains sound advice on performance metrics [PATT90]. The concepts they develop have broad applicability to hardware and software systems. The journal *Supercomputing Review* [SR] publishes LINPACK, SPEC, SLALOM, and other scientific benchmark results.

Within the field of software performance evaluation there are many texts. The one most relevant to the topics discussed here is Bill Highleyman's *Performance Analysis of Transaction Processing Systems* [HIGH89]. It discusses both the relevant analytical techniques, as well as measurement tools and techniques. The classic text in this field is Domenico Ferrari's *Computer Systems Performance Evaluation* [FERR78].

Many of the benchmarks here assume the reader is familiar with the basics of databases and transaction processing. They assume a reading knowledge of SQL and C. In addition, they use terms such as "ACID transactions", "degrees of isolation", "prefetch", "join", and so on. Fortunately, several textbooks are emerging that explain and consolidate these concepts. Tamer Ozsü and Patrick Valdez's recent book *Principles of Distributed Database Systems* gives an excellent treatment of the transaction and database concepts that underlie most of the benchmarks in this book [OZSU90]. Andreas Reuter and I are writing the book *Transaction Processing Systems, Concepts and Techniques*, that complements the Ozsü-Valdez by focusing more on the low-level implementation issues of transaction processing systems [GRAY92].

For more timely information, Omri Serlin's monthly newsletter, *FT Systems*, "analyzes company and product developments in fault-tolerant computers and transaction systems" to quote its masthead [FT]. It gives regular and insightful reports on the activities of the TPC. It also gives regular reports of benchmarking activities, reporting on the latest transaction processing and database systems benchmark wars.

11. Future Database and Performance Benchmarks

This is the first edition of a handbook that is likely to evolve over the years. As the TPC defines new benchmarks, we hope to incorporate them in this handbook. The field is now in its infancy. There is consensus that standard benchmarks will appear in the following areas.

Complex Queries: The Wisconsin query set, the AS³AP query set, and the Set Query Processing Benchmark all try to model complex queries against a relatively large and complex database. The TPC is currently considering a benchmark that characterizes these same issues. This is the most likely next step in the efforts of the TPC.

Complex Transactions: Within the transaction processing sphere, there is a sense that the TPC BM™ A transaction is too simple -- involving only two messages and four database records. There is a desire to define a more complex transaction profile. One that is message intensive, one that is database intensive, and perhaps one that is highly distributed.

Utility Operations: As terabyte databases become common, the cost of managing them becomes a central issue. Dumping them at a rate of a megabyte a second will require over 12 days (and 8,000 magnetic tape pairs using conventional technology). Building an index on such a 10 billion record file will break many sort programs, not to mention the elapsed time such sorts would require. A benchmark characterizing the key utility operations on such large databases is needed. Transaction processing systems have similar online utility needs, such as adding new transaction programs while the system is operating.

Mixed Workloads: Most benchmarks measure an isolated aspect of a system. AS³AP and the Business Benchmark are exceptions. They measure systems performing a mixed workload of simple transactions, queries, reports, and utilities (such as database load and index creation). AS³AP is the only publicly specified benchmark that includes utility operations. The ability to support online operations and to support mixed workloads against a single shared database is widely viewed as a critical performance issue.

New Applications: The Engineering Database Benchmark and the Set Processing Benchmark have their roots in problem-specific domains. They are just the first in a series of problem-specific domains needing a special benchmark. Domains such as image processing, geographic databases, real-time transaction processing, and so on, will likely create new benchmarks.

Transaction and database system performance measurement is a dynamic field. It is a key component of the multi-billion dollar industry -- the procurement of database and transaction processing systems. For many years the benchmark wars, claims, and counter claims created chaos. But gradually standard terminology and metrics have emerged and are now endorsed by the major hardware and software vendors. There is every sign that this trend will continue, and that each new industry will develop domain-specific benchmark standards that allow approximate performance and price/performance comparisons of products in that area.

References

- [AIM] AIM Technology Procurement Guide, AIM Technology, 4699 Old Ironsides Dr. Suite 150, Santa Clara, CA.
- [ANON85] Anon, et al, "A Measure of Transaction Processing Power", *Datamation*, V. 31.7, April 1985, pp. 112-118.
- [BERR88] Berry, M.D., D. Chen, P. Koss, D. Kuck, "The Perfect Club Benchmarks: Effective Performance Evaluation of Supercomputers", CSRD Report No. 827, Center for Supercomputing Research and Development, U. Illinois at Urbana, Nov. 1988.
- [BITT83] Bitton D., DeWitt D.J., and Turbyfill C., "Benchmarking Database Systems, a Systematic Approach," *Proc. Ninth International Conf. on Very Large Data Bases*, November 1983.
- [DMV89] *Database Redevelopment Project: DBMS Operational Assessment Final Report, Volume 1: Project Summary, Volume 2: Test Specification, Volume 3: Vendor Results*. California Department of Motor Vehicles, State of California, Sacramento CA. January 1989.
- [FERR78] Ferrari, D., *Computer Systems Performance Evaluation*, Prentice Hall, Englewood Cliffs, NJ., 1978.
- [FT] *FT Systems*, a monthly newsletter published by ITOM International, POB 1450, Los Altos, CA 94023.
- [GRAY92] Gray, J.N., Reuter, A., "Transaction Processing: Concepts and Techniques, to be published by Morgan Kaufmann, San Mateo, CA., 1992.
- [HIGH89] Highleyman, W.H., *Performance Analysis of Transaction Processing Systems*, Prentice Hall, Englewood Cliffs, NJ., 1989.
- [OZSU90] Ozsu, T., P. Valdez's, *Principles of Distributed Database Systems*, Prentice Hall, Englewood Cliffs, NJ., 1990.
- [PATT90] Patterson, D. A., J. L. Hennessy, *Computer Architecture, a Quantitative Approach*, Morgan Kaufmann, San Mateo, CA., 1990.
- [PCWE90] "Database Shootout: 17 Database Vendors Go Head-To-Head To Build a Real-World Computing Solution," *PC Week*, Aug 27, 1990, pp. 70-78.
- [GUST90] Gustafson, J., D. Rover, S. Elbert, M. Carter, "SLALOM, The First Scaleable Supercomputer Benchmark", *Supercomputing Review*, Vol. 3, No. 11, Nov. 1990, pp. 56-61.

[SPEC90] "SPEC Benchmark Suite, Release 1.0," *Supercomputing Review*, Vol. 3, No. 9, Sept. 1990, pp. 48-57. Also available from SPEC, c/o Waterside Associates, 39510 Paseo Padre Pkwy., Suite 350, Fremont, DA 94538.

[SR] *Supercomputing Review*, 8445 Camino Santa Fe, San Diego, CA 92121.