An Annotated Bibliography of Benchmarks for Object Databases

Akmal B. Chaudhri
Systems Architecture Research Centre
Computer Science Department
The City University
Northampton Square, London EC1V 0HB, UK

akmal@cs.city.ac.uk

Abstract

This annotated bibliography presents a collection of published papers, technical reports, Master's and PhD Theses that have investigated various aspects of object database performance.

1 Introduction

Developing performance metrics for any database system is always a difficult task. Generally, a benchmark is designed to be representative of the characteristics and load of a "real" application, or to test some specific components of a DBMS. An example of the former is DebitCredit [Anon et al. 1985] which was designed to measure the Transactions Per Second (TPS) performance (or system throughput) of different transaction processing systems. Using the TPS measure, various systems could then be fairly compared using a standard price/performance ($/TPS) ratio. An example of the latter is the Wisconsin Benchmark [Bitton et al. 1983] which was designed to test the performance of relational database systems (specifically selections, joins, projection, aggregates, updates). It was perhaps the first attempt to measure relational database performance using a standard methodology.

Both DebitCredit and Wisconsin have been very popular, because of their timeliness, simplicity and portability. However, they have also attracted significant criticism, e.g. [Khoshafian et al. 1992]. This is a general problem which any benchmark designer faces [Stonebraker 1988]. However, there are some good reasons for developing benchmarks. For example, benchmarks can provide a "level playing field", allowing DBMS systems to be compared on a standard collection of commands [Stonebraker 1988]. Additionally, performance is what underlies the success and measure of competitiveness of different database products [Inmon 1989]. This is particularly relevant for object databases, where performance is typically more important than functionality [Cattell 1991], and is probably among the top three selection criteria for users when deciding which object database to purchase [Rotzell & Loomis 1991].

Benchmarks have also served to highlight problems with database systems. For example, the Wisconsin Benchmark showed how one relational database system responded to a test query in a few seconds, whilst another system took several hours, as its query optimiser had used a different strategy. Similarly, the Sun Benchmark [Rubenstein et al. 1987] discovered problems with the indexing system for one commercial relational database product. More recently, [Carey et al. 1993] reported finding at least one major problem with each object database system that was tested with the OO7 Benchmark.

For users, standard benchmarks can alleviate the cost of designing and implementing their own benchmarks. [Barry 1994], for example, has commented that it is easy to spend $100,000 implementing a benchmark on several object databases. However, as standard benchmarks become more sophisticated (and hence more expensive to implement), fewer results may actually be available. Additionally, it becomes more difficult to interpret the results. Another drawback with a standard benchmark is that it may not be representative of user applications. This is a problem particularly for object databases, which are being used in many diverse areas such as healthcare, financial modelling, engineering, etc. Treating performance and its measurement generically is wrong, and can lead to incorrect conclusions [Inmon 1989].

Benchmarks can therefore serve a useful purpose for both vendors and users alike. Vendors can use them for testing new algorithms, quality assurance and diagnostics, whereas users will typically use them for choosing a particular system. One of the dangers, however, is that a benchmark can lead to so-called "database wars", since vendors will use a benchmark on each new release of their product, highlighting improvements in performance over the previous version, besides claiming superior performance over their competitors. This resulted with the Wisconsin Benchmark, and appears to be the direction that some vendors are taking with the OO7 Benchmark.

2 Bibliography

The bibliography is organised alphabetically by benchmark name. Then within this category, alphabetically by surname.

The author welcomes any corrections or additions to this list.

Altair Complex-Object Benchmark (ACOB) and Derivatives

[DeWitt et al. 1990] D. DeWitt, P. Futtersack, D. Maier and F. Velez (1990) A study of three alternative workstation-server architectures for object oriented database systems. *Proceedings of the Sixteenth Very Large Data Bases Conference*, Brisbane, Australia, 1990, pp. 107-121.

Also available as Technical Report No. 936, Computer Sciences Department, University of Wisconsin-Madison, 1990.

Also available as Technical Report No. 42-90, Altair-INRIA, Le Chesnay Cedex, France, January 1990.

Also in *Building an Object-Oriented Database System: The Story of O2*, F. Bancilhon, C. Delobel and P. Kanellakis (Eds.), Morgan-Kaufmann, San Mateo, California, 1991.

ACOB was a simple benchmark used to compare *object*, *page* and *file server* architectures. Three operations (sequential scan, read, update) were used on a database consisting of sets of complex objects, with the read operation being based on observed access patterns for VLSI tools. The results showed that although each architecture had particular strengths and weaknesses, overall there was no clear winner.

[Teeuw 1993] W.B. Teeuw (1993) Parallel management of complex objects. The design and implementation of a complex object server for Amoeba. PhD Thesis, University of Twente, The Netherlands, September 1993.

[Teeuw et al. 1993] W.B. Teeuw, C. Rich, M.H. Scholl and H.M. Blanken (1993) An evaluation of physical disk I/Os for complex object processing. *Proceedings of the Ninth International Conference on Data Engineering*, Vienna, Austria, 1993, pp. 363-371.

Also available as Report No. 183, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, September 1992.

The paper by [Teeuw et al. 1993] describes an analytical model to estimate disk I/Os. The model is validated by experiments using four storage models for complex objects. In [Teeuw 1993], the same performance metrics are used as in the paper. Additionally, some measurements with the benchmark are shown.

Benchmark to Scale Behaviorally Object-Oriented Databases

[Kemper & Chriesten 1990] J. Kemper and J. Chriesten (1990) A benchmark to scale behaviorally object-oriented database systems. *Proceedings of the Fifth Jerusalem Conference on Information Technology*, Jerusalem, 1990, pp. 677-687.

Most well-known benchmarks for object databases focus on the structural dimension of object modelling and do not consider the behavioural dimension. This paper details work to consider both aspects, with objects and behaviour modelled on actual CAD/CAM applications. The benchmarks were implemented on a relational database (SQL/DS), and an experimental non-first normal form database system that incorporated the notion of abstract data types. Although the latter did perform worse than SQL/DS on some tests, the developers concluded that improvements in performance could be achieved by the close integration of object behaviour into the database query language.

Benchmarks for Language Based Editors (LBEs)

[Pedersen 1993] M. Pedersen (1993) An analysis of introducing UQ LBEs to persistent storage environments. Working Paper No. 1, Dept. of Computer Science, University of Queensland, Australia, 6 August 1993.

This paper summarises the work on benchmarking persistent language based editors at the University of Dortmund and focuses on approaches to persistence for recognition editors. The work here should also be compared with [Emmerich & Kampmann 1992] and [Emmerich & Schafer 1993], both mentioned later.

[Pedersen 1994a] M. Pedersen (1994) Developing benchmarks for recognition editors in persistent storage environments. Working Paper No. 2, Dept. of Computer Science, University of Queensland, Australia, 31 March 1994.

This paper describes strategies used to develop a benchmark for recognition editors as a first step towards migrating them to a persistent storage environment.

[Pedersen 1994b] M. Pedersen (1994) Benchmarking the UQ1 editor. Working Paper No. 3, Dept. of Computer Science, University of Queensland, Australia, 31 March 1994.

This paper presents results of a benchmarking investigation, using a recognition editor (UQ1) developed at the University of Queensland.

[Pedersen 1994c] M. Pedersen (1994) Towards persistence for recognition editors. Working Paper No. 4, Dept. of Computer Science, University of Queensland, Australia, 8 April 1994.

Benchmarks to Compare Relational and Object Databases

[Lakey 1989] B. Lakey (1989) Developing benchmarks for comparing relational and object-oriented database systems. Master's Thesis, Oregon Graduate Center, Beaverton, Oregon, 31 July 1989.

[Lakey et al. 1987] B. Lakey, D. Maier and J. Stein (1987) Benchmarking methodology for object-oriented data management systems. Unpublished.

The thesis and paper suggest a comparison based on complete operations at the application level, as opposed to measuring low-level primitives. Text document and Hypertext benchmark results are reported for University Ingres and a very early version of GemStone. Some of the conclusions suggest that direct comparisons will always be difficult as some factors, such as logical and physical tuning, will have a major impact on performance.

[Larsen 1992] A.B. Larsen (1992) A test evaluation procedure for object oriented and relational database management systems. Master's Thesis, Institute of Informatics, University of Oslo, Norway, 5 February 1992.

This work describes a Test Evaluation Procedure (TEP) based on a simplified version of the HyperModel Benchmark. The TEP was implemented on one relational and two object databases and includes both single- and multi-user tests. The TEP was specifically designed with scalability, simplicity and portability in mind. The thesis also contains a useful survey of object and relational database benchmarks.

British Telecom Benchmark

[Baker & Salman 1991] S. Baker and M.A. Salman (1991) Performance comparison between an object-oriented, a relational and an object management database. *Proceedings of Advanced Information Systems*, London, UK, 1991, pp. 61-67.

In this paper, the database schema was designed to represent a hypothetical network consisting of *multiplexors*, *repeaters* and *PBXs*. Six operations were defined (one select, three traversals, two structural modifications), and the databases were populated with approximately 100,000 objects. The superior performance of the object database was attributed to two main reasons. Firstly, the application itself was object-oriented and was therefore easier to implement on the object database, whilst the relational database incurred a cost for trying to support a paradigm it wasn't designed for. Secondly, since the object database was suited to navigational queries, the required objects could be retrieved directly rather than the expensive "search and match" approach used in the relational database.

Engineering Database Benchmark (EDB), Object Operations 1 (OO1) and Derivatives

[Cattell 1988] R.G.G. Cattell (1988) Object-oriented DBMS performance measurement. *Proceedings of the Second International Workshop on Object-Oriented Database Systems*, Lecture Notes in Computer Science, 334, pp. 364-367 (Berlin: Springer-Verlag)

Following experience with the Sun Benchmark [Rubenstein et al. 1987], this paper suggested a number of modifications. Firstly, the schema was changed to reflect a database with a better engineering "flavour", consisting of *parts* and *connections* (typical of a circuit board), as some confusion had resulted from the original *person*, *author* and *document* schema. Secondly, the number of operations were reduced to just three, as it was found that measurements for some operations differed by a constant time and therefore added little value.

[Cattell 1991] R.G.G. Cattell (1991) An engineering database benchmark. In: *The Benchmark Handbook for Database and Transaction Processing Systems*, J. Gray (Ed.) Morgan-Kaufmann, San Mateo, California.

Also in *Readings in Database Systems (Second Edition)*, M. Stonebraker (Ed.), Morgan-Kaufmann, San Mateo, California, 1993.

These papers are both derivatives of the [Cattell & Skeen 1992] paper.

[Cattell & Skeen 1992] R.G.G. Cattell and J. Skeen (1992) Object Operations Benchmark. *ACM Transactions on Database Systems*. 17 (1):1-31.

Also available as a Technical Report, Database Engineering Group, Sun Microsystems, Mountain View, California, April 1990.

In this paper, the final version of OO1 is described. The superior performance demonstrated by object databases over relational databases was attributed to efficient access methods (e.g. parent-child links), minimised concurrency control, use of a local cache and no interprocess communication for database calls. The authors noted that these architectural differences dwarfed any data model differences. Another important contribution of the benchmark was to demonstrate how an implementation using parent-child links provided superior performance than B-trees on traversal operations. This is significant, since B-trees are used extensively in relational database systems, and techniques such as direct physical links may be the way forward for relational systems. Direct links, for example, have been used to enhance nested table performance in the UniSQL DBMS [Finkelstein 1993].

[Duhl & Damon 1988] J. Duhl and C. Damon (1988) A performance comparison of object and relational databases using the Sun Benchmark. *Proceedings of the ACM International Conference on Object-Oriented Programming Systems, Languages, and Applications*, San Diego, California, 1988, pp. 153-163.

This paper reports on results for Vbase (one of the early object database systems) using the Sun Benchmark [Rubenstein et al. 1987]. The results showed that Vbase was generally faster than relational databases. The paper also contains an interesting critique of the original benchmark.

[Rubenstein et al. 1987] W.B. Rubenstein, M.S. Kubicar and R.G.G. Cattell (1987) Benchmarking simple database operations. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, San Francisco, California, 1987, pp. 387-394.

This paper describes an early attempt to develop benchmarks more representative of engineering applications. Additionally, the authors were interested in identifying database systems that could support one or more of the following:

- Caching the entire database in main memory.
- Avoiding the overhead of query optimisation.
- Using pre-computed links.
- Alternative database server architectures.

The benchmark schema consisted of the entities *person*, *author* and *document*, with *author* as an intersection entity between *person* and *document* to map many-to-many relationships. Two database sizes were used, with the smaller one being able to fit entirely within workstation memory. Seven operations were tested against three database systems (Ingres, Unify and Rad-Unify). Rad-Unify (a modified version of Unify) used a simplified locking scheme to more closely model the requirements of engineering applications (one database writer at a time), plus caching to keep as much of the database in memory as possible. The results showed that for the small database, Rad-Unify produced up to an order of magnitude better performance, whilst for the large database, the advantages of main memory residency and the other modifications were lost.

[Sequent 1993] Sequent (1993) A benchmark for object storage. White Paper, Sequent Computer Systems Inc. and Servio Corporation, April 1993.

This paper describes Sequent's extensions to OO1 to produce a multi-user benchmark. Some results are also reported using GemStone on Sequent SMP hardware.

Grid-Benchmark for Object Clustering

[Gerlhof et al. 1992] C. Gerlhof, A. Kemper, C. Kilger and G. Moerkotte (1992) Clustering in object bases. Technical Report No. 06/92, Faculty of Informatics, University of Karlsruhe, Germany, June 1992.

This paper describes how application behaviour can be exploited in an object database to develop a better clustering strategy. Firstly, access behaviour is determined using dataflow analysis techniques. Then heuristics are applied to graph partitioning. The benchmark results show that this approach can outperform other clustering strategies.

HyperModel Benchmark and Derivatives

[Anderson et al. 1990] T.L. Anderson, A.J. Berre, M. Mallison, H.H. Porter and B. Schneider (1990) The HyperModel Benchmark. *Proceedings of the Second International Conference on Extending Database Technology*, Lecture Notes in Computer Science, 416, pp. 317-331 (Berlin: Springer-Verlag).

Also available as Technical Report No. 89-05, Tektronix Inc., Beaverton, Oregon, 3 August 1989.

The HyperModel Benchmark used the node-and-link graph structure common in Hypertext applications. Each node (text or small graphic) at level n was linked to five other nodes at level $n+1$ (referred to as the "fan out"). In addition, two hierarchies were added over the nodes - *parent/children* (1:N cardinality) and *partOf/parts* (M:N cardinality). Benchmark operations included those of the original Sun Benchmark, with additional operations to test transitive closure and complex computation. Results were reported for Vbase and GemStone.

[Berre & Anderson 1991] A.J. Berre and T.L. Anderson (1991) The HyperModel Benchmark for evaluating object-oriented databases. In: *Object-Oriented Databases with Applications to CASE, Networks, and VLSI CAD*, edited by R. Gupta and E. Horowitz (Englewood Cliffs, NJ: Prentice-Hall)

This paper describes algorithms and code fragments from the HyperModel Benchmark for Vbase.

[Berre et al. 1988] A.J. Berre, T.L. Anderson and M. Mallison (1988) The HyperModel Benchmark. Technical Report No. CS/E 88-031, Oregon Graduate Center, Beaverton, Oregon, 1988.

This report discusses data model and database system requirements for hypertext applications. Schema design and benchmark operations for HyperModel are also described.

[Harrus et al. 1991] G. Harrus, V. Benzaken and C. Delobel (1991) Measuring performance of clustering strategies: the CluB-0 Benchmark. Technical Report No. 66-91, Altair-INRIA, Le Chesnay Cedex, France, January 1991.

Adapts the HyperModel Benchmark to investigate clustering, and also proposes performance indices to measure the efficiency of clustering decisions. The paper reports implementation on an early version of O2, together with detailed performance results.

[Hormann et al. 1990] H. Hormann, D. Platz, M. Roschewski and L. Schope (1990) The HyperModel Benchmark, description, execution and results. Internal SWT Memo No. 53, University of Dortmund, Germany, September 1990.

This paper describes efforts to implement the HyperModel Benchmark on GRAS (a graph storage system) and PCTE. According to [Dewal et al. 1990], two to four months were required per system to implement the benchmark. The results are tabulated for comparison purposes with those for GemStone and Vbase from the paper by [Anderson et al. 1990].

JUSTITIA Generic Object Database Benchmark

[Schreiber 1994] H. Schreiber (1994) JUSTITIA: a generic benchmark for the OODBMS selection. *Proceedings of the Fourth International Conference on Data and Knowledge Systems in Manufacturing and Engineering*, Hong Kong, 1994, pp. 324-331.

In this paper, a multi-user, parameterised benchmark is described. Database operations seem to be drawn from OO1, HyperModel, OO7. However, some higher-level operations are user-definable. Whilst no performance results are reported, a number of reference implementations are available for some of the major object database products.

OO7 Benchmark

[Carey et al. 1993] M.J. Carey, D.J. DeWitt and J.F. Naughton (1993) The OO7 Benchmark. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Washington DC, 1993, pp. 12-21.

Also available as Technical Report No. 1140, Computer Sciences Department, University of Wisconsin-Madison, 12 April 1993, revised January 1994.

The OO7 Benchmark was specially designed to test the following characteristics of object databases:

- Speed of pointer traversals.
- Update efficiency.
- Query processor or query programmer.

The benchmark schema is far richer than either OO1 or HyperModel and is based on the idea of a design library consisting of parts and assemblies, and is suggestive of a number of applications such as CAD, CAM, CASE, etc. However, OO7 should really be considered as a *system benchmark*, since it is not based on any performance studies to identify the schema or data manipulation characteristics of such applications. Benchmark results and reference implementations are available for a number of major object database systems, such as Objectivity/DB, ONTOS, VERSANT and the University of Wisconsin research prototype EXODUS.

[Carey et al. 1994] M.J. Carey, D.J. DeWitt, C. Kant and J.F. Naughton (1994) A status report on the OO7 OODBMS benchmarking effort. *Proceedings of the ACM International Conference on Object-Oriented Programming Systems, Languages, and Applications*, Portland, Oregon, 1994, pp. 414-426.

This paper summarises the work presented in the original paper, takes a retrospective look at some of the benchmark decisions, and describes some of the on-going work to improve and extend OO7.

[Objectivity 1993] Objectivity (1993) A guide to interpreting and applying the University of Wisconsin OO7 Benchmark. White Paper, Objectivity Inc., 23 August 1993.

[ODI 1993] ODI (1993) Understanding the OO7 research project. White Paper, Object Design Inc., 4 April 1993.

[VERSANT 1994] VERSANT (1994) ODBMS performance and the OO7 Benchmark. White Paper, Versant Object Technology.

Collectively, these three white papers are vendors reporting OO7 results for their own products. The results in [ODI 1993] have not been officially "blessed" by all members of the original OO7 team.

Opus-Merlin Benchmarks for Software Engineering Environments (SEEs)

[Emmerich & Kampmann 1992] W. Emmerich and M. Kampmann (1992) The Merlin OMS Benchmark: definition, implementation and results. Internal Memo No. 65, Department of Computer Science, University of Dortmund, Germany, October 1992, revised 22 July 1993.

[Emmerich & Schafer 1993] W. Emmerich and W. Schafer (1993) Dedicated object management system benchmarks for software engineering applications. *Proceedings of Software Engineering Environments*, Reading, UK, 1993, pp. 130-142.

Also available as Technical Report No. 63, Department of Computer Science, University of Dortmund, Germany, 1992.

The ESPRIT-III project GoodStep was aimed at enhancing the O2 object database to make it particularly suitable as a database for Software Engineering Environments (SEEs). To assess the performance of an object database (which dominates the performance of a SEE built on top of it), the Opus-Merlin Benchmarks were defined. The Merlin report contains results for GemStone and GRAS (a prototype graph storage system). This work should also be compared to that discussed earlier in the category **Benchmarks for Language Based Editors**.

Performance Evaluation System for Object Stores (PESOS)

[Rabitti et al. 1993] F. Rabitti, R.S. Sferrazza, M.G. Tori and P. Zezula (1993) Performance evaluation system for object stores. *Proceedings of the International Conference on Database and Expert Systems Applications*, Lecture Notes in Computer Science, 720, pp. 289-300 (Berlin: Springer-Verlag)

This paper describes a software tool for estimating the cost of associative retrieval in very large and complex object stores. The tool, called PESOS, can model various object storage organisations with respect to specific characteristics of the secondary memory hardware. It provides a simple language to specify queries on the object store structures and returns quantitative evaluations, using analytical methods, of the access performance.

SEQUOIA 2000 Benchmark

[Stonebraker 1993] M. Stonebraker, J. Frew, K. Gardels and J. Meredith (1993) The SEQUOIA 2000 Storage Benchmark. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Washington DC, 1993, pp. 2-11.

Also in *Readings in Database Systems (Second Edition)*, M. Stonebraker (Ed.), Morgan-Kaufmann, San Mateo, California, 1993.

The SEQUOIA 2000 Benchmark is based on real data and queries typical of those found in many geographic systems. The authors claim that the benchmark is also representative of other applications, such as engineering, and can therefore be used in a more generic manner. The benchmark consists of ten queries, oriented towards four data types (raster, point, polygon, directed graph) and one query for database load. Results were reported for GRASS (a public domain GIS), IPW (a raster image processing workbench), and POSTGRES (a DBMS research prototype investigating the feasibility of extending the relational model with objects). The most complete benchmark results were available for the POSTGRES implementation, which found a number of problems and limitations of this system.

Simple Benchmark

[Dewal et al. 1992] S. Dewal, W. Emmerich and K. Lichtinghagen (1992) A decision support method for the selection of OMSs. *Proceedings of the Second International Conference on Systems Integration*, Morristown, New Jersey, 1992, pp. 32-40.

[Dewal et al. 1990] S. Dewal, H. Hormann, U. Kelter, D. Platz, M. Roschewski and L. Schope (1990) Evaluation of Object Management Systems. Internal SWT Memo No. 44, University of Dortmund, Germany, September 1990.

[Kelter 1989] U. Kelter (1989) The Simple Benchmark. Technical Report No. UNIDO/UK/89/02, Department of Computer Science, University of Dortmund, Germany, 11 September 1989.

The Simple Benchmark is so named because it focuses on measuring simple operations (create, delete, read, write) on a simple data model, rather than complex operations on a complex data model (e.g. as found in the HyperModel Benchmark). The collection of papers here describe the data model and operations of the Simple Benchmark, with [Dewal et al. 1990] reporting results for a number of systems, such as DAMOKLES, GRAS, PCTE.

University of Southern California (USC) Benchmark

[Ghandeharizadeh et al. 1993] S. Ghandeharizadeh, V. Choi and G. Bock (1993) Benchmarking object-based constructs. *Proceedings of the Eighth Brazilian Symposium on Databases*, Campina Grande, Brazil, 1993, pp. 207-221.

This paper describes a synthetic benchmark with queries that reference inherited functions, and traverse complex object sub-components. Queries are described using terminology found in the Functional Data Model (FDM). Whilst no results are reported, the authors speculate on the behaviour of Iris, ObServer/ENCORE, O2, and GemStone using their benchmark.

Acknowledgements

Thanks to Mike Carey, Carsten Gerlhof, Jim Gray, Udo Kelter, Mark Pedersen, Hartmut Schreiber, Jacob Stein, Wouter Teeuw and Pavel Zezula for pointers to some of the benchmarks presented here.

References

[Anon et al. 1985] Anon et al. (1985) A measure of transaction processing power. *Datamation.* 31 (7):112-118.

[Barry 1994] D. Barry (1994) Should you take the plunge? *Object Magazine.* 3 (6):24-27.

[Bitton et al. 1983] D. Bitton, D. DeWitt and C. Turbyfill (1983) Benchmarking database systems: a systematic approach. *Proceedings of the Ninth Very Large Data Bases Conference*, Florence, Italy, 1983, pp. 8-19.

[Cattell 1991] R.G.G. Cattell (1991) Object data management: object-oriented and extended relational database systems, Addison-Wesley, Reading, Massachusetts.

[Finkelstein 1993] R. Finkelstein (1993) Breaking the mold. *Database Programming & Design.* 6 (2):49-53.

[Inmon 1989] W.H. Inmon (1989) Benchmarking the benchmarks. *Database Programming & Design.* 2 (8):54-59.

[Khoshafian et al. 1992] S. Khoshafian, A. Chan, A. Wong and H.K.T. Wong (1992) *A guide to developing client/server SQL applications*, Morgan-Kaufmann, San Mateo, California.

[Rotzell & Loomis 1991] K. Rotzell and M.E.S. Loomis (1991) Benchmarking an ODBMS. *Journal of Object-Oriented Programming.* 4 (1):66-72.

[Stonebraker 1988] M. Stonebraker (1988) Performance and database machines. In: *Readings in Database Systems*, M. Stonebraker (Ed.), Morgan-Kaufmann, San Mateo.