Benchmarking Object-Oriented DBMSs for Workflow Management*

Anthony J. Bonner[1] bonner@db.toronto.edu

Adel Shrufi[1] shrufi@db.toronto.edu

Steve Rozen[2] `steve@genome.wi.mit.edu`

[1]University of Toronto
Department of Computer Science
10 King's College Rd
Toronto, ON, Canada
M5S 1A4

[2]Whitehead/MIT
Center for Genome Research
One Kendall Square
Building 300, Floor 5
Cambridge, MA 02139, USA

Presented at the *OOPSLA'95 Workshop on Object Database Behaviour, Benchmarks, and Performance*, October 15 1995, Austin, Texas.

This and related papers are available at the following web page:
`http://www.db.toronto.edu:8020/people/bonner/bonner.html`

1 Introduction

Workflow management is a ubiquitous task faced by many organizations, and entails the coordination of various activities. This coordination is increasingly carried out by software systems called *workflow management systems* (WFMS). An important component of many WFMSs is a DBMS for keeping track of workflow activity. This DBMS maintains an audit trail, or event history, that records the results of each activity. Like other data, the event history can be indexed and queried, and views can be defined on top of it. In addition, a WFMS must accommodate frequent workflow changes, which result from a rapidly evolving business environment. Since the database schema depends on the workflow, the DBMS must also support dynamic schema evolution. These requirements are especially challenging in high-throughput WFMSs—*i.e.*, systems for managing high-volume, mission-critical workflows. Fortunately, the flexibility and modelling power of object-oriented databases can ease the development of such systems. However, their performance in a workflow environment remains to be seen. One reason is that existing database benchmarks do not account for the combination of flexibility and performance required by high-throughput WFMSs.

To address this need, we have developed *LabFlow-1*, the first version of a benchmark that concisely captures the DBMS requirements of high-throughput WFMSs. LabFlow-1 is based on the data and workflow management needs of a large genome-mapping laboratory, and reflects their real-world experience with an object-oriented DBMS. In addition, we have used LabFlow-1 to test the usability and performance of two object storage managers. These tests revealed substantial differences between

*This work was supported by funds from the U.S. National Institutes of Health, National Center for Human Genome Research, grant number P50 HG00098, and from the U.S. Department of Energy under contract DE-FG02-95ER62101.

these two systems and highlighted the critical importance of being able to control locality of reference to persistent data. Details of the benchmark and the test results can be found in [3], and to a lesser extent in [4].

1.1 Workflow Management

Examples of workflow management are found in a wide range of industries, from banking and insurance, to telecommunications and manufacturing, to pharmaceuticals and health care [13, 33, 24, 11]. The task is to coordinate the various activities involved in running an enterprise. The activities themselves may use a variety of software components, including files, databases, application programs, and legacy systems, which may run on a variety of hardware platforms and operating systems, which may be located at a variety of sites. For example, in a large genome laboratory, workflow management software knits together a complex web of manual and automated laboratory activities, including experiment scheduling and setup, robot control, raw-data capture, multiple stages of preliminary analysis and quality control, and release of finished results. Appropriate software is necessary to make coordination of these activities both intellectually manageable and operationally efficient, and is a prerequisite for high-throughput laboratories. This software includes a DBMS component for tracking and controlling workflow activity. This paper addresses the performance requirements of this DBMS, and examines the ability of object database technology to meet these requirements.

The LabFlow-1 benchmark concisely describes the database requirements of a WFMS in a high-throughput genome laboratory. Although based on genome-laboratory workflow, we believe that LabFlow-1 captures the database requirements of a common class workflow management applications: those that require a *production workflow system* [19]. In a production workflow system, workflow activities are organized into a kind of production line, involving a mix of human and computer activities. Examples in business include insurance-claim or loan-application processing. Production workflow systems are typically complex, high-volume, and central to the organizations that rely on them; certainly these characteristics apply to the laboratory workflow-management systems used in high-throughput genome laboratories. Many production workflows are organized around central materials of some kind, which the workflow activities operate on. Examples of central materials include insurance claims, loan applications, and laboratory samples. As a central material is processed, workflow activities gather information about it.

Production workflow systems include the class of *Laboratory Information Management Systems*, or LIMS [26, 1, 24]. LIMS are found in analytical laboratories in a wide range of industries, including pharmaceuticals, health care, environmental monitoring, food and drug testing, and water and soil management. In all cases, the laboratory receives a continual stream of samples, each of which is subjected to a battery of tests and analyses. Workflow management is needed to maintain throughput and control quality [23]. In addition, LIMS, like many scientific information systems, must frequently deal with complex-structured data. For instance, workflow in genome laboratories requires database support for object-oriented features, such as complex data types, class hierarchies, and user-defined methods [15]. In fact, object-oriented data models have been specifically developed with laboratory workflow in mind [8].

Much of the research on workflow management in computer science has focussed on developing extended transaction models for specifying dependencies between workflow activities, especially in a heterogeneous environment [13, 25, 10, 18, 17, 12, 31]. However, the *performance* of WFMSs has so far received little attention. The need to study performance arises because commercial products cannot support applications with high-throughput workflows. As stated in [13],

> Commercial workflow management systems typically support no more than a few hundred workflows a day. Some processes require handling a larger number of workflows; perhaps a number comparable to the number of transactions TP systems are capable of handling. For example, telecommunications companies currently need to process ten thousand service provisioning workflows a day, including a few thousand service provisioning workflows per hour at peak hours. Commercial workflow management systems are currently not capable of handling such workloads.

Note that each workflow may involve many transactions.

High throughput workflows are also characteristic of large genome laboratories, like the Whitehead Institute/MIT Center for Genome Research (hereafter called "the Genome Center"). Workflow management is needed to support the Genome Center's large-scale genome-mapping projects [9]. Because of automation in instrumentation, data capture and workflow management, transaction rates at the Genome Center have increased dramatically in the last three years, from processing under 1,000 queries and updates per day in 1992 [14], to over 15,000 on many days in 1995. Of course, peak rates can be much higher, with a rate of 22.5 updates and queries per second recently observed over a 5-minute period. These rates are expected to increase by another order of magnitude in the near future if the Genome Center begins large scale sequencing of the Human genome [9]. Moreover, unlike the simple banking debit/credit transactions of some TPC benchmarks [32], these transactions involve complex queries, and operations on complex objects, such as arrays, sequences, and nested sets. For these reasons, the Genome Center utilizes an OODBMS in tracking workflow activity [15]. LabFlow-1 is motivated by their experience with this system.

2 DBMS Requirements

Workflow management has numerous DBMS requirements. First, it requires standard database features, such as concurrency control, crash recovery, consistency maintenance, a high-level query language, and query optimization. In addition, workflow for scientific laboratories (like other advanced applications) requires database support for object-oriented features, as mentioned earlier. A DBMS supporting production workflow management must provide this functionality on a mixed workload of queries and updates. In addition, it must provide two other features, which are characteristic of production workflow systems:

Event Histories. The DBMS must maintain an audit trail, or event history, of all workflow activity. From this history, the DBMS must be able to quickly retrieve information about any material or activity, for day-to-day operations. The history is also used to explore the cause of unexpected workflow results, to generate reports on workflow activity, and to discover workflow bottlenecks during process re-engineering. The DBMS must therefore support queries and views on an historical database.We note that many commercial laboratories are legally bound to record event histories, since "Accountability is critical in tracking who is responsible for data and its approval for release" [22]. Salient examples include clinical drug trials and environmental testing.

Dynamic Schema Evolution. A hallmark of modern workflow management is that workflows change frequently, in response to rapidly changing business needs and circumstances [11, 33, 13]. Typically, a workflow will acquire new activities and existing activities will evolve. In both cases, the changed workflow generates new kinds of information, which must be recorded in the database. This requires changes (usually additions) to the database schema, preferably while the workflow is in operation (which is called *dynamic workflow modification*).

It is worth observing that because the database is historical and the schema is evolving, data at different points in the history will be stored under different schemas. Thus, an historical query or view may access data with many different schemas. This presents a challenge both to database design and to the formulation of queries and views. For instance, an application program may have to query an object's schema, as well as its value.

Existing database benchmarks do not capture the above requirements. This should not be surprising, as it has been observed by researchers working on OODBMS benchmarks that advanced applications are too complex and diverse to be captured by a single benchmark [5, 7]. A quick glance at several recent benchmarks illustrates their diversity of characteristics and requirements. For instance, the OO1, OO7 and HyperModel benchmarks [6, 5, 2] are concerned with the traversal of large graphs, which is a requirement of engineering and hypertext applications. In contrast, the SEQUOIA 2000 benchmark [36] is concerned with the manipulation of large sets of spatial and image data, such as those found in geographic information systems (GIS). The Set Query benchmark [28] is concerned with queries for decision support, including aggregation, multiple joins and report generation. (Such queries also arise in workflow management—for process re-engineering—but they are only part of the story.) Like these benchmarks, LabFlow-1 specifically targets a broad application area: workflow management. This application is characterized by a demand for *flexible* management of a stream of queries and updates, and of historical data and schema.

On the surface, LabFlow-1 might appear similar to some TPC benchmarks [32, 21], which are also based on a stream of transactions that construct a history. However, the way in which the stream is generated (the workload) is very different. Moreover, the history stored in TPC databases is not queried or indexed,[1] and there is no concern for flexibility and schema evolution. Finally, TPC databases are explicitly relational and contain no complex data types, class hierarchies or user-defined methods. The TPC benchmarks were not intended to capture the requirements of workflow management systems.

Unfortunately, many DBMSs do not yet fully support the requirements of production workflow systems as described above. Certainly, a surprising number of commercial products showed serious flaws in simple tests performed by the Genome Center in 1991. Fortunately, one can build a specialized DBMS that supports workflow on top of a storage manager that does not. This approach is taken at the Genome Center. Their specialized DBMS (called *LabBase* [30, 35, 29]) provides the needed support for event histories and schema evolution on top of an object storage manager. LabBase provides a historical query language, as well as structures for rapid access into history lists. It also transforms data from the user's database schema (which is dynamic) into the storage manager's schema (which is static). Besides providing support for workflow, this approach also provides portability, since different object storage managers can be "plugged into" the DBMS. In this way, we can test a wide range of

[1] In particular, the "`history`" relation never appears in the `from` clause of an SQL statement.

existing storage managers. Our implementation of the LabFlow-1 benchmark is based on this idea [3]. We emphasize, however, that LabFlow-1 does not depend on LabBase, which is an implementation detail. In the future, we hope to use our benchmark to test the support for workflow management in "off-the-shelf" DBMSs.

3 The LabFlow-1 Benchmark

The LabFlow-1 benchmark is based on the data and workflow management needs of the Genome Center, and reflects their real-world experience. The goals of the benchmark are two-fold. One goal is to provide a tool for the Genome Center to use in analyzing object storage managers for LabBase. Other goals are (*i*) to provide a general benchmark for databases supporting workflow-management systems, and (*ii*) to provide developers of next-generation DBMS technology with a set of requirements based on the characteristics of a real-world application domain. Developers of new technology often have few if any realistic applications against which to measure their systems. As with any benchmark, the challenge in designing LabFlow-1 was to create a database schema and workload that are realistic enough to be representative of the target class of applications yet simple enough to be feasibly implemented on a variety of systems.

Although LabFlow-1 is intended to be a general benchmark for DBMSs, we have so far used it to compare storage managers only. This is achieved by running the benchmark on versions of LabBase implemented on top of different object storage managers, as described above. In [3, 4], we compare ObjectStore (version 3.0) [20, 27] and Texas (version 0.3) [34, 37]. Compared to relational systems, these storage managers have been used in few production applications, so this analysis is interesting in its own right. Since they are a relatively novel technology, we compare these storage managers not only in terms of performance, but also in terms of client interface, tuning options, and system-administration facilities. In a similar fashion, LabFlow-1 can be used to compare other DBMS components, such as query optimizers and historical access structures.

LabFlow-1 is a preliminary benchmark, intended for a single database user (hence the name). This choice arises from the architecture of the Genome Center's production DBMS, and also from a belief that it is useful to understand single-user performance before attempting to understand multi-user performance. In the future, we plan to extend LabFlow-1 to a multi-user, client/server benchmark.

In sum, LabFlow-1 is the first version of a benchmark for DBMSs that control and track workflow. It is designed for applications with the following characteristics and requirements:

- high volume, mission critical workflows;
- frequent workflow change and process re-engineering;
- an audit trail of workflow activity;
- complex-structured data.

An overview of the benchmark can be found in [4], and a detailed description can be found in [3]. Benchmark software is available at the following web site:

`ftp://db.toronto.edu/pub/bonner/papers/workflow/software/`

References

[1] *Standard Guide for Laboratory Information Management Systems (LIMS).* American Society for Testing and Materials, 1916 Race St., Philadelphia PA 19103, U.S.A, 1993.

[2] T.L. Anderson, A.J. Berre, M. Mallison, H.H. Porter, and B. Schneider. The hypermodel benchmark. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, pages 317–331, Venice, Italy, March 1990.

[3] A. Bonner, A. Shrufi, and S. Rozen. LabFlow-1: a database benchmark for high-throughput workflow management. Technical report, Department of Computer Science, University of Toronto, 1995. 53 pages. Available at `http://www.db.toronto.edu:8020/people/bonner/bonner.html`.

[4] A. Bonner, A. Shrufi, and S. Rozen. LabFlow-1: a database benchmark for high-throughput workflow management. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, number 1057 in Lecture Notes in Computer Science, pages 463–478, Avignon, France, March 25–29 1996. Springer-Verlag.

[5] M.J. Carey, D.J. DeWitt, and J.F. Naughton. The OO7 benchmark. Technical report, Computer Sciences Department, University of Wisconsin-Madison, January 1994. Available at `ftp://ftp.cs.wisc.edu/oo7/techreport.ps`.

[6] R.G.G. Cattell. An engineering database benchmark. In [16], chapter 6, pages 247–281.

[7] A. Chaudhri. An Annotated Bibliography of Benchmarks for Object Databases. *SIGMOD Record*, 24(1):50–57, March 1995.

[8] I-Min A. Chen and Victor M. Markowitz. The Object-Protocol Model, version 3.0. Technical Report LBL-32738, Lawrence Berkeley Laboratory, 1 Cyclotron Road, Berkeley, CA, 94720, USA, December 1994. This document and others on OPM available via `http://gizmo.lbl.gov/DM_TOOLS/OPM/opm.html`.

[9] *Communications of the ACM*, 34(11), November 1991. Special issue on the Human Genome Project.

[10] U. Dayal, H. Garcia-Molina, M. Hsu, B. Kao, and M.-C. Shan. Third generation TP monitors: A database challenge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 393–397, Washington, DD, May 1993.

[11] E. Dyson. Workflow. In *Forbes*, page 192. November 23 1992.

[12] A. K. Elmagarmid, editor. *Database Transaction Models for Advanced Applications*. Morgan Kaufmann, San Mateo, CA, 1992.

[13] D. Georgakopoulos, M. Hornick, and A. Sheth. An overview of workflow management: From process modeling to infrastructure for automation. *Journal on Distributed and Parallel Database Systems*, 3(2):119–153, April 1995.

[14] Nathan Goodman. An object oriented DBMS war story: Developing a genome mapping database in C++. In Won Kim, editor, *Modern Database Management: Object-Oriented and Multidatabase Technologies*. ACM Press, 1994.

[15] Nathan Goodman, Steve Rozen, and Lincoln Stein. Building a laboratory information system around a C++-based object-oriented DBMS. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, September 1994. Available at `ftp://genome.wi.mit.edu/pub/papers/Y1994/building.ps.Z`.

[16] Jim Gray, editor. *The Benchmark Handbook for Database and Transaction Processing Systems*. Morgan Kaufmann, San Mateo, CA, 1991.

[17] M. Hsu, Ed. Special issue on workflow and extended transaction systems. *Bulletin of the Technical Committee on Data Engineering (IEEE Computer Society)*, 16(2), June 1993.

[18] M. Hsu, Ed. Special issue on workflow systems. *Bulletin of the Technical Committee on Data Engineering (IEEE Computer Society)*, 18(1), March 1995.

[19] Setrag Khoshafian and Marek Buckiewicz. *Introduction to Groupware, Workflow, and Workgroup Computing*. John Wiley & Sons, Inc., 1995.

[20] Charles Lamb, Gordon Landis, Jack Orenstein, and Dan Weinreb. The ObjectStore database system. *Communications of the ACM*, 34(10):50–63, October 1991.

[21] S. Leutenegger and D. Diaz. A Modeling Study of the TPC-C Benchmark. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 22–31, Washington, D.C., May 1993.

[22] D.C. Mattes. LIMS and good laboratory practice. In [24], pages 332–345. 1985.

[23] R.D. McDowall. Introduction to laboratory information management systems. In [24], pages 1–16. 1985.

[24] R.D. McDowell, editor. *Laboratory Information Management Systems: Concepts, Integration, Implementation*. Sigma Press, Wilmslow, U.K., 1985.

[25] C. Mohan. Tutorial: A survey and critique of advanced transaction models. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, page 521, Minneapolis, MN, May 1994. Tutorial.

[26] Allen S. Nakagawa. *LIMS: Implementation and Management*. Royal Society of Chemistry, Thomas Granham House, The Science Park, Cambridge CB4 4WF, England, 1994.

[27] Object Design, Inc., 25 Burlington Mall Rd., Burlington MA 01803-4194, USA. Manual set for ObjectStore Release 3.0 for UNIX Systems, December 1993.

[28] P. O'Neal. The set query benchmark. In [16], chapter 5, pages 209–245.

[29] S. Rozen and L. Stein and N. Goodman. Labbase User Manual. Available at `ftp://genome.wi.mit.edu/pub/papers/Y1994/labbase-manual.ps`.

[30] Steve Rozen, Lincoln Stein, and Nathan Goodman. Constructing a domain-specific DBMS using a persistent object system. In M.P. Atkinson, V. Benzaken, and D. Maier, editors, *Persistent Object Systems*, Workshops in Computing. Springer-Verlag and British Computer Society, 1995. Presented at POS-VI, Sep. 1994. Available at `ftp://genome.wi.mit.edu/pub/papers/Y1994/labbase-design.ps.Z`.

[31] M. Rusinkiewicz and A. Sheth. Specification and execution of transactional workflows. In W. Kim, editor, *Modern Database Systems: The Object Model, Interoperability, and Beyond*. Addison-Wesley, 1994.

[32] O. Serlin. The history of debit credit and the TPC. In [16], chapter 2, pages 19–117.

[33] A. Sheth. Workflow automation: Applications technology and research. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, page 469, San Jose, CA, May 1995. Tutorial. Slides available at `http:/www.cs.uga.edu/LSDIS`.

[34] Vivek Singhal, Sheetal V. Kakkad, and Paul R. Wilson. Texas: an efficient, portable persistent store. In *Proceedings of the Fifth International Workshop on Persistent Object Systems (POS-V)*, San Minato, Italy, September 1992. Available at `ftp://cs.utexas.edu/pub/garbage/texaspstore.ps`.

[35] Lincoln Stein, Steve Rozen, and Nathan Goodman. Managing laboratory workflow with LabBase. In *Proceedings of the 1994 Conference on Computers in Medicine (CompMed94)*. World Scientific Publishing Company, 1995. In press. Available at `ftp://genome.wi.mit.edu/pub/papers/Y1995/workflow.ps.Z`.

[36] M. Stonebraker, J. Frew, K. Gardels, and J. Meredith. The Sequoia 2000 storage benchmark. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 2–11, Minneapolis, MN, May 1993.

[37] Paul R. Wilson and Sheetal V. Kakkad. Pointer swizzling at page fault time: Efficiently and compatibly supporting huge address spaces on standard hardware. In *International Workshop on Object Orientation in Operating Systems*, 1992. Available at `ftp://cs.utexas.edu/pub/garbage/swizz.ps`.