SMART: Making DB2 (More) Autonomic

Guy M. Lohman

IBM Almaden Research Center
K55/B1, 650 Harry Rd.
San Jose, CA 95120-6099
U.S.A.
lohman@almaden.ibm.com

Sam S. Lightstone

IBM Toronto Software Lab
8200 Warden Ave.
Markham, L6G 1C7 Ontario
Canada
light@ca.ibm.com

Abstract

IBM's SMART (Self-Managing And Resource Tuning) project aims to make DB2 self-managing, i.e. autonomic, to decrease the total cost of ownership and penetrate new markets. Over several releases, increasingly sophisticated SMART features will ease administrative tasks such as initial deployment, database design, system maintenance, problem determination, and ensuring system availability and recovery.

1. Motivation for Autonomic Databases

While Moore's Law and competition decrease the per-unit cost of hardware and software, the shortage of skilled professionals that can comprehend the growing complexity of information technology (IT) systems increasingly hampers the growth of IT systems and dominates their total cost of ownership. More than half a century ago, the telephone industry responded to a similar projected shortage of telephone operators by automating its switching systems. The continued growth of the IT industry demands that we automate many of the functions that are now performed by humans, both to save cost and to penetrate new markets. Such automation has been called *Autonomic Computing* in a recent manifesto by IBM's Director of Research, Paul Horn [http://www.research.ibm.com/autonomic/], by analogy to the autonomous nervous system in mammals that regulates digestion, pupil dilation, heart rate, sweating, etc. without conscious thought.

**Proceedings of the 28th VLDB Conference,
Hong Kong, China, 2002**

The database community has already made many significant contributions toward autonomic systems. Separating the logical schema from the physical schema, permitting different views of the same data by different applications, and the entire relational model of data, all simplified the task of building new database applications. Declarative query languages such as SQL, and the query optimizers that made them possible, further aided developers. But with the exception of early research in the late 1970s and early 1980s on database design algorithms, little has been done to help the beleaguered database administrator (DBA) until quite recently, with the founding of the AutoAdmin project at Microsoft [http://www.research.microsoft.com/dmx/autoadmin/] and the SMART project at IBM, described herein.

2. The SMART Project

IBM's SMART (Self-Managing And Resource Tuning) project aims to make DB2 autonomic [6]. SMART is the database part of IBM's overall Autonomic Computing Initiative, and project eLiza is the server part. Founded in the spring of 2000, the SMART project is a formalization of a technology theme that DB2 has been pursuing for many years with prior features [9], including its cost-based query optimizer for automatic plan selection [5], its Configuration Wizard, Index Advisor, and several features for automatic integrity verification and failure recovery. The project now involves 2 IBM Research Centers, 2 IBM development laboratories, and university partners at over 6 universities through IBM Canada's Centre for Advance Studies. The work targets DB2 on both the mainframe (z/OS and OS/390) platform and the Linux, Unix, and Windows platforms, but due to space constraints, in this paper we will primarily highlight features in DB2 for Linux, Unix, and Windows.

SMART encompasses five major components:

- "Up and Running", including initial capacity planning, installation, configuration, deployment, etc.
- Design, including both logical and physical design

- Maintenance, including on-going tuning and physical maintenance such as reorganization, backups, statistics collection, etc.
- Problem determination, notification, and correction
- Availability and disaster recovery

3. Some SMART Features

While current versions of DB2 already have many autonomic features, the SMART project will produce an increasingly sophisticated stream of new autonomic features over the next few releases of DB2. Some of these are infrastructure that enables more autonomic behavior, such as mechanisms to permit changing most configuration parameters dynamically without having to quiesce all applications. Several are *advisors*, which systematically model alternative designs or configurations, and recommend a preferred configuration. The most sophisticated are autonomous and self-validating systems that actually learn from past experience to improve their decisions. Our approach is to "walk before we run", by developing tools that ease the DBA's load but keep the DBA in the loop. Once we earn the DBA's trust, we can increasingly automate the function. We will briefly describe some of the major features we are working on.

3.1 Up and Running

The shipped version of DB2 for Linux, Unix, and Windows (V7.2) already has a Performance Configuration Wizard, which will set approximately 36 of the configuration parameters most important to system performance. Significant enhancements have been made to the model of this wizard, which will be called the Configuration Advisor in the next release. The settings chosen automatically by the Configuration Advisor have achieved performance on industry-standard OLTP benchmarks that have come within 98.4% of the performance achieved by DB2's performance experts, and in a customer benchmark actually outperformed the settings of the customer's DBAs. [10]

3.2 Design

In Version 6 of DB2 for Linux, Unix, and Windows, we introduced DB2's Index Advisor [1][9], which automatically determines the best set of indexes for a given workload and space constraint. Like Microsoft's Index Wizard [3,4], our Index Advisor uses the optimizer's model to analyze "what if?" scenarios using the query optimizer. However, we also have the optimizer recommend good candidate indexes, significantly improving the quality of candidates and reducing the search space. In the next release, we will be extending the Index Advisor to also recommend simultaneously Materialized Query Tables (i.e., materialized views) [2] and partitioning of tables in a shared-nothing environment [8]. This combined tool will be called the Design Advisor, headquarters for all design assistance.

3.3 Maintenance

A Maintenance Advisor, available in the next release, will utilize database statistics to determine which objects need which kind of maintenance, to estimate the time needed to perform that maintenance, and schedule that maintenance within a given time window, if need be.

DB2 for z/OS v7 has introduced real-time statistics that track how each database object has been modified, enabling autonomic identification of objects needing various kinds of maintenance. The real-time statistic approach is being investigated for adoption within DB2 for Linux, Unix, and Windows as well. Lastly, the LEarning Optimizer (LEO) project is developing ways to make the DB2 query optimizer self-validate its cardinality model, using the actual cardinalities measured by executing queries with similar predicates, actually learning from prior experience [7].

3.4 Problem Determination and Correction

One of the most daunting DBA tasks is detecting, isolating, and correcting problems. The Health Center will attack this problem by continuously monitoring DB2's health and alerting the DBA by e-mail, pager, or logs of any problems, as well as possible solutions. Great care has been taken to minimize the overhead of this monitoring, "drilling down" to get more detailed information only when a problem is suspected. Ultimately, the goal is to have the Health Center actually take action when it is certain of the correction, but we must first earn the DBA's trust. Perhaps the most challenging task is to have the Health Center learn from human DBAs over time, so that fixes to problems it couldn't solve will be added automatically to its (and others'!) knowledge base over time.

3.5 Availability and Disaster Recovery

Already announced, the DB2 Recovery Expert is a tool that automatically determines the optimal way to recover a set of tables to a specific point in time, by analysing the log and the recovery assets available. For example, depending upon when each backup was completed, this might be either a rollback from the present, or a recovery of a backup followed by a roll-forward to the desired time.

4. Summary

DB2's SMART project is developing a stream of autonomic technology – not just GUIs and handholding -- that will actually make DB2 more self-managing. As part of IBM's Autonomic Computing initiative, SMART will

also cooperate with autonomic features in other IBM components (e.g., Websphere) to synergistically make the overall system autonomic. While some of SMART's projects are DB2-specific, some of the most challenging problems are much more generic, and would benefit from academic collaboration. The ultimate goal of complete automation may be many years away, but we are already making life a lot better for DBAs, and reducing total cost of ownership for enterprises.

References

1. G. Lohman, G. Valentin, D. Zilio, M. Zuliani, A Skelly, "DB2 Advisor: An optimizer smart enough to recommend its own indexes," Proceedings, 16th IEEE Conference on Data Engineering, San Diego, CA, 2000.
2. Sanjay Agrawal, Surajit Chaudhuri, and Vivek R. Narasayya. "Automated Selection of Materialized Views and Indexes for SQL Databases," *Proceedings, 26th International Conference on Very Large Databases*, 2000, pp. 496-505.
3. S. Chaudhuri, V. Narasayya, "AutoAdmin 'What-if' Index Analysis Utility," *Proceedings, 1998 ACM SIGMOD Conference*, Seattle 1998, pp. 367-378.
4. S. Chaudhuri, V. Narasayya, "Microsoft Index Tuning Wizard for SQL Server 7.0," *Proceedings, 1998 ACM SIGMOD Conference*, Seattle 1998, pp. 553-554.
5. P. Gassner, G. M. Lohman, K. B. Schiefer, Y. Wang. "Query Optimization in the IBM DB2 Family", IEEE Data Engineering Bulletin 16(4), 1993, pp. 4-18.
6. D. C. Zilio, S. Lightstone, K. A. Lyons, G. M. Lohman, "Self-Managing Technology in IBM DB2 Universal Database", Procs. Of 2001 CIKM, 2000, pp. 541-543.
7. M. Stillger, G. M. Lohman, V. Markl, M. Kandil, "LEO - DB2's LEarning Optimizer", Procs. 27th Intl. Conf. On Very Large Databases, Rome, Italy, 2001, pp. 19-28.
8. J. Rao, C. Zhang, G. Lohman, N. Megiddo, "Automating Physical Database Design in a Parallel Database System", Proc. 2002 ACM SIGMOD Intl. Conf. On Management of Data, Madison, WI, 2002
9. B. Schiefer and G. Valentin. "DB2 Universal Database Performance Tuning", IEEE Data Engineering Bulletin 22(2), June 1999, pp. 12-19.
10. E. Kwan, S. Lightstone, A Storm, L. Wu, IBM Server Group, "Automatic Configuration for IBM DB2 Universal Database", online, http://www.redbooks.ibm.com/redpapers/pdfs/redp0441.pdf