A Survey of Open Source Tools for Business Intelligence

Christian Thomsen and Torben Bach Pedersen

Department of Computer Science, Aalborg University

Abstract. The industrial use of open source Business Intelligence (BI) tools is not yet common. It is therefore of interest to explore which possibilities are available for open source BI and compare the tools.

In this survey paper, we consider the capabilities of a number of open source tools for BI. In the paper, we consider three Extract-Transform-Load (ETL) tools, three On-Line Analytical Processing (OLAP) servers, two OLAP clients, and four database management systems (DBMSs). Further, we describe the licenses that the products are released under.

It is argued that the ETL tools are still not very mature for use in industry while the DBMSs are mature and applicable to real-world projects. The OLAP servers and clients are not as powerful as commercial solutions but may be useful in less demanding projects.

1 Introduction

The use of Business Intelligence tools is popular in industry [25,29,30]. However, the use of open source tools seems to be limited. The dominating tools are closed source and commercial (see for example [25] for different vendors' market shares for OLAP servers). Only for database management systems (DBMSs), there seems to be a market where open source products are used in industry, including business-critical systems such as online travel booking, management of subscriber inventories for tele communications, etc. [23]. Thus, the situation is quite different from, for example, the web server market where open source tools as Linux and Apache are very popular [38].

To understand the limited use of open source BI tools better, it is of interest to consider which tools are available and what they are capable of. This is the purpose of this paper. In the European Internet Accessibility Observatory (EIAO) project, where accesibility data is collected, it is intended to build a BI solution based on open source software. It is therefore of relevance for this project to investigate the available products.

In the survey we will consider products for making a complete solution with an Extract-Transform-Load (ETL) tool that loads data into a database managed by a DBMS. On top of the DBMS, an On-Line Analytical Processing (OLAP) server providing for fast aggregate queries will be running. The user will be communicating with the OLAP server by means of an OLAP client. We limit ourselves to these kinds of tools and do not consider, for example, data mining tools or Enterprise Application Integration (EAI) tools. Use of data mining tools would also be of relevance in many BI settings, but data mining is a more

A Min Tjoa and J. Trujillo (Eds.): DaWaK 2005, LNCS 3589, pp. 74–84, 2005.

advanced feature which should be considered in future work. EAI tools may have some similarities with ETL tools, but are more often used in online transactional processing (OLTP) systems.

The rest of the paper is structured as follows. Section 2 gives a primer on open source licenses. Section 3 presents the criteria used in the evaluation of the different tools. Section 4 considers ETL tools. Section 5 deals with OLAP servers, while Section 6 deals with OLAP clients. Finally, we consider DBMSs in Section 7 before concluding and pointing to future work in Section 8.

2 Open Source Licenses

To make the findings on licenses more comprehensible, we include a description of the open source licenses that will be referred to later in the paper. The *GNU General Public License (GPL)* [11] is a classic, often used open source license. Any user is free to make changes to the source code. If the changed version is only used privately, it is not a requirement that its source code is released. If it, however, is distributed in some way, then the source code must be made available under the GPL (i.e. also released as open source that any user is free to change and copy). It should be noted that a library released under the GPL will require any program that uses it to be licensed under the GPL. This is not the case when using *GNU Library General Public License (LGPL)* [12] which apart from that is much like the GPL. The *Common Public License (CPL)* [7] was developed by IBM as an open source license. Like the GPL, the CPL requires that the source code for a modified version of a program is made publicly available if the new version is distributed to anyone. Programs and libraries released under the CPL may be used from and integrated with software released under other (also closed source) licenses.

The *Mozilla Public License* [22] is also an open source license that requires the code for any distributed modified works to be made publicly available. It is allowed to use a library under the Mozilla Public License from a closed source application. Thus the license has some similarities with the LGPL. The *Apache License* [1], [2] allows the code to be used both in open source, free programs and in commercial programs. It is also possible to modify the code and redistribute it under another license under certain conditions (e.g. the use of the original code should be acknowledged). Version 1.0 and 1.1 of the Apache License [1] included requirements about the use of the name "Apache" in documentation and advertising materials. That meant that the license should be modified for use in non-Apache projects. This was changed with version 2.0 [2]. The *BSD License* [5] is a very liberal open source license. It is permitted to use source code from a BSD licensed program in a commercial, closed source application. As long as any copyright notices remain in the modified code, there are no requirements saying that modifications of the code should be BSD licensed or open source.

3 Conduct of the Survey

In this section, we present the criteria used for the evaluation of the considered products. The criteria with a technical nature have been inspired by the

functionality offered by the leading commercial BI tools. Other criteria, such as the type of license, are interesting when looking at open source tools. Based on the criteria given below, we collected data about the products (all found on the Internet) by examining their source code, available manuals, and homepages including any forums. The findings were collected in Nov-Dec. 2004.

Criteria for All Categories. When deciding between different products, a potential user would most often prefer a product that is compatible with his[1] existing operating system and hardware. Thus, for all the products, it is of interest to investigate which hardware and software platforms the tools are available for. In this survey we will only look at open source products. As described in Section 2 there are, however, many different open source licenses that have different permissions and restrictions. Therefore, the license used by a product is also of great interest.

Criteria for ETL Tools. When comparing ETL tools, there are several criteria to consider. First, it should be considered which data sources and targets a given tool supports. Here, it should be considered whether the tool is for loading data into ROLAP or MOLAP systems, i.e. into relational tables or multidimensional cubes [30]. In many practical applications, it should be possible to extract data from different sources and combine the data in different ways. Further, it should be possible to load the data into different tables/cubes. Therefore support for these issues should be considered. It should also be considered which types of data sources an ETL tool can extract data from and whether it supports incremental load in an automatic fashion (not requiring two separate flows to be specified). It is also of interest how the ETL process is specified by the user, i.e whether a graphical user interface (GUI) exists and if the user can specify the process directly by means of some specification language. Another important issue for ETL tools is their capabilities for data cleansing. Here it should be considered how data cleansing is supported, i.e. if predefined methods exist and how the user can specify his own rules for data cleansing.

Criteria for OLAP Servers. For an OLAP server it is of interest to know how it handles data. It should thus be considered whether the tool is ROLAP, MOLAP, or HOLAP oriented, where HOLAP is short for Hybrid OLAP [30]. Further, it is of interest if the product is capable of handling large data sets (for example, data sets greater than 10 gigabytes). It should also be taken into account whether an OLAP server has to be used with a specific DBMS or if it is independent of the underlying DBMS. Precomputed aggregates can in many situations lead to significant performance gains. It is therefore relevant to see whether an OLAP server can use aggregates and if so, whether the user can specify which aggregates to use. Finally, it is of relevance to investigate which application programming interfaces (APIs) and query languages an OLAP server supports. A product that uses standards or de-facto standards is much more useful with other tools than a product using a non-standard API or query language.

[1] We use "his" as short for "his/her".

Criteria for OLAP Clients. For an OLAP client it should be considered which OLAP server(s) the OLAP client can be used with. As for OLAP servers, it should also be taken into account which API(s) and query language(s) the OLAP client supports. With respect to reports, it is interesting to see if the OLAP client supports prescheduled reports, perhaps through a server component. If so, the user could, for example, make the OLAP client generate a sales report every Friday afternoon. When a report has been generated (manually or as prescheduled report), it is often useful to be able to export the report to some common format that could be emailed to someone else. Therefore, it should be investigated which export facilities an OLAP client offers. In generated reports, different types of graphs are often used. It should thus also be considered how well an OLAP clients supports different kinds of graphs.

Criteria for DBMSs. There are many possible criteria to consider for DBMSs. In this survey we will, however, only look at criteria directly relevant for BI purposes. First of all, a DBMS should be capable of handling large data sets if the DBMS is to be used in BI applications. Thus this is an issue to investigate. When choosing a DBMS for BI, it is also of relevance which performance improving features the DBMS offers. In this survey we will look into the support for materialized views that can yield significant performance gains for precomputed aggregates. Many commercial ROLAP systems use bitmap indices to achieve good performance [30]. It is also of interest to find out whether these are supported in the considered products. Further, in a typical schema for a data warehouse, star joins may be a faster to use and, thus, the support for these is an issue. Finally, we will consider partitioning which can yield performance improvements and replication which may improve performance and reliability.

4 ETL Tools

In this section, we will consider the three ETL tools Bee, CloverETL, and Octopus. These were all the available tools we found. We found many other open source ETL projects that were not carrying any implementation, but more or less only stated objectives. Examples of such projects are OpenSrcETL [27] and OpenETL [26]. Another disregarded project, was cplusql [8] which had some source code availble, but for which we did not find any other information.

Bee. Bee version 1.1.0 [3] is a package consisting of an ETL tool, an OLAP server, and an OLAP client web interface. The ETL tool and the OLAP server of Bee are ROLAP oriented. Bee is available under both an open source GPL license and a commercial license. Bee is implemented mainly in Perl with parts implemented in C. Therefore, the access to data is provided by the Perl module DBI. Bee comes with its own driver for comma-separated files. To extract data, Bee needs a small server application (included) to be running on the host holding the data. Bee is primarily written for Linux, but is also running on Windows platforms. The mentioned server application needed for extracting data runs on different varieties of UNIX and Windows. The ETL process can be specified by

means of a GUI. The GUI will create an XML file defining the process. It is thus also possible for the user to use Bee without using the GUI by creating the XML file manually. It is possible to have several flows and combine them. It is also possible to insert into more than one table in the database. There seems to be no support for automatic incremental loading. The possibility for data cleansing is introduced by means of allowing the user to write custom transformations in Perl. A standard library of transformations is not included. Thus the user needs to program any needed transformation.

CloverETL. CloverETL version 1.1.2 [6] is also a ROLAP oriented ETL tool. Parts of it are distributed under the GPL license whereas other parts are distributed under the LGPL license. CloverETL is implemented in Java and uses JDBC to transfer data. The ETL process is specified in an XML file. In this XML file, a directed graph representing the flow must be described. Currently, CloverETL does not include a GUI, but work is in progress with respect to this. CloverETL supports combination of several flows as well as import to several tables in the database. There is no support for automatic incremental load. With respect to cleansing, CloverETL supports insertion of a default value, but apart from this, the user will have to implement his own transformations in Java.

Octopus. Octopus version 3.0.1 [9] is a ROLAP oriented ETL tool under the LGPL license. It is implemented in Java and is capable of transferring data between JDBC sources. Octopus is bundled with JDBC drivers for XML and comma-separated files. Further, it is possible to make Octopus create SQL files with insert and DDL statements that can be used for creating a database holding the considered data. Like Bee, Octopus is shipped with a GUI that creates an XML file specifying the ETL process. Octopus can also be used without the GUI and as a library. Octopus is created for transferring data between one JDBC source and another. It is apparently not possible to combine data from one database with data extracted from another. It is possible to extract data from more than one table in the same database as well as insert into more than one table in the target database. There is no direct support for automatic incremental loading. Basic data cleansing functionality is provided. It is possible to make Octopus insert a default value, shorten too long strings, replace invalid foreign key values, find and replace values, do numeric conversions, and change date formats. These cleansings are done by predefined transformations. The user can also implement transformations on his own in Java and JavaScript.

General Comments. The considered open source ETL tools are still not as powerful as one could wish. For example, most of the data cleansing to be done must be coded by the user (with the exception of Octopus which provides some default transformations for very basic data cleansing). Further, the products do not support automatic incremental load which would be very useful for everyday use of the products. In general, the quality of the documentation for the described products is not very good or comprehensive. Further, not much documentation is available. An exception is again Octopus for which a manual of more than 120 pages is available. However, also this manual is not complete. For example, it

explains how to set up which *logger* to use but does not tell about the differences between the available loggers. Thus the quality of the open source ETL products is still not as high as the quality of many commercially available products. Indeed it would probably be difficult to use one of the open source ETL tools for a demanding load job in an enterprise data warehouse environment.

5 OLAP Servers

In this section, we will consider the three OLAP servers Bee, Lemur, and Mondrian. Another possible candidate for consideration would be pocOLAP [31] which, however, in the documentation is said not to be an OLAP server. It provides access to data from DBMS through a web-interface but is not intended provide advanced OLAP functionality or real-time data analysis. OpenRolap [28] is a related tool which generates aggregate tables for a given database. Apart from these tools we did not find any candidates. As for the ETL tool category, there exist other projects that currently carry no code, but only state objectives. An example of such a project is gnuOLAP [13].

Bee. The OLAP server of the Bee package is, as previously stated, a ROLAP oriented server. It uses a MySQL system to manage the underlying database and aims to be able to handle up to 50GB of data efficiently [4]. Despite this, it does not seem to be possible to choose which precomputed aggregates to use. From the documentation, it is not clear which query language(s) and API(s) Bee supports. In general, there is not much English documentation available for Bee, neither from the homepage [3], nor in the downloadables.

Lemur. Unlike the other OLAP servers considered in this paper, Lemur [17] is a HOLAP oriented OLAP server. It is released under the GPL license and is written in C++ for Linux platforms, but is portable. Lemur is a product under development and still has no version number. The homepage for the Lemur project [17] states that for now, the primary goal is to support the developers research interests and that their goals are believed to be too ambitous to deliver usable code now. This is also reflected in the fact that the API is still being designed and in reality is not available for use from outside the Lemur package. Further the user would need to implement methods to load data from a database on his own. It is also not possible to specify the aggregates to be used. No information on how well Lemur scales when applied to large data sets has been found. In summary, the Lemur project is not of much practical use for industry projects so far. However, the goal of eventually producing a HOLAP oriented server outperforming Mondrian (see below) is interesting.

Mondrian. Mondrian 1.0.1 [19] is an OLAP server implemented in Java. It is ROLAP oriented and can, unlike Bee, be used with any DBMS for which a JDBC driver exists. Mondrian is released under the CPL license [7]. The current version of Mondrian has an API that is similar to ADO MD from Microsoft [20]. Support for the standard APIs JOLAP [14] and XMLA [39] is planned. Further, the MDX query language [35], known from Microsoft's products, is supported by

Mondrian. By default Mondrian will use some main memory for caching results of aggregation queries. It is, however, neither possible for the user to specify what should be cached nor which aggregates should exist in the database. The documentation states that Mondrian will be able to handle large data sets if the underlying DBMS is, since all aggregation is done by the DBMS.

General Comments. The Mondrian OLAP server seems to be the best of the described products. Lemur is for the time being not usable for real world applications while it is difficult to judge Bee because of its lack of English documentation. Mondrian is, however, a usable product which works with JDBC-enabled DBMSs. For none of the products, it seems possible to choose which aggregates to use. In most environments this feature would result in significant performance improvements.

6 OLAP Clients

In this section, we will describe the OLAP clients Bee and JPivot. These were the found open source OLAP clients that are actually implemented.

Bee. The Bee project also provides an OLAP client. The client is web-based and is used with the Bee OLAP server. Currently, Microsoft Internet Explorer and Mozilla browsers are explicitly supported in the downloadable code. Again, it has not been possible to determine which API(s) and query language(s) Bee supports. The Bee OLAP client can interactively present multidimensional data by means of Virtual Reality Modeling Language (VRML) technology [37]. Bee can generate different types of graphs (pie, bar, chart, etc.) in both 2D and 3D. It is possible to export data from Bee to Excel, Portable Document Format (PDF), Portable Networks Graphics (PNG), PowerPoint, text, and Extensible Markup Language (XML) formats. Connection with the statistical package R [33] is also evaluated. It does not seem to be possible to preschedule reports.

JPivot. JPivot version 1.2.0 [16] is a web-based OLAP client for use with the Mondrian OLAP server. However, the architecture should allow for later development of a layer for XMLA [39]. As Mondrian, JPivot uses MDX as its query language. It is written in Java and JSP. JPivot generates graphs by means of JFreeChart [15] which provides different kinds of 2D and 3D graphs. With respect to export of reports, JPivot is limited to Portable Document Format (PDF) and Excel format. Support for prescheduled reports has not been found. JPivot is released under a license much like the Apache Software License Version 1.1 [1] (but without restrictions regarding the use of the name "Apache"). However, other software packages are distributed with JPivot and have other software licenses, e.g. JFreeChart which uses the LGPL license.

General Comments. Both the considered OLAP clients are to be used with specific OLAP servers, namely Bee with Bee and JPivot with Mondrian. Both of them are web-based such that specific software does not have to be installed at client machines already equipped with a browser. Both products are capable of

exporting generated reports to other commonly used file formats such as PDF, but neither of them supports prescheduled reports.

7 DBMSs

In this section we consider four open source DBMSs: MonetDB, MySQL, MaxDB, and PostgreSQL. Other open source DBMSs are available, but these four were chosen as they are the most visible, well-known high-performance DBMSs.

MonetDB. MonetDB, currently in version 4.4.2, is developed as a research project at CWI. MonetDB is "designed to provide high performance on complex queries against large databases, e.g. combining tables with hundreds of columns and multi-million rows" [21]. To be efficient, MonetDB is, among other techniques, exploiting CPU caches and full vertical fragmentation (however, the fragments must be placed on the same disk). It thus uses very modern and often hardware-near approaches to be fast. MonetDB is mainly implemented in C with some parts in C++. It is available for 32- and 64-bit versions of Linux, Windows, MacOS X, Sun Solaris, IBM AIX, and SGI IRIX. MonetDB comes with a license like the Mozilla Public License (references to "Mozilla" are replaced by references to "MonetDB") [21]. With respect to features often usable in a BI context, it is interesting to notice that MonetDB does not support bitmap indices, materialized views (normal views are supported), replication, or star joins. However, this does not mean that MonetDB is not usable for BI purposes. On the contrary, MonetDB has been successfully applied in different BI contexts [21]. Currently, the developers are working on improving the scalability for OLAP and data mining in the 64-bit versions of MonetDB.

MySQL. MySQL is a very popular open source database with more than five millions installations [24]. The latest production release is version 4.1, and version 5.0 is in the alpha stage. MySQL is implemented in C and C++ and is available for a large variety of 32- and 64-bits platforms. Users of MySQL can choose between an open source GPL license and a commercial license that gives permissions not given by the GPL license. For BI purposes, MySQL lacks support of materialized views (even ordinary views are not available until version 5.0), bitmap indices and star joins. However, one-way replication (i.e. one master, several slaves) is supported and partitioning is to some degree supported by the *NDB Cluster* (NDB is a name, not an acronym) on some of the supported platforms. Further, MySQL is capable of handling data sets with terabytes of data as documented in case studies available from [24].

MaxDB. MaxDB [18] version 7.5 is another RDBMS distributed by the company MySQL AB which also develops MySQL. Formerly, MaxDB was known as SAP DB (developed by SAP AG). MaxDB is developed to be used for OLTP and OLAP in demanding environments with thousands of simultanous users. It is implemented in C and C++ and is available for most major hardware plat-

forms and operating system environments. As MySQL, it is licensed under two licenses such that users can choose between an open source license (GPL) or a commercial. MaxDB is designed to scale to databases in the terabyte sizes, but there is no user controlled partitioning. It is, however, possible to specify several physical locations for storage of data, and MaxDB will then automatically divide table data between these partitions. There is no support for materialized views (ordinary views are supported), bitmap indexes, or star joins. MaxDB supports one-way replication, also with MySQL such that either of them can be the master.

PostgreSQL. PostgreSQL [32] is also a very popular open source DBMS. At the time of this writing, version 8.0 is just about to be released. PostgreSQL is implemented in C and has traditionally only been available for UNIX platforms. From version 8.0, Windows is, however, natively supported. Originally, PostgreSQL is based on the POSTGRES system [36] from Berkeley and has kept using a BSD license. PostgreSQL supports large data sets (installations larger than 32 terabytes exist) and one-way replication. A multiway solution for replication is planned. There is no support for partitioning, bitmap indices or materialized views (ordinary non-materialized views are supported). However, materializations of views may be done in PostgreSQL by using handcoded triggers and procedures [10]. Further, in a research project at North Carolina State University, materialized views are integrated into a derived version of PostgreSQL [34]. Bitmap indices are planned to be supported in a future release.

General Comments. The considered DBMSs have different strengths and weaknesses, and so there is not a single of them to be chosen as *the best*. In general, these open source products support more advanced features such as partitioning and replication. Further, the DBMSs are capable of handling very large data sets, are available for a number of platforms, and are very reliable. The category of DBMSs is thus the most *mature* of the considered categories.

8 Conclusion and Future Work

Of the considered categories of open source tools (ETL tools, OLAP clients, OLAP servers, and DBMSs), DBMSs are the most mature. They offer advanced features and are applicable to real-world situations where large data sets must be handled with good performance. The ETL tools are the least mature. They do still not offer nearly the same functionality as proprietary products. With respect to the OLAP servers, there is a great difference in their maturity. A product like Lemur is still very immature, while a product like Mondrian is usable in real-world settings. However, important features, such as the opportunity to specify which aggregates to use, are still missing. The OLAP clients are also usable in practical applications. However, they are not very general and can only be used with specific OLAP servers. In general, one of the largest problems for many of the tools is the lack of proper documentation, often making it very difficult to decide how a specific task is performed in a given product.

If one were to create a complete BI installation with open source tools, it would probably be created with JPivot and Mondrian as OLAP client and server, respectively. Which one of the DBMSs should be used would depend on the situation. The ETL tool would then probably be CloverETL, if one did not handcode a specialized tool for the installation. In many BI installations, data mining solutions would also be interesting to apply. The available open source data mining applications should therefore be explored in future work.

Acknowledgements

This work was supported by the European Internet Accessibility Observatory (EIAO) project, funded by the European Commission under Contract no. 004526.

References

1. Apache SW License v1.1, www.apache.org/licenses/LICENSE-1.1 as of 05/30/05.
2. Apache SW License, v2.0, www.apache.org/licenses/LICENSE-2.0 as of 05/30/05.
3. Bee, sourceforge.net/projects/bee/ as of 05/30/05.
4. Bee Project, bee.insightstrategy.cz/en/ as of 05/30/05.
5. BSD license, opensource.org/licenses/bsd-license.php as of 05/30/05.
6. CloverETL, cloveretl.berlios.de as of 05/30/05.
7. CPL v1.0, opensource.org/licenses/cpl1.0.php as of 05/30/05.
8. cplusql ETL tool, sourceforge.net/projects/cplusql/ as of 05/30/05.
9. Enhydra Octopus, octopus.objectweb.org/ as of 05/30/05.
10. J. Gardner. Materialized Views in PostgreSQL, jonathangardner.net/PostgreSQL/materialized_views/matviews.html as of 05/30/05.
11. GNU GPL License, www.gnu.org/copyleft/gpl.html as of 05/30/05.
12. GNU LGPL License, www.gnu.org/copyleft/lgpl.html as of 05/30/05.
13. gnuOLAP, gnuolap.sourceforge.net/ as of 05/30/05.
14. Java OLAP Interface (JOLAP), www.jcp.org/en/jsr/detail?id=69 as of 05/30/05.
15. JFreeChart, www.jfree.org/jfreechart as of 05/30/05.
16. Jpivot, jpivot.sourceforge.net/ as of 05/30/05.
17. Lemur OLAP library, www.nongnu.org/lemur/ as of 05/30/05.
18. MaxDB, www.mysql.com/products/maxdb/ as of 05/30/05.
19. Mondrian components, perforce.eigenbase.org:8080/open/mondrian/doc/index.html, as of 05/30/05.
20. Mondrian FAQs, perforce.eigenbase.org:8080/open/mondrian/doc/faq.html, as of 05/30/05.
21. MonetDB, monetdb.cwi.nl/ as of 05/30/05.
22. Mozilla Public License v1.1, www.mozilla.org/MPL/MPL-1.1.html as of 05/30/05.
23. MySQL Case Studies, www.mysql.com/it-resources/case-studies/ as of 05/30/05.
24. MySQL Database Server, www.mysql.com/products/mysql/ as of 05/30/05.
25. The OLAP Report Market share analysis, www.olapreport.com/market.htm as of 05/30/05.
26. OpenETL, openetl.tigris.org as of 05/30/05.
27. Open Source ETL (OpnSrcETL), opnsrcetl.sourceforge.net/ as of 05/30/05.

28. OpenROLAP, openrolap.sourceforge.net as of 05/30/05.
29. T. B. Pedersen. How Is BI Used in Industry?: Report from a Knowledge Exchange Network. In *DaWaK*, pp. 179–188, 2004.
30. T. B. Pedersen and C. S. Jensen. Multidimensional Database Technology. *IEEE Computer*, 34(12):40–46, 2001.
31. pocOLAP - the "little" OLAP-project, pocolap.sourceforge.net/ as of 05/30/05.
32. PostgreSQL, www.postgresql.org as of 05/30/05.
33. The R Project for Statistical Computing, www.r-project.org as of 05/30/05.
34. Self-Organizing Databases, research.csc.ncsu.edu/selftune/ as of 05/30/05.
35. G. Spofford and E. Thomsen. *MDX Solutions: With Microsoft SQL Server Analysis Services*, Wiley, 2001.
36. M. Stonebraker and L. Rowe. The Postgres papers, TR, UC Berkeley, 1987.
37. J. Vacca. *VRML: Bringing Virtual Reality to the Internet.* Academic Press, 1997.
38. Web Server Survey Achives, news.netcraft.com/archives/web_server_survey.html as of 05/30/05.
39. XML for analysis, xmla.org/download.asp?id=2 as of 05/30/05.