DATABASE USAGE AND PERFORMANCE FOR THE FERMILAB RUN II EXPERIMENTS

D. Bonham, D. Box, E. Gallas, Y. Guo, R. Jetton, S. Kovich, J. Kowalkowski, A. Kumar, D. Litvintsev, L. Lueking, N. Stanfield, J. Trumbo, M. Vittone-Wiersma, S. P. White, E. Wicklund, T. Yasuda, FNAL, Batavia, IL 60510, USA,

P. Maksimovic, Johns Hopkins University, Baltimore, MD 21218, USA

Abstract

The Run II experiments at Fermilab, CDF and D0, have extensive database needs covering many areas of their online and offline operations. Delivering data to users and processing farms worldwide has represented major challenges to both experiments. The range of applications employing databases includes, calibration (conditions), trigger information, run configuration, run quality, luminosity, data management, and others. Oracle is the primary database product being used for these applications at Fermilab and some of its advanced features have been employed, such as table partitioning and replication. There is also experience with open source database products such as MySQL for secondary databases used, for example, in monitoring. Tools employed for monitoring the operation and diagnosing problems are also described.

INTRODUCTION

Run II has been underway since early 2001 and CDF and D0 have made extensive use of their database resources. Oracle was chosen as the product of choice for all mission critical database applications because it is a robust, well supported product, with a feature set that matches the Run II needs. The databases satisfy a broad range of requirements ranging from production code tracking to the physics data file catalogue. The information in the database is used extensively in the online systems for data-taking, as well as offline by users and processing farms located around the globe.

REQUIREMENTS

Both CDF and D0 maintain Oracle database servers at the experimental halls, and these instances are critical for data taking. Offline databases are maintained in the Feynman Computing Center and are essential for data processing and analysis. High availability for both the online and offline systems is required. Generally, the database applications fall into two categories, 1) detector and physics related information, and 2) data handling information. In the first broad area are detector calibration, trigger configuration, data luminosity, detector slow controls, Run configuration and Run quality information. Data Handling includes physics metadata, file catalogues, file replica management, and processing information such as versions of the programs and the processing chains used to create each file or data set. Figure 1 shows the growth of the D0 Offline database in several of these areas, and Figures 2 and 3 chart the growth of CDF's online and offline databases respectively.

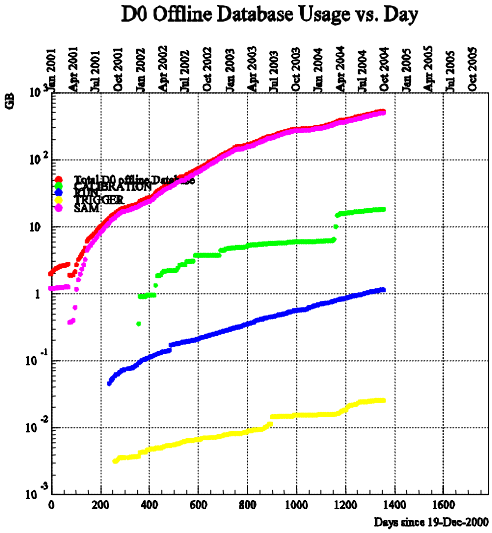

Figure 1 D0 Offline Database Size (GB, log scale) growth versus day of the Run.

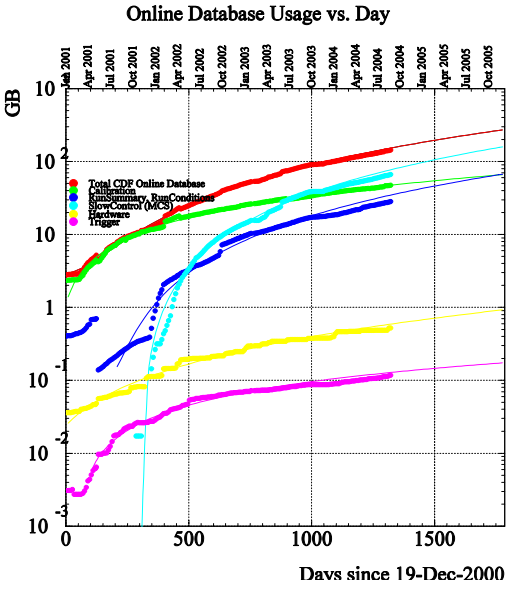

Figure 2 CDF Online Database size (GB, log scale) growth versus day of the Run.

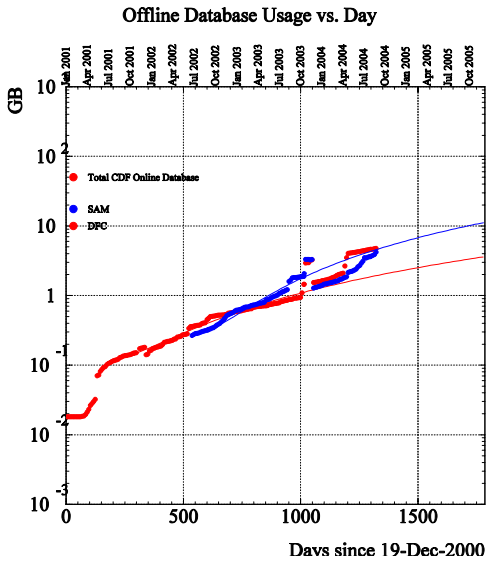

Figure 3 CDF Offline Database Size (GB, log scale) growth.

ORACLE USE AND ADMISTRATION

The pervasive use of Oracle in the Run II software represented a major advance over previous Tevatron running periods. Both CDF and D0 used Oracle exclusively for data consistency, integrity, and high availability access. Many of the advanced features of the product were also needed due the very large size and high demand for the servers. In the following sections we describe the use of Oracle table partitioning, replication, backup and recovery, and monitoring tools.

Oracle Table Partitioning

Partitioning has been implemented for very large table(s) in the database. The D0 Events table, which records trigger information for each event, has been partitioned to accommodate its large size. This table is run-range partitioned with each partition containing 50M events (rows). Each partition is stored in its own tablespace and corresponding indexes are also partitioned and also stored in their own tablespaces.

Benefits of partitioning are twofold, first it improves query optimization, and second it increases backup performance. The query improvement is achieved because the optimizer, knowing that the table has been partitioned, directs the server to only access data from that partition. The improvement in backup performance is achieved because once a partition is "rolled over", its data and index tablespaces are converted to read-only. Read-only tablespaces are only backed up twice a month, and are excluded from read-write tablespaces backup, thus reducing the time for daily database and tape backups. Currently, over 1 billion events are distributed over 24 partitions in the database, and a new partition is added roughly once every month. The growth of partitions in the D0 Event table is shown in Figure 4.

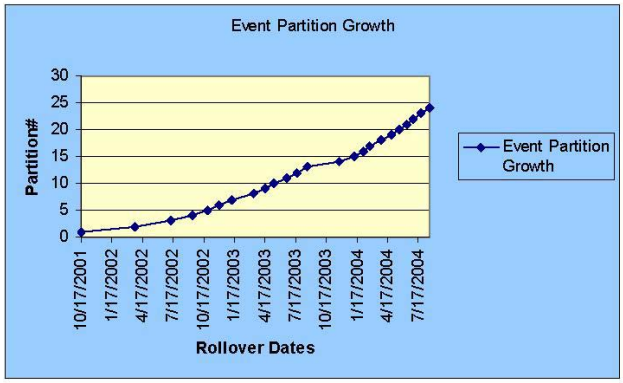

Figure 4 Partition growth for D0 Event table.

Oracle Replication

Throughout the course of Run II, CDF successfully employed Oracle replication to provide offline replicas of its online database. A key feature of the CDF replication is fail-over from one replica to another for high reliability. Replication also provided a mechanism to scale the database server resources to meet the ever growing needs of client processing. This will be discussed further in a later section. During the course of Run II, several of Oracle's replication technologies were tested for performance and stability including, 1) basic replication, 2) multi-master replication, and 3) advanced streams replication.

CDF is currently using Oracle's asynchronous replication where the off-line tables are refreshed from on-line tables at predetermined intervals. Data changes are replicated from on-line database to offline databases every 15 minutes. CDF On-line database is replicated to two locations as shown in Figure 5.

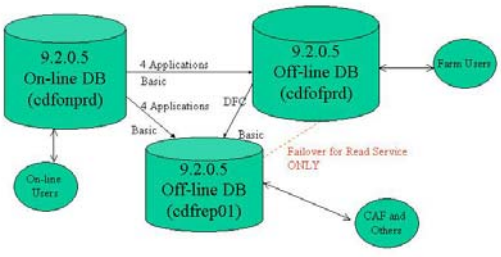

Figure 5 CDF Oracle Replication configuration.

One of the replicas is used for Reconstruction Farm Users and other by CDF Analysis Facility (CAF) and other read-only users. Refresh time for each replica is offset by 7-8 minutes to make best use of caching at the source side. This asynchronous replication is implemented using snapshots and is in uni-directional mode.

Oracle Monitoring

The Run II Databases are closely monitored using Oracle Enterprise Manager (OEM) as well as by in-house developed tools called "TOOLMAN". OEM is a tool provided by Oracle Corp. to monitor the oracle databases. OEM provides a graphical interface enabling the DBA to administer the database, typically adding space, creating users, assigning roles, et cetera. OEM can also be used for event monitoring, which we use in conjunction with TOOLMAN. Events such as space nearing capacity, database down, node down, and database alerts will result in an email being sent to the DBA team for resolution.

The TOOLMAN utility was designed to provide an alternative Oracle monitoring method that does not share a common point of failure with the Oracle Enterprise Manager. TOOLMAN is implemented as shell and SQL scripts and has no GUI interface. It is hoped that this will provide a higher level of database monitoring, at the expense of redundant alerts. It also has login trace feature which tracks who, on which node, the program being run, when it was run, and the CPU used. It has a sniping feature which terminates sessions that have been inactive more than the specified time. Rules for such actions are easily customized.

TOOLMAN also provides ongoing routine maintenance for a database by executing a series of cron jobs that automatically perform basic Oracle maintenance activities. Further, TOOLMAN provides a source of historical data and monitoring should a machine be isolated from the network and otherwise unavailable to OEM. It is primarily used in conjunction with cron. TOOLMAN can be customized in several ways for the machine and databases it monitors.

Oracle Backup and Recovery

The databases are backed up using Oracle's RMAN tool. Our production databases are typically run in archive mode, and we store the RMAN files, as well as the archive logs, to tape during the backup procedure. These backups are made nightly for all active transaction tables, and twice per month for read-only tables. At the present time, as an example, the D0 production database has 375 GB in active transaction mode, and 200 GB in read-only mode. They require about 3 hours for the first, and 45 minutes for the later, to make the RMAN backups to disk. Legato Backup Software is used with an Exabyte tape library to store the disk backups to tape in 3 and 2 hours respectively.

Instance recovery is practiced several times each year in order to verify and update the procedures. As a result of this practice, we have identified several items that are useful when preparing for a recovery, such as knowing database file pathnames and sizes. A tool has been built to collect this data as a step in the nightly backups. During Run II, there have been two occasions when instance recovery was needed for production databases that we manage. In both cases successful recoveries were performed with no loss of data.

We are experimenting with the use of a Storage Area Network (SAN) for centralized storage of backups. We believe that this will reduce recovery time by eliminating the need to read RMAN files from tape in most recovery situations. The SAN will allow us keep more than one day of RMAN files on "local" disk and will allow us to start a recovery immediately when a loss occurs.

DATABASE ACCESS APPROACHES

CDF employed Oracle replication to copy the online instances of the database to the offline, as discussed earlier and shown in Figure 5. They are planning to move to Oracle Advanced replication to provide additional functionality in the near future. They are also pursuing a web caching technique through the Frontier project [1] that will greatly improve the performance of their read-only database access, especially to clients located geographically distant to Fermilab.

The D0 approach was to accumulate online detector and run information in schemas specific to the online operation. This information was then transferred through a queuing system, reformatted, and loaded into the offline database. All access to the offline database was through a middle tier server. In the case of the Data Handling system, the SAM server incorporates business logic methods to access and manipulate the information in the database. For static read-only data, such as detector calibration, D0 employs caching servers called Database Access Network (DAN)[2]. These servers provide connection management to the database, and object caching which significantly reduces database access.

PERFORMANCE MONITORING

Performance monitoring is an important part of the database operation, alerting us to problems, and helping to identify trends, which enable planning for future needs. In the summer of 2002, CDF began building a system that would collect UDP messages sent from clients accessing their database. In the fall of 2002, it was recognized that many of the tools being developed for CDF could be shared with D0, and other experiments, and a general framework for monitoring, archiving, and presentation was established. A project called DBSMON [3] was defined, and tools were built to perform the data collection and automate the graphical presentation of the information. Significant effort has been devoted to understanding the kinds of information to be monitored, and collecting a useful historical summary of the database operation.

The main idea behind the project was to provide a way to monitor database usage, highlighting problems, stability, and trends. The main effort was to find commonality between the experiments, D0 and CDF, in order to produce a general utility that could also be extended to future experiments. The focus was to identify useful information such as top users, top applications, top tables accessed, duration information for both queries and connections, and the number of connections. It was

decided to use MySQL to store the monitoring information, based on the simplicity of the schema and ease of administration. Java was chosen as the language for preparing the plots for presentation based on its portability and straightforward interface to MySQL. The choice of plotting tools, JFreeChart [4], was guided by the nature of data to be represented and the need for a user-friendly display.

CDF and D0 each have their own MySQL servers and databases on separate machines. The monitoring is provided through several jobs that run on a daily basis to produce plots accessible via the WEB, and archived for historical usage. In addition to daily information summary plots are provided which cover longer periods of time. The complete set of plots is available from the web site [3]; a few examples are presented in the following sections for illustration.

CDF Performance

Information on CDF performance comes either from Oracle tables or directly from the client, and is tracked at the individual client level. Since clients connect to the CDF central database facility, monitoring the number of connections is important, and this is shown in Figure 6 for the last 18 weeks. Figure 7 shows the top users of CPU on the Oracle database server.

Figure 6 CDF Number of connections per minute. The lower line is Reconstruction Farm connections, and the upper line is all other offline client connections.

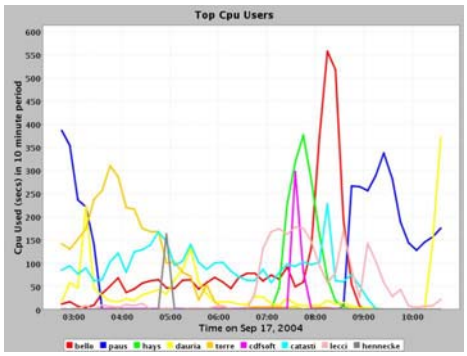

Figure 7 Top CPU users on CDF Database Applications over an 8 hour interval.

D0 Performance

D0 performance is monitored at the DB server level. Figures 8 shows the number of Oracle queries made by the D0 calibration DAN servers. The server's memory and disk caching features enable more efficient response to client requests and reduce the number of database queries. Figure 9 shows the query counts for the D0 SAM database servers, some of which make over 10^4 queries per hour to the D0 database.

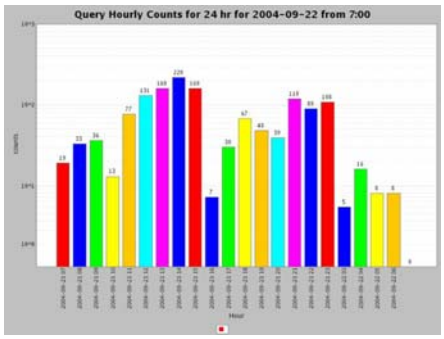

Figure 8 Number of queries per hour for D0 Farm and Non-Farm calibration DAN servers.

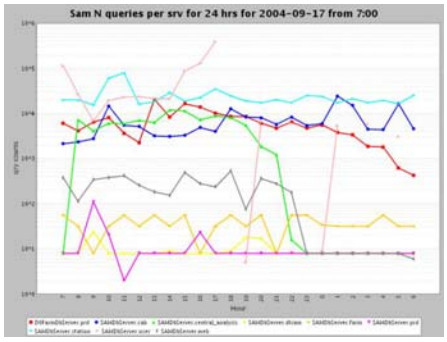

Figure 9 D0 Sam Servers query counts over a 24 hour interval.

ACKNOWLEDGMENTS

We would like to thank the Fermilab Computing Division staff who have contributed to the success of the Run II database operation, including the Database Support Group (DSG) and Database Applications Group (DBS). We thank the CDF and D0 experiments for their cooperation in the development of the tools and methods described here.

REFERENCES

[1] S. Kosyakov, et. al., "Frontier: High Performance Database Access Using Standard Web Components," CHEP04, Interlaken Switzerland, Sept. 27-Oct. 1, 2004.

[2] J. Kowalkowski, et. al., "Serving Database Information Using a Flexible Server in a Three Tier Architecture," CHEP03, UCSD, La Jolla CA, March 24-28, 2003, THKT003.

[3] The DBSMon page http://dbsmon.fnal.gov .

[4] JFreeChart home page http://www.jfree.org