CIMS LAB, INC.

CIMS Server Reporting Performance White Paper

CIMS Server includes a robust Web reporting component that performs extremely well when the system is tuned properly. Some CIMS customers have experienced variability in performance that is caused by a number of factors. Surprisingly, the size of your organization, your operating system platform, the DBMS, the Report Writer, and even the volume of data does not necessarily determine the performance of the CIMS Server Web Reporting system.

The purpose of this white paper is to discuss the causes of reporting performance issues and various methods for resolving these issues. The key factors determining performance include the following in order of importance:

1. Customer expectations/requirements
2. Level of identifier summarization in Detail data
3. Hardware
4. Level of date summarization
5. Level of account code in Summary data
6. Individual report, stored procedure, and index tuning
7. Database maintenance
8. Number of months of data retained in the database

Customer Expectations/Requirements

Because CIMS Server is a Web-based reporting application, users often perceive that all reports should be created in time frames as short as 2 to 5 seconds. Unlike e-commerce or news sites found on the Internet (which typically have a very fast response time), CIMS Server is trying to process and aggregate very large volumes of data and then return the data to the client's browser. The end user running reports must understand what is being requested and have realistic expectations of the performance.

A careful examination of reporting requirements is sometimes required. A Detail report that runs in 15 minutes might be processing one month of Detail data that could contain 2 million rows. Through extensive additional indexing, performance tuning, aggregation, and hardware upgrades, the report could complete in 10 minutes. However, this is not a significant improvement and many users might still consider this unacceptable.

In this case, CIMS Server includes two features that enable users to quickly view reports: Published Reports and Batch E-Mail Report Distribution. These features provide the best avenue for displaying a report in the sub-second range that many users might expect.

Recommendations

- Set expectations for performance. CIMS Lab can assist you in this process after all other avenues to increase performance have been exhausted.
- If the performance is still unacceptable after a report and its corresponding stored procedure have been tuned, consider using the Batch E-mail Report Distribution feature. This tool, introduced in 4.0, enables administrators to generate one or more reports during off hours and then distribute the reports to users via e-mail or distribute links to the reports via e-mail. There are many advantages to this methodology. The main advantages are the off-loading of processing to off hours and the ability to have reports delivered directly to a user's mailbox. A secondary advantage is the potential tightened security, because each end user would only receive his or her reports and could be restricted from using the Web site entirely.

  For more information about the Batch E-mail Report Distribution feature, refer to the *CIMS Server Administrator's Guide*.
- The Published Reports feature enables users or administrators to create a report and then save it for future viewing by the report creator or other users in the organization. This powerful feature eliminates the time consumed by running the same report with the same parameters multiple times.

  For more information about the Published Reports feature, refer to the *CIMS Server Web Reporting User's Guide*.

Level of Identifier Summarization on Detail Data

The number of identifiers used, more than any other factor, can determine how well an individual report will perform.

An identifier denotes the source of a resource that has been consumed. For example, a server name, user name, jobname, process name, etc. CIMS Server aggregates data by identifier values.

Although the input file records can contain many possible identifiers, you can define the specific identifiers that you want to use for aggregation. These are the identifiers that will appear in the output CIMS Server Resource (CSR) records and the Detail records that are stored in the database. In general, the more identifiers that are defined for aggregation, the greater the number of records that are stored in the database. If aggregation is performed on millions of records, the number of identifiers used for aggregation will dramatically increase or decrease the number of records stored in the database. An increased number of records can increase the time required to generate reports.

For example, the following raw input records contain seven identifiers (Start Date through Process Name) and two resources (CPU Time and Reads):

Start Date	Start Time	End Date	End Time	Server Name	User Name	Process Name	CPU Time	Reads
1/1/05	12:01:01	1/1/05	12:01:04	Server1	User1	Process1	.25	12
1/1/05	12:01:02	1/1/05	12:01:05	Server1	User1	Process2	.5	22
1/1/05	12:01:03	1/1/05	12:01:06	Server1	User2	Process3	.5	10

In most cases, time identifiers are not used for aggregation, so that leaves the Start/End Date, Server Name, User Name, and Process Name identifiers.

If aggregation is performed on the Start/End Date, Server Name, User Name, and Process Name identifiers, there would be no record aggregation and three Detail records would be stored in the database.

If aggregation is performed on Start/End Date, Server Name, and User Name identifiers, the first two records would be aggregated and two Detail records would be stored in the database. The aggregated record would contain .75 CPU seconds and 34 reads. The second record would contain .5 CPU seconds and 10 reads.

If aggregation is performed on the Start/End Date and Server Name identifiers only, all three records would be aggregated into one Detail record in the database. The aggregated record would contain 1.25 CPU seconds and 44 reads.

Recommendations

Evaluate which identifiers are needed for reporting and use those identifiers for aggregation. Be aware, though, that if you do not use an identifier for aggregation, it will not be included in the output record and cannot be added back to the record at a later time unless the raw data is retained.

If you decide at a later date that one or more identifiers are not necessary for reporting, the identifiers can be removed from the database and the loads using CIMS scripts.

To reduce the number of identifiers that appear in Detail data:

- Mainframe – make changes to the number of identifiers being retained using the CIMSEXTR program.
- Unix – make changes to the Collector scripts.
- Windows – make changes to the Collector scripts.
- To remove extraneous identifiers out of the database or the flat file feeds, CIMS provides the following scripts in the ..\CIMSLab\Scripts folder: DropIdentFromDB.wsf, DropIdentBeforeLoad.wsf, ListUnusedIdentifiers.wsf, and CSRAggregator.wsf. For more information about these scripts, refer to the *CIMS Server Administrator's Guide*.

Hardware

CIMS Server's minimum requirements of a 1 GHz processor with 1 GB of memory on a single server will work, but might not perform at an acceptable level. Processor speed and memory are the two biggest components in determining the CIMS Server application performance. SQL Server will consume as much of the system memory as possible. Therefore, it can be very advantageous to have a separate machine for the SQL Server database. This allows the main CIMS Server machine to allocate its resources to Crystal Reports and IIS while the second machine can be dedicated to SQL Server.

Recommendations

- Add an additional machine that contains SQL Server.
- Add additional memory to the machine hosting SQL Server.
- Add a larger processor or an additional processor to the machine hosting Crystal Reports.

Level of Date Summarization

Like identifier summarization, date summarization is highly configurable in CIMS Server. However, in most cases, date summarization is much less important in reducing data size.

Most installations prefer to report usage on a daily basis. The exceptions are the "snapshot" type collectors like Disk/Tape Usage (CIMSWinDisk, mainframe CIMSDISK and CIMSTAPE, and Unix Filesystem) and Database Size (DBSpace, Oracle Tablespace, etc.).

Recommendations

- Assess how frequently snapshot data should be collected. If the collection is for a small amount of Windows folders or database tablespaces, the collection probably will not contain a lot of daily data. If the collection is for every data set and tape in a mainframe shop on a daily basis, there may be millions of records collected per day. Sometimes this data can be averaged or collected on a weekly or monthly basis instead of daily.
- For Summary reports, CIMS provides a script (..\CIMSLab\Scripts\ConvertSummaryDailyToMonthly.wsf) that aggregates a month's worth of daily data for a feed in the CIMSSummary table to a single record of monthly data. This will typically reduce Summary record volume for a single month by a factor of about 21:1, depending on data distribution. For more information about this script, refer to the *CIMS Server Administrator's Guide*.

Level of Account Code on Summary

Most CIMS reports are run at the Summary level of the data. Like other aggregation points in CIMS, CIMSBILL automatically aggregates this data using the Account Code, Start Date, Shift, and Rate Code.

"Raw" identifiers from the CSR record are not included automatically in the aggregated Summary records. These records contain an account code derived from the combination of identifiers and the account code conversion table.

The CIMS Server administrator has complete control over the account code. Usually, the account code is built using business type values like company, business unit, application, agency, etc. However, it is also possible to include identifiers like jobname, server, or even a date in the account code. Adding identifier values like jobname, server name, or date will usually dramatically increase the number of Summary records and consequently slow down Summary report performance.

Recommendations

Closely evaluate the account code for extraneous fields. A longer, more detailed account code usually means more records in the CIMSSummary table. The only reason to include identifier information in the Summary account code is if the resource usage needs to be costed at that level. Be aware that any identifier that is in the CSR record is automatically included in the CIMSDetail table. Therefore, the identifier is always accessible in reports that use the CIMSDetail table.

Individual Report, Stored Procedure, and Index Tuning

If most reports appear to be performing acceptably, but one report is not, the problem could lie in a few areas. CIMS Standard reports are fully optimized to perform acceptably. However, because of data distribution issues, a report may perform worse at one installation versus another.

Recommendations

- If it is a Summary-based report (costs are included), be sure that other Summary reports appear to perform acceptably. If all Summary reports are performing poorly, refer to one of the other sections in this paper for the possible cause.

- If it is a Detail-based report (costs are not included), be sure that other Detail reports appear to perform acceptably. If all Detail reports are performing poorly, refer to one of the other sections in this paper for the possible cause.

- If it is only one report that is performing poorly, run the report with different date ranges. Does this make a perceptible difference in performance? If a smaller date range greatly increases the performance, it is likely too much data is being selected and the performance is what is to be expected.

- Is it a custom report/stored procedure? If so, please contact CIMS Lab for assistance in optimizing the report/stored procedure. It is possible that certain indexes are being used improperly or that the stored procedure could be tuned to work more optimally.

- Is the data being selected a large part of the underlying Summary or Detail table? Mainframe CICS will often be a very large part of the Detail data. A Rate Group report running against CICS data might take much longer than a Rate Group report running against Unix Process data because there is a much larger set of data being returned to Crystal Reports. Test the report with different parameters to better assess its performance bottleneck.

- Is a main report performing acceptably, but the drill down is very slow? This is likely to be a problem with an index. Be sure to go to the Database Objects Maintenance dialog box in the CIMS Server Administrator application and verify that all objects are built and exist in the database (`Yes` appears in the **Exists** column for the object). If an object does not exist, click the check box for the object and then click **Create Selected Objects**.

  To open the Database Objects Maintenance dialog box, click **System Administration ▸ Database ▸ Database Objects Manager** in CIMS Server Administrator.

Database Maintenance

DBMS products like MS SQL Server, Oracle, and DB2 can work quite well for a time without significant database maintenance being performed. However, over time, there will be a performance degradation if standard database maintenance procedures are not performed regularly.

The following are tasks that need to be performed in the MS SQL Server database. Ideally, your organization will have a DBA function that can perform these tasks.

- Regular backup
- Index rebuilding
- Check database (DBCC CHECKDB)
- Check the allocation of data and index pages for each table (DBCC NEWALLOC)
- Check for consistency in and between system tables (DBCC CHECKCATALOG)
- Rebuild indexes
- Update statistics on each table
- Database shrink (DBCC SHRINKDATABASE)

Recommendations

Ideally, the SQL Server DBA will perform the preceding tasks on a regular (at least weekly) basis. Many shops, however, do not have an available SQL Server DBA function. For this reason, CIMS Lab provides a script for MS SQL Server (..\CIMSLab\Scripts\SampleSQLMaint.bat) that performs all of the these tasks. For more information about this script, refer to the *CIMS Server Administrator's Guide*.

The same functions that are in the SampleSQLMaint.bat script are also available from MS SQL Server Enterprise Manager and from third-party SQL Server performance tuning products.

Executing these maintenance tasks on a regular weekly basis will usually have your database running at an optimum performance level.

Number of Months of Data Retained in the Database

Removing older Summary and Detail data from the database is useful, but is usually not as important for improving report performance. Most CIMS Server reports are run with date selection criteria that will restrict the data reported using the End Date. This selection is done early in the query process using indexes, so the performance gain from removing older data from the CIMSSummary and CIMSDetail tables is not as great as other measures detailed in this paper.

The larger benefit of removing older data is a reduction in hard disk space consumption. This is still a critical requirement at many IT shops, so even without the significant performance gains, older data should be removed.

Recommendations

CIMS provides both a Windows GUI method for removing older data and a batch method using either CIMS Job Runner or scripts.

In CIMS Server Administrator, you can do either of the following to remove data from the database:

- Permanently delete Summary/Detail data using the Load Tracking dialog box (click **Chargeback Administration ▸ Database Loading ▸ Load Tracking**).
- Delete and archive Summary/Detail data using the Load Archive dialog box (click **Chargeback Administration ▸ Database Loading ▸ Load Archiving & Purging).**

For batch removal, CIMS includes the ..\CIMSLab\Scripts\PurgeLoads.wsf script that can be launched from scripts or from CIMS Job Runner. This script removes data based on a date range or a date keyword. For more information about this script, refer to the *CIMS Server Administrator's Guide*.

Conclusion

Following the measures listed in this paper will usually result in a CIMS Server implementation that performs at an acceptable level. Years of tuning the CIMS Server application, coupled with user education, has yielded a product that can grow over time and still maintain an acceptable performance level.

Please contact CIMS Lab at support@cimslab.com or (800) 283-4267 for assistance with any of the measures documented in this paper. These steps are not always easily implemented without assistance from CIMS Lab.