

Business Planning and Consolidation Performance Tuning for BPC 5.x



Applies to:

SAP Business Planning and Consolidation version 5.0 and 5.1.

Summary

SAP Business Planning and Consolidation 5 is based on a Microsoft platform. As such there are many opportunities to tune the performance. The guidance provided in this document is based on our experiences with SQL 2000 and recently SQL 2005. In addition, BPC itself can be tuned during the application design and subsequent to design.

All tuning should be tested in a development environment.

Each part of the document should be reviewed but not all will be applicable for a given implementation.

Author: Laura Di Tomasso

Company: SAP Labs, LLC

Modified on: 19 May 2008

Author Bio

Laura Di Tomasso, a solution manager with the Corporate Performance Management team at SAP. Laura has extensive knowledge of the BPC solution as she was with OutlookSoft Corporation (acquired by SAP May, 2007) since the version 1.0 release.

Table of Contents

Architecture	4
Server Components	4
Web server	4
Application server	5
OLAP Server	5
Microsoft SQL Server	5
File share	5
Microsoft SQL Reporting Services	5
Basic install configurations	5
Single Server	5
Multi-server	6
Client	6
Development environment	7
Disaster Recovery	7
High Availability	7
Tuning Introduction	10
The Tuning Layers	10
Tuning - Minimum Consideration	10
Hardware Configuration	10
How Many Servers	10
What's in the Server?	11
Don't forget the clients	11
Operating System Configuration	11
Memory Configuration	11
Disk Configuration	11
Virus Scanning Software	11
Setting Connection Timeouts (IIS 6.0)	12
AppSet Database Configuration	12
Partitioning for very large applications	12
When and Why	12
How	12
Cube Processing by Partition	12
SQL Fact Table Indexes	13
Customizing the Indexes	13
Dependencies	13
Partition Aggregation Strategies	14
System Software Configuration	15
SQL Server Configuration	15
Analysis Services Configuration for SQL 2000 only	15
Analysis Services Configuration for SQL 2005 only	16
Operating System Services	16
BPC Parameters	17

SAP BPC Send Governor Configuration Modifications	17
Background.....	17
Configuring the Send Governor	17
Application Customizations.....	17
Consolidations Package Changes	17
Intercompany Package	17
Currency Translation Package.....	18
Data Manager Package Changes.....	18
Application Design	18
Dimensions	18
Alternate Hierarchies	18
Number of Dimensions	19
Reports and Input Schedules.....	19
General Guidelines	19
“Asymmetric Queries”	19
EVDRE	19
Logic type (MDX, SQL, or SQL Server Stored Procedures).....	20
Application Maintenance.....	21
Compression	21
Lite Optimization	21
Storing Historical Data	22
Keep the Wizards directory small	23
A: Identifying Performance Bottlenecks	24
Monitoring Tools	24
Memory.....	24
Memory \ Pages/sec	24
Memory \ Available bytes.....	24
Disk I/O	24
Analysis Services: Agg Cache \ Evictions / Sec.....	24
Physical Disk \ Avg. Disk Bytes/Read	25
Processor.....	25
Processor \ % Processor Time.....	25
Analysis Services.....	25
Analysis Server: Connections \ Current Connections in Progress	25
SQL Server	25
SQL Server: Buffer Manager \ Buffer Cache Hit Ratio	25
Appendix B: Cache Evictions and Locking	25
Cache Evictions and Cache Thrash.....	25
Appendix C: FACT, FAC2 and WB tables	26
Appendix D: Other Resources and References.....	27
Copyright.....	28

Architecture

SAP Business Planning and Consolidation (BPC) for SQL Server at a high level is three tiers:

- The data base tier – SQL Server and Analysis Services
- The application tier – BPC services
- The web tier



Some key advantages of BPC architecture:

- Web-based platform with centralized database.
- Designed for scalability to thousands.
- Zero-footprint, Web-based application leveraging cutting-edge technologies
- Next generation Service-Oriented Architecture (SOA) providing easy integration.
- Multi-platform capabilities.
- Patented, enterprise-scale Microsoft Office integration.

Server Components

To help you understand the configurations let's review the technology components used by SAP BPC. For further details on the hardware requirements and underlying software requirements please refer to the service market place.

SAP Business Planning and Consolidation – Master & Installation Guide

<https://websmp110.sap-ag.de/instguidescpm-bpc>

Web server

The Web server provides launch pad for all BPC functional areas including the Web interface to zero footprint BPC functionality.

- Microsoft IIS
- Uses SOAP (Service Oriented Application Protocol) calls for communication using XML between web and application server.
- Host for ZFP (Zero footprint) Pages
- Performs user authentication

Application server

- Microsoft IIS
- Supports all SOA (Service Oriented Architecture)
- Collection of BPC Services
- Where BPC Application Business Logic Resides
- Performs user authentication

OLAP Server

- Analysis Services Microsoft SQL Server – Enterprise Edition 2005 (2000 still supported but not standard)
- High performance multi-dimensional database component
- Provides detailed analysis & drill down
- Can be 64 bit when separate from Application/web services

Microsoft SQL Server

- Enterprise Edition 2005 (2000 still supported but not standard)
- Supports data management activities
- 100% Open Standard
- Can be 64 bit when separate from Application/web services

File share

- Storage location for all application files, UNC or local drive.
- Book Repository
- Templates
- Unstructured Data
- Conversion & Transformation Files

Microsoft SQL Reporting Services

- Relational report generator
- Supports Journals, Work Process Flow, Comments and Audit reporting capabilities
- Provides Drill Through reporting to source details
- Can be 64 bit when separate from Application/web services

Factors impacting installation configuration is based on licensing. All servers with SQL Server components must have SQL licenses. All servers where IIS is located

Require license. IIS on the Analysis server only required if other servers are separated through a firewall and required ports cannot be opened on the firewall between servers.

Basic install configurations

Single Server

- When the number of concurrent users is low
- If you are setting up a development environment
- If there are no requirements for separating the MS IIS component from the SQL Database component
- If the server is in a Workgroup not a Domain
- No requirements for High Availability (clustering)

Single Server basic guidelines:

	Development Environment		Production Environment	
	Minimum	Recommended	Minimum	Recommended

CPU's	2	4	4	4 Dual Core
RAM GB	2 GB	4 GB	4 GB	8 GB
Disk GB	36 GB	72 GB	72 GB	144 GB
RAID Level	RAID 5	OS–RAID 1 Data–RAID 1+0	RAID 5	OS–RAID 1 Data–RAID 1+0
OS	Windows 2003 Standard and Enterprise Edition SP2/R2 (32 bit only)		Windows 2003 Enterprise Edition SP2/R2 (32 bit only)	
Contact an BPC Certified Technical Consultant for more information				

Multi-server

- When the number of concurrent users is high
- Corporate policies dictate separating certain components – (e.g. IIS and SQL must not share the same server)
- Need to separate functional components according to corporate departmental structure
- Web Services must be hosted in a DMZ
- Database Services must be hosted internally

Multi-server basic guidelines with tier centric configuration:

	Minimum			Recommended		
	Web Tier	App Tier	DB Tier Can be 64 bit	Web Tier	App Tier	DB Tier Can be 64 bit
CPU's	2	2	2 Dual Core	2 Dual Core	2 Dual Core	4 Dual Core
RAM GB	2	2	2	2	4	8
Disk GB	36	36	72	36	72	144
RAID Level	1	1	5	1	1	OS – 1 Data -1+0
OS	Windows 2003 Standard and Enterprise Edition SP2/R2			Windows 2003 Enterprise Edition SP2/R2		

Client

It is important to also consider the end-user (client) hardware. Basic guidelines:

	Standard User	Power User	Admin User
CPU	x86 (600 MHz >)	x86 (1.4 GHz >)	x86 (1.6 GHz >)
RAM	1 GB >	1 GB >	1 GB >

Hard Drive	150 MB	500 MB	1 GB
Other	NIC 100 MBit SVGA 800 x 600		

Client software recommended:

- Window XP SP1 (or later) or Windows 2000 SP4 (or later)
- Internet Explorer 6.0 SP1 (or later)
- Microsoft Office 2003/2007
- 2000 SR1 (or later)
- Office XP SP2 (or later)
- Office 2003 SP3 (or later)
- Microsoft XML 3.0 SP1 (or later)
- Microsoft .NET Framework 1.1
- PDF Reader (optional)

Development environment

A development environment is an integral part of a successful implementation. It provides an environment for:

- Testing third party software updates: Windows, SQL, Analysis Services, Reporting Services
- Can be used as a staging area for new BPC releases that require conversions
- Testing BPC application set design changes
- Environment to develop future BPC applications/ application sets
- Provides Fail Over and/or Disaster Recovery hardware
- Reduces potential impact on production users resulting from any of the above activities

Production and Development environments do not have to be identical in hardware or configuration

Disaster Recovery

Disaster Recovery is the ability to restore the system back to working order in a timely manner.

- Includes the recovery of data, hardware, and software
- Includes a DR plan with a time specified for maximum acceptable recovery time

Use Development Environment as a DR Environment

Your decision is then based on sizing – an exact replica or less performance and access for the time it takes to recover.

Use Backups

You need to see how often you need to backup based on the time frame you need to meet.

High Availability

High availability is the automatic failover of the system.

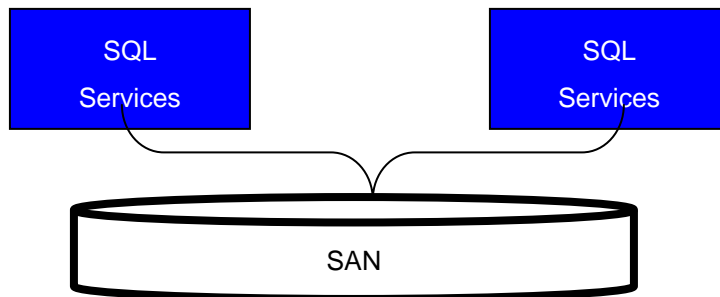
- Includes identical replicas of complete database and application servers with continuous real-time failover
- Should a server fail or have to be shut down for maintenance a synchronized replica will instantaneously and automatically take its place to ensure continuous application availability and business continuity

Complete Redundancy for each component in the BPC 5 Technology Stack

- Microsoft SQL Server 2005
- Microsoft Analysis Services 2005
- Microsoft Reporting Services 2005
- BPC Application Services
- BPC File Share
- BPC Web Services

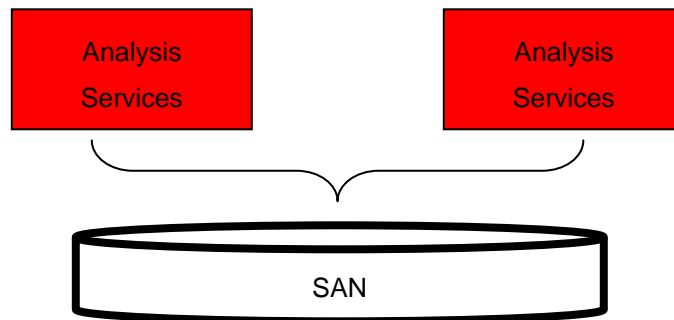
Microsoft SQL Server 2005 Clustering

- SQL Server 2005 Failover Clustering. Can be 64 bit when separate from web/application tier
- Minimum of 2 node cluster, maximum number of nodes dependent on operating system (Windows 2003 STD vs. ENT)
- Microsoft Recommends Clustering for data protection and high availability



Microsoft Analysis Services 2005 Clustering

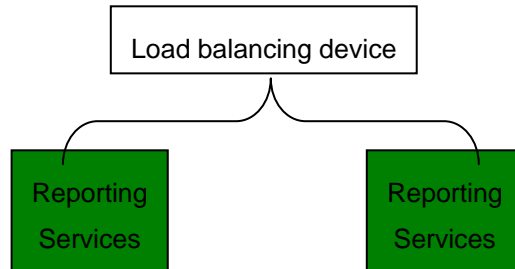
- SQL Server 2005 Failover Clustering. Can be 64 bit when separate from web/application tier
- Minimum of 2 node cluster, maximum number of nodes dependent on operating system (Windows 2003 STD vs. ENT)
- Microsoft Recommends Clustering for data protection and high availability



Microsoft Reporting Services 2005 Load balancing – all nodes active

Support many third party load balancing solutions, for example:

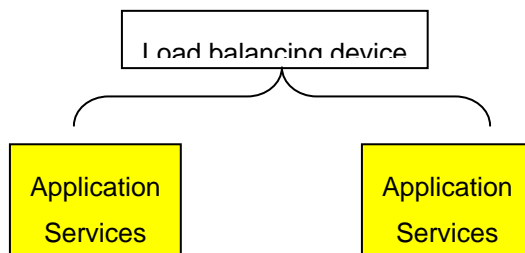
- Microsoft Network Load Balancing
- F5 Big IP
- Cisco



BPC Application Services Load balancing – all nodes active

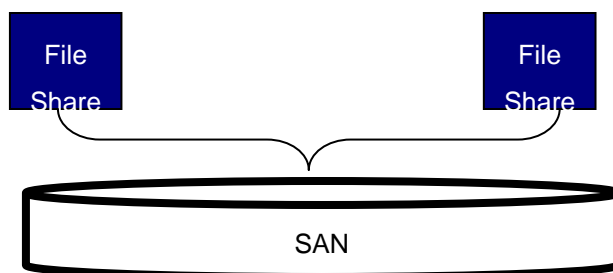
Support many third party load balancing solutions, for example:

- Microsoft Network Load Balancing
- F5 Big IP
- Cisco



BPC File Share using Microsoft File Clustering

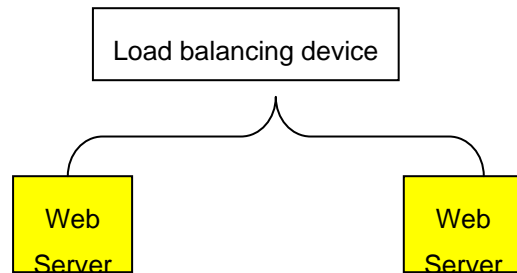
- Active / Passive Failover Clustering
- Minimum of 2 node cluster, maximum number of nodes dependent on operating system (Windows 2003 STD vs. ENT)
- Microsoft Recommends Clustering for data protection



BPC Web Services Load balancing – all nodes active

Support many third party load balancing solutions, for example:

- Microsoft Network Load Balancing
- F5 Big IP
- Cisco



Tuning Introduction

This is a document designed to provide guidance and specific techniques beyond the basics above for developing high-performance applications.

The document is organized practically: what can be done to tune each layer, from the hardware up through the system software to the application and the reports.

Please note that not all techniques are necessary, or even appropriate, for all situations. The practices in this document will help to provide you with a solid understanding of your application and a foundation for a high performance solution.

The overall organization of the document is in layers, checklist style. By working through this document from beginning to end, you can determine which best practices have been applied, and which can be.

The Tuning Layers

The layers addressed are:

- The Hardware
- The Operating System
- SQL Server
- Analysis Services
- The AppSet database
- Data load processes
- Reports and Input Schedules
- Application Maintenance

Tuning - Minimum Consideration

If nothing else, you should focus on these things:

1. Get big servers
2. Eliminate HTTP connections
3. Use EVDRE for Excel templates

Hardware Configuration

Hardware configuration can be very complex, and its relationship to performance may not be readily apparent. You should engage technical consulting to ensure all details are in order. But your review should touch on all these points:

How Many Servers

While your performance profile may vary, for large applications you should have at least three servers: A web, application and database server (SQL box and an OLAP). You may consider load balancing of the web and application tier. If possible a fourth server is recommended, by

splitting the OLAP and SQL components on different servers will provide the most scalable solution.

In release 4.2, the web and application tier could not be separated. When testing 4.2, we had found that having 1 web/app server with about 300 users accessing it at one time, we would max out the web server and started getting connectivity error messages. While we found the breaking point at 600 users for a single server, we found 300 users took a while to get to the breaking point.

With release 5.x, the web and application tier have been split allowing better scalability. By splitting the web from the application tier there are additional options to scale out both or either tier via load balancing. The appropriate configuration can only be determined when the application design is known. Working with the SAP resources will provide the experience to assist.

Another test on release 4.2, with 4 hyperthreaded (8 virtual) 3.5 GHz CPUs, multiple high speed I/O channels, and 6 GB of RAM, SQL and AS are both on the same server. We have good performance on this server, as many of the best practices in this document have been implemented. Similar results are expected with release 5.x.

What's in the Server?

It is no secret that screaming fast hardware will typically provide the biggest performance boost per dollar spent. In particular:

- Lots of really fast CPUs, dual core or hyperthreaded.
- Enough RAM to hold the entire database and all processes.
- High speed, properly configured, I/O devices.

For any customer purchasing hardware today and concerned about performance, they should consider no less than:

- Four dual core 3GHz or faster CPUs
- 8 GB RAM Minimum (16 or 32 is better)
- Three or four separate RAID drives and controllers for operating system and files, data, temp and log files.

Don't forget the clients

Because BPC is a client application, client workstations should have as much RAM as possible. Informal testing showed that upgrading a client machine RAM could reduce complex expansion and report times by over 50%.

In case the clients are using Citrix on shared hardware, please contact technical consulting.

Operating System Configuration

You should be running Windows 2003 Enterprise Edition to take advantage of large memory and multiple CPUs.

Memory Configuration

'PAE' or Physical Address Extension is an operating system memory switch that allows Windows to effectively address memory in excess of 4 GB. Details are beyond the scope of this document; you should make sure all memory is addressable.

Disk Configuration

In general, you should have data, logs, tempdb and the OS on separate physical drives.

Virus Scanning Software

Make sure Virus scanning is limited to the files absolutely needed (so don't virus scan the transaction log or the reads/writes done by SQL and Analysis Services).

Setting Connection Timeouts (IIS 6.0)

Connection time-outs help reduce the loss of processing resources consumed by idle connections. When you enable connection time-outs, IIS enforces the time-outs at the connection level.

To set a global WWW or FTP service connection time-out value:

1. In IIS Manager, expand the local computer, right-click the Web Sites or FTP Sites folder, and click Properties.
2. On the Web Site or FTP Site tab, in the Connection timeout box, type the maximum number of seconds that IIS should maintain an idle connection before resetting the connection.
3. For the WWW service, verify that the Enable HTTP Keep-Alives box is selected.
4. Click Apply, and then click OK.

AppSet Database Configuration

Partitioning for very large applications

Partitioning the main cube has had a dramatic impact on performance at some customers. The default partition model is to partition by time, as below. Partitioning is perhaps the single most significant thing you can do to improve performance in a very large application.

Try it in development! The division of a cube into multiple partitions is transparent to the user. When a query requests data that spans multiple partitions, Analysis Services uses aggregations or scans data in each partition and then combines this information to resolve the query and return the results to the user. You can significantly increase query responsiveness and processing performance by horizontally segmenting the data by one or more keys, such as date or department, and dividing the cube into multiple partitions.

When and Why

How do you know you'll benefit from Partitioning?

- When a partition file exceeds 2.5 GB or more than 10 million records, these are two general rules of thumb and may vary with circumstances
- When you have many alternate hierarchies

The runtime performance degradation is typically due to having to refresh the cache from disk over and over. If there are lots of alternate hierarchies, it is especially problematic, as the time to reprocess each partition grows with the number of hierarchies.

Other Benefits include:

- Multiple partitions can be processed in parallel
- Only some partitions will need to be processed
- Different partitions can have different storage modes
- Partitions can be refreshed individually

How

There is a separate document which contains details on how to do the partitioning. Please work with SAP technical services.

Cube Processing by Partition

Once you've partitioned, you can choose one (or both) of the following:

- Only process the partitions which have changed, even when you process the FACT table.
- Process the partitions in parallel, rather than one at a time (the default).

How to decide which partitions and when to process? Here are a few guidelines:

If the customer modifies one or more dimensions before starting each reporting cycle, this requires a full processing, and in the presence of multiple partitions, it may end up being slower. However, if the change to

the dimension is limited to adding new members below existing parents, an “incremental” processing may be enough. This may allow for some fine tuning of the processing to be performed.

Please remember that our optimization tools are not as sophisticated as the above possibilities allow, and the user may need to revert to DTS and Analysis Manager to do the processing. There is a custom utility from Microsoft (<http://www.microsoft.com/downloads/details.aspx?FamilyId=A2EEF773-6DF7-4688-8211-E02CF13CBDB4&displaylang=en>) which will provide this capability.

SQL Fact Table Indexes

Customizing the Indexes

The Indexing scheme should be changed so that the WB, FACT and FAC2 table indexes are optimized for performance. To set the index for a given application, use the BPC Web Administration > Set Application Parameters > DimsForFactTblIndex. This will ensure that any BPC processing of the tables will set the index appropriately. Manually adjusting the index at the SQL tier will result in the lose of customized indexes upon BPC processing.

Fact tables index change:

Changes:

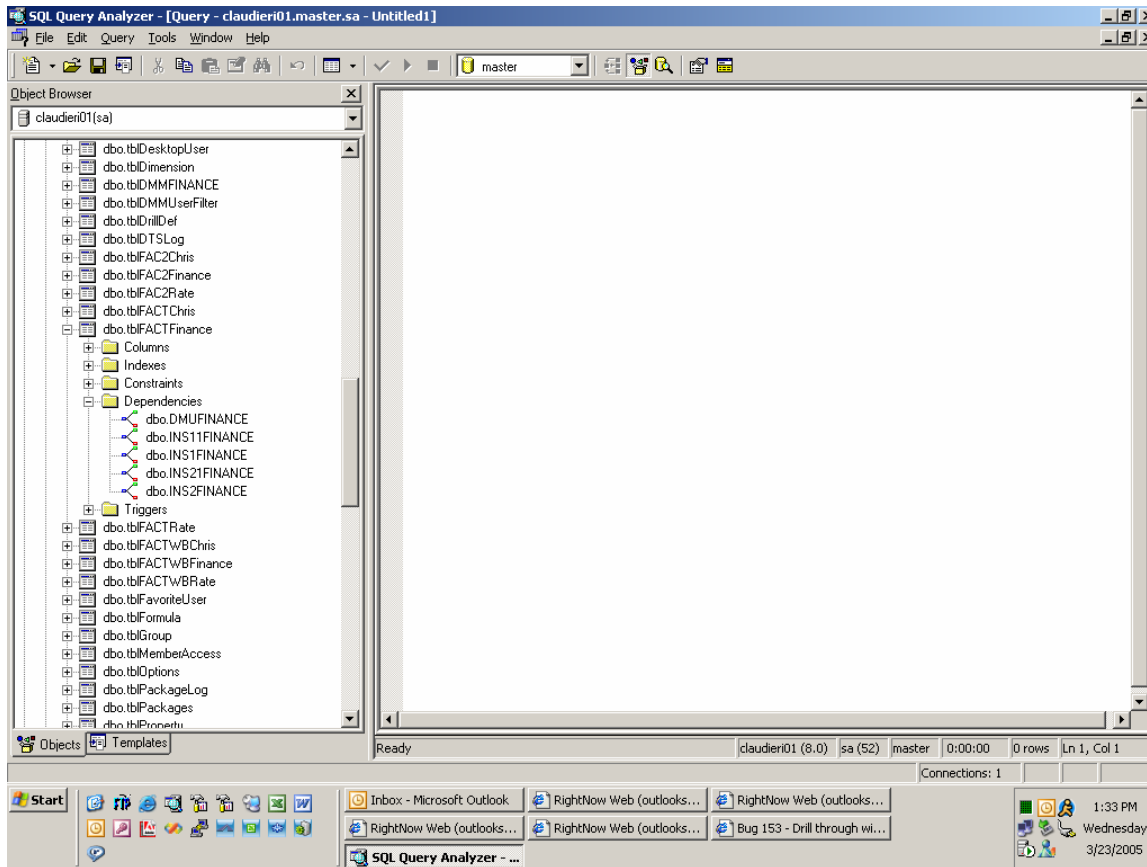
1. Get rid of all individual indexes on Fact, Fac2 and Write-back table
2. Create one clustered index on Fact table on FIVE most selective columns with fill factor 85%. The suggested columns and order are as following (if applicable):
Category, Time, Entity, Account, RptCurrency
3. Create one clustered index on Fac2 table on FIVE most selective columns with fill-factor 100%. The suggested columns and order are:
Category, Time, Entity, Account, RptCurrency
4. Create one composite index on Write-back table on FIVE most selective columns. The suggested columns and order are:
Category, Time, Entity, Account, RptCurrency

If you want to modify columns and order of the indexed fields in fact table, you can use following application parameter”

DimsForFactTblIndex: input dimension names separated by comma, and go to admin console and run modify application to apply the change

Dependencies

Dependencies link indexes to tables. You can verify that they are in place by going to the object browser in SQL Query Analyzer as per the screen shot below; your dependencies should look like this:



Partition Aggregation Strategies

When a cube is created in Analysis Services, there are initially no aggregations defined. Only data at the base level member is physically stored in the cube, so all values at aggregate levels of the dimension hierarchies are calculated on the fly when the cube is queried – you are then not benefiting from OLAP's ability to physically store pre-aggregated values to increase query performance.

Why is this so? It would be impossible to calculate and store all possible cells at aggregated levels; the physical storage needed for the cube would be huge and it would take a very long time to process. And then, many of the cells would never get queried.

What is needed is to selectively define aggregations for the parts of the cube which will be queried most often. This can only be done on a case by case basis when the application is finished and stable. If you are always creating new reports, they can be slow and you may have to retest and recreate new aggregations. The manual steps required for maintaining aggregations can be included in the process for transferring an application update from a development to a production environment, which with BPC is a largely manual process anyway. The only difference is that you need access to the Analysis Manager application on the server, so IT involvement is required.

Finally, be aware that BPC does not support aggregations natively. You will lose your aggregations when you:

- Process dimensions with multiple hierarchies
- Save applications
- Optimize with full process (with or without compression)

System Software Configuration

System software refers to the SQL, Analysis Services, and other layers above the Operating System but not including BPC and the application itself.

SQL Server Configuration

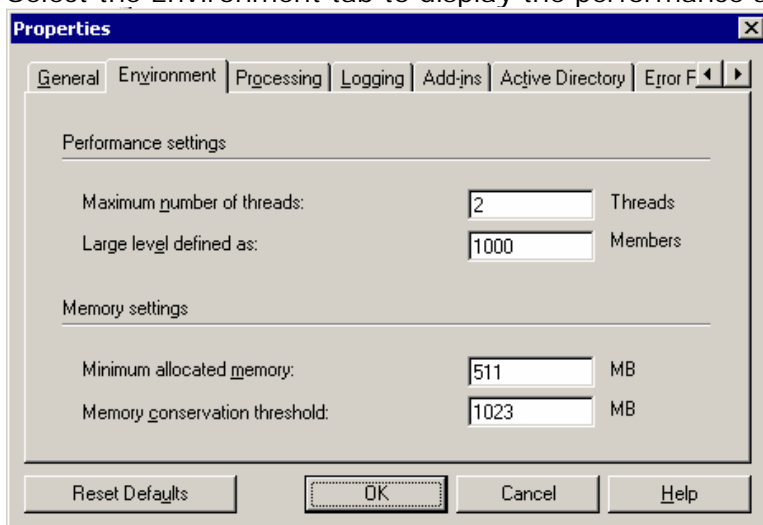
Make sure you are running the SAP supported SQL Server version for both SQL and OLAP (see software master installation guide).

Make sure AWE is configured for the large amount of memory you've installed in the hardware. If disk I/O is an issue, look into using file separation techniques.

Analysis Services Configuration for SQL 2000 only

The following procedure sets up Analysis Services so that BPC can give you greatly improved performance on data sends and retrieves.

1. On the server running Analysis Services, select Start > Programs > Microsoft SQL Server > Analysis Services > Analysis Manager.
2. Expand the Analysis Servers folder, then right-click on the Analysis Server, and select **Properties**.
3. Select the Environment tab to display the performance and memory settings:



4. Consult the following table to change these values based on your server setup:

Setting	Recommend value
Maximum number of threads	Two times the number of physical processors in the system (not including hyperthreading).
Large level defined as	Maximum number of levels in the largest dimension, or 10,000 if the largest dimension has more than 10,000 members.
Minimum allocated memory	One-half of the server's RAM not to exceed 3GB (3072)
Memory conservation threshold	Maximum of 2047MB for 2GB of RAM or Maximum of 2.7GB if set for 3GB with the 3GB switch with 4GB of RAM (Enterprise edition OS)

These parameters assume Analysis Services is installed alone. If Analysis Services is installed with other base components please contact an SAP technical consultant for proper consideration.

The max number of members "Large Level..." is by default set to 1,000. To be precise, this relates to the number of members in a single level. It will not hurt to set this to at least the number of members in your largest dimension, or 10,000, whichever is less.

5. Click OK, and then close Analysis Manager.

Analysis Services Configuration for SQL 2005 only

We have found the majority of the setting for SQL2005 should follow the Microsoft documented guidance. We have found recently that Query/MaxThread value should be 10 (default value) plus the total number of BPC databases (AppSets) plus 2 times the number of CPUs. Without making the appropriate setting, processing time will be significantly slower. For more details contact the SAP support team.

Operating System Services

Turning off services that are not required will save memory and CPU. Following is a list of services that we don't need on BPC server.

- Alert
- Application Management Transfer service
- ClipBook
- Computer Browser
- Distributed Link Tracking client
- Distributed Transaction Coordinator
- Fax Service
- Indexing Service
- Internet Connection Sharing
- Logical Disk Manager Administrative Service
- Messenger
- Microsoft NetMeeting® Remote Desktop Sharing
- Network DDE
- Performance logs and alerts
- Protected Storage
- QoS RSVP
- Remote Access Auto Connection Manager
- Remote Access Connection Manager
- Remote Procedure Call (RPC) locator
- Routing and Remote Access
- RunAs service
- Security Account Manager
- Server
- SmartCard
- SmartCard Helper
- System Event Notification
- TCP/IP NetBIOS Helper Service
- Telephony
- Telnet
- Uninterruptible Power Supply
- Utility manager
- Windows Installer
- Windows Time

You should work closely with technical consulting and your IT team to ensure that these services are not actually in use by other processes on the server, and that their disabling will not adversely affect functionality or performance.

BPC Parameters

SAP BPC Send Governor Configuration Modifications

Background

Some background on what the Send Governor (SG) accomplishes, so that you can understand better how this works. The SG is designed to manage the Microsoft Analysis Services locks. This ensures consistent performance for the user and avoids deadlocks.

Configuring the Send Governor

Send Governor values to set in tblDefaults are:

Send Governor Settings (KeyID)	Default	Batch mode	Real time mode
THREAD_MAXNUM_SG:	3	0	Increase
INTERVAL_CHECK_SEND:	3000 msec	Increase	Decrease
MAXCELLS_THREAD_SG:	1000000	Increase	Decrease
UNITPER_SP:	1000000	Increase	Decrease

For an AppSet which will have a lot of concurrent send activity, you should consider adding an additional 1 to 2 seconds (on average) during sends in order to help improve data read and lite optimize (specific function of BPC administration) performance. Please note that the time to do a Send from BPC for Excel will include both the time to "send" plus (on average) half the Interval_Check_Send time.

We saw a significant drop in locks by decreasing the THREAD_MAXNUM_SG to 1:

A customer quote:

"As soon as I saw more than 35 users concurrent I could see the locks higher than 10 for more than 20 seconds. I tried switching the thread_maxnum_sg to 1 and immediately (after about 10 sec) the locks dropped...At that moment the wbtrigger was at 10k rows. Noticing this I tried putting the wbtrigger to 20K and did not notice any huge locks anymore...

During concurrency of about 70 users all remained well...

Seeing this I would need more time for testing with even higher wbtriggers, at this stage it looks like the thread_maxnum_sg parameter is making the real difference"

Application Customizations

A core area of planning for large applications consists of knowing what data will flow into which of the FACT, FAC2 and WB tables when, and how. By proactively managing this flow, you are taking the most significant step to ensuring a smooth performance profile for your applications. [Appendix C](#) provides a little more background on the Fact tables and how they're used.

Consolidations Package Changes

Intercompany Package

You should configure the package to write to FAC2 not to WB.

Currency Translation Package

Currency translation will generate a lot of records in the database. These can be created in the WB table by default, but this can result in (worst case) millions of records in the WB table, which would create a serious performance issue. At one customer in Europe, the currency translation process generated 2.5 million WB records.

The package should be configured to insert into the FAC2 table. The default FXTRANS logic can also be modified to performance this function.

Note: both packages can support the insertion of records into FACT after which we can process just the FACT partition. This would be seriously considered for a very large production environment where the FAC2 table will be heavily used by other processes.

Data Manager Package Changes

As mentioned above, smart use of the WB, FAC2 and FACT tables is essential.

By default our import package writes into FAC2 and triggers a processing of the associated partition. When many users concurrently import their data, all the processing will easily lock the system. This situation is exacerbated if other processes are being run, such as FX conversion or Intercompany eliminations.

This condition has been particularly acute when users are allowed to run packages whenever they choose, often resulting in a large number of submissions in a very short period of time (at budget or closing deadline, naturally).

The following techniques can be used to mitigate the performance impacts:

1. You can import into WB table if you are importing so frequently that locking has become an issue, especially if the imported files are relatively small. A scheduled light optimization will take care of periodically emptying the WB.
2. "Throttling" the throughput. In a customer case, the submitted files are not immediately imported, but are put in a queue. A separate process merges the files and imports the merged file. This process is both more efficient and reduces locks.
3. You should also look to use bulk data loads into the FACT table if there is a maintenance window which allows for such loading.

Application Design

While by no means an exhaustive discussion, a few points are worth mentioning.

Dimensions

- 1) The "UNARYOPERATOR" property should be avoided if at all possible.
- 2) The choice of concurrent dimension is key. BPC locking is by three to five dimensions: for example a customer uses the Entity, Category and Time type for the other two locking dimensions. The combination of Entity, Category and Time is locked by BPC when data is sent. (This is one of several kinds of locks in the system: SQL has locks, Analysis Services has locks, and BPC has locks. Here we are talking about BPC Locks). At one customer, they were budgeting and forecasting by function – Sales, Marketing, IT, Legal. A Function dimension was secured and used for this purpose. Unfortunately, when a user hit "send" they were sending to all entities. So everyone else was completely locked out until that send ended. The sends queued up quickly and response time went from a minute to 10 or 20 minutes. Solution: The concurrent dimensions were changed to the function dimension instead of the entity type dimension. Therefore the legal entity dimension was not considered in the lock (albeit still a secured dimension). The problem went away.

Alternate Hierarchies

Alternative hierarchies in BPC 5 will show dimension processing to be faster than 4.x since it uses a different schema. However query performance will still be impacted. Consideration of the need for alternative hierarchies is necessary to ensure optimal performance for reporting.

Number of Dimensions

More dimensions create more complex joins and retrievals. If you can break a large cube up into “business process specific” cubes with simpler dimensionality, you can gain performance (and usability) improvements. This is anecdotal.

Reports and Input Schedules

General Guidelines

When building or reviewing sheets:

- See how much data is to be entered: many small sheets vs. one big workbook have advantages and disadvantages and should be discussed.
- Avoid refreshing after sends if possible. Use Excel formulas to calculate totals instead of retrieving them. This makes the process even more real-time.
- Use park-n-go if applicable. This discourages multiple send and refresh.
- Try to avoid using multiple EVGETs in a single cell. (i.e. EVGETs + EVGETs)
- Try to avoid having the dimension parameter of an EVGETs formula depend on the result of another EVGETs formula. See reporting best practices on the Corporate Performance Management Community on SDN (<https://www.sdn.sap.com/irj/sdn/bpx-cpm>)

“Asymmetric Queries”

Be Aware of

- Sheet calculation order
- How Row and Column ID headings are created
- Nested Expansions

BPC for Excel queries are designed to optimize the refresh from the Column/Row grid; meaning it expects some commonality in the columns (i.e. Time dimension members across all columns and most, if not all, columns point to the same CV). When reports vary from this format the performance degrades. An example of a report that doesn't follow a ‘clean’ grid format is one where each column has a different set of dimensions mapped to it. In other words, it is not just one dim across the columns but many dimensions referenced with consistency.

For example:

Assuming a basic 4 dimension application, (To really see the performance degradation you need an 8 to 12 dimension application)

If you have time in the columns and accounts in the row, all other dimensions are in the page key (a single reference per query). This is ideal.

Now if you have category and time in the columns and accounts in the row... you're still good, because entity is in the page key.

If you have a different entity, category, and time period in each column and accounts in the row, performance is terrible.

So the bottom line is the nested expansion cannot assist here. It does not depend on expanding up to 3 dimensions in a column or row. It is related to varied dimension members for each cell. There is no way to optimize the query for the data retrieval.

This applies to the older expansion functions EVEXP, EVENE, etc., and, as stated below, it is not quite true with EVDRE. Yet, certainly any function will have a harder time doing its things as the number of dimensions to nest increases.

EVDRE

EVDRE can have a huge impact on performance in BPC. Example:

We were working on a customer that had to have the dreaded „triple-nested expansion“ and it was taking > 60 minutes to expand and zero suppress to a result set of ~2000 rows.

The same triple-nested expansion under using EvDRE now takes less than 30 seconds to fully expand and refresh.

EVDRE can also provide benefits from its built-in query engine. For example EVDRE goes directly to SQL when retrieving values that exist in the FACT tables. This may have a dramatic impact on scalability, because OLAP does not scale as well, especially during periods of heavy data input activity.

At a European customer we did a stress test of their system using EVDRE. This was a complex schedule which took 3 minutes to refresh when you had the system to yourself (a high-end multiserver environment). Running sixty simultaneous users, the query time increased to only 3 minutes 45 seconds. This is very good scalability.

Logic type (MDX, SQL, or SQL Server Stored Procedures)

Generally, most applications strictly use two versions of "logic" to solve calculations in BPC, either MDX or SQL logic (a proprietary syntax of BPC). The big difference between MDX and SQL is that MDX logic, generally speaking, is not scalable. It means that when the number of concurrent users increases the performance degradation is huge. For this basic reason the SQL logic is preferred to MDX.

The approach to using MDX is:

1. Dimension logic is useful in the account dimension for the ratio (KPI) account only. All other formulas should be better defining on SQL
2. Dimension logic which "spans" time (such as dynamic open balances, etc) should be used with great care. These are often better performed in SQL logic.
3. Due to the big impact on the performance, do not define a large quantity of hierarchies in the dimension.
4. The source/destination regions defined in the logic must be small as much as possible.
5. Reduce as much as possible the number of COMMITS
6. Be very cautious about using MDX-based logic calculations dependent upon large dimensions. At base-level members, these MDX functions will perform well and often are desirable; however at parent levels in large hierarchies, these calculations often result in massive Analysis Server CPU utilization.

The approach to using SQL is:

1. Select the right trigger in order to work only with the existing records and not with all possible cases. For example; in the budget application you have to calculate all revenues by region and product. In this case you have REVENUES=UNITS*PRICE. The right trigger is units because normally price is define for all products vice versa units are defining per regions/product, this because not all products are sold in all regions. In this case the system will apply the formula only for the existing cases, vice versa if you define PRICE as trigger the system will execute the calculation also in case the product is not filled, of course the result is the same but in the second case (Price as trigger) the system has to execute more calculations.
2. Load in memory only the information that you need. For example if you have to calculate REVENUES you define as a region only PRICE and UNITS, you do not need to load in memory the account REVENUES, the system will be faster because will scan less records.
3. Use properties when possible to write more efficient & "compact" logic. For example if you have to apply the same calculation on a specific set of accounts instead of specifying all single accounts, you can define a property and in the SQL logic you can test the property (using a SELECT statement) in order to define the account set. This approach is much faster than specifying all accounts and then using a WHEN criteria to perform the "filtering".

4. Reduce as much as possible the number of COMMITs – the GO command can be a useful alternative but care should be given here as well.

Important: even if at a first glance it may seem that the GO instruction can be used in place of a COMMIT instruction, this is not quite true. All instructions that are COMMIT-specific (like for example XDIM_MEMBERSET) are still COMMIT-specific and not GO-specific. In other words you cannot redefine the data region to process for each GO instruction, but only for each COMMIT instruction. The GO instruction only sets a stop-and-go point between WHEN / ENDWHEN structures of the same COMMIT section, i.e. of the same data region.

Any GO instruction defines the end of a logic section, more or less like a COMMIT instruction, in the sense that the logic is executed normally up to that point.

Do not leave too many calculations in default logic. Typically currency conversion or Intercompany Eliminations can be left to a later phase.

Finally, as a third alternative to standard BPC-based SQL logic is pure SQL Server stored procedures. BPC SQL Logic allows for the calling of SQL stored procedures directly from the BPC logic, thus it makes database-only processing a real possibility. This last alternative might be useful under the following circumstances:

1. There is a large amount of data to be allocated
2. Standard SQL Logic requires heavy use of temporary "memory variable" for its processing

Of course, making the decision to use SQL Logic versus SQL stored procedures is a careful balancing act. SQL stored procedures do not allow for simple administration like the BPC SQL Logic does, so this can often require BPC Administrators to rely on IT-oriented database resources. However, from a scalability and performance standpoint, the SQL stored procedures method will often times be preferable.

Application Maintenance

While by no means an exhaustive discussion, a few points are worth mentioning.

Compression

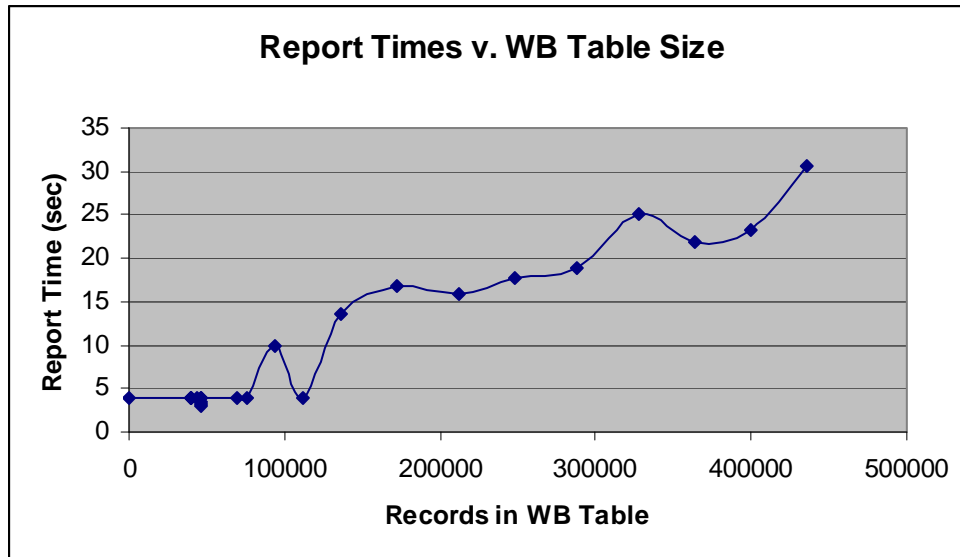
Data compression (through eAdmin, not other mechanisms) can have a very significant impact on cube and partition processing times. It is nearly linear: If compression can reduce the number of records in the Fact tables by 50%, you can expect a nearly 50% reduction in the time to process the cube. This will, of course, vary with alternate hierarchies, etc.

Here are three customer case studies:

Test	Pct reduction in records	Pct reduction in process time
1	75% (from 40 to 10 million)	70% (from 2.5 hrs to 45 min)
2	50% (from 32 to 16 million)	67% (from 2 hrs to 40 minutes)
3	75% (From 4 to 1 million)	(from 30 min to min)

Lite Optimization

As data grows in the Writeback table, it has a noticeable impact on report performance. At the Microsoft lab in the UK, we were using a test application based on a real customer. The time to run the MIF21 report grew along with the WB Table size. The scatter plot of all data points from both tests 1 and 2 is shown below. Note that keeping the WB table below 80,000 records kept the report running in under 5 seconds.



Lite Optimizes were run while reports were being run, with no system downtime. By regularly performing Lite Optimizes, report performance can be kept reasonable.

Storing Historical Data

A very large amount of data can be a problem for processing times, especially if the maintenance window is small.

Many times you will want to store more information than needed on a regular basis. This could be in the form of versions of a Budget or Forecast, number of years worth of data, storing LC and USD for USD entities (or what ever the primary reporting currency is), or before and after allocation results. It is important to re-consider this need so that the Application Set in use shows what you will need on a regular basis.

- 1) Clear data that is no longer in use
 - a) After the Budget is finalized, you can typically clear the data from the version categories.
 - b) Additionally, after the Budget for the current year is complete, prior year's Budgets can be removed.
 - c) Store only Forecast versions that you will be analyzing
- 2) If you want to have those versions for reference or audit, move them to a separate appSet (which will almost never be used) and clear them from the active one. Remember multiple appsets on the same server will take resources, please evaluate this accordingly.
- 3) Archive historical data
 - a) Many companies only want to analyze a certain number of year's worth of data. If you have data outside that threshold, extract the data or backup the database so that you can always get back to it and then delete it from the database.
 - b) You can also use Book Publication to display historical reports on an as needed basis without storing the data in the fact table.
 - c) Create a historical appSet to hold historical information that few people need to see. This is often the best choice, as you can also maintain the historical entity and other structures for historical reference. At a customer, they create a historical appSet at the end of each fiscal year, so they have each year's "snapshot" available when needed. Disk is cheap.

Keep the Wizards directory small

This reduces the initial load time. Not a big deal in many cases, but can affect user experience.

Appendices

A: Identifying Performance Bottlenecks

Monitoring Tools

There are two basic performance monitoring tools provided with the Windows server operating system: the Task Manager and the Performance Monitor.

The Task Manager should be tracking on the Processes tab, with the "show processes from all users" box checked. The CPU header should be selected, so that the processes using the most CPU time will be identified at the top of the list. Critical to the success of this test

The performance monitor provides much more detail than the task manager, but it must be configured to capture the relevant metrics. Performance Monitor can also capture the performance profile for analysis later. This document doesn't document how to use performance monitor, but will identify what to monitor.

Other Monitoring tools include:

- a) Client and server logs
- b) Black box – SAP product support
- c) PSS Diag – Microsoft Professional Services

Memory

Although all of the counters in the Memory performance object are useful, two stand out when measuring Analysis Services overall performance:

Memory \ Pages/sec

This performance counter indicates the number of I/O operations needed to support virtual memory. Ideally, this number should be as low as possible; a high number demonstrates too little available physical memory. Increasing available physical memory should reduce the number of page faults, and therefore reduce the amount of virtual memory used to support active processes such as Analysis Services.

Memory \ Available bytes

This performance counter indicates the amount, in bytes, of available physical memory. Combined with the Pages/sec system counter, this counter can be used to further quantify the amount of available physical memory.

Disk I/O

Disk storage performance is central to Analysis Services performance. The following counters can be compared directly with various performance counters maintained by the Analysis Services performance objects to determine if disk storage represents querying or processing performance impact.

A common cause of poor disk storage performance is the performance of other applications running on the system. Disk storage supports all applications running on a given system; active applications draw resources away from Analysis Services. These performance counters can provide a better picture of absolute querying and processing performance by comparing all of the disk storage activity on the system with the disk storage activity tracked by Analysis Services.

Analysis Services: Agg Cache \ Evictions / Sec

Evictions measure how frequently Analysis Services is flushing its cache and refreshing from the database / disk. "Cache Thrash" can make performance very poor.

Physical Disk \ Avg. Disk Bytes/Read

This represents the average number of bytes transferred from disk storage during a single read operation.

Physical Disk \ Current Disk Queue Length and Physical Disk: Average Disk Queue Length

This counter represents the current number of queued disk operations. If this number spikes during poor processing performance, especially during the base or aggregating phases, then the current disk storage solution may not be adequate to support the needs of Analysis Services. Ideally, the value of this performance counter should be as low as possible at any given time.

Processor

Scaling up to multiprocessor servers allows for much greater Analysis server performance. Scaling up, however, does not necessarily provide a linear increase in performance. Other factors, such as physical memory or disk storage, can affect the increase provided by scaling up an Analysis server.

Processor \ % Processor Time

If you are consistently above 80-90% find out what process is taking all of the CPU time. The Task Manager will provide the information regarding which process is actually using the CPU. Please contact SAP Services and Technical Consulting if the total CPU is regularly running over 90%, or is steadily using up 100% of one or more of the CPUs in the system.

Analysis Services

Analysis Server: Connections \ Current Connections in Progress

This counter indicates all the connections waiting on OLAP at any given point.

Analysis Server: Locks \ Current Locks

This counter is particularly useful to Everest. It measures the current number of locked objects, and can indicate when locks are preventing users from accessing data in a timely fashion.

Analysis Server: Locks \ Current Lock Waits

This counter measures the number of clients waiting for a lock.

SQL Server

SQL Server: Buffer Manager \ Buffer Cache Hit Ratio

The buffer cache hit ratio measures the percentage of pages that were in memory and did not require a disk access to get at the data. A buffer cache hit ratio should be nearly 99 percent. If the ratio is lower, there may be memory constraints that affect performance.

Appendix B: Cache Evictions and Locking

Cache Evictions and Cache Thrash

Copied from an Analysis Services performance tuning Microsoft tech guide:

Cache evictions are an indication of memory pressure. If the cleaner thread is evicting cache entries, increasing the amount of memory on the Analysis server and increasing the memory conservation threshold may improve query performance. If cache evictions are occurring during parallel processing, consider processing fewer partitions in parallel.

Cache evictions can also be a sign of memory constraints, and you can check for memory limitations other ways. But when data is invalidated (through inserts to the WB table or

through data loads, for example) the cache is also evicted because the data in the cache is no longer valid. That's the issue we usually encounter.

Appendix C: FACT, FAC2 and WB tables

At a summary level it could be said that there are 3 tiers of storage:

- WRITEBACK table
- FAC2 table
- FACT table

This scheme was adopted as prior experience in developing and working with other Corporate Performance Management (BPC) products showed that large applications experience degrading performance as data volumes and concurrency of users increase. The intention, successfully achieved, was to allow a scalable solution no matter how far a client needed to take the application.

In contrast other vendors adopted a 'black box' approach which, when a certain scale is reached, necessitates moving to different software from the same vendor when but with a changed architecture, data model or size of database albeit with the same functionalities. Having to 're-scale' as the vendor does is not uncommon. As an example, Hyperion FM reached the limit of 100.000 records in their custom dimension sub-cubes in several large applications. This limit has been removed in version 4.1 but with an associated impact on performance.

The distinction comes down to whether or not a product has been designed to scale or has inherent design limitations necessitating redevelopment of the software.

Each application will have three tables are associated with it. For the application 'Finance', the three tables are named 'tbIFACTFinance', 'tbIFAC2Finance' and 'tbIFACTWBFinance'.

WB – **real time** data input (ROLAP partition)

This is data that is the most current data sent to the system. Data sent by BPC for Excel data sends and Investigator browser data sends is placed in real-time storage.

FAC2 – **short term** and Data Manager imports (MOLAP partition)

This is data that is not real-time data, but is also not in long-term storage yet. When you load data via Data Manager (automatic data load from external data sources), it loads the data to short-term storage so that the loaded data does not affect system performance. Only the cube partition associated with this table is processed, so the system is not taken offline.

Fact – **long term** history (MOLAP partition)

This is the main data storage. All data eventually resides in long-term storage. Data that is not accessed very often remains in long-term storage so that the system maintains performance

This structure allows SAP BPC to maintain the same performance over time even when there is a large increase in data volumes.

Periodically clearing real-time data greatly optimizes the performance of the system and an "Optimization" process is required (this could be scheduled automatically based on given parameters like a numbers of records threshold).

- Lite Optimization:
 - Clears Real-time data storage (WRITEBACK) and moves it to short-term data storage (FAC2). This option doesn't take the system offline, and can be scheduled during normal business activity.
- Incremental Optimization:
 - Clears both real-time and Short-term data storage (WB and FAC2) and moves both to Long-term data storage (FACT).
 - This option should be run when the system is offline, but it will not take the system offline so it should be run during off-peak periods of activity.

- Full Process Optimization:
 - Clears both real-time and short-term data storage and processes the dimensions.
 - This option takes the system offline and takes longer to run than the incremental optimization.
 - It is best run scheduled at down-time periods – for example after a month-end close.

The **Compress Database** option is available to rationalize the Fact Tables. “Compress” sums multiple entries for the same CurrentView into one entry so that data storage space is minimized. Compressed databases also process more quickly.

Appendix D: Other Resources and References

You should definitely check out the TechNet Web site:

<http://technet.microsoft.com>

Microsoft SQL Server 2000 RDBMS Performance Tuning Guide for Data Warehousing:

<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/rdbmspft.mspix>

Microsoft Analysis Services Performance Guide.

<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/ansvcspg.mspix>

HOW TO: Create and Configure Performance Monitor Trace Logs in Windows 2000

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q302552&sd=tech>

HOW TO: Create and Configure Performance Monitor Alerts in Windows 2000

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q302558&sd=tech>

http://www.sql-server-performance.com/performance_monitor_counters_sql_server.asp

Copyright

© Copyright 2007 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.