Performance Analysis and Tuning Guide for SBO Planning and Consolidation version for Netweaver

Applies to:
SAP BusinessObjects Planning and Consolidation version for Netweaver 7.x. For more information, visit the Enterprise Performance Management homepage.

Summary
This best practice guide describes different aspect of performance relevant topics in the context of customer defined functions of SAP BusinessObjects Planning and Consolidation version for Netweaver application. It highlights some of the design aspects that can influence the performance of a specific application as well as providing general guidelines, recommendations and best practices.

Author: Sreedhar Kandagatla
Company: SAP AG, Germany.
Created on: 14 April 2011

Author Bio
Sreedhar Kandagatla works for SAP AG, Germany. He is a senior specialist in EPM Solutions in the Customer Enagagement Unit since 2008 and works in close co-operation with the development, the consultant and pre-sales mainly. He mostly support ramp-up activities and performs expert advice in Pre-sales and consulting activities.
# Table of Contents

Introduction ........................................................................................................................................... 3

System Landscape ................................................................................................................................... 3
  Solution Components .......................................................................................................................... 3
  Server Components ............................................................................................................................. 4
  Load Balancing and Scalability for BPC ............................................................................................. 6

Performance Considerations ................................................................................................................ 7
  Active Concurrent Users in the system ................................................................................................. 7
  Data Volume ......................................................................................................................................... 7
  Complexity of any script logic used for calculations ........................................................................... 7
  Complexity of any Consolidation Logic ............................................................................................... 8
  Complexity of any member access rules ............................................................................................. 8
  Data Volume per query or volume of any operation transporting data ................................................. 8
  Report Design ....................................................................................................................................... 8
  Network .............................................................................................................................................. 10

Performance Analysis .......................................................................................................................... 10
  Resources Utilization .......................................................................................................................... 10
  How to use Logs and Traces of BPC ................................................................................................... 13
  How to use UJSTAT (BPC Statistics feature) ..................................................................................... 22
  Other Performance analysis tools ...................................................................................................... 27

General Performance Recommendations ............................................................................................. 28
  General ................................................................................................................................................ 28
  Hardware and Software settings ......................................................................................................... 29
  NW settings ......................................................................................................................................... 29
  BPC Settings ....................................................................................................................................... 29
  .Net server ......................................................................................................................................... 31
  Load Balancing ................................................................................................................................. 32
  BW Accelerator ................................................................................................................................. 33
  Modeling ............................................................................................................................................ 33
  Reporting ........................................................................................................................................... 37
  Asymmetric Queries ........................................................................................................................... 39
  Logic Best Practice Guidelines .......................................................................................................... 39

Related Content ..................................................................................................................................... 44

Copyright ............................................................................................................................................... 46
Introduction

SAP BusinessObjects Planning and Consolidation, version for SAP NetWeaver (SAP BPC) is a robust planning and consolidation application designed to meet organizational budgeting, planning, consolidation, and reporting requirements. It supports a wide array of top-down and bottom-up financial planning needs, as well as consolidation processes necessary to ensure a smooth and timely financial close, all through a single application. Through the implementation and utilization of SAP BPC, organizations are empowered to meet increasingly stringent regulations, effectively plan strategically and tactically, and obtain the information necessary to gain important insights.

It is a solution dedicated to fulfilling your planning, budgeting, and forecasting requirements, as well as enabling legal and management consolidations and advanced reporting capabilities. Furthermore, it enables all of these functions to operate through a single unified application with a best-in-class user interface (UI).

The primary goal of this document is to help the consultants and customers to analyse the performance issues they might face in SAP Business Planning and Consolidation, version for SAP NetWeaver. This guide also provides some recommendations, best practices and guidelines on how to use, how to correct, how to configure and how to fine tune the product for performance gains. This document can also help consultants understand product technical details, performance influence factors, and performance analysis tools to help guide them on their customer implementation projects.

System Landscape

Solution Components

SAP Business Planning and Consolidation, version for SAP NetWeaver, runs on the SAP NetWeaver BI platform, and also requires a Microsoft .NET application server.

At a high level, it has four tiers:

- **Client tier** – Used to run Zero Footprint (aka ZFP, Web), Office (Excel, Word, PowerPoint), or Admin (Standalone .NET client application)
- **.NET Web Server and Application Server tier** – Used for data mapping and pass through between client and SAP NetWeaver App Servers and perform user authentications
- **SAP NetWeaver application tier** – SAP NetWeaver BW Application Servers (OLAP Engine and Application Logic)
- **Database tier** – Any Database including SAP NetWeaver BW physical data storage
Server Components

To help you understand the configuration 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 marketplace.

SAP Business Planning and Consolidation – Master Guide & Installation Guide:

https://service.sap.com/instguidescpm-bpc

1. **Database Tier**
   - Can be any database supported by SAP NetWeaver (MS SQL, Oracle, DB2, MaxDB, etc…)
   - Full support of 64 bit and Unicode capabilities
   - Common Processing
     - All reads of data
     - All write back of data
     - Metadata read/write
     - File Storage Read/Write
       - Storage location for all application files
       - Book Repository
       - Report & Input Templates
       - Unstructured Data
       - Conversion & Transformation Files
2. SAP NetWeaver ABAP Application Server
   - Supports any Application Server configuration supported by SAP NetWeaver
   - Common Processing
     - Script Logic, calculations, and MDX Parsing
     - Parameter Driven Logic
     - Data Manager Loads and ETL
     - Data Tables to XML Conversions and vice versa
     - OLAP Engine and Calculations

3. Microsoft IIS Web Server
   - Microsoft IIS
   - Collection of CPM Services
   - Performs user authentication
   - Common Processing
     - Send and receive data between client and ABAP Application Server
     - Rendering of System Reports
     - Live Reports Rendering
     - PDF Rendering for Distributor
     - SOAP Processing and conversion of data formats
     - Hosts ZFP

Note: this tier contains 32-bit components that can be installed on a 64 bit platform when utilizing Windows On Windows (WOW), though will still have the same constraints as when installed on a 32 bit platform.

4. BW Accelerator
   - Optional component
   - Used to optimize database read times for large data volumes
Load Balancing and Scalability for BPC

1. **Scalability**
   
   For BPC product Scalability information, refer to this White paper document:
   

2. **Load Balancing**
   
   Depending on the size of the user base and active concurrent users for your BPC application there might be a need to load balance certain parts of your environment in order to deal with the performance needs of your users.

   There are two different components that can be load balanced to enable Load Balancing:
   
   - Load balancing of the BPC .NET Application Server
   - And/Or Load balancing of the Netweaver System (BW System)

3. **.NET Server Load Balancing**:
   
   A load balancer would need to be placed in front of the two BPC .NET application servers to distribute the incoming calls
   
   The load balancers can be acquired from third party vendors, for example the Microsoft Network Load Balancer (hardware or software network load balancers can be used)
   
   A Hardware load balancer is recommended for better performance and in most cases a more accurate distribution of the load
   
   The load balancer would determine which server has fewer loads and send the next user to that server

4. **SAP Netweaver Application Server Load Balancing**:
   
   Load balancing in the Netweaver environment usually means creating additional dialog instances of your SAP System. The Netweaver Message Server automatically handles the load balancing between the different instances.

   You need to setup a logon group in the Netweaver System to which you assign all the system instances to be used by BPC (transaction SMLG) In BPC 7.0/7.5 we have the possibility to point the .NET Server directly to the Netweaver Message Server for the load balancing capability.

5. **SAP Netweaver Application Server High Availability**:
   
   As the High Availability topic on the Netweaver Platform can get very complicated and implemented differently depending on the customer landscape and their needs, please refer to the following links for additional information:

   - Achieving high availability for SAP solutions (SDN):
   - SAP High Availability (SAP Help):
Performance Considerations

Active Concurrent Users in the system

**Concurrent Users:** With the term concurrent users we mean the number of users currently working with the BPC application at any given point in time. This might be users either using one of the BPC Clients (Excel or Admin) or the BPC Web interface. The users are actively working in the system at a normal pace.

**Influence:** The higher number of concurrent users actively using the BPC application the more amounts of resources will be required to handle their requests by the system. It is important to also distinguish the users performing simple tasks, like opening basic reports vs. users performing “heavier” tasks, such as inputting data or triggering data manager packages. The second will generally generate a heavier load on the system.

**Recommendation:** Refer to the BPC sizing guide ([https://service.sap.com/sizing](https://service.sap.com/sizing)) to get a high level view of what amount of hardware you will need in order to handle your user load. In general, you can have approximately 100 concurrent BPC Users per each Application Server instance, so if you have a higher number of concurrent users, at the very minimum you would need to add some additional server dialogue instances. Keep in mind you will need to factor the number of .NET servers as well, and whether or not you will be sharing any of these resources with other installations and server components (like the database itself). These are just a few starting things to discuss when analyzing user load and should be properly discussed and planned for accordingly with the appropriate sizing resource before attempting productive use.

Data Volume

**Data Volume:** The data volume can have a significant impact on performance. Here we mean both transaction and master data. Always remember to properly filter and select the data being operated on.

**Influence:** Large volumes of Master Data (Dimension Members) have an impact on performance for several aspects of BPC functionality, like Logon, Member Processing, Report formatting etc...

Large volumes of Transactional Data have a direct impact on data retrieval times for reports and script logic. When building reports that need to read large data volumes make sure that you keep your BPC cubes optimized. Refer to the “Light Optimization” Section of this document for further details on the optimization routine part of BPC.

**Recommendation:** We recommend only keeping the necessary data in your cubes. Make sure you archive that data not used anymore in your application. In some case it might be of benefit to split your application into two cubes. For example, if most of your reporting is based on the current and last year and for only very few reports you might have to refer to data that is older than 2 years, in this case splitting that data older than 2 years could be of benefit for the performance of your most used reports.

Complexity of any script logic used for calculations

**Script Logic:** The amount of logic that you are using for running custom calculations, currency translations, default logic during write back, etc. can vary from project to project. Especially in planning scenarios customers tend to create large calculations for their specific business needs.

**Influence:** Script Logic can create a significant load on the BW application server if heavy calculations need to be performed. This load can also impact the Database if large amounts of data are being read and written. Remember that the execution of default logic is executed each time that a user inputs data. Depending on the complexity of the code this can significantly extend the write operation for any single user sending data into BPC.

**Recommendation:** Keep the default logic simple and properly scoped to only the records needed for the calculation. Remember to always use efficient coding to reduce the amounts of reads, writes, and processing steps in your logic. This will help to reduce the resource load on the Database and the Application Server. Also make sure to only calculate what you need – high numbers of complex calculations can impact the CPU load on your Application Server. When possible, you should use simple tuple logic in the place of heavy when/rec/endwhen statements.
Complexity of any Consolidation Logic

Consolidation Logic: Consolidation logic configuration mainly depends on business rules, Script logic and Validations. In general, the more complex the business rule, Script logic and Validation are, the more complex the Consolidation logic in general. According to the business scope these things need to be defined in an optimal way. For example: when the group hierarchy structure in the group dimension is very complex. When you run currency translation, you can indeed get a lot of data which might be too more than what you need. Users also need to be very careful with Validations since it can have a negative impact on performance.

Influence: Consolidation Logic has potential to impact performance if the configuration is not optimal. When the script logic runs it can duplicate records based on the group structure and number of group currencies.

Recommendation: In order to reduce the volume of data generated by currency translation, in your ‘G’ type dimension, i.e. GROUPS, find the property ENTITY and clear values in that property for all group members. In your dimension “GROUPS”, you can also find the property STAGE_ONLY, if you have left this property blank for all group members then change it to “E”.

Complexity of any member access rules

Member access rules: The security model put into place will affect performance of BPC functionality such as: data retrieval and data writing, script logic execution, BPF usage, Client login times, consolidation execution, etc.

Influence: The amount of member access profiles and dimensions that are defined as secure will have a direct impact on the performance of the system (as mentioned above). Try to keep these to a minimum.

Recommendation: Security is critical in each project; our recommendation is only to keep it as simple as needed since it does have an impact on performance. Making frequent changes to your member access profiles will adversely affect the logon times for the BPC Excel Clients, as this will require a resynchronization of the local dimension member files each time that a change to the users access has been made.

Data Volume per query or volume of any operation transporting data

Data Volume: The data volume can have a significant impact on performance. Here we mean both transaction and master data. Always remember to properly filter and select the data being operated on.

Influence: Refer to section 3.2

Recommendation: Refer to section 3.2

Report Design

Report: When reading the data from SAP NetWeaver BI InfoCubes, SAP BPC uses two different SAP NetWeaver technologies:

Online Analytical Processing (OLAP) BAPIs, which allow SAP BPC to access the data in SAP NetWeaver BI InfoCubes via the Multidimensional Expressions (MDX) language.

Internal APIs, which enable SAP BPC to use private APIs (unsupported for third-party or customer use) to read data from SAP NetWeaver BI InfoCubes.

Note: The Shared Query Engine (SQE), a module of SAP BPC, determines the optimal technology to use: internal APIs or OLAP BAPIs.

The Shared Query Engine or SQE is the central module for reading all of the data from an SAP BPC application. (Ex: SAP BPC Excel client, the Web, and the data manager). Technically, the SQE in SAP BPC 7.x NW consists of the following four sub-modules. These sub-modules are completely invisible to the end users.
- **RSDRI query**
  RSDRI is the function group in SAP NetWeaver BI that deals with access to InfoProviders. In the SQE, an RSDRI query means calling the internal SAP NetWeaver BI functions to read data from an InfoProvider (Fastest).

- **MDX query**
  SAP NetWeaver BI exposes OLAP BAPIs to access data from an InfoProvider through MDX. The MDX query method in SAP BPC generates the MDX statements required to retrieve data with the SAP BPC application logic applied. It is always used when there are dimension member formulas, measure formulas, or hierarchy parent values to be retrieved.

  - **Axis query**
    To request a contiguous region of data (e.g., an array of data to be displayed in a grid in SAP BPC Excel), the system uses the Axis query, which contains internal logic that determines (among other things) whether an RSDRI query or an MDX query will do the actual retrieval of data. Then, the Axis query calls the appropriate method. In some cases, the Axis query even breaks one request into multiple queries, each of which might call an RSDRI query or an MDX query, and then merges the results.

  - **Cell query**
    In direct contrast to an Axis query, this method specifically deals with requests for sparse data. A Cell query is required because SAP BPC supports users building highly formatted reports by retrieving several different individual cells from an SAP BPC application.

**Note:** As of BPC 7.5 NW SP5 and BPC 7.0 NW SP8, we no longer use Cell query, see SAP Note 1329178

The **RSDRI Query** is only possible if:

- No member formula has been defined on the requested members.
- You use the Periodic measure for an Application of type PER storage, or you use the YTD measure for an Application of type YTD storage, (respectively). No parent node has been requested on the Time dimension.
- No multiple members on account dimensions are requested when account dimension is not set on ROW/COL.
- Only base members have been requested for the dimension which is defined on the ROW/COL in the evDre report.

**Front end formatting – report design considerations:**

BPC could also potentially spend a lot of time in the Excel Client as well. This time will ultimately depend on how you build your Report/Input Schedule.

The EvDRE processing sequence is designed so that the formatting instructions of the Format range do not override the formats of the Before and After ranges as well as of the formatted sets. If some portions of the report are already formatted, the format instructions do not break the predefined formats.

When the system expands an EvDRE report, it processes the tasks in the following sequence:

1. Expands the member sets to apply (this includes the members of the formatted sets, if defined)
2. Performs suppressions (optional — based on the setting in the expansion or option ranges)
3. Refreshes data
4. Applies formatting instructions
5. Applies sorting instructions
6. Inserts Before and After ranges of sorting, with their formats
7. Applies formats of the formatted sets
8. Inserts Before and After-ranges of the expansions, with their formats

Factors that can cause long run times in the front end for reports/input schedules:

1. Formatting
2. VBA or Macros
3. Calculations in Excel (Formulas)
4. Validations in the front end
5. Applying Before/After Ranges
6. Zero Suppression
Network

**Network:** What type of network connections are we talking about? These can be the client connections or the connections between the .NET server and the ABAP server and the Database as well.

**Influence:** It is important that the user accessing the BPC system with the BPC Client or over the BPC Web has a solid network connection. With this we mean that a good bandwidth is guaranteed and that that latency should not be too high.

**Recommendations:** When planning the architecture of the BPC landscape and the Client rollout it is important to consider the location of the users (in case of a global user base) and the kind of access that they will have to the site hosting the BPC servers.

The access speed to the BPC local client cache directory is also very important. The Excel Client is working quite a bit with the local cache. This is generally not an issue if the local cache is on the local hard drive of the Workstation where the BPC Client is installed. However, we see a lot of customers where the My Documents directory hosting the BPC local client cache is stored on a network share. This network share is then accessed by the local PC installation as well as if a Terminal Server (Citrix) client rollout is used. Depending of the location of this network drive compared to the actual workstation/terminal server on which BPC is running, the access times to this share could be quite slow. If the access to the network share is not optimal for the BPC Excel Client we recommend having the BPC local cache directory moved to a directory stored on the local hard drive of the particular machine hosting the BPC client.

Performance Analysis

**Resources Utilization**

**Front End Clients:**

![Diagram](url)

The BPC Client tier takes user requests/actions and converts them into SOAP requests. These SOAP requests are processed by the .NET tier and communicated via HTTP and/or HTTPS protocols. BPC for Excel (and other office clients) are synchronous, i.e. the request is processed immediately and the client waits for a response to be sent back from the .NET Tier. Some SOAP requests from the BPC Administration console are asynchronous. These requests start a process, and then the client continues to check the status of the process via an additional SOAP request until a “Success” or “Error” condition is encountered. SAP BPC clients don’t connect directly with SAP NetWeaver AS ABAP; they always go through the .NET Web server or .NET application server. Based on our experience, the end user machine should not have less than 1GB of physical memory. This is actually quite low and a safer limit would be 2GB.
BPC .NET Tier:

Communication between the BPC .NET and ABAP AS Tiers is performed via RFC calls. Each RFC call requires a dialog worker process. In SAP BPC, little processing remains in the .NET application server. The .NET application server in SAP BPC mostly repackages (i.e., converts) the data from SAP NetWeaver Application Server (SAP NetWeaver AS) into the required format that the SAP BPC clients expect.

ABAP Application Server- Reading data:

When reading the data from SAP NetWeaver BI InfoCubes, SAP BPC uses two different SAP NetWeaver technologies:

- Online Analytical Processing (OLAP) BAPIs, which allow SAP BPC to access the data in SAP NetWeaver BI InfoCubes via the Multidimensional Expressions (MDX) language.
- Internal APIs, which enable SAP BPC to use private APIs (unsupported for third-party or customer use) to read data from SAP NetWeaver BI InfoCubes.

Note: The Shared Query Engine (SQE), a module of SAP BPC, determines the optimal technology to use: internal APIs or OLAP BAPIs.

Database Tier:

The Database Server Tier is responsible for data storage and retrieval. There is nothing unique in the Database Server Tier with BPC 7.x for NetWeaver. The only tier that communicates directly with the Database Server Tier is the ABAP Application Server Tier. The SAP BPC application also creates a number of new tables in the database in which to store its file system data; for example, script logic, excel templates, reports, data manager files, etc.

Note: there are also various files stored on the client side OS file system for caching.

Consolidation scenarios tend to have higher resource consumption within the SAP NetWeaver dialog instances, while planning scenarios tend to have higher resource consumption within the database. In the performance project for sizing guide testing we observed the following:
In the planning scenario testing (at peak load), approximately 70% of overall CPU was consumed by database processing, while the remaining 30% was consumed by processing in the NetWeaver dialog instances.

In the consolidation scenario testing (at peak load), approximately 20% of overall CPU was consumed by database processing, while the remaining 80% was consumed by processing in the NetWeaver dialog instances.

Logon:

All dimension members are cached locally on logon, both for the Admin Client and Office Client. For the Office Client, this process reads your security model: complex security models for member access security can adversely affect performance of the BPC logon. For the Admin Client, security model for member access profile is ignored, as only administrators can logon to the Admin Console. If you have a large number of dimension members and you have a slow WAN connection, logon performance can suffer. So make sure that you have better bandwidth or use Citrix for low bandwidth areas. Dynamic Report Templates are cached locally on logon based on the “template_version” field set in Admin. If you update dynamic report templates frequently, this can impact logon time for the BPC Office Client. Consequently, if these are cached and not updated frequently, then performance of logon shouldn’t be impacted

Note: Logon performance has improved a lot from BPC 7.5 NW SP6, detailed logon process analysis information is provided in Modeling and Reporting recommendations section.

Read Operations on Data:

During read operation dimension members are cached locally on logon, both for the Admin Client and Office Client. Read operation on dimension members will primarily impact logon times. End user runtime should be fast for dimension member retrievals, as they are cached locally. Web client responses may be more directly impacted by dimension members as they are read from the backend (not local cache). Transactional data performance will be impacted based on the number of records per InfoCube (application). Executing “Light Optimize” compresses the cube, rebuilds all cube indexes, and updates DB statistics for the cube. Executing “Full Optimize” will automatically remodel the cube –

Caution: The light optimize could run a very long time if you have a high volume of data in the fact table. BWA can help read times for transactional data. Check the section 5.7 BWA for more details on the usage of BWA.

Write Back Operations on Data:
BPC write back performance is primarily affected by:
Work Status Locks
Validation Framework
Security Model
Concurrency Lock parameters
Generally, the higher the degree of complexity in these areas, the more overhead placed on write-back

Front End Client:
For Front end Macros, Calculations, and Excel Formulas will affect the overall end user performance.
How to use Logs and Traces of BPC

Log and Tracing capabilities are possible for three layers: BPC Client, BPC .NET Server, and the ABAP Application Server system.

BPC Client Tier:

Recapitulation of Client Logs in BPC NW 7.0:
- In order to get detailed client logs in BPC 7.0, it is necessary to switch the Web Appset Parameter “LOGLEVEL” from the default value 3 to 4 (or higher).
- LOGLEVEL controls not only client logs, but also logs on the .Net and ABAP server (SLG1).
- The logs can be found under „My Documents” -> BPC -> Logging
- For every day with activity in BPC a new log file is created. The format is log<mm-dd-yyyy>.txt

Client Logs and Traces in BPC NW 7.5:
- In release 7.5 the client logs are controlled by the xml-file log-configuration.xml. This file is located in the installation folder of the BPC frontends (e.g. C:\Program Files\SAP Business Objects\PC_NW).

```
<%xml version="1.0" encoding='utf-8' %>
  <LogTraceConfig>
    <!-- Trace Setting -->
    <TraceSeverity value="ERROR" />
    <TraceDestination value="C:\Program Files\SAP BusinessObjects\PC_NW\Logging\Trace" />
    <TraceFileCount value="10" />
    <MaxTraceSize value="10" />
    <TraceUser value="" />
    <!-- Log Setting -->
    <LogSeverity value="ERROR" />
    <LogDestination value="C:\Program Files\SAP BusinessObjects\PC_NW\Logging\Log" />
    <LogFileCount value="10" />
    <MaxLogFile value="10" />
    <!-- Application Set setting -->
    <AppSet value="" />
  </LogTraceConfig>
```

- There is no tool available for modifying the parameters; it needs to be done within the file itself (open with notepad).
- In order to analyze the performance of the BPC excel client, the “Trace Setting” is important, rather than the “Log Setting”.

Log-configuration.xml:

Trace Severity:
- “TraceSeverity” is used to do trace filtering; the default value is “ERROR”.
- Currently there are six severities supported:
  - DEBUG
  - PATH
  - INFO
  - WARNING
  - ERROR
  - DONOTHING
- Most detailed client trace information can be obtained by setting TraceSeverity to “DEBUG”; this is the recommended setting for analysis (similar to LOGLEVEL = 4 in BPC 7.0)
- After having finished the analysis the TraceSeverity should be switched back to “ERROR”!
Trace Destination:
- While in BPC 7.0 the location of the log files was fixed (My Documents -> BPC -> Logging), however in BPC 7.5 it can be customized.
- By default the logs and traces are located in the installation folder of the BPC frontends e.g.: <TraceDestination value="C:\Program Files\SAP BusinessObjects\PC_NW\Logging\Trace"/>
- The trace destination can be changed to any other location.

Other Parameters:
- TraceFile Count: The new logging system uses the Round Robin to write traces. TraceFileCount stands for count of Trace File in the Round Robin. Here is an example. Let's assume TraceFileCount is 2. So firstly traces are written into BPCTRACE.0.LOG. (Trace file name pattern: BPCTRACE.x.LOG, x starts from 0 and end with TraceFileCount -1) If the size of this file reaches to MaxTraceSize, a new file BPCTRACE.1.LOG is created. When BPCTRACE.1.LOG reaches the MaxTraceSize BPCTRACE.0.LOG will be overwritten. Default value is 10.
- MaxTraceSize: The max size of single trace file. Default value is 10, unit is MB.
- TraceUser: Can be used to restrict traces to a specific BPC user. Default value is empty.

.Net Tier:

The .NET Tier offers three primary logging mechanisms
- BPC .NET Server Logs
- Microsoft IIS Server Logs
- Microsoft Windows Event Viewer

Each of these logs gives different information that can be used to troubleshoot issues

.Net server logging is quite similar to client logging. The same two types logging occurs on the .NET server as the client side, standard logging and trace files. Both are located in subdirectories under the .NET server installation\Logging directory by default.

The same trace and log levels from the client tier apply here. The log settings can be configured by modifying the log-configuration.xml file in the BPC server installation\Websrvr\web directory.

You can monitor the utilization of the .NET server by reviewing the processes dllhost.exe, which represent the COM+ objects and w3wp.exe, which represents the core IIS/WWW service. These processes can be monitored via Task Manager.
ABAP Tier:

The ABAP Application Server Tier code (written in ABAP) is stored within the UJ package. You can display the package contents in transaction (SE80). This code performs all of BPC’s business logic and can be debugged in real time. It also provides MDX parsing capabilities and is responsible for communication with the underlying database as well as communicating back to the BPC .NET Tier.

Tracing BPC with Netweaver (ABAP):

- NetWeaver offers tracing capabilities as well as debugging capabilities.
- An overview of the Tracing Capabilities within a NetWeaver system which will help troubleshoot BPC issues are as follows:
  - Step 1: Check ST22 to see if any ABAP dumps occur. In general, these would be caused by BPC programming bugs and should be reported to SAP. Some may be caused by system issues, so check the details of the errors to see the root cause.
  - Step 2: Check SM21 to see any connection errors between the .NET Server and the ABAP Web Application Server.
  - Step 3: Check SLG1 or SLG2 for BPC specific logs.
  - Step 4: Enable RFC Trace using ST05.
  - Step 5: Enable any detailed trace using ST01.
  - Tracing and Logging should be used as the first option to troubleshoot ABAP issues within BPC.
  - Debugging can be used if Tracing and Logging don’t solve your issue.

ST22: Check for ABAP Dumps:

Check ST22 to see if any ABAP dumps occur. In general, these would be caused by BPC programming bugs and should be reported to SAP. Some may be caused by system issues, so check the details of the errors to see the root cause.
Check system connections using transaction SM21:

As BPC communication occurs through RFC, all RFC system connections can be checked through transaction SM21. Transaction SM59 can also be used to verify the RFC connection configuration.

Tracing RFC Connection using ST05:

As the BPC Services users are setup as communications users, they do not allow debugging. However, RFC trace in ABAP can be set in ST05.

In order to get a more detailed picture about what is happening on the data base, it is recommended to perform a SQL trace (transaction ST05).

For recording a trace during execution of a refresh/expand, the SQL trace must be activated for the user BPC_SYSADMIN: ST05 -> Activate trace with filter -> User Name: BPC_SYSADMIN (or your backend user if you start a step from there) -> Enter.
Execute the step in BPC excel client. 
Click on “Deactivate Trace” and then “Display Trace”

Search for long running SELECT statements to your Cube. Highlight the SELECT statement by double click.
You can display the execution plan by putting the cursor on the SELECT statement of the OPEN operation and clicking on the “Explain” button.

The same functionality is available in transaction ST12. Additionally it is possible to perform an ABAP trace at the same time.

For more information about Performance Traces please check here:
http://help.sap.com/saphelp_nw04/helpdata/en/5a/ace273ca0211d194b500a0c94260a5/content.html
ABAP Trace:

The ABAP trace (transaction-ST12) is used to measure runtimes in ABAP coding.

Enter a title in the Comment field and the user name BPC_SYSADMIN (or your backend user if you start a step from there).

Set the flag “with internal tables”, click on the button “Further opt.” and increase “Max. size of file”. Click on “Start Trace”, execute the steps to be measured and click on “End Traces & Collect”.

After the measurement you can collect the trace file at the bottom and evaluate ABAP and SQL Trace.

At the top of the ABAP trace you can see the percentage of CPU time in ABAP and on the data base.

Sorting the column “Net” in descending order shows you the longest running parts of the coding on the top.
BPC Application Logs using SLG1:

- Using SLG1, you can view detailed logs written by the BPC Software within ABAP.
- The logs within SLG1 can show different types of information:
  - Error Message
  - Warning Message
  - Debug Message
- To view all logs related to BPC, enter object "UJ" and execute SLG1.
The output of the logs is as follows:

- In this case, you can see all details of the SQE call "UJQ_RUN_XML_QUERY" from the .NET tier in the system logs.
- The SQE logs all details based on the type of call. Therefore, you can see all the MDX statements executed within the system logs as well.
- All calls to SQE are centralized and write to system logs, and therefore, this will give you the log for all SQE calls, whether it is BPC for Excel, Script Logic, Data Manager, etc…
SLG1 – SubObjects:

- To view all logs related to BPC, enter object "UJ" and execute SLG1.
- The SubObject also allows you to only see logs for a specific BPC Module. Choose the F4 Help to get the list of modules you can view subobject logs for.
Use transaction SLG2 to maintain system logs

All other traces can be executed through ST01
- Authorization Check (for ABAP auth objects, not BPC authorizations)
- Kernel Functions
- General Kernel
- DB Access (SQL Trace)
- Table Buffer Trace
- RFC Calls
- Lock Operations

End to End trace instruction is available in 7.5 ops guide:
https://service.sap.com/~form/sapnet?_SHORTKEY=01100035870000726399&_SCENARIO=011000358700000000202&

How to use UJSTAT (BPC Statistics feature)

The Statistics Framework in SAP Business Planning and Consolidation is a central, generic framework designed to help monitor performance for SAP Business Planning and Consolidation modules (such as SQE, Write-Back, etc).

The Statistics Framework in SAP Business Planning and Consolidation is mainly designed to do performance monitoring of SAP Business Planning and Consolidation modules.

As a prerequisite for performance analysis it is necessary to switch on the performance statistics for the involved Application. The customer can do this in the Application Parameters in Web Administration:

After executing a report/Inputschedule on the application you will find an entry in the statistic table, which can be visualized with transaction UJSTAT:

Performance Statistics Report

The statistics report shows counters like runtime and number of records for data retrieval, write-back and script logic processing (for Data Manager Packages).

For a detailed documentation about the BPC statistics please take a look at the RKT-Material:
https://service.sap.com/~sapidb/011000358700001267532008E
Many of the performance issues in backend might be connected to the data retrieval via OLAP BAPI. This interface supports MDX syntax and is always called when the user requests any kind of calculated data that does not exist physically on the database (such as a BPC Hierarchy node).

From the statistics you can isolate the MDX statement. Open the complete Hierarchy, double click on the event "Mddataset_create_object" and copy the MDX statement from the right Window into a Notepad (as shown in the following screenshot):
MDXTEST:

With the MDX test tool (transaction MDXTEST) it is possible to run the MDX statement directly. Here it is possible to measure the response time, check the result, and modify the statement and to debug the whole execution process directly.

Paste the MDX statement into the window on the right as shown in this screenshot:

It may be necessary to influence the MDX execution. Choose MDX Commands -> Set Debug Flags.

- If you want to disable the OLAP cache, set flag Others -> Do Not Use Cache
- If you want to display BW statistics directly after execution, set flag Others -> Display Statistics Data
- If you want to switch off BWA usage, set flag BIA Server -> Do Not Use BIA Index
BW Statistics:

One way of displaying BW statistics is to use the flag in MDXTEST or access the BW statistics table directly: transaction SE16 -> Enter table RSDDSTAT_OLAP/RSDDSTAT_DM

RSDDSTAT_OLAP: Contains statistics about the BW calculation layer
RSDDSTAT_DM: Contains statistics about the BW Data Manager

In the following screenshot you can see that the BW Data Manager needed 3.5 seconds for selecting 31,013 records and transferring 26,357 records.

All the records have been selected from the e-fact table:

You can find more detailed information about BW statistics under the following link:
http://help.sap.com/saphelp_nw70/helpdata/en/43/e3807a6df402d3e10000000a1553f7/content.html
STAD:
In the ABAP statistics (STAD) you can find out exactly how much time is spent for reading the data from the database (DB req. time (ms)) and for processing on the application server (CPU time (ms)). Refer to the following screenshot:

When you look for STAD entries after refreshing data in the BPC excel client you need to filter on user name BPC_SYSADMIN:

For more information about STAD please take a look at the corresponding help page:
http://help.sap.com/saphelp_sm32/helpdata/de/3d/7b5f3c31727d59e10000000a114084/content.html

Note: All detailed error analysis methods can also be found in the RKT materials: https://service.sap.com/RKT-EPM
Other Performance analysis tools

Fiddler:
http://www.fiddler2.com/fiddler2/

The Fiddler tool is a free analysis tool which needs to be installed on the same PC as the BPC Client. It can be used to track down the HTTP communication between the BPC Client and the .NET Server. This can be useful to troubleshoot problems but also for tracking the amount of time spent for the communication between the BPC Client and the Server vs. the time spent in the client itself.

HTTP Watch:
http://www.httpwatch.com/

This is a similar tool like Fiddler that can help you analyse the HTTP requests sent from the BPC Clients.

Load Runner:

Load Runner is a tool from HP used for executing load tests on many different products. It is the industry standard for executing load and performance tests. It is also the only officially supported product for running load tests on BPC for NetWeaver.

The primary purpose, or goal, of load testing is to observe how a landscape will perform under different scenarios and to verify the stress handling of the system. A lot of useful information can come from these observations including:

- Identifying/Verifying Hardware and sizing needs
- Identifying Tuning or Parameter requirements
- Identifying inefficient Business Processes, or Logic

Effective load testing will identify any of the preceding issues, and allow you to test alternative solutions in a controlled environment before going live or releasing new changes into production. This is the best way to find out how the high user concurrency after the Go-Live will affect the overall performance and where needed changes or re-designs are necessary in order to meet the user’s expectations.

SAP has released two very useful How to Guides on the usage of Load Runner with BPC and even created a custom tool that will help you create BPC scripts for the Load Runner.

How to Guides:

How To Plan and Execute LoadRunner Scenarios for SAP BusinessObjects Planning and Consolidation:

How-To Use the SAP BusinessObjects Planning and Consolidation (BPC) Toolkit for LoadRunner:
http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/b064211f-0b5f-2c10-ce8f-c53836e060d2

Solution Manager Diagnostics:

The Diagnostics tool is an additional option for analysing the performance of BPC NW. With BPC NW 7.0 only work load analysis is possible but with BPC NW 7.5 End to End trace is available.

For more details on Solution Manager Diagnostics check the SAP Note: 1461749.
General Performance Recommendations

General

- In order to support a large number of concurrent users, an appropriate basis configuration must be in place. An SAP NetWeaver dialog instance can support a maximum of 100 dialog processes, and each dialog process consumes a certain amount of memory. Thus, in order to support a higher number of users, multiple SAP dialog instances may be required (including the possibility of several dialog instances on one server).

- Ensure that an appropriate amount of swap space is configured for SAP NetWeaver instances – SAP recommends 2 times physical memory.

- Significant performance improvements were observed when the latest BPC support package was applied. In particular for BPC 7.5, Support Package 6 includes many performance related fixes.

- Based on our experience, the end user machine should not have less than 1GB of physical memory. This is actually quite low and a safer limit would be 2GB.

- Significant performance improvements were observed when database statistics were updated with brttools, including running with the settings “all”, “all_part” and “infocubes”. In addition, you should always keep a few requests in the fact table so that statistics are properly calculated on a representative data set. You can change the number of requests to keep in the fact table with the “compress” process type in the BPC Optimize process chains.

- Oracle DB Only: Contrary to guidance provided in note 830576, we observed significant performance degradation when we increased the size of the Oracle parameter PGA_AGGREGATE_TARGET to a very large size (multiple GB). We saw optimal results with this parameter set to 600 MB in conjunction with high concurrency tests (greater than 100 users at a time).

- Parameter settings in accordance with SAP notes 192658 (Setting parameters for BW systems), 1044441 (Basis parameterization for NW 7.0 BI systems), and 830576 (Parameter recommendations for Oracle 10g) were utilized effectively in our testing.

- In order to accommodate higher numbers of concurrent users, SAP note 316877 (“Maximum number of conversations exceeded”) was applied, in particular – the environment variable CPIC_MAX_CONV was added on the .NET servers, with the value “200”.

- Our tests included a very wide degree of variation of selections on queries and of datasets used in write back. Because of this very high degree of variation, the performance overhead of the OLAP cache mechanism outweighed its potential performance benefits. In our example, performance improved somewhat significantly when we turned off the OLAP cache. It is worth noting that real world scenarios will not likely have nearly the same type of wide degree of variation of selections like the ones utilized in our tests, and therefore the cache may be beneficial for performance. In summary, it is worth monitoring the effects of the OLAP cache in your own environment to determine if it is beneficial for performance in your particular case.

- While the .NET layer can be installed on a 64 bit platform utilizing the “Windows On Windows” environment, the processes continue to be executed in a 32-bit namespace. This limits the amount of addressable memory to no more than 4 GB of RAM.

- BPC Support packages can include performance improvements. The following note has the list of all major performance improvements included in each individual BPC support pack:

  - Note 1479436 - BPC NW 7.0 Collective Note for Performance Improvement
  - Note 1475520 – BPC NW 7.5 Collective Note for Performance Improvement

- BPC NW 7.x Performance Consulting Notes:
  - 1448882 - Necessary/Useful BW Notes for BPC7.x
  - 1457423 - Analyzing EVDRE problems (Back-end)
Hardware and Software settings

Networking: A general recommendation is to have a minimum of a 1 MB bandwidth connection and a latency of less than 200ms

Processes: Communication between the BPC .NET and ABAP AS Tiers is performed via RFC calls. Each RFC call requires a dialog worker process and there is an upper limit of 100 dialog worker processes per ABAP AS instance.

One area to review in poor performing systems is the availability of idle worker processes. If a BPC installation requires more than 100 worker processes, an additional ABAP AS instance will be required.

IIS: As IIS is part of the .Net server, please refer to the .Net server section 5.5

Antivirus firewall: Care should be taken with Antivirus settings as they may block or delay some of the content and communication between the server and the client.

NW settings

Even without the SAP BPC add-on installed on SAP NetWeaver BI, you can configure numerous profile parameters to optimize your ABAP application server (transaction RZ10 – Maintain Profile Parameters). Note that it’s possible to have hundreds of parameters to adjust here. No new parameters have been introduced specifically for SAP BPC, so this section highlights a couple of parameters that are important to keep in mind when optimizing the SAP BPC system:

- The rdisp/max_wprun_time parameter determines the timeout (in seconds) for a dialog process. The recommended value for an SAP NetWeaver BI system is 3,600 seconds [verify]. This value might need to be adjusted for SAP BPC.
- The abap/heap_area_* parameters control how much memory your system allocates to work processes. With 64-bit platforms, individual work processes are no longer limited to just 2GB of memory. See SAP Note 146289 – Parameter Recommendations for 64-bit SAP Kernel for more details.

Check the standard performance documents and notes on NW settings related to performance. In addition it is always helpful to review the ABAP Application Server Memory Help Section:


BPC Settings

BPC Technical Parameters:

- BPC has some nice hidden technical parameters that can be set internally (not exposed in Web Admin)
  - These parameters are different from the already existing SAP NetWeaver parameters, such as RSADMIN, and RZ10 parameters
  - These parameters are stored in ABAP tables: UJ(X)_PARAM where (X) is the BPC module (see UJ Package overview for this value)

Write back Parameters:

Table UJR_PARAM stores parameters for write-back. These values are maintained at the individual application level, and they are:

- PACKAGE_SIZE (default = 40,000): size to break packages into
- RECLEVEL_NR (default = 10): If the number of records being saved is less records than or equal to this value, record based locking will be implemented.
- MULTIPLY_COEF (default = 50): In the situation where record level locking is not being implemented and a sparse data set is being saved, this value tells you the total number of members you can implement record level locking for (i.e. when to swap to using a BETWEEN range in the lock table). This is also a “less than or equal to” check.
- INTERVAL_NR (default = 10): In the situation where record level locking is not being implemented and the data set being saved is NOT sparse, this value tells you which dimensions can be locked by individual values and which dimensions should just be locked using the BETWEEN range). This is also a “less than or equal to” check.
These entries with their default values will actually be inserted into table UJR_PARAM whenever they are not present (e.g. these value do not get inserted when a new application is created – rather, they are always checked during write back, and if a value is not found, the default will be inserted).

**Concurrency Locking:**

All mechanisms that write data into an InfoCube (e.g., manual user-planning in Excel, on the Web, loading and logic from data manager) must execute a concurrency-locking check before saving data to the SAP BPC application. There are no exceptions! You cannot bypass this concurrency-locking check under any circumstances! This is a built in functionality within SAP NetWeaver which cannot be bypassed in a standard setting.

When you lock regions of the InfoCube, you might not want to lock each individual record that you save. When locking records, there is a trade-off between the users' and systems points of view:

- The more granular the locks, the better it is for users because they are less likely to lock unnecessary records (i.e., they only lock the specific records they are updating).
- The more granular the locks, the greater the amount of memory that the system requires to store the locks and the more time it takes to check them.

SAP BPC helps to manage the locking trade-off, and parameters are available to tweak its behavior. You can change the parameters in transaction UJR0 (mention above in Write Back Parameters Section) in your SAP system.

In addition to BPC Specific Lock settings, there are also BW Specific Lock settings for the InfoCube itself. Transaction RSPLSE allows you to manage these settings for each InfoCube. For instance you can set the Lock Relevant characteristics (or BPC Dimensions) as you need. Refer to the Help site for further details: http://help.sap.com/saphelp_nw70ehp1/helpdata/en/44/240f7182974face10000000a155369/frameset.htm

**Shared Query Engine Parameters:**

The SQE is the central module for reading all of the data from an SAP BPC application. (Ex: SAP BPC Excel client, the Web, and the data manager). Technically, the SQE in SAP BPC 7.x NW consists of the following four sub-modules. These sub-modules are completely invisible to the end users.

- **RSDRI query.** RSDRI is the function group in SAP NetWeaver BI that deals with access to InfoProviders. In the SQE, an RSDRI query means calling the internal SAP NetWeaver BI functions to read data from an InfoProvider (Fastest).
• **MDX query.** SAP NetWeaver BI exposes OLAP BAPIs to access data from an InfoProvider through MDX. The MDX query method in SAP BPC generates the MDX statements required to retrieve data with the SAP BPC application logic applied. It is always used when there are dimension member formulas, measure formulas, or hierarchy parent values to be retrieved.

• **Axis query.** To request a contiguous region of data (e.g., an array of data to be displayed in a grid in SAP BPC Excel), the system uses the Axis query, which contains internal logic that determines (among other things) whether an RSDRI query or an MDX query will do the actual retrieval of data. Then, the Axis query calls the appropriate method. In some cases, the Axis query even breaks one request into multiple queries, each of which might call an RSDRI query or an MDX query, and then merges the results.

• **Cell query.** In direct contrast to an Axis query, this method specifically deals with requests for *sparse* data. A Cell query is required because SAP BPC supports users building highly formatted reports by retrieving several different individual cells from an SAP BPC application.

**Note:** As of BPC 7.5 NW and BPC 7.0 SP8 we don’t use Cell queries. Check SAP Note: 1329178.

**BPC Settings Notes:**

Apart from the BPC settings mentioned above there are also some other parameters settings mentioned in these notes:

- SAP Note 1421710 – By Pass MAX AMOUNT field check
- SAP Note 1395793 – Enhance performance of UJP Classes

Lot of other important BPC notes can be found in the Related Content section.

**.Net server**

Within the .NET application server, there are numerous optimization techniques available, mostly in the area of optimizing connectivity. There are two major techniques to consider: Number of connections, processes, and threads for the .NET application and Web server. This technique is used for setting Web server parameters, such as threads and connections. This technique is used for persisting connections for COM+ objects. These settings ensure that your system performs reliably without significant cost to concurrency and performance response times.

Number of Connections, Processes, and Threads for the .NET Application and Web Server: To ensure that users can gain access to the .NET application and Web server when numerous concurrent requests exist, set the number of threads and processes in the Machine.Config file correctly. The Machine.Config file is located in the `<WINSYSIR>\Microsoft.NET\Framework\v1.1.4322\Config\` directory. (For more details about defining these parameters, see the Microsoft knowledgebase article at http://support.microsoft.com/kb/821268.)

The configuration options include:

- **maxWorkerThreads.** The maximum number of threads for traditional unrestricted threads, the default is 20, but the recommended default value for SAP BPC is 100.

- **maxIOThreads.** The maximum number of threads for I/O objects, such as a stream or a pipe. The default is 20, but the recommended default value for SAP BPC is 100.

- **minFreeThreads.** The minimum number of free threads required to execute a request. If a sufficient number of threads aren’t available, the request remains queued and periodic checks for thread availability continue until the required number of threads is present. The default value is 8, but the recommended default value is calculated as “88 x # of CPUs on server.”

- **minLocalRequestFreeThreads.** The number of free threads that Microsoft’s Web application framework, ASP.NET, keeps available to allow new local requests to execute. (ASP.NET is the successor to Microsoft’s Active Server Pages [ASP] technology.) The intent is to avoid a possible deadlock with recursive reentry into the Web server. The recommended default value is calculated as “76 x # of CPUs on server.”
appRequestQueueLimit. The maximum number of requests that ASP.NET queues for the SAP BPC application. ASP.NET queues requests when there are insufficient free threads to process the requests. The recommended default value is 500.

collectionManagement. If deploying where the Web servers do not reside on the same servers as the application servers, adjust these HTTP TCP/IP connections settings. This controls the maximum number of connections allowed to the server.

executionTimeout. The maximum number of seconds a request can execute before ASP.NET automatically shuts it down. The default is 90 seconds. However, for ABAP dialogs, the default timeout is set to 300, so you should logically set executionTimeout to match your ABAP timeout.

One more configuration to allow users to request/query large amounts of data is to edit the metabase.xml file located at C:\Windows\system32\inetsrv\MetaBase.xml. Set the following parameters to the stated values:

- ASPBUFFERINGLIMIT = 10485760
- MAXREQUESTENTITYALLOWED = 10485760

Load Balancing
The higher the user load, the higher the number of worker processes you will need. If a BPC installation requires more than 100 worker processes, an additional ABAP AS instance will be required.

All client communication for BPC generally falls into two categories.

- Synchronous SOAP Requests – In Synchronous SOAP Requests, the client requests an operation in a SOAP message. The SOAP message is sent to the BPC .NET Server over HTTP or HTTPS and the BPC .NET Server responds to the request with the resulting information.
  - For example, the response may include data sets for a report or details on the success or failure of a data load.
  - The BPC for Office (Excel, Word, PowerPoint) clients use Synchronous SOAP Requests.

- Asynchronous SOAP Requests – In Asynchronous SOAP Requests, the client still requests an operation in a SOAP message that is delivered to the BPC .NET tier via HTTP/HTTPS, but in this case the results are not immediately returned to the client. Instead, the status for this request is entered into the MSMQ with a unique identifier, and the unique identifier is returned to the client to periodically check the status.
  - Some example asynchronous operations are processing dimensions, or processing an application

The BPC Admin Client uses asynchronous operations.

The key to ensuring that Load Balancing works correctly with BPC 7.x for NetWeaver is to verify that:

- Asynchronous communications always get directed to the same .NET BPC Server
- Limits the types of Load Balancing that can be used, or impacts the Landscape Design
- All .NET BPC servers store their status in the same MSMQ (Requires SP1, Patch2)
- Concern – Storing all status messages in the same MSMQ introduces a Single Point of Failure

Note: Some load balancers offer advanced load balancing techniques where cookies are injected in the HTTP response. The purpose of this technique is to stick a client to the specific application server. This Load Balancing technique does not work with BPC because the BPC clients are not web browsers and have not been written to honor cookies.

This forces us to use lower level technologies (for example, client IP address) if we decide to use a load balancing scheme that relies on clients sticking to the same BPC .NET server.

Important Information: One latest finding is about the distribution of load across multiple NW application servers, when using Logon Groups. As BPC uses extended RFC connection for the communication between .Net server and ABAP server, sometimes the load is not properly distributed across the servers. As of NW release 6.x, you can use a new load balancing procedure for RFC logons. In transaction SMLG, you must select the "Ext. RFC-enabled" field for the maintenance of the group attributes for this. For more details check the SAP Note: 593058
BW Accelerator

You can use the SAP NetWeaver BW Accelerator (BWA), although it is not mandatory, for SAP BPC Applications.

You can add this hardware appliance to SAP NetWeaver in order to provide consistently fast response times, even as data volumes and the number of users increase. SAP NetWeaver BW Accelerator performance improvements can extend to SAP BPC, too. It is important to note that BW Accelerator will improve Read performance only, so if you have a very high number of Writes you will not see the same benefit as when reading data from a BW Accelerator index.

We have also seen the BWA significantly reducing the load on the Database when it comes to the read operations, which can have a positive impact on the overall system performance. This load decrease can also mean a significant reduction in the hardware sizing needs when data is accessed via BWA and not directly from the Database. This benefit might not be as significant if the particular BPC implementation is relying on many write operations consistently adding large data amounts.

The overall improvement seen by using the BWA on read performance relating to Reports, Input Schedules, Script Logic, etc. are depended on the actual query executed as part of that operation and the data volume in the cube.

Important to know when BPC Cubes are indexed on the BWA:

The Automatic Rollup needs to be turned on for the particular BPC Cube otherwise a regular job needs to be scheduled to rollup all closed requests to the BWA.

Having a BPC Cube on the BWA does not mean that no additional performance maintenance is needed on this cube. These cubes should still have a daily job running to update the statistics of the cube, as with BPC cubes there will almost always be an open request which is not yet on the BWA. This can still have a significant performance impact, if the statistics on the cube are not up to date. In scenarios where a lot of data is being written in a short time period with many requests generated, the BWA rollup might not be able to keep up with rolling up the requests as fast as new requests are being added. This will add more data to the cube that is not yet on the BWA, which will require updated statistics for optimal access times. A regular job compressing the requests to keep the volume of the cube to a minimum is also recommended.

Modeling

The design of an application set has a significant impact on the performance of the Business Planning and Consolidation 7.X for NetWeaver BPC application. The purpose of this section is to highlight some of the design aspects that can influence the performance of a specific application as well as providing general guidelines and design best practices.

The following design factors have been identified to have impact on the overall performance of SAP Business Objects Planning and Consolidation 7.X for NetWeaver (BPC) and should be carefully considered during the initial design phase of the implementation.

1. Application sets
   - The impact of adding additional applications to an application-set.

2. Applications
   - The number of applications in an application-set.
   - The difference between having a large application instead of having multiple smaller applications.

3. Dimensions
   - The number of dimension members per application.
- The impact of the number of properties on performance.
- The impact of the size of the name and the size of the properties on performance.
- The impact of the number of dimensions hierarchies
- How many levels can a hierarchy have before it starts impacting performance

4. Security
   1. The impact of having a secure dimension in the design.
   2. The impact of having a wide access profile vs. a specific access profile.

5. Work status
   3. The impact of using work status on performance.

Note: The findings in this section are only relevant to SAP Business Objects Planning and Consolidation 7.5 for NetWeaver (BPC) SP4 and below. Significant performance improvements have already been identified by the BPC development team and are planned to be delivered in future Support Packs.

BPC users can experience performance issues during logon, or when they are running reports or input schedules. The same applies when BPC Administrators perform maintenance in an application such as adding a new property or adding new dimension members.

To understand the reasons for any performance issues we need to understand how BPC works, and the impact of certain design decision on the performance.

The BPC Logon Process

The high level logon process for BPC is as follows:

1. When a user logs on to BPC for the first time, the user’s client machine (PC) will send a request to the BPC .NET Application Server in order to retrieve the application relevant information.
2. The BPC .NET Server then converts the request into an appropriate RFC call to the BPC ABAP Application Server.
3. The ABAP server locates the details in the BW File Storage (UJFS), and converts the data into an xml format which it sends back to client. Note: As of BPC 7.5 NW SP6, we have csv file format instead of xml format.
4. These xml/CSV files will be downloaded/created on the user’s client machine under the “My Documents” folder.
5. For any subsequent logons BPC will identify any dimensions that have been changed since the previous logon and will reload new XML/CSV files for only those dimensions. The only exception is the measure.xml file that is reloaded with every logon.
6. BPC loads the local xml/csv files into memory that enables users to work “stateless”, that is without being connected the BPC application server.

The dimensions files are being downloaded/created for each application they belong to in the Application Set. The following picture shows an example of an xml file (Note the filename has the Application Name as a prefix):

<table>
<thead>
<tr>
<th>Filename</th>
<th>Size</th>
<th>Type</th>
<th>Date - Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIENTE.xml</td>
<td>16KB</td>
<td>XML Document</td>
<td>21/07/2010 10:03 a.m.</td>
</tr>
<tr>
<td>DirectCost_02~ATTIVITA.xml</td>
<td>10,252KB</td>
<td>XML Document</td>
<td>21/07/2010 10:03 a.m.</td>
</tr>
<tr>
<td>DirectCost_02~MEASURE.xml</td>
<td>1KB</td>
<td>XML Document</td>
<td>21/07/2010 11:07 a.m.</td>
</tr>
<tr>
<td>DirectCost_02~TOPDATA.xml</td>
<td>26KB</td>
<td>XML Document</td>
<td>21/07/2010 10:03 a.m.</td>
</tr>
<tr>
<td>DirectCost_02~UC.xml</td>
<td>0KB</td>
<td>XML Document</td>
<td>21/07/2010 10:03 a.m.</td>
</tr>
</tbody>
</table>

The application specific reports and input templates in the eExcel folder will only be downloaded when you logon to BPC for Excel for that specific application. This process may take a few seconds longer when you access the application for the first time, depending on the size of the templates in this folder.
Every time you logon to a new application, the size of the XML/CSV file in the client My Documents folder will increase slightly due to the downloading of the application specific reports and input templates.

The high level process for maintaining a dimension is as follows:

1. When a BPC administrator processes a dimension, he/she can either save the dimension, or process the dimension.
2. The request is sent to the .NET server, which in turn converts the request into an RFC call on the ABAP Application Server where the actual work is being done.
3. BPC will retrieve the data from Excel and store it to the BW File Storage.
4. Processing can either be from the dimension's membersheet, or from the previously saved file stored in the BW File Storage. Processing from the BW File Storage is significantly faster since the data is already in the required format.
5. During the processing, each dimension member is validated to prevent duplicate members and to ensure that the assignment to parent members are consistent for all hierarchies.
6. The ABAP server will then convert the file into an xml/csv format and send it back to client through the .NET Server.
7. These xml/csv files are then downloaded/created on the user's client machine My Documents folder.

Overall Findings: (Based on BPC 7.5 NW SP4)

BPC for Admin interface did not have any significant performance issues because of this login process. However, it is impacted by various factors such as the number of concurrent users, network activity, server activity etc.

BPC for Excel however, was significantly impacted by all the test factors identified. The following diagram shows the detailed analysis for logging on to BPC for Excel for a test scenario with 130,000 members in 5 different applications.

The following factors were identified that contribute to the increase in the logon time:
1. The size of the XML file.
2. The ABAP server needs to convert BW data into xml and send back to client (first logon or changed dimensions)
3. BPC transfers and loads the local xml files into memory.
4. The same dimension will be loaded n times if it belongs to n applications

- The XML File Size:
  The logon time is directly related to the size of the XML it has to download to the client.
  The size of the file for a specific dimension is impacted by the number of members, the number of properties, the field length of the properties and the number of hierarchies. According to our analysis the size of the XML file can be calculated according to the following formula:
  \[ \text{Constant} \times (\# \text{members} \times (1 + \# \text{properties} \times \# \text{hierarchies})) \].

- Manage Application Info:
  Dimensions are being loaded for each application they belong to in the application-set. This is time consuming as all the dimensions have to be reloaded multiple times, even though when you logon to BPC for Excel you are only working in one application at a specific point in time.

- Loading Dim Info:
  Currently BPC transfers every node of an XML file and loads them into temp tables in memory. This could potentially have long runtimes for larger dimensions and will require very high CPU utilization. For many concurrent users, converting and sending back multiple xml responses can create bottlenecks on the server and a potential increase in runtimes.

Recommendations:

The above findings are based on BPC 7.5 NW SP4 and since then there were lot of improvements like changing the xml format to CSV format to bring the overall file size down, downloading Application Info only for that Application where the user is logged and loading only required Dimension info into memory with each SP.

So the recommendation is to move to the latest SP of BPC 7.5 NW to take benefits of performance improvements. Even the customers who are on BPC 7.0 NW are recommended to move to BPC 7.5 NW as these improvements are only available here.

The following recommendations can also help to address performance issues:
- Design your BPC application with performance in mind.
- Consider having multiple smaller applications such as Sales Planning, Production Planning, and Cost Centre Planning etc... With a summarized Finance Cube instead of one large overall application.
- Limit the number of dimensions per application to no more than 15.
- User access should be limited to required applications and dimensions.
- For very large dimensions consider using an "aggregated or group" level member that is still relevant to planning. Actual data is often at a more granular level than planning, and could be summarized during the loading of the data into BPC.
- Restrict the number of hierarchies as it impacts the size of the XML file as well as the time that it takes to process a dimension.
- Restrict the number of properties you add to a dimension.
- Delete unnecessary properties.
- Restrict the length of each property.
EvDRE ("Data Range Expansion") is a powerful and flexible WebExcel function that can be used to generate different kinds of SAP BPC-based reports and schedules in Excel, all with one easy to use interface.

EvDRE combines and extends the functionality of several other "EV" functions like EvGet, EvSnd, EvExp, EvNex, etc. Historically, EvGet was the original function that was used to retrieve data from the database. To increase performance, the enhanced function EVDRE was developed. While the older functions are still supported for backwards-compatibility of workbooks that some users may not want to re-design, EvDRE should be used in the greatest majority of situations and as a best practice.

The Shared Query Engine (SQE) automatically decides the most efficient way to access the database to retrieve the requested data. In most cases this will greatly improve the performance and scalability of reports and schedules, because whenever possible, EvDRE reads base-level data directly from the database. This will have the effect of reducing the workload on the OLAP engine. In cases in which the use of SQL queries is not possible it interrogates the OLAP cube using very efficient, dynamically-optimized MDX queries.

Best Practices When Using EvDRE Function

1. **Be careful with multiple dimensions in Row and ColKeyRange:**
   
   Performance degrades rapidly in a large application if you have more than 1x2 dimensions in rows x columns. Reports with two dimensions in both the row and column become very slow when the concurrency level in the system is medium or high. Reports with more than 2 x 2 dimensions in the row and column may not finish, even if you refresh them during peak concurrency. As a result, BPC users may experience poor performance. Therefore, it is a good idea to reduce the number of row and column dimensions using multiple EvDRE functions. Nested report can be created using one or more row or column expansions but avoid having more than 7 dimensions for a nested expansion.

2. **Sum base members via the PagekeyRange:**
   
   When a report contains only base members (members at which data input is done), use EvDRE because this is the quickest method.

3. **Remove parent members from the key range when using Excel formulas for parent calculations:**
   
   If you have a report with, for example, Accounts in the rows, and you have all base members in your PagekeyRange, and some calculated members in the RowKeyRange, EvDRE creates an MDX query for these calculated members. However, if you create a formula, for example, a SUM formula, and do not remove the calculated members from the RowKeyRange, you will not benefit from the SUM formula, because EvDRE will generate a MDX query anyway for the parent member.

4. **Limit the Number of EvDRE functions in one sheet:**
   
   It is possible to use more than one EvDRE in a worksheet, but the performance should always be tested very carefully. There are however, some specific situations where multiple EvDRE functions can increase performance, because instead of building one EvDRE with two column dimensions and one row dimension, you can build two EvDREs with one row dimension and one column dimension. This makes the query easier and in these specific situations it is faster.

5. **Avoid overlapping ranges when refreshing:**
   
   Overlapping ranges causes the last executed EvDRE to overwrite the data of the EvDRE functions that were executed before and have the same overlapping data range. Overlapping ranges can be used for sending data (although not recommended, but sometimes necessary), but never for refreshing.

6. **Dimension formula:**
   
   Avoid retrieving MDX calculated formulas as much as possible and take advantage of script logic.
7. **Avoid the CellKeyRange option, if possible:**

When you have a complex report where you need to retrieve, for example, accounts that are not the same for the whole column or the whole row, it becomes difficult to create a report without using the CellKeyRange. This function allows you to define individual members for each cell in the DataRange of EvDRE. But the query is less efficient in comparison to a “normal” EvDRE where you only use row and ColKeyRanges.

8. **Use data dump sheet & Excel-based calculations:**

In some cases, for performance reasons it may be an option to set up one big data dump sheet that is generated by one or more EvDRE functions. For example, if you have a workbook with ten sheets and all the sheets retrieve data from the same accounts, but for different entities per sheet. In this situation, instead of making ten EvDRE’s in the whole workbook, you can dump all your data in one hidden sheet and build your reports with links to the data dump sheet.

Also, setting up easy KPIs like “Account1/Account2” via Excel formulas can increase the performance of your reports as it is faster than MDX queries. However, when building a report this way, keep in mind to choose a unified approach throughout the whole report, because maintaining these data dump reports is harder then maintaining a single report that retrieves its own data. In addition, you should be aware that excel formulas are local to the report and maintenance can become a factor when dealing with a large number of reports.

9. **Note handling of text within EvDRE datarange:**

If you want to present text within an EvDRE DataRange, use: =”[yourtext]” because otherwise the text disappears after a refresh. EvDRE overrules all cells that do not have formulas in them. So if you enter text in a cell, it is empty after refreshing and you have to enter everything again.

10. **Avoid EvDRE and named ranges:**

The use of named ranges generally decreases performance. Excel has to do a lot of calculations to convert the named range to a real cell reference. The use of named ranges in combination with EvRNG or EvDRE functions may work with a few workarounds, but is not recommended.

11. **Turn off the “automatic calculation before refresh” in big workbooks:**

For workbooks with a lot of sheets that contain multiple EvDRE’s, it is a good practice to create a custom refresh button with VBA code behind it. The code looks like the following:

   Application.Run ("MNU_eAnalyze_Refresh")
   Application.Calculation xlCalculationAutomatic

This code turns the automatic Excel calculation off before refreshing and enables it after the refresh. You should use this approach if you have workbooks that contain references to named ranges in the EvDRE function. If not, you may get “unable to get the text property of the range class” error during refresh. Although the EvDRE looks OK before and after the refresh, it still does not refresh the right data.

12. **Avoid using different data retrieval functions within one workbook (EvDRE & EvGET):**

When (re)building reports, you can use both EvDRE and EvGTS functions within one workbook. For example, if you have a large report that retrieves finance data from the Finance application using EvDRE and rates data from the Rate application using the EvGTS function.

In some cases performance may be better when you use a combination of both these functions. However, this does not mean that it is a good idea to combine EvDRE and EvGET in all your reports. Since BPC needs to start up two different retrieve processes, there is overhead time added to the query. We recommend you avoid the use of both these functions within one workbook for two reasons:

- To prevent possible performance issues. All reports using mixed functions should be tested in detail.
• To avoid maintenance. The more you have mixed functions, the more difficult it will be to maintain your report in the future, because first you have to investigate which data is retrieved in which way before you can really change your report.

13. Using multiple EVDREs:

The following methods to improve performance are recommended when building multiple EVDRE Templates. Based on analysis and experiences during customers' EVDRE templates testing, we notice data retrieval is faster for multiple EVDREs than data retrieval for one EVDRE if the server is close to workstations (Client PCs).

While opening an EvDRE report that does not have the Expand on workbook open or the Refresh workbook on workbook open option in the workbook options, you must refresh or expand it first.

If you retrieve several members that contain many dimensions in the one axis, this may slow down performance.

Multiple EvDREs in a worksheet may overlap each other. Check the member list when creating a worksheet with multiple EvDREs.

Formatting Range:

- Format Ranges will be used unless it is necessary.
- Adjust Cell and font sizes

Asymmetric Queries

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 performance should still be 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.

Logic Best Practice Guidelines
This section is presenting best practices for Script and will mainly cover the following areas

- Options to perform calculations
- Usage considerations
- LookUp

The findings, comments and recommendations are based on SAP BPC 7.5 SP03. Remember that with each new release more enhancements and improvements are introduced into the product and some of the recommendations may need to be adjusted.
Available options:
There are various options available in BPC to perform calculations, namely:

- On the fly calculations
  - Dimension Formulas (MDX)

- Stored results calculations
  - Script Logic with "SQL" Syntax
  - Script Logic with "MDX" Syntax
  - Business Rules
  - Allocation Engine
  - BADI

When setting up calculations in BPC, it is important to determine the choice of which option to use based on performance considerations and not based on the effort required to setup each of the options.

It is very difficult to precisely assess the relative performance of each of these options, as they cannot be used for the same purpose and each one has specific usage case.

**Dimension Formulas**
The Dimension Formula calculations are performed on the fly and the results of these calculations are not stored. This type of calculation should in general be avoided, as it the calculations are quite slow and they need to be executed on each report. This type of calculation is however useful to calculate KPI, that need to be calculated on node levels (and not only on base levels, like for the stored calculations)

**Business Rules**
The Business Rules should be used whenever possible, as they have been optimized and they can be maintained through parameter tables (no need to write any script here). For cases like Currency Conversion or Consolidation, their usage is highly recommended. Other Business Rules like "Account Transformation” can be used to perform simple calculations (like “addition/subtraction”) and when possible, should be used in place of SQL/MDX Script Logic

**Allocation Engine**
The allocation Engine can be used to allocate figures (distribute existing figures over a structure) and it should be used for that purpose over SQL/MDX script logic whenever possible. It has also been observed that using the allocation for simple calculations like additions or multiplications gave better performance than using the MDX/SQL Script Logic

**MDX/SQL Script Logic**
It is important to notice that MDX/SQL Script Logic should only be used as the last option, as its performance can be worse than the other options. It is also important to notice that, contrary to the Microsoft Release, the MDX Syntax is not always slower than the SQL Syntax. It was observed that in some examples, the MDX type syntax could perform faster than or at least as fast as the SQL type syntax, especially when some parent members are used by the calculation.
If you for example want to copy the value of a parent account (Acc10000) that is the parent of large number of account Acc1 to Acc9999 to Acc10001 (we assume that there is a parent property that indicated the parent)

MDX Syntax:

\[[Account][#Acc10001]=[Account].[Acc10000]\]

SQL Syntax:

\*WHEN Account.parent
\*IS Acc10000
\*REC(Account=Acc1001)
\*ENDWHEN

BAdI:

A well written BAdI can probably outperform any other calculation method, as it can be configured to operate only on the exact data set required and can then perform all relevant operations before writing it back. Most of the other methods mentioned in this section will have multiple cycles of data reading / writing and that will lead to longer execution times.

Two important aspects are:

1. A badly designed BAdI can lead to poor performance, so the design and the algorithm used to apply the calculations here is important. There is a real design process involved when writing a BAdI, it's not just writing a script with a list of calculations.
2. There might be a substantial effort required to develop (and maintain) a BAdI and this might not always be worth the effort. So not every calculation needs to be written as a BAdI, but the potential performance gain needs to be assessed against the effort it takes to develop it.

There are, however, situations where the usage of a BAdI is mandatory. For example, performing calculations on data before this data has been written back to the database. This can be achieved with the Write-back BAdI and can for example be used to allocate parent values to children.

Using Script Logic:

When Script Logic still needs to be used, it is important to take into account the following points to make it the most performing:

1. Control the scope
2. Limit the number of *COMMIT statements
3. Limit the content of Default Logic

Control the scope:

The scope of a calculation will define which source records will be read to perform the calculations. It is important to only include the records that are needed in the calculation in the source region.

Here is an example of what should not be done:

\*XDIM_MEMBERS ACCOUNT ACC1, ACC2.ACC3
\*WHEN ACCOUNT
\*IS “ACC1”
\*REC (...)
\*ENDWHEN
\*COMMIT
In this example, “ACC2” and “ACC3” have been included in the scope and the system could potentially perform the “WHEN / *IS test on these records, although they will never be used for any calculations. You have two negative effects here:

- The size of the source region query is increased
- Processing is slowed (more records need to be tested)

The example above is of course not so significant, as there are only very few members in the "*XDIM_MEMBERSET statement of the account dimension. But imagine when the same thing is done for multiple dimensions with a large number of members (and a high number of unused ones in the calculations).

It is also important to understand how the system sets the Initial Scope. There are different cases here:

1. On data sending (default logic), the scope is derived from the data that was sent. The system will create a list of all the unique members per dimension and the cross product of all these members is the scope.
2. Using “Data Manager”, the system will take the member selections for the dimensions that have user selections and all non-calculated members for the other dimensions.

The scope can then be further manipulated in the logic script during execution (Execution Scope):

- Overwriting the scope (*XDIM_MEMBERSET)
- Extending the scope (*XDIM_ADDMEMBERSET)
- Filtering the scope (*XDIM_FILTER)

Note that the scope will be reset to the original scope after a *COMMIT instruction

Limit the number of *COMMIT statements:

Each time a *COMMIT instruction is used, the system will send back the record to the database. Technically, the system will first need to query the destination to compare it with the calculation results and then only send back the delta values. Once the records have been sent back, the system will query the source region again.

All these operations are quite resource intensive and time consuming and should therefore be kept to a minimum. It’s recommended to limit the number of *COMMIT as much as possible

Limit the content of Default Logic:

The Default logic is executed on each data send, so it is highly recommended to limit its size and only include what is necessary. A huge Default logic file can have a negative impact on the user experience (send takes a long time to complete) and on the scalability of the system (will limit the number of concurrent users that can send data).

The basic rule would be to only include in the Default logic what is absolutely necessary and to add more if the performance / response time allows it.

A long script can also take a longer time to validate. The script is validated each time it gets executed and this process can slow down the execution of the script.

Log File:

The log file is the starting point of any investigation concerning the behaviour (or misbehaviour) of any logic.

While the log file is critical to fixing logic, it is also a best practice to review the log file when your logic runs without errors. This will ensure you understand the performance profile of the logic, and might point out areas for improvement.

Recommendations:
- Minimize Use of MDX Dimension Member Formulas AS much as possible
• Measure Formulas Perform Better than Member Formulas
• Any time you specify a Member with an associated formula, it will ALWAYS call the MDX engine causing additional Overhead.
• When Possible, use ‘Calc and Store’ (Default Logic) vs. ‘Calc on the Fly’ (MDX Formulas).

TIP: How to force the RSDRI_QUERY when reading data for Script Logic?
  • The SQE should automatically determine WHAT query type would be best performing based on your query.
  • In order to enforce an RSDRI_QUERY read you need to:
    • Only select Base Level Members
    • Do not include any MDX in your logic (base member formulas, measure formulas, parent node requests, etc)

Shared Query Engine – Decision Tree:

IF Query is REQUESTING specific Set of Cells → CELL QUERY (Note: As of BPC 7.5 NW and BPC 7.0 NW SP8 Cell Query is not used)
IF Query is REQUESTING an AXIS of data CALL → AXIS QUERY
  IF Query contains NO MEMBER FORMULAS
  IF Query contains ONLY BASE LEVEL MEMBERS
  IF Queries Measure type IS SIGNEDDATA OR PERIODIC
  THEN Call RSDRI QUERY
ELSE Continue with AXIS QUERY
Related Content

SAP Notes:
- BPC NW 7.0 Collective Note for Performance Improvement
- BPC NW 7.5 Collective Note for Performance Improvement
- Script Logic Performance Note
- Setting parameters for BW systems
- Basis parameterization for NW 7.0 BI systems
- Parameter recommendations for Oracle 10g
- Maximum number of conversations exceeded
- Parameter Recommendations for 64-bit SAP Kernel
- Solution Manager Diagnostics with BPC
- Necessary/Useful BW Notes for BPC 7.x
- Improving proper load distribution across multiple application servers
- Issue with Lite optimization note
- Analyzing EVDRE problems (Back-end)

How-To Guides:
- General EPM How-To Guides link
- BPC 7.X for Net Weaver Performance Considerations for BADI UJ_CUSTOM_LOGIC
- BPC 7.X for NetWeaver: Performance Improvements for BADI UJ_CUSTOM_LOGIC Using Buffer Objects
- How To Plan and Execute LoadRunner Scenarios for SAP BusinessObjects Planning and Consolidation:
- How-To Use the SAP BusinessObjects Planning and Consolidation (BPC) Toolkit for LoadRunner

Help Documents
- General RKT link:
  http://service.sap.com/rkt
- On the left hand side, navigate to SAP Ramp-Up Knowledge
- Transfer -> SAP BusinessObjects EPM Solutions -> SAP BO PC 7.5, version for SAP NetWeaver
- BPC Installation Guides: https://service.sap.com/instguidescpm-bpc
- BPC Sizing Guides: https://service.sap.com/sizing
- SAP High Availability
- Information about Standard NW Performance Traces
End to End trace instruction guide
BPC Statistics RKT material
ABAP Statistics (STAD)
BW Statistics
ABAP Application Server Memory Management
BW Specific Lock settings
SAP Buffers
SAP Number Range Buffering
SAP Database Support (All Databases supported by SAP and their Help Site)
Oracle Database Guide
Oracle Statistics

Other Documents
Microsoft knowledgebase article - Defining parameters for the .NET application and web server
Fiddler Tool
HTTP Watch Tool: http://www.httpwatch.com/

SDN Blog Wiki
The following wiki compiles blogs specifically targeting BPC topics
http://wiki.sdn.sap.com/wiki/display/CPM/BPC+Blogs

SAP BPC Help
http://help.sap.com has an extensive library of help documentation for all SAP products.
Help documentation for BPC 75 NW can be found at the following location:
For more information, visit the Enterprise Performance Management homepage.