

Backend Performance Improvement Measures in BI



SAP BI 7.0. For more information, visit the [EDW Homepage](#).

Summary

This article discusses 3 methods for improving the performance for BI.

Author: Kiran Gasmar

Company: Appexus Software Solutions

Created on: 17 September 2010

Author Bio

Kiran Gasmar is working as SAP Technology Consultant with Appexus Software Solutions (P) Ltd. He has an experience of over 1+ year in SAP ABAP and 6 months in SAP BI.

Table of Contents

Introduction	3
Aggregates.....	3
Steps to build aggregates:	3
Read Mode	6
Step to set the read mode:-	7
OLAP Cache.....	8
Related Content.....	9
Disclaimer and Liability Notice.....	10

Introduction

The performance aspects of a data warehouse project should be of primary consideration, as performance is directly related to the degree of user acceptance of the solution, as well as the overall effectiveness of its analytical capabilities. If it's time for some fine tuning on your BI engine, you can focus your effort on just a few areas for the greatest impact. These include:

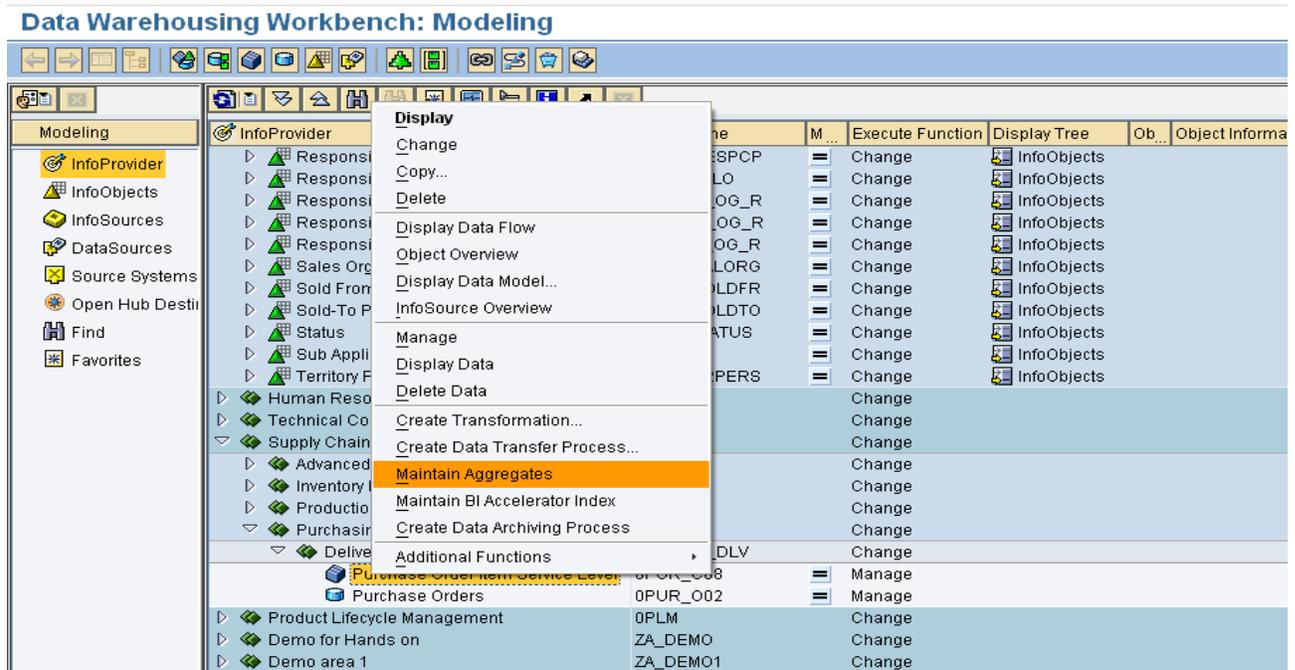
- Aggregates
- Read Mode
- OLAP Cache

Aggregates

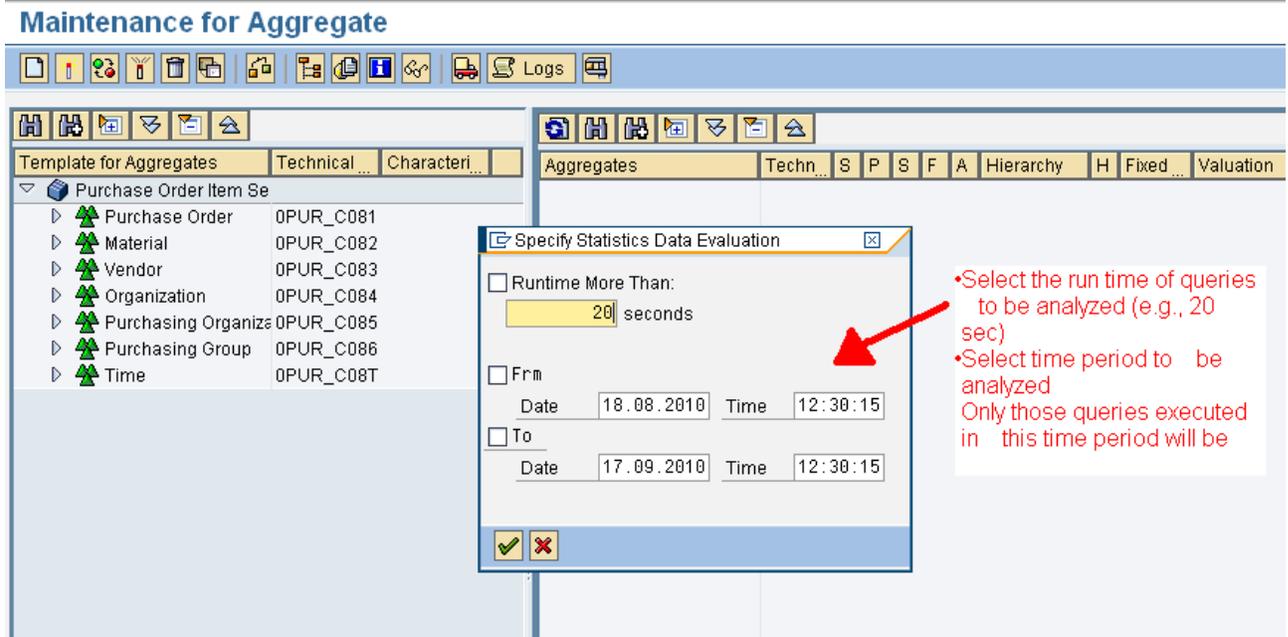
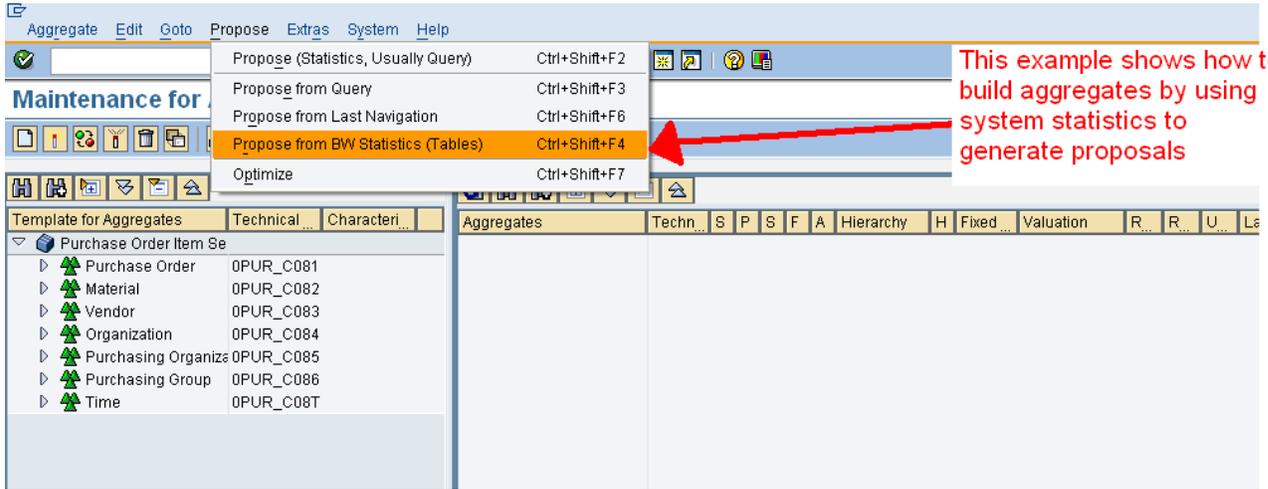
Aggregates are mini-cubes built on top of InfoCubes. An aggregate is a highly summarized version of the data in the underlying InfoCube. During query execution, if applicable, an aggregate is hit before the InfoCube, and this serves to improve performance. It is important to note that unlike multiproviders, aggregates actually store the aggregated data in the database. A multiprovider enables users to run reports off more than one InfoProvider, by combining the information that resides in them in a “union” operation. When a query is run on a multiprovider, the processor spawns parallel sub-queries to the various InfoProviders, and this in turn expedites query execution. Users of aggregates (especially first-timers), tend to think of it as a silver bullet for the “query is taking forever to run” problem. However, the sole rationale for using aggregates shouldn't be that a query is taking too long to run. Unless the planning and design process is properly followed, aggregates may degrade performance rather than improve it. An example case of this is frequent change to master data attributes. Keeping these updates to your InfoCube synchronized with the corresponding aggregate(s) involves frequent “change runs”, which take longer and longer as the number of aggregates increases. The interface to build the summary tables (aggregates) is intuitive and easy to master, but few are taking real advantage of them.

Steps to build aggregates:

Drill down to the InfoCube on which you want to create an aggregate. Highlight this InfoCube; then from the context menu (right mouse click), select “Maintain Aggregates”.



Displays the menu path that takes you to aggregates creation



We can also create proposals from the Query user statistics. To make this work, a representative number of queries must be executed to gather the statistics to optimize from.

We can also create proposals for aggregates based on individual queries that are performing

Reduce the overlapping proposals by optimizing them. This may reduce the proposals from 99 to less than a dozen

Object Name	Long Description	Techn	S	P	S	F	A	Hierarchy	H	Fixed	Valuation	R	R	U	Lastused	Las
STAT 20											****				3	
STAT 21											*				1	
STAT 22											*				1	
STAT 23											*				1	
STAT 24											*				1	
STAT 25											*				1	
STAT 26											***				2	
STAT 27											*				1	
STAT 28											****				3	
STAT 29											*****				10	
STAT 30											****				3	
STAT 31											*				1	

High valuation and high usage is what we are looking for. This indicates high reduction of records in aggregate and high benefits to users.

Warning: When using 3rd party query tools and ODBC to query directly into the DSO, you are bypassing the OLAP Processor. Therefore, you cannot accurately performance tune the system using aggregates (statistics), nor will the 3rd party tool benefit from aggregates.

Before they can be used in reporting, aggregates must be active and filled with data. If you are prevented from using one or more aggregates, there are three options available:

- Switch off the aggregate. The aggregate is still filled with data, but is not used by the OLAP processor. You still need to roll up new data.
- Deactivate the aggregate. The aggregate data is deleted, but the aggregate definition remains.
- Delete the aggregate. The aggregate data and the definition of the aggregate are deleted.

An InfoCube can contain more than one aggregate, but usually only one aggregate can be used in each query step. If several aggregates are created for a InfoCube, they are checked by the OLAP processor when a query is run. During this check, the OLAP processor reads the InfoCube aggregates and selects the aggregate that is suitable for the query definition.

Read Mode

Read mode determines the manner in which data are returned by the OLAP processor when you execute a query. SAP provides three read options. These are:

- (H) – Query to read, when you read or expand hierarchies: In H mode, the detailed information within the nodes and sub-nodes of a hierarchy are read on demand (i.e., when you expand a hierarchy node). The amount of data transferred to the OLAP processor is the least of all the three categories. The downside is that the number of database reads increases as you keep expanding hierarchy nodes. SAP recommends setting queries to the H read mode. This is based on the empirical evidence of the relative performance of queries set to each of these modes.
- (X) – Query to read data during navigation: In this mode, data are requested by the OLAP processor as dictated by every navigational step. The difference between X mode and H is that in the case of hierarchies, all the data on a leaf level are completely retrieved by the OLAP processor.
- (A) – Query to read all data at once: There is only one read process in this mode. When you execute the query in the Business Explorer, all data in the main memory area of the OLAP processor that is needed for all possible navigational steps of this query is read. During navigation, all new navigational states are aggregated and calculated from the data from the main memory.

The read mode Query to be read when you navigate or expand hierarchies significantly improves performance in almost all cases compared to the other two modes. The reason for this is that only the data the user wants to see is requested in this mode.

For queries involving large hierarchies with many nodes, it would be wise to select Read data during navigation and when expanding the hierarchy option to avoid reading data for the hierarchy nodes that are not expanded.

Reserve the Read all data mode for special queries—for instance, when a majority of the users need a given query to slice and dice against all dimensions, or when the data is needed for data mining. This mode places heavy demand on database and memory resources and might impact other SAP BW processes and tasks. A query read mode can be defined either on an individual query basis or as a default for new queries using the query monitor (transaction RSRT).

Step to set the read mode:-

Go to transaction RSDIPROP; specify the info provider name and change.

InfoProvider: Edit Properties

InfoProvider	KI_IC_PRD
Description	Products

Query / Cache	Roll Up	DB Performance
---------------	---------	----------------

This decides how many records are read during navigation. ✓

Read Mode	H Query to Read When You Navigate or Expand Hierarchies
Cache Mode	1 Main Memory Cache Without Swapping
Persistence Mode	0 Inactive
Data Integrity	<input type="checkbox"/>
<input checked="" type="checkbox"/> Delta Cache	
Delta Cache Class	CL_RSD_DC_SUPPORT_INFOCUBE

System Default

OLAP Cache

The OLAP cache is architected to store query result sets as highly compressed cluster data and to give all users on an application server access to those result sets. If the same query (or a subset) is then executed by another user, the subsequent query request can be filled by accessing the result set already stored in the OLAP cache. OLAP Cache has a system default of 100 MB for local and 200 MB for global cache. This may be too low for a system that can be optimized via broadcaster.

Step to change OLAP cache settings:-

Go to transaction code RSCUSTV14 and edit the settings

Change View "OLAP: Cache Parameters": Details

Customizing ID: Business Information Warehouse

OLAP: Cache Parameters

Cache Inactive

Local Size MB	125	
Global Size MB	200	
Persistence Mode	Flat File	Alternative Cache storage medium when Cache memory is exhausted (used When Swapping is enabled)
Flat File Name	BW_OLAP_CACHE	Application server specific logical file name
ComprehensiveFile	BW_OLAP_CACHE_SPAN	Cross Application server specific logical file name

Related Content

[Backend Performance Improvement Measures](#)

[Query performance](#)

[Query Performance](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.