Logical Data Partitioning of the Info-Cubes

Applies to:

SAP BI 3.x and 7.0, data base consider MS-SQL service pack1

Summary

Document explains about Logical partitioning of the info-cubes and its significance. Why this process is necessary and the problems individuals usually will be facing in the system if we don't' avert this situation.

Author: Vinay kumar H.S

Company: Infosys

Created on: 7 March 2011

Author Bio

Vinay is working for Infosys technologies since from 2009. Have a 1.5years of experience mainly on the SAP BI/BO tools.

Table of Contents

Introduction

As we all know that the SAP BI system will have a backend database to store the data records which comes into the system through the various data sources like R3, Standalone Legacy Systems etc.,. Whenever a data target is created in the SAP BI/BW system, some portion of the database memory will be allocated to its (data targets) in-terms of physical address.

This allocation of physical memory will differ based on the type of the data target.

For example:

- DSO is a flat-table and its capacity is entirely based on the total capacity of the database ideally.
- Info-cube is a multi-dimensional structure and its capacity is based on the number of dataload requests it has or indirectly number of the data partitions happened at the database level because of the data-loads happening or happened to the respective info-cubes.

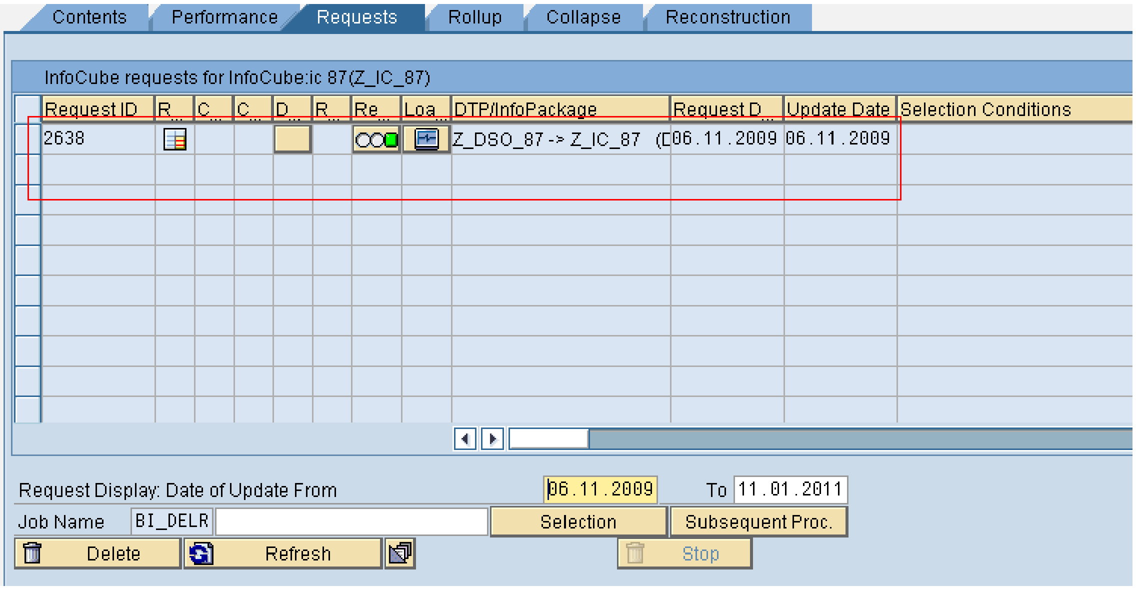
As the number of the dataload requests increases, the respective partitions at the database level also increases. This limit is database dependent.

Example, for SQL data base, it is 1000 and for DB2 it is 1200 approx based on their respective service packs installed.

Throughout this document I have consider the Microsoft's SQL database for the explanation, i.e., whose data partition limit is equal to 1000

Illustration of partition of the database w.r.t the Infocube data loads

As mentioned above the database partition of the database for the Infocube happens based on the number of dataload requests that cubes has. I.e. one new dataload request into the respective cubes creates a partition at the database multiply by the number of dataloads happened into it till date.



InfoCube requests for InfoCube:ic 87(Z_IC_87)

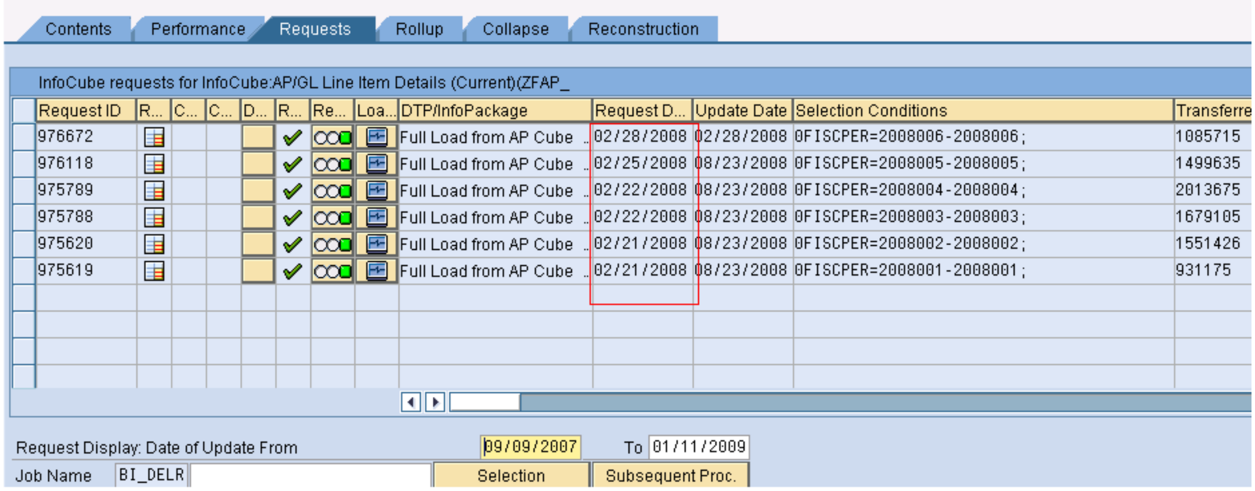
Request ID	R...	C...	C...	D...	R...	Re...	Loa...	DTP/InfoPackage	Request D...	Update Date	Selection Conditions
2638						○○🟩		Z_DSO_87 -> Z_IC_87 (D	06.11.2009	06.11.2009	

Request Display: Date of Update From 06.11.2009 To 11.01.2011

Job Name BI_DELR

FIG1: Illustration to indicate the dataload request for the Infocube.

As the number of the dataload requests increases, the number of the data-base partitions w.r.t info-cube also increases since both are directly proportional. Once this reaches the limit of the database used, no more data loads can happen for the respective Infocube (data-target) according to the property of the database.



InfoCube requests for InfoCube:AP/GL Line Item Details (Current)(ZFAP_

Request ID	R...	C...	C...	D...	R...	Re...	Loa...	DTP/InfoPackage	Request D...	Update Date	Selection Conditions	Transferre
976672	▤				✓	○○●	▣	Full Load from AP Cube ..	02/28/2008	02/28/2008	0FISCPER=2008006-2008006;	1085715
976118	▤				✓	○○●	▣	Full Load from AP Cube ..	02/25/2008	08/23/2008	0FISCPER=2008005-2008005;	1499635
975789	▤				✓	○○●	▣	Full Load from AP Cube ..	02/22/2008	08/23/2008	0FISCPER=2008004-2008004;	2013675
975788	▤				✓	○○●	▣	Full Load from AP Cube ..	02/22/2008	08/23/2008	0FISCPER=2008003-2008003;	1679105
975620	▤				✓	○○●	▣	Full Load from AP Cube ..	02/21/2008	08/23/2008	0FISCPER=2008002-2008002;	1551426
975619	▤				✓	○○●	▣	Full Load from AP Cube ..	02/21/2008	08/23/2008	0FISCPER=2008001-2008001;	931175

Request Display: Date of Update From 09/09/2007 To 01/11/2009

Job Name BI_DELR

Fig2: Illustration example used to explain the database partition limit.

Data base partition limit calculation

In the above considered cube, the dataload is happening from 21st Feb of 2008 as depicted in the figure (shown above).

Assumptions: Dataloads are happening every day.

Hence in a non-leap year, ideally 365 data loads =➔ 365 data partitions at the database level.

[SQL database is considered whose upper partition limit is 1000.]

Therefore till Nov 18, 2010 (313 days of 2008+ 365 days of 2009 + 322 days of 2010) the dataloads will happen successfully to the considered Infocubes, later to this the loads will fail as there will be no physical space to store the records.

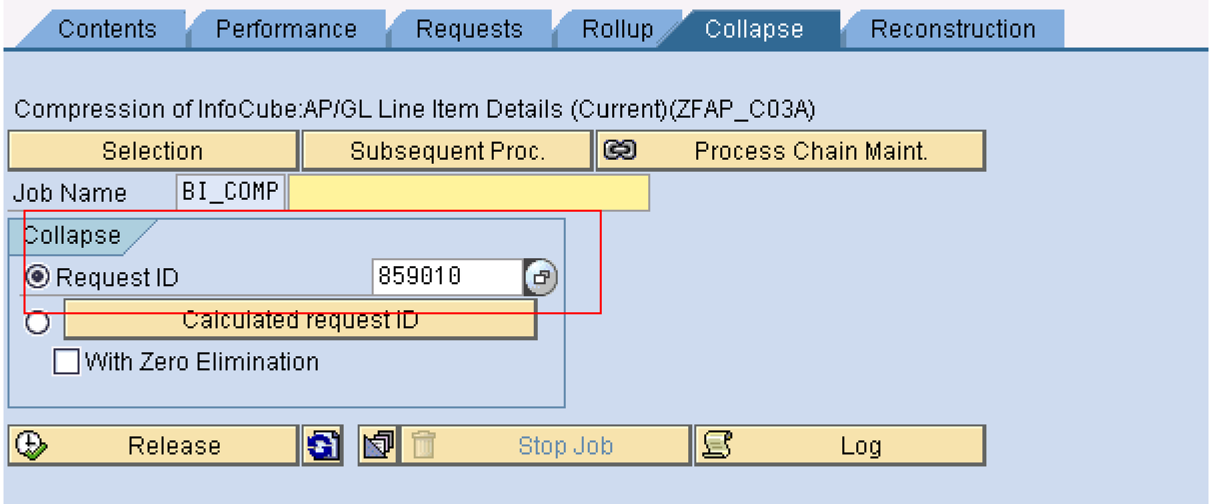
And also the SAP's standard table "**RSMSSPARTMON** " provides the details like the number of partitions happened at the database level w.r.t to the data targets(info-cube), these details can be used to get the accurate result w.r.t Info-cubes.

How to avoid the database partition limit consequences

One method is to use the logical partition of the data-targets which includes the backup of the data into some other data-targets from the actual ones and deleting the respective data (which is replaced to the other cubes) from the actual data-targets and compression of the actual data-targets dataload request whose data is replaced from the actual to the another newly created (copied) data-targets.

This compression activity will delete the data-load request details from the database level and hence indirectly reducing the number of partitions at the data-base level previously had happened w.r.t individual data-targets.

Steps to compress the Infocube (data)



Compression of InfoCube:AP/GL Line Item Details (Current)(ZFAP_C03A)

Job Name BI_COMP

● Request ID 859010

○ Calculated request ID

□ With Zero Elimination

Fig: Screenshot indicating the compression process of the infocube which reduces the number of partition at the data base level.

Select the required request ID or range of request ids for which the compression has to be carried, and then schedule the process.

This will delete the respective data-load request id/ids and reduce the number of partitions w.r.t the infocube/data-target created at the database level, thus increases the number of available partitions. [This compression activity can be controlled to the required level like by giving the individual request numbers or the range of the requests as depicted in the above screen shot.]

Refer this link for the complete details of the Infocube Compression activity. http://help.sap.com/saphelp_dm40/helpdata/en/ca/aa6437e7a4080ee10000009b38f842/content.htm

The compression of data-targets, for e.g. for info-cubes, will make the data deletion process using the data-load request id/ids impossible by moving the contents of fact table into E table, but this will not have any impact on the abovementioned logical process as the compression is done for the request/requests whose data is already replaced from the actual target to the other target and deleted from the actual target into which the data loads will be happening on regular basis.

Constraints for Logical partition of the data-targets

Following are the constraints we should make sure before we do the logical partition of the cubes.

- Check if there are any queries based on the current Infocube. If any, make sure that those queries are not affected due to Logical partitioning process.

  E.g.: when the data is replaced from the actual data-target to the other data-targets and deleted from the actual data-target and this other target into which the data is moved is not connected to the queries or the multi-providers of the actual data-target, then the respective queries will not display the replaced and deleted data.

  So in case this data (replaced and deleted from the actual data-target) is required for the business, suitable steps must be taken to ensure the data is not lost and the other data-targets created because of the logical partition process is connected to the respective multi-providers of the actual data-target and queries are working as intended according to the business norms.

- Check if, is there any dataproviders restrictions for the considered infocube's queries (Queries based on the multiprovider whose data provider is the underlined Infocube). If any, please make sure to add the newly created cubes having the previous data which is moved from one data target to the another one under the restriction or take the suitable steps by taking the business scenarios into the consideration like if the business don't want this replaced data, i.e. no business reports uses this data in the future etc.

  E.g.: If BW queries have few data-providers restriction and the logical partition for the underlined data-targets are done (which have restriction at the query level) and business wants the replaced and deleted data from the actual data-targets to be available for reporting, then the other data-targets developed because of the logical partition process should be connected or added under the filter restrictions of the BW queries.
- Any data access restrictions, if any, need to provide the same to the newly developed cubes/data targets (same as the original cubes) because of the logical partition.

  E.g.: There will be certain data access restrictions of the few of the data-targets for the security reasons, and in case the partition had been done on these data-targets, then the suitable data access restriction should be provided to the other data-targets developed because of the partition process.

Differences between the logical data partition and the

Standard data archiving process

Standard Data Archival	**Logical Partition of the infocubes**
Generally deals with the removal of the data from the production environment which is no longer in use.	Done to avert the impact of the data base partition limit
Complete migration of the data from the production environment which improves the performance of the system and also lessen the consumption of resources	Reduces the database partition limit by logically transferring the data from one data-target to the other followed by the compression activity
Enhances the performance of the system	Avoids the possible dataloads impact, no performance enhancements

Practical implementation of the Logical partitioning process

We had a requirement in one of the project to avert the effects of the logical partitioning of the data base because of the limit it had (data-base used) under it (project).

The process we had followed is explained through the following steps:

- The partition limit of the respective data-targets are identified through the contents of the standard SAP table **RSMSSPARTMON**
- The equivalent data-targets/info-cubes are developed w.r.t the actual ones and all the data loaded till the current fiscal year is moved/replaced into the newly created data-targets
- The data which was replaced from the actual to the other newly developed data-targets is deleted from the actual data-targets into which the regular data loads were happening
- Data load requests pertaining to the data deleted from the actual data-targets are compressed. This step was essential to reduce the number of data base partitions w.r.t the data-targets
- All the constraints mentioned in above section 4 are taken care to ensure that no data is lost during the process
- And finally we have kept the process of replacing the data from the actual data-targets to the other equivalent ones and deletion of the replaced data as an year end activity to avoid any possible effects due to this database partition limit

Note: Deletion of the data from the actual data-targets will take more time in the production environment based on the number of data-records it has. Hence the deletion activity has to be planned logically to avoid long duration unavailability of the production system to the users.

Conclusion

- Memory will be allocated to the different datatargets of the SAP BI system. Example: For DSO a memory in terms of a simple flatfile will be allocated and for the Infocube, a separate memory block will be created having the partitions limit which depends on the type of the database.
- Partition of the database happens based on the number of data requests that infocube has, each requests creates a partition.
- The number of partitions is database dependent
- As number of dataload requests for Infocube increase , the number of partitions at the data-base level also increases
- The dataloads to the infocubes will not happen once the partition limit is exceeded. So suitable steps must be taken to reduce the number of partitions. Example by replacing the data from one data-target to the other followed by the compression activity as illustrated in above section
- Once the required data is replaced logically, delete those requests from the respective infocube and do the compression of the data for the entire deleted requests. Without this compression activity, even though we have deleted the data from the datatargets, number of partitions will not be decreased. This mandates the compression process.

Related Links

www.help.sap.com
www.sdn.sap.com
SAP service market place

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.