Supportlink January-March 2006

Time-Based Partitioning Across Multiple Servers

Andy Rachmiel, Partner, BrightStar Partners Incorporated, Park Ridge, Illinois

This document illustrates how time-based partitioning can be used to overcome processing challenges for large Cognos PowerPlay cubes.

The Challenge: A client wishes to take advantage of multiple servers to process the data in several large PowerPlay cubes. The cube requirements include 60 months of data, detailed down to the day within the cube. Furthermore, the cube is built from flat files, and each month of data is contained within a separate flat file. The goal is to build five months of the cube on each of 12 servers, enabling them to build all 60 months across the machines. They would then consolidate each "monthly" slice onto a 13th server, which would host the cube to the end users.

Solution: Through the use of time-based partitioning and a little creativity, we can meet this challenge. The time-based partitioned cubes option offers an alternative method for updating time-segmented data, with many advantages over traditional cube groups or incrementally updated cubes. Instead of having multiple cubes with similar structures across the time dimension, or a single large cube containing all data, you can define a time-based partitioned cube that combines multiple member cubes.

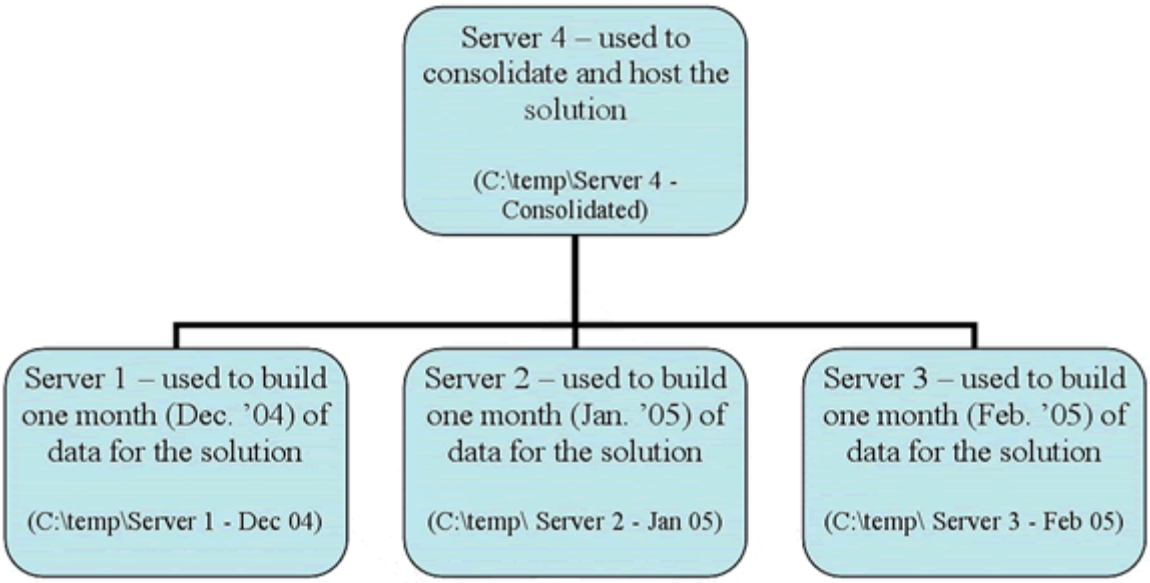
A time-based partitioned cube is defined on the time dimension, and each member cube is partitioned, or split, at the appropriate level, such as year, quarter, or month. Compared to traditional cube groups, time-based partitioned cubes allow PowerPlay users to view each cube independently, or to access the entire collection of member cubes as one time-based virtual cube. This means that reports can be viewed across the entire time dimension, or across only one level in the time dimension, such as a specific month.

With some creativity, the member cubes can be generated on distributed machines, and copied back to the location of the "master" model.

Important Considerations: The following items must be considered when utilizing this approach:

1. The model must be designed with dimension queries and one or more fact queries. The full set of dimension categories will need to be generated in the "master" model / cube / control file each time new categories show up. This is to prevent any confusion with the dimensions within in the control cube.
2. There must be one and only one time dimension within the model.
3. The "master" model must be built first in order to generate the control cube and control file. Note: this master model should be built with a subset of the data (for example, one record from each underlying member cube) to ensure that the control file is properly generated.
4. A copy of the "master" model should be used for each distributed build. Once the master model is built and the initial control file, control cube, and member cubes are built, the model should be saved and then copied to each location where the individual cubes will be built.

Example: The following example uses the Go Sales.mdl and Go Sales.asc files from the default PowerPlay installation (<install location>\cer4\samples\PowerPlay\Discovering Transformer). We will assume that there are four machines in the architecture; three that will be used to build cubes and a fourth that will host the results to the end users. For this sample, we will distinguish the various servers by creating different folders to store the information.



1. Copy the Go Sales.mdl and Go Sales.asc files from the location above to Server 4 in the example above.

2. Since this example uses three months of data, parse out the three months from above (December 2004 to February 2005) from the Go Sales.asc file. Note: for our example, we converted this file to an Excel format so that it would be easier to work with. This step is not necessary, but will be assumed going forward.

3. Based on the dimensionality of the model, create separate files that contain all possible combinations of each dimension, and add them to the model as dimension sources. Note: for this example, we converted the time dimension to go down to the day. This step is not necessary, but will be assumed going forward.

All Dates	Products	Regions
Year	Product Line	Region
Quarter	Product Type	Country
Month	Product	Branch Name
Week		Sales Representative
Day		

a. For example, the Products dimension has three levels, so a dimension source (flat file) was created with three columns for Product Line, Product Type, and Product.

b. This step was repeated for All Dates (Order Date) and Regions (Region, Country, Branch Name, Sales Representative).

4. From the file in step two above, parse out the information into four separate flat files.

a. December 2004 data

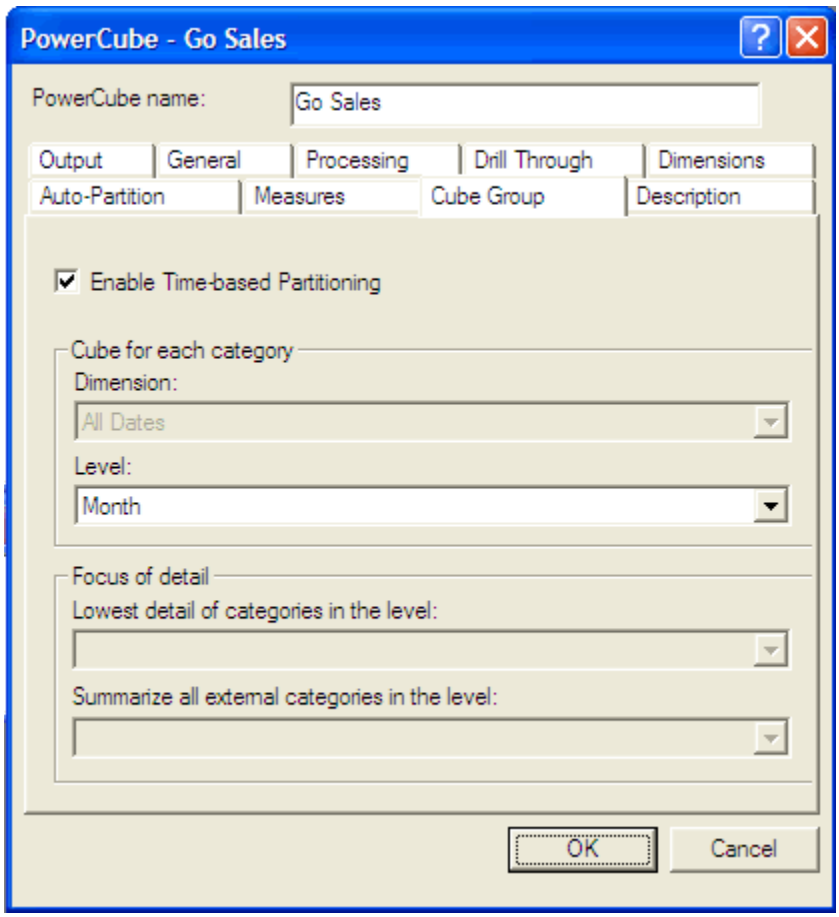
b. January 2005 data

c. February 2005 data

d. A three-record flat file that contains one record from each of the three files in a to c above. The date of each record does not matter as long as each of the three represent a date in the three files above, respectively.

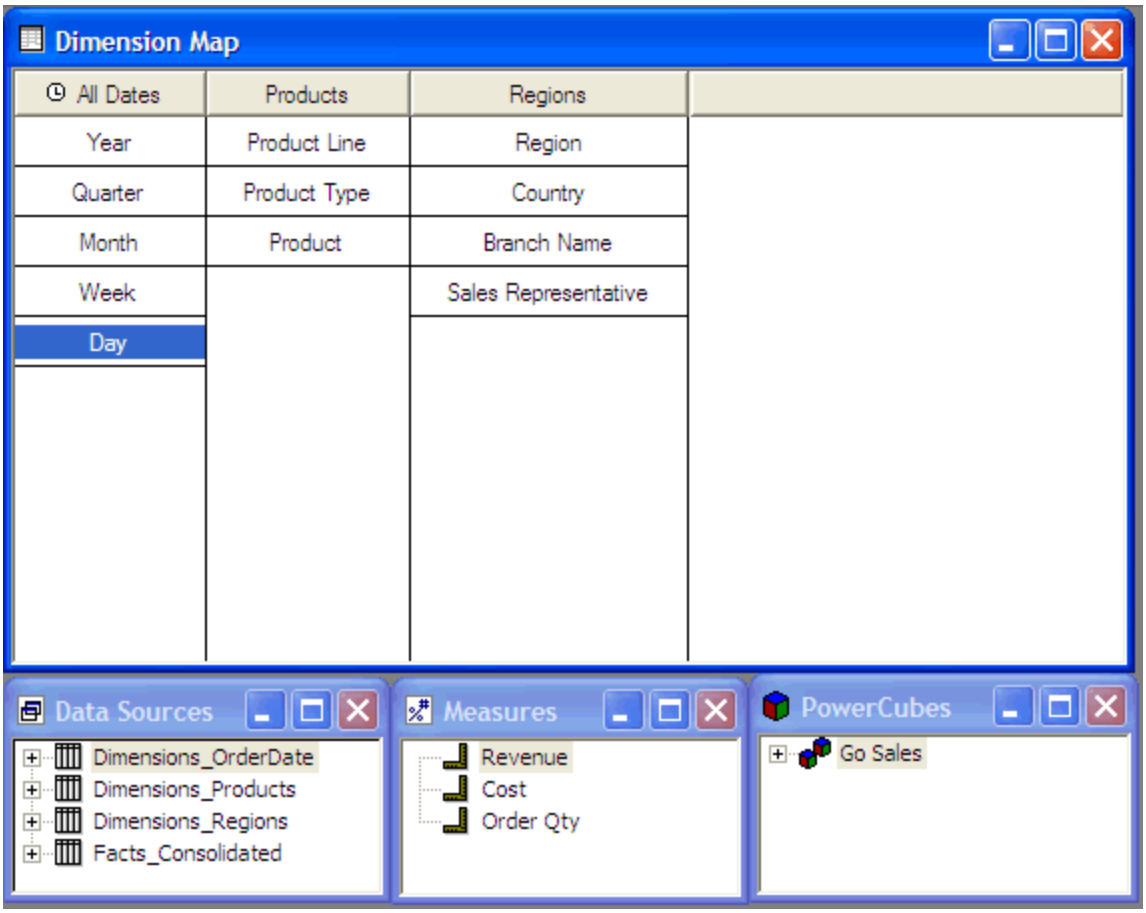
5. Modify the Go Sales.mdl from above on "Server 4" so that it is built from the three dimension sources and the one flat file containing three data records (4d).

6. Remove the existing PowerCube and add a new PowerCube, making sure that in the new PowerCube, the Cube Group tab appears as shown below. Note: in this case the member cubes will be formed at the month level. This can be any of the levels from within the time dimension in the model.



7. Build the model and once complete, save it.

8. The model should now appear as follows:



9. Note that in the location where the cube is built, you will have two files and one folder:

a. go sales.mdc – this represents a control cube (.mdc) which:

- contains the high-level metadata about the overall structure of the cubes.
- maintains a list of all measures, dimensions, root categories, currency records, and the entire structure of the time dimension.
- contains the control information to gather the data from the member cubes.
- serves as the entry point for PowerPlay users to access the time-based partitioned cube.
- shares the same filename and location as the .vcd file.

b. go sales.vcd – this represents a time-based control file which is an editable ASCII-format text file that maintains an index of the separate cubes in the time-based group.

c. go sales – this file folder is the location of the member cubes, which holds the following cubes in our example:

- 20041201-20041231.mdc
- 20050101-20050131.mdc
- 20050201-20050228.mdc

Note: currently these member cubes contain only one record per cube. These will be replaced by the member cubes built on the distributed servers.

10. Copy the model along with the existing dimensional data source files from step seven above to each of the other three “servers” described in the diagram above. In addition, copy each of the three monthly flat files from step four to the respective servers.

11. On each of the three distributed “servers”, open the model from step 10 and change the fact source from the one used on the consolidated server to the one for the month that the server is going to build. For example, on “server 1”, the fact should point to December 2004 data.

12. In addition, modify the location of the PowerCube to the appropriate location on the server.

13. Generate the cube for each of the three distributed “servers”. Note that this will create a control cube, control file, and member cube similar to that seen in step 9.

14. Copy the member cube from each of the distributed “servers” above in place of the original member cubes on the Consolidated “server”.

For more information, refer to the help within PowerPlay Transformer.