Panorama NovaView™

Improving SAP NetWeaver® BI Platform Performance

NovaView Enables Proactive BI

Most decision-makers in operational environments – finance, production, HR, sales and marketing, and procurement – have little inclination or need to learn complex reporting and analysis tools. What they require, and NovaView provides, is:

- User-friendly and easily understood reports that support effective operational decisions
- Fast access to relevant, insightful information
- Ability to quickly interact with reports and KPIs, such as drill down from summary to detail data, display additional attributes, and view information from different perspectives
- Search capabilities that let them quickly find specific items such as a customer, supplier, distributor, product, or part number

Many of these interactive reports and KPIs have standard content and formats, but contain information that is updated as often as hourly. Another attribute is that any one report can be accessed by many different individuals. The information in these standard and frequently used reports can be stored in computer memory (or on a disk) via a process called caching. The original query doesn't have to be rerun because the result set is available from the cache.

NovaView SmartCaching enables quick interaction with KPIs, dashboards, views, and reports by reducing the need to have queries answered by the SAP BW OLAP processor. Instead, many queries can be resolved directly from the NovaView cache, using techniques described below.

NovaView-generated MDX queries are optimized to work directly against the OLAP processor. For queries that aren't resolved from the cache, the optimized MDX results in fast and efficient query performance in SAP BW.

Connection pooling consolidates multiple user requests into a single connection to the OLAP processor, reducing network overhead and making queries faster and more efficient.

This combination of SmartCaching, optimized MDX and connection pooling enables operations managers to find information quickly so they can make faster, smarter, and more profitable decisions.

Query Performance in SAP NetWeaver® BI

Performance tuning in any data warehouse environment is a combination of science and art. SAP recommends a number of ways to speed up queries:

- Follow best practices when designing InfoCubes
- Partition large InfoCubes into several smaller ones (by year, for example) using MultiProviders
- Create effective aggregates
- Load InfoCubes into SAP BI Accelerator
- Use the SAP BW OLAP cache and pre-calculate complex queries

SAP stresses that there is no "silver bullet" for performance tuning, including use of the BI Accelerator. All factors must be considered in what SAP refers to as "Strategic Performance Management." An overview of each of these techniques is included at the end of this whitepaper. A list of SAP Knowledge Base documents relating to query performance tuning is also included.

These techniques require a lot of expertise, which is not always readily available. Even with them, response times can be slow because the OLAP processor is generating SQL queries to the InfoCube using a technique called Relational OLAP (ROLAP).

Panorama NovaView can overcome performance limitations in most SAP BW implementations, and on top of a well-tuned SAP BW system, its performance-enhancing technology can generate lightning-fast responses.

A New "Smart" Layer of Caching

SmartCaching speeds query performance by using the cache on the NovaView server first, before sending the query to the SAP BW OLAP processor. The result sets from previously used NovaView queries for views and reports are stored in highly compressed tables on the NovaView server, providing nearly instantaneous response times, even for complex queries.

SmartCaching builds on the SAP BW strategy of limiting the number of queries run by the OLAP processor. Using advanced techniques, NovaView adds a very fast layer of caching that can handle the majority of user queries.

If a query can't be resolved by the NovaView cache, it will then go to the OLAP processor, which in turn will try to resolve the query from the SAP BW OLAP cache.

NovaView SmartCaching can be faster and more efficient than OLAP caching because it stores all the intermediate steps a user takes in creating a query. These expanded result sets increase the probability that a query can be resolved independently of the OLAP processor and cache.

There are two modes of SmartCaching that can and should be used together: *Standard* and *Predictive.*

Standard SmartCaching: An automated process fires off MDX queries for a predefined set of frequently used NovaView views, such as daily production, inventory or sales reports. The reports are accessed frequently by multiple people, making them good candidates for caching. It is also possible to cache all the reports in a briefing book.

For example, a sales organization could have ten basic daily reports that are accessed by sales managers, sales persons and senior executives. A database administrator would create a SQL Server job that simulates opening these ten NovaView reports, filling both the NovaView cache and the SAP BW OLAP cache. These particular views would be executed with drill down to the detail level, such as for each individual sales person, product, or time period. There is nothing fancy or

complicated about this operation, but it will provide very quick response times for most basic reporting and user interactions.

Predictive SmartCaching: The second mode uses an algorithm to determine what views should be cached, and in what order. For each user action, NovaView generates a separate MDX query. Each result is saved separately as a view in the NovaView cache. The final result set comprises each of these MDX queries, collectively referred to as the "extended state" of a cached NovaView query. In this way, every step a user takes to get to the final answer becomes part of the cache. Any step of the query can be accessed by another user from the cache, without the query going back to the OLAP processor.

Additional information for each view also is captured:

- Average Process Time: How long did the query take on the SAP OLAP server (You want to cache slow queries and let the OLAP server handle fast ones).
- Hit Count: How many times each cached view is hit, denoting its popularity.
- Hit Frequency: How often a view is hit in a given time period; for example, one hit every three minutes or one hit per week.
- Last Access: How long ago a cached view was last accessed. If a formerly popular view hasn't been accessed in several weeks, it probably shouldn't be cached.

Linear scores are calculated for each measure. For example, the longer a query takes on the OLAP server, the higher the score it will get. The combination of these scores can be used to set priorities for caching views. Obviously, views that are used frequently, have been accessed recently, and have slow query response times are candidates for caching.

Based on knowledge of how managers use reports, a database administrator can set weights for each of these measures (with a total of 100) and then schedule a job to run these queries during off hours (after the SAP InfoCubes are updated). Currently, assigning the weights is a manual process, but Panorama is working on a new algorithm that will assign the weights automatically.

Ad Hoc Queries Using On-Demand Retrieve

Even with standard and predictive caching, there will always be ad hoc queries that can't be resolved from the NovaView or SAP BW OLAP cache. As mentioned earlier, every action taken by a NovaView user results in a separate MDX query to the OLAP processor. If, for example, each step generated a query that took 15 seconds to complete, and the user took six steps in all, the total query time would be 90 seconds. Users are likely to be frustrated by a process that takes this long.

Panorama has developed a new technique that lets the user go through each of these six steps before generating the final query to the OLAP processor. In On-Demand Retrieve mode, NovaView will fire off very lightweight and fast MDX queries to populate the rows and columns of a view, set slicers, create filters, enable drill down, and do other activities that don't result in filling the cross tab grid with data. Once the user is satisfied with what she wants, she can launch the final query, which may take only 15 – 20 seconds in total to complete, instead of 90.

Optimized MDX Queries

NovaView SmartCaching and On-Demand Retrieve are very powerful ways to improve performance. NovaView also uses highly efficient MDX to query SAP InfoCubes, previously defined queries, and data store objects.

Queries from NovaView to SAP BW are rendered using MDX, which is rapidly becoming the lingua franca of multidimensional analysis. NovaView connects to SAP BW using the OLE DB for OLAP (ODBO) applications programming interface, an industry standard. Each user action in NovaView, such as slicing, drilling, filtering, calculating, or sorting, generates a separate MDX query. It's these individual actions that provide the interactivity required for executives, analysts and operations managers.

MDX originated with Panorama. The Company sold its OLAP server technology to Microsoft in 1996, which became "OLAP Services" in Microsoft SQL Server 7.0. NovaView was written to execute highly efficient MDX queries to OLAP Services, while hiding the complexity of MDX from the end user. OLAP Services has become Analysis Services 2005, part of the SQL Server 2005 BI platform.

SAP adopted MDX and the ODBO API in 1998, but its implementation of MDX differs from Microsoft's. Panorama engineers have worked closely with the SAP BW team over the past two years to enhance the use of MDX in SAP BW. Panorama also rewrote the NovaView query engine to generate MDX code that is optimized to work with SAP BW and NetWeaver BI. NovaView MDX queries are translated to SQL by the SAP BW OLAP processor.

Connection Pooling

Panorama NovaView Server supports connection pooling in which queries from multiple web clients are routed to a single OLE DB connection. All requests are serviced with a relatively small group of connections maintained for extended periods.

This approach greatly reduces the amount of data the needs to be moved around, reducing network and server load and improving response time. It also minimizes the number of connections done to the OLAP platform.

The combination of SmartCaching, optimized MDX and connection pooling in NovaView can have dramatic effects on query response times. The various techniques described in this article work best, however, with a well-tuned SAP BW system. While it can improve performance, NovaView by itself cannot overcome badly designed InfoCubes or a poorly maintained data warehouse. NovaView SmartCaching warms up both the SAP BW OLAP and NovaView caches, but the queries still need to be run first by the OLAP processor against the InfoCubes or queries. You want these queries to run as efficiently as possible, even if they are being run in off-peak hours.

With a well-tuned SAP BW system, NovaView can provide lightning-quick responses, providing every one in the organization with the information they need, when they need it.

Addendum: SAP BW Query Performance Tuning

These are the primary factors that affect query performance according to SAP:

- Design of the InfoCube data model: SAP recommends ensuring that dimension sizes are <= 10 percent of the fact table size, maintaining dynamic characteristics (such as customer and material) in separate dimensions, using many smaller dimensions as opposed to a few big ones, making sure fact and dimension table indices are rebuilt after data loads, and compressing the cubes. There are many other factors, but these are the most important.

- Using MultiProviders to logically partition homogeneous InfoCubes: SAP recommends breaking a large InfoCube into several small ones, such as one InfoCube for 2006 and a second InfoCube for 2007. A query seeking data from 2007 would not hit the 2006 InfoCube. SAP strongly recommends that MultiProviders be used with InfoCubes instead of direct queries to an InfoCube.

- Carefully designing aggregates, particularly for navigational characteristics: As with SSAS, SAP BW will make recommendations based on the structure of InfoCubes. Unlike SSAS, however, an administrator can fine-tune aggregates. SAP BW aggregates are actually maintained in InfoCube structures. The BW OLAP processor will create SQL (sub) queries for each of these aggregate InfoCube structures, taking advantage of parallel processing in the DB. There are variety of techniques for creating and maintaining aggregates. Norbert Egger notes in his book, SAP Business Intelligence, that the creation of effective aggregates "requires a certain degree of technological know-how and experience regarding user behavior in relation to the use of analytical applications." For this reason, he writes, aggregations frequently are not implemented well in SAP BW. But, he says, they are critical to improving query response times.

- MOLAP: SAP BW offers an option to export InfoCube data to Microsoft SQL Server Analysis Services. The connection between the BW server and Analysis Services is via an SAP MOLAP bridge on the DB server. It uses remote function calls to transport InfoCube data via MDX.

- Global OLAP Cache and Pre-Calculation: SAP states that the OLAP Cache should be used whenever possible. Every query fills the cache, either in memory or on disk (or both using swapping). Subsequent identical queries (or subsets of the query) will hit the OLAP cache and not have to read the database tables. In addition, you can "warm up" the cache by issuing queries that include drill-down paths (pre-calculation). For example, a query with the selection parameter Calendar Month= 'Unrestricted' fills the OLAP cache with all the months. A subsequent query for Calendar Month = '05.2007' will be resolved from the cache, not from the database tables. You can use the SAP Reporting Agent or the OLAP trace to warm up the Global OLAP Cache. The Reporting Agent typically is used to pre-calculate BEx Web templates and the pre-calculated query is stored in the Reporting Agent storage. However, any query, whether from BEx, a Web template, or a third-party reporting tool (such as NovaView), will also warm up the OLAP cache. The Global Cache is invalidated after every data load and must be rebuilt.

- BI Accelerator: Basically, this is a "black box" that combines blade servers (HP or IBM) and SAP's proprietary TREX search and classification software in one system. It is a highly scalable analysis server that, in effect, loads all the contents of an InfoCube into memory on a 64-bit Linux platform. The SAP OLAP processor will still check the Global OLAP cache first to answer a query and, if it can't, it will send the query to the BI Accelerator (if it's available.) Tests by SAP have shown that BI Accelerator response times are approximately as fast as the Global OLAP cache.

- Hardware: It goes without saying that the SAP BW server has to be on the right hardware (multiple fast processors, lots of memory, fast disks in RAID 5, multiple application servers as required, etc.). The database engine (SQL Server, Oracle, DB2) must be set up and tuned properly.

Additional information about query performance tuning can be found in these documents:

- Performance Tuning for SAP BW
- Multi-Dimensional Modeling with BI
- Performance Gains in SAP BW 3.x using Caching and Pre-calculation
- SAP BW Query Performance Tuning with Aggregates
- SAP Business Intelligence (SAP Press 2007): This is an excellent SAP BW/NetWeaver technical resource.

These documents, except for the SAP BI book, can be found in the SAP BI Performance Tuning Knowledge Center on SAP's website.