



# Panorama NovaView - Enterprise BI Performance and Scalability

Technical Whitepaper

## Contents

Abstract .....	3
Introduction .....	3
High Performance and Scalability .....	3
Intelligent Data Management .....	4
Dynamic Memory Management .....	5
Low Client Overhead .....	5
Behavior-Oriented Design .....	5
MDX Query Optimization .....	6
Panorama NovaView Scalability Benchmark .....	6
Background .....	6
Interpreting Results: How Does A Strong BI Architecture Perform? .....	6
Benchmark Results .....	7
Efficient Performance on Multi-CPU Servers .....	7
Increasing User Load .....	9
Increasing Data Size .....	9
Panorama vs. ProClarity .....	10
Conclusions .....	10

## Abstract

This whitepaper describes how Panorama NovaView™, the leading business intelligence solution for the Microsoft® and SAP platforms, has been architected from the ground up for enterprise scalability. It also outlines key architectural features that enable NovaView's scalability. Panorama NovaView is a comprehensive Proactive business intelligence solution that integrates analysis, enterprise reporting, scorecards and dashboards into a single unified solution.

This paper also details the results of two independent benchmark studies on Panorama's solution. The study demonstrates how Panorama NovaView scales to an unprecedented degree, serving thousands of OLAP users in a production class environment. NovaView outperformed any third-party ISV running on a single 32-bit server, as well as distributed across 32-bit and 64-bit environments.

## Introduction

Scalability is a key criterion for technical evaluators when deciding on a business intelligence solution, and for good reason. Business intelligence applications are high-investment, high-value initiatives with the potential to dramatically improve business processes such as supply chain management, customer marketing, and retail analysis. But the value of BI can only be realized if BI systems reach across the extended enterprise, providing employees and partners with a single, integrated, and actionable view of the business. An effective enterprise BI application must provide fast response times while supporting thousands of users around the globe as they access and analyze large sets of data.

All BI vendors claim to scale, but as Gartner Group notes, "Enterprises should not take these claims at face value, but instead must question vendors and customer references and understand how product scalability is achieved."<sup>1</sup>

## High Performance and Scalability

Scalability and performance were key considerations in the design and implementation of the Panorama NovaView architecture, and continue to be key points of consideration throughout the implementation of the Panorama product roadmap. Many of the key scalability-enhancing features reside in Panorama NovaView's Server, a powerful middle-tier application server that lies between Panorama's applications and Microsoft SQL Server Analysis Services and SAP Netweaver BI (BW) platforms. These features include:

- ▶ Intelligent queuing, connection pooling, and caching
- ▶ Dynamic memory management and process control
- ▶ Low client overhead
- ▶ Behavior-oriented design
- ▶ Optimized MDX queries

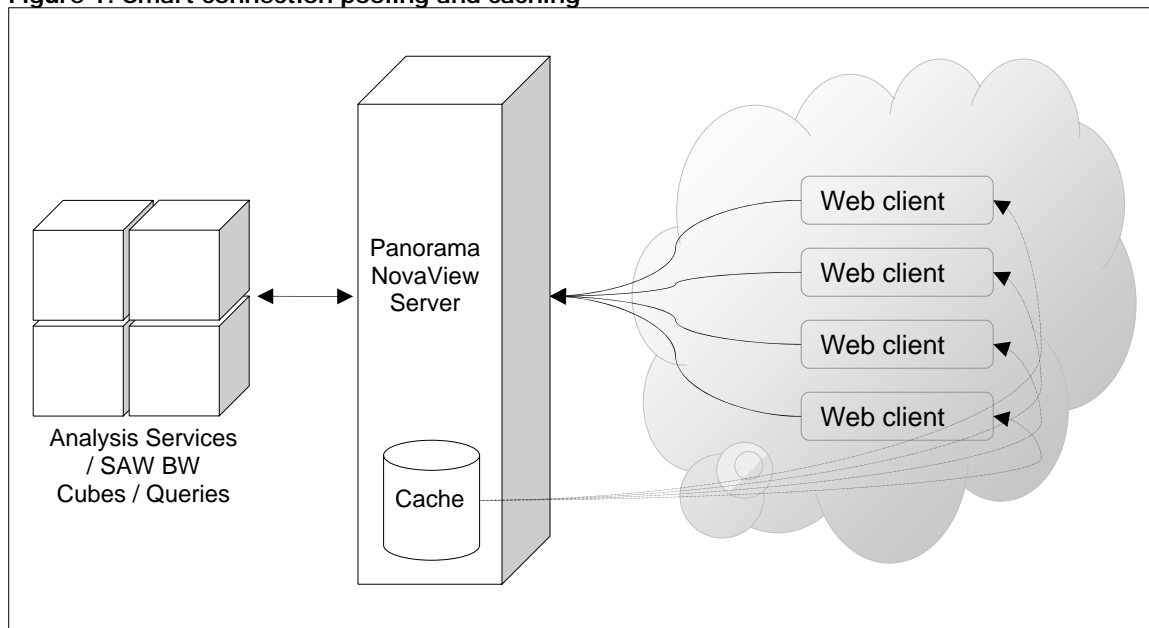
## Intelligent Data Management

Panorama NovaView handles all data management services for users of the client interfaces. It receives requests from clients, queues them for access to data source connections, caches the resulting data sets, and delivers them back to the clients.

When providing these services, Panorama NovaView takes into account the current user and data context, and ensures that data is queried and delivered as efficiently as possible. This is a key contributor to Panorama NovaView's overall scalability because it reduces overhead in the form of unnecessary activity against the data tier (SQL Server Analysis Services or SAP BW).

For example, NovaView Server routes queries from multiple web clients to a single OLEDB connection if they have similar characteristics, such as user roles and slicing authorizations. In essence, it services all requests to the SQL or BW platforms with only a relatively small group of connections which are kept alive for extended periods. Then it caches the query results and uses the cached data to service any identical requests over a configurable time interval. This enables the server to distribute the results to the largest community of users with the least amount of load.

**Figure 1: Smart connection pooling and caching**



This approach greatly reduces the amount of data that needs to be moved around, reducing network and server load and improving response times. It also minimizes the number of connections to the backend platform that the NovaView Server needs to open and close.

## Dynamic Memory Management

In an enterprise-scale BI application, very large queries can quickly hit Windows memory limitations when pooled together into a single Windows process. For this reason, older process-oriented architectures often run into a scalability wall when they attempt to handle a sudden rush of big queries.

To address this, Panorama NovaView controls the number of connections running on each Windows process, while still maintaining the efficient connection pooling mechanism described above. This allows queries to consume as much memory as needed—up to 2-3 GB per process—limited only by the amount of memory the machine has.

NovaView dynamically generates new processes as required based on its own real-time analysis and parameters you can configure to maximize server performance.

## Low Client Overhead

Panorama NovaView uses a number of strategies to maintain the lowest possible overhead when servicing client requests. This ensures the fastest response times and the best user experience, even when exploring large data sets in a large, multi-user environment.

For example, NovaView uses a state-based architecture to make smart, centralized decisions about how to service user requests. It tracks and stores the state of each user in its database, enabling the server to reduce the amount of data forwarded to clients down to only what is necessary. Moreover, because the information is stored centrally, it is used to service requests intelligently across multiple servers in a distributed server farm environment.

Also, because all user interface elements are created on the client side, the server passes the data set only, without a large amount of metadata describing the required visualization controls.

## Behavior-Oriented Design

Many of the scalability-enhancing architecture features in Panorama NovaView are based on observed user behavior in large enterprise environments. These features were designed specifically to respond to common scenarios in which users inadvertently create extreme load on the application.

For example, analysts can often create queries that consume large amounts of memory and/or take a long time to execute due, in part, to the physical characteristics of the underlying application. When the query doesn't come back quickly, they will often close their browsers, re-open them, and try again. They may attempt this several times, unaware of the fact that their queries continue to run after the browser is closed, and that they have submitted several redundant, heavy queries that will backlog the middle tier and block other requests.

To address this common scenario, Panorama NovaView includes a Query Governor that both users and administrators can use to easily cancel queries at the push of a button. Alternately, administrators can establish policies that ensure queries are automatically cancelled after a configurable time interval has passed.

## MDX Query Optimization

Panorama has been involved with the evolution of the MDX query language since its inception. As with any query language, drastic improvements in performance can be gained by sophisticated optimization of the MDX syntax. These enhancements can dramatically decrease the time it takes to run certain complex queries, thus allowing the mid-tier server to handle more queries faster.

For example, analysts often create a “top ten” style report against multiple dimensions: “Show me the top 50 customers in Europe in 2007 across all products.” To accomplish this, they nest the dimensions along a single axis and filter to show the top N combinations.

	Cust 1	Cust 2	Cust 3	Cust 4	Cust 5	Cust 6	Cust ...
Geo = Europe							
Time = 2007							
Product A							
Product B							
Product ...							

For queries like these, non-optimized MDX can take gigabytes of memory and hours to execute. NovaView optimizes these queries using specialized MDX functions that enable it to reduce memory consumption to less than 5MB and get response times down to seconds.

## Panorama NovaView Scalability Benchmark

### Background

Panorama NovaView’s multi-tier architecture, combined with specific architectural features and enhancements, provide best-of-class scalability in enterprise environments around the world today. To validate and demonstrate that scalability and performance, Panorama Software has engaged in numerous third-party benchmark studies.

During one benchmark, Microsoft SQL Sever Analysis Services and Panorama NovaView were deployed on a 32-bit server with 16GB of RAM and with both 8 and 16 processors. The team tested along three broad parameters: increasing the number of CPUs, increasing the number of concurrent users, and increasing the size of the data set.

### Interpreting Results: How Does A Strong BI Architecture Perform?

To best appreciate the benchmark results, it helps to have an understanding of how an efficient, modular, contemporary architecture performs as test parameters change. While it is important to

evaluate an architecture based on absolute maximums (e.g., maximum number of concurrent users, maximum rows of data), it is also important to notice how the architecture responds to a changing environment. Specifically:

*Strong BI architectures make the best use of additional CPUs on SMP boxes.*

As processors are added, expect to see a proportional increase in the server's ability to handle increasing user load. This indicates that the server handles multi-threading, memory management, and process management intelligently.

*Strong BI architectures handle increased user load elegantly.*

As the number of concurrent users on the system increases, expect performance to degrade incrementally in a predictable and linear fashion. This indicates that the system handles connection pooling and caching intelligently, without arbitrary programmatic limits or unexpected internal blocks.

*Strong BI architectures handle increased data sizes elegantly.*

As the size of data in a query increases, expect performance to degrade incrementally in a predictable and linear fashion. This indicates that, given a fixed degree of query complexity, the system handles data processing and management in an effective way.

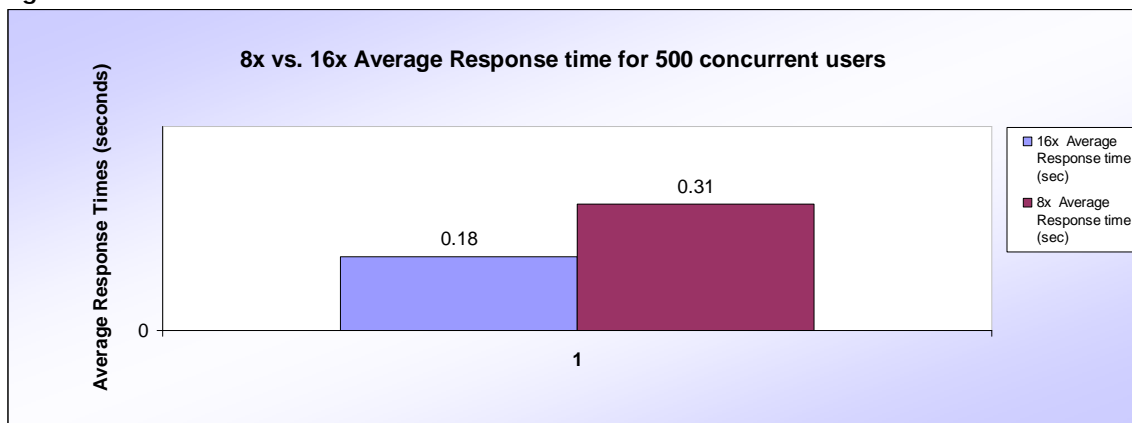
## Benchmark Results

### Efficient Performance on Multi-CPU Servers

In the first set of tests, Panorama NovaView performance was tested on an 8-CPU server and a 16-CPU server across load scenarios of 500 users, 1,000 users, and 1,500 users.

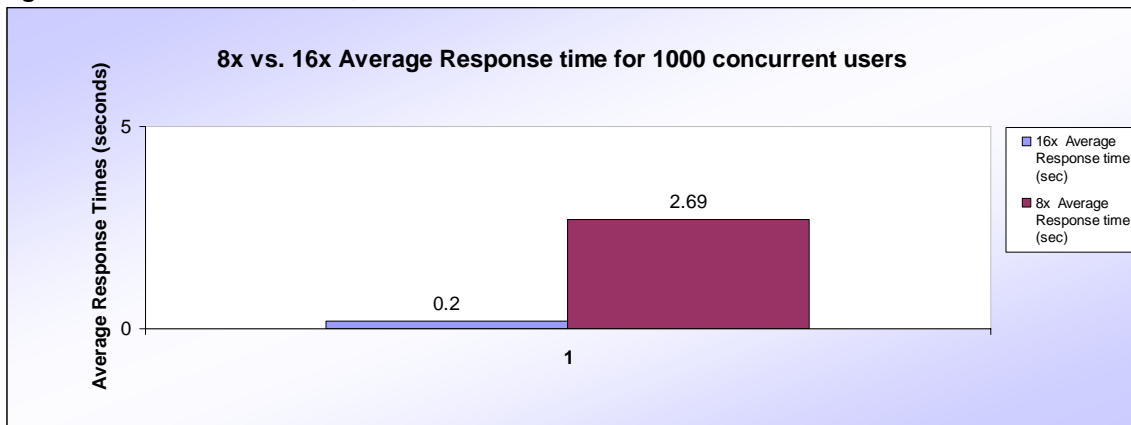
With 500 concurrent users, the NovaView Server response times were 1.72 times faster on the 16-CPU server.

Figure 2: 8 vs. 16-CPU with 500 users



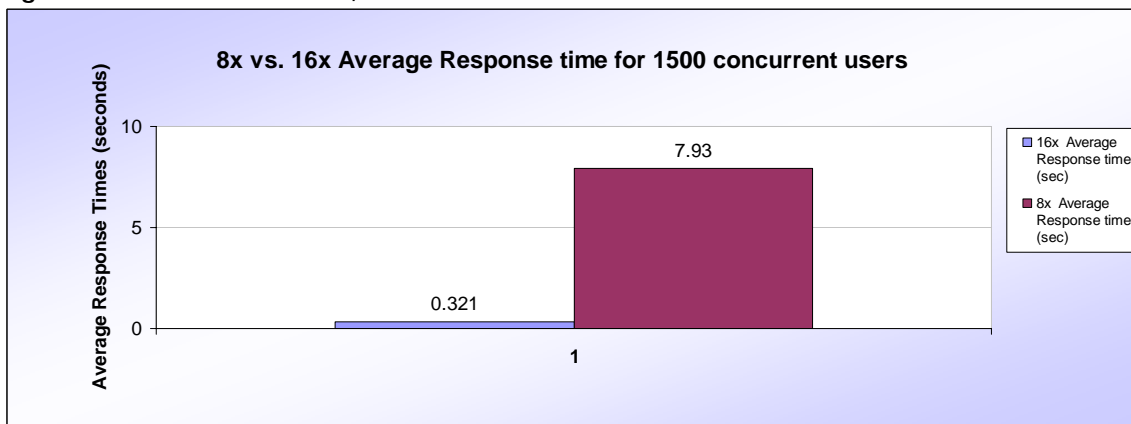
With 1,000 concurrent users, the NovaView Server response times were about 13 times faster on the 16-CPU server.

**Figure 3: 8 vs. 16-CPU with 1,000 users**



With 1,500 concurrent users, Panorama NovaView response times were about 24 times faster on the 16-CPU server.

**Figure 4: 8 vs. 16-CPU with 1,500 users**



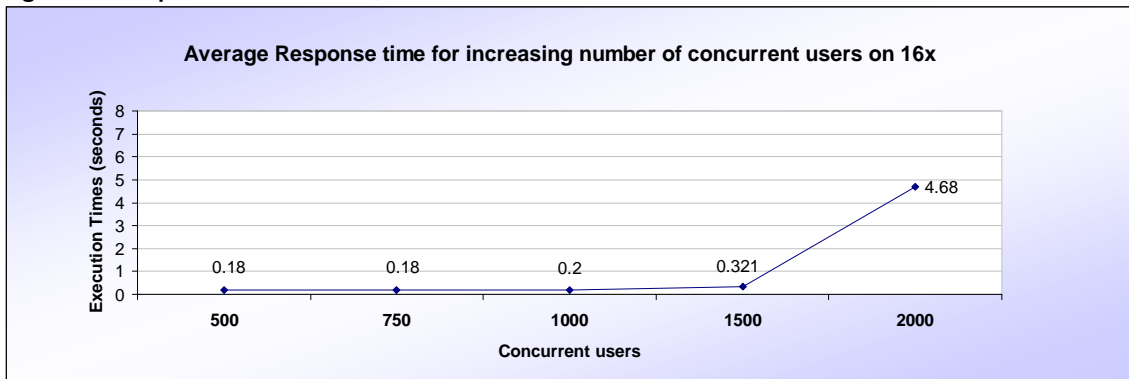
Clearly, NovaView performs and scales in a large IT data center scenario in which the BI application is hosted on a large, dedicated SMP box that services a large user community. Note the average response time of well under half a second for 1,500 concurrent users and how NovaView takes maximum advantage of additional processors to service heavy user demand.

## Increasing User Load

In the second set of tests, NovaView Server response times were benchmarked as the number of concurrent users increased. For these tests, the team deployed Panorama NovaView on a 16-CPU ES7000 server.

With 1,000 concurrent users, Panorama NovaView responded in well under a second. Even with 2,000 concurrent users, NovaView response times were still well within the acceptable range.

Figure 5: Response time with 2,000 users

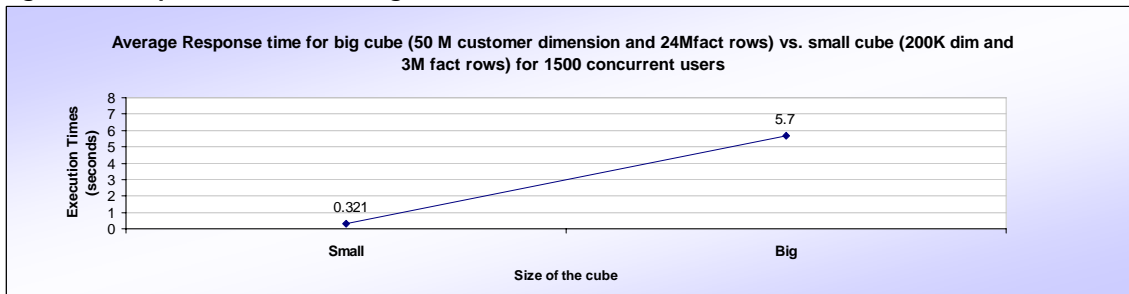


## Increasing Data Size

In the third set of tests, response times were benchmarked on a very large cube with SQL Server Analysis Services running on a separate 64-bit server (the larger address space was necessary to handle the volume of data—the cube contained an OLAP dimension with 50 million members, and the fact table contained 24 million rows).

Tests determined that NovaView provided very good response times even on massive cubes with a large number of concurrent users. The team also noted that despite the data size, NovaView caused relatively little load on Analysis Services because of its intelligent caching and connection pooling capabilities.

Figure 6: Response time on a large cube



## Panorama vs. ProClarity

In another independent third party evaluation of Panorama NovaView compared to ProClarity 6.0, both running on SQL Server Analysis Services 2005, the following results were obtained:

	Microsoft SSAS 2005 with <b>ProClarity</b>	Microsoft SSAS 2005 with <b>Panorama NovaView</b>
100 Concurrent Users	49 seconds	0.5 seconds
500 Concurrent Users	Crashed	8.1 seconds

## Conclusions

Increasingly, technical decision makers understand that business intelligence applications must scale in large, enterprise class environments. BI is becoming more pervasive across the extended enterprise—it touches employees, branches, partners, and even customers around the world. BI systems link multiple data sources and expose tremendous volumes of information to thousands of concurrent users. The key to success in this type of environment is an efficient, modular, and contemporary BI architecture that scales well.

Panorama NovaView provides that architecture. Unlike other vendors that have repurposed desktop code for server environments, Panorama built NovaView from the ground up to integrate closely with Microsoft SQL Server Analysis Services and SAP Netweaver BI (BW). NovaView provides best-of-breed performance through smart caching, connection pooling, and data management. As a result, NovaView scales extremely well in environments with large user communities and massive volumes of data, thereby providing superior query response in any environment. Moreover, NovaView behaves in a predictable and linear fashion, managing load elegantly where other solutions grind to a halt.

\* Independent benchmarks conducted by Unisys and a leading international consumer packaged goods manufacturer.