NETEZZA
The Power to Question Everything™

## Netezza Performance Server® Data Warehouse Appliance:

# An Architectural Comparison

Whitepaper

1 866 NETEZZA
WWW.NETEZZA.COM

NOVEMBER 2006

All rights reserved. © 2006 Netezza Corporation.

CONFIDENTIAL

## Executive Summary

Data is flowing into organizations at unprecedented rates. With the growth of e-business, wireless communications, RFID and other technologies, many large enterprises collect hundreds of gigabytes or even terabytes of detailed data every few weeks. Fast, comprehensive analysis of this data can provide vital information about customers, products and operations, helping increase profitability and market share.

But data volumes are so huge, and the need for business-critical information so sweeping, that this creates challenges for organizations across many industries:

- For the major wireless telecommunications carrier, it means running CRM analyses while collecting 250 million call detail records daily in order to proactively market to at-risk customers before they churn.
- For the healthcare provider committed to providing the best and most efficient care, it's the need to conduct sophisticated data mining of terabytes of operational and patient data, even as data grows exponentially.
- For the large online retailer, recording every click of millions of customers' shopping habits, it's the need to thoroughly analyze billions of rows of data in order to develop targeted promotions.
- For the grocery retailer, it means being able to perform complex market basket analyses against detailed line-item level transactional data to understand customer behavior and develop optimal merchandising and promotional strategies.

However, none of these goals can be met successfully when queries for information stored in the enterprise data warehouse take hours or even days to complete. Current data warehouse systems are based on older architectures that weren't designed to handle today's demands for querying enormous amounts of data. As a result, many business intelligence solutions are abandoned or severely under-utilized by the users that they were intended to help.

Netezza overcomes the limitations of traditional architectures adapted for data warehousing with a unique approach that delivers orders of magnitude improvements in performance, affordability and ease of use. The Netezza Performance Server® family of data warehouse appliances are designed specifically for powering complex ad-hoc analysis of terabytes of dynamic, detailed data. In delivering 10-100 times the query speed at half the cost of competing solutions, Netezza allows companies to do the types of detailed analysis that previously would have been impossible or cost prohibitive.

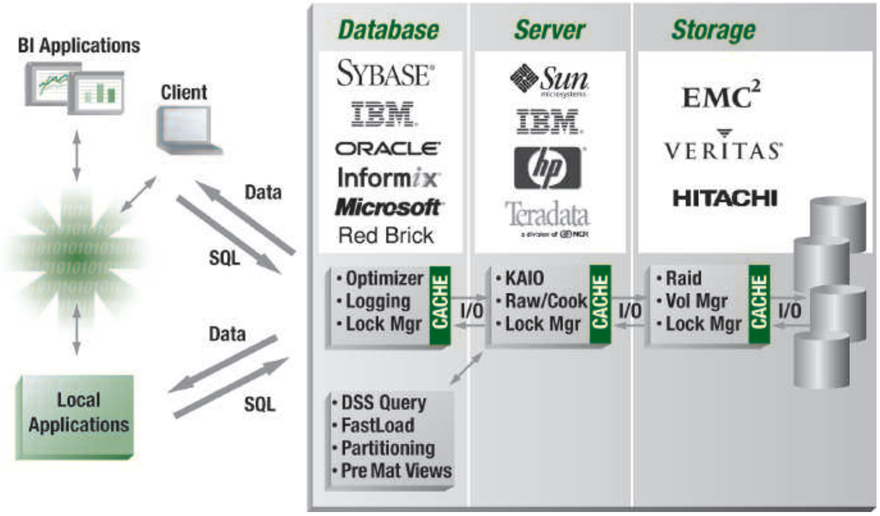
This paper compares the architecture of the Netezza Performance Server (NPS®) appliances to other data warehouse technologies used today. It addresses the inherent limitations of traditional architectures that result in bottlenecks when processing complex queries on enormous quantities of data. It then examines the methods used by some well-known vendors, which are all variations of the same inherently limited design. Finally, it shows how Netezza's approach is fundamentally different, from the affordability of its data warehouse appliance to the lightning speed of Intelligent Query Streaming™ technology. Netezza customers gain real-time intelligence based on almost unimaginable amounts of data – fundamentally changing the way they leverage their data warehouse to make better decisions.

# Traditional Data Warehouse Systems

## The High Cost of General-Purpose Solutions

A data warehouse consists of three main elements – server, storage and database software – interacting with external systems to acquire raw data, receive query instructions and return results. In traditional systems, these core elements are a patchwork of general-purpose products from multiple suppliers, configured to function together as a data warehouse. These solutions are built from systems that were originally developed for on-line transaction processing (OLTP). They were not designed to handle large and complex Business Intelligence (BI) analysis, and have inherent constraints that result in limited performance and high costs of acquisition and ownership.

*Figure 1*



## Barriers to Performance

- **General-purpose servers:** These are the same computers used in data centers as web servers, email servers or application servers. They use architectures that originated in the eighties and nineties for OLTP applications, and are designed for efficient, RAM-based operations on individual data elements (such as the contents of a field). They are not designed to run quickly or efficiently as part of a data warehouse solution, where processing can involve extremely large sets of data, and query requirements are quite different.
- **General-purpose storage:** Most general-purpose storage arrays require time-consuming, careful synchronization of loaders and data striping mechanisms to ensure that data is distributed so that it can be accessed efficiently by business intelligence users. Finding the specialized expertise to properly configure the storage system usually means engaging a costly professional services firm. (Technical services are a lucrative "cash cow" for suppliers of traditional data warehouse systems.)
- **General-purpose database:** Obtaining maximum performance from a data warehouse requires a close marriage between its software and hardware architectures. The full power of general-purpose database management systems (DBMS) such as DB2 or Oracle is lost when they are simply embedded within general-purpose hardware and used for data warehousing. The software is not designed to extract optimal performance out of even the most advanced servers and storage.

## Barriers to Efficiency

The sheer inefficiency of patchwork solutions creates cost, complexity and waste:

- **Inefficient use of administrators' time:** Configuring storage devices from EMC (for example), servers from HP or IBM, and database management software from Oracle for the demands of tera-scale query processing requires a great deal of time from system and database administrators. Regardless of administrators' efforts, delivering acceptable performance becomes a losing battle as scalability limits are reached.
- **Inefficient installation:** Patchwork data warehouse solutions invariably mean more hardware, less reliability, higher power requirements and wasted floor space. Set-up often takes weeks, involving testing, debugging and fine-tuning of system parameters.
- **Inefficient system management:** Patchwork solutions become increasingly difficult to manage as core products evolve, especially as vendors upgrade their offerings at different times.
- **Inefficient data flow:** Query processing on a general-purpose system is extremely cumbersome, requiring the shuttling of huge quantities of data from storage to memory – multiple times for complex queries. This topic is explored in more detail in the "Traditional Data Flow" section.

## Inability to Scale

There are a number of dimensions of system scaling in an enterprise data warehouse. Each of these dimensions is growing faster than "Moore's Law," which states that processor speed doubles every 18 months. This means that general-purpose hardware and software solutions that advance at the rate of Moore's Law will be unable to keep pace with growing user and data demands. The key factors affecting system scaling in a data warehouse setting include:

- **Data volumes:** Data volumes in many organizations double in less than a year, and sometimes much sooner – far exceeding the pace of Moore's Law. The implications can be seen in companies that spend millions of dollars in upgrades, only to be immediately swamped as data volumes continue to soar.
- **Complexity of queries:** The use of enterprise data warehousing is growing continually more sophisticated – moving from a historical view in the form of reports, to analysis of up-to-the-minute data and finally to strategic predictive analytics. Increasingly, this requires analysis of detailed, granular data, where richness of the data may provide key information glossed over in data aggregations or summaries. This additional complexity in analysis increases the processing burden on the data warehouse.
- **Real-time response:** Increasingly, the need for access to and analysis of data is toward real time or near real time ("right time"). Driven by the exploding use of sub-transactional data for everything from financial trading to credit card users' purchasing patterns to telecommunications call data records, the need to analyze and detect timely patterns of business opportunities or fraudulent behavior is critical.
- **Number of users:** Enterprises continue to allow growing numbers of users (both internal employees and external partners) to query the data warehouse for vital business information.

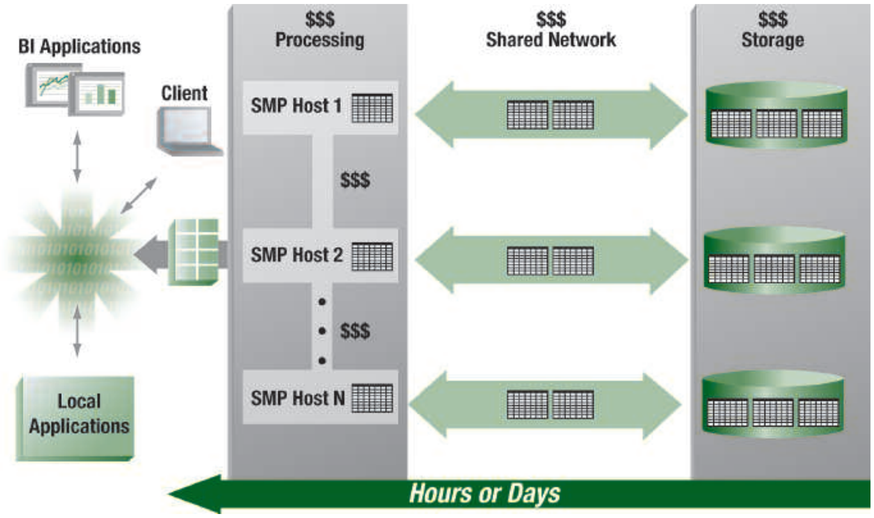
## Traditional Data Flow

### Bringing the Data to the Query

Traditional database platforms operate by reading data off disk, bringing it across an I/O interconnection and loading it into memory for processing. Data normally flows smoothly for routine OLTP applications (e.g. processing invoices or looking up patient records) that are characterized by random I/O operations on individual data elements. However, moving data across multiple backplanes and I/O channels works poorly when the amount of data to be queried is enormous, when the query involves complex joins requiring multi-phase processing and when data is changing rapidly.

Unlike OLTP, data warehousing is all about data shuffling: moving large quantities of data through the system's analysis and processing engine as efficiently as possible, with a minimum of internal thrashing. Where OLTP systems might be optimized to reduce data latency, data warehouse systems are typically much more concerned with data throughput. As a result, bringing OLTP-optimized system designs to build a data warehouse results in greatly reduced performance.

*Figure 2*



For example, a complex join or other complex query may require a number of processing steps. Consider the sheer inefficiency of delivering multiple enormous tables (billions of rows) off disk, across the network and into memory for processing by the DBMS – all to perform one step. The partial results then have to be moved ("materialized") back to disk in a temporary storage location prior to bringing in another huge bundle of data for the next step. These massive flows of data overwhelm shared and limited resources including disks, I/O buses (LAN, SAN, gigabit Ethernet, Fibre Channel, etc) and especially backplane interconnects.

All large-scale legacy data warehouse systems used today (including those from IBM, Oracle, Teradata, and SQL Server) are burdened with these fundamental limitations to performance. Accordingly, as traditional data warehouses grow, they become "victims of their own success" – with more users, data and queries, response slows to unacceptable levels and user frustration becomes inevitable. To squeeze out more performance, DBMS systems typically employ complex schemas, indices, aggregates and advanced partitions to attempt to limit the amount of data required for movement and analysis within the architecture.

### Compromises and Trade-offs

Configuring general-purpose processors, storage and database software for a purpose for which they were not designed involves compromises and optimization challenges. How long are users willing to wait for a query to complete? How much granularity can be sacrificed through aggregation or averaging in order to return results within a reasonable time? How does the IT department's high-availability strategy affect query response? Managing these conflicting requirements can consume a great deal of database administration time, yet reaching an acceptable compromise is often impossible.

Data warehouse systems built with general-purpose components usually require careful modeling of the database schema to accommodate performance constraints. The result is that system users don't get the level of information they need. Forcing users into aggregated data is one example, improving response time but at the expense of data depth on which analysis is based. This compromise approach has actually influenced "best practices" in data warehousing, which call for a multi-level structure, with an enterprise warehouse of fine-grained data feeding smaller, summarized schema and data marts. Another common method involves extensive use of indexing or cubes to reduce the working set of data. However, this requires ongoing tuning to keep indices current as data evolves and grows.

## Traditional Multiprocessing Architectures

### SMP vs MPP

An enterprise data warehouse will use some form of multiprocessing architecture, as there is just too much information for a single processor and system backplane to handle. The two main forms, Symmetrical Multiprocessing (SMP) and Massively Parallel Processing (MPP) were originally developed as competing architectures in the eighties and nineties, and have served as models for generations of high-performance computing systems ever since.

SMP systems consist of several processors, each with its own memory cache. The processors constitute a pool of computation resources, on which threads of code are automatically distributed by the operating system for execution. Load is balanced across the processors, so one doesn't sit idle while another is overloaded. Resources such as memory and the I/O system are shared by and are equally accessible to each of the processors. The single coherent bank of memory is useful for efficiently sharing data among tasks. The strength of SMP lies in its processing power; however, the architecture is limited in its ability to move large amounts of data as required in data warehousing and business intelligence applications.

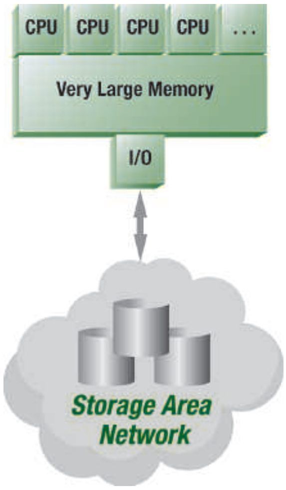
In general, MPP systems consist of very large numbers of processors that are loosely coupled. Each processor has its own memory, backplane and storage, and runs its own operating system. The no-shared-resources approach of pure MPP systems allows nearly linear scalability, to the extent that the software can take advantage of it and is parallelizable. High availability is another advantage – when one node fails, another can take over (again, if accommodated by the software architecture).

Pure MPP systems are rare in practice due to the costs of additional memory and I/O components, as well as the administrative challenges in setting up and managing many semi-independent systems. Typical MPP systems are implemented virtually in clusters of SMPs, often with some sharing of I/O resources. The intent is to preserve some of the performance and scalability advantages of MPP while reducing costs and administration time.

## Multiprocessing Variations for Data Warehousing

Traditional large data warehouse systems are typically based on one of the following variations of the SMP and MPP forms. Both methods were developed to deliver high-performance for general-purpose computing, but have major drawbacks when used to process queries requiring massive data movement.

*Figure 3*



## Large-Scale SMP

A small SMP system consisting of a few processors is not capable of handling large-scale query processing. However, larger SMP systems with additional processors and shared memory are available that deliver much higher computing power, and large-scale SMP systems are frequently deployed for data warehousing.

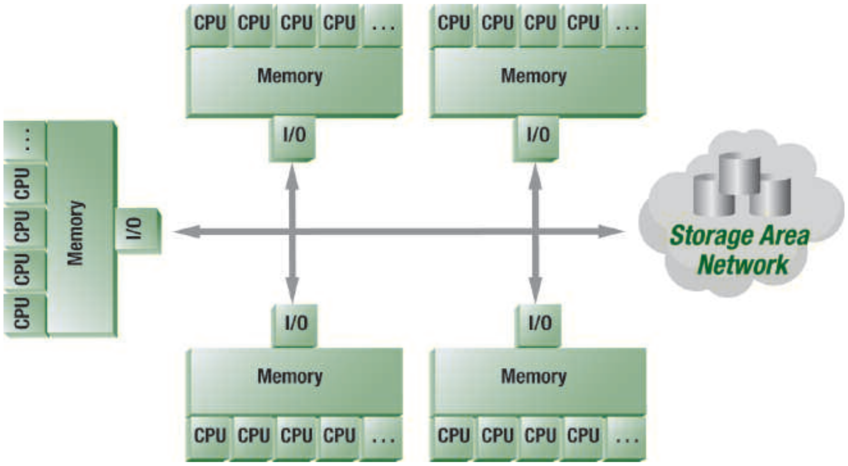
As shown in the diagram, dozens of processors are sharing memory and storage. When data volumes are huge and growing quickly, systems based on SMP architectures tend to outgrow their memory, backplane and I/O resources. As processors take turns accessing massive amounts of data in memory, the memory bus becomes a bottleneck that results in poor performance.

Because memory bus bandwidth is limited, increasing the number of processors and RAM to handle the workload becomes futile as the bus becomes saturated. I/O bus bandwidth is also limited, and can become congested as the amount of data sent from the storage area network to process a query increases. Hence, a traditional disadvantage of SMP systems is less-than-linear scalability and a progressive decline in performance as the system grows.

### MPP on Clustered SMP

The example in the diagram consists of small SMP clusters operating in parallel while sharing a storage area network and management structure. Each CPU within an SMP node shares RAM with its neighbors, as well as access to the storage network over a shared I/O bus. A number of vendors offer database solutions using this approach, including Teradata and the IBM DB2 Integrated Cluster Environment (ICE).

*Figure 4*



The resource-sharing built into this approach imposes a bottleneck that limits performance and scalability. As the diagram shows, SMP nodes are contending for access to storage over a common I/O bus. This architecture is intended for traditional database applications – not for scenarios where enormous amounts of data are pushed through a shared pipeline. Even a purely parallel architecture with separate I/O paths is not designed to handle terabytes of data flowing from storage to an SMP cluster for processing.

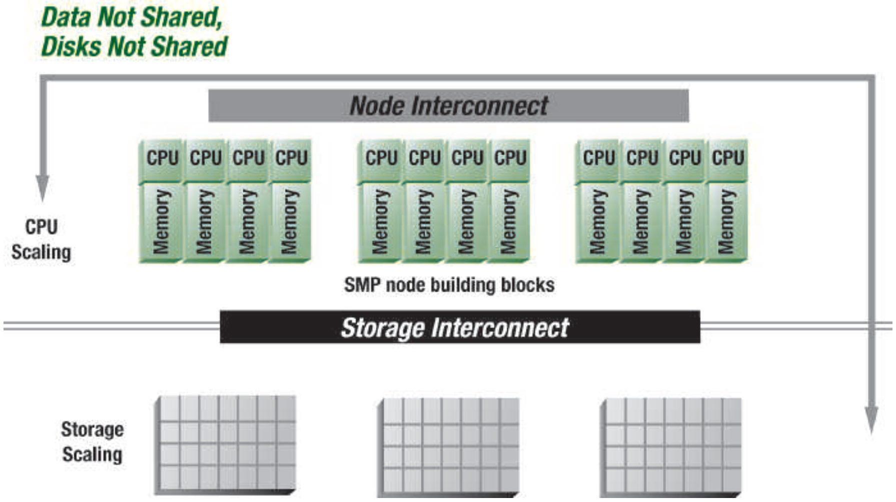
## Mainstream Examples of Traditional Architectures

This section examines three architectures used today by well-known data warehouse solutions. All three are based on a hybrid combination of MPP on SMP clusters, but vary in their approach to sharing data and storage resources between MPP nodes. The solutions also vary in their degree of integration, with storage and in some cases servers provided by third-parties. All three share inherent limitations in performance, resulting in heavy dependence on complex and costly administration and tuning.

## “Shared Nothing” MPP

In a “shared-nothing” architecture, processor-disk pairs operating in parallel divide the workload to execute queries over large sets of data. Teradata follows this approach. Each processor communicates with its associated disk drive to get raw data and perform calculations. One processor is assigned to collect the intermediate results and assemble the query response for delivery back to the requesting application.

*Figure 5*



With no contention for resources between MPP nodes, this architecture does allow for scalability to tera-scale database sizes. A major weakness of this architecture, however, is that it requires significant movement of data from disks to processors for BI queries. While the processor-disk pairs operate independently, they typically share a proprietary common interconnect which becomes clogged with traffic, adversely affecting response time. A typical scenario involves moving a 64K block of data, of which only about 1K is required to respond to the SQL statement. The balance is overhead – unrelated data, project columns, join columns and other extraneous material that is wrapped around the relevant data and has to be filtered out by the processor.

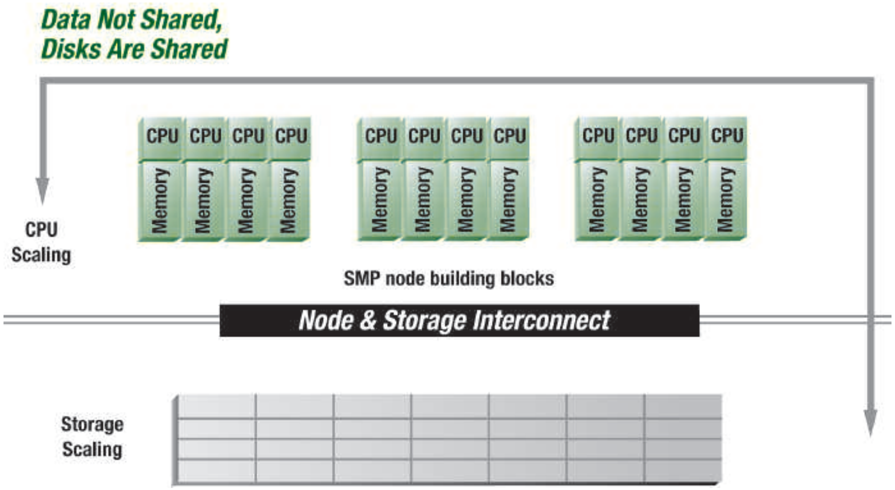
This high overhead has consequences because of a classical problem that occurs when MPP architectures are used for large-scale query processing: the internal backplanes, busses and I/O connections between processor and storage cannot handle the amount of traffic. The inability of data transfer speeds to keep pace with growing data volumes creates a performance bottleneck that inhibits scalability. Teradata has admitted publicly that this trend of diminishing returns makes it difficult to take advantage of abundant storage capacity while maintaining acceptable performance levels.[1]

Performance limitations are compounded when systems are based on legacy components. For example, in developing its data warehousing offering, Teradata opted to use its own generic servers, with storage provided by third parties. In order to squeeze performance out of this older system architecture, the solution relies heavily on system tuning through complex choices of primary and secondary indexes and table denormalization as well as space allocation. However, the relationship between complex indexing and query speed is difficult to optimize, and highly dependent on the data and query. As a result, indices are often mis-configured, increasing query response time instead of decreasing it. (As an indication of the complexity and expected time commitment, Teradata devotes a four-day training course and nearly 300 pages of documentation specifically to indexing and tuning of the system.)

## Separate Data, Shared Storage

In this approach, processors operating in parallel share the same storage media, which is partitioned so that processors don't contend for the same data. Systems sharing common storage are less costly than a shared-nothing architecture, where each processor has its own dedicated storage device. However, performance suffers from the classical disadvantage of MPP architectures: system resources are overwhelmed as query data is transferred from disk to the processors. In addition, the common storage leads to scalability issues as data volume grows.

Figure 6



IBM has developed a data warehousing solution based on this design using its DB2 database management system. This is a hardware-independent solution, consisting of a core database supported by a variety of server and storage options. The independence comes at a cost, with the range of products supported creating a complex mix of choices when designing a solution. For example, storage options include SANs or direct attached storage, requiring customers to understand the I/O dependency of their warehouse on completely different types of storage architectures.

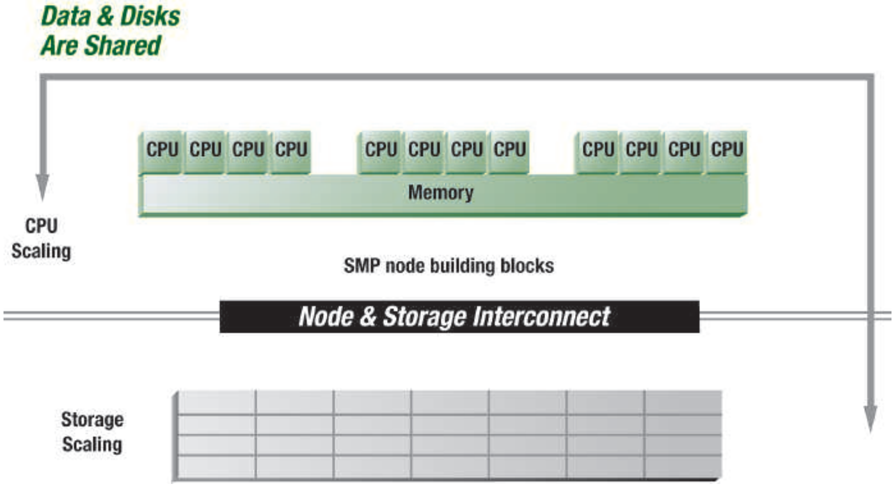
A mature RDBMS such as DB2 comes with elaborate tools to manage the database and its network of storage and processors. These tools are used extensively to tune and optimize the disparate system elements for acceptable query performance. Heavy administrative workload can be expected, with complexity growing as SMP cluster nodes are added.

Another way to compensate for performance constraints is to use indexing to limit the amount of data examined in a query. While DB2 provides a number of indexing strategies for warehouse applications, the combination of partitioning and indexing across distributed nodes makes configuration and loading considerably more complex. The range of tuning and management challenges often proves too daunting for most customers, requiring assistance from a high-priced professional services firm and dramatically altering the cost-benefit equation of the system.

## Shared Data and Storage

In this design, multiple processors operating in parallel access shared data residing on a common storage system. A lock manager is used to prevent simultaneous access to the same data by multiple query processes. Access to shared data is coordinated via messaging between processes and the lock manager. Oracle has built a solution based on this approach using its 9i or 10g Real Application Cluster (RAC) relational database management system.

*Figure 7*



In theory, the shared data architecture means that DBAs do not have to worry about partitioning strategies that may affect query performance. However, like the two previous designs, this approach requires the transfer of massive amounts of data from storage to processors. The problem is exacerbated by the shared data architecture, where contention issues further limit performance and scalability. For example, the locking and caching mechanisms used by Oracle to prevent processor contention effectively put a ceiling on the data scalability of its RAC.

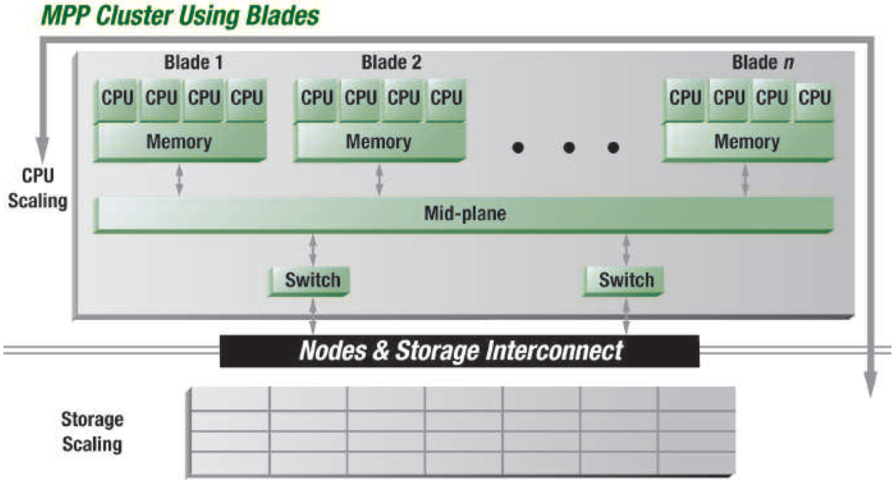
Oracle themselves recommend that users deploy partitions and indices to help improve query performance, thus eliminating the simplicity that the shared data architecture was meant to achieve. These indexing schemes and their interactions with partitioning greatly increase set-up and maintenance complexity.

As with DB2, the Oracle 9i and 10g RAC solutions are hardware and operating system-independent, capable of running on a variety of HP-UX, AIX, Linux and Windows servers. The number of hardware platforms and OS choices can result in multi-week installation times, requiring assembly, testing, debugging and fine-tuning of system parameters.

## Blade Servers

Blade servers provide a new level of high-density computing, packing an enormous amount of computing power into a compact frame. Each blade has its own collection of processors, memory and I/O capability – in short, “a server on a card.” Dozens of blades are installed in a single chassis sharing storage, network and power resources. The result is an integrated, consolidated infrastructure for high-performance computing, with a common management framework providing control as a single virtual system.

Figure 8



A number of vendors offer data warehouse solutions based on blade technology. In a typical scenario, each blade functions as an SMP cluster of processors and shared RAM, within a matrix of blades operating in parallel. This amounts to a tightly consolidated version of the MPP on clustered SMP architecture described earlier.

However, the blade architecture contains elements that work to its disadvantage for the specialized requirements of BI query processing. In the example shown in the diagram, each blade communicates over the system midplane to a storage area network – a route shared with all the other blades in the rack. Accordingly, blades suffer from the same problem as traditional architectures configured for a data warehouse – massive amounts of data have to be delivered from storage to processors over a common I/O pathway. With these traffic volumes, bottlenecks occur as individual blades contend for access to shared resources.

It is also critical that software used in data warehousing be written to exploit the benefits of the hardware architecture on which it operates. In the case of blade servers, simply taking legacy OLTP-optimized software and running it on a blade processing architecture will generally not result in added performance. In fact, the case could be made that because blade servers consolidate processing into a single shelf or rack with shared backplanes, I/O channels and the like, deploying legacy software may exacerbate the bottleneck issues seen previously.

In short, today's first generation blade servers typically provide a general-purpose computing platform with better form-factor and cost profiles than legacy SMP and clustered-SMP/MPP implementations, but still suffer from the same inefficiencies and complexity of traditional data warehouse solutions. To truly harness the processing power of blade technology, blade architectures must evolve to become optimized for specific applications. An example in a different industry segment is the "Google Search Appliance," a custom-designed blade server developed to enable ultra-high speed content searches for the enterprise. Similarly, the next section will discuss how the innovative intelligent storage node architecture developed by Netezza enables dramatic improvements in BI performance and cost of ownership.

# Netezza's Data Warehouse Appliance

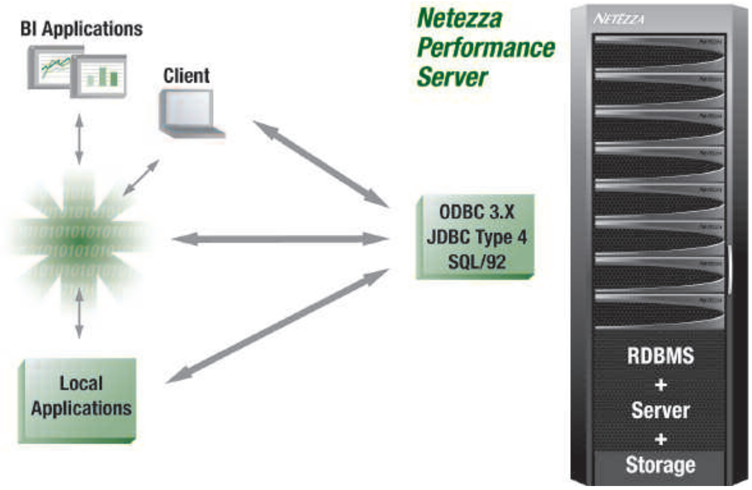
## Performance, Value, Simplicity

In developing its Netezza Performance Server (NPS®) system, Netezza took a fresh look at the challenges of tera-scale data warehousing and created an architecture that eliminates the barriers to performance of traditional systems. The NPS system is a data warehouse appliance – a fully integrated device built for a single purpose: to enable real-time business intelligence and analytics on terabytes of data.

The NPS systems combine server, storage and database in a single scalable platform based on open standards and commodity components. The architecture, rather than expensive, proprietary components, provides the dramatic performance advantage – ten to one hundred times faster than other data warehousing systems. The NPS system leverages commodity components throughout, delivering a huge cost advantage – half the cost of competitive systems.

The simplicity of the Netezza approach also eliminates the high operating costs of general-purpose systems adapted for data warehousing. Its "load and go" implementation process takes hours, not weeks; and there's no need for intensive database administration and system management.

*Figure 9*



# Data Flow - The Netezza Way

## Bringing the Query to the Data

The architecture of the NPS appliance is built upon two guiding principles:

- Performance and scalability goals can be met using elements of both SMP and MPP, applying each method where it is best suited to meet the specific needs of BI applications operating on terabytes of data. Netezza has named this architectural approach Asymmetric Massively Parallel Processing™ (AMPP™).
- Moving processing intelligence to a record stream adjacent to storage produces much better performance and scalability than the traditional approach of moving sets of records to a processor. This Netezza innovation is called Intelligent Query Streaming™ technology.

By putting these two principles into practice, the result is tremendous real-time performance and scalability at a fraction of the cost of other systems on the market.

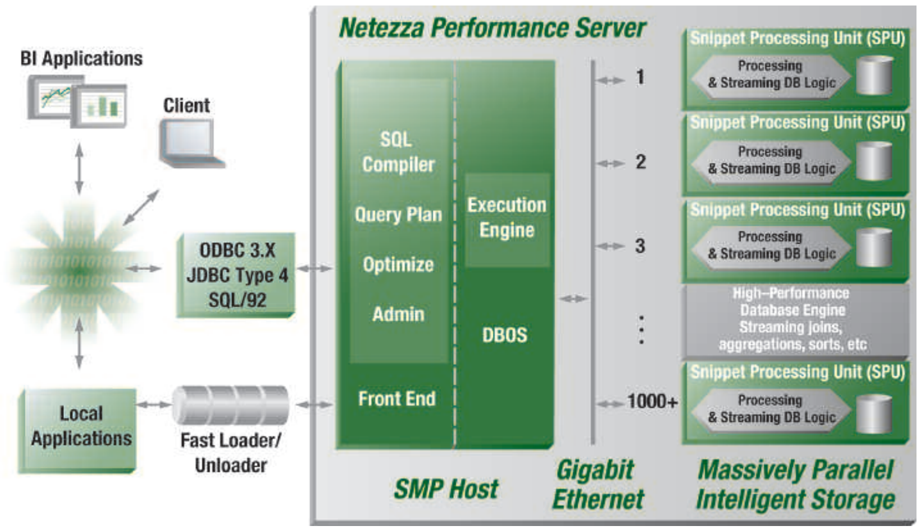


*Figure 10*

Netezza's AMPP architecture is a two-tiered system designed to handle very large queries from multiple users. The first tier is a high-performance Linux SMP host. (A second host is available for fully-redundant, dual-host configurations.) The host compiles queries received from BI applications, and generates query execution plans. It then divides a query into a sequence of sub-tasks, or *snippets*, that can be executed in parallel, and distributes the snippets to the second tier for execution. The host returns the final results to the requesting application.

The second tier consists of dozens to hundreds or thousands of Snippet Processing Units (SPUs) operating in parallel. Each SPU is an intelligent query processing and storage node, and consists of a powerful commodity processor, dedicated memory, a disk drive and a field-programmable disk controller with hard-wired logic to manage data flows and process queries at the disk level. The massively parallel, shared-nothing SPU blades provide the performance advantage of MPP.

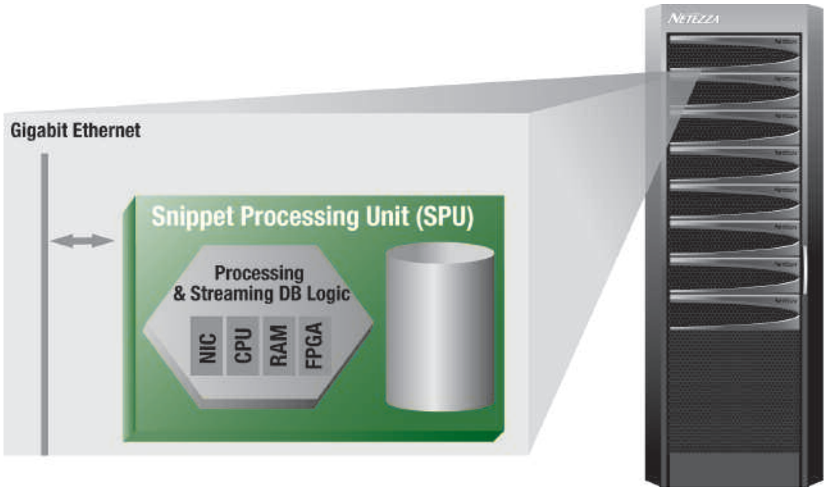
Nearly all query processing is done at the SPU level, with each SPU operating on its portion of the database. All operations that lend themselves easily to parallel processing (sometimes referred to as "embarrassingly parallel") including record operations, parsing, filtering, projecting, interlocking and logging, are performed by the SPU nodes, significantly reducing the amount of data required to be moved within the system. Operations on sets of intermediate results, such as sorts, joins and aggregates, are executed primarily on the SPUs, but can also be done on the host, depending on the processing cost of that operation.

The real power of the Netezza solution lies in the strength of its software to optimize performance and throughput. While the SPUs respond to requests from the host, they are highly autonomous, performing their own scheduling, storage management, transaction management, concurrency control and replication. This significant degree of autonomy reduces the coordination requirements on the host. It also relieves DBAs from low-level but time-consuming maintenance tasks.

## Intelligent Query Streaming™ Technology

A second key approach in the NPS system architecture is Intelligent Query Streaming technology, which greatly reduces the data traffic among SPU nodes, and between SPU nodes and the SMP host. The design streamlines the flow of information by placing silicon processors right next to the storage device. Rather than moving data into memory or across the network for processing, the technology intelligently filters records as they stream off the disk, delivering only the relevant information for each query. By performing this first level processing right at the disk, Netezza is at least ten times faster than conventional systems, with disk access speed providing the only limiting factor.

*Figure 11*



Intelligent Query Streaming is performed on each SPU by a Field-Programmable Gate Array (FPGA) chip that functions as the disk controller, and is also capable of basic processing as data is read off the disk. The system is able to run critical database query functions such as parsing, filtering and projecting at full disk reading speed, while maintaining full ACID (Atomicity, Consistency, Isolation, and Durability) transactional operations of the database. Data flows from disk to memory in a single laminar stream, rather than as a series of disjointed steps that require materializing partial results.

With the Netezza approach, the pathways used by traditional architectures to deliver data to the host are streamlined and shortened. Because the SQL is “understood” by the disk drive in a Netezza system, there is far less reliance on CPUs, data modeling or bandwidth for performance:

- The storage interconnect, a bottleneck on traditional systems, is eliminated by direct attached storage – data streams off the SPU disk and straight into the FPGA for initial query filtering.
- Intermediate query tasks are performed in parallel on the SPUs, where streaming processing sharply reduces CPU workload.
- The gigabit Ethernet network connecting SPUs to the host and each other is used only for transmitting intermediate results, rather than massive amounts of raw data. Network traffic is reduced by approximately two orders of magnitude.
- The I/O bus and memory bus on the host computer are used only for assembling final results, eliminating previous congestion.

# Key Differences and the Netezza Advantage

The Netezza Performance Server appliance offers several fundamental advantages over traditional data warehouse architectures:

## Data Flow

- **NPS appliance:** Netezza's AMPP architecture applies elements of SMP and MPP to deliver high performance for enterprise-scale BI applications. Most processing is handled by the massively parallel snippet processing units, as early in the data flow as possible. This approach of "bringing the query to the data" eliminates extraneous traffic and resulting delays.
- **Traditional systems:** SMP and MPP architectures developed for general-purpose systems (including blade servers) are based on moving data from storage to the processors ("bringing the data to the query"). When performing BI queries of massive databases, the flood of data creates bottlenecks that result in slow (and often unacceptable) response times.

## Storage Connection

- **NPS appliance:** Netezza's Intelligent Query Streaming technology filters out unnecessary information as data streams off the disk, greatly reducing the processing burden downstream. There are no storage interconnects in the traditional sense – the disk controller handling the initial processing is hard-wired to the disk drive. System performance is limited only by disk speed (the NPS system runs at "physics speed").
- **Traditional systems:** Storage system interconnections simply function as a conduit to deliver data from storage to its associated processor. System performance is limited by the capacity of the I/O bus.

## Degree of Integration

- **NPS appliance:** Server, storage and DBMS are integrated in a compact, efficient unit designed specifically for data warehousing. The system installs in hours, not weeks, and deploys quickly with no need for indexing, tuning, physical modeling or other time-consuming tasks. There is one vendor to manage, and none of the unnecessary components, awkward cabling or conflicting parameters that traditionally cause problems with patchwork solutions.
- **Traditional systems:** Patchwork solutions based on general-purpose products mean a myriad of headaches, including multiple vendors to manage, lengthy and difficult implementations, complex tuning, lower reliability, higher power requirements and extra floor space.

## Advantages of the Netezza Architecture

### Performance

By "bringing the query to the data," the NPS appliance delivers at least an order of magnitude performance improvement for BI applications analyzing terabytes of data. Traditional delays are eliminated – analyses that previously took hours now take just seconds. Even "queries from hell" to uncover deeply buried patterns are handled with ease.

### Low Acquisition and Operating Costs

As a purpose-built appliance, the purchase price of the NPS appliance is significantly lower than competing systems. Cost savings are even more attractive over the long term. While the care and feeding of traditional systems often requires several highly paid DBAs or system administrators, an NPS system supporting tens of terabytes is usually managed by a part-time administrator. Instead of partitioning table spaces, designing indices and performing all the other optimization tasks previously required, DBAs can devote their time to developing business-critical analyses that help their companies succeed.

### Linear Scalability

While I/O bottlenecks are commonplace as general-purpose systems scale to accommodate complex queries, additional arrays of snippet processing units can be added to the NPS system without impacting performance. This is because query processing using the NPS architecture involves a minute fraction of the data traffic associated with traditional systems, and because storage and processing are tightly coupled into a single unit. The autonomy of the SPUs creates further conditions for a highly scalable system, allowing SPUs to be added without worrying about coordination with other units. As a result, growing data volumes can be planned for and accommodated without the sudden, unexpected need for costly purchases.

## The Power to Question Everything

General-purpose architectures developed for online transaction processing were not designed for detailed analysis of terabytes of data. Users of traditional systems continue to pay the price – in poor performance, limited scalability and complex administration. Ultimately, the highest price comes from limited and delayed business intelligence: off-target forecasts, lost revenues, missed opportunities.

By providing an architecture built for the specific challenges of tera-scale analytics, the Netezza Performance Server appliance delivers the performance, value and ease-of-use that business users demand and expect. For the growing number of Netezza customers, the benefits are dramatic:

- For a wireless carrier, accelerating the analysis of 120 days of CDR records from six hours to less than 30 minutes, it means capturing millions of dollars through improved billing and more profitable network utilization.
- For a healthcare provider, reducing query time of its two-billion row patient database from five hours to just over a minute, it means the ability to identify the most effective treatments from a cost/benefit perspective for hospitals and patients.
- For an online retailer, reducing query time against 5.4 billion rows from 50 hours to 21 minutes, it means more effective analysis of web site visits in order to adjust promotions.
- For a leading grocery retailer, reducing the query time of a complex market basket analysis report from over three days with many manual processes to a one-step process that completes in less than four hours, it means empowering business users to understand customer purchasing behavior for improved operational efficiency and larger average purchases.

Companies across many different industries benefit from the tremendous speed of Netezza's data warehouse appliance. The exceptional performance of the NPS system is matched only by its remarkable simplicity and ease of use. For users accustomed to the performance constraints and administrative burden of general-purpose systems, there's no going back.

---

[1] "With capacity growing more quickly than disk bandwidth, the bandwidth per GB of storage capacity has actually decreased by 50%. Given this trend, it is challenging to take advantage of abundant storage capacity while maintaining required performance levels." – Ron Yellin, Director of Storage Product Management, Teradata (*Teradata Magazine Online*, Vol. 4, No. 1, 2004).

## About Netezza

Netezza, the global data warehouse appliance market leader, enables enterprises to make all of their data actionable - quickly, simply and affordably. The Netezza Performance Server family of products delivers breakthrough performance, unmatched ease of deployment and operation, and innovative flexibility and scalability at a fraction of the cost of traditional data warehouse solutions. By architecturally integrating database, server and storage within a single appliance, the NPS system delivers 10 to 100 times the performance at half the cost of existing systems. Based in Framingham, Mass., Netezza has offices in Washington, DC, the United Kingdom and Asia Pacific. The Company is backed by leading venture capital firms, including Matrix Partners, Charles River Ventures, Battery Ventures, Orange, Sequoia Capital and Meritech Capital Partners. **For more information about Netezza, please visit www.netezza.com.**

Netezza Corporation : 200 Crossing Boulevard : Framingham, MA : 01702-4480
+1 508 665 6800 tel : +1 508 665 6811 fax : www.netezza.com

NETEZZA
The Power to Question Everything™

© Netezza 2006. All rights reserved. All other company, brand and product names contained herein may be trademarks or registered trademarks of their respective holders.