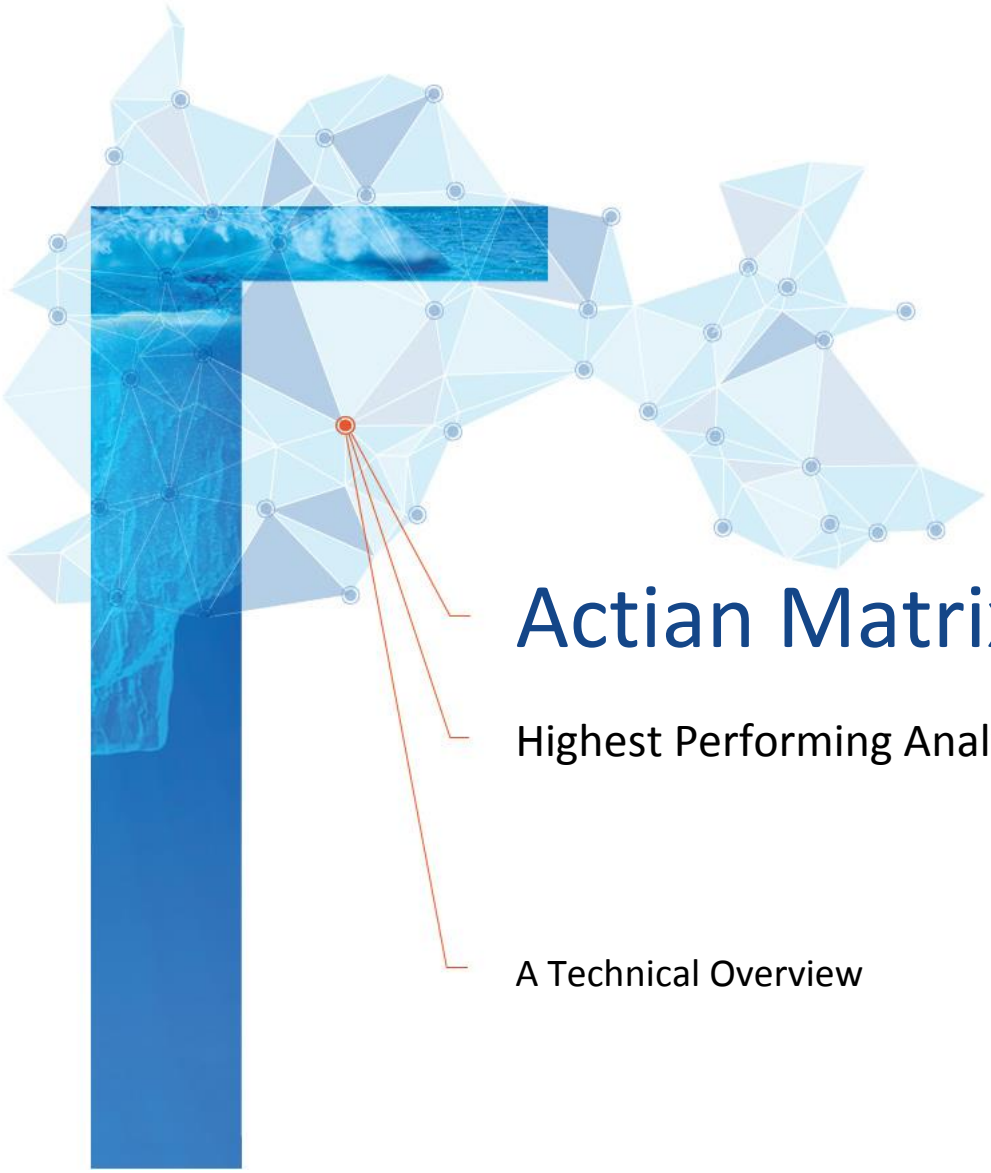


Actian Matrix

Highest Performing Analytics Database

A Technical Overview

Contents

Introduction

The Actian Matrix™ Analytics Database™ is a next generation high performance relational database management system (DBMS) that combines leading-edge innovations with best practices to deliver the fastest, simplest, most cost-effective solution for analytic processing.

Enterprises today use commonly accepted data warehouse tuning techniques such as specialized physical schemas, aggregation tables, indexes and materialized views that inflate data volumes and cause extra administration effort to manage the resulting data redundancy and tuning structures. In contrast, Actian Matrix is designed to deliver the highest performance while alleviating that complexity. The core Actian Matrix design philosophy is to provide an advanced analytic DBMS that delivers the highest possible performance and takes best advantage of the investment protection and raw computing power of industry-standard hardware.

The high-maintenance tuning techniques mentioned above have traditionally been necessary to shore up analytic query performance, but they increase maintenance time and complexity and often do not deliver the desired performance. These workarounds are customarily used to improve the analytic performance of DBMS products that were designed for operational processing (e.g., with row-oriented, shared-memory architectures like Oracle, SQL Server, MySQL, PostgreSQL, et al), but they may be found in use with "purpose built" analytic products as well (e.g., Teradata, IBM Netezza). With Actian Matrix, these types of tuning structures are unnecessary.

Actian Matrix allows companies to rely on a more normalized schema which can reduce data redundancy as compared to other analytic solutions. This heightens extensibility and efficiency, and facilitates adoption of server technology improvements as they occur (e.g., Moore's Law). With schema flexibility, database administrators can devote more time to providing new business applications and serving the ever broadening needs of users, instead of tuning existing applications for small incremental gains in performance.

In customer environments, Actian Matrix continues to outperform all other so-called high performance analytic solutions. In addition to honing the efficient utilization of I/O, CPU and internode communication, each Actian Matrix feature is individually optimized for performance. At the highest level, Actian Matrix is architected as a columnar, compressed, massively parallel relational database management system that is capable of all-in-memory processing, if desired.

This paper describes the high level architecture and key features of the Actian Matrix analytics database. Product collateral is available online at www.actian.com.

System Architecture

At the highest level, Actian Matrix has four main architectural components: A Leader Node ("leader"), Compute Nodes ("compute"), the Parallel Communication Fabric, and an optional Storage Area Network (SAN).

The leader node controls the execution of the compute nodes, and all nodes communicate with each other via the fabric. Leader and compute nodes are standard x86 servers running Linux. Users and applications communicate with the system via the leader by using standard interfaces – ANSI SQL via ODBC/JDBC.

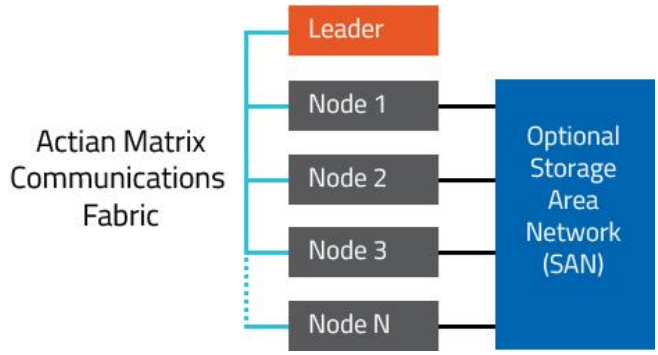


Figure 1: High Level Architecture

Leader Node

The leader sits on the customer's network and is the only Actian Matrix node intended to interface with external applications and the rest of the IT infrastructure. The leader communicates with applications and users via standard ODBC or JDBC, and recognizes ANSI SQL plus Actian Matrix extensions (See "Standard Interfaces and Tools").

A leader is required to manage communication with the compute nodes. The leader is responsible for controlling sessions, parsing and optimizing queries, and scheduling execution of the workload, but the Leader does not participate in data operations.

Architectural workload separation by node type (leader and compute) allows for better throughput optimization – the leader's bandwidth is optimized for outward communication and handling of query overhead so each compute node's bandwidth is dedicated to data operations. Workload separation is important for parallel efficiency because a parallel system is only as fast as its *slowest* unit of parallelism. Adding a "different" task to a single unit of parallelism's workload can therefore impact overall system throughput. This may be mildly consequential in smaller configurations, but becomes more significant as a system grows. For example, assume in a leaderless architecture that adding leader responsibility to a compute node can take up 10% of its bandwidth. In a 10-node cluster, an entire compute node's power is lost as 10 nodes deliver the power of 9. That may be acceptable at 10 nodes, but in a 50-node cluster a leaderless architecture would lose the bandwidth of 5 compute nodes.

Compute Nodes

Compute nodes are the high level component responsible for processing and storing data. Each node stores and manages a subset of the rows of each table. For example, if a table has 1 billion rows and there are 20 compute nodes, then about 50 million rows are distributed to each node. Data is distributed to a particular node based on a hashing algorithm applied to a distribution key, or by round robin. Distribution keys, such as the primary key or other popular join column, are good for even distribution of data, especially when queries will benefit from collocated joins by using the same distribution key. In cases where an inherently balanced distribution key isn't obvious or doesn't exist, round robin distribution can be used to balance the data. By offering multiple methods of data distribution, it is possible to maintain the appropriate balance between data distribution and performance so Actian Matrix can take best advantage of its resources and provide good parallel efficiency.

Actian Matrix performance is driven by how many compute nodes are present. For example, with most applications, a 50-compute node system will perform 5X faster than a 10-compute node system. Therefore, performance and price-performance are inextricably linked on an Actian Matrix system. For highest node performance, Actian Matrix customarily stores data on fast direct-attached storage (mechanical or flash drives onboard the server) to eliminate the connection bottleneck associated with external storage devices (See "Highest Performance with Direct-Attached Storage"). Nodes can also be configured to deliver optimal performance in SAN environments (See "Optional Storage Area Network" and "High Performance in SAN-based Environments").

Compute nodes are logically subdivided into a set of parallel processes called "slices" that include a CPU core, an allocation of memory and portion of each disk. Slices work in parallel regardless of the work they are processing. When loading, slices parse the data into columns, then sort, compress and write the data to disk. Slices communicate with other slices via the fabric, but they are not directly accessed by end user applications.

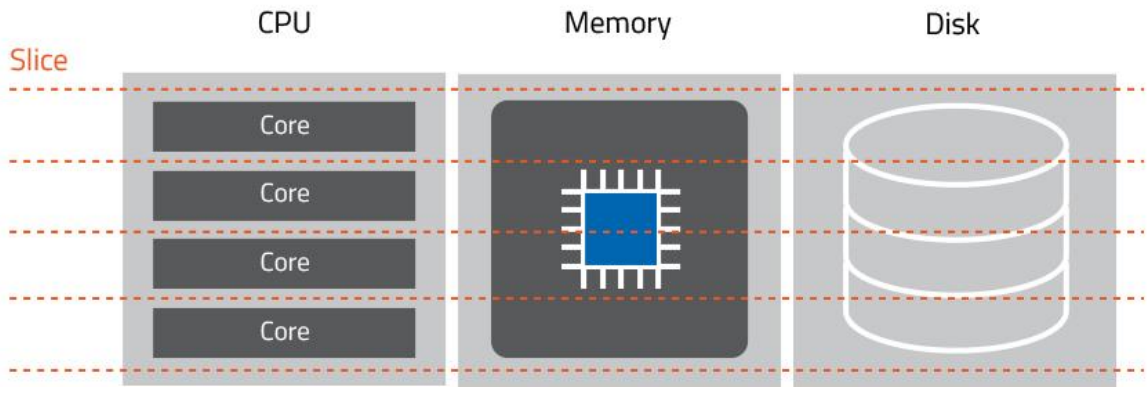


Figure 2: Compute Node Slides

Communication Fabric

The Actian Matrix Communication Fabric is a low cost, high performance fabric based on standard, ubiquitous, Gigabit Ethernet (GbE) and standard multi-port switches that have full crossbar support. It uses a custom protocol (See "Actian Matrix Interconnect Protocol") to enable highly efficient communication among each of the nodes (leader and compute). It delivers maximum interconnect performance because it is specifically designed for how traffic moves in a complex, parallel database environment (e.g., large intermediate result sets, data redistribution) and therefore uses multiple links simultaneously running multiple data streams. The fabric is implemented internally as multiple independent networks all working on behalf of the database, and while at least two GbE fabrics are required for high availability, Actian Matrix will utilize as many fabrics as are available for increased performance (See also "Shared-Nothing Massively Parallel Processing (MPP)", "Actian Matrix Interconnect Protocol" and "Complex Join Processing").

Optional Storage Area Network (SAN)

While Actian Matrix is by default configured to use fast direct-attached storage (mechanical or flash drives) within the compute nodes, it also offers patent-pending features that enable differentiated performance in SAN-based configurations. Actian Matrix can also take advantage of SAN enterprise readiness features such as snap-shot, backup and restore. (See "High Availability").

Actian Matrix Features

Actian Matrix features can be conveniently grouped into three high-level categories for discussion:

- Performance (including Scalability)
- High Availability
- Solution Simplicity

Performance

The most distinctive characteristic of the Actian Matrix system is performance. The database is built for the highest analytic performance, above all other considerations, and outperforms even highly-optimized proprietary analytic computing environments. It consistently achieves gains of 50X or more in customer scenarios that span many analytic processing needs across many industry sectors (telecommunications, retail, online, marketing, information services and so on).

Actian Matrix is configurable to run in two modes: a disk-based mode, or an all-in-memory mode. Many organizations run their analytic platforms as disk-based environments, where data is retrieved from disk to answer business questions. By reducing the amount of data retrieved using both data compression and parallel processing, Actian Matrix delivers a tremendous performance improvement over typical disk-based DBMS configurations. In addition, Actian Matrix offers an all-in-memory configuration where the data is queried from memory and not scanned from disk (though the data persists on disk to be initialized into memory). This configuration eliminates data movement bottlenecks from comparatively slow disks and provides even greater performance.

Actian Matrix has proven its record-breaking performance and price-performance in customer benchmarks and in audited, industry-standard analytic benchmarks administered by the Transaction Processing Performance Council (www.tpc.org). Industry-standard benchmarks complement customer benchmarks as a useful means for technology evaluators to select platforms for further evaluation.

Actian ushered in a new era of performance expectation when it became the first MPP-columnar DBMS vendor to publish TPC-H.

Columnar Orientation

Storing data on disk in a columnar form is widely regarded as delivering better query performance for the bulk of analytic query processing. Actian Matrix's columnar storage implementation is transparent to applications and users – data is retrieved via SQL as with any relational database. The catalyst for column-oriented storage is that functions like scans, joins, aggregations, and sorts are cumbersome for standard row-wise databases to perform because they must read an entire record to use any of its fields. Row-wise storage is highly appropriate for transaction processing, which involves quickly writing or retrieving a whole record by its ID (e.g., a customer account record). During analytic processing, or example, a business analyst is determining the set of customers with a certain characteristic in common, this would result in what is known as a "full table scan" (all columns and all rows must be read).

Row-wise databases offer workarounds to improve performance for read-mostly queries (e.g., indexes, materialized views, summary tables, subsets), but with increased costs. The costs are attributable to increased system and administration resource requirements as well as system size and data redundancy. Specifically, adding indexes and other workarounds reduces load performance and requires attention to design, implement, and maintain the structures. These elements add latency to data availability. According to Forrester's Boris Evelson, these types of structures and redundancy can increase system storage requirements by as much as 8-to-1 or 16-to-1.

Contributors to Data Growth	What Happens and Why?
Moving Raw Data into DBMS	Data volume increases by a factor of two to four by adding indexes, temporary tables aggregate tables, metadata and other structures.
Database Replication	Data is replicated to support distributed architectures, backup, and disaster recovery requirements. Data is replicated by consolidating multiple databases in operational data stores (ODSes) for operational processing and reporting as well as in data warehouse (DWs) and data marts (DMs) for analytical reporting.
Database Optimization	DW and DM are optimized for reporting and analytics, frequently turning third normal form or "normalized" data models into "denormalized" or flattened data models, which replicate data in multiple tables, columns and rows to avoid time-consuming database joins.

Source: Forrester Research

Figure 3: Contributors to Data Growth

In sharp contrast, column-wise storage is very much aligned with analytic queries because they are typically concerned with only a fraction of the columns defined in a table. By reducing the processing to only the columns of interest, Actian Matrix greatly reduces I/O processing, which directly improves performance. Furthermore, Actian Matrix avoids indexes and other data structures because columnar storage, by design, lends itself to the same kind of performance that these structures are intended to provide.

Column vs. Row Example

The following is a simple example using the concept of U.S. Census Data to show how storing data by columns reduces the amount of data to be read. The example assumes there are 300 million people and 100 demographic fields collected for each person. An average size of 10 bytes per attribute yields approximately 300GB of raw, uncompressed data.

*Based on a system with 10 compute nodes, each with 400MB/sec scanning speed.

Illustrative Query of U.S. Census Data: What is the average age of people by State?

	Common Row-based DBMS	Actian Matrix Column-based DBMS
Number of Records	300M people @ 1K bytes / person	300M people @ 1K bytes / person
Scan Volume	300GB	8GB
Efficiency	2%	100%
Performance*	75.0 seconds	1.5 seconds

*Based on a system with 10 comute nodes, each with 400MB/sec scanning speed

Figure 4: Scan Savings of Columnar Storage

The example illustrates how columnar storage provides significant performance improvement independent of compression, parallelism, in-memory processing, and so on. To calculate the average age by state, only two attributes are required. A row-wise database would read all data, but Actian Matrix reads only Age and State—a 50X I/O improvement. When compression is applied, the I/O savings are even higher.

How Does Columnar Orientation Impact Design Considerations?

Actian Matrix's columnar implementation is transparent to applications so designers can focus on application functionality and not physical design. Tables are created with SQL, the same as with any relational database, but the data blocks are written by column instead of row. Actian Matrix automatically converts the records into columns as it writes the data to disk. To an end user, it behaves no differently than a row-wise database, except it's faster.

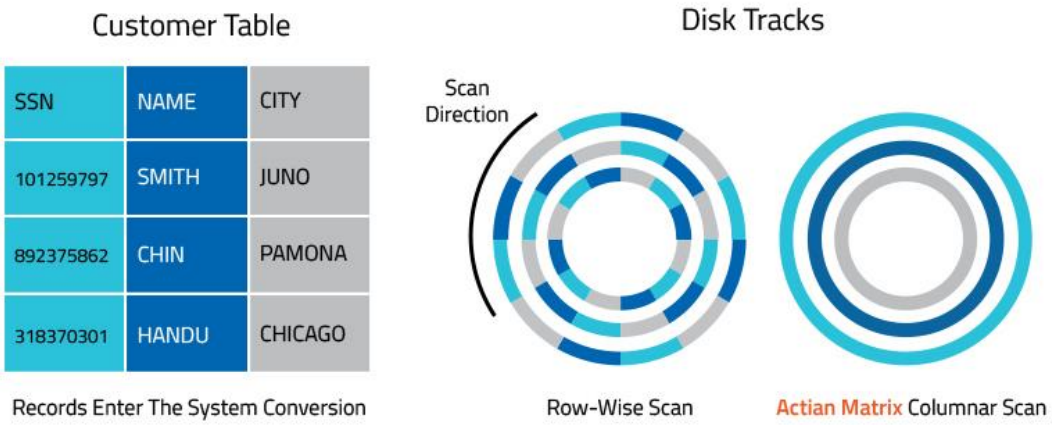
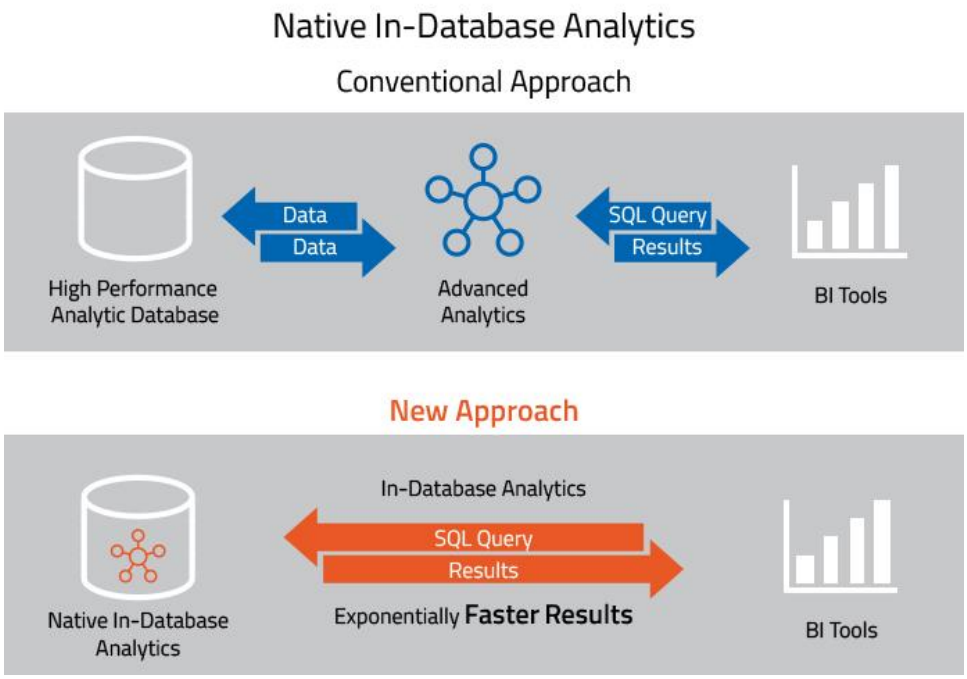


Figure 5: Row Scan vs. Column Scan

Extensible Analytics

Traditional approaches to advanced/custom analytics typically require data to be shipped between different servers. The requirement to read and move the data constantly slows down analytic performance, creating a bottleneck for advanced analysis. Newer approaches package up the advanced analytic functionality and move it to the same server that hosts the relational database. This delivers better performance but still requires the same data to be read/used/processed multiple times.



Actian Matrix enables advanced/custom analytics to be run directly inside the database as a "Class A" object. This ensures that the data is accessed only once, delivering much better performance.

In addition, Actian Matrix's Extensible Analytics delivers better time-to-analysis through the ability to create/port advanced analytics using existing skill sets (such as C/C++). Actian Matrix also enables re-use of the resulting analytic modules by multiple teams across multiple scenarios – no need to "reinvent the wheel". Actian Matrix's approach to Extensible Analytics also supports polymorphic functionality which enables a WORM (Write Once, Reuse Many) approach that provides re-use of the same functionality across multiple input/output formats. For instance, a transpose function can be written once and used to process data in 4x4, 5x5 or 6x6 formats. The exact structure is defined at runtime, rather than requiring specific functions to be written for each format.

Query Compilation

Most analytic processing systems are I/O bound (the bulk of a query's processing time is spent on data retrieval) and therefore the CPUs are not kept fully busy. However, the columnar nature

and specialized fabric protocol in Actian Matrix reduce the I/O cost to the extent that the CPU becomes the next possible performance limitation.

Actian Matrix approaches CPU utilization differently than most other database vendors. Specifically, Actian Matrix compiles queries for far deeper optimization. This level of optimization adds a small up-front cost, but allows most queries to run many times faster in Actian Matrix because it reduces the number of instructions that the CPU must execute. For simple database activities, like aggregation, a CPU can otherwise spend more time determining what action to take than performing the action itself. Actian Matrix's fine-grained compilation takes full advantage of modern CPU technology (e.g., 64-bit, multi-core, etc.). Previously compiled segments are automatically re-used to forego compilation overhead for follow-on queries that may have similar characteristics. Actian Matrix's compilation is transparent to applications and end users.

Query compilation makes sense for workloads that analyze large volumes of data, and especially benefits aggregation queries, which make up the bulk of data analysis. The greatest benefit from compilation is seen on queries that would otherwise be long-running. For example, a thirty-second query in other columnar MPP systems can become a few-second query, or less, in Actian Matrix.

Shared-Nothing Massively Parallel Processing (MPP)

Shared-nothing MPP is a divide-and-conquer processing technique that coordinates processing simultaneously across separate but equal parallel units each with self-contained processing resources (e.g., each unit is a separate server with its own memory, CPU, I/O and interconnect access). Actian Matrix utilizes a unique software architecture that takes advantage of commonly available servers and benefits from the raw power of the fabric, which can sustain rates of 150MB per second per node. This highly efficient MPP grid is implemented on standard hardware to eliminate proprietary lock-in and thereby provide the benefit of sensibly priced components that leverage hardware performance advances as they enter the market. In this divide-and-conquer model, each node operates principally with its own disks' data against its own memory and CPUs. Only intermediate result sets are transferred across the interconnect to be combined into final result sets to be returned to the application.

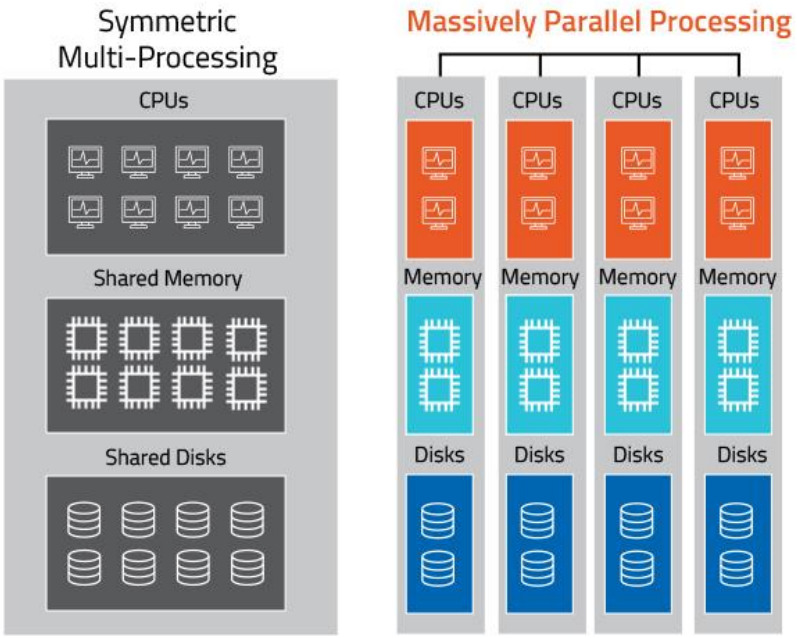


Figure 6: SMP vs. MPP

Actian Interconnect Protocol

The performance of any parallel system is directly tied to the efficiency and scalability of its internode communications. Actian Matrix's MPP implementation is further differentiated by the Actian Interconnect Protocol (PIP), a highly optimized, custom protocol that is specially designed to leverage low-cost, standard Gigabit Ethernet (GbE) with standard switches more efficiently than other parallel database systems. In contrast, MPP interconnects that are based on standard TCP/IP can suffer up to 95% packet loss during heavy inter-node communication activity that is typical of MPP environments – as packet losses increase, sending attempts increase, performance degrades and, eventually, problem queries impact the performance of otherwise well-behaved queries.

Actian Matrix is specifically designed to handle the problematic many-to-many communications found in a parallel system during redistribution of intermediate result sets. During such redistribution every node must send to and receive from every other node. With inferior fabric protocols this would cause chaos as packet losses drain resources. By contrast, Actian Matrix is designed to behave and scale predictably to eliminate this chaos without the workarounds that other databases must employ (e.g., indexes, materialized views, heavy reliance on projections) and which consume maintenance effort and available storage. As with its columnar and compiled features, Actian Matrix's shared-nothing MPP implementation is transparent to applications and users.

Parallel Data Distribution, Query Planning and Execution

In parallel, shared-nothing systems, the distribution of data is balanced to maximize query processing efficiency. In addition, query planning and optimization methods dynamically evaluate

the requirements of a query and the distribution of the data to determine the best processing steps for optimal performance. For example:

Parallel Server Grid

Using multiple high performance servers in a direct-attached or mixed direct- and SAN-attached storage configuration scales system capacity linearly and balances query performance. The grid is easily extended to adapt to capacity and performance changes over time. When servers are added, Actian Matrix rebalances data distribution according to the new configuration.

Intra- and Inter-Query Parallelization

The advanced scheduling features of Actian Matrix ensure that all CPUs, memory, I/O bandwidth, and network switch capacity are fully utilized at all times. Concurrency levels are configurable to balance latency with throughput to suit mixed-use scenarios.

Balancing and Interleaving

Parallel analytic systems must balance resource utilization to maximize throughput in all of the processing of each query. This involves eliminating CPU overhead and maximizing bandwidth while reading, caching, and moving data. Actian Matrix is designed to reduce potential bottlenecks in all of these areas by interleaving movement of data and processing of analytic operations.

Incremental, Linear Scaling

Linear scalability – that is, each added unit of parallelism delivers a full unit's worth of power – is important for predictability, capacity planning and investment protection. Actian Matrix's MPP implementation ensures scaling simplicity. Actian Matrix has demonstrated linear scalability in customer environments and audited industry benchmarks. To get additional Actian Matrix capacity, simply add nodes. When nodes are configured into the system, they are invoked and given their portion of the data. The system is then ready for queries and will run commensurately faster. Expanding the system can yield much better than linear benefit. Queries that had to shuffle intermediate results to disk on smaller systems can now run more in-memory on larger systems. Thus, a 2X larger system can sometimes have a 5X benefit on performance if the smaller system was under configured.

Acceleration vs. Traditional DBMS Platforms

The performance features discussed thus far provide fundamental and measurable performance differentiation for analytic processing, but they are not found in general-purpose Symmetric Multi- Processing (SMP) DBMS platforms (such as Oracle, SQL Server, et al) which were designed originally for transaction processing (aka OLTP).

The performance of Actian Matrix is determined by the nature of the workload. Therefore, this section can only provide a rough rule of thumb based on standard scenarios. Customers can use this to estimate overall performance benefit compared to SMP systems, but the final proof is in loading the data and executing the queries.

Acceleration from Columnar Storage

Columnar acceleration is largely the ratio of the number of columns in the schema as compared to the number of columns that participate in the queries. Aggregation queries rarely use more than a half-dozen fields (columns), but the tables they access often have several hundred columns (e.g., marketing demographics). Therefore, the scan performance can be several hundred times faster for columnar than for row-based scans. Alternatively, clickstream data can consist of only a few dozen columns, so a 10X scan improvement is more common for that type of data. Finally, it is possible, though uncommon, to see very narrow tables for queries that use all of the columns. For those cases, Actian Matrix users will see the same I/O-related performance as row-based systems, because both are ultimately performing the same amount of I/O.

Acceleration from Query Compilation

With fully compiled queries, the number of CPU instructions (clock ticks) to perform fundamental analytic operations like aggregation processing can be reduced by 20X. Dimensional join processing usually sees at least a 10X improvement. Complex arithmetic, restriction and CASE expressions can similarly yield 5X to 20X improvement, depending on their nature. For these operations, the speed of the memory bus is often the limiting factor. Actian Matrix has been publicly benchmarked performing aggregation and joins at full memory bus speeds, which translates into throughput ratings of multiple gigabytes per second per node.

Acceleration from MPP

MPP is what provides a fully scalable system based on inexpensive hardware. In an MPP system, the memory is inexpensive single-port memory coupled to only a few CPUs. SMP systems use proprietary, complex and costly memory subsystems. Similarly, in MPP systems, the disks are "owned" by each node and do not require expensive disk interconnect structures which bottleneck easily. Combining the lower price of the hardware and the better utilization of memory and I/O, MPP solutions typically yield a 10X improvement in price-performance, and scale to dramatically more powerful systems (hundreds of nodes).

Overall Acceleration

Some schemas and queries benefit from all of the accelerations described above. Some benefit from less. Because of the inherent analytic performance in Actian Matrix's overall architecture, Actian Matrix relies less on pre-built tuning assists, (e.g., table projections, etc.) and is therefore well-suited for true ad hoc query. Compared to traditional solutions, Actian Matrix customers have seen performance benefits ranging as high as 4000X for ad hoc queries.

Compression

Adaptive compression is another major contributor to Actian Matrix performance. Compression speeds up I/O (e.g., *n* times compression reduces I/O by *n* times) and reduces traffic on the fabric. Many times data can be processed while it is compressed, which saves compute cycles and boosts query performance. Compression's performance benefit is almost entirely based on the nature of the actual data stored in the tables. In general, fields that are highly redundant from record to record compress very well, and thus better leverage I/O, CPU and interconnect throughput. Similarly, fields with a limited number of distinct values (compared to declared field type) compress quite well. For the most part, highly normalized schemas dominated

by surrogate keys usually offer less compression because normalization is, itself, a compression technique.

Actian Matrix supports an extensive variety of compression methods, including techniques such as run-length encoding (RLE), delta, dictionary, Lempel-Ziv (LZ), and others. Actian Matrix's Compression Analyzer is an efficient tool to take the time and guesswork out of compression analysis so implementers can proceed to "load and go". Compression Analyzer selects the optimal compression scheme for each column based on the data. The administrator can override this selection to compress for "least space" (optimal for disk-based processing) or "best performance" (optimal for all-in-memory processing).

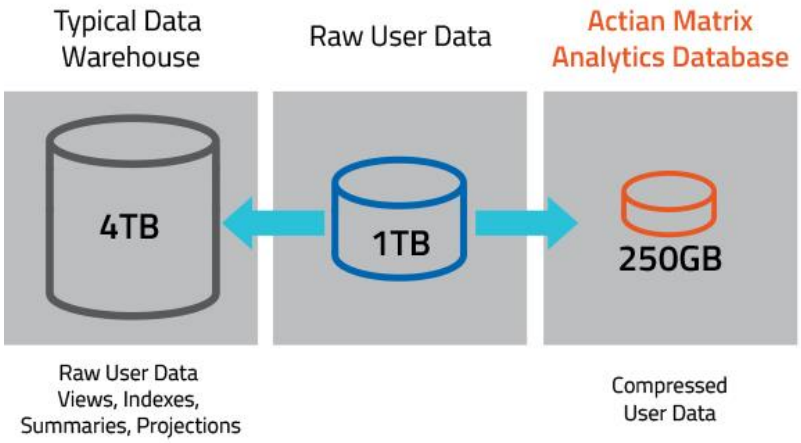


Figure 7: Actian Matrix Compression

Compression also helps shrink the storage footprint required for the system. Consider that a typical warehouse has 8:1 total data to raw data footprint (after indexes, materialized views, summaries, and so on) and, being conservative, consider that compression can reduce the size of the raw data by a factor of four. Therefore, the storage required for 1TB of raw data is reduced to 250GB. The reduced data size enables increased performance, with the capability to run all-in-memory processing, where only memory-and-disk-cache processing was viable before. By eliminating the extraneous data structures and compressing the data, Actian Matrix better enables all-in-memory processing for exceptional performance. Actian Matrix customers see compression ratios averaging 4X and going as high as 12X, depending on the data.

Cost-Based Query Optimizer

Actian Matrix's exceptional query performance begins with the Actian Optimizer Framework, a set of advanced parallel query optimizers built to handle complex queries within the context of Actian Matrix's unique architecture. Specifically, the optimizer is both cost-based and rule-based; has advanced view folding, column pruning, native correlated subquery decorrelation, and query rewrite capability; and is MPP-, columnar-, and compression- aware. It can perform join order optimization for queries with an unlimited number of joins (tested to 1000-way), and evaluates viable alternatives to provide consistently efficient query plans even for scenarios such as outer joins, and correlated sub-queries.

Most optimizers struggle with many-way joins and intermediate result sets – classic analytic query problems. As a competitive advantage, Actian has invested in a robust, world-class, MPP query optimizer because it makes data more accessible.

Complex Join Processing

The need to join large sets of data in complex ways is a defining characteristic of an analytic query. Therefore, efficient complex join processing is a defining characteristic of an analytic DBMS. Rather than be join averse (e.g., rely on the existence of a restricted schema or tuning structures, both of which shift the burden from the DBMS to the customer's designers and ETL processes), Actian Matrix's Optimizer Framework includes a patent-pending design to make good algorithmic choices about how to perform joins most efficiently.

Additionally, the Actian Matrix Communication Fabric also facilitates good join processing because Actian Matrix can rely on its ability to quickly distribute or broadcast join data at query time. Without a robust communication fabric, a parallel DBMS must rely on extensive data replication in order to avoid a broadcast or redistribution. Through smart engineering, Actian Matrix builds into the DBMS platform elegant solutions for tough join processing issues. This allows for ongoing scalability and performance without defaulting to artificial structures, data replication or denormalization.

Query Concurrency

Actian Matrix is designed to maximize overall throughput. The design philosophy is to give each query unfettered access to the resources it needs so it may run as quickly as possible. The number of concurrent queries is configurable, but, conceptually, the approach is to allow many sessions and to throttle the query flow so the system resources are kept fully busy without creating inefficiencies (e.g., swapping) that may limit overall throughput.

Highest Performance with Direct-Attached Storage

Because analytic systems are I/O intensive, the storage architecture is a very important consideration when designing for performance. When implemented with mechanical disk drives, each node benefits fully from the server's own disk controller and I/O bus. This way, the data will flow directly along the server's high speed bus without being bottlenecked by a secondary storage connection. Additionally, mechanical drives are very cost effective and positively impact price-performance. When implemented with solid state flash technology (e.g., Fusion-io), each CPU is directly connected to the drives for ultimate scan performance, typically each flash card is 15X faster than a mechanical drive. Flash drives are more prevalent in high performance environments where capacity and price are secondary.

Using Memory to Boost Performance

In the past ten years, CPU performance has increased by about 30X and memory prices have dropped by approximately 50X. In comparison, disk speeds have only increased by about 2X. This economic fact heavily influences how Actian views best-practice DBMS architecture.

When disks were reasonably fast and memory was precious, large database design orthodoxy was to implement the principle join and aggregation algorithms entirely using disk-based meth-

ods because, in general, the intermediate result sets of these queries could not fit in memory. However, the price per byte of memory has dropped so dramatically that it is now very reasonable to scale a system so that intermediate results can fit in memory.

Actian Matrix decides at runtime whether to use memory-based or disk-based algorithms, based on the behavior of the individual queries. Where possible, Actian Matrix uses memory-based algorithms because running in memory accentuates the performance benefits of its other features. Memory-centric processing positively impacts performance of processing intermediate results and also cost-of-ownership.

Since disks use the most power on seek-type operations, Actian Matrix consumes less power by using memory over disk. Furthermore, memory-centric processing means that the disks are more available for the scan needs of other concurrent queries.

High Performance in Optional SAN-Based Environments

SANs are important data center components for mass storage and enterprise data management, but in MPP environments they are limited by their finite channel throughput. Conversely, an MPP server's direct-attached storage (DAS) is fast and scalable, but is an unmanaged island of storage requiring separate backup and disaster recovery support. Actian Matrix's patent-pending Blended Scan interleaves SAN I/O with DAS I/O to overcome these disadvantages and make better use of the configuration's entire DAS- and SAN-based storage capacity.

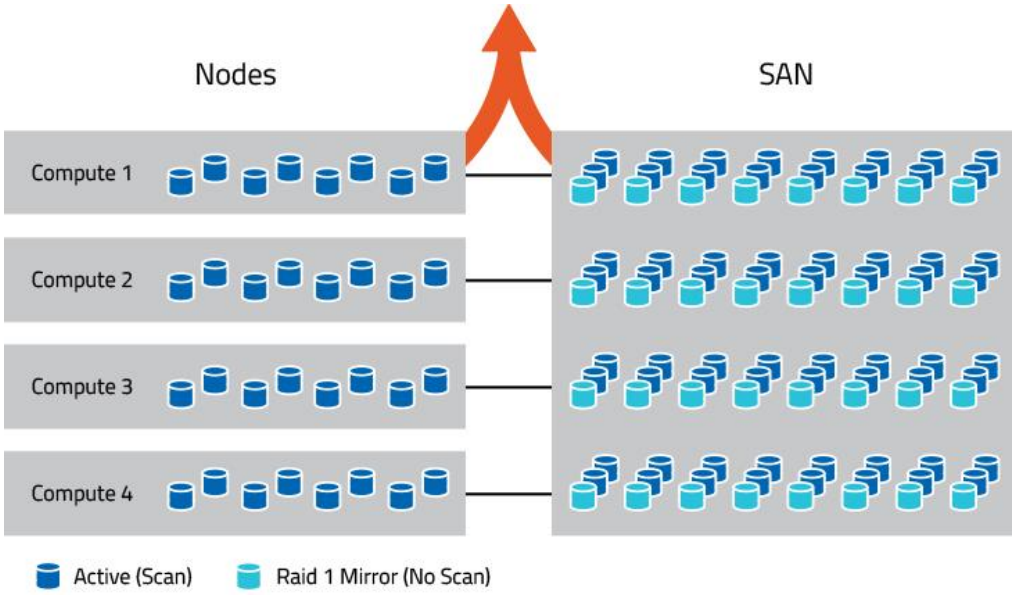


Figure 8: Actian Matrix's Blended Scan

As shown above, Actian Matrix can integrate the two storage environments (DAS and SAN) using the SAN for the server's RAID 1 pair and for a non-equal portion of the total I/O processing. This way Actian Matrix can benefit from the DAS I/O throughput and take full advantage of the SAN I/O throughput, too. Plus, it can now leverage the enterprise data management features of SANs (See "SAN-Based Failover with HSN").

Parallel Loading and Unloading

Data loading is a chronic performance bottleneck in large data warehousing environments due to large data volumes combined with tight load windows. A manually-tuned environment (e.g., with summaries, indexes, and materialized views) exacerbates the problem. To speed up load processing, Actian Matrix avoids tuning structures and comes standard with scalable, high performance loading and unloading.

Actian Matrix offers exceptionally fast loading; load performance has been measured at 9TB per hour. However because it's scalable, higher load rates can easily be achieved. Actian Matrix offers two modes of parallel loading – standard parallel loading and massively parallel loading. Both employ the same process but are initiated differently. Loading occurs in one or two steps depending on whether a table's distribution method is via distribution key or round robin (See "Compute Node"). Source files are commonly CSV or pipe-delimited and can be flat files or ETL processes (e.g. named pipes).

Standard Parallel Load

Standard parallel loading uses the leader node as a round-robin distribution mechanism. When loading initiates, the leader automatically fans out records to the compute nodes to balance them evenly across the system (step 1). If the table uses round robin distribution then it's a one-step process and the data is parsed, validated, sorted, compressed, and written to disk. If the table uses a distribution key, then each node parses and validates the data, then hash distributes the data to the receiving node where it is sorted, compressed, and written to disk (step 2). Standard parallel loading is throttled only by the leader's pass-through bandwidth onto the dual GbE fabric. It can achieve speeds of up to 700GB/hour.

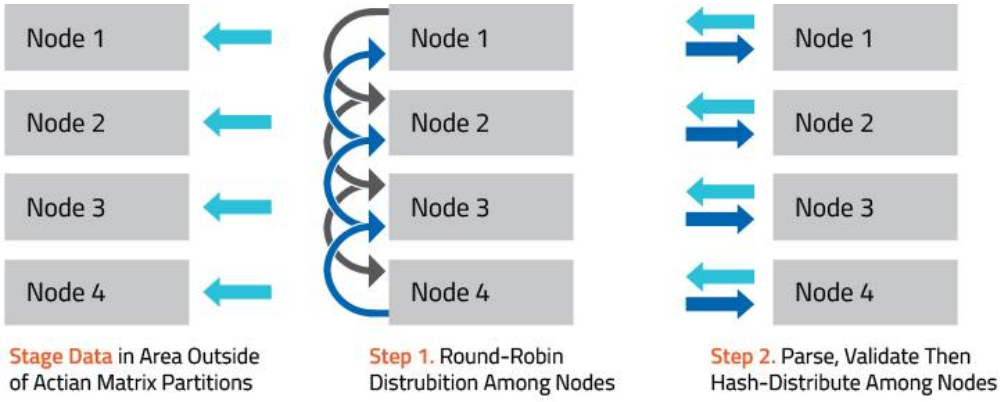


Figure 9: Standard Parallel Load

Massively Parallel Load

Massively parallel loading bypasses the leader and reads data staged directly on compute nodes from a disk staging area that is outside of Actian Matrix's disk partitions. (Note: it is not mandatory to use all compute nodes for staging). When loading is initiated, the compute node pulls the data from its staging area and proceeds with step 1 (round robin distribution) then step 2 (if required). Because the leader is not involved, the load is not throttled by its bandwidth and can

instead leverage the entire cluster's bandwidth for greater throughput. In customer environments, massively parallel loading achieves speeds of up to 100GB per hour per node.

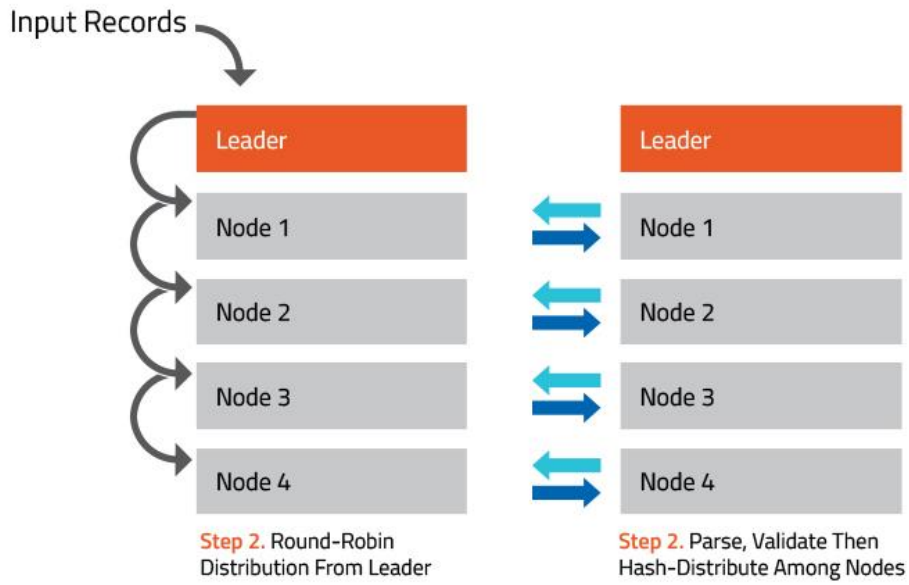


Figure 10: Massively Parallel Loading

For massively parallel loading in SAN-attached environments, data is staged onto the SAN. For step 1, the compute nodes will pull data from the SAN. For step 2, data will be written to both the direct attached and SAN drives as appropriate (See "High Performance in SAN-Based Environments").

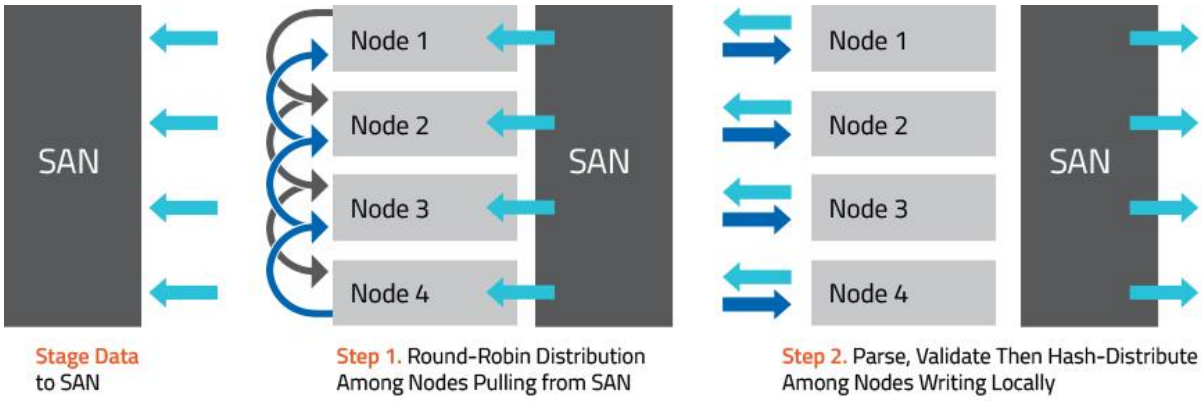


Figure 11: Massively Parallel Loading in Blended SAN Environment

Parallel Extract

In addition to high-speed, parallel loading, Actian Matrix uniquely offers high-speed parallel data extract. Parallel extract provides the best performance for quickly populating downstream data structures with data from the data warehouse. Extracting is essentially the reverse of loading and is very flexible. It can be accomplished with or without the leader, and can write to a flat file or streaming process, or any other network destination.

Mini-Batches and Stream Loads

Actian Matrix can easily employ mini-batch loading and stream loads to achieve near-real-time data freshness (few seconds). Batches can be created by size (every n records) or time (every n seconds). A best practice for mini-batch loading is to ensure that the batches involve enough records to involve at least one entire data block for each column on each of the nodes in order to maintain good parallel efficiency. Actian Matrix maintains a transactional-consistent view of the database with snapshot isolation. (See "Snapshots (Snaps)").

High Availability (HA)

Data warehousing and business intelligence have become mission critical. To eliminate single points of failure, dual or mirrored components (e.g., fabrics, nodes, disks, and so on) are staple offerings. For enterprise readiness, Actian Matrix leverages availability and recoverability features including hot standby nodes, snapshots and backup based on parallel unload/reload. Overall, fault tolerance and high availability can be appropriately configured to meet the customers' service level requirements in order to balance the business need with the size of the hardware investment.

Availability During a Disk Failure

To achieve availability with disk or full-node failures, Actian Matrix mirrors data to other nodes within the cluster. Each disk's storage contains the "primary" disk image on the outer (higher performance) tracks and mirror material on the inner tracks. Therefore, in normal operation, the full throughput of all disks is available to the database.

If a disk fails, the node obtains that disk's primary data across the Actian Matrix interconnect from the sibling nodes whose disks provide the mirror data. Since the additional query processing load is incremental to the sibling disks, performance is not significantly impacted. When the failed disk is replaced, Actian Matrix automatically populates the disk in the background with the appropriate primary and mirror data to restore full performance and HA. The nodes can also be configured with a "spare" disk which is automatically similarly populated when a disk fails. Disk failure and recovery is completely invisible to the users of the system. The in-flight queries seamlessly obtain their data from the mirror disks, without interruption. Only the administrator is notified that a disk has failed and should be replaced.

In addition to Actian Matrix's mirroring, the customer can apply standard hardware or operating system RAID to the disks to take advantage of an additional level of HA. These RAID techniques typically require more disks to obtain the same performance level of Actian Matrix's intrinsic mirroring.

Node Failover with a Hot Standby Node (HSN)

An important criterion in enterprise-class database systems is consistent performance to meet SLAs even after a node fails. Actian Matrix achieves this by including one or more hot standby nodes to take over if a node fails. With hot standby nodes, Actian Matrix leverages inexpensive server hardware to provide a consistent user experience and maintain required SLAs.

Compute Node Failover

If a compute node fails, the HSN takes over and can immediately access its data through each sibling's failover copy. In the background, a copy of each sibling's failover data migrates to the HSN in order to resume normal performance levels.

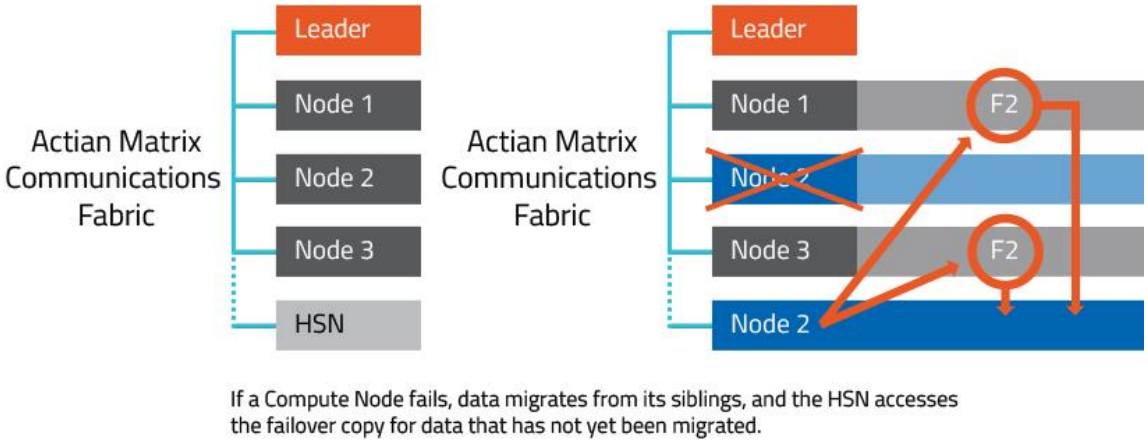


Figure 12: Compute Node Failover with HSN

Leader Node Failover

Leader node high availability is facilitated with third party monitoring software that continuously monitors availability of both the leader and the HSN. If a leader node fails, the HSN takes over the leader function, but data migration is not needed. Alternatively, if the HSN fails, the administrator is notified so repair can be made and the configuration will again have a healthy HSN available.

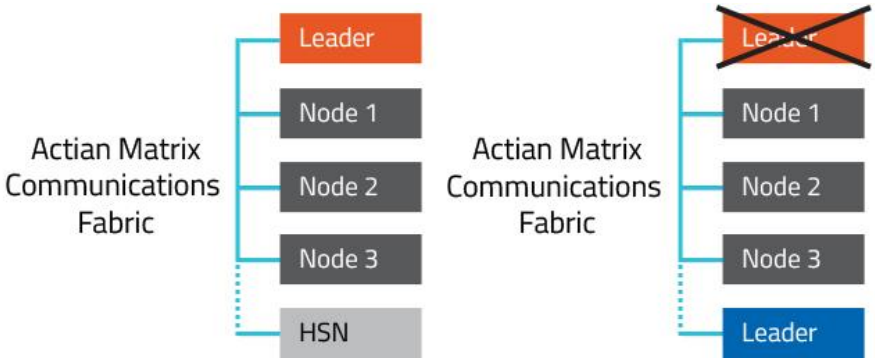


Figure 13: Leader Failover with HSN

SAN-Based Node Failover with HSN

Actian Matrix's enhancements for SAN environments (see "High Performance in SAN-Based Environments") allow it to leverage a SAN-specific approach for high availability (failover/failback) if server hardware fails. In this scenario, all storage is mirrored on the SAN so the hot spare can immediately take over processing by accessing the SAN-based copy of the data. Similar to the non-SAN environment, each node's internal disks are mirrored to multiple logical units (LUNs) in the SAN. However, the SAN's LUNs are always configured with RAID so disk spares or RAID are not needed for the server's internal drives.

For the purposes of snapshot backups and disaster recovery, the mirror images on the SAN are considered the "data of record". Thus, the server's internal disks do not directly participate in the HA model, but instead function as a "cache" for part of the SAN. Actian Matrix stores temporary tables and other transient data on SAN partitions that do not participate in the snapshot or Disaster/Recovery (D/R) system. This significantly reduces the overhead of snapshot maintenance and reduces D/R connectivity requirements. (See "Disaster Recovery (D/R)").

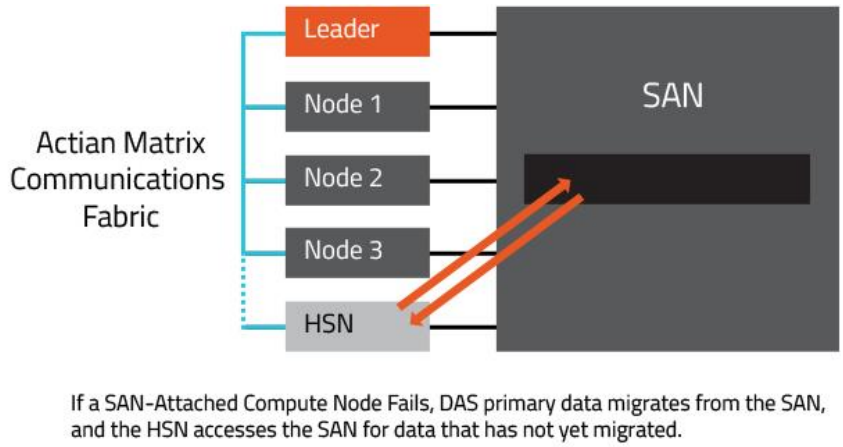


Figure 14: Compute Node Failover with SAN

Disaster Recovery (D/R)

For higher assurance of continued processing, disaster recovery systems can be implemented locally or remotely by leveraging a SAN's enterprise software features. In this scenario, the enterprise software handles data replication and synchronization.

Snapshots (Snaps)

Snaps instantly create and maintain a record of the database at a point in time to provide for recoverability from processing errors. Snaps provide a low-overhead way for customers to restore the database system to a prior state in a matter of seconds, rather than experience a time-consuming restore from backups.

Actian Matrix leverages the SAN's enterprise management software to initiate and manage snaps. Snaps can be scheduled to run periodically, or invoked dynamically by the management software or a SQL instruction to Actian Matrix (which in turn notifies the management software).

Actian Matrix's ability to initiate snaps is a unique advantage. Managing the snap transaction from within the database ensures the snap data is at the same transaction commit level even across multiple SAN and D/R instances *without acquiescing the database*. This allows processing to continue while also eliminating the possibility of D/R systems being out of synch.

Solution Simplicity

Changing business demands, requests for new applications and data sources, and extended data volumes continue to exert pressure on existing resources. Meeting business needs within the required time frames requires a manageable yet flexible environment. Also, as analytic implementation costs skyrocket, solution simplicity becomes as important as performance. IT organizations must continually find ways to improve productivity and accomplish more with the same resources. With a high performance product like Actian Matrix, the time spent remedially tuning traditional database systems for analytic processing can instead be directed to delivering new analytic capability.

Adaptive Workload Management

Actian Matrix automatically and intelligently adapts to share resources between groups and applications based on user-defined priorities enabling more efficient sharing of analytic resources to maximize use of the analytics database. Resource priorities are specified by simply giving each group different relative weights; resource groups can be defined by user, department, application, or query type (e.g. read/write, short/long). The Adaptive Workload Management capability ensures the analytics most critical to the enterprise receive prioritized access to deliver timely insights to the organization.

Load-and-Go Design

Actian Matrix is designed to be simple and flexible for reduced cost of ownership. Exceptional performance without tuning structures (materialized views, aggregates, and so on) means that customers don't need to spend a lot of time thinking about physical design before they can begin analyzing data.

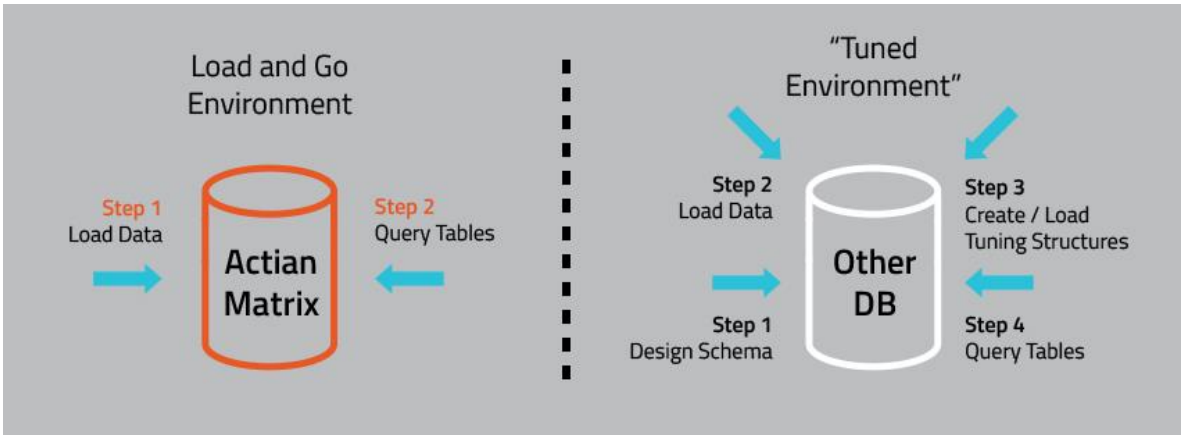


Figure 15: Load and Go vs. Tuned Environment

Schema Neutral

A key contributor to Actian Matrix's load-and-go simplicity is its schema neutral design. Unlike other analytics databases, Actian Matrix is not limited to simple star schemas and simplistic query constructs; it delivers high performance for normalized, de-normalized and dimensional designs, and complex queries. By being schema neutral, Actian Matrix offers ultimate flexibility for quick implementation.

Normalization

The process of normalization is a systematic way to ensure that a database structure is suitable for general-purpose querying and free of certain undesirable characteristics—insert, update, and delete anomalies—that could lead to a loss of data integrity. Normalized designs, such as third normal form (3NF) can be hard for many DBMSs to process, so their implementations will often include redundant structures to avoid joins (these structures also increase load time). As discussed earlier, Actian Matrix is not join-averse, and provides efficient join processing so that the flexibility of a normalized design is not outweighed by detrimental join performance, tuning redundancy, or increased load times.

Denormalization

During denormalization, a designer attempts to improve query performance (and avoid joins) by grouping data or adding redundant data. Improving query performance through denormalization can negatively impact write and load performance and consume extra space for redundancy (e.g., materialized views, pre-built summaries). Actian Matrix, like any DBMS, can benefit from denormalization, but the organization must evaluate the design to determine whether the negative impacts outweigh the potential performance benefits. Because it is built for analytics, Actian Matrix can deliver solid performance without denormalization to save significant effort over time.

Dimensionalization

Dimensional Modeling (DM) is a design technique often used in data warehousing environments. As an analytic system, Actian Matrix is well-suited to dimensional designs though it does not heavily rely on them as do some DBMS products. Dimensional designs create some amount of data redundancy. As with the other designs the negative impacts should be weighed against the positive ones.

Ad Hoc Query

Readiness for ad hoc query is a key requirement for data warehouses to deliver ongoing business value. Actian believes strongly in making ad hoc query more feasible by reducing the need for performance-related design and maintenance that would add latency and limit scalability and extensibility. Actian Matrix includes a number of architectural features that directly benefit ad hoc query performance without query processing "hints" or hardware tuning tied to specific queries or data volumes. These include automatic de-correlation of correlated sub-queries (CSQs) and the detection of opportunities to dynamically select table projections (if they exist) on a query-by-query basis. In addition, the Actian Matrix exhaustive query plan optimization methodology uses configuration-specific performance metrics to tune each query's I/O, network and memory workloads across the cluster to take full advantage of system resources.

Standard Interfaces and Tools

Actian Matrix uses standard interfaces (ANSI SQL, ODBC, JDBC) and supports all of the popular commercial and open-source BI and data integration tools (e.g., IBM Cognos, SAP Business Objects, Informatica, Information Builders, JasperSoft, Microsoft, MicroStrategy, Pentaho, Tableau, Talend, YellowFin, and more).

Actian Matrix is SQL92 compliant and supports the most important analytic extensions in the SQL99 and SQL:2003 standards including analytic functions, window functions and correlated sub-queries. The database also provides Actian-specific features, and recognizes and accepts a variety of common stored procedure languages used in traditional BI environments.

SQL Coverage

Actian Matrix combines straightforward ANSI schema definition, simple administrative functions and a very robust query processing capability including support for specific Oracle, Microsoft, and Postgres SQL functions.

Schema Definition (Data Definition Language - DDL)

Actian customers use standard ANSI CREATE TABLE statements to define their schema. In addition, Actian Matrix's DDL allows, but does not require, users to specify per-column compression and per table sort and distribution keys.

Primary and Foreign key attributes, if specified, help the query optimizer to choose the optimal query plan. Actian Matrix implements a core set of data types, with support for variable- and fixed-length character strings, integers, decimals, dates, timestamps and Boolean values.

SQL Query and Modification (Data Manipulation Language)

Actian Matrix supports the key query constructs described in the SQL standard, including full support for all join types, set operations such as UNION/MINUS/INTERSECT, and a rich expression capability. Also included are the SQL:2003 window functions such as SUM, AVG, and RANK. These are implemented with ANSI-compatible OVER clauses with full partitioning and range specification.

Data is modified using SQL CREATE, TRUNCATE, INSERT, DELETE, UPDATE commands. The COPY command provides a broad range of options for parallel loads from delimited or fixed-length input files. All data modification commands are transactionally ACID with full snapshot isolation based on Actian Matrix's internal record versioning system. Read-only transactions are never blocked by write/update transactions.

For compatibility and ease of application development, Actian Matrix supports a number of SQL functions that are implemented in the Oracle and Microsoft SQL Server databases, such as functions for analyzing date-time data and manipulating character strings. Actian Matrix recognizes key aspects of the Microsoft TSQL language, including stored procedure definitions and control-of-flow commands for conditional and sequential processing.

Correlated Sub-Queries (CSQs)

A CSQ is one that contains an embedded query that depends on the outer query for its values. This means that the sub-query must be executed once for each row that is selected by the outer query. CSQs are common in complex analytic environments, but they are challenging for MPP databases because they don't inherently scale well.

Actian Matrix automatically de-correlates these complex queries internally, translating them into joins and aggregations of intermediate result sets. This dramatically reduces the time-consuming burden other DBMS systems place on application developers to manually de-correlate. It also allows the use of newer BI application packages that emit correlated sub-queries to the DBMS.

Stored Procedures

Stored procedures allow designers to include non-SQL programmatic constructs (e.g., C++) into their database applications. Actian Matrix supports two stored procedure languages: Microsoft T-SQL and PostgreSQL.

User Defined Functions (UDFs)

In SQL databases, UDFs provide a mechanism to extend the vendor's SQL language in unique ways by adding functions that can be evaluated in SQL statements. Once created, a user defined function may be expressed in SQL statements. Actian Matrix's current UDF support allows customers to embed C or C++ scalar functions directly in the database so they may take advantage of the attributes of MPP.

Identity Columns

Actian Matrix supports identity columns. This features enables surrogate key implementation through self-incrementing integer columns.

Security Features

Actian Matrix offers column-level encryption to aid applications that require extra data security (e.g., Financial Services, Healthcare, etc.). The encryption implementation takes advantage of the columnar nature of the DBMS and particularly its compression to offer exceptional performance in encryption scenarios. Actian Matrix also runs in Payment Card Industry (PCI) and Security Technical Implementation Guides (STIG) compliant environments.

Appliance Simplicity on Standard Hardware

Actian Matrix is designed to be an advanced analytic software engine that optimizes standard, highly available hardware components from any major vendor to deliver exceptional performance without the lock-in of proprietary hardware. This allows tremendous cost of ownership advantages as customers benefit from economies of scale of highly available, ubiquitous components that align with their standard operating environments.

Typically, nodes are standard 2-CPU, quad-core servers running Linux or Solaris with 32GB or more of memory and a varying number of internal drives, depending on the manufacturer and model (less storage is required for systems intended to run all-in-memory). Compute nodes must be of identical specification for optimal parallel efficiency (the leader may be configured

differently depending on workload characteristics). As stated earlier, the fabric software is designed to run efficiently on cost effective GbE.

Actian Matrix is easily deployable in all enterprises, including virtualized standard operating environments. Actian Matrix is available as enterprise software and be deployed on-premise, in the cloud, or in a hybrid configuration.

Platform Check

Actian Matrix offers a simple, seamless platform installation that includes the OS and the DBMS. To facilitate appliance simplicity, a platform check feature will ensure that a system is properly cabled and configured and that all components are functioning.

Summary

The Actian Matrix Analytics Database has combined best practices with new engineering to become the fastest, simplest, most cost-effective RDBMS for analytic processing on the market today. Actian Matrix has proven itself in customer scenarios and audited benchmarks to have leading-edge performance, best price-performance and scalability, and is aggressively changing expectations about analytic processing. In Actian Matrix, we are offering a new generation, differentiated analytic DBMS combined with high availability and solution simplicity to give leading enterprises a modern solution to handle all their analytic workloads.

About Actian: Accelerating Big Data 2.0

Actian transforms big data into business value for any organization – not just the privileged few. Actian provides transformational business value by delivering actionable insights into new sources of revenue, business opportunities, and ways of mitigating risk with high-performance in-database analytics complemented with extensive connectivity and data preparation. The 21st century software architecture of the Actian Analytics Platform delivers extreme performance on off-the-shelf hardware, overcoming key technical and economic barriers to broad adoption of big data. Actian also makes Hadoop enterprise-grade by providing high-performance ELT, visual design and SQL analytics on Hadoop without the need for MapReduce skills. Among tens of thousands of organizations using Actian are innovators using analytics for competitive advantage in industries like financial services, telecommunications, digital media, healthcare and retail. The company is headquartered in Silicon Valley and has offices worldwide. Stay connected with Actian Corporation at www.actian.com.