Memory-Centric Database Acceleration

Achieving an Order of Magnitude Increase in Database Performance

A FedCentric Technologies White Paper

September 2007

Executive Summary

Businesses are facing daunting information challenges. Business processes such as revenue protection, fraud detection, RFID, social network analysis, automated query enrichment, biometrics, complex event processing, and real-time customer interaction analysis have the potential to yield huge cost savings and important information about a business' health and status if performed in a timely manner. However, this real-time class of application often exceeds the capability and performance of disk-based database systems.

The mandate to automate these critical business processes and leverage increasing quantities of information in ever shorter timeframes have left many enterprises drowning in the data these processes create, unable to ingest, process, and analyze the ever-increasing data stream. Many have attempted to address these challenges by "throwing hardware at the problem", deploying data warehouses of large server farms with only to find that certain high value business critical processes exceed the capacity those traditional approaches.

Time and risk collide, resulting in lost revenues, opportunity costs, dissatisfied customers, and unnecessary operations and maintenance costs. And in some data centers, power and space consumption are nearing, or at, capacity, making additional hardware resource deployment practically impossible.

Fortunately, a solution for providing transformational database performance is available today. Memory-Centric Database (MCDB) Acceleration uses commodity hardware components and random-access memory (RAM) to deliver breakthrough performance, typically an order of magnitude or greater, using less power, space and cooling than traditional disk-centric approaches. MCDB uses disk for capacity not performance, thereby enabling tremendous results.

Customers adopting MCDB have the potential to realize dramatic increases in database performance. FedCentric Technologies has experienced 10 to 10,000 times speedup in real application performance. MCDB allows businesses to process vast quantities of data in near real-time.

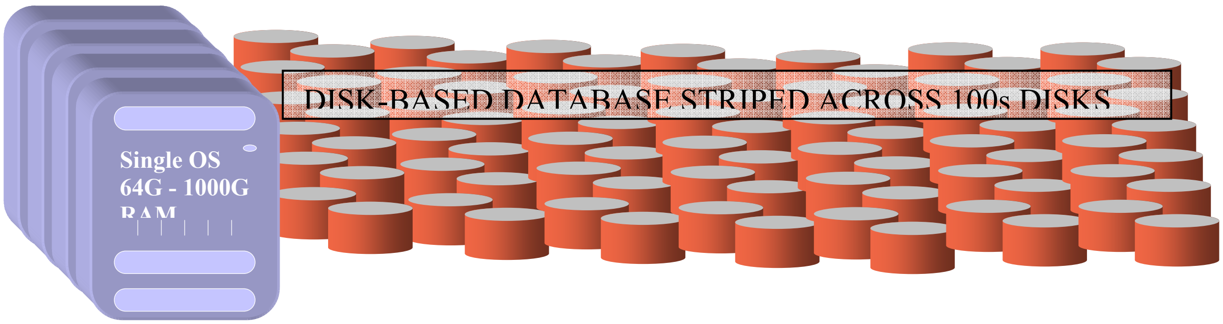
What Would You Do With an Order of Magnitude Increase in Database Performance?

MCDB can provide from one to many orders of magnitude increase in performance and is complementary and compatible with your existing Oracle disk-based RDBMS system. MCDB targets applications that have proven difficult or impossible to implement using traditional approaches.

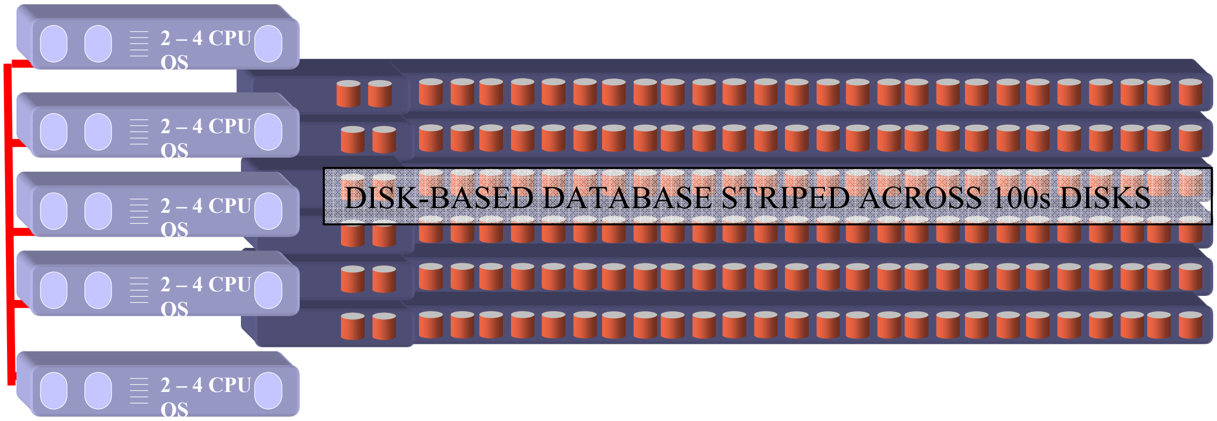
Traditional Approaches

Historically, information systems have been built using a set of disk-centric assumptions, namely that for most operations, some data resides in random access memory (RAM), but most data remains on disk and must be retrieved into memory and managed once there. All disk-based databases (Oracle, SQL Server, DB2, MySQL) were designed with these assumptions in mind. Disk-based databases maintain a logical-to-physical RAM-to-disk address map (called a buffer cache), and must therefore spend significant CPU cycles to manage and maintain the integrity of the buffer cache. Even if all the data is cached in memory, the disk-based RDBMS must perform logical-to-physical lookups to find the cached data, spending CPU cycles accessing data, indexes, and sort spaces.

Application speed up requires better performance from the disk subsystem. Disk optimization involves striping small amounts of data onto each disk and using striping algorithms to spread the data over as many disks as possible. This technique permits parallel seeks over a broad array of disks, thereby minimizing seek and latency times. However, using disks to optimize performance dramatically increases the amount of disks required for the application.



In an attempt to better balance and manage computing resources, others have embraced clusters of small computers and network attached storage. Unfortunately, these approaches include the same disk-based performance constraints (and costs), while introducing network I/O as the new bottleneck.



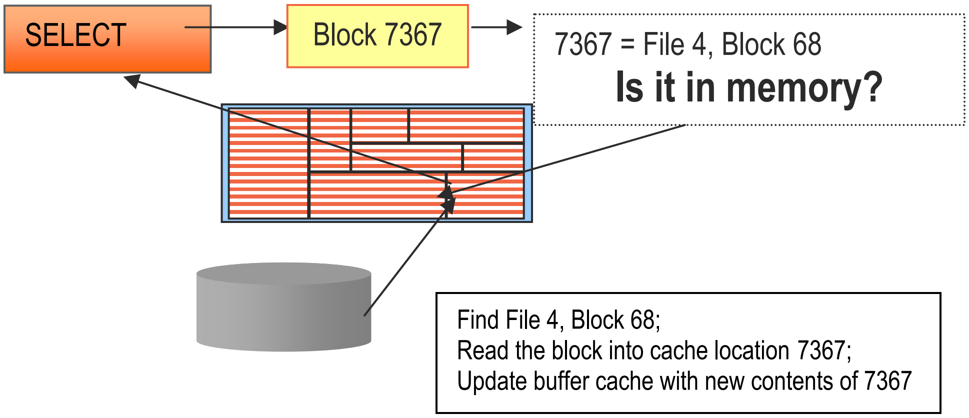
So how does one build a system that provides an order of magnitude better performance without the associated cost and space requirements associated with disk-based approaches?

For the purposes of our discussion, we will define "performance" as the ability to accomplish some quantity of work per unit of time. The performance of disk-based systems is restricted by the laws of rotational physics associated with disk drives. Once a request for a disk block occurs, the process making the request must wait for the disk to spin to the proper cylinder and sector, and for the head to move to the proper location to read or write the data to or from disk. This disk latency is typically in the area of 2 – 10 milliseconds (thousandths of a second).

Over the years disk drives have gotten denser and less expensive, but not faster. Modern disk drives can perform 100 - 120 or so disk I/O operations per second (IOPS). Building modern disk-based information systems involves an analysis of the expected throughput required, given these constraints. If a system's throughput requirement is 10,000 IOPS, it will require file systems striped across at least 100 disks, regardless of the capacity of the disk drive. If the throughput requirement is 100,000 IOPS, a file system striped across at least 10,000 disks will be required. Large disk farms consume floor space and tremendous amounts of power.

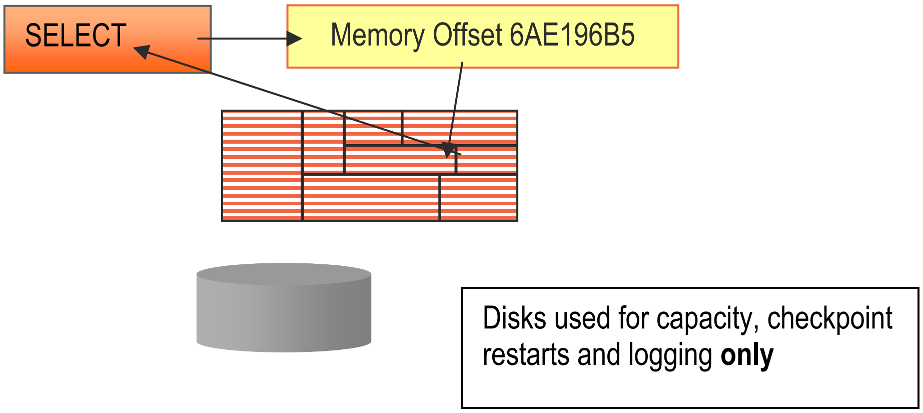
The use of disks to increase performance has greatly contributed to the power and floor space problems at data centers. And it leads one to ask, "Are wide stripes across myriad disks the appropriate solution for high throughput/low latency applications?"

Some system architects attempt to address disk latency and throughput by moving some or all of the file systems to solid-state (or RAM) disks. Putting data in RAM reduces or eliminates the latency associated with "spinning" disks, and can improve system performance by 2 to 4 times. However, the lengthy code path associated with disk-based RDBMS still applies. Even though all the data is in memory, the disk-based RDBMS must maintain and translate logical to physical block addresses and perform memory transfers. The length of the buffer cache management code path logic continues to limit the performance of RAM-disk technology. There is a better way to make use of RAM: MCDB Acceleration.


SELECT
Block 7367
7367 = File 4, Block 68
Is it in memory?
Find File 4, Block 68;
Read the block into cache location 7367;
Update buffer cache with new contents of 7367

The MCDB Acceleration Approach

MCDB starts with a different set of assumptions: all data structures (tables and indexes) are in memory. This means that MCDB systems do not impose multi-millisecond waits for I/O. Once the data is placed into RAM, the location of that data is a constant memory address offset. Indexes are simpler, smaller, faster to build and maintain because they are constructed from the physical addresses, requiring no logical to physical translations. MCDB systems leverage the potential value of large quantities of RAM without layering performance-limiting disk-based constructs (file systems, files, buffer cache management, etc.) on top of the database management system. MCDB systems tend to be an order of magnitude faster than fully cached disk-based systems or RAM-disk solutions, and can be two to four orders of magnitude faster than disk-based systems that perform I/O. MCDB systems use disks for persistence and logging of transaction data, not for performance.


SELECT
Memory Offset 6AE196B5
Disks used for capacity, checkpoint restarts and logging **only**

Though MCDB Acceleration involves putting all data into volatile shared memory, we provide persistence of all MCDB data by logging transactions and writing checkpoint information to multiple files on disk. The MCDB will read all data from the last checkpoint file and roll forward/roll back any uncommitted transactions from the transaction log files.

MCDB can also be configured to automatically cache and synchronize some or all of the data stored in an Oracle database, and can be configured in a redundant configuration for high availability.

By starting with these different assumptions and combining these capabilities we can provide a robust solution to very high throughput and very low latency applications, the class of application that simply exceeds the capabilities of a disk-based approach. This is the class of application targeted by FedCentric Technologies' MCDB.

Quantifying Disk-Centric vs. Memory-Centric Efficiencies

MCDB can significantly improve the performance of certain applications. However, a common assumption is that MCDB will be too expensive to add to the existing system. In Table 1, we compare two Transaction Processing Council benchmark (TPC-C) results for disk-based systems with a similar transaction processing benchmark for an MCDB. The disk-based systems achieve high performance by striping data across large numbers of spinning disks. Spinning disks require significant floor space, consume lots of power and conversely generate heat, which must be cooled. For 16 - 25% of the hardware costs and less than 10% of the operations cost of disk-based solutions the MCDB produces 12 times more work per unit of time. The resulting MCDB price/performance (cost to achieve a certain level of performance) was between 2% and 6% of the disk-based systems' price/performance.

	Disk-based Transaction Benchmark	Disk-based Transaction Benchmark	In-Memory Transaction Benchmark
Processor type, count	SPARC64, 64	Itanium2, 16	Itanium2, 16
Database engine	Oracle 10g	Oracle 10g	TimesTen IMDB
RAM	512 GB	1024 GB	1024 GB
Total storage	64,800 GB	77,184 GB	4,096 GB
Server cost	$3,982,226	$1,083,648	$946,659
Storage cost	$2,130,176	$3,357,040	$135,226
Avg response time	0.131 – 0.598 sec	0.103 – 0.453 sec	0.0048 - 0.0173 sec
Performance	595,702 tpmC	1,238,579 tpmC	7,283,908 trans/min
$ Price/Performance	$12.42	$3.94	$0.25
KWH Consumption/yr	1,436,747	1,559,165	139,858
Annual power cost	$181,368	$196,822	$17,655
Server ft^3	88.6	51.7	51.4
Storage ft^3	506.3	614.1	1.4
Perf/KW	3,634.5	6,963.6	456,541
Perf/ft^3	1001.4	1860.2	138,099.2

Table 1: Comparison of Disk-Based vs. Memory-Centric Transaction Processing

Source: Transaction Processing Council Executive Overviews

http://www.tpc.org/results/individual_results/Fujitsu/fujitsu_pw2500_20040106_es.pdf
http://www.tpc.org/results/individual_results/Fujitsu/fujitsu_primequest540_16p32c_tpcc_es.pdf

Customer Prototypes and Use Cases

The results produced in the preceding analysis have been replicated using real applications. Over the past several years, FedCentric has worked with a number of customers to quantify the business value of solving demanding data management problems with MCDB Acceleration.

MCDB Case #1: Disk-based, RAM-Disk & MCDB String-Search

FedCentric performed initial tests on random string searches on a 1 billion row table using a traditional disk-based RDBMS system, a disk-based RDBMS on a RAM disk, and MCDB. The results, shown in Table 2, demonstrate that the MCDB was 13 times faster than a fully cached Oracle database, and 5 times faster than Oracle implemented on RAM disk.

	Disk-based Database	Disk-based RDBMS in RAM Disk	Memory-Centric Database Acceleration
Processor type, count	Itanium 2, 16	Itanium2, 16	Itanium2, 16
Database engine	Oracle 10g	Oracle 10g	TimesTen IMDB
RAM	256 GB	256 GB	256 GB
# rows	1 billion	1 billion	1 billion
Queries processed/min	10,257	27,488	137,836

Table 2: Comparison of Disk-Based, RAM Disk, and MCDB Systems Performance

MCDB Case #2: Global Florist Model

In this use case, MCDB provides for high-speed analysis of data as it is being ingested. The data model resembles a world-wide florist, with PERSONs placing ORDERs for flowers at a combined rate of 250 million records per hour. The customer wanted to perform three types of queries against the data. A SIMPLE QUERY would return all the info about a PERSON selected at random; a more complex PERSON/ORDER join would return all the sender and receiver PERSON information and the ORDERS they shared; the most complex query, a SUBQUERY analyzed the sending/receiving order pattern where receivers would then turn around and become senders.

Table 3 shows the performance results for both ingest and query on a MCDB of over 15 billion rows (1.5 terabytes in size). FedCentric's MCDB Acceleration approach exceeded the expected ingest performance by 5 to 12 times, and the expected query rates by 91,000 to 775,000 times.

Performance Target	**Observed (single threaded)**
Ingest 200 million ORDERs/ hour	Ingested 1.006 billion ORDERs/hour
Ingest 50 million PERSONs/ hour	Ingested 611 million PERSONs/hour
SIMPLE QUERY < 1 sec.	SIMPLE QUERY at rate > 91,000/sec
JOIN PERSON/ORDER < 1 min.	JOIN PERSON/ORDER at rate > 13,000/sec.
SUBQUERY in < 5 minutes	SUBQUERY at rate > 2500/sec

Table 3: Ingest and Query Performance

MCDB Case #3: Real-time Complex Event Analysis

This application identifies real time fraudulent usage patterns in a supply chain. The customer originally constructed a data warehouse, with bulk ingest and batch querying of supply chain transactional data several times throughout the day. Unfortunately, this approach created enough delay that items exited the supply chain before potentially fraudulent usage patterns were identified. And the amount of hardware thrown at this problem quickly created power, space, and cooling issues within the data center. FedCentric transitioned a portion of the system from a batch data warehouse to a real-time fraud detection system, processing a series of complex event analyses on data in real time as it moved through the supply chain. Table 4 shows the performance results of the prior system and the MCDB performance. MCDB performed real-time fraud detection of data on ingest 19 times faster than the prior system's batch-oriented item processing.

Batch-oriented System Performance	**Real-Time MCDB Performance**
Batch ingest of 2200 items/sec	Real-time ingest of > 42,000 items/sec
Batch queries on hours of data	Real-time analysis & reporting on each item
Bulk load data rate of 7200 rows/sec	Bulk load data rate of 138,000 rows/sec
	1024 GB RAM MCDB

Table 4: Real-Time Complex Event Analysis Performance

MCDB Case #4: Real-Time Data Enrichment

This application involved real-time data enrichment as customer interactions took place and contact information moved throughout the enterprise. The rate at which contacts occurred was somewhere around 25 million per day (less than 300/sec). FedCentric constructed an MCDB and replicated the customer's data enrichment environment, and built an Enterprise Targeted Marketing System that simulated the required data enrichment. The results from this prototype are shown below in Table 5. FedCentric's MCDB Acceleration approach produced results that were 672 times the expected query performance.

Performance Target	Observed
• Perform 290 queries/second	• Performed 195,000 queries/sec
• Perform 25 million queries/day	• Performed 16.8 billion queries/day
• Scale up from 1 to 4 simultaneous threads	• Demonstrated scale-up ○ 1 thread: 65,000 queries/sec ○ 2 threads: 132,000 queries/sec ○ 4 threads: 195,000 queries/sec

Table 5: Data Enrichment Query Performance

The preceding customers each needed at least a 10x improvement in their current performance, and found it with MCDB Acceleration. In each case, the business mandate to do more work in ever shortening time windows compelled these customers to look beyond the traditional disk-centric approach to building high performance systems. Though the business problems were different, they had one thing in common: the application requirements exceeded the performance thresholds of disk-based RDBMS.

Summary & Conclusion

What would you do with an order of magnitude increase in database performance? Accomplish more work in less time? Spend less on hardware? Use less power, space and cooling to accomplish the same amount of work? Meet critical business process timelines and service level agreements? Find more value in the data that flows into and floods your organization?

If your organization has implemented a disk-based data management approach and the performance of your system is satisfactory, congratulations, and thank you for taking the time to read our white paper. We wish you continued success.

If your organization is experiencing significant performance challenges with your disk-based system, MCDB Acceleration may provide a valuable enhancement to your existing systems. Please contact FedCentric to explore the advantages that MCDB may provide to your business. Together we can discover what you could do with an order of magnitude increase in database performance.

Joe Conway
Chief Technology Officer
FedCentric Technologies, LLC
joseph.conway@fedcentric.com
703 628 7264

Gerry Kolosvary
Chief Executive Officer
FedCentric Technologies, LLC
gerry.kolosvary@fedcentric.com
301 263 0083

About FedCentric Technologies, LLC

FedCentric is a recognized thought leader and practitioner in the area of memory-centric database acceleration. We combine the unique capabilities of SGI's open standards-based Altix hardware with the simplicity and raw speed of Oracle TimesTen In-Memory Database (IMDB) to help customers realize the benefits of Memory-Centric Database Acceleration.   FedCentric is an authorized reseller of SGI Altix computers and Oracle TimesTen IMDB software. MCDB Professional services are provided by Integrated Computer Concepts Incorporated, a FedCentric partner.

Please visit us at our website http://www.fedcentric.com

All product or company names mentioned are used for identification purposes only and may be trademarks of their respective owners.