Grid Computing: Business & Technical Value Validated at Bank of America

IBM Grid Computing Helped Bank of America Attain a 10+-Fold Increase in Processing Speed and Capacity to Improve Customer Service

A Look at the Problem

Bank of America and its broker-dealer, Banc of America Securities LLC, provide strategic advice and corporate & investment banking solutions and services to corporations, financial institutions, government entities, and institutional investors worldwide.

The client roster includes 97 percent of the U.S. 2005 Fortune 500 and 79 percent of the Global 2005 Fortune 500. To serve these clients, Bank of America's analysts depend on IT to process their calculations for financial analysis in risk, credit, and other areas.

So massive are these calculations, they typically take many hours to process -- creating a "number crunching" bottleneck in serving the bank's analysts and their customers.

Like many financial services firms, Bank of America's IT architecture consisted of dedicated silos of static resources.

Bank of America wanted to increase computing power so that calculations could be done faster – *without* spending millions of dollars adding more processors and storage devices to its data centers.

In early 2005, Bank of America's Global Markets Trading Technology Architecture team conducted a series of road show discussions with development teams in Charlotte, Chicago, London, and New York. These meetings highlighted several ubiquitous data distribution challenges common to the Bank's businesses.

One of the major issues identified as a universal problem across front, middle, and back offices was *data latency* – the time it takes for an application or user to access cached data. Long latency times can adversely affect the Bank's ability to deliver the fast quoting and execution today's demanding customers require.

Speed of quoting and execution are critical for remaining competitive in the investment banking marketplace. When a broker wants to place a trade, he may contact several sources. Whichever vendor can quote a price fastest -- and then execute the trade most swiftly -- typically gets the order.

In addition, by enabling computations to be executed faster and more frequently, grid computing can eliminate processing bottlenecks in portfolio analysis, wealth management analysis, risk reports for capital markets, actuarial analysis, and other financial applications.

Another key IT objective is to achieve greater *data liquidity*, defined as the ability for all users to access all of the bank's data, anywhere and at any time, regardless of where the data resides or how it is structured. Achieving data liquidity is critical to achieving the bank's goal of becoming a dominant player in electronic trading -- and to establishing real-time risk and P&L capabilities across businesses.

The Solution: Grid Computing

Bank of America opted to increase computing power not by adding processors, but through more efficient usage of existing IT resources – attained with a type of distributed computing architecture known as *grid computing.*

In the mid-1990's, IT professionals working on advanced science and technology projects began using the term "grid" to refer to their large-scale, highly powerful distributed computing deployments.

This grid architecture enabled high-performance computing to evolve from the use of large, proprietary, expensive machines that used vector computing to "commodity Intel" computers available at very low prices, providing a high compute capability to dollar ratio.

By pooling computing resources, grid computing also allowed CPU-intensive applications to process data much more quickly. For example, the Search for Extraterrestrial Intelligence (SETI) project uses grid computing to analyze signals from the world's largest radio telescope with idle processing time donated by more than 2 million volunteers from all over the world. And the Human Genome Project used grid technology to perform sequencing tasks in cracking the human genetic code.

Yet even as those super-scale projects were being brought under control, business IT infrastructures also began exploding. Complexity has seemingly grown exponentially. Speed, power, and agility are needed as never before.

Resource sharing and problem solving must take place in dynamic, multifaceted "virtual organizations" within and across enterprises. Grid computing offers exciting and effective ways to address these issues. That's why grid computing is now a critical component of day-to-day business.

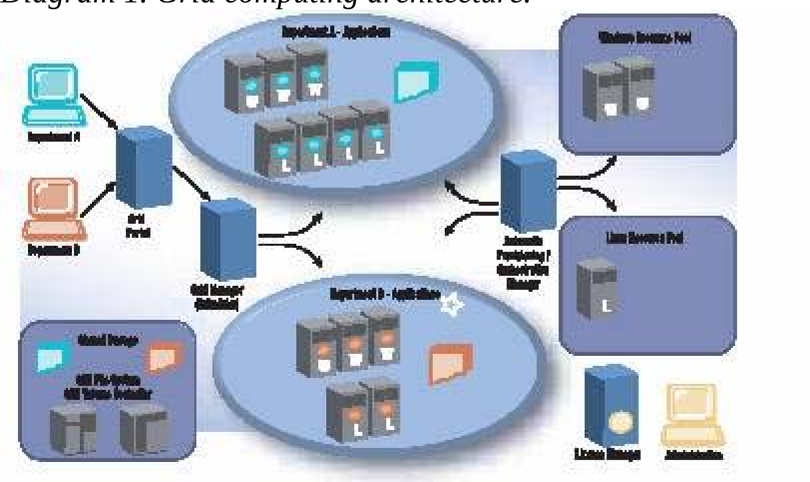
How Grid Computing Works

The current on-demand business climate, bringing a multitude of benefits to companies and customers alike, requires continuous innovation in order for a company to differentiate its products and services in the marketplace.

Grid computing can help businesses to adjust dynamically and efficiently to market shifts and customer demands. It enables collaborators to rely on today's advanced heterogeneous tools, speed time to market, and tap new levels of processing power and storage space. The grid computing evolution is a critical component of e-business on demand.

In grid computing, processors, storage devices, applications, data, and other network resources are coordinated and dynamically shared throughout an enterprise (see Diagram 1). Resources are pooled and made available to any location -- whether in a different building or campus, or across time zones and international borders. The grid connects disparate computers into one large, integrated computing system, resulting in a higher level of utilization and service.

Diagram 1. Grid computing architecture.



The idea is simple: very few of your network's resources are working at full capacity around the clock. Grid computing enables you to dynamically allocate existing resources to handle tasks whenever and wherever needed.

By using IT resources more efficiently, grid computing can significantly lower computing costs on a price to performance basis. And, by providing processing power as needed, grid computing increases processing speed, giving users better results.

Grid computing creates a framework in which a massive number of servers are dynamically balanced and managed as a pool of resources – creating a virtualization of storage, networking, and computing services and devices. MIPS are pooled, dynamically managed, and delivered to needed applications and services regardless of system load.

Instead of dedicated silos of static resources, grid computing pools all resources to create a services-oriented environment throughout the entire enterprise. It also enables a utility model for charge-back of IT services to users.

IBM views grid computing as critical to the ongoing development of on-demand operating environments. Strategic IBM products that often form part of grid solutions include:

- IBM BladeCenter eservers.
- IBM xSeries Servers.
- IBM DB2 Information Integrator (DB2 II).
- Global Parallel Filesystem (GPFS).
- Network File System (NFS).
- SAN/FS.
- Virtualization Engine.
- Components within WebSphere for scheduling.
- Tivoli Provisioning Manager (TPM) from the Tivoli suite of products.

Computational grids share computing resources to distribute data and speed processing. *Information grids* enable computing resources to process data stored in multiple formats in multiple places and present it in a common form for analysis.

IBM offers solutions for both computational and information grids. Financial services institutions using IBM grid solutions include Bank of America, Higo Bank, and Unicredit.

The IBM Grid Computing Solution for Bank of America

The grid computing solution IBM built for Bank of America uses IBM and Intel hardware running GigaSpaces software.

IBM BladeCenter

The IBM BladeCenter™ server features seven blades and three architectures, which can be mixed and matched on a single chassis. After evaluating Dell, HP, and IBM servers, Bank of America purchased more than 3,700 IBM blades, most running Linux and some running Windows. Table I shows total IBM blade sales in 2004 and 2005. To date, more than 400,000 units have been shipped.

Table I. IBM blade sales 2004-2005.

Quarter	Q104	Q204	Q304	Q404	Q105	Q205	Q305	Q405
Units sold	19,029	25,288	30,206	51,057	36,288	44,343	52,612	68,841
Revenues (USMS)	$82	$113	$140	$227	$172	$195	$258	$306

Intel Processors

Bank of America's grid system is based on Intel's low-voltage, dual-core HS20 Xeon® processor. This multi-core, 64-bit chip features a proprietary Intel I/O Acceleration Technology (iO/AT) that reduces latency between nodes in the grid.

Ideal for deployments requiring high compute density and power optimization, the processor excels at handling demanding multi-threaded, multi-tasking applications such as high-performance financial services. Total dissipated power is just 31 watts.

Intel is a strategic collaborator in the engineering of the BladeCenter product line, working with IBM in chassis and chipset design. The design balances processing vs. memory and other I/O capabilities to provide top overall capability per power used.

The decision to run grid computing applications on BladeCenter is focused not only on today's technology but also on planned improvements to the hardware. These include:

* *64 Bit LV Xeon Processors* – these dual-core, 64-bit processors have maximum efficiency in the 40-watt range and consume only half the power of conventional x86-64 processors.

* *Intel Direct Connect Technology* – in 2006, the BladeCenter will feature a dedicated connection from the chipset to each processor socket, tripling the bandwidth to the processor socket over previous generations.

* *I/O Acceleration Technology* -- the chipset evolution is towards reducing latencies of Ethernet connections. While Ethernet is ubiquitous in grids today, the ability to use acceleration technology will bring to Ethernet low latencies that were previously achieved only by using expensive proprietary connections. Reduced latencies are accomplished through the use of intelligent switching and offloading in the supporting chipset on each blade.

GigaSpaces Software

Bank of America's financial applications run on GigaSpaces infrastructure software. GigaSpaces Enterprise Edition represents a new generation of "network-resident" application server technologies, built from the ground up to address the performance, scalability, and reliability requirements of the most demanding distributed systems and service-oriented architectures.

GigaSpaces is the *only* commercially available third-party software for grid computing that can provide end-to-end, on-demand scalability in a single platform *at the application level* -- and not, as with other enterprise grid platforms, at just the data center level.

The advantage is that GigaSpaces solves *all* bottleneck areas -- data access, data processing, and messaging -- while other solutions address only one or two of these bottlenecks, mainly data access.

How does it work? GigaSpaces enables three-dimensional virtualization through a dynamically scalable *object bus* called the IMDG (In-Memory Data Grid) layer. An "object bus" is simply a big container of objects scaled across any number of nodes in a grid or network.

“Three-dimensional” means that virtualization takes place across the three application axes: vertical (scaling up), horizontal, (scaling out), and time (performance, throughput, and speed).

“Scaling up” refers to the ability to run multiple applications within the same server – achieved by dividing the application processes into multiple threads, all running within the same Java Virtual Machine. “Scaling out” means running applications across a cluster, network, or grid of multiple physical machines.

How does the IMDG object bus provide data access to distributed applications? GigaSpaces partitions the data on multiple machines. Dynamic allocation and re-allocation of the data on those machines permits more efficient use of memory, with scaling of capacity on a pool of machines available on an on-demand basis.

The object bus enables a meta-tier (“beyond tiers”) topology to be managed across the application’s three axes -- performance, scalability, and reliability – at the same time.

Conventional tier-based applications have separate tiers -- for instance, a database tier, a business logic tier, a presentation tier, and maybe a messaging tier and a caching tier around the data and business-logic tiers.

By comparison, the GigaSpaces multi-tier topology virtualizes and dynamically scales data, business-logic, messaging, and caching simultaneously. The software’s core clustering technology provides a common middleware stack for all tiers.

GigaSpaces’ innovative “space-based architecture” combines advanced services such as distributed data caching, distributed messaging bus, parallel processing, and grid-enablement with open standards and cross-language support to enable a “write once, scale anywhere” approach to the development and deployment of distributed applications.

The flexible architecture permits customers to scale the system almost transparently when load and number of users increase – *without* degrading performance.

In addition, the GigaSpace infrastructure software has a rich set of APIs, enabling Bank of America to plug in its “best in class” messaging and transport system with no product modifications required.

“GigaSpaces is pure Java, and it can run on any platform that supports a Java Virtual Machine (JVM),” says Geva Perry, Executive Vice President of Business Development, GigaSpaces. “However, GigaSpaces has been extensively tested on Intel and IBM platforms and consistently offered excellent results on these platforms.”

He adds that “IBM’s BladeCenter is a perfect platform for GigaSpaces Grid, with built-in high availability for maximum up-time, fault-tolerance at the server level, and innovative deployment and diagnosis tools.”

Among the large financial institutions running GigaSpaces are Bank of America, the Chicago Mercantile Exchange, the New York Stock Exchange, and FXAII.

Starting in London, Growing in Chicago

Recognizing the need for the development of a global grid strategy, a team led by Andy Bishop, Managing Director, in London designed the foundation for the bank’s grid infrastructure.

Building on this foundation, Michael Oltman, VP of Advantage Risk Technology, launched Bank of America’s U.S. grid computing initiative with IBM as a limited-scale pilot project – a risk-management system in the Chicago office -- before rolling out to other locations.

The Chicago-based risk management system processes hundreds of millions of calculations involving terabytes of data and runs 23 hours a day.

“Not only did we want to take what we do and make it better; we also wanted to do things we weren’t able to do with our existing siloed IT infrastructure,” says Oltman.

Both goals were achieved through the IBM grid computing solution. “Resources that we were paying for but using only 8 to 12 hours a day could be used around the clock,” says Oltman.

The most significant, tangible benefit that IBM’s grid computing solution has delivered for Bank of America is increased processing speed. Grid computing not only makes more efficient use of existing processing capacity, but it also enables Bank of America analysts to be more effective at their jobs by completing calculations faster.

“Since the implementation of the IBM grid computing architecture, we’ve gone from a few million to more than a billion calculations processed in a 24-hour period,” says Oltman.

“Jobs that used to take 90 minutes can now be completed in 20 minutes. Computations for a single trade have been reduced from 4 hours to 40 minutes.”

Shifts Processing Into High Gear

One benefit of faster processing is the ability of a user to get results earlier. Another is that users now can perform many more “what-if” scenarios.

“Scenarios that would have taken 24 hours to run on our old infrastructure now run in 5 hours,” says Oltman. “Thanks to grid computing, our Chicago location can run up to 600 scenarios daily.”

More efficient use of computing resources not only helps Bank of America process calculations faster, but it also defers investments in additional IT infrastructure.

“Thanks to the increased utilization, grid computing will save Bank of America tens of millions of dollars in IT costs over the next 3 years with better application convergence,” says Oltman.

The new system's architecture is three-tiered: data services, workflow applications, and grid technology.

"The grid enables us to focus IT resources to serve the business," says Oltman, "and process data from workflow applications at high volume."

"It used to be that an analyst running a scenario would have his data the next day. Now, a user who asks for data at 9am can have it 20 minutes later."

In addition, costly contingency resources normally sit idle unless needed for an emergency. With implementation of the IBM grid computing solution, Bank of America can put its investment in contingency resources to work increasing bandwidth in its production environments at any time.

Says Oltman: "I am a big fan of grid technology. I think this is a wonderful solution to make your organization more efficient – increasing agility and flexibility while ensuring full utilization of IT resources throughout the enterprise."

After successful implementation of the pilot project in Chicago, Bank of America expanded its grid computing infrastructure to encompass investment banking operations in London, Tokyo, and Mexico.

"Grid computing is easy to scale," says Oltman. "You can distribute tasks to available computing resources without changing applications or writing new software unless a new feature is needed."

Proof That Grid Computing Scales to Support the Business

In December 2005, Bank of America, IBM, Intel, and GigaSpaces conducted a large-scale proof of concept (PoC) test with GigaSpaces software running on Intel/IBM hardware.

"The methodology used in the PoC test was to create data grids with distributed caches capable of loading extremely large amounts of data in memory," says B.J. Fesq, Senior VP of Architecture, Bank of America Securities, "and to then have a lot of applications and users consuming that data."

The primary objective of this test was to validate the scalability of both the GigaSpaces technology and its underlying approach to managing large environments.

A secondary objective was to evaluate Intel and IBM options for flexible, consistent deployment of GigaSpaces technology in a large-scale deployment environment. The results of this proof of concept testing validate an initial production deployment of up to 2,000 CPUs within the Bank of America **Global Corporate and Investment Banking Group**'s (GCIB) Global Markets Trading Technology (GMTT) group.

With offices in more than 150 countries, the GCIB Group focuses on companies with annual revenues of more than $2.5 million; middle-market and large corporations; institutional investors; financial institutions; and government entities. Services include M&A, equity and debt capital raising, lending, trading, risk management, treasury management, and research.

Benchmark* tests were conducted at an IBM Computing Center test facility to prove scalability prior to roll-out. The IBM test lab was equipped with 512 IBM X-355 servers running dual Xeon processors with 3 GB RAM on a 1 GB Ethernet network. Testing was executed and controlled from Bank of America at 50 Rockefeller Plaza using secure VPN connection into the IBM test lab facilities.

** The results from a "benchmark test" provide a point of reference for performance against which future performance can be measured.*

Performance was compared using different Java development kits (JDKs) including IBM JDK 5, JRockit JDK 5, Sun JDK 4, and Sun JDK 5. The JDKs ran on the Java run-time environment in a Java virtual machine.

Large-Scale Deployment Benchmark Tests

The first objective of the PoC testing was to prove that the IBM BladeCenter servers are a high-performance platform for running Bank of America's financial applications in a GigaSpaces grid computing environment.

The most common topology used for the proof of concept testing was a partitioned cache with 32 partitions. Cache topology was mirrored using 32 primary partitions and 32 back-up partitions.

The total number of objects in the cache was 15 million when using 1 KB objects, 1.5 million with 10 KB objects, and 150,000 with 100 KB objects. All transactions per second (TPS) numbers are based on single-object operations (no batch operations).

The cache tests used remote (over the network) cache access, with no local cache in the clients. The API used was the IMPA API.

In each case, the total cache size was 15 GB, which is approximately 42.8 percent of the maximum cache capacity of 35 GB. Since server performance degrades as you approach the memory limit, the benchmark tests were deliberately performed using caches of less than half the maximum cache capacity.

Table II shows a sample comparison with 1 KB objects between IBM, Sun, and JRockit Java development kits. The results are plotted on Fig. 1.

Two conclusions can be drawn from the data. First, while all JDKs performed well, the IBM 5 JDK outperformed Sun 1.4.2 and Sun 1.5 JDKs consistently. It also processed more transactions per second than JRockit 5 when 20 "writers" were used. A *writer* is a server that loads data from a variety of databases into the caches.

The superior performance on IBM and JRockit vs. Sun JDKs is to be expected, since the IBM and JRockit JDKs are optimized to take advantage of the Intel processing environment, and the Sun JDKs are not.

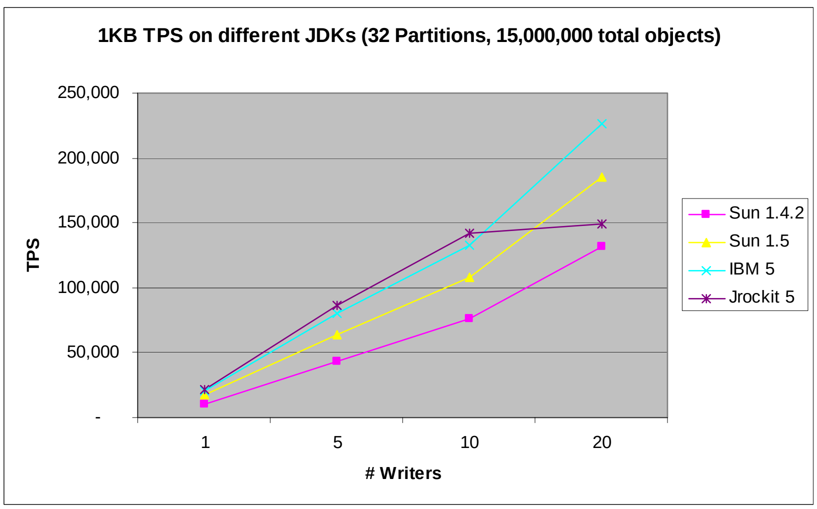
Second, the grid computing architecture scaled well on all JDK platforms, IBM in particular. For instance, when the number of writers doubled from 10 to 20, the transactions per second (TPS) on the IBM 5 JDK nearly doubled as well, from 132,210 TPS to 226,840 TPS.

These test results indicate that IBM BladeCenter servers perform extremely well in the GigaSpaces grid environment – and that you can scale up processing speed by adding more servers to the grid.

Table II. Sample comparison 1KB write on 4 different JDKs.

# Writers	Sun 1.4.2	Sun 1.5	IBM 5	JRockit 5
1	10,460	17,103	20,495	21,386
5	42,735	63,512	80,750	86,663
10	75,759	108,340	132,210	141,989
20	132,096	185,678	226,840	149,314

Fig. 1. Performance comparison of 4 different Java development kits.



Proving Scalability: Reading Data from the Cache

As discussed, a key objective of the proof of concept testing was to ensure that the CPU-intensive applications Bank of America runs are scalable in an IBM blade grid computing environment. That is, by adding more IBM blade servers to the grid, can you increase processing speed, capacity, and power proportionally?

Fig. 2 shows the throughput measured in terms of actual operations (transactions per second) performed, for three different object sizes (1K, 10K, and 100K), as a function of the number of "readers" being deployed.

A *reader* is a server that loads data already present in the cache into the CPU. As you can see, processing capacity scales linearly as you add readers to the grid, until you reach about 50 readers.

At that point, server and bandwidth limitations at our IBM test lab facility cause the grid to scale at a slightly reduced rate. Once 50 servers were in the grid computing network, adding more servers continued to enhance the grid computing system's capacity, but not in linear proportion to the number of servers added.

The conclusion: as long as the underlying network and computing resources allow, the grid computing solution scales to give linear increases in transaction processing and data throughput. Once the server and network bandwidth limitations kick in, adding more servers continues to increase performance, but not as dramatically.

Table III. Read performance (transactions per second).

Readers	1K	10K	100K
1	3,031	2,035	496
10	24,400	18,820	4,760
50	74,400	72,000	15,400
100	99,500	108,800	18,500

Fig. 2. Read performance scales linearly to around 50 readers.

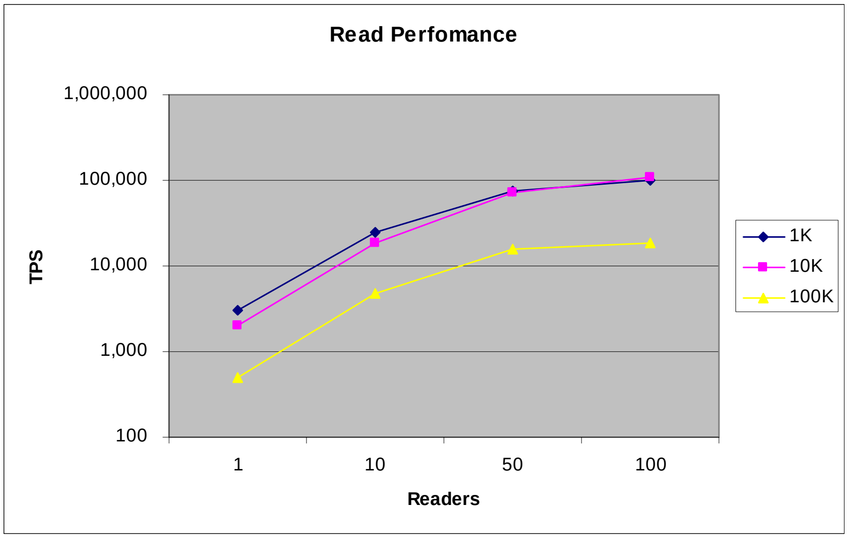


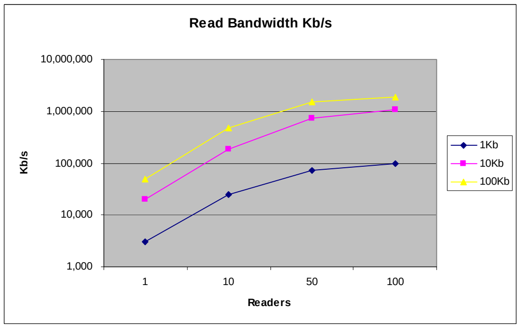
Fig. 3. shows throughput measured in actual data -- kilobytes per second -- transferred from the cache to the client, for three different object sizes (1K, 10K, and 100K), as a function of the number of readers being deployed. As you would expect, the larger the "object payload" – the net size of each object read from the data grid – the slower the throughput: it takes more time to read bigger data.

Once again, processing capacity (kilobytes per second) scales linearly as you add readers to the grid, until you reach about 50 readers. At that point, server and bandwidth limitations caused the scalability to flatten out. Adding readers continued to increase the grid computing system's capacity, but at a slower rate.

Table IV. Read performance (kilobytes per second).

Readers KB/s	1	10	100
1	3,031	20,350	49,600
10	24,400	188,200	476,000
50	74,000	720,000	1,540,000
100	99,500	1,088,000	1,850,000

Fig. 3. Read performance scales linearly to around 20 readers.



Proving Scalability: Writing Data to the Cache

We also tested the scalability of grid computing for Bank of America when it comes to *writers*, which we've defined as the servers loading data into the caches. Performance was measured in both transactions per second (Table V and Fig. 4) and kilobytes per second (Table VI and Fig. 5).

Again, the results show no surprises: the larger the objects being written into the cache, the more time it takes to load them -- and the more writers being used to load the cache, the faster the data can be written to the memory.

Once again, the system scales up in linear fashion. For instance, when you double the number of servers in the grid from 10 to 20, the throughput with 100K objects approximately doubles, from 7,770 to 14,100 TPS.

Table V. Write performance, transactions per second.

Writers	1K	10K	100K
1	20,495	7,750	931
5	80,750	35,265	4,195
10	132,210	63,520	7,770
20	226,840	114,620	14,100

Fig. 4. Transactions per second increase as more writers are added to the grid.

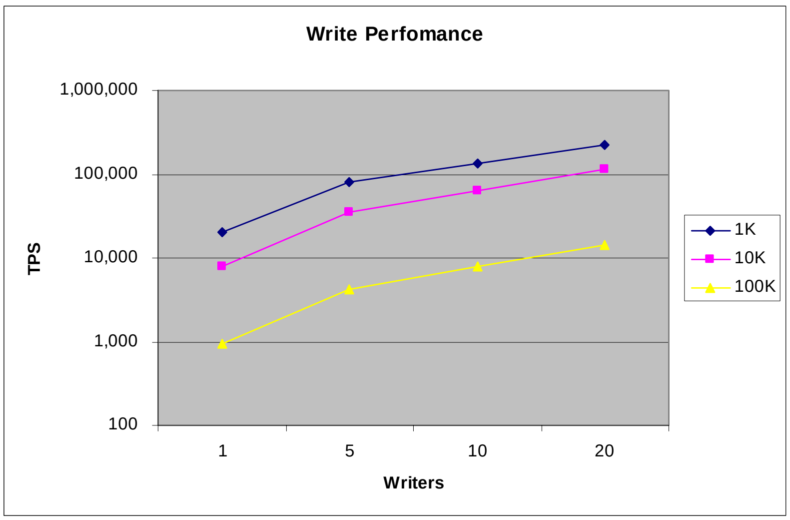
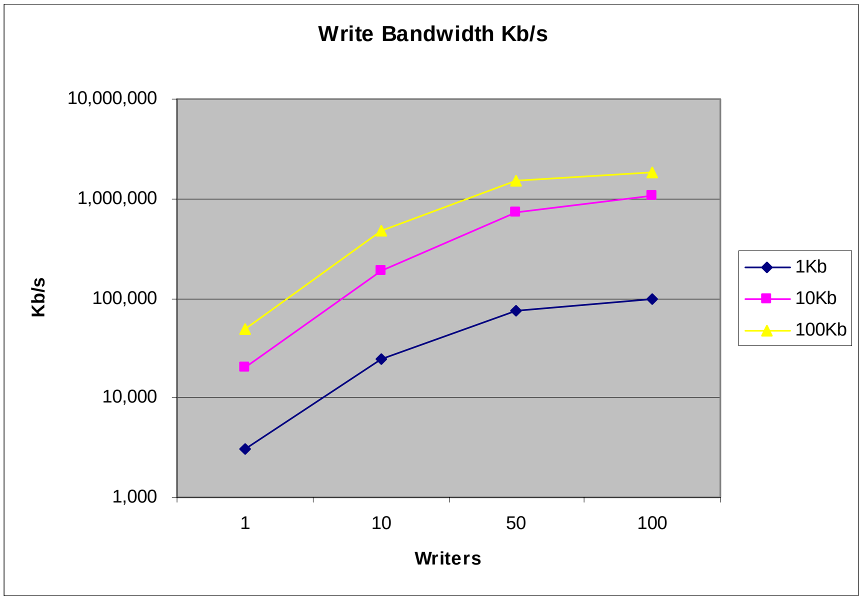


Table VI. Write performance, kilobytes of actual data per second.

Writers KB/s	1	10	100
1	20,495	77,500	93,100
5	80,750	352,650	419,500
10	132,210	635,200	777,000
20	226,840	1,146,200	1,410,000

Fig. 5. Data throughput increases as more writers are added to the grid.



Large Cache Size Testing

Would loading the cache with a huge amount of data slow down the grid system? Or could the grid architecture scale to process massive data caches? Finding the answer to these questions was the goal of the large cache size test.

Large cache size tests were performed with 192 partitions holding 150 GB and 1,022 partitions holding 1 TB of data.

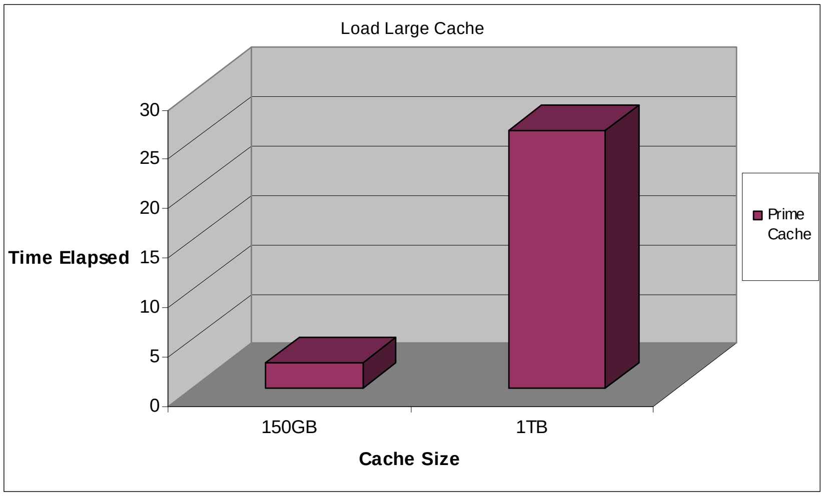
In this test, performed on the Sun JDK 1.5, 20 writers first loaded a 150 GB cache consisting of 15 million objects. The load time was 2.5 minutes.

Next, the 20 writers loaded a 1 TB cache consisting of 102.4 million objects – approximately 6.7 times the size of the first load. The load time was 26 minutes, or 10.4 times longer than the original load, as shown in Fig. 6.

The test demonstrates that even with a large cache size in the terabyte range, the grid computing system can scale to process large amounts of data quickly and efficiently.

The scaling is not perfectly linear – the second load was approximately 7 times larger than the first, but took approximately 10 times longer to process – but it’s close enough: the grid computing architecture can scale performance to handle data loads in the terabyte range.

Fig. 6. Loading of large-scale cache.



Parallel Processing Benchmarks

In financial applications, a grid computing system can break a complex calculation to its component parts and distribute these sub-calculations among servers in the grid network. Multiple servers then process these calculations in parallel (meaning simultaneously rather than in linear or serial fashion), greatly compressing the time required to complete the computation.

In the parallel processing test, we wanted to see whether adding more servers for processing increased speed in a linear fashion.

The parallel processing benchmark used a master worker pattern where a master submitted a task to be executed by multiple workers.

In the tests performed, 5 masters submitted 5 parallel tasks to 2, 32, 128, and 448 workers. The task was to determine whether an extremely large number was a prime number.

Workers performed different computations in parallel. Results, shown in Table VII, were aggregated back to the masters to yield the answer.

Table VII. Parallel processing results.

Workers	Processing Time (Seconds)	(Minutes)
2	1,387	23.11
32	129	2.15
128	50	0.84
448	20	0.34
1000	10	0.13
2000	5	0.07

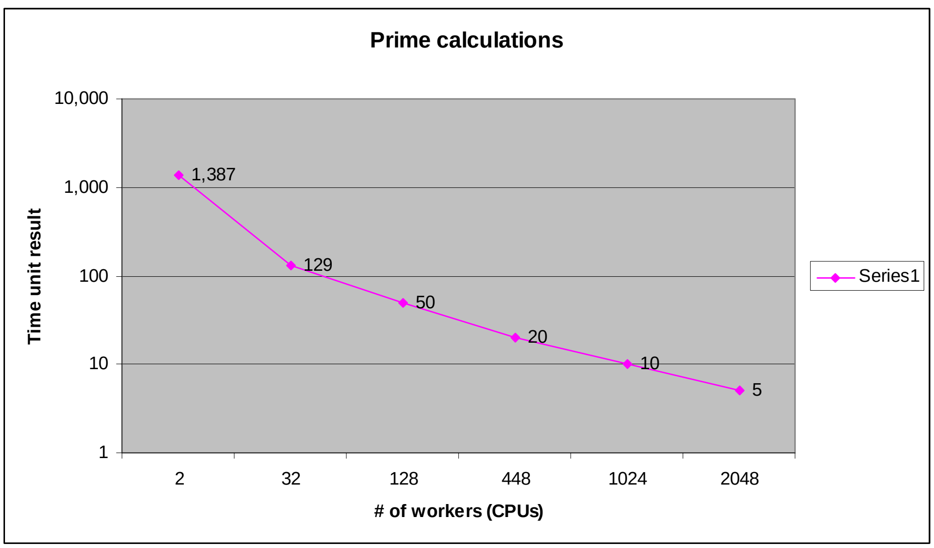
When just two workers were used, the task took 23 minutes and 7 seconds to complete. But when 32 workers were used, multiplying client processing power eightfold, completion took just 2 minutes and 9 seconds – more than a tenfold increase in speed.

Scaling up from 32 to 448 workers further reduced processing time from 2 minutes and 9 seconds to just 20 seconds, giving an increase in speed of more than six fold while using 14 times as many servers. As illustrated in Fig. 7, the speed of parallel processing in the grid computing network scales linearly as more servers (workers) are added to the grid.

The test proves that parallel processing on a grid computing network can dramatically reduce the time to complete a complex calculation, such as the financial calculations routinely performed by bank analysts. The grid computing solution can reduce compute-intensive task time significantly by running computations in parallel on multiple servers.

The problem is split into a number of separate tasks, all of which can be completed simultaneously, in seconds, by distributing them over the entire cluster. The more computers you add to the grid, the faster you can complete the calculation and get the result.

Fig. 7. Parallel processing on a grid computing environment.



Conclusions

The proof of concept testing demonstrates that GigaSpaces infrastructure software, running on IBM servers with Intel processors, can consistently and reliably scale-up for large deployments in data-intensive applications – meeting Bank of America’s requirements for speed and throughput.

The tests show that a grid computing system working with distributed caches of data can easily scale up to meet greater demand. By adding more servers, the grid computing solution can handle extremely massive amounts of data in memory, with many users and applications reading and writing to those large data caches.

In this series of benchmark tests, the grid computing system was able to process object payloads as large as 100 KB at rates of up to 100,000 transactions per second and higher, with cache sizes ranging from 15 GB to 1 TB, in configurations ranging from one to 500 client nodes.

Both IBM and JRockit outperformed the Sun JDK 5 on most tests. There was no clear winner between IBM and JRockit JDKs. On the JRockit JDK, 100 readers read 1 KB objects at a rate of approximately 150,000 per second. On the IBM JDK, 20 loaders wrote 1 KB objects at a rate of approximately 227,000 per second.

The results demonstrated GigaSpaces ability to support large cache size made of many partitions with multiple readers and writers running on Intel-based IBM servers. In most cases, GigaSpaces performed better on IBM and JRockit JDKs than on Sun JDKs.

Parallel processing, reading data from the cache, writing data to the cache, and the ability to load extremely large caches were all proven scalable with GigaSpaces infrastructure software running on the IBM grid computing system. As many as 200,000 objects were being written and read to the data grid concurrently, without glitches or errors.

“Grid computing is easy to scale,” says Oltman. “You can distribute tasks to available computing resources without changing applications or writing new software unless a new feature is needed.”

Benefits of grid computing include improved service levels, leverage of existing computing infrastructure, increased asset utilization, lower operating costs, scalability, and infrastructure acquisition avoidance. IBM’s grid computing projects have, so far, generated an average return on investment within 6 to 9 months and a total IT savings of 3 to 10 times their cost.

“As global capital markets continue to become increasingly electronic, the reliability, scalability, and sheer performance of our technology solutions are repeatedly underscored,” says Bank of America’s B.J. Fesq.

“The most successful firms in electronic trading don’t always have the best prices, the smartest traders, or the latest algorithms. Rather, the e-trading leaders are typically those firms who are continually investing in technology that will make them fastest to market in quoting and execution, with uncompromised availability.”

The increased throughput, computing efficiency, and processing power provided by the IBM grid computing solution can enable Bank of America to be the fastest to market in quoting and execution for electronic trading. Mirroring of data, with automatic rerouting to a different node in case of failure, ensures the highest levels of system availability.

The Next Step

Lines of business (LOBs) across capital markets – including equities, derivatives, mortgage-backed securities, global foreign exchange, electronic trading, and the rates business – have GigaSpaces-based projects going into production in 2006.

The success of the proof of concept testing should give LOBs the confidence to use the Intel/IBM platform for their 2006 GigaSpaces projects. A number of these LOBs expect to purchase additional hardware, providing the increased capacity needed to support various cache sizes with appropriate quality of service.

For more information on IBM grid computing solutions, contact your IBM representative today. Or visit the URL below now:

www-1.ibm.com/grid/about_grid/ibm_grid/products_services.shtml

For more information on Gigaspaces, Inc. grid solutions, visit:

www.gigaspaces.com

###