WRITE ONCE.
SCALE ANYWHERE.

GigaSpaces XAP and Cisco UCS Joint Solution

Massive Performance at Unbeatable Cost

© Copyright 2010 GigaSpaces. All Rights Reserved.

Abstract

This whitepaper provides a detailed description of the joint value proposition behind Cisco Unified Computing Resources (UCS) and GigaSpaces eXtreme Application Platform (XAP).

The executive summary provides an overview of the joint solution and value proposition.

The remainder of the paper provides technical architects and engineers with an in-depth technical review of the joint solution. The technical section is structured in the following way:

- Introduction: Summarizing the market opportunity for this joint solution
- A closer look into GigaSpaces & UCS – providing a short technical overview of UCS and the GigaSpaces platform with specific emphasis on Cisco's unique extended memory technology.
- The benchmark – Measurement of the performance and scaling gain provided through the combined solution.
- Use case scenarios –Specific application scenarios that most benefit from the joint solution. In this section we apply the performance gain as measured in the benchmark section and translate those numbers into the actual value in terms of efficiency for each particular scenario.
- Total Cost of Ownership (TCO) benefits – In the summary section we will provide a TCO-driven summary of the value proposition behind the joint solution for existing users of GigaSpaces, as well as users that are using legacy application server and database centric solution.

Table of Contents

Executive Summary

A Better Platform for Memory- and Data-Intensive Applications in Today's Data Centers

Enterprises, Infrastructure-as-a-Service (IaaS) providers, and various other organizations are faced with managing an increasing portfolio of applications that require higher throughput and lower latency. As loads grow, systems are scaled up or scaled out to meet performance requirements. But scaling presents its own set of challenges in terms of cost and performance efficiency.

By taking advantage of larger memory footprints, such as that provided by the Cisco Unified Computing System (UCS) Extended Memory Technology, along with a deterministic network fabric provided by Cisco's UCS 6100 series Fabric Interconnects and Nexus switches, organizations can improve application performance and reduce latency, while minimizing their physical footprint and operational overhead.

To further improve cost and performance efficiencies, Cisco and GigaSpaces have joined forces to design a system that addresses the issues presented by scaling applications across the data center. By combining GigaSpaces In-Memory Data Grid (IMDG) technology with Cisco's UCS platform, organizations can:Greatly reduce overall data center costs

- Manage large amounts of memory using less hardware
- Reduce application latency by a over 90%
- Simplify application delivery and infrastructure operations
- Enable applications to scale up (add more memory) or scale out (add more servers) easily and automatically
- Improve mean time to recovery (MTTR) with advanced systems event integration and workload management
- Span applications across data centers
- Leverage multiple development languages and styles independent of system architecture
- Get simplified parallel programming and enable applications to leverage multi-core devices more efficiently
- Take advantage of system and network improvements including: 10GE, I/O virtualization, application partitioning, dynamic scaling, bare-metal or virtualization designs

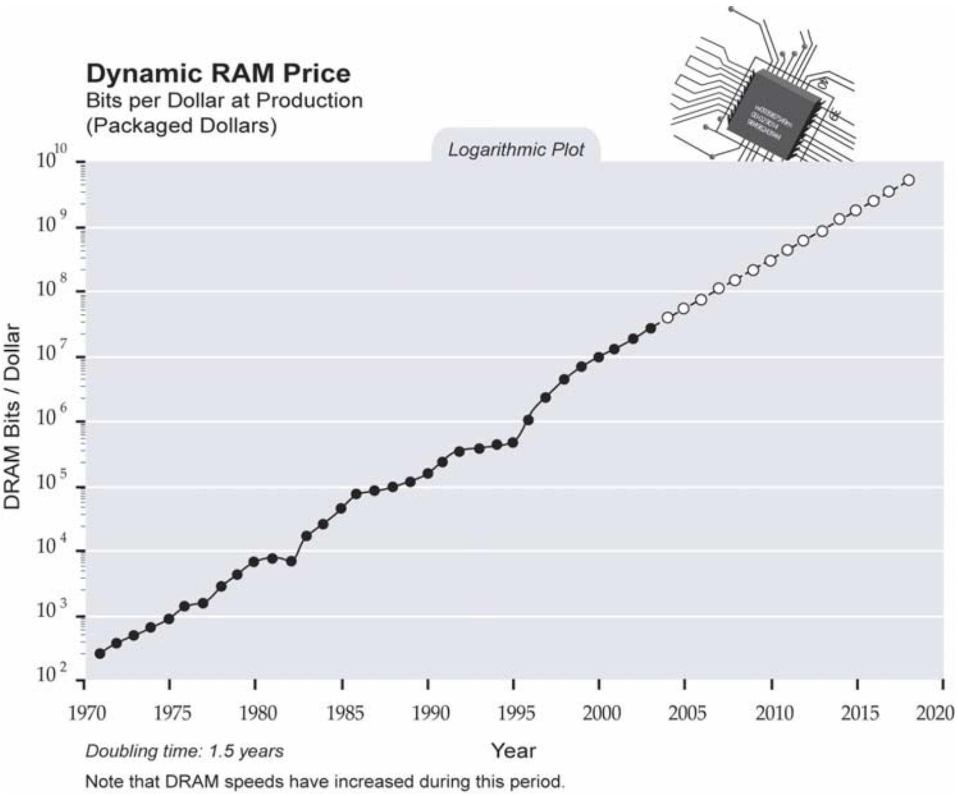
In short: *Enable systems to scale beyond anything available on the market today*.

Together, Cisco and GigaSpaces provide the most advanced system for application scaling, reliability, and performance, simplifying the application delivery life cycle, and enabling organizations to focus on their core business.

1. Introduction

Improving Efficiency by 100x/20x Through Better Utilization of New Memory, Network, and Computing Resources

Memory resources are now available at greater capacity and lower cost. With new large memory capacity it is now possible to manage terabytes of data completely in-memory. At the same time, network bandwidth is growing exponentially with 10GE already available, and 40G and 100G coming soon. Multi-core computing resources are becoming mainstream, with 8- to 12-core systems already available, providing computing power that was previously available only on super-computers at an affordable cost.



The increase in available memory capacity per $ over a period of 50 years.

The reality, however, is that most existing applications, databases, and messaging platforms were designed with the assumption that the network is a bottleneck, and that memory is expensive and of limited capacity. These applications rely heavily on disk storage to manage their data. However, disk-based devices cannot keep up with the increasing demand for higher performance and scaling.

Relational databases have been the storage system of choice for several decades, but they do not scale to the level required by today's large-scale applications. Virtually every popular Web application has found that a single relational database cannot meet its throughput requirements. As the site grows it must undergo a series of massive revisions, each introducing ad hoc techniques to scale its storage system, such as partitioning data among multiple databases. These techniques work for a while, but scalability issues resurface when the site grows to the point of needing a new scale yet again, or a new feature is introduced that requires yet more special-purpose techniques.[1]

It becomes clear that to take full advantage of new memory, network, and computing resources, a new class of platform is needed--one that does not rely on disk storage for managing data, and which can exploit newly available increased network and multi-core capacity through a combination of distributed scale-out and scale-up patterns.

As noted in a recent research paper published by Stanford University, it is expected that with full exploitation of these new resources, organizations can improve their overall application utilization and efficiency by a factor of 100-1000x compared to existing disk-based solutions.

"RAMClouds become much more attractive for applications with high throughput requirements. When measured in terms of cost per operation or energy per operation, RAMClouds are 100-1000x more efficient than disk-based systems and 5-10x more efficient than systems based on flash memory." [1]

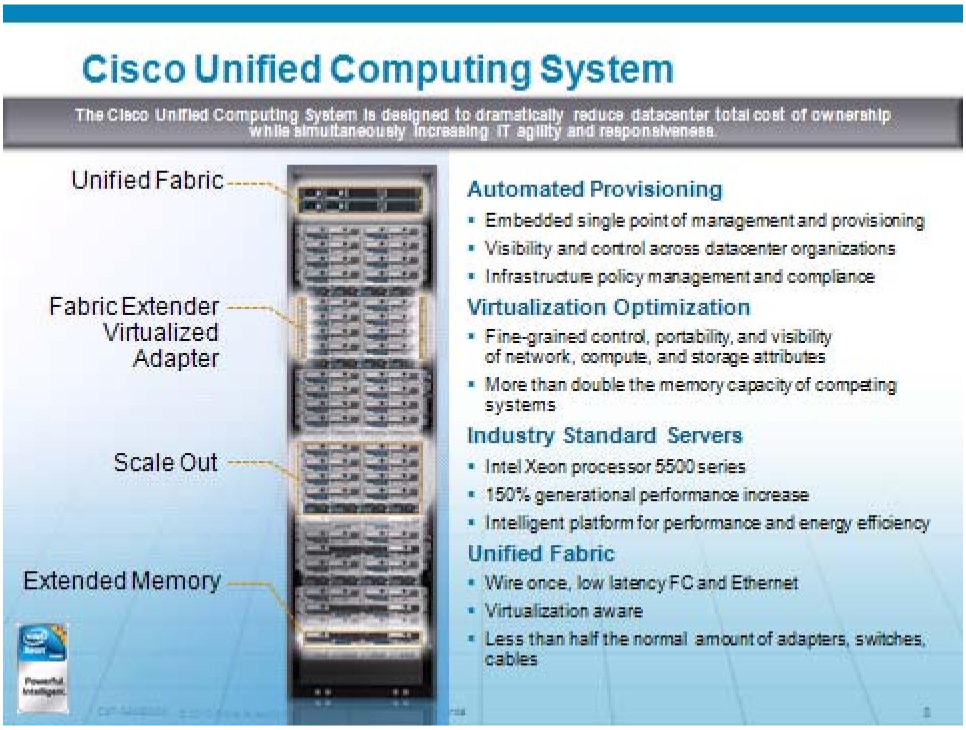
Beyond the value of memory- vs. disk-based architecture, the use of a fully optimized stack from the application middleware down to the hardware – such as the one provided by GigaSpaces and Cisco – can yield an additional 20x improvement in terms of capacity (the amount of data that can be managed on the same number of machines), and a 3-6x throughput improvement compared to existing memory-based deployments that are running on current commodity network and hardware devices.

---

[1] Stanford University – *The Case for RAMClouds*

2. A Closer Look at Cisco and GigaSpaces

Cisco Unified Computing System (UCS) is a single integrated system that brings together computing, network, virtualization, and storage access into one cohesive system. Unlike most existing servers, it was primarily designed for virtualized and large-memory workloads, and as such it provides a high degree of density. With up to 384GB of RAM per server and the latest in Intel server technology, UCS makes it possible to store terabytes of data entirely in-memory over a significantly smaller amount of resources (as little as 1/20 compared to existing hardware).

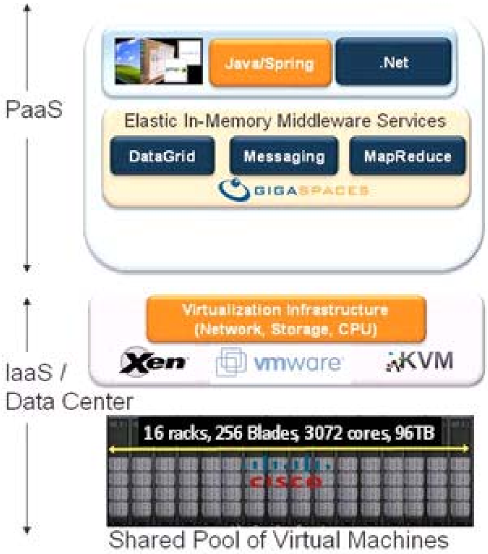


GigaSpaces eXtreme Application Platform (XAP) is a unique in-memory middleware platform that supports leading industry standards, and enables applications to fully exploit the new hardware capabilities available with the UCS. It does this by taking full advantage of the memory resources in each node, as well as enabling distribution of the data across all nodes.

The GigaSpaces framework provides many high-end features, such as parallel query processing, at no extra cost. The power of this approach comes in part from familiar concepts like partitioning. But the high speed of memory as compared to disk also enables entirely new levels of performance and reliability, in a package that is relatively simple and easy to understand and deploy.

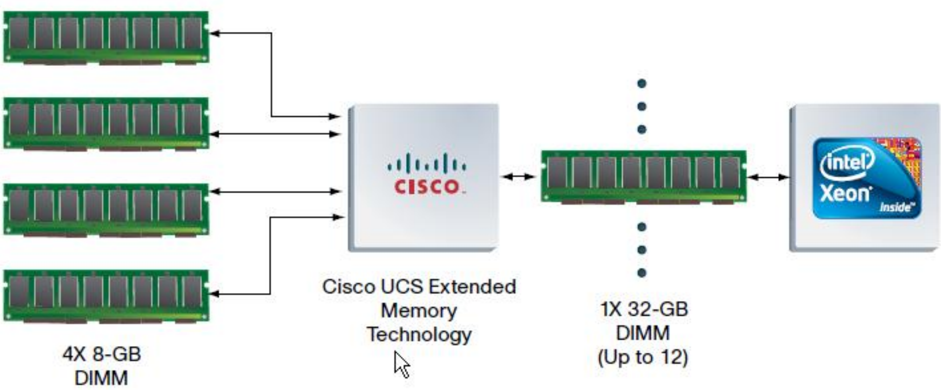
The In-Memory Stack for the Next-Generation Data Center

The combination of UCS as the in-memory and network resource, and GigaSpaces as in-memory middleware forms a full in-memory stack that fits the needs of the next generation data center.



2.1. Understanding the Unique Value of Cisco UCS Extended Memory

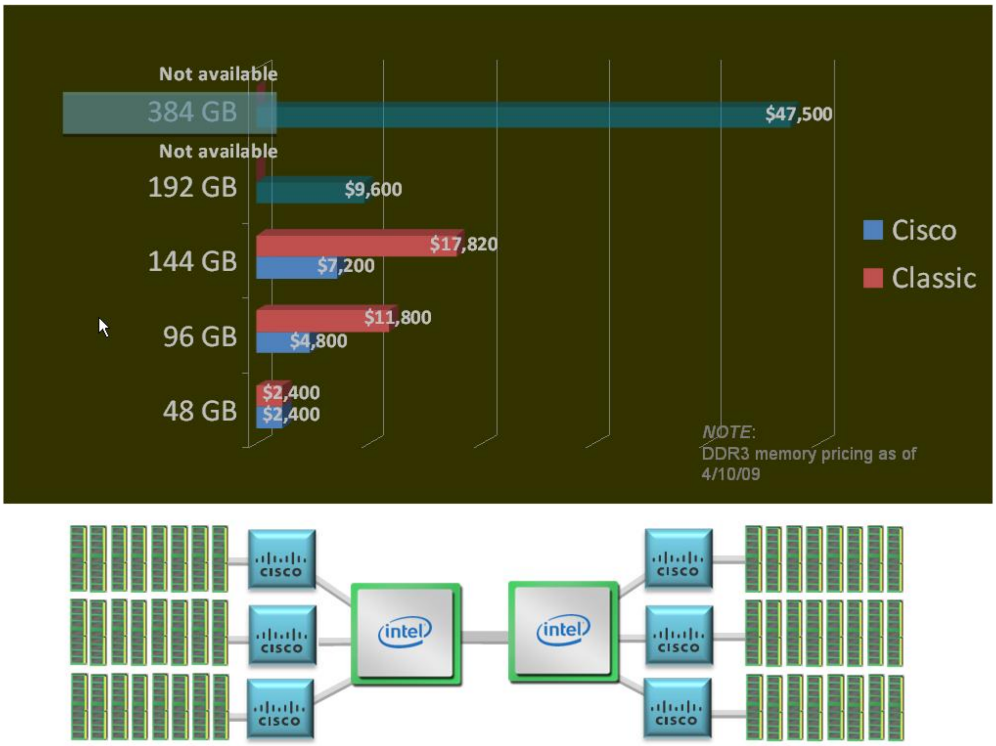
Cisco UCS Extended Memory Technology provides flexibility between memory cost and density, or in other words, the number of machines required to manage a particular set of data. Extended Memory Technology uses a high-performance ultra-fast technology that is implemented in its ASIC to allow 48 memory modules (DIMMs) to be addressed at high speed. The total memory address space per blade jumps to 384 GB at 1333 MHz speed compared to 96 GB at 1066 MHz, or 144 GB at 800 MHz, on alternative hardware provided by other x86 based 2-socket server vendors that can use up to 18 memory modules (DIMMs).



CISCO EXTENDED MEMORY TECHNOLOGY MAKES FOUR PHYSICAL DIMMS APPEAR TO THE CPU AS A SINGLE, LARGE, LOGICAL DIMM

UCS Extended Memory Technology architecture provides two options that ensure its cost effectiveness compared to alternatives:

- **Low-cost option:** Delivers a memory footprint of up to 192 GB using low-cost 4-GB DIMMs, rather than more expensive 8GB DIMMs; enables data center operators to save up to 60 percent on memory costs compared to traditional two-socket servers. These savings come without the performance penalty incurred when traditional systems use 18- rather than 12-DIMM slots.
- **Large-footprint option (density):** Can accommodate extremely memory-intensive workloads. With up to 384 GB of memory available using 8-GB DIMMS, the Cisco UCS B250 M2 and UCS C250 M2 servers deliver the largest memory footprint available in any two-socket sever using Intel® Xeon® 5600 series processors. This capacity rivals that of current four-socket x86-architecture servers and provides an economical two-socket solution alternative to larger, more expensive four-socket servers. (See graph, 192GB-384GB)



THE CISCO EXTENDED MEMORY TECHNOLOGY LOW-COST AND LARGE-FOOTPRINT OPTIONS

2.2. Getting Better Flexibility Through Distributed Computing

The use of Cisco extended memory in combination with distributed computing models provided by GigaSpaces XAP enables extending the two options (low cost and large footprint) beyond a single blade instance. In other words you can create a virtual in-memory data cloud that spans from a single blade to 1000 blades and looks as one big memory cloud to the application that is using it. As with single blade setup we have two deployments options:

- **Low-cost option:** Users can use multiple-low cost memory blades configuration to reduce the total cost per GB/RAM
- **Large-footprint option:** Users can easily manage terabytes of data in memory on the same number of machines.

3. GigaSpaces Benchmark

The benchmark measures the performance and scalability gain achieved by combining UCS extended memory and GigaSpaces in-memory middleware.

The test measured the two ends of the spectrum of a typical application deployment: *embedded*, where the entire application resides in a single process, and *distributed,* where a cluster of machines is grouped together to form a large memory cloud.

A typical enterprise application deployment combines embedded and distributed units. We expected that creating embedded units with higher capacity would translate into smaller deployments, **reducing the total cost of ownership**. At the same time, we expected larger embedded units to **enable a significantly larger deployment** that could easily manage terabytes of data in-memory, making it possible to hold the entire application data in-memory.

From a performance perspective, we expected embedded units to perform significantly better than distributed units. While a typical enterprise application would not fit into a single embedded unit, we expected that with larger embedded units and the ability to collocate the relevant business logic with the data, we could **reduce the number of network hops, gain greater overall performance, and lower latency**. We also expected to gain substantial performance and scalability compared to current disk-based systems. With pure memory deployments we expected to gain greater efficiency and therefore use less hardware to meet given performance goals.

The next two sections discuss the performance results of benchmarks done for both embedded and distributed scenarios. The exact configuration of the tests is provided in Appendix A.

3.1. Embedded (Scale Up) Benchmark

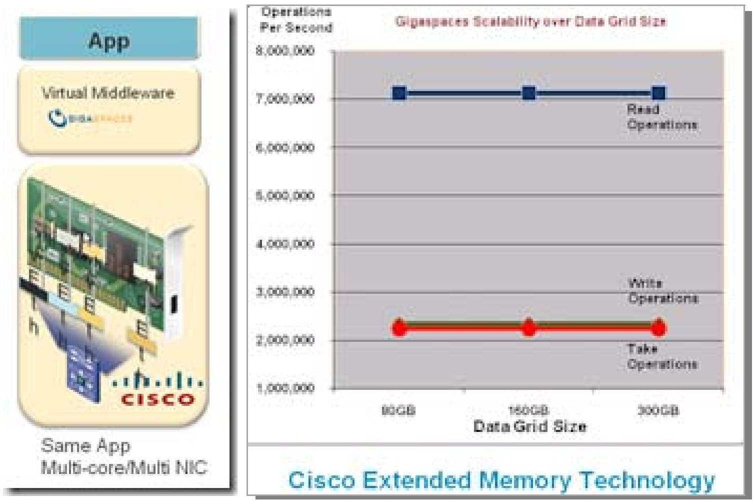
This *embedded* test measured a typical application deployment (processing unit) where the application is collocated with the data to minimize network and latency overhead.

This deployment type presents two main challenges:

- Scaling at the capacity level: How to increase application data in a single process without causing GC hiccups and CPU starvation (there is a correlation between the amount of data and CPU capacity required to process that data).
- Concurrency on multi-core: How to avoid locking on shared data in order to enable scaling through multi-threading.

GigaSpaces provides unique optimization to address the challenges mentioned above:

1. **Seamless collocation optimization:** The GigaSpaces runtime automatically detects if the data space is collocated or distributed, and optimizes access to the data in runtime. In this specific case it detects that the data is collocated and uses a direct reference to the data, avoiding serialization or network calls.
2. **Lock-free data model:** To enable full exploitation of the underlying multi-core resources, the GigaSpaces data structure is designed with a unique lock-free data model. This enables scaling up even stateful transaction applications very simply. GigaSpaces also provides an API abstraction similar to the those provided by functional languages such as Scala, which greatly simplifies parallel programming.



The benchmark results as outlined in the diagram above show that we were able to achieve throughput of 2.3M write/sec and 7.1M read/sec. We were also able to increase capacity up to 280 GB data per VM without any performance decrease.

Embedded Benchmark Analysis

The following conclusions were reached by comparing the benchmark results with our current benchmark (see more on the GigaSpaces public benchmark page).

Throughput results

	Current (Sun4450 Intel 7460 CPU)	UCS (Cisco UCS B250 M2 with Intel X5600 CPU)
Read	**1.8M**	**7.1M**
Write	**1.1M**	**2.3M**

Test results, as shown in the table above, indicate that UCS provides a **100 percent performance gain** compared to Sun/Intel-based hardware.

Capacity Density Analysis

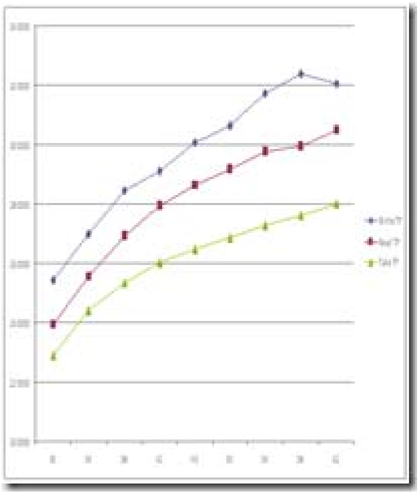
A single JVM can hold up to 300 GB of data with no performance drop. Current production systems in GigaSpaces run at a maximum capacity of 10 GB per VM (most run on a smaller capacity). Based on this, with UCS **we can reduce the number of machines required to manage the same capacity by 95%**.

3.2. Distributed (Scale Out) Benchmark

The *distributed* test measures a typical application deployment, where data is spread across a cluster of machines. In this case, the application is remote to the data. Access to the data, therefore, involves networking and serialization. This model is also referred to as a scale-out model – scaling is achieved by adding more application instances (partitions) and spreading the load and data among those instances. The main challenges in this scenario relate to the transition between the embedded (scale-up) to distributed (scale-out) scenario.

Typically, a scale-up application is written completely differently than a similar distributed application. In many cases, moving from scale-up architecture to scale-out architecture requires a complete redesign, making it almost impossible to measure and compare the two scenarios.

GigaSpaces provides a seamless transition between embedded (scale up) and distributed (scale out) modes. In embedded mode, the application accesses data through a specific GigaSpaces abstraction we call *space*. The space detects if the data is collocated or remote. If it detects remote data-space, it uses network calls to access the data. All this is done implicitly, outside the user code. In addition, the remote space is cluster-aware and is therefore capable of routing and load-balancing if the application calls between clusters of machines, providing a seamless scale-out model. In other words, the application does not need to “know” whether the system is embedded or remote.



The diagram above illustrates the linear system scaling that occurred as the number of concurrent clients grew. The system reached approximately 320,000 operations/sec. A detailed analysis is provided below.

Distributed Benchmark Analysis

The following conclusions were reached by comparing the benchmark results with our current benchmark (see more on the GigaSpaces public benchmark page).

Throughput results

	Current (Sun4450 Intel 7460 CPU)	UCS
Read	90K	320K
Write	45K	305K

The results indicate an improvement of 600% on write throughput, and 300% on read throughput. This can be translated to a savings factor of 6x (write) and 3x (read) in terms of hardware resources required to achieve the same throughput goals.

Capacity Analysis

The tests shows that the system can scale linearly in terms of capacity as the number of machines grows; or, adding more machines increases system capacity in linear proportion, as expected.

3.3. Benchmark Results Summary

Below is a short summary of the results from the benchmark.

Maximum Throughput	
Embedded (scale up)	7.1M reads/sec, 2.3M on write/take
Distributed (scale out)	~300,000 ops/sec
Capacity	There was no performance drop while increasing the capacity
Scaling	The system scales linearly as long as CPU resources are not exhausted

4. Use Cases

This section illustrates use case scenarios that show how the benchmark numbers translate into concrete value.

4.1. Business Application Use Case: Real-Time Analytics

Analytics applications, as the name suggests, are applications that analyze a given set of data and produce a report of the analyses outcome. Real-time analytics applications are applications that analyze the data in near real time and keep continuously update the report as changes occur.

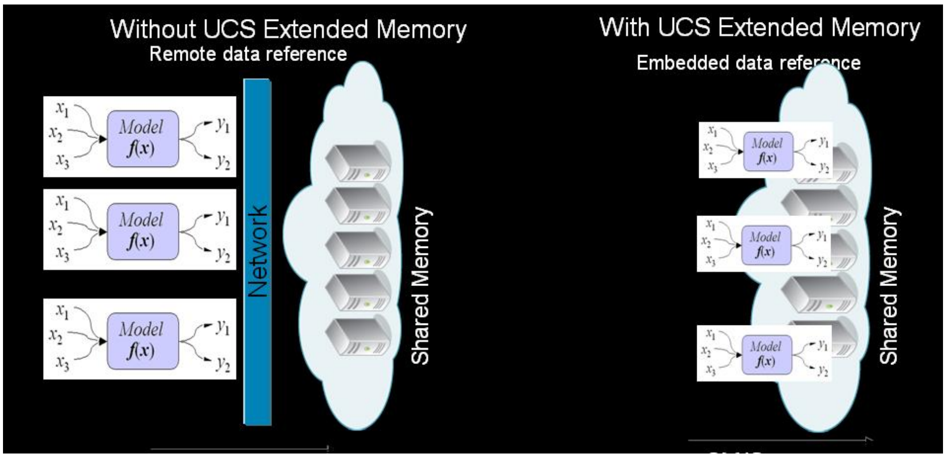
One of the most common types of real-time analysis applications is risk management. Risk management tries to predict current risk based on historical data, also known as reference data.

A common pattern for scaling a risk management application is by partitioning the analytics process based on portfolio. Each portfolio is processed independently in parallel to other portfolios, often on different servers.

All the portfolio analysis processors need access to the same reference data.

To gain higher accuracy in risk management the application often tries to look deeper into the history, which in turn requires a larger amount of reference data.

Often, the amount of storage space needed to store the reference data exceeds the size of memory available in a single process. It is therefore a common practice to store that data in a remote database or memory storage. In this use case, we refer only to the scenario using distributed memory storage to store the reference data, as illustrated in the left side of the diagram below.



Scenario *without* UCS Extended Memory ("Before")

The diagram above illustrates the typical real-time risk management application architecture. Risk processes are distributed into partitions based on portfolio. All processes access shared reference data to calculate risk. Because the reference data does not fit into the memory of each unit, it is stored in remote storage, and each analytics process must access the data through a remote call. According to our benchmark, that would be at a maximum rate of 300K/sec.

Scenario *with* UCS Extended Memory ("After")

With UCS, extended memory enables easy storage of the entire set of reference data in-process. Each analytics process can access the data without going through a network call. According to our benchmark, that would occur at a rate of 2-7M reads/sec, which is approximately ten times faster than a distributed scenario.

Summary of UCS Benefit for Risk Management Applications

UCS extended memory enables storing the entire set of reference data in-memory, which translates into a performance boost of up to 10x, reducing the time it takes to produce an analytics report. From a business perspective, this translates into greater accuracy in how risk is managed, as more simulations can be run, or the same number can be run with better real-time resolution – in other words, knowledge of risk exposure would occur closer to the time the risk is formed, thereby minimizing actual exposure.

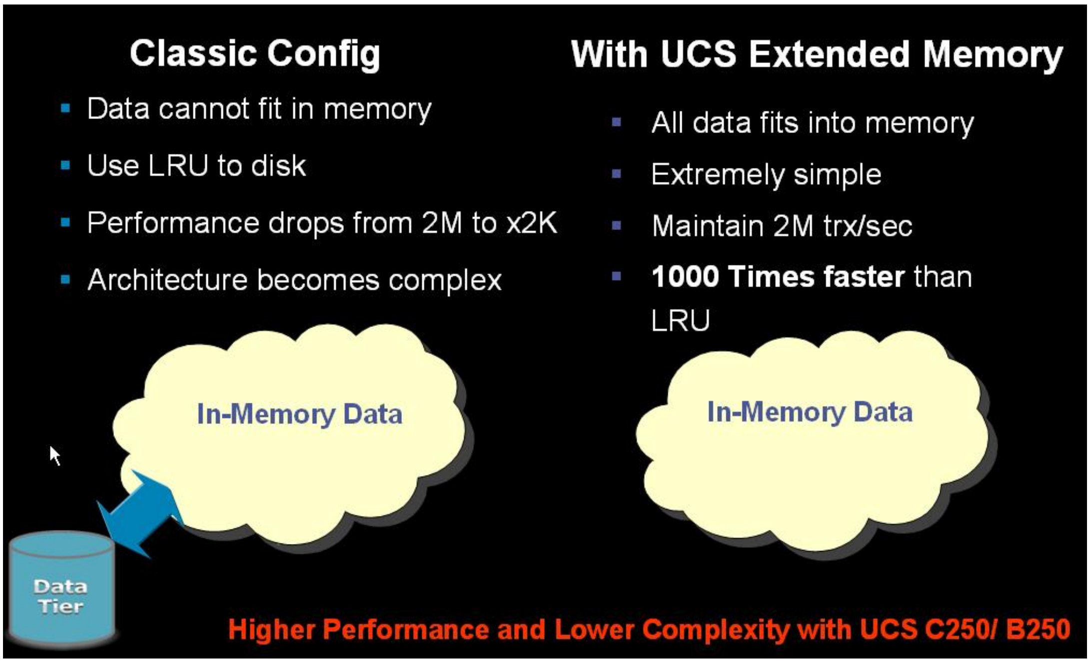
> "Real time drives database virtualization – Database virtualization will enable real-time business intelligence through a memory grid that permeates an infrastructure at all levels."
> (James Kobielus, Forrester Research Senior Analyst)

4.2. Memory-Based Solutions Use Cases

This section does not refer to a specific business application, but rather to horizontal use cases that are applicable to a wide variety of applications – typically high-performance applications that commonly face database bottlenecks.

We will compare two memory-based solutions designed to accelerate data performance. The first (left side of diagram) uses memory to cache part of the data. In the second scenario we leverage the UCS large memory capacity and ability to store the entire data in-memory.

4.2.1. In-Memory-Database



***Without* UCS (Before)**

Most existing databases use memory to reduce disk access contention. But these databases are not designed to fully exploit memory resources. According to the Stanford research[2], even a **1% miss ratio for a DRAM cache costs a factor of 10x in performance** (reduces performance by 90%). The main challenges that prevent full use of memory resources in most current databases are cost and available memory capacity. As a result, performance is severely limited by the time it takes to access the remote disk. These limitations also lead to extremely complex architecture, due to continuous tuning of the application to ensure that the *right* part of the data resides in-memory. Because data access patterns tend to change quite frequently, this is a costly operation. To maintain the performance gain of memory vs. disk access, a write-behind approach is often used, in which updates to disk are asynchronous. In this mode, consistency is compromised as well.

***With* UCS (After)**

According to the benchmark, UCS extended memory makes it possible to manage 15-20x the amount of data in-memory, per partition. This makes it possible to store the entire data set in-memory, and gain not only 10x the performance but also great simplicity, because the application no longer needs to deal with a *miss* ratio in the cache; and at the same time, there is no need to negative effect on consistency as all the data resides in-memory.

Online Retailer		Airline Reservations	
Revenues/year:	$16B	Flights/day:	4000
Average order size	$40	Passengers/flight:	150
Orders/year	400M	Passenger-flights/year:	220M
Data/order	1000 - 10000 bytes	Data/passenger-flight:	1000 - 10000 bytes
Order data/year:	400GB - 4.0TB	Passenger data/year:	220GB - 2.2 TB
RAMCloud cost:	$24K-240K	RAMCloud cost:	$13K-130K

Table 3. Estimates of the total storage capacity needed for one year's customer data of a hypothetical online retailer and a hypothetical airline. In each case the total requirements are no more than a few terabytes, which would fit in a modest-sized RAMCloud. The last line estimates the purchase cost for RAMCloud servers, using the data from Table 1.

As the table above illustrates, storing the entire yearly data of an online reservations system or an online retailer can be mapped into a fairly simple deployment with UCS:

- Online Retailer – 4 UCS chasses
- Airline Reservation – 2 UCS chasses

4.2.2. Active RAM Storage

In this case we will compare the use of memory as an alternative to disk- or flash-based storage.

***Without* UCS (Before)**

Most existing storage systems (disk or flash-based) are “passive” resources, meaning that every meaningful operation on the data must go through a cycle of copying the data into application memory, and then re-storing it in the storage device. This leads to costly serialization/de-federalization and network overheads. In a real-time application this places severe limits on the latency and throughput of the system.

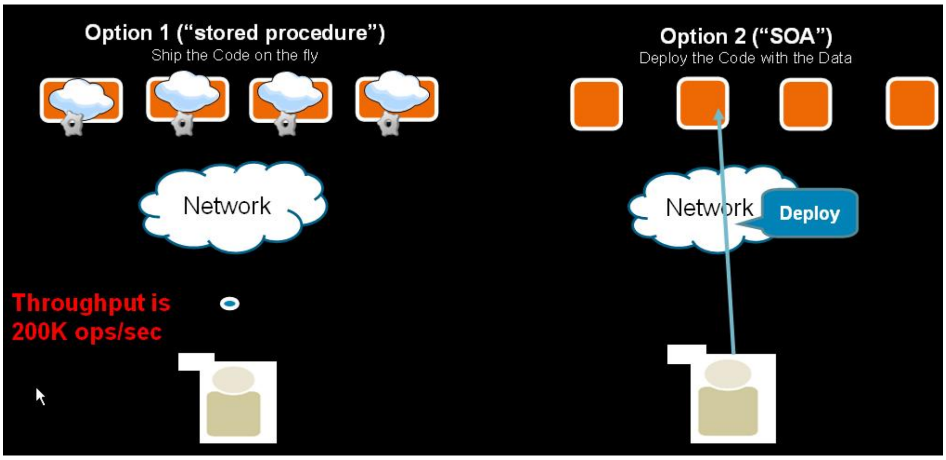
---

[2] Stanford University – *The Case for RAMClouds*

With UCS (After)

With UCS, the entire data set "lives" within the application memory. As a result, data can be accessed by reference. This can translate into extremely more efficient performance than disk or flash memory, as noted in the Stanford[3] research:

“RAMClouds become much more attractive for applications with high throughput requirements. When measured in terms of cost per operation or energy per operation, RAMClouds are 100-1000x more efficient than disk-based systems and 5-10x more efficient than systems based on flash memory.”



BETTER LATENCY AND THROUGHPUT BY PUSHING THE APPLICATION LOGIC INTO THE DATA

The fact that the data reside in-memory means application logic that depends on that data can be pushed into the machine hosting the data. For data-intensive applications, pushing the code to the data is significantly more efficient then trying to bring the data to the application. As the benchmark shows, this can translate into 10x the performance in access time – from 200K ops/sec to 2M ops/sec.

There are two models for pushing the data into the UCS memory cloud:

1. **Map/Reduce:** Code is passed dynamically per request, as shown in the left side of the diagram above.
2. **SOA:** Logic is deployed into the data server permanently, as shown in the right side of the diagram above.

4.3. Taking Advantage of UCS Multi-Core Computing Power

Because UCS provides tremendous data capacity in memory coupled with high-speed network and computing power, not only is it possible to store huge amounts of data in-memory, it is also easy to exploit the full potential of UCS computing power by running application logic on the data machines.

> “It's quite clear that the business intelligence/data-warehouse industry is moving toward a new paradigm wherein the optimal data-persistence model will be provisioned automatically to each node based on its deployment role -- and in which data will be written to whatever blend of virtualized memory and disk best suits applications' real-time requirements.“
> (James Kobielus, Forrester Research Senior Analyst)

[3] Stanford University research – *The Case For RAMClouds*

5. Total Cost of Ownership (TCO) Benefits

This paper provided an overview of the Cisco UCS and GigaSpaces XAP joint value proposition, and presented high-level proof points that demonstrate the actual performance and scaling gains of the joint solution as well as a set of use cases in which these gains would best be realized.

This section summarizes those benefits in terms of what most businesses consider a top priority - impact on the bottom line, or total cost of ownership (TCO). Because TCO comparisons can vary based on the number of alternative solutions available and the number of options in each alternative, this section focuses on providing the reference material that should be taken into consideration when calculating the specific TCO in each of the scenarios rather than a fixed set of numbers for a particular scenario.

TCO is analyzed for users in the following categories:

- TCO for existing GigaSpaces users: Users already working with GigaSpaces (or an alternative solution with similar capabilities) on non-Cisco UCS hardware platforms.
- TCO for legacy applications: Users working with legacy application servers or other database-centric solutions, and are considering alternative approaches to meet their efficiency and scaling demands.

5.1. TCO for Existing GigaSpaces Users

This section provides a TCO-reduction outlook for users of GigaSpaces (or an alternative solution with similar capabilities) over non-Cisco UCS hardware as compared with the Cisco/GigaSpaces joint platform.

Based on benchmark results with the Cisco UCS, GigaSpaces users can expect the following TCO reductions:

- **Cost reductions through Memory Density:** The ability to store more data within a single JVM (up to 270GB) enables a reduction in the total number of machines and GigaSpaces instances required by the application. Internal testing has shown a reduction of up to 95% in certain scenarios.
- **Cost reductions through better throughput:** With UCS and XAP, users can achieve 3-6x better throughput than with existing hardware configurations. In most cases this translates into a similar proportion of consolidation of the number of servers and/or additional existing resource scaling.
- **Cost reductions in maintaining large-scale systems:** With the combination of UCS and XAP, users can store terabytes of data in a single physical configuration. This reduces architectural complexity, system maintenance, and overall requires fewer moving parts in large-scale deployments, which are typical in distributed environments. In addition, UCS' unified, embedded management model – which extends across multiple chassis and thousands of virtual machines – further reduces the operational burden normally associated with large scale-out environment. Assuming a standard server configuration of 10GB of RAM, up to a 97% reduction of physical servers, not to mention associated management modules, is possible.

The TCO impact of putting your entire data in-memory:

Many existing large-scale GigaSpaces deployments can store only part of the data in-memory due to the memory capacity limitations of existing server configurations. With the expansive memory capacity and extended memory management capabilities of Cisco UCS combined with GigaSpaces' In-Memory Data Grid technology and market leading architecture, the entire data set can easily be mapped in-memory. This greatly improves application performance, decreases transactional latency, and reduces the overall development and maintenance costs associated with the complexity of the legacy solutions.

5.2. TCO for Legacy Applications

In this section we will cover the TCO reduction possible by moving from a legacy application server or other database-centric solution to a memory-based distributed solution provided through the combined Cisco UCS and GigaSpaces XAP offering. The TCO reduction is represented by the following categories:

- Memory vs. disk-based solution TCO: In this section we will discuss how the efficiency gained through the use of memory versus disk based devices can translate to significant cost savings.
- Latency TCO: Latency can have varying impacts on the business depending on the type of business and the purpose of the application. In this section we will outline how the use of GigaSpaces and UCS can significantly reduce your application latency and the cost associated with it.
- Scaling TCO: In this section we will cover the cost benefit of moving from a non linearly scalable system such as the one provided today by most application servers and database centric solutions to a fully linearly scalable solution provided through GigaSpaces and UCS.
- Downtime costs: Downtime can have a significant cost impact on the business. In this section we will discuss how the use of GigaSpaces and UCS can reduce the downtime cost by eliminating the main inhibitors for failure - number of moving parts and human errors.

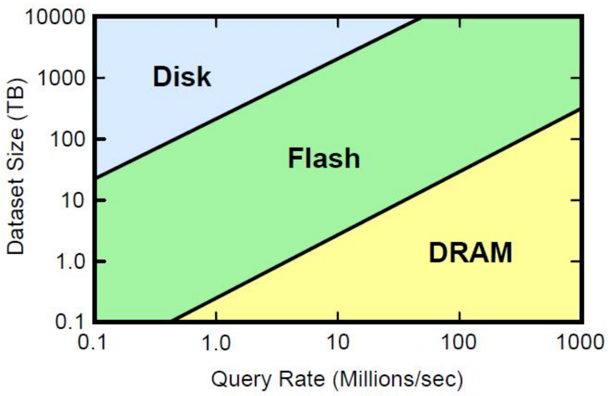
5.2.1. TCO of Memory vs. Disk-Based Solutions

Disk space is growing at a very high rate. However, the speed of access to the disk is not growing at the same rate. The result of this increasing gap – the time it takes to access data stored on disk is getting *longer* rather than shorter. This also suggests that if an application needs to reach data at certain performance level, or on a given number of concurrent servers, cost cannot be measured strictly on the basis of volume, but must also consider the time/speed factor.

In other words, higher storage capacity does not necessarily increase performance, and cost is incurred either in lower performance or in additional investment to meet performance requirements.

Note that even a *1% miss ratio for a DRAM cache can cost a factor of 10x in performance*.

RAM-based storage can be 100-1000x more efficient than disk storage, and 5-10x more efficient then flash disks, as shown in the diagram below.



This figure indicates which storage technology has the lowest TCO over a three-year period, including server costs and energy usage given the required dataset size and query rate for an application (assuming a random access workload).[4]

The combination of Cisco UCS as the memory machine and GigaSpaces XAP as the in-memory middleware enables organizations to maximize the utilization benefits of memory-based devices.

UCS Memory vs. Alternative Memory Solutions

UCS provides a unique extended memory solution that delivers a lower cost per GB for managing data in either small-scale or large-scale deployments, as shown in the table below.

Memory Capacity	Typical System Memory Cost ($US)	Cisco UCS Memory Cost ($US)	Savings ($US)	Savings (Percent)
96	11.880	4800	7080	60
144	17,820	7200	10,620	60
192	Not Available	9600		
384	Not Available	47,520		

SAVINGS WITH CISCO EXTENDED MEMORY TECHNOLOGY
BASED ON PUBLICLY ADVERTISED PRICES OBTAINED AUGUST 2009

[4] Stanford University – [The Case for RAMClouds]

5.2.2. Latency TCO

As the trend towards a real-time world continues, latency becomes an important factor of businesses' bottom line costs. The actual impact can vary between various industries:

"A brokerage can lose up to $4M per 1ms of latency" (Tabb Group)

"An additional 500ms delay resulted in -20% traffic" (Google)

"An additional 100ms in latency resulted in -1% sales" (Amazon)

Cisco UCS Memory Latency Characteristics

Intel Xeon 5500/5600 series processors have a built-in memory controller that supports three memory channels per processor. In a two-socket server, the two processors are interconnected through the Intel QuickPath Interconnect (QPI). Memory latency is characterized by local access (to memory directly attached to the processor) and remote access (to memory connected to the other processor, requiring traversal of the interconnect). With Cisco Extended Memory Technology, latency to local memory is marginally higher than in a system without memory extension, and it is significantly lower than the latency for remote memory access. This feature can result in significant performance improvements based on memory access times in addition to the performance improvements that can be achieved because of a larger memory capacity in the server.

Cisco UCS 10GE Interconnect Latency Characteristics

Cisco UCS 6100 Series Fabric Interconnects comprise a family of line-rate, low-latency, lossless, 10GE interconnect switches that consolidate I/O within the system. Both 20-port one-rack-unit (1RU) and 40-port 2RU versions accommodate expansion modules that provide fiber channel and 10GE connectivity.

GigaSpaces Latency Characteristics

GigaSpaces positively affects latency TCO as follows:

- **Providing in-memory latency to existing applications:** GigaSpaces provides a middleware stack that resides entirely in-memory. All disk access occurs asynchronously outside of the user transaction, providing the latency benefit of memory to existing applications.
- **Reducing network overhead:** GigaSpaces' "secret sauce" is the use of Space Based Architecture (SBA). SBA is a software architecture designed primarily to improve application scaling and latency through better packaging of the application components. With SBA, latency can be reduced as follows:
  - **Reduce latency through optimized deployment and packaging model:** Collocation of the application tiers (logic, data, messaging, presentation) eliminates network calls between these tiers and reduces the associated transactional latency involved.
  - **Reduce latency of remote applications by pushing the application logic to the data:** If the entire application cannot be collocated with the data, GigaSpaces can push the relevant element of the application code into a remote data service. This can reduce the network overhead associated with the iteration between the logic and the remote data. This pattern is also known as Map/Reduce. According to the benchmark tests performed as part of the development of this paper, this optimization can yield up to 10x improvement in latency and throughput.

- **Reduce latency under load (backlog):** One of the major causes of latency under load is backlog. GigaSpaces dynamic scaling enables the application to balance the load among multiple server instances, as well as spawn new instances dynamically, reducing – and in some cases entirely preventing – the impact of backlog on latency.

5.2.3. Scaling TCO (Linear Scaling vs. Non-Linear Scaling Systems)

Scaling TCO is a measurement of the cost of scaling. Most existing disk-based database and application server solutions cannot scale linearly due to their inherit contention on disk I/O.

Cisco UCS and GigaSpaces XAP provide a full linearly scalable system from the application middleware down to the hardware. Cisco UCS provides management scalability across multiple chassis and thousands of virtual machines, and the capability to scale I/O bandwidth to match demand. GigaSpaces enables users to scale their applications linearly by eliminating any disk contention from the application level.

The following table shows how much hardware can be saved with linearly scalable systems vs. non-linearly scalable systems. Savings depend on the level of scalability being attempted, because more scaling means the purchase of more hardware for each incremental increase in throughput. It also depends on the contention – how congested your application architecture is, the severity of the bottlenecks, and as a result – how difficult it is to scale.

Applications scenarios	Event-driven, no H/A	Reliable messaging	DB-centric web apps	Integrations, ESB	
Scaling Your App By / Contention	**10%**	**20%**	**30%**	**40%**	**50%**
Scaling by 1.5X	Save 15%	Save 30%	Save 45%	Save 60%	Save 75%
Scaling by 2X	Save 20%	Save 40%	Save 60%	Save 80%	*
Scaling by 3X	Save 30%	Save 60%	Save 90%	*	*
Scaling by 4X	Save 40%	Save 80%	*	*	*
Scaling by 5X	Save 50%	*	*	*	*
Scaling by 6X	Save 60%	*	*	*	*

These figures were computed using the most broadly-accepted model of software scalability – see our definitions and assumptions.

5.2.4. Reducing the Cost of Downtime

The impact of downtime on cost

25% of companies in a Forrester Research survey estimated the cost of each downtime event $500K-$1M.

According to IDC, the total cost of downtime in 2009 is estimated at $400 billion, an average of $8,000 per server per year.

The main causes of downtime costs are the number of moving parts and human errors.

Cisco UCS and GigaSpaces XAP reduces the downtime cost through:

- **Reducing the number of moving parts**: Cisco UCS provides a fully integrated environment that reduces the number of physical parts, including cabling, as well as exponentially reducing the number of associated points of management. GigaSpaces reduces the number of application instances and tiers through co-location and through a unique shared clustering that uses the same underlying cluster across all the application tiers and API.
- **Reducing manual work:** Cisco UCS comes out of the box with unified, embedded management that provides a central view of all system resources for easily tested and replicated provisioning templates and rapid troubleshooting. GigaSpaces provides a fully self-managed middleware solution through its SLA-driven middleware and management APIs, with most of the manual work associated with deployment, scaling, relocation, resizing, provisioning new machines being fully automated. In addition, the fine-grained monitoring provides visibility through the entire stack that enables the detection of problems and taking proactive corrective measures.

Appendix A: Benchmark Configuration and Environment

This appendix shows how SBA would be implemented for a simple distributed application: a trading system that accepts buy and sell orders from clients. This application is transactional (there are two sequential steps in the business process), and requires very low latency, so that buy and sell orders can be executed in real time.

UCS Model name	Cisco UCS B250 M2 Blade Server
Memory configuration	384GB (48 x dual-rank DDR3-1333 8GB DIMMs)
CPU configuration	2-socket hexa-core Intel x5680 3.33GHz processors (a total of 2 x 6 = 12 physical cores, hyperthreading to 24 cores)
Number of blades/ Chassis	UCS B250 M2 Blade Server inside one UCS 5108 Chassis (that can hold up to a total of 4 UCS B250 M2 blade servers)
OS model	RedHat 5.4 x86_64, 2.6.18 kernel
Java Version	jdk1.6.0_18
GigaSpaces Version	GigaSpaces-XAP-premium-7.1.0-m6-b4287
Object Qty (POJO)	80M
Object Size	1024B
Avg. Test Time	209574 (ms)
Java Threads	24 to 80
HyperThreading	Enabled

U.S. Headquarters
GigaSpaces Technologies Inc.
317 Madison Ave, Suite 823
New York, NY 10017
Tel: +1-646-421-2830
Fax: +1-646-421-2859

California Office
GigaSpaces Technologies Inc.
101 Metro Drive, Suite 350
San Jose, CA 95110
Tel: +1-408-878-6982
Fax: +1-408-878-6149

Europe Office
GigaSpaces Technologies Ltd.
2 Sheraton St.
London, W1F 8BH, United Kingdom
Tel: +44-207-117-0213
Fax: +44-870-383-5135

International Office
GigaSpaces Technologies Ltd.
4 Maskit St., P.O. Box 4063
Herzliya 46140, Israel
Tel: +972-9-952-6751
Fax: +972-9-956-4410