

# Oracle Coherence: Providing Extreme Performance, Predictable Scalability, and Continuous Availability for Mission-Critical Java Applications

*An Oracle White Paper  
Updated August 2007*

# Oracle Coherence: Providing Extreme Performance, Predictable Scalability, and Continuous Availability for Mission-Critical Java Applications

**To optimize the financial impact, you must build your mission-critical applications on an infrastructure that delivers consistent high performance, no matter how massive or erratic the throughput demand. Users demand continuous availability of information with 100 percent reliability, even in the event of hardware or software outages. And the cost of scaling must be predictable and economical, even with explosive user and data growth.**

## INTRODUCTION

Organizations can gain significant advantage by managing and exploiting their information systems more effectively than their competitors. But with today's tight budget constraints and technology limitations, many miss the opportunity—sometimes with disastrous results. This white paper addresses how you can gain a measurable competitive edge without compromise—today.

## TIME IS MONEY

It is not just a cliché. For most mission-critical applications—such as trading systems, risk analysis applications, logistics management, and online order and support systems—time is money. To optimize financial impact, you must build your mission-critical applications on an infrastructure that delivers consistent high performance, no matter how massive or erratic the throughput demand. Users demand continuous availability of information with 100 percent reliability, even in the event of hardware or software outages. And the cost of scaling must be predictable and economical, even with explosive user and data growth.

Faced with such tremendous demands, many institutions accept compromise. But instead, you can cost-effectively address these challenges in Java-based systems through the use of commodity hardware with clustered data-management technologies.

Java, commodity hardware, clustered servers, and grid computing are the key components that organizations apply to today's most-demanding applications. But simply combining these technologies will not deliver value. Applications exist to process data—without data, applications are useless. Applications built without consideration for data access will very quickly become starved for data. A capable data management strategy is a required element for applications to operate with extreme scalable performance and reliability.

## THE TRUTH ABOUT JAVA

**The limitations of Java should not discourage its use for critical applications.**

**Rather, the IT architect targeting exceptional performance must consider Java's strengths and weaknesses during the design phase and plan accordingly. IT organizations need to select vendors that have Java expertise, understand its limitations, and have built their solutions to protect users from any negative impact of the potential pitfalls.**

Java, including Java Platform, Enterprise Edition (Java EE), has become the platform of choice for mission-critical applications for a number of important reasons.

- Java is the premier platform for cutting-edge Web services, clustering, and grid technologies—all vital to today's competitive landscape.
- Substantial standardized infrastructure is available for building, deploying, and managing Java-based applications.
- There is a large pool of professionals available to build and support Java-based systems.
- Almost every new component, library, and application has an API available for Java.
- Java provides an unprecedented number of flexible deployment options—more than any other software platform—including the ability to upgrade applications and services without downtime; the flexibility to implement phased transitions, such as the transition from big iron and UNIX servers to scale-out, commodity hardware; and support on every major server hardware platform and operating system.

All these factors make Java a very attractive platform for institutions looking to gain competitive advantage. However, Java is not without its limitations and potential pitfalls.

- Java's automatic memory management causes garbage collection pauses that can make it impossible to guarantee real-time behavior for complex systems.
- Some of the existing infrastructure and applications in an environment may not seamlessly support interoperability with new Java applications.
- Compared to C++ and CORBA-based applications, many Java technologies are new and may still be evolving.
- Regardless of how well the technologies perform, each new generation of application software is generally more complex than the previous, resulting in slower performance when running on the same modern hardware.

The limitations of java should not discourage its use for critical applications.

Rather, the information technology (IT) architect targeting exceptional performance must consider the strengths and weaknesses of Java during the design phase and plan accordingly. IT organizations need to select vendors that have Java expertise, understand its limitations, and have built their solutions to protect users from any negative impact of the potential pitfalls.

**Although commodity hardware has helped control the cost of information services, it has not directly addressed any of the most significant challenges facing mission-critical applications—such as information integrity, regulatory compliance, and continuous information availability.**

## **THE TRUTH ABOUT COMMODITY HARDWARE**

Commodity hardware is another important component of today's systems. As is the case with Java, it is important to recognize both its strengths and limitations.

- Commodity hardware has dramatically lowered the hardware element in the cost of computing.
- Although x86-based servers have improved dramatically, they still do not provide features that have been standard on UNIX servers for a decade.
- Switching to Linux as a business application platform has turned out to be more expensive than anticipated, and it is still not as mature as many had hoped it would be.

Although commodity hardware has helped control the cost of information services, it has not directly addressed any of the most significant challenges facing mission-critical applications—such as information integrity, regulatory compliance, and continuous information availability. Users must complement commodity hardware with other capabilities that address these issues.

## **CLUSTERING IS “IN”—BUT DOES IT ALWAYS WIN?**

Server clustering has become a popular approach to improving performance and achieving competitive advantage. It seems logical that Java EE performance problems can be solved by adding more application servers to create or expand a clustered server configuration. After all, won't this provide the ability to execute more operations per second?

On the contrary, adding servers often degrades performance. The first time users notice the performance change is often when they scale up from a single-server environment to a cluster. In these cases, databases are made to assume too much of the transient data load, and as a result, performance degrades.

When moving to a cluster environment, you must disable application server caches and optimizations that assume a single server. You also need to introduce extra overhead, such as server synchronization. Within a cluster, a typical approach to supporting failover is to persist application state to the database server. At first glance, this makes sense because the database server is both shared and reliable. But almost every action a user performs changes the application state, and this can quickly and dramatically increase the load on the database server. The load on the application server also increases substantially in order to manage these excessive database reads and writes. In addition, the application state typically includes a large amount of transient data, whose lifetime does not exceed the lifetime of the application. However, databases are optimized for storing transactional and persistent data, not transient data.

**Grid computing has become an important technology for competitive institutions because of its potential to deliver extremely high throughput. To capture the full scalability and availability potential that grid computing offers, you must design your applications for grid environments.**

## **GRID COMPUTING: BUZZWORD CONFUSION**

The confusion surrounding grid computing can result in organizations inadvertently selecting components with inherent limitations that will seriously inhibit their ability to achieve their objectives. For example, many software vendors are quick to use buzzwords such as “failover,” “utility computing,” and “Web services interoperability” in marketing their products. To separate fact from fiction in such claims, it is wise to ask vendors for several production references to make sure that these references have solved business problems similar to your own.

Claiming “virtual” capabilities is another clever marketing ploy. When you hear the word “virtual,” replace it with “not.” Virtual reality is *not* reality. A project that is virtually complete is *not* complete. And virtually seamless is *not* seamless. In the context of grid computing, don’t end up with something that is *virtually* a grid. It is *not* a grid!

Grid computing has become an important technology for competitive institutions because of its potential to deliver extremely high throughput. Therefore, it demands an extra degree of caution to verify vendor claims. Here are a few questions that will help you confirm the viability of any advertised grid computing solution.

- Does your grid computing solution support dynamic allocation and server management?
- What do you mean by “scalability?” (Be certain that the vendor uses “scalability” relative to the throughput of the application and not just the price tag.)
- How does your solution handle infrastructure failures? (At the very least, infrastructure failures must be localized. Preferably, they should also be detected and handled automatically.)

To capture the full scalability and availability potential that grid computing offers, you must design your applications for grid environments. Otherwise, you may miss most of the benefits you could gain—in spite of your best efforts and the grid infrastructure capabilities you selected.

## **JAVA ON COMMODITY HARDWARE—A POTENTIALLY POTENT COMBINATION**

Notwithstanding the reality checks just mentioned, Java and commodity hardware make a potent combination when properly deployed. Commodity hardware has provided phenomenal reductions in the cost of computing. Applications can effectively utilize Java with clustered cache or grid computing technology on commodity hardware and eliminate many of the related concerns, such as hardware, operating systems, and virtual machine reliability.

The challenge is to achieve consistent high performance with predictable, economical, and unlimited scalability while minimizing or eliminating downtime.

With localization of failure—or preferably, transparent server failover—applications can achieve incredible scale without sacrificing availability.

## **THE BOTTLENECK PROBLEM**

**There is no greater infrastructure challenge for most mission-critical applications than solving the data bottleneck. The data bottleneck prevents predictable and cost-effective scalability, and directly impacts the performance and throughput of every data-intensive application.**

After deploying the latest technology, including clustered servers or grid computing, results still may be disappointing. The bottleneck to peak performance is no longer computational throughput—commodity hardware has solved that problem by providing cost-effective horizontal scaling for compute-intensive applications. Now the throughput limitation for most applications is the rate at which they can be fed the data they need to complete their processes.

Applications, the computers on which they are operating, and end users spend most of their time waiting on data. If central processing units (CPUs) have to “wait in line” for data when there are only two or three servers, the wait will only get longer as more servers are added.

In fact, there is no greater infrastructure challenge for most mission-critical applications than solving the data bottleneck. The data bottleneck prevents predictable and cost-effective scalability, and directly impacts the performance and throughput of every data-intensive application.

Remember: time is money for most mission-critical applications. There can be no compromise in solving the data bottleneck problem.

## **THE SOLUTION THAT MAKES THE DIFFERENCE**

To eliminate the data bottleneck, you must distribute data among many resources and perform computations where the data lives. The ability to do precisely this is what makes Oracle Coherence the solution of choice for many of the world’s largest financial, communications, insurance, logistics, travel, and entertainment companies.

Oracle Coherence allows you to store all your transient application data in a high-speed, shared cache with fault-tolerance and failover. Since the data is managed in the application tier rather than the database tier, database load is reduced and the application has much more efficient access to the data. Because peer-to-peer clustering is used, there is no single point of failure or bottleneck.

With Oracle Coherence, Java applications running in a grid do not have to wait in line for data because the compute grid also becomes the data grid. Data resides safely and securely in memory, where it is instantly available when needed. All the delays associated with data retrieval are eliminated.

Oracle Coherence provides a simple, standard API that makes sharing data among hundreds of servers as simple as sharing that data on a single server. And by explicitly providing redundancy, instantly detecting server failure, and automatically managing failover and transparent redistribution of the load, Oracle Coherence eliminates single points of failure and helps applications achieve the highest levels of availability.

**Oracle Coherence is used by many of the world's largest companies to help them achieve extreme, scalable performance with continuous availability for their most important applications.**

## **COMMON THEMES FOR MISSION CRITICAL APPLICATIONS**

Oracle Coherence is used by many of the world's largest companies to help them achieve extreme, scalable performance with continuous availability for their most important applications. These applications often share the same basic requirements and challenges.

- The amount of data involved is voluminous and has been steadily increasing.
- Business and legal requirements have caused the demand for data throughput to increase exponentially.
- Automating these systems has only compounded the increase in data.
- Recent regulatory changes and the increased focus on business continuity and disaster recovery impose additional urgency.

Despite all the new demands, one requirement remains unchanged: the data must be available instantly, and it must be correct. The following examples illustrate how a variety of application types can achieve extreme scalability using Oracle Coherence.

### **Financial Trading Systems**

Financial trading systems require instant access to the current state of the data as well as instant notification of changes to that data. Trading systems highlight the critical necessity for scalable performance in financial services applications. Performance must not degrade as users are added to the system. This is especially imperative for automated trading systems in which windows of opportunity may be measured in milliseconds.

The elimination of single points of failure, the ability to predictably and economically scale trade throughput, and support for rich-client trading systems have made Oracle Coherence the grid infrastructure of choice for new trading systems.

### **Financial Compliance Systems**

Real-time compliance places immense demands on the trade and execution cycle of financial systems, highlighting the need for immediate access to data. Compliance queries may be particularly involved, often requiring multidimensional aggregation. As compliance rules are executed, they must be provable against a particular point-in-time data set. There are no second chances. If the rules cannot be evaluated immediately and predictably, the remainder of the trade and execution chain will be affected. And obvious opportunities will be missed completely.

In addition to eliminating single points of failure, Oracle Coherence enables compliance systems to utilize large sets of aggregation data in memory across a grid and keep the data up to date—all in real time.

**Oracle Coherence can manage the many gigabytes of data and the thousands of updates per second that large feeds entail, as well as the distribution of events and data to internal clients. Because of its self-partitioning architecture, Oracle Coherence can accomplish this without any single points of failure or measurable garbage collection pauses.**

## **Market and Reference Data**

Market and reference data offers huge opportunities for optimizing data infrastructures. This type of data typically encompasses the full gamut of data flows, including long-term-persistence reference data, data on demand, instantaneous feeds, and all the residuals in between.

For short-term reference purposes, the feed data is held as last-known residuals, providing an organizationwide bulletin board. Applications within the organization express interest in all or parts of the bulletin board and expect instant notification of changes to their subjects of interest. Because this data service is capable of delivering consistent response times even as additional capacity is added, new applications can be built on the same data service without impacting existing applications.

As a central data hub and distributor, the service is also responsible for requesting missing data, either on demand or on a schedule. As a centralized system, it must provide scalable performance and cannot be compromised by single point of failure. Downtime would be disastrous.

Oracle Coherence can manage the many gigabytes of data and the thousands of updates per second that large feeds entail, as well as the distribution of events and data to internal clients. Because of its self-partitioning architecture, Oracle Coherence can accomplish this without any single points of failure or measurable garbage collection pauses.

## **Financial Risk Management**

Risk management applications are a perfect fit for grid computing. These applications require a grid solution that can linearly scale its coherent data store as the grid grows. The solution must also ensure that the processing of that data occurs in parallel with each individual piece of processing, gravitating to the grid node responsible for the inputs needed for that processing. This approach achieves linear scale for throughput while nearly eliminating intermediate network traffic. In one Oracle Coherence case study, the calculation process was reduced from more than nine hours to a few minutes.

## **Online Sales and Service**

A broad range of industries has implemented Oracle Coherence clustered cache to support their online sales and service applications. GEICO, one of the nation's largest and fastest-growing auto insurers, is a good example of how Oracle Coherence provides competitive advantage through these applications.

**Using Oracle Coherence, GEICO estimates that in addition to delivering continuous availability and extreme performance, they saved more than US\$700,000 in deferred system and database resource costs. Database load dropped by 80 percent and application server CPU utilization dropped by almost 40 percent.**

GEICO maintains a market position of excellent coverage, low cost, and outstanding customer service. They are focused on providing continuous availability with exceptional service response for their online users. The volume of their Web-based activity is increasing at twice the rate of the company's overall growth. Their Web site now originates nearly 50 percent of sales revenue while supporting half of the customer-support activity.

Their Web site, which is rated the number one auto e-service site in the country by Gomez and received a Webby award as the best insurance industry Web site, has operated without interruption since Oracle Coherence was implemented. Upon its deployment, Oracle Coherence reduced users' page-turn time by an entire second, delivering continuous performance well within GEICO's defined service-level standards. There was only one brief exception.

That exception occurred the day GEICO opened their online service to the state of New Jersey. Even though the introduction was well orchestrated, the demand had been grossly underestimated. All servers were maxed out by 10 a.m. And although the volume was 3.5 times the maximum expected, the system did not crash. Not one sale or transaction was lost. The only impact felt by users was that during the peak load, page-turn time deteriorated to five seconds.

GEICO scrambled to double their Web site's capacity, adding servers wherever they could find them. This expansion was made without any service disruption. Oracle Coherence automatically recognizes when a server has been added to the cluster and redistributes the load across all available capacity. Had a similar volume been imposed on the database tier, the impact would have been disastrous.

Using Oracle Coherence, GEICO estimates that in addition to delivering continuous availability and extreme performance, they saved more than US\$700,000 in deferred system and database resource costs. Database load dropped by 80 percent and application server CPU utilization dropped by almost 40 percent.

## **A FINAL WORD OF CAUTION**

It is important to select the caching methodology appropriate to your application environment. Each cache technology has its own performance characteristics. And applying an inappropriate caching methodology to your applications will not ensure scalability. A fully replicated cache, for example, provides the highest speed of data access up to a point, but does not scale linearly as the cluster grows.

Oracle provides an extensive range of caching services including replicated cache, distributed (partitioned) cache, and near cache. Oracle also offers a wide variety of unique functions—such as write-behind caching, HTTP session management, and distributed query facility—tailored to further enhance scalable performance for specific applications.

The ability to tailor Oracle Coherence's broad range of caching methodologies to your specific application needs is an important element in achieving optimum linear

**The ability to tailor Oracle Coherence's broad range of caching methodologies to your specific application needs is an important element in achieving optimum linear scalability, extreme performance with virtually unlimited capacity, and significant competitive advantage.**

scalability, extreme performance with virtually unlimited capacity, and significant competitive advantage.

## **SUMMARY**

Oracle Coherence is a proven solution that provides measurable competitive advantage. It eliminates data bottlenecks and provides mission-critical Java applications with the infrastructure to achieve predictable, scalable performance and high availability by supporting the broadest range of deployment options, including commodity hardware, clustered server, and grid computing environments.



Oracle Coherence: Providing Extreme Performance, Predictable Scalability, and Continuous Availability for Mission-Critical Java Applications  
August 2007

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[oracle.com](http://oracle.com)

Copyright © 2007, Oracle Corporation and/or its affiliates. All rights reserved.  
This document is provided for information purposes only and the contents hereof are subject to change without notice.  
This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.