ORACLE
FINANCIAL SERVICES

An Oracle White Paper
August 2011

PERFORMANCE, TIME TO MARKET, SUSTAINABILITY

Building Trading Applications using the Oracle Extreme Java Trading Platform

ORACLE

Introduction

The Oracle Extreme Java Trading Platform is an engineered system with a unique combination of technologies that allow developers to quickly build high performance trading applications that are low latency, deterministic, and can be deployed rapidly

Increasingly, the key differentiator between electronic trading market participants is the speed at which they can process market data into opportunities, executed orders, and fills; so the performance of their trading platforms is key to success. Fundamental to both market venues and trading firms is minimizing latency across the entire trade execution process.

Applications performing specific functions – such as market data handling, algorithmic trading, execution management, order routing, and connectivity – are typically created as a number of functional modules which are physically distributed across a series of servers, or run on a single multi-core server. The optimal architecture is determined by analysis of the application itself.

Traditionally, there has been a truism of software development – 'if you build it quickly, it runs slowly', the corollary being that if you want your application to run fast then you had better invest significant time writing it in C++, and in tuning and optimisation. In the world of trading, where getting the algo onto the desk as quickly as possible is vital, firms are searching for technologies which can alleviate this paradox – how to build applications quickly that run fast.

The 80:20 Pareto rule applies inasmuch as it takes a disproportionate amount of time, effort, and specialist knowledge to extract that last little bit of performance – and by extension, cost. Many firms have mitigated this by buying 'innovative' technologies from 'early stage' software and hardware vendors – for innovative and early stage, read risky and difficult to integrate and support. Furthermore, the firms themselves have no choice but to take on the integration risk – the effort of making so many technologies with heterogeneous APIs work together as a single, high performance, unbreakable whole.

Whilst performance is at the top of the list, these firms live according to the principles of Alpha - the relationship between risk and reward. Regardless of the promises of available performance, running what is essentially a technology business on top of risky technology cannot be contemplated. Also, the realities of life in an investment bank are such that cost is always front of mind.

Consequently it is important to understand the realities of how fast the trading application really needs to be. It is certainly true that not all requirements are equal – so the extent to which the trading

application needs to exhibit extreme performance can make a huge difference to the costs and risks associated with the technology platform.

For example, whilst some Options trading and Arbitrage applications may need to give microsecond and millisecond latency to be competitive, most latency sensitive applications can be competitive with performance at around 10 milliseconds. Clearly the kind of technology required to achieve deterministic 1 microsecond performance is significantly different, and more expensive to achieve, than that required for 10ms.

Trading System Requirements

Given the development and challenges of the electronic trading landscape, one can derive a shortlist of requirements for the technology platforms that will address those challenges:

- **High performance, high throughput, and low latency.**
- **Deterministic and consistent performance in terms of latency.**
- **High availability and resilience.**
- **Fast to deploy, easy to maintain and upgrade.**
- **Minimal number of moving parts to test and re-test.**
- **Easy to integrate with existing and proprietary technologies and applications.**
- **Low TCO, including power/cooling/space and management. Small physical footprint for deployment in Co-Lo sites.**

Java Performance

Whilst the latency characteristics and performance of electronic trading applications are paramount, there are trade-offs that architects and developers must keep in mind. There is little point in building the world's fastest algorithm if the opportunity to capitalize on the market innovation has already passed before the algo is finally ready to deploy. Equally, it makes no sense to develop the fastest application if it cannot be easily and quickly adapted to address the next trading opportunity as the market changes. So flexibility and time to market are considerations that must be treated as seriously as raw performance.

It is estimated that more than 80% of electronic trading applications are written in Java. The reason that this phenomenon has arisen is that the Java platform offers the best balance between performance, speed of application development, and ease of flexibility and deployment. Amongst the most significant examples is the fact that nearly all of the FIX engines in the world are written in Java.

It is also true that well written applications in Java should be every bit as performant as those written in C++ or any other language. Controlling the ways in which Garbage Collection impacts latency and determinism is a consideration that the programmer must take direct responsibility for, assisted by the features of the Java Virtual Machine. The dynamic run-time compilation optimisations that the Java VM offers can, in many cases, result in programs that exhibit better performance than those statically compiled by a C++ compiler.

Architecture and Performance

InfiniBand Network Communications

Trading applications are typically architected in a distributed fashion with different modules connected using high-speed network interconnects for exchanging messages. This messaging either takes place on a 'message bus', which is a specialist infrastructure component built for messaging, or using a data grid that does the messaging for you. Either way, distributed trading apps require the conversation and chatter between the modules to be extremely high performance. Essentially, at the heart of any trading application sits a high performance network; a scalable network which enables high-speed communication between constituent applications and components.

Ethernet combined with TCP/IP has long been the standard networking 'stack' within datacenters, but for low latency applications Infiniband (IB) technology is fast becoming the networking fabric of choice. This is because it is extremely fast and absolutely deterministic, which is not true of Ethernet and TCP/IP. Infiniband today offers raw bandwidth rates of 40Gb/sec (QDR IB) and latencies typically in the order of a few microseconds.

A key difference between IB and TCP/IP is that the IB architecture allows applications direct access to the underlying messaging service by bypassing the operating system (OS) kernel and networking stack using a technique commonly referred to as 'kernel bypass'. Message processing in IB is not done by the

CPU and OS but by IB networking software and hardware interface, which are called an HCA. With the OS and CPU freed from having to process network traffic, more compute cycles are available to do application related tasks. Not only is the CPU freed up, but also memory bandwidth, often times a more constrained resource than CPU cycles, that would otherwise have been allocated for processing network IO. In a typical TPC/IP network, each inbound packet is placed in placed in an anonymous OS buffer pool and then copied to the application's virtual memory. A single 10Gb/s Ethernet link can thus consume more than 20 Gb/S of memory bandwidth. IB has a "Zero Copy" architecture and memory bandwidth and message latencies are drastically cut because messages do not need be copied between OS memory buffers and application buffers. It's important to note that although the OS is bypassed, IB architecture ensures that applications have the same degree of isolation and protection that an operating system would otherwise provide, and it even provides for higher networking Quality of Service (QoS) management than a typical OS would provide.

Another difference between TCP/IP and IB technologies is that TCP/IP is a stream-based protocol where as IB is block or message based. While TCP/IP transmits packets of data and notifies the receiving application on receipt of each transmitted package, Infiniband software and hardware does not notify the receiving application until the entire message has been received and assembled. This vastly simplifies the interactions between the application and the networking stack and reduces the number of interrupts that the OS and application must process.

In addition to sending and receiving messages, Infiniband allows applications to directly and securely read from and write to the virtual memory of other applications, even applications running on remote servers. This additional communication semantic is called Remote Direct Memory Access (RDMA) message transport service.

RDMA, asynchronous send and receive, zero buffer copies, kernel bypass, and lossless flow control are some of the key architectural features that make Infiniband the network technology of choice for trading applications that require low latency, deterministic, execution.

In-Memory Databases and Data Grids

For reasons of performance, traditional disk-based databases are not used for low latency applications. Whenever an application needs to read or write data from a hard disk drive, the concept of low-latency is undermined. With microprocessors with large addressable memory spaces and affordable servers with as much as 2TB of DRAM, in-memory architectures are increasingly appropriate to support real-time and low-latency applications, and customers successfully use both In-Memory database and Data Grid technologies. When it comes to real-time use cases, there are technical differences between the two approaches:-

- In-Memory Database (IMDB) is designed for very demanding latency requirements, and can execute many operations in less than 100µs (microseconds). For relational database use cases that require predictably low-latency data access on the order of tens of microseconds, IMDB is an ideal solution.
- Data Grid is a data management architecture for application objects that are shared across multiple servers, require low response time, very high throughput, predictable scalability, continuous

availability, and information reliability. A Data Grid is a system composed of multiple servers that work collaboratively together to manage information and related operations – such as computations – in a distributed environment. A Data Grid achieves low response times and very high performance for data access by keeping the information in-memory and in the application object form, and by sharing that information across multiple servers.

Application objects are the actual components of the application that contain information and are shared across multiple servers. Unlike a relational schema, application objects are often hierarchical in nature, and may contain information pulled from many database tables.

One of the major benefits of developing trading applications using Data Grid is that the low level messaging between components is taken care of by the Data Grid technology on behalf of the developer, rather than it having to be custom built as part of the application using expensive specialist messaging products.

Architecture - Java and Component Choices for Performance

Architectural choices and selection of key component at all levels impact the performance of all trading systems. Some technologies to consider include:

- **Java is the language of choice for electronic trading – providing a platform for rapid development, testing and deployment that underpins fast time to market. More than 80% of electronic trading applications are written in Java, including most FIX engines.**
- **InfiniBand networking, supporting kernel bypass and RDMA, provides low latency and highly deterministic messaging for distributed applications.**
- **In-Memory databases boost performance over traditional storage by several orders of magnitude, with no changes to application logic.**
- **Data Grid middleware provides an efficient mechanism for sharing data objects across components in a distributed application, without requiring knowledge of the complex data management and messaging that supports it.**

The Benefits of an Integrated Stack

Many firms have committed huge resources to building high performance trading infrastructures by taking head-on the mammoth task of integrating the most innovative technologies that they could find in the market. These are complex, long, risky and expensive, custom built solutions. Often this has involved working with niche 3rd party, early stage, and exotic technology components with multiple heterogeneous APIs to achieve integration. Significant optimization effort goes into achieving low latency, performance, and scalability because it is so difficult to instrument all the moving parts and interfaces, and difficult to assess impact on performance of future modifications.

Procurement – Time, Cost, and Risk. It is very expensive to buy different infrastructure components from individual best of breed vendors. Nearly all of the individual component suppliers

are very small, high risk, early stage software houses and hardware manufacturers. Customers ideally prefer to buy mission critical technology from a small number of world-class vendors.

Deployment and Manageability – Imagine buying the software components for a trading system from 10 different vendors and one of them is upgrading a network component; what happens with the other 9 moving parts in terms of re-optimization and testing? There is advantage from buying pre-integrated components from a single vendor!

These complex, expensive, heterogeneous systems have proven very difficult to optimize for low latency and high throughput, and the on-going effort for both engineering and testing makes the cost of ownership extremely high. Also, the number of moving parts makes them hard to maintain and upgrade, resulting in a high risk of unforeseen impact on performance from even the smallest modifications. In fact, the resources required to develop and keep a trading infrastructure competitive using disparate technologies are so substantial that they can represent a barrier to entry into the market that only the tier 1 banks can afford.

So there are many reasons why it would be preferable to be building such complex trading infrastructures on top of a single pre-integrated, pre-optimized stack of technologies – assuming that the whole can consistently deliver performance better than that achievable from an assembly of disparate technologies.

The Oracle technology stack is a pre-integrated suite of technologies which deliver:-

- Best performance at each layer of the stack; out of the box.
- Optimization – Where stack components are sourced from many vendors there is often duplication in functionality amongst components that makes the overall solution heavy and slow, and the interfaces between components suffer from heterogeneous APIs. Through ownership of all the components in the stack, Oracle is able to take a higher level view of how best to architect and optimize entire systems rather than focusing on individual components in isolation, and to design and implement the individual components from the ground up as part of a greater whole. Oracle thoroughly tests and benchmarks the components operating in unison rather than isolation, thus ensuring that they work together optimally as a single whole. Large performance gains can be realized by smart design and standardization of interfaces between components, and by a thorough understanding of where best to locate functionality within the stack.
- Modular construction.
- Protection of existing technology investments.
- Integrated into a comprehensive pre-integrated, pre-optimized, pre-tested solution.
- Faster time to market (reduced integration and testing efforts) – gets trading apps into production more quickly.
- Create a platform for growth that scales with trading volumes.
- Reduced risk and lower TCO – pre-integrated and tested.
- World Class vendor.

Oracle offers the complete stack. Buying the components from Oracle means that you will get world-class technologies that remain at least as performant as any competitive solutions. It means that you can replace individual components without throwing away your existing investments. And by investing in the Oracle stack you can derive the benefits of a pre-integrated and pre-optimized stack; it also means that you will be able to derive the future benefits of Oracle's enormous on-going investment in R&D, and transit easily in the future into technologies and systems which will deliver further huge uplifts in performance and manageability without redeveloping your applications or infrastructure.

Serviceability - The ability to field service the system is extremely important when it comes to providing a fault tolerant system. The ability to perform reliable field upgrades as well as the swapping of failed components is important in maintaining uptime in the overall system. Serviceability is not only evident in the mundane (but important) aspect of how the cable design has manifested itself in Oracle Exalogic, but also the choice for individual hardware components such as redundant InfiniBand switches and storage heads that can be replaced without requiring any downtime.

Consolidation - A primary goal is to provide a platform optimally suited for the consolidation of heterogeneous applications. This implies the ability to support multiple application stacks running on various operating systems.

The Integrated Stack Philosophy

Traditional 'best of breed' approaches no longer offer optimized performance and require significant resource for deployment and management, increasing time to market, operational risk, and TCO. On the other hand, adopting an integrated stack delivers a number of benefits:

- **Optimal performance, since all components are built and configured to work efficiently together.**
- **Fast time to market, since the stack is pre-configured and tested.**
- **Modular design allows for integration with existing systems, thus protecting investment in them.**
- **Low maintenance and high serviceability, meaning high availability and low TCO.**
- **Allows IT function to concentrate on application development to support business goals, instead of dedicating resources to infrastructure deployment.**

Components of the Oracle Extreme Java Trading Platform

Java SE

Low latency - a short time delay between receiving a market update, processing it, and sending an order to the market - is a pre-condition for success, but not sufficient alone. Consistently low latency - deterministic latency – is the challenge. The answer is Oracle Java SE Suite, which includes the converged Oracle / Sun Java Virtual Machine and the JRockit Real Time (JRRT) VM.

JRRT is a foundational platform for Java applications that require consistent and predictable low latency response times. By using the real-time VM and best practices for coding, customers can

develop extremely high performance, deterministic trading applications using Java. JRRT can take any standard Java application and provide microsecond response performance, with a five nines guaranteed maximum industry standard for latency-sensitive environments. In the current environment of electronic trading, milliseconds can mean the difference between profit and loss and with Oracle JRockit Real Time, electronic trading with Java-based systems is a reality.

With proper configuration, the JRRT JVM pauses can be consistently kept under 100ms, and with advanced coding techniques the pauses can be reduced to less than10ms.

TimesTen

Oracle's TimesTen In-Memory Database offering is a full relational database that stores data in DRAM for microsecond-level transactional access. Standard SQL commands are used to store and query data, so that applications written to work with hard disk databases require no code modifications to take advantage of TimesTen's performance - tests have shown read operations to complete in less than 5 microseconds, and write operations in less than 15 microseconds. Thus, throughput is measured in tens of thousands of transactions per second on commodity x86 servers. Consequently, TimesTen is an appropriate solution for boosting performance of trading applications that operate on relational database structures. These might include order routing and management applications, where counterparty information needs to be accessed and updated in real-time.

TimesTen is provided as a shared library, which is linked into applications, to provide a fully in-process runtime environment – i.e. no latency-costly context switches are required to perform database operations. Row-level locking and committed read isolation techniques allow multiple applications to interact with the same in-memory database. Despite the in-memory architecture, TimeTen uses agent processes to log data and transactions to disk for full recovery in case of a server outage. TimesTen can also be configured for high availability and instant failover, via replication. Moreover, TimesTen can either be deployed in a configuration where the entire database is memory resident, or as a cache to a larger disk-based Oracle enterprise database.

According to Matthias Schamun, Senior Expert, Product Manager Xentric Order, Deutsche Börse Systems AG, "with the implementation of Oracle TimesTen In-Memory Database to supplement our Xentric Order routing system, we can consistently offer quick transaction times to our customers. All order messages are processed within a matter of milliseconds using Oracle TimesTen."

Coherence

Complementing the single server, relational, approach of TimesTen is Coherence, the world's leading distributed in-memory object Data Grid. Coherence is designed to store vast quantities of in-memory data, well beyond what can be stored on a single server. By splitting up and spreading the in-memory data cache across multiple servers using a design known as partitioning, Coherence can scale by simply adding servers into the network – providing low-latency access to large datasets. Using Coherence, a read request for a data object is serviced either from the partition on the same server as the requester,

or from another partition on a different server across the network. In either case, performance is significantly faster than retrieving data from hard disk. Once accessed, a data object is held on the requesting server in a local, or 'near' cache, for fastest future retrieval. Also, a variant on the near cache is a 'continuous query' cache, which keeps subsets of data locally, also for fastest retrieval.

Behind the scenes, Coherence manages backup copies of partitions, storing them on different servers for fault tolerance. It also manages storage of data to hard drives, again as a background function that does not impact the performance of real-time applications. By holding data in its cache as objects – user defined data structures – it does not need to conform to the relational model to store that data. Thus it's ideal for time series datasets, or those consisting of complex multi-branch structures which are generally used to represent derivatives and other complex financial products.

By layering Coherence into the application stack, performance is significantly improved by the in-memory data access provided. At the same time, processing power is cut by the parallelization techniques, and by not requiring disk access. This feeds through into savings on hardware footprint, and on power and cooling costs.

In one example at a global investment bank, a derivatives pricing function was accelerated from taking 17 hours, down to 20 minutes. At the same time, the number of servers was cut to 60 from a planned 400. Such performance improvements allowed the trading operation to vastly increase the volume of trades that could be handled.

Because of its support for different cache architectures – partitioned and replicated – augmented by near and continuous query caches, Coherence is very appropriate for geographically dispersed trading environments, where 'follow the sun' trading takes place. Other functions are likely to benefit from distributed caching, such as real-time pre-trade risk management, which needs to access large amounts of data for processing in low-latency time.

Complex Event Processing

Oracle Complex Event Processing (OCEP) is a technology specially designed to build very high performance applications that are processing streams of very fast moving data. This technology offers the best of both worlds - time to market because CEP allows apps to be developed relatively quickly, plus performance and determinism. OCEP is the product of choice because of the close integration with standard Java componentry and the other components in the stack.

A recent OCEP benchmark demonstrated continuous querying on 100,000 messages/second, with processing of just 46 microseconds.

Oracle's CEP engine is implemented as a 100% pure Java application server, engineered for high and deterministic performance. Through standard Spring binding, proprietary Java code – in-house developed or third party - can be integrated with CQL to perform complex queries or execute logic.

Within the CEP engine, each query – including proprietary Java code – runs as a single process thread, with no context switching required. Thus, a trade execution function can run in the same process as the logic that generates the trade, minimizing latency.

A single instance of a CEP engine can execute multiple queries concurrently, and securely, so that several trading strategies can run alongside one another. Moreover, CQL queries running within a CEP engine can be updated and modified in situ, without the need to restart the engine. This feature – know as Dynamic Injection – is particularly useful for trading system implementations, where strategies need to be updated frequently, perhaps even on an intra-day basis.

CEP can be integrated with the TimesTen in-memory database where persistence of data is required with high performance, as is often the case when complex calculations are being performed and intermediary results need to be captured. Calculations also often require access to historical data – such as a time series of trade prices – and for these instances, integration between CEP and Coherence is appropriate, with the latter providing high performance in-memory capabilities for storing such datasets. Coherence integration also provides a design solution for running multiple CEP engines in active/active standby mode for high availability, so that in the event of a server outage, another is ready to immediately take over processing.

The Oracle Extreme Java Trading Platform

Combining Java technologies with other high performance components provides a platform for deploying mission critical trading applications. To recap, the platform comprises:

- **JRockit Java Virtual Machine boosts Java application performance to microsecond processing times.**
- **The TimesTen IMDB provides multiple orders of magnitude performance improvement compared to traditional databases, with no application code changes. This allows relational database functionality to be leveraged for high performance low latency applications.**
- **Coherence offers easy to use, highly performant functionality to distribute data across components in a distributed system. Underlying leverage of InfiniBand network offers very low latency.**
- **Complex Event Processing offers rich functionality for data stream processing, and for encapsulating business logic rapidly. Allows business users and analysts to design, test, and modify trading system functionality via a GUI.**
- **Components integrate to provide optimal performance, out of the box.**

Bringing it all together – Oracle Exalogic

The Oracle Extreme Java Trading Platform is a high performance engine designed specifically for running Java trading applications, with all of the components optimized to run on the Oracle Exalogic platform.

Oracle Exalogic is a complete hardware and software platform for enterprise applications, delivered as pre-assembled building blocks that are easy to buy, deploy, and operate.

Oracle Exalogic is an Engineered System: an assemblage of best-of-breed storage, compute, network, operating system, and software products that are integrated, tested, tuned, optimized, delivered, and supported by Oracle as a single factory-assembled unit

Oracle Exalogic is designed to provide extremely high performance, reliability, ease-of-use, and versatility without being a proprietary, closed system with high total cost of ownership or vendor lock-in. Oracle Exalogic is everything enterprises love about both mainframes and open systems with none of the stuff they don't. Oracle Exalogic is the realization of a new way of looking at the role of IT in the modern enterprise.

Oracle Exalogic has been designed from the ground up to provide the ideal environment for enterprise Java applications and Java-based infrastructure.

Oracle Exalogic is specifically designed to provide financial markets enterprises with a foundation for secure, mission-critical, infrastructure capable of virtually unlimited scale, the highest performance, and management simplicity. Ideal as platform for a wide range of applications, from specialized trading business units to the entire front-to-back office enterprise, Oracle Exalogic is optimized for Java, OCEP, Coherence, and TimesTen, and for the Linux and Solaris operating systems.

Oracle Exalogic hardware is pre-assembled and delivered in standard 19" 42U rack configurations. Each Oracle Exalogic configuration contains a number of hot-swappable compute nodes, a clustered, high-performance disk storage subsystem, and a high-bandwidth InfiniBand interconnect fabric comprising the switches needed to connect every individual component within the configuration, as well as to externally connect additional Oracle Exalogic or Oracle Exadata Database Machine racks.

All Oracle Exalogic configurations are fully redundant at every level and are designed with no single point of failure. Each Oracle Exalogic compute node is a fully self-contained unit of compute capacity in a standard 1U enclosure containing two Intel x86 6-core processors, power supplies, fast ECC DIMM memory, and redundant InfiniBand Host Channel Adapters. Each compute node also contains two SSDs, which host the operating system images used to boot the node and act as high-performance local swap space and storage for diagnostic data generated by the system during fault management procedures.

InfiniBand networking is fundamental to Oracle Exalogic. In addition to providing an extremely fast, high-throughput interconnect between all of the hardware units within a deployment, it also provides extreme scale, application isolation, and elasticity. InfiniBand provides a reliable high performance network interconnect while at the same time offloading compute cycles from host CPUs, further increasing the performance of the overall system.

Up to eight complete Oracle Exalogic racks can be coupled together without the need for interconnects. This expands Oracle Exalogic's power to 2,880 CPU cores and 320TB of hard disk storage. Interconnects allow Oracle Exalogic to scale to hundred of racks.

Oracle Exalogic has been designed from the ground up to provide the ideal environment for enterprise Java applications and Java-based infrastructure – the most popular IT platform for execution venue matching engines and firm's electronic trading suites.

Oracle Exalogic's systems software suite includes a number of optimizations and enhancements made to the core products, including Coherence, TimesTen, OCEP, and JRockit. These include:

- Run-time load balancing - JDBC operations and inter-process communications via SDP.
- Enhanced buffer handling for InfiniBand.
- An optimized multi-core scheduler.

Benchmarks have shown that transactional Java applications perform 60% faster – in terms of operations per second – on Oracle Exalogic, compared to a traditional platform

Database-intensive applications are also boosted by Oracle Exalogic, with a 2x to 3x increase in the number of database operations that can be performed compared to a traditional platform

In addition to extreme performance, Oracle Exalogic's hardware/software integration means that customers minimize the time and effort taken to set up and configure the environment. Oracle Exalogic is tuned for a wide range of workload types – processing intensive, data intensive, I/O intensive, etc. – and thus there is no need to configure parameters for a particular environment. This translates to a reduction in configuration time by as much as 95%. And, because all customers run the same Oracle Exalogic configuration, as tuned by Oracle, errors are reduced and fault diagnosis is simplified. The result is a TCO reduction of up to 60%.

The Secret Sauce

From a performance point of view, the real value for application owners is that the platform comes with systems software that is specifically optimized to take full advantage of the underlying hardware. The 'Exalogic Elastic Cloud' software is an anthology of software, enhancements, and optimizations developed exclusively for and only available on the Oracle Exalogic hardware platform.

Included in the package are a number of unique new features that allow Java applications to take advantage of the underlying InfiniBand networking fabric. For the first time, Java applications can take advantage of low latency Infiniband network features hitherto reserved for C++ applications. With ECS, messaging between Java applications natively utilizes InfiniBand's built-in guaranteed delivery features, zero buffer copy operating mode, and kernel bypass technologies. Socket based communications transparently use the 'native' Socket Direct Protocol (SDP). Oracle Fusion applications such as Coherence take full advantage of RDMA primitives and high-speed messaging APIs, which helps to reduce message latencies from milliseconds to microseconds when running on the Oracle Exalogic platform.

Conclusion

The business outlook for electronic trading calls for extreme performance and agility to leverage opportunities which are short-lived whilst maintaining cost competitiveness within an environment where the only constant is change.

The technology environment supporting electronic trading is a complex one, combining hardware, systems software, applications software, and both local and wide area networking.

No longer is it enough to have the lowest-latency, most deterministic, highest performance, IT platform to underpin one's trading strategies. That IT platform must also be deployable ahead of the competition and be designed in an agile way so that changes and updates can be made on the fly, quickly, securely, and in a cost-efficient manner.

Exalogic delivers Oracle's pre-integrated stack philosophy, combining hardware and software, offers greater performance, faster deployment, and lower cost of ownership. Specifically, a pre-integrated stack provides the following benefits:

- **A higher performance platform, since all components are designed and tuned to work together for optimum latency and throughput, as well as stability and flexibility.**
- **Includes specific functionality to boost the performance of Java applications.**
- **Faster deployment with less deployment risk, because all components have been pre- integrated, configured, and tested.**
- **Ongoing ease of maintenance, since component upgrades are pre-tested for optimum performance within the entire stack.**
- **Ease of management.**
- **Overall, a much lower cost of ownership through ease of deployment, maintenance and management.**

For a vast majority of low trading application developers the tradeoff has been between speed of deployment and speed of execution. The Java platform has been the platform of choice for applications that need to be developed and deployed quickly and cheaply, while C++ has been the language of choice for applications that needed ultra low latency and deterministic execution. With the introduction of the Oracle Extreme Java Trading Platform these developers no longer need to make a tradeoff between efficiency and speed.

Building Trading Applications using the Oracle Extreme Java Trading Platform

August 2011
Author: Ian Pearl

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

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together