An Oracle White Paper
March 2013

Oracle Solaris: The Best Platform for Enterprise Java
Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.
Introduction ............................................................................................. 1
Start with the Right Foundation................................................................. 2
Take Advantage of Leading Performance and Scalability ..................... 3
  3.4x Performance Advantage over IBM pSeries at 3.7x Lower Cost........... 3
  Up to 4x Faster Per Chip than HP ProLiant Servers with Microsoft Windows 2008 R2 .............................................................. 4
  Optimizations to Boost Java Application Performance and Scalability ......................................................................................... 4
  Continued Optimization of the Java Virtual Machine ...................... 6
Enable Mission-Critical Availability for Enterprise Java Applications .. 7
  Oracle Solaris Predictive Self Healing ............................................. 8
  Increased Predictability with Oracle Solaris Image Packaging System (IPS) ................................................................. 8
  Oracle Solaris Cluster ...................................................................... 9
  Enhance Data Integrity and Data Availability with Oracle Solaris ZFS ......................................................................................... 10
Tighten Security ....................................................................................... 11
Simplify Management of Java Services ............................................... 12
  Oracle Solaris DTrace Probes Available in Oracle JRockit Mission Control .................................................................................. 12
  Manage Java Services Within a Cloud Context .............................. 13
Conclusion .............................................................................................. 13
For More Information............................................................................. 14
Introduction

Java, invented by Sun Microsystems – now Oracle, has helped enterprises reduce the cost and complexity of developing and maintaining applications since the mid 1990s. Today, Java not only permeates the Internet, it is the invisible force behind many of the applications and devices that power our day-to-day lives. It is commonly used in mission-critical applications where performance, scale, and ease of management are vital requirements. Oracle Fusion Middleware and Oracle Applications such as Oracle E-Business Suite, PeopleSoft, and JD Edwards EnterpriseOne, are all developed in whole or in part using Java technology.

However, execution of Java applications is very dependent on the features and characteristics of the operating system that supports the Java Virtual Machine (JVM). Oracle Solaris is the most reliable and feature-rich enterprise operating system available on the market today and has been evolving alongside Java in close partnership with the development teams and the Java community. It has many features and capabilities that are specifically designed to make it the best available operating system for Java applications. The most demanding, mission-critical applications today are implemented using a combination of Oracle Fusion Middleware, Oracle Database, and the Oracle Solaris operating system.

Oracle Solaris 11 continues to drive enterprise Java to new heights of security, reliability, diagnosability, developer productivity, efficiency, and raw application performance. For example, recent Java benchmarks detailed in this paper illustrate that Oracle Solaris 11 and Oracle’s SPARC servers offer world record performance for Java.

This white paper describes the key features of Oracle Solaris that make it the best platform for deploying Java applications. The paper is written for an audience of system administrators who want to understand the technologies that make Oracle Solaris the best platform for enterprise Java.
Start with the Right Foundation

This section describes the Oracle Solaris technologies that provide the right foundation for running enterprise Java applications on an Oracle stack comprised of hardware and software technologies that have been engineered and tested to work together.

Oracle Solaris has been designed for enterprise applications and cloud deployments. In addition to the performance optimizations that will be discussed in the next section, Oracle Solaris includes important technologies such as virtualization, security, and high-availability features to enhance Java application service levels and simplify provisioning and management. It's the first fully virtualized OS in the industry, delivering full-service virtualization along with storage and networking virtualization.

Some of the key technologies that support more efficient Java implementations include:

- **Oracle Solaris 11 network virtualization** is a capability of Oracle Solaris Zones, a feature of Oracle Solaris 11. It enables administrators to create a fully virtualized network for more efficient sharing of network resources as well as virtualized network interfaces. Network virtualization gives an administrator the ability to manage bandwidth allocation across virtual network interfaces so that Java services running in Oracle Solaris Zones can still be mapped to service level agreements (SLAs) even with a virtualized network. In addition, the management features enable Java services to be assigned to a specific and dedicated network stack—all the way down to the physical network interfaces.

- **High-availability** features are built into Oracle Solaris 11 to keep applications running. Oracle Solaris predictive self healing automatically detects and recovers from many hardware and application faults. This enables business-critical applications and essential system services to continue uninterrupted in the event of software failures, major hardware component failures, and even misconfigured software. Intelligent zone migration and clustering with Oracle Solaris Cluster provide additional tools for keeping applications running even in the event of a hardware failure.

- **Oracle Solaris Zones** is the cornerstone of Oracle’s server virtualization, providing a low-overhead method for consolidating multiple services onto a single server. Oracle Solaris Zones offers an isolation environment for Oracle Solaris 11, providing secure, flexible, scalable, and lightweight virtualized operating system services that appear as distinct Oracle Solaris instances. Oracle Solaris Zones provides zone-specific observability and scales to hundreds of zones per node.

- **Oracle Solaris DTrace** is a comprehensive monitoring facility that can be used to observe, debug, and tune system behavior. The enhanced visibility offered by DTrace allows administrators to find and fix issues before user service levels are impacted.

- **Oracle Solaris ZFS** provides an integrated file system and volume manager with exceptional data integrity, scalability, and nearly zero administrative overhead. Oracle Solaris ZFS takes advantage of traditional disk storage and solid-state disk storage with a hybrid storage pool approach, providing extremely efficient read and write caches to accelerate accessing and writing data. Volumes can be created on remote storage and shared through iSCSI and FC as a block device. As a result, Oracle
Solaris Zones can be hosted on shared storage in the cloud, providing virtualization of the storage environment along with the virtualized servers.

- **Tools for fast software installation and deployment.** New features in Oracle Solaris 11 simplify and speed the installation and deployment of applications and services. Oracle Solaris Automated Installer cases and facilitates rapid deployment. The Oracle Solaris Image Packaging System simplifies software lifecycle management and thereby minimizes downtime. With Oracle Solaris ZFS, rollback with snapshots provides a built-in safety net while data reduction lowers the amount of storage required. In addition, Oracle Solaris Zones and Oracle VM Templates enable exceptionally fast rollout of new and existing services.

By combining Java applications with Oracle Solaris, administrators and developers can do their jobs with superior out-of-the-box performance, high-quality implementations, and innovative tools designed to simplify code debugging and performance tuning tasks.

**Take Advantage of Leading Performance and Scalability**

Oracle Solaris 11 is the result of more than 25 years of ongoing development and a relentless focus on achieving high service levels for enterprise applications while simplifying overall deployment and management. Oracle Solaris can support tens of thousands of CPU threads and hundreds of terabytes of memory capacity. It offers practically limitless virtualization, and exceptional reliability, availability, and serviceability for future SPARC and x86 servers. The multithreaded capabilities of Oracle Solaris also leverage the performance features of Oracle SPARC T4 and Oracle SPARC T5 servers. Recent benchmarks have demonstrated the performance advantages of using Oracle Solaris with Java applications.

**3.4x Performance Advantage over IBM pSeries at 3.7x Lower Cost**

A recent new world record SPECjEnterprise benchmark result with Oracle Solaris and SPARC T5 servers demonstrated a 3.4x performance advantage over IBM pSeries systems. The SPARC T5-8 server provides 3.4x better performance compared to the 8-chip IBM Power 780 results of 16,646.34 SPECjEnterprise2010 EjOPS. The SPARC T5-8 is also 3.7x less expensive for the application hardware list compared to the IBM Power 780 8-chip configuration. For more information, visit [https://blogs.oracle.com/BestPerf/entry/20130326_sparc_t5_8_specjenterprise2010](https://blogs.oracle.com/BestPerf/entry/20130326_sparc_t5_8_specjenterprise2010).

---

1 SPEC and the benchmark name SPECjEnterprise are registered trademarks of Standard Performance Evaluation Corporation (SPEC). SPARC T5-8 (SPARC T5-8 server base package, 8xSPARC T5 16-core processors, 128x16 GB-1,066 DIMMS, 2x600 GB 10K RPM 2.5” SAS-2 HDD, 4x power cables) list price $268,742. IBM Power 780 (IBM Power 780 9179 Model MHB, 4x3.8 GHz 16-core, 64xone processor activation, 4xCEC enclosure with IBM bezel, I/O backplane and system midplane, 16x 0/32 GB DDR3 memory (4x8 GB) DIMMS-1,066 MHz Power7 CoD memory, 12x activation of 1 GB DDR3 Power7 memory, 5x activation of 100 GB DDR3 Power7 memory, 1x disk/media
Up to 4x Faster Per Chip than HP ProLiant Servers with Microsoft Windows 2008 R2

In a recent SPECjbb2013-MultiJVM benchmark, Oracle Solaris 11 and Oracle Java Development Kit (JDK) were used with Oracle's SPARC T5-2 server to deliver a new two-socket server world record result. The result of 75,658 SPECjbb2013-MultiJVM max-jOPS and 23,334 SPECjbb2013-MultiJVM critical-jOPS beat both HP's four-socket DL560p Gen8 server and HP's two-socket ML350p Gen8 server running Microsoft Windows 2008 R2.2

These results showed the Oracle configuration to be:

- 3.2x faster per chip than the four-chip HP ProLiant DL560p Gen8 SPECjbb2013-MultiJVM max-jOPS results
- 4x faster per chip than the HP ProLiant DL 560p Gen8 SPECjbb2013-MultiJVM critical-jOPS results
- 1.9x faster than the HP ProLiant ML350p Gen 8 on both SPECjbb2013-MultiJVM max-jOPS and SPECjbb2013-MultiJVM critical-jOPS results

For more information, visit

Optimizations to Boost Java Application Performance and Scalability

Administrators can improve their own efficiency and increase application service levels by leveraging the following capabilities from Oracle Solaris for increasing Java application performance:

- **Dynamic threading.** When Java applications are being run on a SPARC T4 or SPARC T5 platform, Oracle Solaris takes advantage of their dynamic threading capabilities and automatically allocates additional CPU resources to the threads that most need them, thus boosting overall application performance. The Oracle Solaris scheduler automatically spreads threads across as many cores and sockets as possible to maximize per-thread performance when the system is less than 100 percent utilized. Oracle Solaris also allows administrators or developers to define a specific thread as “critical.” Threads designated as such will be allocated a full CPU core when running on the SPARC T4 or SPARC T5 platforms. This is a good way for administrators to boost performance of single-threaded applications because the dedicated CPU core enables uninterrupted execution.

backplane. 2x 146.8 GB SAS 15K RPM 2.5” HDD (AIX/Linux only), 4x AC power supply 1,725 w) list price $992,023. Source: TPC.org and IBM.com, collected 03/20/2013. See http://www.spec.org for more information.

• **Large page support.** Enterprise Java applications normally require a large heap and can benefit from storing the heap in large pages of physical memory. If memory is accessed in small pages, additional processing is required because every virtual memory address must be looked up in the translation-lookaside buffer (TLB) to find the corresponding physical memory page. The TLB is a page translation cache that holds the most recently used virtual-to-physical address translations. With large pages, the TLB can effectively reference more physical address space with fewer entries in its cache. Not being able to find an address in the TLB cache can be costly—the processor must read from a hierarchical page table or make multiple memory accesses, impacting performance. A larger page size places less pressure on the TLB and results in more efficient usage of processor memory so that applications spend less time waiting for memory references. Large pages are enabled by default on Oracle Solaris and the default page size is 4 MB. This page size parameter can be easily changed with the command line option `-XX:LargePageSizeInBytes=n`.3

• **Non-uniform memory access (NUMA) architectures and NUMA I/O.** In the Java HotSpot VM, the NUMA-aware allocator takes advantage of NUMA systems running Oracle Solaris, providing automatic memory placement optimization. It can provide up to 15 percent performance gain on NUMA machines such as Oracle’s SPARC T4 and SPARC T5 servers. When NUMA awareness is enabled in the JVM, the eden in the young generation is divided into regions, one region per NUMA node available to the Java process. The eden is where all new allocation happens and with NUMA awareness, the JVM will make sure that each object allocated by a thread will be placed on the local NUMA node where the Java thread is currently executing. This lowers the latency for both allocation and subsequent access to the new object. In addition, "from" and "to" survivor spaces of the young generation, the old generation, and the permanent generation have page interleaving turned on for them. This ensures that all threads have equal access latencies to these spaces on average. NUMA awareness is not enabled by default, but can be enabled with the command line option `-XX:+UseNUMA`.4 In addition, when I/O devices are connected to the system, Oracle Solaris remembers which NUMA node the device is associated with so that when DMA memory is allocated for that device, it is placed on the same NUMA node in which the device resides. This ensures that device I/O operations that utilize DMA do not have to traverse between NUMA nodes because the DMA is always resident on the local NUMA node. This improves I/O throughput for I/O devices by minimizing I/O latency.

• **Oracle Solaris 11.1 virtual memory subsystem.** The new virtual memory subsystem in Oracle Solaris 11.1 automatically scales with the size of the system memory of the underlying platform. It optimizes the assignment of memory resources to make memory allocation much more efficient and predictable on machines that support large page sizes. It includes a built-in memory predictor, which

---

3 Refer to [http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/java.html](http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/java.html) for additional details about standard and non-standard options in the Java application launcher.

4 See [http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/java.html](http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/java.html) for additional details.
monitors large memory page usage and adjusts the size of the memory pages created during collection based on application usage patterns. This performance optimization speeds allocation of large memory pages because there is no need to assemble a large page from many smaller pages when a page is allocated. The large page is often already available from within a pool of large pages. This same feature also helps smaller memory systems because memory page sizes are closely matched to size of the memory pages that the application needs. As a result, enterprise applications and databases are transparently speeded up when migrated to a platform that supports large memory pages.

- **Accelerated cryptography.** The libraries of the Oracle Solaris Cryptographic Framework feature provide a set of cryptographic services and API support whereby both kernel and user-level application consumers can transparently delegate the cryptographic operations to hardware without adding any new code to the application. When Java applications make use of Java cryptographic library calls, these calls are transparently converted to Oracle Solaris Cryptographic Framework libraries, which will then utilize hardware encryption resources on SPARC or x86 platforms where such capabilities are available. This enables wire-speed encryption without any changes to Java applications and without wasting sever CPU resources to implement various packet processing and security algorithms. To learn more about cryptography performance acceleration with Oracle Solaris on SPARC and x86 platforms, visit the related blog sites at:
  - [https://blogs.oracle.com/BestPerf/entry/20130326_sparc_t5_2_zfsencrypt](https://blogs.oracle.com/BestPerf/entry/20130326_sparc_t5_2_zfsencrypt)
  - [https://blogs.oracle.com/DanX/entry/sparc_t4_openssl_engine](https://blogs.oracle.com/DanX/entry/sparc_t4_openssl_engine)
  - [https://blogs.oracle.com/DanX/entry/intel_aes_ni_optimization_on](https://blogs.oracle.com/DanX/entry/intel_aes_ni_optimization_on)

- **Network performance.** Network performance plays a critical role as more Java applications and services go online. An enhanced TCP/IP stack in Oracle Solaris lowers overhead by requiring fewer instructions to process network packets. By optimizing the Internet protocol stack and making it easier to develop drivers supporting new hardware technologies, users have seen speed increases of as much as 50 percent when moving network-based applications to Oracle Solaris. An enhanced software stack not only reduces CPU overhead when processing network packets, it also improves scalability, interrupt handling, and Secure Sockets Layer (SSL) traffic. More network connections can be supported, and throughput can scale linearly with the number of CPUs and interface cards, helping enable the latest 10 GbE adapters to deliver throughput converging on wire speed. The TCP and IP layers in Oracle Solaris are partially merged, resulting in significantly improved network performance for Web applications. The threading and queuing code in the TCP/IP stack is streamlined to process a single packet through both layers, improving processor locality, increasing processor cache performance, and reducing context switch overhead. The result is the network throughput levels that enterprise environments require.

### Continued Optimization of the Java Virtual Machine

Oracle currently offers two Java Virtual Machines (JVMs) that address different types of application requirements. As part of an ongoing effort to optimize the Java environment, Oracle is in the process
of combining the code bases from these two JVMs. The new unified JVM will combine the best features of both JVMs.

The key capabilities in the two current JVM offerings are as follows:

- **Java HotSpot VM** is the primary Java implementation for systems produced by Oracle. It provides performance-enhancing technologies such as just-in-time compilation and adaptive optimization. The name derives from its ability to continuously analyze the Java program's performance for code segments that are frequently or repeatedly executed. These *hot spots* are then targeted for optimization, leading to high-performance execution with a minimum of overhead for less performance-critical code.

- **Oracle JRockit** is a family of Java runtime solutions that features industry-leading real-time infrastructure capabilities and unparalleled JVM diagnostics. Oracle Solaris 11.1 now provides enhanced support for real-time applications including Oracle JRockit. The default resolution of application timers and timeouts is now 1 msec in Oracle Solaris 11.1, down from 10 msec in previous releases. Oracle JRockit Real Time offers deterministic response times for real-time Java applications on the order of milliseconds, making it suitable for the latency requirements of the most demanding financial and telecom applications. Oracle JRockit Mission Control, which comes with Oracle JRockit, is a set of powerful JVM diagnosis and management tools that deliver advanced, unobtrusive application monitoring suitable for use in both development and production environments. Oracle JRockit Mission Control profiles the Java runtime environment with the lowest possible impact on the running system. The software also enables Java applications to run at full speed once the tool is disconnected from the JVM.

Key features from the Oracle JRockit JVM such as Oracle JRockit Flight Recorder, Oracle JRockit Mission Control and real-time, low latency garbage collection are being converged into the Oracle Java HotSpot VM. The convergence is being done iteratively with the initial features already being available in the current Oracle JDK 7 release and more features coming in both JDK 7 updates and the upcoming JDK 8 release.

In the meantime, organizations can continue to use either the Oracle JRockit JVM or the Oracle Java HotSpot VM depending on their requirements. Organizations that already rely on Oracle JRockit Mission Control or have low-latency requirements for real-time Java applications will want to use Oracle JRockit. The Java HotSpot VM will continue to be developed and enhanced to include new capabilities and the Oracle JRockit JVM will continue to be fully supported.

**Enable Mission-Critical Availability for Enterprise Java Applications**

Java applications are often mission-critical and central to daily operations. Deploying Java applications on Oracle Solaris enables organizations to maintain the highest standards for enterprise application service delivery.
Oracle Solaris Predictive Self Healing

Oracle Solaris predictive self healing is designed to maximize the availability of the system and application services by proactively diagnosing, isolating, and recovering from both hardware and software failures. Rather than providing a stream of error messages that can be difficult to decipher, Oracle Solaris predictive self healing automatically initiates appropriate responses such as dynamically taking a CPU, a region of memory, or an I/O device offline—before the component can cause a system failure. If an application should fail, the Oracle Solaris service management facility feature can automatically restart the application.

Another integral part of the Oracle Solaris predictive self healing software, the Oracle Solaris fault manager feature isolates and disables faulty components and helps ensure uninterrupted service—and it does so before administrators even know that a problem exists. In addition, remote service agents can retrieve information that’s vital to diagnosing the root cause of the failure.

Increased Predictability with Oracle Solaris Image Packaging System (IPS)

Environments and resources for enterprise Java applications need to be upgraded and altered to respond to the needs of the users. Historically, patching has been a complex manual process requiring manual patch analysis to understand dependencies and to determine and apply appropriate patches. Administrators sometimes inadvertently introduced problems when the correct patches were not applied or when patch dependencies were not well understood. Testing was therefore required before a new OS environment could be put into production, making the overall upgrade process very time consuming.

The Oracle Solaris Image Packaging System (Figure 1) defines a new approach to package management, offering a smarter and safer means to perform OS software updates. In addition to dramatically reducing the administrative effort to install patches, it also reduces the amount of downtime needed to update a system and virtually eliminates the risk of errors in the patch update process.

In contrast to many other Unix and Linux packaging models, IPS eliminates the need for patching altogether. Relying on the use of software repositories, IPS dramatically changes how an administrator updates system and application software.
The Oracle Solaris Image Packaging System provides increased predictability that system updates are executed correctly because of dependency checks and automatic dependency resolution. This reduces the risk of human error introducing incompatibilities or problems since all updated OS packages are fully integration-tested before download and installation.

In addition, administrators can quickly and easily revert to a previous state because the old boot environment (the current system state) is automatically saved when a new boot environment is created. The clone of the current boot environment is created using Oracle Solaris ZFS snapshots and packaging changes are applied to the clone. By default, this newly cloned environment is activated on reboot. If a problem occurs, the administrator can easily roll back to the previous boot environment image, making the system quickly available in its previous state. In this way, Oracle Solaris 11 provides an administrative safety net for OS upgrades and software changes, helping to improve system and application availability.

Oracle Solaris Cluster

Most enterprise Java applications are so critical that they must continue to operate even in the event of a server failure. Oracle Solaris Cluster provides the capability for these applications to run in a highly available cluster environment that is integrated with Oracle Solaris failure detection technologies and Oracle Real Application Clusters (Oracle RAC) to maximize overall Java application availability.

Oracle Solaris Cluster enables failover to secondary local server in less than a second after a primary server failure. This rapid response makes underlying system failures virtually transparent to Java
application users. The comprehensive failure detection technologies built into Oracle Solaris work with sophisticated policy-based and application-specific recovery templates in Oracle Solaris Cluster to determine the appropriate response to specific types of failure conditions. In addition, Oracle Enterprise Manager Ops Center automates cluster topology discovery, provisioning, cluster-wide patching, and configuration updates to help administrators manage their high-availability cluster deployments.

Enhance Data Integrity and Data Availability with Oracle Solaris ZFS

Many of today’s Java applications create or utilize vast repositories of data that represent valuable intellectual property, making it important that the IT infrastructure provides reliable data access and flawless data integrity.

Basic data protection techniques such as RAID and component or interconnect redundancy are commonly employed, but don’t offer full protection. Oracle Solaris ZFS has several capabilities that help increase data availability and integrity. It combines a copy-on-write approach (data is written to a new block on the media before the pointers to the data are changed and the write is committed) with end-to-end checksumming (explained below) to keep the file system internally consistent. Because the file system is always consistent, time-consuming recovery procedures such as fsck(1) are not required if the system is shut down in an unclean manner, thus improving service levels for users and applications.

Oracle Solaris ZFS constantly reads and checks data to help ensure that it is correct. And, if it detects an error in a mirrored pool, the technology can automatically repair the corrupt data. This relentless vigilance on behalf of availability protects against costly and time-consuming data loss—even previously undetectable silent data corruption. Corrections are facilitated by a RAID-Z implementation that uses parity, striping, and atomic operations to aid in the reconstruction of corrupted data.

As shown in Figure 2, conventional file systems use a block-based checksum whereas Oracle Solaris ZFS performs checksum operations across the entire data tree. In the conventional approach, shown on the left in Figure 2, checksums are stored with the data block, allowing any self-consistent data block to pass the checksum. This approach can only verify media data integrity and can’t even detect stray writes. Oracle Solaris ZFS validates the entire data block tree, thus validating the entire I/O path. Since it validates much more than the media (bit rot), it can catch issues such as phantom writes, driver bugs, and accidental overwrites. With Oracle Solaris ZFS, not only are more corruption cases detected but also they are automatically corrected.
**Tighten Security**

A properly designed IT infrastructure builds in security from the ground up to prevent, detect, and respond to threats. Oracle designs security into the entire operating system, which itself is securely configured by default so that administrators don’t need to spend time configuring in security features. The Oracle software assurance process also includes security checkpoints as part of the quality assurance process. In addition, Oracle Solaris provides granular security features to ensure isolation of users and applications, protect data, and integrate into security event monitoring frameworks. This combination of security features allows organizations to address a wide spectrum of risks with unmatched protection of business interests and intellectual property.

Oracle Solaris offers some of the most advanced security features of any operating system, including the following capabilities that enhance security for running Java on Oracle Solaris:

- **Java Cryptography Extension.** Java applications that utilize Java Cryptography Extension and run on Oracle Solaris automatically take advantage of the Oracle Solaris Cryptographic Framework, which, in turn, leverages the crypto acceleration features in the underlying server hardware. The Oracle Solaris Cryptographic Framework transparently brings the power of advanced, streamlined encryption algorithms and hardware acceleration to Java applications.

- **Application and user isolation.** Oracle Solaris hardens systems against attacks by preventing unauthorized access to data and applications. The software limits and selectively allows applications and users to access only those system resources necessary to perform their function. This capability dramatically reduces the possibility of attack from a poorly written application by eliminating inappropriate access to the system. Even if hackers gain access to an application, they are unable to...
affect the rest of the system, thus limiting the opportunity to inject malicious code or otherwise damage data.

- **Application security policies.** Using the same role-based access control framework discussed above, administrators can enforce strict control over application behavior. For example, a Web server can be given the security policy to bind to only port 70, allowing it to do its job while preventing it from doing more. This feature is new to Oracle Solaris 11.1 and enables application behavior to be modified without modifying the source code.

- **Encryption for data at rest and data in motion.** Oracle Solaris protects data both on disk and in transport, ensuring the confidentiality of data from disk to client. ZFS provides block-level encryption to provide an extra layer of protection against unauthorized access or theft of physical storage. The encryption property is activated at the time an Oracle Solaris ZFS file system is created, and it includes a security check against a passphrase or numeric key when mounting the file system. ZFS encryption can use the Oracle Key Manager for enterprise management across thousands of data sets. Data in motion is protected using the same hardware-assisted cryptography via utilities (SSH, SCP, Krb), cryptographic APIs, and applications that use Java Cryptography Extension for client/server communication (e.g., SSL, IPsec, Kerberos).

**Simplify Management of Java Services**

Managing service levels and maintaining the underlying infrastructure for Java services does not have to be arduous. Having the right tools to do the job efficiently can make a big difference. Oracle technologies such as those described below can simplify management of Java services.

**Oracle Solaris DTrace Probes Available in Oracle JRockit Mission Control**

Oracle JRockit Mission Control provides detailed visibility that enables administrators to monitor the JVM internal state and activities as well as the running Java application. It uses the following instrumentation probes in Oracle Solaris DTrace to provide this visibility:

- **VM Lifecycle Probes**—for VM initialization and shutdown
- **Thread Lifecycle Probes**—for thread start and stop events
- **Class Loading Probes**—for class loading and unloading activity
- **Garbage Collection Probes**—for system-wide garbage and memory pool collection
- **Method Compilation Probes**—for indicating which methods are being compiled and by which compiler
- **Monitor Probes**—for wait and notification events, plus contended monitor entry and exit events
- **Application Probes**—for fine-grained examination of thread execution, method entry/method returns, and object allocation

These probes can be safely used by administrators on live production systems to examine the behavior of both user programs and of the operating system itself. When DTrace is used with Oracle JRockit
Mission Control (Figure 3), it allows users to monitor their Java applications, examine the behavior of user programs and the OS, uncover performance bottlenecks, and troubleshoot runtime issues.

The DTrace plugin for Oracle JRockit Mission Control is available to Oracle Java SE Advanced customers.

Figure 3. Oracle JRockit Mission Control utilizes DTrace probes to provide detailed visibility for Java applications running in production.

Manage Java Services Within a Cloud Context

Oracle Enterprise Manager Cloud Control provides centralized monitoring, administration, and lifecycle management functionality for the complete IT infrastructure, including systems running Oracle and third-party technologies. Oracle Enterprise Manager Cloud Control enables monitoring and management of the complete Oracle IT stack from a single console. In addition, it provides support for business-driven IT management and business-centric top-down application management to manage business services, user experience, and infrastructure. Managing Java applications within this context improves administration efficiency and makes it easier to map resources to business priorities.

Conclusion

Oracle Solaris and Oracle’s integrated hardware and software stack deliver optimized performance and deployment efficiency for enterprise Java applications. Oracle Solaris and Java components are specifically designed and extensively tested to work together to provide the best environment for mission-critical Java applications. The industry benchmarks referenced in this paper illustrate that Oracle Solaris delivers leading Java performance. Oracle Solaris optimizations such as cryptographic processing, NUMA awareness, and large page support provide seamless integration from the Java application all the way to the server hardware, enabling Java applications running on Oracle Solaris to achieve optimized performance without any code changes.
For More Information

Information on the products and technologies discussed in this paper can be found at the following sources.

<table>
<thead>
<tr>
<th>TOPIC</th>
<th>SOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Solaris</td>
<td><a href="http://www.oracle.com/solaris">www.oracle.com/solaris</a></td>
</tr>
<tr>
<td>Oracle Solaris Technical Content on Oracle Technology Network (OTN)</td>
<td><a href="http://www.oracle.com/technetwork/server-storage/solaris11/overview/">www.oracle.com/technetwork/server-storage/solaris11/overview/</a></td>
</tr>
<tr>
<td>Java</td>
<td><a href="http://www.java.com">www.java.com</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://www.jcp.org">www.jcp.org</a></td>
</tr>
<tr>
<td>Oracle JRockit</td>
<td><a href="http://www.oracle.com/technology/products/jrockit">www.oracle.com/technology/products/jrockit</a></td>
</tr>
<tr>
<td>Oracle JDK—Java Platform, Standard Edition (Java SE)</td>
<td><a href="http://www.oracle.com/javase">www.oracle.com/javase</a></td>
</tr>
<tr>
<td>Oracle Services</td>
<td><a href="http://www.oracle.com/services">www.oracle.com/services</a></td>
</tr>
<tr>
<td>Oracle Support</td>
<td><a href="http://www.oracle.com/us/support">www.oracle.com/us/support</a></td>
</tr>
<tr>
<td>Oracle's Sun Servers</td>
<td><a href="http://www.oracle.com/servers">www.oracle.com/servers</a></td>
</tr>
</tbody>
</table>