In a Class By Itself — The Solaris™ 10 Operating System

A Technical White Paper
November 2004

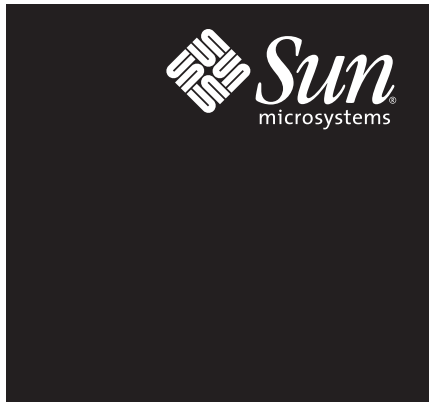


Table of Contents

Chapter 1
Introduction

In today's business climate where dynamic change is the only constant, innovation is the key to success. Businesses are relying more than ever on information technology (IT) services as a way to (1) respond quickly to changing customer needs and (2) build competitive advantage. However, with many organizations having deployed up to a thousand applications in their IT environment, IT managers are challenged to maintain this complex network of systems and applications without exceeding their limited budget and without introducing business risk.

With its customers' concerns in mind, Sun Microsystems has improved its premier operating system to help customers contain costs, reduce complexity, and minimize business risk. Sun's Solaris™ 10 Operating System (OS) is a reflection of Sun's continued commitment to innovation, with many new features and technologies that offer dramatic benefits. The Solaris 10 OS is designed to help organizations optimize system utilization levels, deliver extreme performance, and provide unparalleled security — all with relentless, around-the-clock availability.

- **Reducing Cost**
  The Solaris 10 OS incorporates:
  - New functionality that can help customers reduce costs through better utilization of their Sun systems
  - Higher performance that allows room for growth without upgrading servers
  - More efficient management of the IT environment

  Breakthrough functionality provides the opportunity to:
  - Squeeze up to 80-percent utilization out of each of server
  - Improve application performance by as much as 30 times
  - Automatically configure failing components out of the system
  - Deliver up to 47-percent improvement on Web server workloads through increased network performance
  - Reduce administration costs for file systems by up to 80 percent

- **Reducing Complexity**
  The Solaris 10 OS significantly reduces complexity by delivering a self-managing, self-healing, and self-tuning operating system. Dramatic improvements in manageability give customers the unique ability to consolidate thousands of applications onto a single instance of the OS while at the same time automatically maintaining system, application, and data integrity.

- **Reducing Risk**
  Customers choosing Sun's Solaris 10 OS can minimize risk with industry-leading security, binary and source-code compatibility guarantees, and a multiplatform environment that enables businesses to grow their IT infrastructure using a variety of hardware platforms, including UltraSPARC®, AMD64, and x86 architecture systems.

Breakthrough New Features

Far from being an incremental software release, the Solaris 10 Operating System delivers a breathtaking new set of features designed to dramatically improve performance, availability, and manageability. New key features include:

- **Solaris Containers** — Offering highly-efficient, mainframe-quality partitioning technologies for low-cost systems, Solaris Containers (formerly N1™ Grid Containers) can help consolidate multiple, potentially incompatible applications onto the same server — each securely in its own environment with application-specific resource allocation and an isolated fault zone.
- **DTrace** — Developers and administrators can quickly zero in on performance issues as well as difficult-to-find and difficult-to-re-create bugs using the powerful new Dynamic Tracing (DTrace) facility to quickly understand system behavior. DTrace can be used on running systems to examine application, library, and kernel activity.
- **TCP/IP Stack Improvements** — TCP and IP layers are partially merged in the Solaris 10 OS, resulting in significant performance improvements over the Solaris 9 OS on UltraSPARC processor and x86 architecture-based systems.
- **Predictive Self-Healing** — This innovative new capability automatically diagnoses, isolates, and recovers from many hardware and application faults. Business-critical applications and essential system services can continue uninterrupted in the event of software failures, major hardware component failures, and even software misconfiguration problems. Solaris Fault Manager works in tandem with Solaris Service Manager to form the new Predictive Self-Healing architecture in the Solaris OS.
- **Solaris Cryptographic Framework** — A consistent framework for application-level and kernel-level cryptographic operations, the Solaris Cryptographic Framework can help increase security and performance while giving applications access to the same hardware encryption acceleration devices used by the operating system kernel.
- **Process Rights Management Facilities** — This new security feature significantly reduces the possibility of a compromised application doing real damage to a system by providing fine-grained control over the resources and objects that processes can manipulate.
- **Solaris ZFS File System** — Integrated device and volume management with automatic administration features provides a flexible, secure, scalable, high-performance, policy-driven, and fault-resilient file system.

The unique combination of features found in the Solaris 10 OS allows it to detect and correct faults before they result in application failures, deliver optimized resource utilization, provide higher performance, and protect applications with unparalleled security — all across multiple platform and processor architectures.

In a Class By Itself

Sun designed the Solaris Operating System with a flexible, modular structure so that it can grow and evolve to meet the demands of today's enterprise environments. The exciting new features and functionality available in the Solaris 10 OS were made possible only because the underlying foundation of the Solaris kernel was designed with innovation in mind. Sun continues to make bold progress with the Solaris OS, establishing clear leadership in the areas of network and application performance, server virtualization, file system management, and security.

Scope of this White Paper

This white paper is intended for IT executives and managers. It explores Solaris Operating System design principles and how new Solaris OS features contribute to running business-critical applications with relentless availability, optimized utilization, extreme performance, and unparalleled security. The paper describes a range of features in the Solaris 10 OS, not all of which are available in its first release. Sun reserves the right to alter or eliminate features described in this document at its sole discretion.

Chapter 2

Solaris OS Design Principles

Sun designed the Solaris Operating System for performance and reliability, and to scale and evolve as business needs change. Sun's ability to quickly add support for new processors such as the UltraSPARC IV and AMD Opteron, and maintain feature parity, is testimony to the adaptability of the Solaris platform. The Solaris OS is designed for innovation, resulting in a system that is easily extensible and can grow in an evolutionary fashion without the major, error-prone rewrites that are typically required from one release of software to the next.

This chapter is divided into two primary sections. The first section describes the modular architecture of the Solaris OS and the key technologies that support Sun's continued innovation. The remainder of the chapter discusses the design principles that enable the Solaris OS to provide relentless availability, optimal utilization, extreme performance, and unparalleled security in ways that can help organizations contain costs, reduce complexity, and minimize risk.

Designed to Evolve

The Solaris Operating System is designed in a modular fashion so that it can adapt to new processor and hardware platforms while incorporating new features. It is also designed to provide nondisruptive growth and evolution by enabling new services to be added on top of a stable core. The Solaris kernel is dynamic, composed of a core system that is always resident in memory, with services beyond the core loaded as needed. For customers, this means that Solaris software can evolve to accommodate new devices and services without even rebooting — resulting in less downtime and greater agility to meet changing business needs.

The Solaris OS kernel is a compact code base that is built to be extended. Unlike other operating environments where the delineation between the operating system and applications is confused, Sun designed the Solaris OS so that the line is very clear. It is nearly impossible, for example, for a Web browser to crash the operating system because Solaris software can distinguish between an application and an operating system feature. This clarity enables a smaller code base that results in a more reliable and secure operating environment.

Figure 1 shows an overview of the Solaris OS architecture.

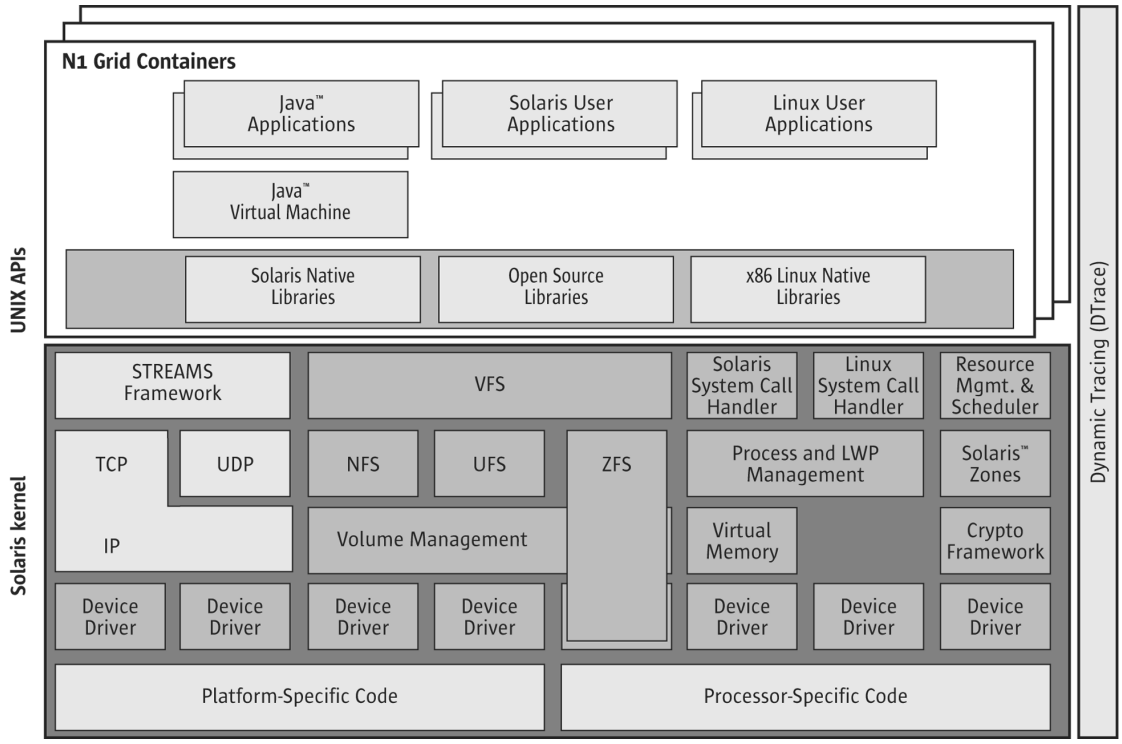


Figure 1. The modular design of the Solaris Operating System helps it rapidly evolve to support new technology and business models.

Some key elements of the Solaris OS architecture include:

- **Processor-Specific and Platform-Specific Code**
  The layer closest to the hardware is platform- and processor-specific code that enables Solaris software to easily support different processors and system architectures. This code comprises less than five percent of the operating system kernel, resulting in an agile, adaptable code base that gives Sun the ability to rapidly bring new SPARC® and x86 architecture systems to market as technologies evolve. It's easy to think that Sun might have difficulty adapting a single common code base to bring out the best in the new processor technologies, but since it has been designed from the beginning to support multiple platforms, the Solaris OS can readily adapt to new technologies, including the new UltraSPARC IV, AMD Opteron, and Intel IA32e processors.
- **Device Drivers**
  Device drivers plug into the Solaris kernel and provide access to I/O devices, including disks, tapes, CD drives, serial ports, and networks. Above these device drivers, software is layered to support device-specific functions. For example, the TCP/IP stack uses networking devices. Volume management and file system software use disk device drivers. In the Solaris OS, new device drivers can be added dynamically, making it easy to reconfigure the kernel without rebooting.

  Device drivers for the Solaris OS are written using published, stable interfaces to the software platform. These interfaces do not change from release to release, so when a vendor delivers a driver for one version of Solaris software, it will continue to work in the next release. Sun is careful to avoid any changes that might undermine the work of its internal and third-party device driver developers.

- **File Systems and Volume Management**
  One very useful way in which drivers are used in the Solaris OS is to manage large numbers of disks as a single volume. Using disk management software like Solaris Volume Manager enables software RAID arrays as well as disk mirrors, striping, and even the capability to add disk volumes to existing logical file systems to be created. Organizations can layer volume management under whatever file system software they wish because the existence of the underlying driver is transparent. Solaris Volume Manager software has been incorporated into the operating system since the Solaris 9 OS release.

  File systems provide another way in which the capabilities of the Solaris OS can be configured dynamically and extended to meet business needs such as managing disk volumes and files. Any third party who writes software that uses the Virtual File System (VFS) interfaces can provide file systems that integrate with the Solaris environment. Examples of the supported extensions include the UNIX® file system (UFS), Network File System (NFS), Sun StorEdge™ SAM-FS file system, Sun StorEdge QFS file system, PC file system (PCFS), ISO 9660 CD-ROM file system known as the High Sierra File System (HSFS), and VERITAS file system (VxFS).

  Sun's new dynamic file system, Solaris ZFS (zettabyte file system), incorporates both volume and device management facilities to help provide a flexible, expandable file system that manages devices and volumes to effect data storage policies defined by administrators. Hence, Solaris ZFS overlays three layers in the Solaris kernel, as illustrated in Figure 1.

  New to the Solaris 10 OS — and available today through the Sun℠ Software Express program for the Solaris OS — is NFS Version 4, including improvements to make it more firewall-friendly, to make it run more efficiently over wide area networks (WANs), and to offer improved security through strong authentication.
- **Solaris Cryptographic Framework**
  The Solaris Cryptographic Framework provides a mechanism and API whereby both kernel- and user-based cryptographic functions can be executed by the same optimized encryption software or transparently use hardware accelerators configured on the system. This new framework brings the power of advanced, streamlined encryption algorithms and hardware acceleration to user-level C and Java™ programming language-based applications.
- **Resource Management and Scheduler**
  Resource management enables fine-grained control over allocation of system resources to processes and tasks. The fair-share scheduler supports task- and project-based scheduling and accounting priorities. Processor sets, implemented as part of the resource management software, also enable specific named processes to be bound to specific hardware processors. System resource usage can be controlled by assigning limits to the rate at which any given task (which can be a collection of processes) can consume resources, including network bandwidth, number of processes, and threads per process.
- **Solaris Containers**
  Solaris Containers software supports multiple isolated environments that appear to processes as their own unique instance of the operating system, when in fact they may be in only one of many Containers configured onto a single instance of the operating system. Solaris Containers support a unique and isolated name space, process hierarchy, and set of resources for each zone.
- **Unified TCP/IP Stack**
  The Solaris 10 OS partially merges the TCP and IP layers in the TCP/IP stack to dramatically improve performance for network applications like Web servers.

- **STREAMS**
  STREAMS provide a network software backplane into which modules can be plugged to dynamically build higher-level services. STREAMS increase flexibility in the use of devices so that new and custom protocols and line disciplines can be configured using the same underlying physical device. They allow various modules to be pushed onto a stack dynamically. An underlying device driver passes data (using standard interfaces) to the next module in the stack, which performs its unique processing and in turn passes data along to the next module and ultimately to a user process. High-level services can be dynamically configured with STREAMS modules. For example, a point-to-point (PPP) protocol stack can be built using a serial device driver and layering in modules to support PPP network control protocols and the TCP/IP protocol itself. No operating system updates or changes to the code are necessary.
- **Linux System Call Handler**
  The Solaris 10 OS kernel includes a handler for Linux system calls for x86 architecture platforms. This in-kernel handler manages system calls issued by Linux applications and dispatches the equivalent Solaris kernel functions to handle the requests. By providing kernel-level support for Linux applications, the Solaris 10 OS can run Linux applications without recompilation on x86 systems — and with maximum efficiency.
- **UNIX APIs**
  On top of the Solaris kernel is a set of UNIX APIs that provide public interfaces to the Solaris kernel. Sun has recently expanded its set of APIs to include x86 Linux Native Libraries to allow Linux applications to run unchanged on the Solaris OS when coupled with a Linux distribution.
- **Java Virtual Machine**
  Sun's Java Virtual Machine is the key technology supporting Web-based services. Java technology enables much of the software that drives the Web today, from Java Servlet and JavaServer Pages™ software to the business logic in application servers. Because the Java runtime environment is so important for most application development, Sun has invested heavily in making its Java Virtual Machine and the Sun Java™ System Application Server Enterprise Edition the best they can be.
- **Dynamic Tracing (DTrace)**
  Overlaying the UNIX APIs and the entire Solaris kernel is a new facility for tracing system behavior. The Dynamic Tracing (DTrace) facility provides an in-depth view of the activities and status information of both kernel- and user-level or application areas. This powerful and easy-to-use tool enables better performance tuning and more rapid resolution of application bugs.

Open Standards-Based Implementation

The Solaris OS adheres to open standards, following Sun's philosophy that healthy competition in an open marketplace fosters superior implementations, higher quality, and lower cost. Indeed, even with innovations that originate at Sun — ranging from NFS to Java technology — Sun's philosophy is to make interfaces public and stable, and successfully compete with other vendors by providing the best implementation rather than locking out rivals by obscure or constantly changing interfaces.

UNIX Platform Open Standards

Portability of applications between UNIX platforms is enhanced by Sun's endorsement and support of open standards. Sun adheres to X/Open®, POSIX, UNIX 98, and POSIX threads standards for interoperability between UNIX implementations from competing vendors. UNIX 2003 certification is in progress.

Interoperability Open Standards

Open standards supporting interoperability between systems is key to today's networked environments. For example, Sun's implementation of the Java 2 Platform, Enterprise Edition (J2EE™) environment is compelling to developers because of the wide range of open standards that the software embraces, including Common Object Request Broker Architecture (CORBA); JDBC™; Remote Method Invocation (RMI); Simple Object Access Protocol (SOAP); Universal Description, Discovery, and Integration (UDDI); eXtensible Markup Language (XML); and XML Remote Procedure Call (RPC) specifications.

Binary Compatibility

Compatibility is one of the hallmarks of the Solaris OS, and is a key feature enabling customers to move up the product line without having to port or recompile their applications, which helps lower costs through investment protection. For years, Sun has supported binary compatibility across its UltraSPARC processor-based product line. Any software, once compiled, runs on every UltraSPARC platform, from entry-level workstations to the 106-processor Sun Fire™ 25K server. This offers customers a high degree of flexibility in deploying applications.

Compatibility Between Releases

Software is one of the most costly investments that an IT organization makes. Operating systems that force frequent application software upgrades due to operating system interface changes can result in skyrocketing costs.

Sun supports a public application binary interface (ABI) that is designed to help properly written applications run on all Sun systems without modification. This implicit contract with developers helps ensure binary compatibility throughout the SPARC product line as well as x86 platforms. It enables Sun to make improvements in the Solaris OS so long as the improvements conform to the ABI. Because the interface is consistent regardless of the underlying hardware platform, the ABI also enables developers to easily move an application from one architecture to another. For example, moving an application from the SPARC platform to the x86 platform requires a simple recompile to generate an executable that will exhibit the same behavior on both platforms.

Sun provides a set of compatibility-testing tools in the Solaris Compliance Assurance Toolkit. This toolkit includes the new Solaris Application Scanner, which quickly assesses whether software that runs on previous versions of the Solaris OS will run on the Solaris 10 OS. The Solaris Appcert tool does a complete job of examining an application's conformance to the Solaris ABI and reports potential release-to-release stability problems.

Designed for Superior Execution

Designing the Solaris Operating System to evolve with changing business needs was only part of the challenge. In order to stand out in a class by itself, the Solaris OS also had to be designed for superior execution. Sun's development efforts for the Solaris 10 OS were therefore prioritized according to how they could impact five major design criteria or design centers:

- Performance
- Security
- Availability
- Utilization
- Platform Choice

Sun's approach to each of these first four design centers is described in the subsections that immediately follow. Platform choice refers to Sun's focus on binary compatibility and industry standards (described previously) as well as Sun's efforts to deliver optimized versions of the Solaris OS for various hardware platforms (Chapter 3, *Platform Choice*).

Extreme Performance

Sun has long held a unique position in the industry by delivering a single operating system that can help organizations *scale up* in capacity through vertical scaling, and *scale out* in capacity through horizontal scaling. Sun believes in empowering its customers to use both scalability dimensions to best meet their critical performance and availability criteria. Sun's Solaris OS supports customers who need to scale up heavyweight applications such as database management systems on servers with more than 100 CPUs. At the same time, for applications such as Web servers where scaling out is the preferred way to grow, the Solaris 10 OS has even better support for volume servers and new processor technologies. Compare Sun's approach to the competition, where some vendors' operating systems don't scale well beyond four processors, and where other vendors require changing operating systems and platform architectures in order to scale up their product lines.

When it comes to performance, the Solaris 10 OS offers breakthrough performance improvements in several areas:

- *Extreme system performance* that unlocks the power of several new processor architectures — including nearly doubling performance on systems with Chip-level Multithreading (CMT) technology like Sun's UltraSPARC IV processor. Sun's internal engineering tests show a scaling factor of 1.85 to 1.98 depending on workload.
- *Extreme network performance* by optimizing TCP/IP stack performance for network-centric workloads with substantial performance improvements for the Solaris 10 OS over the Solaris 9 OS.
- *Extreme application performance* facilitated by the new Solaris DTrace facility that makes it easy for developers to pinpoint performance bottlenecks and zero in on faults in their production environments — without having to create special debug versions of their software.
- *Kernel performance improvements* that reduce overall system call latency through both kernel improvements and special processor-dependent system library optimization — changes that improve all types of applications.

Unparalleled Security

As IT infrastructures have evolved from glass house data centers to distributed networks that are connected to the Internet, new approaches to security have become necessary. The distributed nature of today's data centers, with virtually unlimited points of access, leaves businesses vulnerable to a variety of security threats, including malicious attacks and unintentional acts that compromise security or corrupt data. The only way to successfully protect business interests is to implement a top-down security policy that incorporates the prospect of failure into the security design.

A three-pronged approach is recommended:

- **Prevent** — Prevent malicious attacks and unintentional security breaches through a perimeter security defense system along with security policies and procedures that adequately control users' access, maintain data integrity, and help prevent system administrators from inadvertently breaching security policies.
- **Detect** — Identify an attack in progress by monitoring multiple points on the network including areas behind the firewall, providing a backup to the perimeter security defense.
- **Respond** — Facilitate both manual and automated responses to an attack and enable traceability to the source of an attack.

The Solaris 10 OS adds significant features that can help defend against attacks by preventing unauthorized access to data and applications. For example, Role-Based Access Control (RBAC) provides strict control over the access rights that both users and applications can exercise. Process Rights Management, a feature that was once available only in the Trusted Solaris™ product, further restricts access by preassigning access rights to user processes, significantly limiting the damage that can be done if the process is somehow compromised by an attack. Other features such as Solaris Containers and Solaris Cryptographic Framework provide further levels of security as explained in the following chapter.

By using the right products for their IT infrastructure, businesses can build security into their environments from the ground up to prevent, detect, and respond to security threats. Sun solutions are based on the robust and secure Solaris OS, with its key security features and an open standards approach that allows third-party security solutions to be integrated into the security architecture, and offer unmatched protection of business interests and intellectual property.

Relentless Availability

Downtime, whether planned or unplanned, is something that all IT managers seek to avoid. Sun designed the Solaris OS with a small, compact kernel that limits the exposure to errors that can bring the operating system down, combined with a clear distinction between the kernel, shared libraries, and applications that limits the impact of application failures. Many Sun customers can point to systems running the Solaris OS that have gone months or even years without a restart, a testimonial to the robust nature of the Solaris OS.

Sun doesn't stop with good design. The Solaris 10 OS includes new features that improve it to a point that can only be described as relentless availability. Solaris Fault Manager and Solaris Service Manager build Predictive Self-Healing into the Solaris OS, helping to proactively diagnose, isolate, and recover from both hardware and software failures. Rather than providing a stream of error messages that can be difficult to decipher, Predictive Self-Healing technology is designed to automate the recovery process. It can help reduce and even prevent system failures based on hardware or software faults.

Sun's Solaris Containers can also be used to isolate the impact of application errors by confining one or more services to a secure, fault-isolated zone. Sun also reduces the risk of data loss from hardware, software, and administrator errors with its self-healing Solaris ZFS.

Optimal Utilization

Today's IT managers are looking beyond their previous concerns about scalability, availability, and predictable service levels. They must also find ways to reduce the cost of deploying their IT systems by squeezing extra performance and utilization out of their IT investments.

A common approach to cost reduction is to consolidate business applications onto a smaller number of large servers where economies of scale and improved hardware availability features can provide both cost savings and improved service levels. One of the major advantages of consolidated environments is the ability to utilize shared pools of resources that can provide excess capacity to buffer against temporary spikes in demand. In an unconsolidated IT environment, each application service is deployed with enough excess capacity necessary to handle its own peak loads. When many application services share the same pool of system resources in a consolidated environment, however, they can dynamically borrow resources from each other, reducing the need for extra capacity and improving overall system utilization.

The concept of dynamic sharing of resources has been fundamental to the Solaris OS since the introduction of Dynamic System Domains on the Sun Enterprise™ 10000 server nearly a decade ago. Now, the Solaris OS makes resource sharing even easier with the implementation of Solaris Containers which allow servers and the network to be flexibly partitioned into independent execution environments that can be dynamically resized as workloads change. The following chapter provides more detail about Solaris Containers and their ability to help customers optimize resource utilization.

Chapter 3

Innovative New Technologies

The Solaris 10 Operating System is a major step forward from the Solaris 9 OS because it includes many new technologies as well as major improvements to existing features. All of the improvements are designed to help customers reduce cost, complexity, and risk. The following innovative technologies that set the Solaris 10 OS apart from its competition are described in this chapter:

- Solaris Containers
- Dynamic Tracing facility
- Solaris ZFS file system
- Predictive Self-Healing
- Advanced security features
- An optimized TCP/IP stack
- Technologies that support platform choice

Isolating Applications and Controlling Resources With Solaris Containers

Escalating costs of managing vast networks of servers with software components installed on thousands of nodes have turned the attention of IT executives to cost-reducing measures. They are looking for ways to spend less on IT infrastructure, increase utilization of hardware, and efficiently deliver high end-user service levels. Many customers are pursuing a server consolidation strategy to meet these goals, and Solaris Containers are an ideal approach to implementing this strategy. Sun's solution allows servers and the network to be flexibly partitioned into independent execution environments that can be dynamically resized as workloads change.

Sun's Dynamic System Domains already provide the ability to dynamically partition a single server into multiple independent operating environments, but Solaris Containers take the server virtualization concept to a new level by allowing servers and domains to be partitioned to sub-CPU granularity without replicating the Solaris Operating System image.

A Solaris Container provides a virtualized Solaris OS image, including a unique root file system, a shared set of system executables and libraries, and whatever resources the root administrator assigns to the Container when it is created. Solaris Containers can be booted and shut down just like any instance of the Solaris OS, and rebooted in only seconds if the need arises.

Solaris Containers can provide the following business benefits:

- **Reduced costs through improved utilization** — By allowing applications to borrow resources from other Solaris Containers where resources are not fully utilized, a much more cost-effective implementation can be realized.
- **Reduced complexity** — Solaris Containers can reduce the number of Solaris OS images to be administered and maintained in a consolidated environment by enabling application isolation without giving each application its own instance of the Solaris OS.
- **Reduced risk through isolation** — Multiple applications can be deployed on a single domain while containing the impact of application failures using separate Solaris Containers for each application.
- **Reduced risk of service outages** — Automated dynamic resource sharing helps reduce the risk that service levels will drop below required levels for service-level agreements.

Unlike systems that use high-overhead, virtual machine techniques to implement virtualization, Solaris Containers support mainframe-level partitioning capabilities with almost zero overhead. Sun's approach is unique in the industry in its ability to accomplish sub-CPU partitioning on systems as small as those with only a single CPU. Finally, with feature parity across the platforms supported by the Solaris 10 OS, customers can deploy Solaris Containers on systems using UltraSPARC or x86 architecture processors, including the AMD Opteron family.

Solaris Containers fulfill Sun's vision of bringing high-end system management and partitioning features to even the smallest, single-processor systems. They give administrators the ability to consolidate many applications onto a single instance of the Solaris OS while providing each with independent, virtualized name spaces and resources. This is accomplished by partitioning systems in two key dimensions: *Resources and security*. Solaris Containers are created by combining two complementary, but independent technologies: Solaris Zones and Solaris Resource Manager software.

Solaris Zones

There are three families of technologies for partitioning servers in use today:

- *Hardware partitioning* like Dynamic System Domains, available in high-end Sun Fire servers and competing technologies like IBM LPARs and HP vPars
- *Virtual machines* like those provided by VMware and IBM's VM operating system
- *Operating system-level partitioning*, including Solaris Zones, FreeBSD Jails, and Linux-VServers

Compared to hardware-level partitioning technologies, Solaris Zones software brings software partitioning capabilities to servers of all sizes, even to x86 architecture systems. Unlike IBM LPARS, Solaris Zones partitioning technology enables administrators to reconfigure environments without bringing any one down. And unlike virtual machine mechanisms on Linux systems or IBM mainframes, the performance overhead is nearly zero.

Solaris Zones partitioning technology provides a set of up to 8192 virtualized environments per Solaris OS instance that appear to users, administrators, and applications as independent, isolated systems (Figure 2). A global administrator can create zones, allocate resources to them, and then boot them just like they were an operating system. Once booted, Solaris Zones software provides a secure "sandbox" that includes:

- A virtual platform containing a unique root, shared user, and administrator-configured file systems — plus network interfaces, IPC objects, a console, devices, and resource management facilities.
- Standard system identity settings including host name, time zone, RPC domain, and locale.
- An independent name space including users, roles, and process IDs.
- Secure isolation from other zones enforced at the kernel level. A process in a Solaris Zone, even if compromised, cannot escalate privileges to compromise the system or another zone.
- Fault isolation that can restrict the propagation of software faults to a single zone. Unlike IBM LPARs and HP vPars, if an error does cause a zone to fail, it can reboot in only a few seconds.

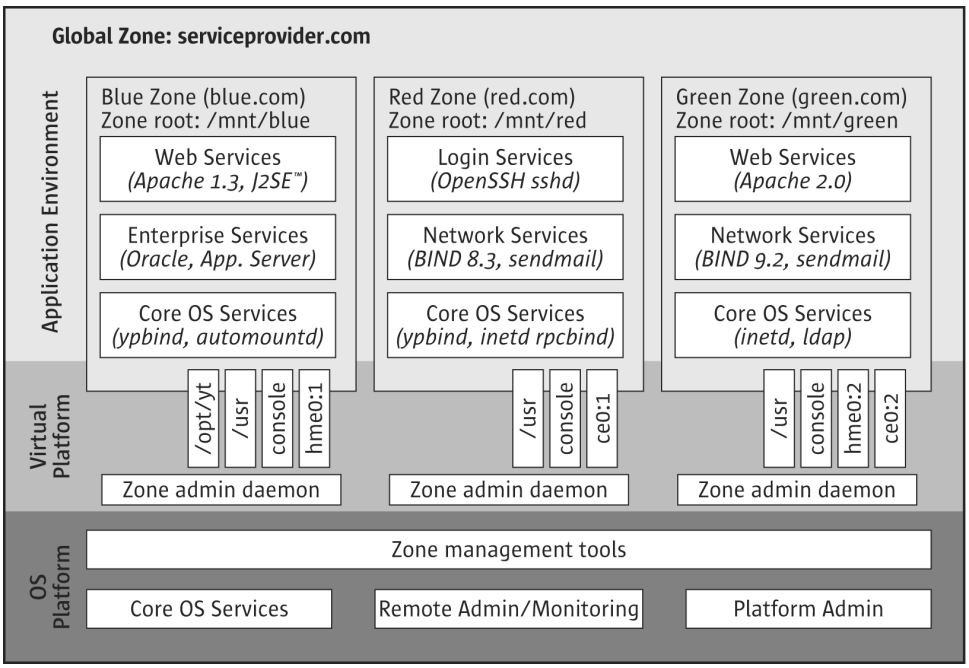


Figure 2. Three zones running on a single Solaris OS instance, each running their own software and each with their own dedicated subset of system resources and system software.

For example, a Solaris Container that hosts a Web server might have an IP address that it can bind to port 80, Web server software, and a disk device containing a file system with the Web site's content. Because Solaris Zones are used to isolate the Container, the Web server cannot see any resources not assigned to it. If the application or even system software within the Container fails, it cannot affect other applications running in other Containers. Similarly, if someone exploits a security vulnerability in the Web server, the intruder can only manipulate resources within the Container, helping prevent failures in other applications running on the same server.

When to Deploy Zones

Solaris Zones software makes it easy to use a single server for multiple purposes in service provider and corporate data center environments. Organizations can host multiple competing applications on the same platform, and multiple organizations can use a single server, each with their own security domain. For example:

- Hosting facilities can give each customer their own virtualized environment, including IP address, disk storage, Web server, applications, and even a root password.
- Data centers can consolidate multiple database instances onto the same server, with each database administrator having complete control of their virtualized environment. Competing applications — such as online transaction processing and data warehousing — can run with resource allocation changing as business needs change.
- Developers can create and test software that requires root access to install or run; different installation scenarios can be exercised, each in their own zone.

- IT organizations can run a production application in one zone and stage a new version in a second zone. Each application has its own environment and isolated set of configuration files. When it's time to make the staged version live, all it takes is changing network addresses and resource allocations.
- Web services environments can consolidate vertically scaled application servers and database servers onto the same multiprocessor server, cutting in half the number of servers required on the back end.

Solaris Resource Management Facilities

Whereas Solaris Zones software provides secure, virtualized environments for application execution, Solaris Resource Manager software provides administrators with almost unlimited flexibility to assign and isolate resources to particular zones, and in many cases can be used to allocate resources within a single zone.

Solaris Resource Manager software gives administrators fine-grained control over the resources that applications use, allowing multiple applications to operate on a single server while maintaining specified quality-of-service (QoS) levels. Fixed resources such as processors and memory can be partitioned into pools on multiprocessor systems, with different pools shared by different projects (a specified collection of processes) and zones.

Solaris Resource Manager supports dynamic resource sharing through its Fair Share Scheduler, allowing different projects to be assigned different scheduling classes. Solaris IP QoS (IPQoS) can be used to manage network bandwidth used by multiple, competing network applications. When resources like CPUs and memory are dynamically allocated, resource capping controls can be used to set limits on the amount of resources used by a project. With Solaris resource management capabilities, organizations can consolidate many applications onto one server, helping to reduce operational and administrative costs while increasing availability.

Combining Virtualization With Resource Management

The combination of virtualization and resource management embodied in Solaris Containers helps organizations improve availability levels by protecting applications from error propagation and security intrusions. They can reduce costs by making server consolidation flexible, simple, and secure. They help protect existing investments by providing the same facilities on existing servers and new Sun UltraSPARC, x86, and AMD Opteron systems. And, Solaris Containers can help organizations optimize their resource utilization by allowing multiple Containers to dynamically and automatically share resources while maintaining agreed-upon quality-of-service levels.

Accelerating Application Performance With Dynamic Tracing (DTrace)

When systems exhibit nonfatal errors or subpar performance, the sheer complexity of the distributed software environment can make accurate diagnosis of the root cause extremely difficult. Transient failures such as not being able to support an anticipated user load or consistently not meeting service-level agreements during peak hours of the day can benefit from deep visibility into system behavior to help diagnose the problem. Traditional approaches for debugging transient failures and tuning performance have involved examining postmortem crash dump files and using trial and error approaches to tuning. Not only are these approaches slow and time-consuming, but they may never completely resolve the problem if the root cause is not found. As a result, many applications languish at performance levels nowhere near optimal levels.

The new Solaris Dynamic Tracing (DTrace) facility is a powerful tool that can help developers quickly tune their applications for maximum performance and rapidly identify the root cause of system and application problems. Tasks that might take days or weeks with traditional approaches can often be accomplished in hours or minutes. DTrace is simple enough to be used by both entry-level and experienced developers, and offers substantial benefits that can help businesses reduce cost, complexity, and risk:

- **Cost** — Resolving system or application performance bottlenecks can be reduced from days to hours, saving on labor costs. DTrace also enables businesses to save on hardware costs in two ways. Improving application performance provides more room for growth without the expense of upgrading hardware. And since DTrace can be safely used on production systems, it virtually eliminates the need to deploy a separate test environment.
- **Complexity** — DTrace's single view of the software stack greatly simplifies the tracing process, enabling developers to follow a thread as it crosses between kernel space and user land and back.
- **Risk** — DTrace works with applications as is; there is no need to modify applications, install a debug utility, reboot the OS, or restart applications before, during, or after the DTrace session. Developers can even use DTrace to analyze or tune applications on the Solaris 10 Operating System and then deploy or redeploy those applications on an earlier version of the Solaris OS while retaining most of the benefits of the performance-tuning exercise.

The primary value for those developing business applications is in delivering higher-quality solutions that offer greater performance and stability. For IT service delivery organizations, it means greater utilization of system resources by making better use of the CPU cycles available.

Expediting Problem Analysis and Performance Tuning Through Enhanced Visibility

The Solaris DTrace facility provides dynamic instrumentation and tracing for both application and kernel activities — even those running in a Java Virtual Machine. It enables developers to explore the entire system to understand how it works, track down performance problems across many layers of software, or locate the cause of aberrant behavior. It even allows creation of custom scripts to dynamically instrument the system and provide immediate, concise answers to arbitrary questions formulated using the DTrace D programming language.

Tracing is accomplished by dynamically modifying the operating system kernel and user processes to record additional data at locations of interest, called probes. A probe is a location or activity to which DTrace can bind a request to perform a set of actions, like recording a stack trace, a timestamp, or the argument to a function. Probes are like programmable sensors scattered in key places throughout the Solaris OS. To explore an area in question, users can turn on the appropriate sensors and program them to record information of interest. Then, as each probe fires, DTrace will gather the data from the probes and report it. If no action is specified for a probe, DTrace simply tracks each time that the probe fires.

DTrace probes come from a set of kernel modules called providers, each of which knows how to perform a particular type of instrumentation to activate probes. When DTrace is run, it invokes a compiler for its D language to look for probes that have been requested by the user and to gather data from providers about the instrumentation that is needed to activate the probes. Providers maintain the system-level information about probes, allowing users to request actions to be taken when the probe fires and leaving the instrumentation of that request to be performed dynamically by the DTrace utility. For example, the user may request that DTrace publish values for specific variables whenever a probe fires and DTrace would execute the required actions to collect that data whenever the probe fires.

Developers can use DTrace D programs to bind their own customized tracing actions to any of the more than 30,000 published probes within the Solaris kernel and to instrument any line of code in an application that runs on the Solaris OS. Figure 3 shows the different components of the DTrace facility, including providers, probes, DTrace kernel software, and the *dtrace* command.

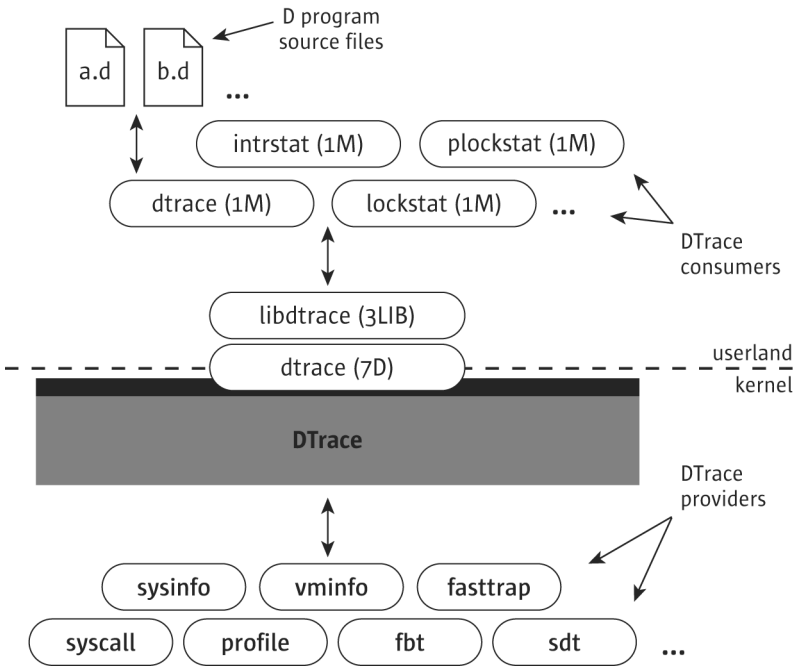


Figure 3. The DTrace facility is architected to enable visibility into both user land and the Solaris kernel.

Reducing Risk

Stability and low-performance overhead are hallmarks of this new utility because DTrace was designed from the beginning to run on production systems. Risk is reduced because users can dynamically turn probes on and off with no need to reboot or otherwise configure the operating system, disable or alter applications, or change user or client access. DTrace is also programmable, so analysis routines can be written and reused.

Safety is enhanced because the DTrace execution environment performs its own error handling and uses proven probes that already exist inside the Solaris OS. D program runtime errors such as dividing by zero or referencing invalid memory are managed directly by DTrace. When such an error occurs, DTrace simply reports the error and disables the instrumentation, allowing the developer to correct the mistake and try again. As a result, developers can never construct an unsafe program that would cause DTrace to inadvertently damage the Solaris kernel or one of the user processes running on the system. These safety features allow DTrace to be used in a production environment without fear of crashing or corrupting the system.

Virtually Eliminating the Performance Penalty for Tracing

Although DTrace is always available and ready for use, it has no impact on system performance when not being used. All of the instrumentation in DTrace is completely dynamic. Probes are enabled discretely only when they are specifically called out by the user. No instrumented code is present for inactive probes, so there is no performance degradation of any kind when DTrace is not in use. Once the DTrace command exits, all of the probes that were used are automatically disabled and their instrumentation is removed, returning the system to its exact original state.

DTrace instrumentation is also designed to be as efficient as possible. When DTrace is executed, the instrumentation for each probe is performed dynamically on the live running system. The system is not paused in any way and instrumentation code is added only for the probes that are enabled. As a result, the probe effect of using DTrace is limited to exactly what DTrace is asked to do; no extraneous data is traced.

Success With DTrace

DTrace has helped Sun improve the performance of both kernel functions in the Solaris OS and customer business applications. For example, for one customer's business-critical trading application, DTrace was run on a live system and the team was quickly able to pinpoint a bottleneck presented by the customer's use of a nonscalable memory allocator. Replacing the component with a more scalable version resulted in a 1000-percent performance increase. Improvements of this scale not only dramatically reduce overall system cost, but they can also have a direct impact on the number of transactions processed each day, adding to top-line revenues.

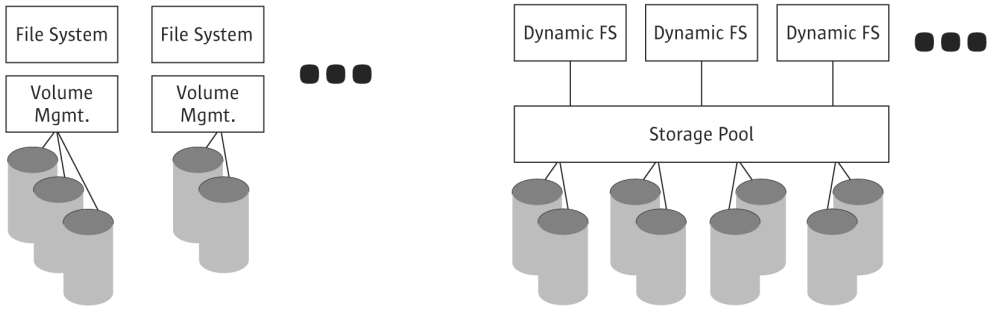
A New Approach to Storage Management With the Solaris ZFS File System

Traditional file systems are made up of data structures overlaid onto a disk volume that is managed independently by a separate volume manager. This approach involves configuring storage devices, virtualizing multiple devices into logical volumes, and finally laying out a file system structure onto them. Each layer must be managed consistently with every other layer, making storage difficult to manage and configurations prone to error. Even when everything goes right, the disk space can be underutilized because each file system must be configured from the beginning to handle an anticipated maximum amount of data.

Sun's Solaris ZFS fundamentally changes the storage equation by integrating devices, storage, and file system structures into a single structure (Figure 4). By integrating the file system with volume management, the risk of a misconfiguration at one layer affecting another layer is virtually eliminated. Sun's file services technology results in more effective use of storage and provides a high degree of reliability and flexibility that can help reduce cost, complexity, and risk.

The primary business benefits of Solaris ZFS include:

- *Improved availability* through online administration and fault resilience features
- *Greater efficiency of administration* through automation features such as dynamic load balancing, logging, and flexible volume management, which can help simplify overall file system management and reduce installation and setup time
- *Improved storage utilization* due to a data compression algorithm and a feature that manages distributed hot space, reducing the need for reserve capacity
- *Investment protection* through adherence to existing standards (POSIX APIs), enabling existing software to run unchanged
- *Greater flexibility* due to massive capacity (up to 2^{128} bytes) and elimination of the requirement for a one-to-one mapping between file systems and volumes
- *Reduced risk of data loss* through data integrity, encryption, and enhanced access control features



Traditional storage management approaches require a separate software stack for each file system

ZFS manages a single large pool of storage, optimizing utilization by allocating storage to file systems on demand

Figure 4. Solaris ZFS manages storage in one large pool, integrating device and volume management.

Key advantages of Sun's new Solaris ZFS as compared to other offerings include:

- **Massive Capacity** — 128-bit compared to most 32-bit file systems
- **Advanced Data Protection** — Checksums and always consistent data structures
- **Always Online** — No need to take storage offline for administration tasks
- **Fast Operation** — Less that 1 second to create a new file system
- **Reduced Administration Costs** — Far simpler administrative model

Flexibility by Design

Solaris ZFS is built to allow file systems to be created on an as-needed basis, so administrators can create a file system per user or a file system per application, as needs dictate. It breaks the bond of a single file system per volume, allowing as many file systems as desired to be allocated from the single pool of storage. This allows more efficient use of storage because different file systems can draw storage from the pool without each one having its own internal fragmentation. This is the perfect complement to Solaris Containers, making it easy for organizations to deploy many Containers with as many file systems as needed.

Solaris ZFS also implements POSIX-compliant file system semantics and stores its structure in an endian-neutral format (where byte order is independent of the CPU architecture). The result is that file systems created with Solaris ZFS on a big-endian UltraSPARC system can later be mounted on a little-endian x86 architecture server, and vice versa. In addition to its standards-compliant behavior, it also supports per-file system attributes. Compression, for example, can be enabled as a file system attribute, resulting in all data stored to the file system being compressed. Solaris ZFS is also written to utilize the Solaris Cryptographic Framework, so future versions may include encryption as a file system attribute.

Dynamic Data Management

Solaris ZFS can dynamically grow up to 2^{128} bytes, automatically configuring the pool as storage devices are added, removed, or reconfigured. Solaris ZFS can also dynamically provision itself according to policies. This means that administrators can add more storage to an existing file system without unmounting, locking, or otherwise interrupting the service that it provides. Removing storage is just as easy. Solaris ZFS enables administrators to state the intent of their storage policies rather than describe all of the details needed to implement them.

In a consolidated environment where multiple applications are running on a single server, there is also a need to more efficiently manage storage resources to deliver high performance, maintain isolation, and avoid fragmentation. Everyone knows the trade-offs: The more volumes needed to support different isolated environments, the higher the cost in terms of fragmentation and underutilized disk space. Solaris ZFS helps administrators partition storage on a fine-grained basis, flexibly using a single pool of storage for multiple file systems and securely allocating them to users, applications, or containers. This allows any single use to grow or shrink, while optimizing overall utilization.

Data Integrity and Availability

Unlike most file systems in use today, Solaris ZFS avoids data corruption due to system failures or power loss by keeping the data on the disk self-consistent at all times. Solaris ZFS manages data using *transaction groups*, which use copy-on-write semantics that always write data to a new block on disk before changing pointers that commit the write. These low-level semantics support file system snapshots at a fundamental level, making it easy for administrators to take consistent backups or roll back all data to a known point in time. They also support future plans for Solaris ZFS, including integrating file system semantics with application semantics so that, for example, a commit operation on a database transaction could eventually be tied to the transaction taking place in the file system itself.

Most file systems implicitly trust the data they read in from disk, which can result in everything from corrupted application data to a system panic. Solaris ZFS incorporates checksums into the file system structure itself. With every block protected, Solaris ZFS can validate every bit of data read in from disk, detecting any data corruption. Because Solaris ZFS is integrated with volume management, it can actually repair faults when they are detected. In the case of a file system configured with disk mirroring, for example, Solaris ZFS can obtain an intact copy of the corrupt block and remap and rewrite the block on the original disk, delivering pristine data to applications. The copy-on-write semantics in Solaris ZFS also help protect RAID 5 volumes from inconsistencies that could otherwise be caused by writing data before parity.

Reducing Hardware and Software Failures With Predictive Self-Healing

When things go wrong on traditional systems, administrators try to make sense of the problem by looking at error messages as well as system and application log files. Unfortunately, these preprogrammed error messages tend to show only the symptoms and do not necessarily reveal the underlying problem. Administrators interacting with a stream of symptoms can find them hard to understand and even more difficult to interpret.

Sun believes that the system — not an army of administrators and engineers — should be responsible for diagnosing and even repairing faults as much as possible. Rather than just providing clues as to what might be wrong, a system should inform administrators on what action to take — and the system may recommend taking action long before there is any visible symptom in terms of actual application failures.

The Solaris 10 Operating System delivers a breakthrough approach to service availability with automatic online error detection and auto recovery for all systems. It proactively removes failing components from the system before failure and automatically restarts failed applications and services. The business benefits include:

- **Lower TCO Through Reduced Downtime** — Automated detection and diagnosis of system component failures allows problems to be diagnosed in milliseconds rather than manually in hours, reducing both administrative costs as well as the cost and lost revenues due to application service failures.
- **Reduced Risk** — Greater availability achieved through proactive fault management translates into reduced risk of business losses such as dissatisfied customers or missed opportunities for revenue.
- **Reduced Complexity** — Automatic management of failing system components and applications on all systems running the Solaris OS can greatly simplify the task of enterprise system management.

The Solaris OS has long incorporated self-healing measures. For example, the kernel memory scrubber constantly scans physical memory, correcting any single-bit errors that it finds so that it can reduce the chance of those problems turning into uncorrectable double-bit errors. The Solaris 10 OS takes a big leap forward in self-healing with the introduction of Solaris Fault Manager and Solaris Service Manager. Businesses can also further extend the reach of Solaris 10 OS self-healing capabilities by adding remote management and support through Sun™ Remote Services (SRS) Net Connect.

Automating Detection and Diagnosis With Solaris Fault Manager

Solaris Fault Manager, a common system that works across all platforms running the Solaris OS, reduces complexity by automatically diagnosing faults in the system and initiating self-healing actions to help prevent service interruptions. It helps increase availability by configuring problem components out of a system before they cause a failure, and in the event of a failure, it can automatically recover and restart applications. The higher availability that can be achieved through these features also translates into decreased costs, including reduced loss of business due to down services, improved resource utilization, and reduced administration costs.

Solaris Fault Manager provides a pluggable, dynamically updated fault management architecture that can be updated with increasingly sophisticated fault management modules as they become available — and without having to reboot the system. It accepts a stream of binary telemetry data regarding the system's health (Figure 5). Data is fed to a set of dynamically loadable diagnosis engines that examine incoming events that are related to their particular function, for example, memory or CPU faults. The diagnosis engine performs synchronous actions such as logging a detected error. It asynchronously produces a fault diagnosis once it has observed discernible patterns from a stream of incoming errors. A specific fault diagnosis is determined using root-cause analysis and by making inferences on a series of incoming events. Much like a good detective solves a crime by piecing together a sequence of events, a fault is detected by observing a sequence of specific error events.

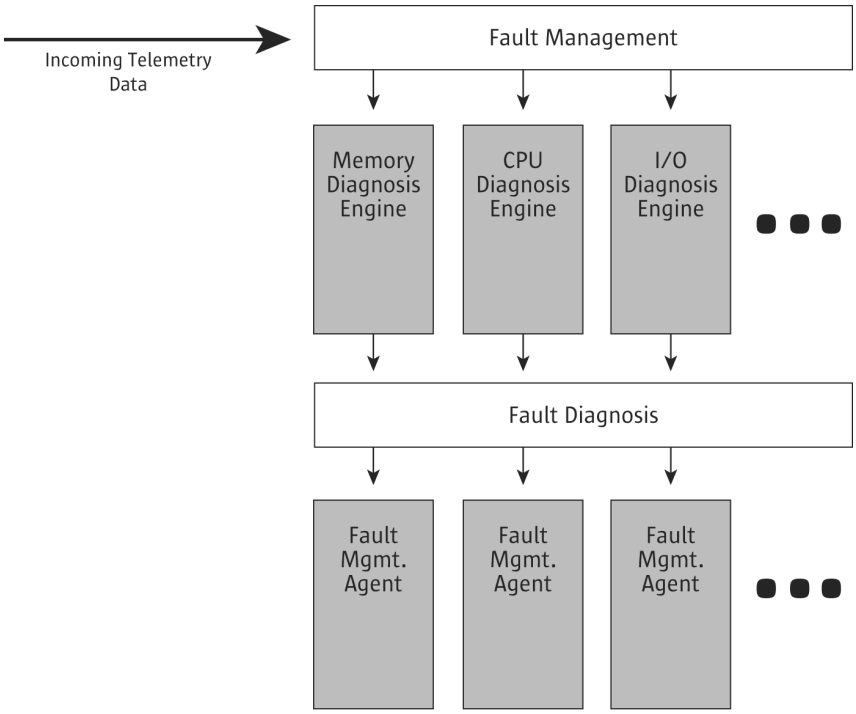


Figure 5. Solaris Fault Manager receives incoming telemetry data, makes diagnoses based on incoming events, and dispatches agents to respond to the error and provide information to administrators and to Sun.

Once Solaris Fault Manager comes up with a specific diagnosis, it provides fault information to agents that know how to respond to specific faults. An agent designed to respond to a memory error might determine the memory addresses affected by a specific chip failure and remove the affected locations from the available memory pool. An agent designed to respond to a specific software error might restart the affected service. Fault agents can also direct administrators to take specific action, and also provide information for Sun Services personnel — all with localized error messages in the administrator's language. New and improved fault agents can be dynamically incorporated into the fault manager so that more sophisticated analyses can be implemented and deployed without forcing a system reboot.

Solaris Fault Manager can grow in sophistication as new diagnosis engines and corresponding fault management agents are created — so organizations can reduce cost and complexity today, and reduce it even further as new and improved components are created.

Automating Service Management With Solaris Service Manager

UNIX systems have traditionally included a set of services, each of which consists of a single process listening and responding to requests to perform certain tasks. Examples include remote logins, file transfers, and e-mail handling. As services hosted on UNIX systems have become more complex, they encompass many processes and a wide range of startup scripts, configuration, and data files — all of which are time-consuming and difficult to manage.

Solaris Service Manager breaks with tradition by helping to make service management fast, easy, reliable, and secure. Beginning with the core set of services packaged with the operating system, Solaris Service Manager converts them into first-class objects that administrators can manipulate with a consistent set of administration commands. Even better, Solaris Service Manager is integrated with Solaris Fault Manager, so when a low-level fault is found to impact a higher-level component of a running service, Solaris Fault Manager can direct Solaris Service Manager to take appropriate action. The result is less service downtime due to hardware and software failures as well as less downtime through automated diagnosis and reduced administrator error.

Using Solaris Service Manager, administrators can take actions on services including start, stop, restart, enable, disable, view status, and snapshot. Service snapshots save a service's complete configuration, giving administrators a way to roll back any erroneous changes they might make to a service. Snapshots are taken automatically whenever a service is started to help reduce risk by guarding against erroneous errors. Organizations wishing to integrate their own business applications into this framework can develop a plug-in that lets them leverage the benefits of Solaris Service Manager.

Solaris Service Manager is implemented with three key components:

- A r*epository cache manager* that stores current service state as well as its manifest. Every service has a manifest that includes administration settings, methods for executing actions on the service, and a dependency graph. Administrators can also define roles and access control lists so that any actions taken on a service are controlled by the kernel's access control mechanisms.
- A *master restarter* can be invoked by administrators or by Solaris Fault Manager as a result of a hardware or a software fault diagnosis. It uses the dependency graph to determine exactly how to restart a service based on its current state.
- A set of *new administration tools* helps administrators maintain and take actions on services. Administration commands are integrated with the kernel's Role-Based Access Control and process privilege mechanisms. This makes it easy to create different roles for maintaining different services, allowing administration chores to be split up and/or delegated to specialists.

Increasing Availability With Sun Remote Services Net Connect

Sun Remote Services (SRS) Net Connect is a collection of services that can help businesses better manage their IT environment. It is designed for anyone who maintains and manages Sun IT environments, and provides a channel for collaborative support as well as direct access to the knowledge of Sun experts. It is Sun's intention to integrate the SRS Net Connect remote monitoring services with Solaris Service Manager and Solaris Fault Manager so that replacement parts can be on their way immediately upon a fault diagnosis.

SRS Net Connect utilizes the Internet to deliver hardware failure, system performance, and trend data to a password-protected SRS Net Connect Web portal. Its Web-based services provide the ability to self-monitor systems and storage, and to create reports to help analyze and proactively manage system performance and availability. An automatic notification feature also enables administrators to respond quickly when a system event occurs. If there is a need to engage Sun for support, important system information is immediately accessible to Sun support technicians, enabling collaboration that can help reduce time to resolution.

SRS Net Connect can help businesses:

- Enhance and optimize availability of Sun systems
- Resolve problems faster by detecting them earlier
- Improve management of the IT environment
- Realize greater value from IT investments

Unparalleled Security

The Solaris 10 Operating System includes many new ways to protect data and applications, including military-grade security features that were once available only in Sun's Trusted Solaris product. The Solaris 10 OS offers User Rights Management, supported through Role-Based Access Control (RBAC), to provide strict control over the access rights that both applications and users are allowed to exercise. Sun has also increased system and application security through its Process Rights Management mechanism that limits process access to resources for fine-grained security control while reducing the risk of system-breaching exploits. Process Rights Management is a key technology that supports Solaris Containers. Granting processes and applications rights to only the resources they require to get their job done, Solaris Containers limit the resources that can be accessed should a successful intrusion take over the process. Thus, the security features of the Solaris 10 OS enable:

- *Secure environments* through more easily hardened operating system software, protection for user data, and fine-grained limits on process rights
- *Secure applications* that are isolated from one another using Solaris Zones software, part of Solaris Containers
- *Secure communications* through standards-based interoperability, the new Solaris Cryptographic Framework, and a new integrated stateful packet-filtering firewall built into the Solaris OS

Secure Execution Environments

Most UNIX operating system kernels are littered with checks for a zero userID that indicate a request by the root user. Once a process successfully passes through such checks, it gains unfettered access to most of a system's resources. An amazing number of processes must run as root, if even for a short while. Web servers need root access in order to bind to port 80. The `ping` command needs network stack access in order to send ICMP packets. File system backup and restore commands need to run as root in order to access raw disk devices and to set userIDs on restored files. Each of these commands (and there are many more) act as a potential attack vector, because if an intruder is able to hijack the process with a buffer overflow or a more subtle semantic-based attack, the process can do whatever the intruder wishes it to do. A rogue process with access to I/O devices, memory areas, and other processes can corrupt or compromise business-critical data. It can install Trojan horses that can operate long into the future, and it can destroy application and system files that result in a complete service denial.

In the environment governed by Process Rights Management, processes with less than root privileges can have access to a restricted set of system objects, significantly limiting the damage they can do if they are somehow compromised. Adapted from the Trusted Solaris OS, Process Rights Management is integrated with User Rights Management through Solaris OS Role-Based Access Control so that administrators can assign specific rights that constrain user or process actions. For example, a Web server can be given only the specific right to bind to a low-numbered port (such as port 80), allowing it to do its job while preventing it from doing more. Sun's solution is particularly elegant because it allows existing programs to run unchanged, giving developers the opportunity to adapt this new rights management model at their convenience.

Process Rights Management implements a model that allows a process to inherit a set of rights known as its *limit set* (Figure 6). Once it begins executing, it can shrink the set of rights that it allows itself by reducing its *permitted set*. It can further reduce its own set of rights by shrinking an *effective set*. A process may expand and contract its effective set as needed to perform different functions, but its effective set can never be larger than its permitted set, and its permitted set can only shrink, never grow.

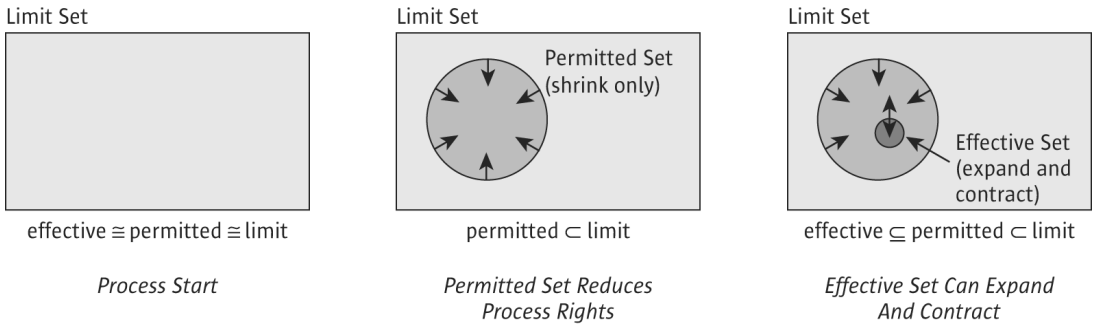


Figure 6. Process Rights Management allows processes to reduce their rights but only expand and contract them within the scope of their permitted set.

A Web server, for example, can run with the root userID but without full privileges just long enough to complete binding to port 80. Then it can change its userID and drop the privilege to bind to low-numbered ports from its effective set. Now the Web server can run as a nonroot process with its limit set containing the bind-to-port-80 right. At no point in this process does the Web server have full access to all system resources.

Sun implemented Process Rights Management by replacing all kernel checks for zero userIDs with more fine-grained rights checks. Sun also modified most system programs so that they use the new Process Rights Management facilities. The result is that this new high level of security works right out of the box:

- Process Rights Management makes it easy to significantly harden the operating system.
- It is used to restrict actions for processes in zones.
- Most system daemons now run as nonroot users with rights assigned to them that are sufficient to perform their specific function.
- The changes are transparent to applications, allowing administrators to apply privileges to existing applications.
- Developers can integrate privileges into their products to further protect their customers from potential attack through misuse of their applications.

Isolating Applications Using Solaris Zones

Solaris Containers, which can be used to isolate applications in their own virtualized Solaris OS environment, utilize security features in Solaris Zones to establish security boundaries. The secure boundaries protect an application from escaping its zone should it become compromised in some way.

Solaris Zones use Process Rights Management to establish a *limit set* for any process within the zone. Then, even the root user has access to only those resources that are assigned to the zone. For example, the root user would be allowed to mount a file system on a disk device only if that device is assigned to the zone. Similarly, a Web server can bind to port 80 on an IP address, but only the IP address that is configured onto a network interface assigned to the zone.

When organizations add powerful features like those provided by Solaris Zones to existing Solaris OS security capabilities (for example, a nonexecutable stack), they can make application environments even more secure.

Hardening the Environment With the Solaris Security Toolkit

The Solaris Security Toolkit, also known as the JumpStart™ Architecture and Security Scripts (JASS) Toolkit, provides a flexible and extensible mechanism to minimize, harden, and secure systems running the Solaris OS in a simple and automated way. Hardening essentially reduces the footprint of the Solaris OS to only those components that are needed for the specific application at hand. The reduced environment is therefore less vulnerable to an attack because it is less visible and more restricted in terms of what it can do if compromised.

The toolkit is designed to assist in the development, deployment, and maintenance of secured systems. It is comprised of a set of scripts and directories implementing the recommendations of the security-related Sun BluePrints™ OnLine articles. Extensive documentation on its use is available online at *sun.com/blueprints/browsesubject.html#security*.

Secure Communication

From the time users log into a server and access files and applications on the system, to when they venture out onto the Internet, they can be assured of one of the industry's strongest security solutions when they use the Solaris OS. Because Sun designed the Solaris OS with the belief that all systems should be networked, it has been built to securely interconnect with other systems in heterogeneous environments and to be more secure from the viruses and worms that so often plague software designed without security and networking in mind. Secure communication is critical whether building an enterprise network or providing services to millions of users over the Internet.

Interoperability Through Open Standards

Most enterprises today run heterogeneous networks, utilizing workstations, servers, and software from a variety of vendors. Secure interoperability based on open standards is key to organizations that need to contain cost and complexity. The Solaris 10 OS includes improvements to open standards-based networking beyond what was available in previous versions:

- The Solaris 10 OS provides native Lightweight Directory Access Protocol (LDAP) authentication protected by Secure Sockets Layer (SSL)/Transport Layer Security (TLS) encryption and Kerberos authentication.
- Solaris Secure Shell software, based on the open source OpenSSH, is now integrated with Solaris Pluggable Authentication Modules (PAM), providing increased flexibility in the use of authentication mechanisms. Solaris Secure Shell software is also integrated with underlying Solaris OS auditing modules to provide increased accountability.
- Solaris PAM has been enhanced to include password history maintenance and account lock-out features.
- IPSec with IKE has been enhanced to support integration with the Solaris Cryptographic Framework and offers enhanced interoperability for enterprise transparent network encryption.
- The Solaris OS implementation of Kerberos is improved through integration with the Solaris Cryptographic Framework and provides enhanced interoperability with Microsoft Windows-based systems.
- Sun has implemented NFS Version 4 to enhance the security of Network File System activities, with features including strong authentication and greater interoperability with firewalls.

Solaris Cryptographic Framework

Recent relaxations in U.S. export regulations on cryptographic software have allowed Sun to enter a new era for cryptographic security. Sun designed the new Solaris Cryptographic Framework to unify access to cryptographic functions, whether they are used by application programs, libraries, or the kernel itself. By designing the framework to support existing software that uses the *de facto* PKCS #11 standard, applications and libraries can use the new framework transparently, with no code changes necessary. The framework uses a pluggable architecture so that performance improvements, new encryption algorithms, and support for new hardware crypto accelerators can be integrated into its standards-based structure.

Two key benefits that encryption consumers will realize immediately include:

- **Support for Sun Crypto Accelerator Hardware** — The Solaris Cryptographic Framework can provide uniform access to hardware-accelerated cryptographic functions whether needed by the kernel or user applications such as SSL-enabled Web servers. Because of its transparent implementation, libraries (such as OpenSSL) and commands (such as OpenSSH) simply use hardware encryption if configured on the system and allowed by administrators. Better yet, the cryptographic framework can balance workloads across multiple crypto accelerators, significantly increasing throughput of encrypted network traffic.
- **Improved Algorithms** — Sun has significantly optimized several key cryptographic functions, and has deployed them as cryptographic framework plug-ins. Optimized algorithms include RSA, AES, 3DES, and RC4. Sun's improvement on 3DES crypto functions results in an IPSec performance boost of up to 65 percent using ESP.

The Solaris Cryptographic Framework is structured with a user-level and a kernel-level framework (Figure 7). The user-level framework provides cryptographic services to consumer applications and end-user commands. Likewise, the kernel-level framework provides cryptographic services to kernel modules and device drivers. Through the kernel crypto plug-in, user-level applications have access to the same core support — like Sun Crypto Accelerator Hardware — that the kernel uses.

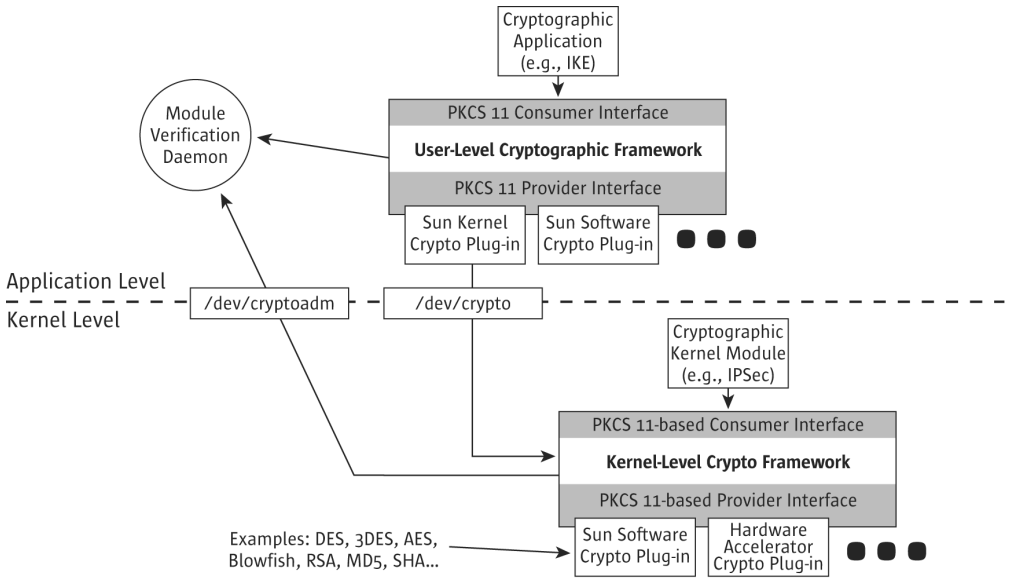


Figure 7. The Solaris Cryptographic Framework uses user-level and kernel-level software to enhance performance and centralize management of cryptographic functions.

The Solaris Cryptographic Framework uses a standard PKCS #11 interface to allow both Sun and third-party cryptographic modules to be integrated into the framework. All modules, whether from Sun or software vendors, are installed through the framework's administration facilities. In order for a module to be installed, it must be cryptographically signed by Sun, helping to ensure the software's integrity as well as its compliance with export controls.

Support for components like the Java System Web Server is provided through the mod_ssl module and the OpenSSL library. Functions including Java 2 Platform, Standard Edition (J2SE™), NFS, IPSec, Kerberos, and the Network Cache Accelerator (NCA) all use the Solaris Cryptographic Framework. The Solaris 10 OS also delivers a version of the OpenSSL cryptographic library— commonly used by open source applications — that is implemented to use the Solaris Cryptographic Framework.

Integrated Firewall

One of the most popular packet-filtering solutions on UNIX systems today is IP Filter, a cross-platform, open source packet filter that provides stateful packet-filtering, network address translation, and accounting facilities. With IP Filter available on a wide range of UNIX operating systems, Sun helps reduce administration costs by allowing customers to use the same software on a variety of OS platforms, including Linux and BSD UNIX variants.

IP Filter software protects systems running the Solaris 10 OS by limiting the type and direction of network traffic flowing to and from a system. It can be used to very quickly allow a workstation to access resources it needs on the network while limiting what network services are exposed by the system. Because IP Filter is configured through a simple text file, administrators can easily implement a standard firewall policy that is installed at the same time the operating system is installed with the Solaris JumpStart™ software.

Sun has worked with IP Filter's author to enhance both the Solaris 10 OS kernel and IP Filter software so that the two work as efficiently as possible. All improvements made by Sun have been merged back into open source IP Filter 4.0 software so that the benefits can be enjoyed across the range of heterogeneous systems found in today's data center environments.

Extreme Network Performance

Growth in Web-based applications, throughput-oriented network systems, network file services, and Secure Sockets Layer (SSL)-encrypted connections has made network performance a critically important system feature, especially for servers deployed in the application server tier and the Web server tier. To address this need, Sun has turned its focus to dramatically improving network performance on the systems that are most used in this application space: 1-4 processor servers. Improved network throughput for application servers and Web servers can have a dramatic impact on end-to-end performance of application services.

For the last 15 years, Sun has incrementally refined its STREAMS-based TCP/IP stack to provide near-linear scalability across Sun's UltraSPARC processor-based product line, ranging from 1-106 processors on a single server. Sun's goal is to improve Solaris networking capabilities to the point where even a small 4-5 processor server can saturate a high-speed network.

In the Solaris 10 Operating System, a new leap in network performance has been achieved through a redesigned implementation of the TCP/IP stack, which partially merges the TCP and IP layers in the stack. The result of Sun's work on STREAMS and the TCP/IP stack has shown quantifiable benefits for network-oriented applications. The following metrics are based on Sun's internal testing:

- Dramatic network performance improvements for UltraSPARC processor-based systems, including up to a 47-percent improvement on Web server workloads using servers with 4 processors
- Up to a 38-percent performance improvement for the Solaris 10 OS over Linux on dual-processor Sun Fire V20z servers running the Apache Web server

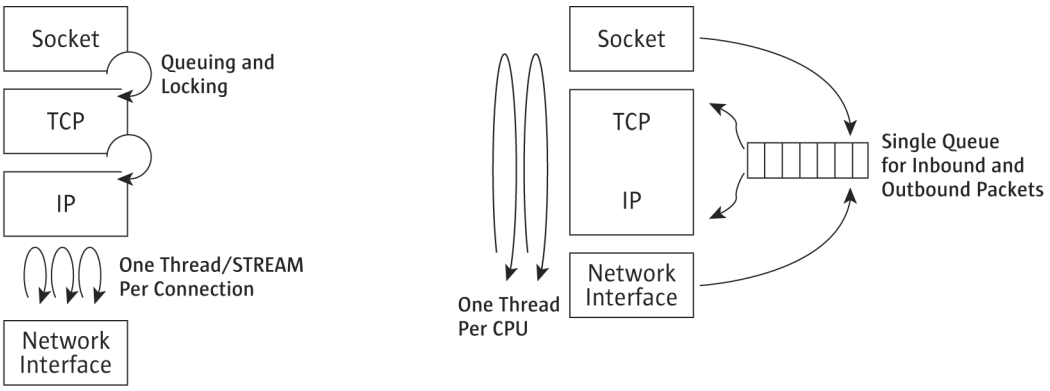
- Substantial improvements on x86 architecture systems, with up to a 34-percent improvement on Web server workloads using servers with 2 processors
- SSL performance improvements as high as 77 percent for DES encryption, and up to 130 percent for 3DES
- Maintaining a high degree of scalability on systems with more than 4 CPUs

Streamlining TCP/IP Processing

The key technology driving Sun's performance gains is a new TCP/IP stack that fundamentally changes the threading and queueing model. In the past, Sun's multithreaded TCP/IP stack was structured as two STREAMS layers with packet queueing (and locks) between layers and one processor thread assigned per connection (Figure 8). On busy multiprocessor servers, this architecture limited performance because of only modest processor locality, significant lock contention, and high context switch rates.

Sun's new TCP/IP stack merges the TCP and IP layers and allocates a single thread per CPU. The code in the new stack is streamlined to process a single packet through both layers, improving processor locality, increasing processor cache performance, and reducing context switch overhead. This new architecture uses a new threading model, binding connections to a specific CPU for their entire life, improving cache locality and vertical separation. Instead of using STREAMs-based mechanisms to protect data structures, the new stack uses a coarse, per-CPU mechanism called a *vertical perimeter* that protects the entire connection. The vertical perimeter is implemented with an IP classifier, serialization queue, and a worker thread so that only one CPU actively processes any specific packet, significantly reducing context switch overhead. Sun's approach is unique and stands in sharp contrast to fully multi-threaded approaches where each connection or incoming packet is assigned an individual thread and where each thread competes for resources. Sun's IP classifier-based approach classifies each packet to a connection and assigns it to a queue to be processed by the worker thread running on that particular CPU.

To further reduce context switch overhead, new interrupt management features allow the software to switch between polling and interrupt-driven processing dynamically. On a busy system, polling device drivers improves throughput by reducing context switch overhead.



Previous Packet Processing Model *New Streamlined TCP/IP Stack*

Figure 8. The new, merged TCP/IP stack results in better processor cache performance and lower context switch overhead.

To improve performance even further, Sun is integrating support for TCP offload engines into the Solaris OS. Today, problems with the way these network interfaces are integrated into operating systems leaves them vulnerable to packet overruns that can result in dropped connections. Sun's approach is to establish all connections in software and then offload packet processing to the hardware. This makes it possible for the Solaris kernel to process packets that might otherwise be lost by the offload engine, virtually eliminating the chance of dropped connections due to overruns. The result is the rock-solid reliability that enterprise environments demand — and the ability to run x86 architecture systems with higher network throughput levels.

For organizations using the Solaris 9 Network Cache Accelerator (NCA), Sun has integrated its functions into the new TCP/IP stack, allowing it to accelerate Web server performance transparently. Sun has preserved NCA's caching functions as well.

Platform Choice

With the Solaris 10 Operating System, Sun offers a single operating system that supports the unique strengths of both SPARC and x86 product architectures. This offers customers and independent software vendors (ISVs) flexibility in selecting the hardware architecture that best fits their needs while utilizing an OS that provides the same features, benefits, and service and support characteristics regardless of hardware architecture. In addition to running on SPARC and x86 platforms from Sun, the Solaris OS also runs on a variety of hardware from other vendors. More than 250 systems are listed on the growing Hardware Compatibility List for the Solaris Operating System (x86 Platform Edition) (see *sun.com/bigadmin/hcl/*). Full support for the Solaris OS is available to customers running it on any of the systems that have been tested and certified through the Hardware Certification Test Suite.

The Solaris OS utilizes a single common code base and delivers the exact same features on all supported platforms. Its architecture includes a layer of processor-specific code that can be quickly adapted to support a new processor architecture. Sun's experience in supporting multiple processors, including large and nonuniform memory architectures, makes the Solaris OS uniquely positioned to best support new leading-edge processors such as AMD's Opteron processor.

With feature parity across platforms and binary compatibility within processor classes, organizations protect their investment when using the Solaris OS. Support for both 32-bit and 64-bit applications includes the ability to run 32-bit applications that were compiled for previous 32-bit SPARC architectures on 64-bit UltraSPARC processors — all without recompiling. Recent efforts toward Linux compatibility with the Solaris Linux Application Environment feature will also allow customers to run Linux binary applications unchanged on the Solaris OS.

Getting the Most from New Processor Technology

Sun's long history of building highly scalable multiprocessor systems with large address spaces puts it in a unique position to best exploit the benefits of new processor technologies with its Solaris OS. For more than a decade, the Solaris OS has been designed from the ground up to support multithreaded applications on multiprocessor systems — and software developers have long been writing their software to utilize the high-performance engine that the Solaris OS provides. Sun was one of the first vendors to support 64-bit processors in a UNIX environment and the large, non-uniform memory spaces that large multiprocessing systems require.

With Sun's experience exploiting new processors and platforms for the Solaris OS, it should come as no surprise that Sun has optimized the Solaris OS for several new processor families:

- Sun's UltraSPARC IV processor utilizes Chip-level Multithreading (CMT) with two cores per chip, with plans to provide up to 8 cores per chip and four threads per core, for up to 32 threads executing on a single chip. The Solaris OS maximizes performance with CMT technology by carefully balancing workload across multiple chips to avoid saturating any chip's connection to main memory. Overall, Sun's internal engineering tests show a near doubling of performance for some applications running on dual-core systems.
- Sun optimizes performance on Symmetric Multithreading (SMT) processors like the Intel Hyperthreading Pentium IV processor by evenly distributing threads across cores. By pausing idle threads, Sun can also reduce chip power consumption by up to 45 percent.
- Since 2001, Sun has optimized performance on systems with hierarchical memory structures using Memory Placement Optimization (MPO) algorithms that store memory pages nearest the processors most likely to use them — and they likewise give that processor an affinity for processing close, low-latency pages. Sun is extending Solaris 10 OS memory placement optimization technology to AMD Opteron processor-based systems with the goal of improving memory latency and reducing traffic on AMD's Hypertransport link.
- The x86 architecture processor family has seen many processor-specific instruction set extensions, including MMX, SSE, and SSE2. The Solaris 10 OS supports the extended instruction sets in these processors, dynamically selecting the optimum implementation of system calls and performance-sensitive C Library routines (like `memcpy` and `bzero`) at boot time to maximize performance on these processors.

Endian Neutral for Choice

The Solaris OS is designed to be endian neutral, and has supported both big- and little-endian byte ordering for more than a decade. The terms big endian and little endian refer to the ways in which a processor orders bytes into words. UltraSPARC processor architectures use big-endian byte ordering, where a word stored in memory begins with its most significant byte first. x86 architecture processors use little-endian byte ordering, where the least-significant byte is stored first. The Solaris OS design allows it to change byte-ordering conventions with the flip of a compilation flag. Indeed, Sun has experimentally compiled Solaris OS software for both byte orderings on both types of processors. The bottom line is that the Solaris OS is compiled to support its industry-standard application binary interface (ABI) in the most natural byte ordering for the processor: Big endian for UltraSPARC processors and little endian for x86 architecture processors.

When big- and little-endian systems exchange data, endian-neutral formats are used, whether it is the byte ordering in TCP/IP packets or the parameters in a remote procedure call. Byte order can, however, prevent a particular file system stored using one byte ordering to be mounted directly onto a system with the opposite byte order. This is not a problem with Sun's new Solaris ZFS where the entire binary file system structure is stored in an endian-neutral format, allowing Sun customers to freely exchange disk media between processors having different byte-ordering conventions.

UNIX APIs

Solaris software provides a set of well-documented, public APIs that are used by the writers of the Java Virtual Machine and non-Java technology applications. In contrast to vendors with proprietary, shifting operating system interfaces, Sun's stable, open APIs invite others to compete for the best product implementation, improving product quality and resulting in greater customer choice. The stability of Sun's APIs frees customers from having to reengineer their applications with every OS release, helping to lower costs through investment protection.

Solaris Native Libraries

The Solaris OS supports extensions through its shared-library mechanism. This technology allows multiple applications to share large amounts of system software. The Solaris OS supports multiple versions of the same shared libraries so that old and new applications requiring different libraries can run side by side. This helps free organizations to consolidate multiple functions on the same server — a key feature for companies under pressure to reduce costs.

Because Sun develops both the SPARC and x86 platform versions of the Solaris OS, Sun is able to make processor-specific and architecture-specific optimizations. In the case of shared libraries, Sun overlays different shared libraries at install time depending on the underlying platform. Transparent to application software, these libraries help to fully exploit the capabilities of the underlying hardware. For example, applications using cryptographic functions transparently use hardware crypto accelerators if configured on the system.

Open Source Libraries

Sun encourages the sharing of software between Linux and Solaris environments, and facilitates developers using the Solaris OS to write Linux applications by including six key Linux libraries with Solaris software. Glib supports useful data types, macros, type conversions, and a lexical scanner. Gtk+ supports graphical user interfaces, including the GNOME desktop. The JPEG, PNG, and TIFF libraries support creating compressed image files in the corresponding formats. And the XML2 library supports structured XML documents through a C language interface. Through support of these and other libraries, many Linux applications can compile and run virtually unchanged in the Solaris environment.

Linux Interoperability

With unwavering support for interoperability and open standards, and a commitment to delivering customer choice, Sun has made Linux interoperability a high priority. Indeed, Sun has built Solaris and Linux interoperability in at every level, from binary and source code compatibility, to portability features in Java technology, to common applications, management software, and user environments.

Hundreds of Linux applications and libraries are provided with the Solaris OS, so Linux users will find almost all of the tools and commands they are familiar with, right down to the GNOME desktop System. Linux systems also interoperate easily with Solaris OS-based servers and workstations because they share many of the same TCP/IP-based network communication mechanisms, including Telnet, FTP, NFS, and Samba.

Solaris Linux Application Environment

Today, Sun is taking Linux interoperability to the next level with a new feature — the Solaris Linux Application Environment — in the Solaris OS for x86 platforms. The Solaris Linux Application Environment allows Linux applications to run unchanged on the Solaris OS when coupled with a Linux distribution, enabling businesses to take advantage of the innovations in the Solaris 10 OS without sacrificing investments in existing Linux applications, and allowing Solaris OS customers to gain access to the vast range of software applications available on Linux.

Customers wishing to run Linux applications on x86 systems running the Solaris OS can utilize Sun's Linux libraries and a Linux system call interface that allows Linux binaries to run without modification. The Linux system call handler in the Solaris OS catches system call interrupts from Linux binaries and dispatches the appropriate Solaris OS modules to handle the requests. The module handles device-specific requests so that even device-dependent Linux software can run unchanged.

Linux Compatibility Assurance Toolkit (LinCat)

Many Linux applications can also be recompiled and executed virtually unchanged on either UltraSPARC or x86 processor-based Solaris OS systems. The Linux Compatibility Assurance Toolkit (LinCat) further simplifies the process of porting Linux applications to run natively on the Solaris OS. The LinCAT utility can be used to identify porting issues in C/C++ source code files. IT analyzes the effort of porting such source files from Linux to the Solaris OS and can help estimate the total time required for a port. Further information and a download link for the LinCAT utility can be found at *developers.sun.com/prodtech/solaris/downloads/lincat*.

Chapter 4
Conclusion

The Solaris 10 Operating System is a significant leap forward from the Solaris 9 OS, establishing it in a class by itself when compared to competing operating systems. It offers many innovative new technologies that fundamentally change the equation for organizations needing to reduce costs, reduce complexity, and minimize risk. The new features in the Solaris 10 OS bring mainframe-quality software to even the smallest single-processor servers and provide a stepping stone into tomorrow's data center.

- **Relentless Availability**. The Solaris 10 OS helps to reduce downtime through Solaris Fault Manager and Solaris Service Manager, designed to work together to not only reduce downtime but also reduce complexity. It helps organizations isolate applications through Solaris Containers, and minimize the risk of data loss with Solaris ZFS.
- **Optimal Utilization**. Solaris Containers technology is a breakthrough approach to virtualization, offering multiple software partitions on a single operating system instance. Solaris Containers make it simple, safe, and secure to consolidate data center servers. They help organizations to align resource allocation with business goals, increase uptime with partitions that can reboot in only a few seconds, and reduce costs by simplifying and accelerating consolidation efforts while reducing system administration and maintenance overhead.
- **Extreme Performance**. The Solaris 10 OS includes a new TCP/IP stack that brings performance of UltraSPARC processor-based servers on par with x86 architecture servers having double the clock rate. It includes system call performance improvements across the board, including many 15x and some 25x speed-ups. Key to overall data center efficiencies is application performance — and Solaris DTrace is a powerful tool to analyze and diagnose elusive problems in real time. Safe and comprehensive, it provides a virtually unlimited number of probes, simplifying the task of locating failures and performance bottlenecks. Requiring minimal overhead when activated, it uses zero overhead when deactivated, making it a powerful tool standing ready to serve.
- **Unparalleled Security**. Incorporated from military-grade Trusted Solaris software, Process Rights Management features make it easy to harden Solaris OS systems and provide the security foundation for Solaris Containers. Solaris Containers restrict applications to using only the specific resources they are authorized to use, and protect errors and even breaches in application security from propagating to other Containers on the same system. The Solaris Cryptographic Framework brings consistent, transparent use of streamlined cryptographic functions and hardware crypto accelerators to both kernel- and application-level software. And an open source firewall helps lock down systems while minimizing administrator learning curves.

- *Platform Choice*. The Solaris 10 OS supports systems based on both the newest UltraSPARC and x86 architecture processors, including the UltraSPARC IV and AMD Opteron chips, giving organizations the choice of platform to best meet their needs. Once the choice is made, Sun supports binary compatibility within processor architectures and source-code compatibility across platforms, giving organizations the investment protection they require in running today's lean and mean data centers. Finally, Linux applications can now run unmodified on x86-based systems, letting organizations use the rock-solid support of the Solaris 10 OS to run some of the most popular Linux applications.

For long-time Sun customers, the new features embodied in the Solaris 10 Operating System should come as no surprise because Sun has been a leader in innovation for more than two decades. For years, the flexible, modular structure of the Solaris OS has allowed it to grow and evolve to meet the needs of today's demanding environments, incorporating new features without the risk of costly and error-prone rewrites. With Sun's years of experience with new processor architectures and nonuniform memory structures, Sun's Solaris 10 OS is well-positioned to support today's data centers and those evolving into the future. For new Sun customers, it is time to experience the relentless availability, optimal utilization, extreme performance, and unparalleled performance that the Solaris 10 Operating System has to offer.

For More Information

Additional information about related Sun technologies and programs are available at the following Web links.

Web Site URL	Description
sun.com/solaris	Solaris Operating System
sun.com/software/javaenterprisesystem	Sun Java Enterprise System
developers.sun.com/prodtech/solaris/downloads/lincat	LinCat (Linux Compatibility Assurance Toolkit)
sun.com/bigadmin/hcl/overview.html	Solaris Operating System (x86 Platform Edition) Hardware Compatibility List
sun.com/software/solaris/solaris-express	Sun Software Express Program

Table 1. Web links for additional information

Please
Recycle

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 1-650-960-1300 or 1-800-555-9SUN **Web** sun.com

The Network is the Computer™

Sun Worldwide Sales Offices: Argentina +5411-4317-5600, Australia +61-2-9844-5000, Austria +43-1-60563-0, Belgium +32-2-704-8000, Brazil +55-11-5187-2100, Canada +905-477-6745, Chile +56-2-3724500, Colombia +571-629-2323 Commonwealth of Independent States +7-502-935-8411, Czech Republic +420-2-3300-9311, Denmark +45 4556 5000, Egypt +202-570-9442, Estonia +372-6-308-900, Finland +358-9-525-561, France +33-134-03-00-00, Germany +49-89-46008-0 Greece +30-1-618-8111, Hungary +36-1-489-8900, Iceland +354-563-3010, India–Bangalore +91-80-2298989/2295454; New Delhi +91-11-6106000; Mumbai +91-22-697-8111, Ireland +353-1-8055-666, Israel +972-9-9710500 Italy +39-02-641511, Japan +81-3-5717-5000, Kazakhstan +7-3272-466774, Korea +822-2193-5114, Latvia +371-750-3700, Lithuania +370-729-8468, Luxembourg +352-49 11 33 1, Malaysia +603-21161888, Mexico +52-5-258-6100 The Netherlands +00-31-33-45-15-000, New Zealand–Auckland +64-9-976-6800; Wellington +64-4-462-0780, Norway +47 23 36 96 00, People's Republic of China–Beijing +86-10-6803-5588; Chengdu +86-28-619-9333 Guangzhou +86-20-8755-5900; Shanghai +86-21-6466-1228; Hong Kong +852-2202-6688, Poland +48-22-8747800, Portugal +351-21-4134000, Russia +7-502-935-8411, Saudi Arabia +9661 273 4567, Singapore +65-6438-1888 Slovak Republic +421-2-4342-94-85, South Africa +27 11 256-6300, Spain +34-91-767-6000, Sweden +46-8-631-10-00, Switzerland–German 41-1-908-90-00; French 41-22-999-0444, Taiwan +886-2-8732-9933, Thailand +662-344-6888 Turkey +90-212-335-22-00, United Arab Emirates +9714-3366333, United Kingdom +44-1-276-20444, United States +1-800-555-9SUN or +1-650-960-1300, Venezuela +58-2-905-3800, or online at sun.com/store

11/04 R1.0