Oracle White Paper
October 2012

Oracle Solaris 11: What’s New for Application Developers

Introduction

Streamlining the development process for enterprise applications is key to business success. To be as productive as possible, enterprise developers need an integrated environment that lets them work on tasks concurrently, collaborate with other developers, and create optimized and parallelized applications in less time. Yet no application is perfect. Studies show that the earlier a software defect is found in the development cycle, the less time-consuming and expensive it is to fix. As a result, sophisticated debugging and analysis tools that enable the construction of quality solutions are a must for enterprise developers.

For over two decades, Oracle Solaris has been the platform of choice for enterprise developers. Providing a rich environment for the creation and deployment of strategic enterprise applications, Oracle Solaris combines key computing elements—operating system, networking, storage management, and user environment—into a stable, high-quality foundation that developers can depend on when creating solutions. Many innovations, including built-in virtualization technology, broad hardware applicability, massive scalability, rich security capabilities, debugging and analysis tools, and more, have made Oracle Solaris the mission-critical operating system.

This paper discusses why Oracle Solaris 11 continues to be the best platform for enterprise application development, offering key advantages to help streamline the software lifecycle through development, test, and production environments and ensure a secure and highly reliable foundation for application deployment. It examines the key advances in Oracle Solaris 11 and their value to application developers.

Note: This article includes some features that are only available in Oracle Solaris 11.1. Those wishing to develop for the Oracle Solaris platform are encouraged to upgrade to this release.

Oracle Solaris 11: The First Cloud OS

Oracle Solaris 11, the first cloud operating system, raises the bar on enterprise innovation bringing the reliability, security, and scalability of the #1 enterprise operating system to the cloud. With simplified administration, designed-in virtualization, scalable data management, and advanced protection capabilities, Oracle Solaris 11 provides a comprehensive platform to run the most demanding of enterprise applications in private, hybrid, or public clouds. Oracle Solaris 11 provides the foundation of Oracle's complete applications-to-disk technology stack and a strategic platform for application developers.

The key areas of advancement in Oracle Solaris 11 include the following:

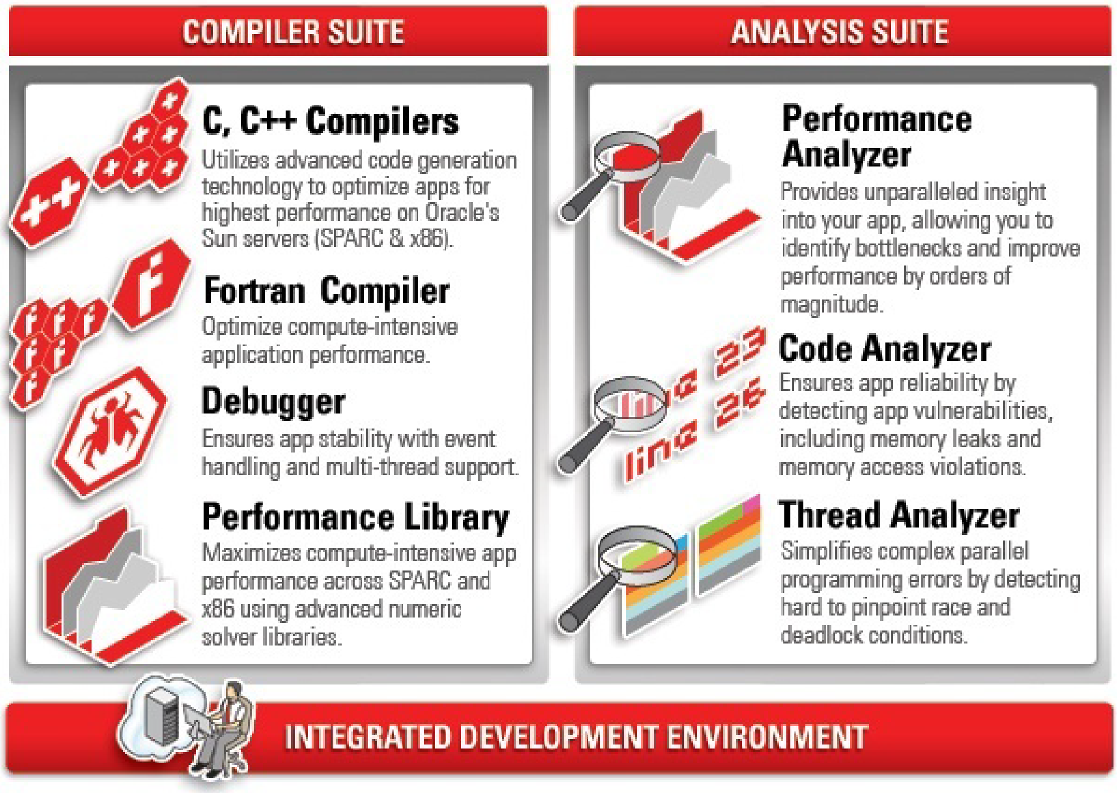
- **Simplified administration:** Oracle Solaris 11 includes simplified package management through the Image Packaging System (IPS), reducing the complexity of keeping software up to date on a system with integrated support for Oracle Solaris Zones virtual environments. With built-in safety, administrators update systems over the network in cloned boot environments reducing planned system downtime and maximizing application availability. IPS provides a fast and efficient way for developers to distribute their software through network-based package repositories.
- **Designed-in virtualization:** Oracle Solaris 11 is a completely virtualized operating environment, with technologies that span network, storage, and server virtualization to support consolidation initiatives effectively. Oracle Solaris 10 Zones provide fully compatible environments running on Oracle Solaris 11, helping developers transition their applications with minimum disruption.
- **Scalable data management:** Oracle Solaris 11 provides for all emerging "big data" needs, as driven by new applications, devices, and data mining methodologies. Oracle Solaris ZFS, the default file system on Oracle Solaris 11, has no practical limits to file system size and has no need for a separate volume manager. Oracle Solaris ZFS also contains built-in data services, such as deduplication, RAID, and compression, helping to prevent additional costs in enterprise development environments. Developers can quickly take advantage of snapshots and clones for increased efficiency in their development environments.
- **Advanced protection.** Oracle Solaris 11 advances data and system security with on-disk encryption through Oracle Solaris ZFS and fully secure-by-default environments. Enhancements to the Oracle Solaris Cryptographic Framework, a unified software interface for encryption services, allow developers to take advantage of next-generation hardware, including hardware-based cryptographic acceleration in Oracle's SPARC T4 processor.

Streamline the Application Development Process

Coding and testing applications is a time-consuming and often complex process. While individual point products can help with certain tasks, building applications with an integrated platform designed for all of the pieces to work together streamlines workflow and results in more robust applications. Oracle Solaris 11 supports a rich set of tools for optimizing the development process.

Oracle Solaris Studio

Oracle Solaris Studio provides developers with everything needed to create high-quality, cross-platform desktop, enterprise, and Web applications. An integrated development environment optimizes the application development process, from creating and building C, C++, Java, or Fortran applications, to debugging problems and tuning for optimal performance. By integrating all the steps programmers take—from GUI design and code generation, to edit-compile-debug-tune cycles—the Oracle Solaris Studio integrated development environment (IDE) makes it easy to rapidly build applications.



COMPILER SUITE

C, C++ Compilers
Utilizes advanced code generation technology to optimize apps for highest performance on Oracle's Sun servers (SPARC & x86).

Fortran Compiler
Optimize compute-intensive application performance.

Debugger
Ensures app stability with event handling and multi-thread support.

Performance Library
Maximizes compute-intensive app performance across SPARC and x86 using advanced numeric solver libraries.

ANALYSIS SUITE

Performance Analyzer
Provides unparalleled insight into your app, allowing you to identify bottlenecks and improve performance by orders of magnitude.

Code Analyzer
Ensures app reliability by detecting app vulnerabilities, including memory leaks and memory access violations.

Thread Analyzer
Simplifies complex parallel programming errors by detecting hard to pinpoint race and deadlock conditions.

INTEGRATED DEVELOPMENT ENVIRONMENT

Figure 1. The Oracle Solaris Studio IDE integrates advanced tools and helps streamline the development process.

Build Applications

Oracles Solaris Studio provides the tools needed to build enterprise applications faster.

- **Gain efficiency with an integrated environment.** All tools in the Oracle Solaris Studio suite—compilers, debuggers, code coverage tools and more—come together in the IDE to provide an optimized development platform for the end-to-end process of building, debugging, analyzing, and tuning single-threaded and multithreaded applications. Developers can exploit platform functionality while accelerating and simplifying the process of application development.
- **Build better and faster applications with sophisticated C, C++, and Fortran compilers.** Currently a leader among IDEs in multithreading development, Oracle Solaris Studio offers high-performance parallelizing compilers that are tuned for Oracle Solaris 11 and underlying hardware, resulting in the best overall development platform for Oracle systems. Indeed, the record-setting application performance of Oracle Solaris Studio C, C++, and Fortran compilers consistently exceeds the performance of open-source alternatives.
- **Create cross-platform applications.** Oracle Solaris Studio compilers provide a solid foundation for building robust, high-performance, parallel code for Oracle systems with single or multicore SPARC, x64, and x86 processors. Utilizing the extensive set of 32- and 64-bit options available in the Oracle Solaris Studio compilers, applications can be built to meet a broad range of needs. For instance, an application can be compiled to provide reasonable performance on any SPARC processor, resulting in a single application binary for all Oracle systems with SPARC processors.

  Alternatively, developers can compile the same application to run with maximum performance by using different compiler options to take advantage of features found in the latest 64-bit UltraSPARC processor. This is accomplished by specifying the type of processor to be used so that the compilers can generate code optimized for different processor instruction sets. As a result, the same application source code can be used to create optimized application binaries for each SPARC architecture, maximizing performance across the entire SPARC product line.

Debug Applications

Oracles Solaris Studio provides powerful debugging solutions for single-threaded and multithreaded application development to enable developers to find problems faster.

The `dbx` debugger is an interactive, source-level, postmortem and real-time debugging tool available through both a command line and graphical interface. It is also integrated into the Oracle Solaris Studio IDE. Ensuring application reliability, the `dbx` debugger is scriptable and multithread-aware.

Analyze Applications

Built-in advanced profiling and observability tools make it easier to identify performance hotspots in serial and parallel applications and to visualize application behavior.

- **Analyzing application performance.** The Performance Analyzer identifies application performance bottlenecks by specifying not only which functions, code segments, and source lines are having an impact on performance but by also providing the tools necessary to tune for optimal performance. From annotated listings showing what optimizations the compiler performed, to optimization status and runtime thread performance, users can easily visualize performance hotspots. The Performance Analyzer can profile single-threaded as well as multithreaded applications. The Performance Analyzer also handles multiple concurrent processes to collect system-wide performance data, providing application insights down to the OS kernel, presenting a graphical identification of bottlenecks and helping improve application performance by orders of magnitude. In addition to supporting C, C++, and Fortran applications, the Performance Analyzer also includes support for Java code. The Performance Analyzer also provides the ability for developers to visualize the performance of distributed MPI-based applications via intuitive graphs to reveal performance hotspots in a clustered environment. Basic application profiling data is available, as are various MPI message statistics, such as message length, internode message count, and message latency.
- **Checking application source code.** The Code Analyzer helps ensure application reliability by utilizing dynamic, static, and code coverage analysis to detect 45+ common coding errors, including memory leaks and memory access violations, faster than competitive alternatives. The Code Analyzer utilizes static analysis at compilation and dynamic analysis from application runtime to identify code quality issues. In addition, the tool incorporates code coverage data to provide information about functions that are not covered by your test suite and guidance on the type of benefit you could achieve by covering those functions. The Code Analyzer provides a comprehensive view of application vulnerabilities by synthesizing the data collected from these three types of analysis, enabling you to improve application correctness and reliability. It also provides advanced error filtering and sorting capabilities, enabling you to track, detect, and fix issues faster.
- **Ensuring correct application threading.** Improving developer productivity and software robustness, the Thread Analyzer tool identifies hard-to-detect threading errors before they occur. It can detect potential race and deadlock conditions at runtime, map them to source lines in the application, and then enable the user to view the results by using command-line or graphical user interface (GUI) options.

More information on Oracle Solaris Studio can be found on the Oracle Technology Network at http://www.oracle.com/technetwork/server-storage/solarisstudio/overview/index.html.

Additional Developer Software

Oracle Solaris 11 includes a wide range of other tools that facilitate application development.

- **Java.** Oracle Solaris 11 helps Java developers create applications with superior performance, scalability, and security. By combining Java and Oracle Solaris, enterprise developers can take advantage of superior out-of-the-box performance, high-quality virtual machine implementations, and innovative tools designed to simplify Java code debugging and performance tuning tasks. More information can be found in Oracle Solaris: The Platform of Choice for Java Application Development and Deployment.
- **Other programming languages.** Oracle Solaris 11 now includes Python and Ruby for high-level application development. These tools are integrated with DTrace, Oracle Solaris analysis tools, and the new Image Packaging System to ease the debugging process and simplify the task of finding the right versions of software.
- **Source code management.** A number of popular tools have been included in the Oracle Solaris 11 network package repository, including Mercurial and Git, for efficient distributed source-code version control, and Meld, for a graphical source code diff and merge utility.
- **Powerful scripting.** Oracle Solaris 11 includes a newer version of the popular Perl language, providing developers with powerful scripting capabilities and text processing facilities.
- **A new default system shell.** For years, the Korn shell (`ksh`) was the default shell environment in Oracle Solaris. With Oracle Solaris 11, `ksh` is superseded by `ksh93`, an updated and improved version of the popular system shell. Note that the .kshrc file is read in addition to the .profile file when the `ENV` environment variable is not set.
- **A new default interactive shell.** To ease familiarity with other operating systems, Oracle Solaris 11 uses the Bourne-again shell (`bash`) as the default interactive shell for users. When used as a login shell, `bash` finds the first instance of a .bash_profile, .bash_login, or .profile file that exists and uses that configuration information. When a non-login interactive shell is used, the .bashrc file is read instead. When a non-interactive shell is used, `bash` uses the `$BASH_ENV` environment variable setting. More information on `bash` can be found in the `bash(1)` man page.
- **Environment variable and path enhancements.** A number of existing Oracle Solaris utilities have been enhanced to include popular GNU-like command line options for user familiarity. GNU utilities have been included in /usr/bin where no conflict with existing Oracle Solaris utilities occurs. In addition, the `MANPATH` environment variable no longer is required. The `man(1)` command determines the appropriate `MANPATH` based on the `$PATH` environment variable setting. Note that `MANPATH` can be set; however, it usually is not necessary to override automatic handling. Finally, files previous located in the /usr/sfw directory now are located in the /usr/bin directory. Links exist so that either path is valid; however, developers can simplify path settings to use the new location.

More information on Oracle Solaris 11 can be found on the Oracle Technology Network (OTN) at http://www.oracle.com/technetwork/server-storage/solaris11/overview/index.html.

Simplify Application Testing

Testing applications is a critical step in the development process. Oracle Solaris 11 includes several features that help developers validate that applications work in multiple operating environments and find and fix problems faster.

Replicate Development, Test, and Production Environments

Creating identical instances for development, test, and production environments is a time-consuming process that is prone to error. Applications need to be able to move throughout the build, test, and deploy chain in a timely manner. With Oracle Solaris Zones and Oracle Solaris ZFS, applications can be developed in an isolated environment, cloned, and packaged for movement to testing systems (Figure 2). Prebuilt, preconfigured, prepatched virtual environments and their application stacks are replicated on new machines. Shared storage makes it possible for the transition to happen quickly, and applications do not need to be duplicated. Once tested, applications can be moved quickly to production systems. With these capabilities, organizations can experience rapid rollout of applications, little downtime, and automatic rollback to development and testing systems when needed.

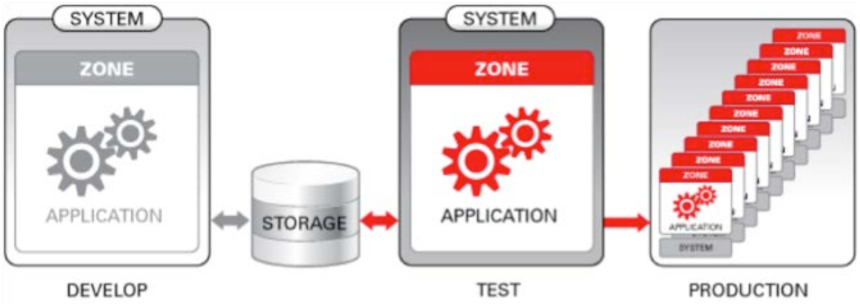


Figure 2. Oracle Solaris Zones facilitate the movement of entire application stacks between systems.

Validate Application Behavior in Multiple Environments

Developers and quality assurance teams need to test applications on multiple systems, each running a different operating system or patch level. Using Oracle Solaris Zones, developers can run multiple environments on a single server running Oracle Solaris 11 (Figure 3). In this solution, the server is divided into distinct areas that each runs an operating system in an isolated application execution environment at the same time on the same system. Each zone provides the functionality of previous-generation operating systems, as well as any application software, and can take advantage of the resources and services provided by the underlying hardware in order to complete tasks.

As a result, the specific configuration needs of applications under test can be addressed individually. In fact, different versions or patch levels of an operating system can be hosted on the system, giving each application access to the specific operating system features needed.

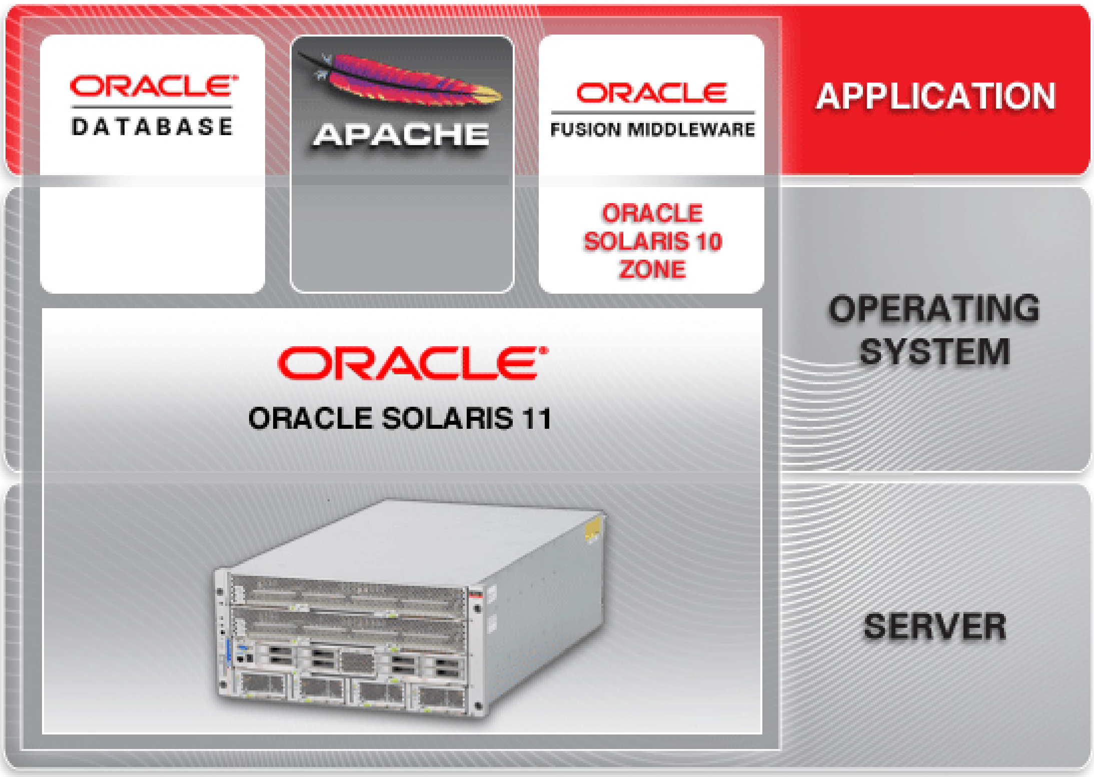


Figure 3. Developers can run applications in virtualized environments, including Oracle Solaris 10, to aid transition to Oracle Solaris 11.

More information on Oracle Solaris Zones can be found in the System Administration Guide: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management.

Migrate Oracle Solaris 10 Applications to Oracle Solaris 11 with Ease

While the binary compatibility guarantee of Oracle Solaris helps ensure applications run from one release to another, there might be times when having a complete runtime environment for Oracle Solaris 10 applications is needed. Oracle Solaris 10 Zones can host Oracle Solaris 10 10/09 and later environments on SPARC or x86 servers running Oracle Solaris 11. Built-in tools, including preflight checking, enable the migration of an existing physical Oracle Solaris 10 system into a virtual environment on the target system or the migration of a virtualized Oracle Solaris 10 environment, facilitating the immediate adoption of Oracle Solaris 11.

Observe Application Behavior and Diagnose Faults

Enterprise applications are sophisticated and complex, often consisting of multiple pieces of software that interact with one another to deliver needed functionality. With software components spread across multiple systems, isolating problems can be difficult. In addition, traditional tools make it impossible for developers to debug live production environments without affecting performance, and entire production environments cannot be replicated due to their size and complexity. The Dynamic Tracing (DTrace) facility in Oracle Solaris 11 lets enterprise developers turn a production deployment environment into a debug environment—without impacting user access or application performance.

DTrace is a dynamic tracing framework that provides top-to-bottom system observability for troubleshooting systemic problems in real time. Designed to quickly identify the root cause of system performance problems, DTrace combines over a hundred thousand trace points with a powerful scripting language and a simple, interactive command-line interface. It works by safely and dynamically instrumenting the running operating system kernel and applications with trace points that are completely passive until enabled. These trace points can be enabled quickly for data collection and then disabled again to minimize performance impacts on the system being examined. With this information, developers can rapidly observe production environments, identify performance bottlenecks, and troubleshoot issues in a secure and safe manner, providing the following benefits.

- **Ease application observation.** Observing applications in production environments often is disruptive. Indeed, many tools that are typically used to observe applications cannot be used on production systems. Unlike traditional debug flags that must be turned on when applications are started, the dynamic nature of DTrace makes it possible to turn debug flags on and off at runtime. With DTrace, developers can turn on select probes or scripts only when needed, eliminating diagnostic overhead when they are disabled.

- **Support QA testing efforts.** Error induction and code coverage testing is expensive to conduct. DTrace scripts can help QA engineers evaluate code coverage rates. In addition, DTrace provides mechanisms for introducing errors to see how systems behave under adverse conditions. For example, the `chill` action can be used to introduce race conditions, and the `copyout` action can be used to modify variables at runtime.

- **Observe all levels of the technology stack.** Unlike tools that focus on specific debugging tasks or application layers, DTrace provides a single tool for observing all levels of the technology stack. For example, device driver developers can use the anonymous tracing feature in DTrace to observe the boot phase, and telemetry can be collected for system management applications. In addition, DTrace includes prebuilt instrumentation for I/O scheduling, signals, networks, iSCSI communication, and applications written in C, C++, Java, Ruby, PHP, Perl, and more.
- **Instrument Java code.** While DTrace includes embedded probes for C and C++ applications, scripts must be used for Java applications. Alternatively, an additional tool can be used to dynamically trace running Java programs. BTrace dynamically instruments target application classes to inject tracing code. Now BTrace and DTrace can be used together to embed DTrace probes in Java applications. The framework creates and compiles a BTrace script and embeds it in Java code.
- **Take advantage of new providers and probes.** A number of new DTrace providers and probes are included in Oracle Solaris 11.
  - A `cpc` provider for connecting CPU events, such as TLB or L2 cache misses, to the cause of the event on a system-wide basis
  - New `tcp`, `udp`, and `ip` providers for tracing the TCP, UDP, and IPv4/IPv6 network protocols
  - An `iscsi` provider for tracing iSCSI activity
  - PHP and MySQL probes
  - A `kerberos` provider for tracing Kerberos messages described in RFC4120

  More information on DTrace can be found in the Oracle Solaris Dynamic Tracing Guide.

Find and Understand Faults Faster

As applications gain complexity, analyzing error conditions becomes increasingly complicated. Developers understand that finding and fixing faults quickly can alleviate resulting side effects. Oracle Solaris 11 includes a framework for a decision system that can analyze, report, and, if possible, work around faults automatically. The Service Management Facility (SMF) provides the ability to view system-wide service status, manage services and their dependencies, and automatically restore services when a failure occurs. The Fault Management Architecture (FMA) enables systems to automatically monitor, report, and recover from hardware errors to maintain data integrity and keep mission-critical applications and services online. Together, these tools can help developers to find and fix faults faster.

Significant improvements have been made to SMF and FMA in Oracle Solaris 11:

- Developers can be notified of service state transitions and fault management events via SNMP traps or e-mail messages.
- With the introduction of new installation and packaging technologies, the SMF configuration repository has been modified to allow better control and auditing of administrative customizations of service and system configuration through *repository layers*.
- FMA reports errors in a standard format that can be analyzed. Root cause can be determined in an automatic way and reported in readable and actionable formats. In addition, Oracle continues to integrate FMA with new hardware features as they are released.

In addition, the following best practices can help developers move applications to the SMF and FMA framework and take advantage of the key benefits it offers:

- Eliminate custom scripts that analyze application health and restart applications. SMF provides a simple way to encapsulate and standardize the methods for starting, stopping, and restarting applications.
- Make applications SMF-aware during initial development, if possible. Identify fault states and create a fault tree. Review error messages and determine if they can become FMA events.
- Convert legacy .rc scripts to SMF manifests.
- Convert custom scripts to SMF manifests and profiles with the help of the `svcbundle` command line utility. Look for instances of the start method, stop method, and check status method. Once these areas are converted, migrating the remainder of the script typically is straightforward.
- Use privileges with SMF to minimize security exposure. Be sure to release the privileges after the SUID process starts.

Detailed information on SMF and FMA can be found in the Managing Services and Faults in Oracle Solaris 11.1 guide.

Improve System and Application Deployment

Oracle Solaris 11 introduces a completely new software installation and deployment architecture that makes it easier for developers to create deployment environments and manage the software lifecycle.

Updated Installation and Configuration Programs

Oracle Solaris 11 provides updated installation and configuration programs—from interactive interfaces to full automation—to give developers the right amount of control when preparing systems.

- **Automated installer.** The automated installer makes it possible to apply an Oracle Solaris 11 installation to multiple systems. Developers can quickly create an installation service that lets systems look for, locate, and install a manifest that matches specifications. The automated install images are bootable, providing an easy way of installing Oracle Solaris 11 on SPARC and x86 systems without the need to set up a server with an active install service. Simply boot off a CD, point the system at an install manifest (or default to the one included in the media), and watch while the system automatically is installed over the network. With this tool, developers can replicate operating environments on systems to speed the application testing and deployment process. See the Installing Oracle Solaris 11 Systems guide for more details.
- **Distribution Constructor.** The Distribution Constructor is a new command-line tool for building preconfigured, bootable Oracle Solaris 11 images that contain a collection of software. The tool takes an install manifest file as input and outputs an ISO image or virtual machine image, allowing developers to create completely customizable golden images for use in development, testing, and deployment environments. See the Creating a Custom Oracle Solaris 11 Installation Image guide for more details.

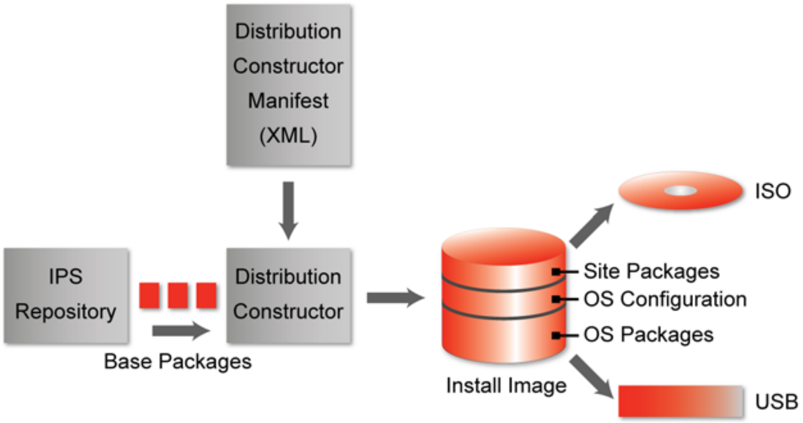


Figure 4. Distribution Constructor makes it easy for developers to create software, configuration, and target media.

- **Graphical LiveCD.** The Oracle Solaris 11 LiveCD for x86 gives developers the ability to boot directly off CD and evaluate the operating system in a complete graphical desktop environment prior to installing it on the system. The LiveCD installs a predetermined selection of software with a full desktop environment.
- **Text-based interactive installer.** A new text-based interactive installer lets developers install systems using a console. It installs a selection of software considered suitable for server deployments. For example, graphical desktop environments and audio and wireless networking drivers are not installed. However, several configuration options are available, including IPv4 and IPv6 network interfaces, a DNS resolver, LDAP clients, and a name service switch.

TABLE 1. COMPARISON OF ORACLE SOLARIS 10 AND ORACLE SOLARIS 11 INSTALL FEATURES

FEATURE	ORACLE SOLARIS 10		ORACLE SOLARIS 11	
Interactive Install	Yes		Yes	LiveCD, text-based user interface, updated look and feel, live desktop
Live Upgrade	Yes	Oracle Solaris Live Upgrade	Yes	Boot Environments
Network Install	Yes		Yes	
Hands-Free Install	Yes	Oracle Solaris JumpStart	Yes	Automated Installer
User-Specified Install Parameters	Yes	Oracle Solaris JumpStart, text file, not easily updated	Yes	Automated Installer manifests, XML-based, easily extended, default provided
Begin, Finish Scripts	Yes		Yes	Derived manifests (Begin), device driver utility (Begin), first-boot SMF services (Finish)
Heterogeneous Environment Support	Limited	Non-standard protocols	Yes	
Software Customization	Limited	User-defined HTTP package repository	Yes	Integration with Image Packaging System, multiple repositories available
System Configuration	Yes	`sysidtool`, `sysidcfg`	Yes	Enhanced SMF profiles, stable system interfaces, more extensible, SMF profile, text-based utility
WAN-Based Installation	Yes	SPARC only	Yes	SPARC and x86, Image Packaging System repository-based installs
Fully Integrated Install Server Utilities	No		Yes	
Derived Profiles	Yes		Yes	Automated Installer derived manifests
Packaging Independent of Media	No	SVR4 packages included with media	Yes	Image Packaging System packages are separated from delivered media, reduces the need for image storage and the burden on developers, provides better performance, WAN-based
Easy to Customize Distribution	No		Yes	Distribution Constructor

New, Modern Software Packaging Model

Oracle Solaris 11 modernizes the software packaging model. The Image Packaging System (IPS) is a new network-based package management system that provides a framework for complete software lifecycle management, including the installation, upgrade, and removal of software packages. IPS takes the guesswork out of configuring systems. Enterprise developers can easily determine what software is installed, learn whether needed patches are installed, and rest assured that dependent patches are identified and installed automatically. In addition, developers can use IPS to get their enterprise applications ready for distribution.

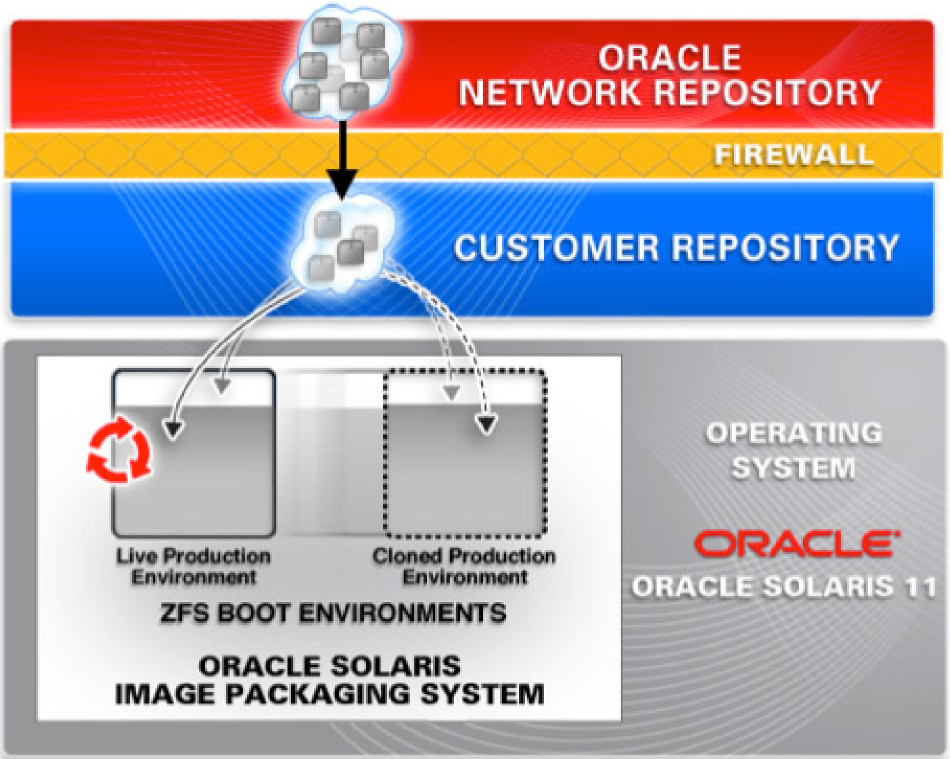

ORACLE NETWORK REPOSITORY
FIREWALL
CUSTOMER REPOSITORY
Live Production Environment
Cloned Production Environment
ZFS BOOT ENVIRONMENTS
ORACLE SOLARIS IMAGE PACKAGING SYSTEM
OPERATING SYSTEM
ORACLE
ORACLE SOLARIS 11

Figure 5. The Image Packaging System (IPS) makes it easy for developers to deliver software on Oracle Solaris 11

- **A new software publishing model.** IPS provides an easy method for publishing software packages. Package content, metadata, and dependent system services are added to a repository upon installation. Software repositories can be created and managed for local software delivery, and multirepository support lets developers pull software and fixes from different sources. While IPS packaging is the default for Oracle Solaris 11, compatibility with previous SVR4 software packages is preserved with continued access to `pkgadd` and related commands. However, the traditional patch tools to manage updates to SVR4 software packages have been removed. IPS also provides the ability for application developers to easily create new packages, or convert over legacy SVR4 packages.
- **A new set of repositories.** Oracle Solaris 11 defines a new set of new repositories that contain additional software packages (see Table 2). The repositories can be added in the Package Manager or by using the `pkg set-publisher` command.
- **More reliable application installation, version control and locking, and minimization.** Package refactoring simplifies minimization efforts and supports version control measures. In addition, developers can use IPS to validate installed packages and make any needed changes.
- **No patching.** IPS eliminates lengthy and complicated patching procedures. Preflight checking and automated downloads ensure only the needed differences are obtained and installed.

- **Safer system upgrades.** IPS and Oracle Solaris ZFS work together to deliver safe system upgrades through boot environments. Developers can install software from a series of network-based package repositories with full automatic dependency checking for additional libraries that might be required during software package installation. This automation is also integrated into Oracle Solaris Zones providing easier and safer upgrades across virtual environments.
- **A choice of interfaces.** IPS provides two different interfaces for interacting with the packaging system. A command-line interface and graphical Package Manager and Update Manager give developers flexibility. In addition, MIME associations allow for single-click package installs while browsing the Web and on-disk archive formats.

TABLE 2. IMAGE PACKAGING SYSTEM REPOSITORIES

REPOSITORY	DESCRIPTION
http://pkg.oracle.com/solaris/release	The default repository for new Oracle Solaris 11 users. This repository receives updates for each new release of Oracle Solaris. Significant bug fixes, security updates, and new software may be provided at any time for users to install at Oracle's discretion.
https://pkg.oracle.com/solaris/support	Provides bug fixes and updates. This repository is accessible with a current support contract from Oracle.

More information on the Image Packaging System can be found in the Adding and Updating Oracle Solaris 11 Software Packages guide and Packaging and Delivering Software with the Image Packaging System: A Developer’s Guide.

Take Advantage of a Fully Integrated and Virtualized Environment

Oracle Solaris 11 is a completely virtualized operating environment, with technologies that span server, network, and storage virtualization, to let enterprise developers make the most of available resources. With these tools, enterprise developers can create minimized environments targeted to development, testing, and deployment processes, yet add in additional functionality whenever needed.

Create a Lightweight Development Environment

Keeping environments small and focused makes it simpler and safer to build applications and move them throughout the build, test, debug, and deploy process. Using IPS, enterprise developers can create a minimized operating environment inside an Oracle Solaris Zone. This lightweight development environment includes only essential operating system components and services. Tools and source code can be installed in the zone to create an application development stack that is compact and complete. The zone can be moved in its entirety to testing and production environments—that might or might not be located on the same server—with ease. Other Oracle Solaris technologies can be layered on top of the minimized environment as needed, giving developers access to greater functionality and more resources in a safe and controlled manner.

Manage Resources

Developers often have limited access to systems during application development and testing phases. Finding ways to share systems and ensure developers, tools, and applications have sufficient resources can be challenging. Oracle Solaris Zones provide fine-grained resource management controls to ensure applications do not starve one another when running on the same system. Developers can create collections of resources, such as CPUs, physical memory, and network bandwidth, which are reserved for exclusive use by an application or set of applications. Virtual environments provide fixed boundaries between workloads, ensuring each has access to a consistent set of resources regardless of resource usage on the rest of the machine. As a result, programmers can create separate environments for development and testing efforts to eliminate competition for resources, helping achieve predictable application and system performance.

Virtualize the Network

Oracle Solaris 11 brings network virtualization and resource control into the operating system architecture, instead of adding on layered functionality with heavy overhead and undue complexity. Within the operating system, network virtualization virtualizes the network stack and allows fine-grained monitoring and control to let applications take advantage of improved network performance. With parallelization of network workloads across multiple processor threads and cores, and the ability to provide dedicated bandwidth and resources to separate services and protocols without a performance penalty, developers can create high-performance multitiered applications on a single kernel.

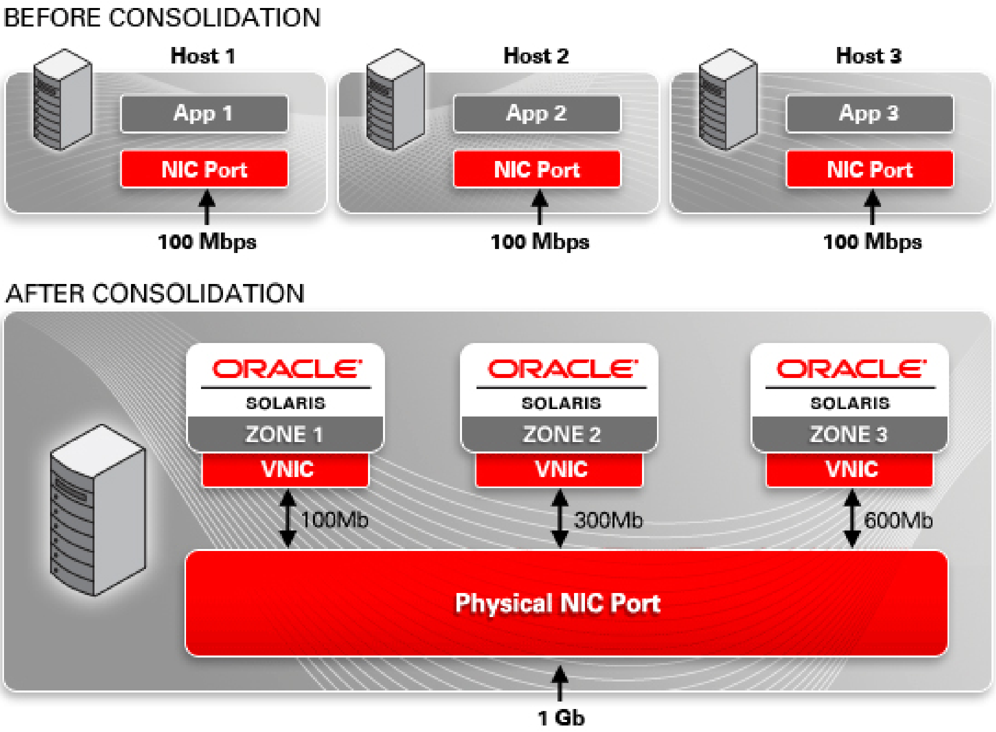


Figure 6. Network virtualization provides developers with the ability to model complex network topologies in developer environments without the restriction of expensive hardware.

- **Virtualize NICs.** A physical network card can be presented to applications as multiple virtual network interface cards (VNICs). Each VNIC acts like any networking device on which an IP interface can be plumbed. In Oracle Solaris 11, a VNIC is created automatically during the creation of Oracle Solaris Zones to allow users to benefit from an exclusive IP networking stack for each virtual environment.
- **Gain flexibility without sacrificing performance.** Network virtualization virtualizes the network stack and network interface card (NIC) around any service protocol, such as HTTP, HTTPS, FTP, and NFS, or virtual environments created with Oracle Solaris Zones or Oracle VM Server for SPARC. Each virtual stack can be assigned its own priority and bandwidth on a shared NIC without degrading performance.
- **Consolidate workloads.** Built-in network virtualization promotes more effective sharing of network resources and enhances the ability to consolidate server workloads. Using the basic building block of VNICs, virtual switches and interconnects, virtual LANs (VLANs), and built-in routing and firewall functionality, developers can consolidate an entire distributed computing environment on a single system for prototyping, testing, and deployment.

- **Control network usage.** Each VNIC can be assigned its own priority and bandwidth on a shared NIC. Traffic for one VNIC can be isolated from other traffic and assigned limits or guarantees on the amount of bandwidth it can use. Placing limits on bandwidth consumption improves network utilization and performance rates and supports operating system virtualization, utility computing, and server consolidation efforts.
- **Model networking environments.** Network virtualization gives developers the ability to model a complete data center networking topology in a virtual network, reducing limitations created by the availability of physical NICs.
- **Secure applications from attacks.** The architecture dynamically manages priority and bandwidth resources, and it can provide a better defense against denial of service attacks directed at a particular service or virtual machine by isolating the impact to that entity.

Expand the Storage Infrastructure

Enterprise applications often need to be tested with extremely large data sets to ensure information can be handled and throughput is sufficient. However, managing a large storage infrastructure can take time away from application development efforts. Oracle Solaris ZFS is now the default (root) file system in Oracle Solaris 11, and it is an enterprise-class, general-purpose file system designed to ease data and storage management complexity. Unlike traditional file systems that require a separate volume manager, Oracle Solaris ZFS integrates traditional file system functionality with built-in volume management techniques and data services.

Several important changes result from Oracle Solaris ZFS becoming the default file system:

- Oracle Solaris 11 now installs only on an Oracle Solaris ZFS root file system.
- Oracle Solaris ZFS eliminates the need for Solaris Volume Manager.
- Oracle Solaris ZFS is now the root file system. The UNIX File System (UFS) now is a legacy file system, but it is still available for use. Note that the `mkfs` and `newfs` commands remain available.

Integrated Volume Management

By breaking free of the typical one-to-one mapping between the file system and its associated storage, Oracle Solaris ZFS decouples the file system from physical storage in the same way that virtual memory abstracts the address space from physical memory, allowing for more efficient use of storage devices. Space is shared dynamically between multiple file systems from a single storage pool. Thousands of file systems can draw from a common storage pool, with each file system consuming only the disk space it needs. Physical storage can be added to storage pools dynamically—without interrupting applications and services—providing new levels of flexibility, availability, and performance. When capacity is freed by one file system in the pool, it becomes available to other file systems. As a result, developers no longer need to deal with the problems associated with partitioning, provisioning, wasted bandwidth, and stranded storage.

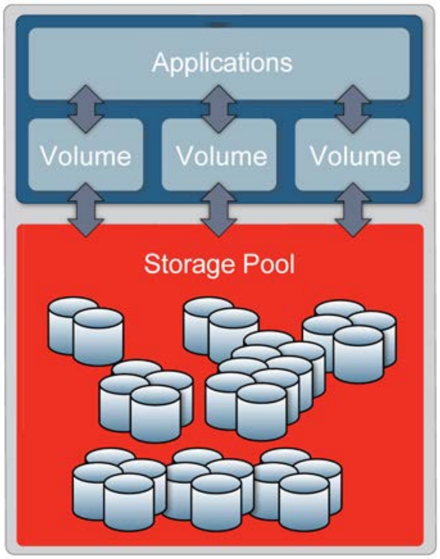


Figure 7. Virtual storage pools let multiple file systems share storage space.

Built-in Scalability and Data Integrity

Providing virtually unlimited storage (up to 21 billion yottabytes of capacity), Oracle Solaris ZFS scales to support large data sets. Several techniques, such as copy-on-write and end-to-end checksumming, keep on-disk data self-consistent and eliminate silent data corruption.

Data is written to a new block on the media before pointers to the data are changed and the write is committed. Because the file system is always consistent, time-consuming recovery procedures, such as `fsck`, are not required if the system is shut down in an unclean manner. In addition, data is checked when read to help ensure correctness, and any errors detected in a mirrored pool are automatically repaired to protect against costly and time-consuming data loss and (previously undetectable) silent data corruption. Corrections are made possible by a RAID-Z implementation that uses parity, striping, and atomic operations to help reconstruct corrupted data and reduce planned and unplanned downtime.

New Data Services

Oracle Solaris ZFS incorporates new data services in the Oracle Solaris 11 release that provide the following benefits:

- **Save space with deduplication.** Modern storage platforms use deduplication technology to reduce the amount of total data stored by eliminating and sharing common components. In Oracle Solaris 11, Oracle Solaris ZFS uses a checksum-based comparison of blocks with optional verification. When common blocks are found in the storage pool, only one copy is maintained, freeing the space associated with duplicate data. While deduplication is optional, it provides benefit in environments containing highly duplicated data, such as virtualized images, tools, home directories, e-mail folders, and more.

- **Protect data with encryption.** Oracle Solaris ZFS now includes support for encrypted data sets to provide data set-level, secured deletion and protection against physical storage theft and man-in-the-middle attacks on the SAN. Because data is encrypted at the data set level, users can mix encrypted and unencrypted data in the same Oracle Solaris ZFS storage pool for maximum flexibility.

I/O Performance Improvements with Flash Device Support

Flash technology is gaining in popularity. Oracle Solaris 11 includes support for solid-state disks (SSDs) to give developers the opportunity to put the right information on faster Flash devices to speed application performance. For example, developers can use SSDs to hold frequently accessed data to minimize the impact of disk latencies and improve application performance. By creating hybrid storage pools—storage infrastructure that uses enterprise Flash devices to handle certain types of I/O with dramatically reduced latency and hard disk drives for high capacity—applications can store and access massive data sets with high performance at lower cost.

Greater Storage Device Support

Developers can take advantage of the wide range of storage devices now available to meet capacity and performance demands for development and deployment systems—from smaller, low-cost systems to high-performance, high-capacity devices, and everything in between. Oracle Solaris 11 supports a variety of protocols and interface technologies and provides key host bus adapter (HBA) drivers with the software distribution.

- **Internet SCSI (iSCSI) target support.** Many storage deployments rely on the iSCSI protocol to move data across intranets and manage devices remotely. With iSCSI target support, the operating system can make SCSI storage devices available to clients over the network.
- **Fibre Channel.** Oracle Solaris 10 includes a number of Fibre Channel packages, including libraries based on the T11 FC-HBA specification, Emulex and QLogic device drivers, debugging aids, an FCIP IP/ARP over Fibre Channel device driver, a Fibre Channel transport layer, and much more.
- **HBA drivers.** A wide range of drivers are included in the operating system, including the Adaptec AdvanceRaid Controller SCSI HBA, Adaptec Ultra320, Advanced Host Controller Interface SATA, LSI MegaRAID SCSI HBA, LSI Hardware RAID HBA, LSI MegaSAS RAID Controller HBA, and Marvell 88SX SATA device drivers.
- **Multiple protocol support.** Several protocols are supported by the operating system, including Serial Management Protocol for Serial Attached SCSI (SAS), which provides a way to communicate with SAS expanders through the Serial Management Protocol (SMP). An SMP target driver lets developers issue SMP requests and receive SMP responses through an interface, giving privileged users the ability to configure and manage SAS domains.

Integrate Storage Management into Enterprise Applications

Many enterprise applications have unique requirements that make it necessary to integrate storage management into the application. However, connecting software components in the data center requires intelligent storage interconnectivity. Developers can take advantage of Oracle Solaris ZFS and the Common Multiprotocol SCSI Target (COMSTAR) framework included in Oracle Solaris 11 to create in-house solutions that are tailored to enterprise storage needs.

Block-based devices are prevalent in storage infrastructures, and many enterprise applications are designed specifically to work with these systems. However, a trend is on to move toward object- and file-based access and utilize virtualization and abstraction techniques for better utilization and simplified management and access as data volumes grow. Yet traditional block-based storage continues to serve these new data access methods.

Supporting both block-based and file-based access, the COMSTAR framework lets developers transform any server that is running Oracle Solaris 11 into a target storage device that can be accessed by initiator hosts over the network (Figure 6). What makes the COMSTAR framework revolutionary is its ability to employ an Oracle Solaris ZFS file system on a target storage device and combine it with off-the-shelf components to create a storage server. All SCSI devices types, including disk and tape, can connect to a transport with concurrent access to all logical unit numbers (LUNs) and a single point of management.

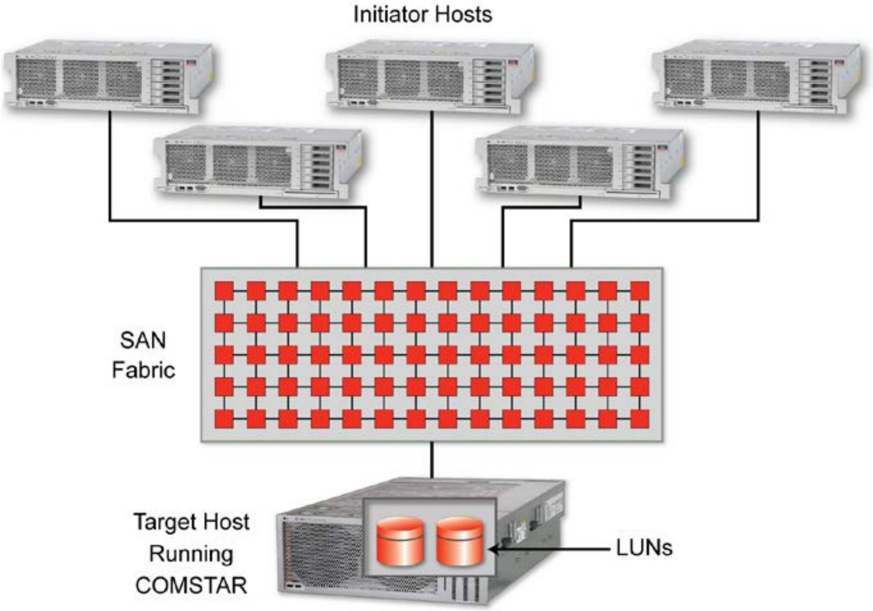


Figure 8. A target host running the COMSTAR framework can be accessed by initiator hosts over a SAN fabric.

COMSTAR provides a single framework for all target storage devices, so enhancements can be made in one place. Developers do not need to update each architecture independently whenever problems arise or new features are implemented, and any port can access any LUN. Each COMSTAR target device contains several components.

- Logical unit (LU) providers implement the functions associated with a SCSI LUN.
- Port providers provide local ports or connection points, such as HBAs and NICs, for transports.
- A SCSI target mode framework (STMF) manages common block storage functionality, including contexts and resources for SCSI command execution, LUN mappings, management of logical unit and port providers, abnormal command termination, and more.
- A management library (`libstmf`) communicates with and configures the COMSTAR framework.

Oracle Solaris 11 adds support for several protocols:

- Internet SCSI (iSCSI) protocol and iSCSI Extensions for RDMA (iSER) protocol
- SCSI RDMA Protocol (SRP), for hosts that include an InfiniBand host channel adapter
- Fibre Channel over Ethernet (FCoE) protocol

In addition, DTrace probes have been added to COMSTAR in the SCSI Target Mode Framework (STMF) and SCSI Block Device (SBD) code.

Work in Heterogeneous Environments

Developers often need to work in, or create applications for, heterogeneous environments. Whether data is located on storage systems connected directly to a server, across a local or remote enterprise network, or on an external Website, one or more protocols likely is needed to make the data available to applications. Oracle Solaris 11 supports a number of block- and file-level protocols, such as iSCSI, Network File System (NFS), SAMBA, and Common Internet File System (CIFS).

Native CIFS support simplifies data access by Microsoft Windows virtual machines. By incorporating an in-kernel CIFS/ Server Message Block (SMB) implementation, the operating system provides a rich set of permission and identity mapping capabilities that allows transparent file access—even to files on the same share—from Microsoft Windows or UNIX virtual machines. By simplifying Microsoft Windows and UNIX interoperability and seamlessly mapping file system credentials, applications, storage servers, and appliances can be integrated more easily into existing heterogeneous environments.

The Oracle Solaris 11 CIFS service includes several new features, including the following:

- Host-based access control, allowing CIFS servers to restrict access to specific clients by IP address
- Access control lists (ACLs) on shares
- Client-side caching of offline files and synchronization when reconnected
- Complete Microsoft Windows file sharing interoperability that pairs with Microsoft Active Directory integration

Secure Applications, Services, and Systems

Oracle Solaris 11 integrates key security features that enterprise developers can leverage to create more secure production environments and applications. More information can be found in the Developer's Guide to Oracle Solaris Security.

- **Secure the environment.** Oracle Solaris ZFS allows the creation of a read-only copy of a file system, including the root file system. Developers can deploy a read-only root file system in an Oracle Solaris Zone to lock down the environment for added security.
- **Encrypt information.** The ubiquity of networked computing and the growing demand for multimedia and rich Internet applications and Web services are pushing developers to implement data encryption capabilities into applications to protect valuable information as it flows across the network. Oracle Solaris 11 includes a cryptographic framework that provides application-level and kernel-level cryptographic operations. Based on the PKCS#11 public key cryptography standard, the framework brings the power of advanced, streamlined encryption algorithms and hardware acceleration in the SPARC T4 processor to user-level C and Java programming language-based applications. New cryptographic enhancements in Oracle Solaris 11 include support for FIPS 140-2 of the Federal Information Processing Standard and the implementation of ECC and other NSA Suite B protocols to meet stringent government standards.
- **Secure data.** Oracle Solaris ZFS now uses the cryptographic framework built into the operating system to enable cryptographic protection of data on a per-data set basis.
- **Create secure by default environments.** Oracle Solaris 11 network services are disabled by default, or set to listen only for local system communications, to limit opportunities for unauthorized access.
- **Audit access.** The Oracle Solaris audit feature provides the ability to log system activity for any auditable Oracle Solaris event—such as system calls on the server machine, packets sent over the network, or a sequence of bits written to disk—at a granular level. As of Oracle Solaris 11, auditing is a service and auditing records are stored in binary files on an Oracle Solaris ZFS file system. Developers do not need to reboot systems to enable auditing. The /etc/security/audit_user and /etc/security/audit_control files, and the bsmconv/unconv script, no longer are used.
- **Set sensitivity levels.** Oracle Solaris includes support for Trusted Extensions, an advanced security feature that implements labels to protect data and applications based on their sensitivity level, not just by who owns or runs them. Labeled objects have an explicit relationship with each other, and an application usually cannot see or access data with a different security label. Oracle Solaris 11 includes two significant enhancements to Trusted Extensions. Security labels now can be created on Oracle Solaris ZFS data sets for explicit mandatory access control (MAC) policies. In addition, the existing trusted networking protocol assumes the underlying network is secure and that packet headers cannot be manipulated or observed in transit. Oracle Solaris 11 introduces labeled IPsec, enabling sensitivity labels to be associated with network traffic. Traffic with different sensitivity levels can be isolated and contained, and labeled networking can take place over untrusted networks.

- **Protect Web-facing applications.** With pressure on to create and deploy more network services faster, developers sometimes are forced to skip or shorten the testing process, making these applications targets for hackers looking to deface Websites or steal information. Developers can use the access control mechanisms built into Oracle Solaris 11 to protect Web-facing applications. These features—such as Oracle Solaris Zones, User and Process Rights Management, and the Oracle Solaris Service Manager—can be used to separate data and Web servers and limit access (Figure 7). Data is hosted in a zone and connected to the corporate intranet or LAN. Another zone holds the Web server and is configured with a reduced set of privileges. While the Web zone is accessible to the public internet through a firewall, it has read-only access to the files contained in the data zone to help protect HTML and data files. As a result, intruders that hack into the system are unable to modify the IP address or corrupt other system data.

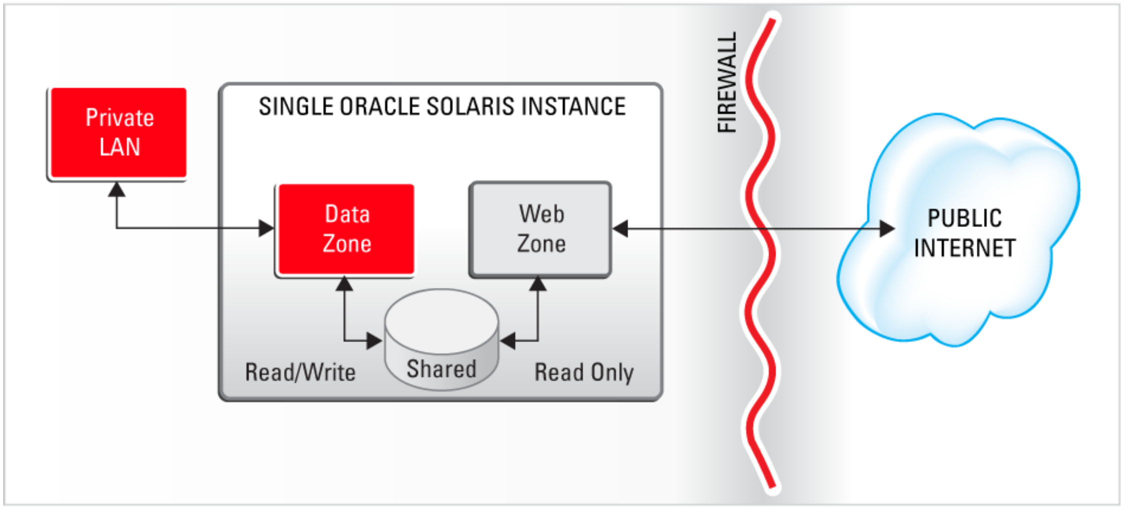


Figure 9. Oracle Solaris 11 technologies can be used to prevent Web-based hijacking.

Port and Migrate Applications to Oracle Solaris 11

Porting applications from one environment to another can be a lengthy and expensive task. Continuing over a decade of tradition, Oracle preserves the long-standing guarantee of binary compatibility—applications that run on previous Oracle Solaris releases can run unchanged on Oracle Solaris 11 within the same processor architecture: x86 or SPARC. In addition, the Oracle Solaris Source Code Guarantee provides assurance that C and C++ applications developed and compiled to run on SPARC or x86 platforms will compile and run on either of these platforms. Together, these compatibility guarantees help ensure applications run from one Oracle Solaris platform to the next, lower development and testing costs, and speed time to deployment.

Developers porting applications from other UNIX environments, such as Linux, should consider the following:

- Placing tools in similar locations in the new environment can help minimize changes to source files.
- Source files should be modified to use Oracle Solaris tools, utilities, and libraries, and care should be taken to ensure the options that are used provide the intended functionality.
- Keep in mind that some compiler options might have the same name but behave differently in the old and new environments. Similarly, some options might have the same functionality but different names in the two environments.
- Determine whether any APIs used by the application are incompatible with Oracle Solaris 11. Inline source code changes can be made; however, Oracle recommends creating a compatibility library that implements the changes needed to resolve issues. Modifications to the source code can be limited to conditional compile directives, ensuring backward compatibility.
- A number of new routines are included in the Oracle Solaris C library to improve familiarity with Linux and BSD operating systems and help reduce the time and cost associated with porting applications to Oracle Solaris 11. Examples of new routines include `asprintf()`, `vsprintf()`, `getline()`, `strdupa()`, `strndup()`, `ascftime()`, and `fnmatch()`. The introduction of /usr/include/paths.h for definitions of pathnames used in Oracle Solaris and /usr/include/sys/paths.h for definitions of pathnames used in the Oracle Solaris kernel help developers familiarize themselves with well-known locations for application porting.
- Changes to source code might be required as a result of the semantic and syntactic differences in how Oracle and other vendors implement the C, C++, and Fortran languages. Oracle's compilers conform to the American National Standard for Programming Languages—C (ANSI/ISO 9899-1990) and also support traditional K&R C.

Developers who are migrating applications across to Oracle Solaris 11 should take note of some of the key differences in environment when compared to Oracle Solaris 10. Table 3 summarizes these key changes.

TABLE 3. ORACLE SOLARIS 10 AND ORACLE SOLARIS 11 DIFFERENCES SUMMARY

COMPONENT	ORACLE SOLARIS 10	ORACLE SOLARIS 11
Hardware	SPARC and x86	SPARC and x86
Distribution Media	Six CDs or one DVD	One CD/DVD; different installation choices available. (Installs as a minimal OS. Use the package repository to add additional components.)
Default User Shell	/usr/bin/sh	/usr/bin/bash
Default User Path	/usr/bin	/usr/bin
Default Desktop	GNOME (JDS)	GNOME
Default X Server	Xsun	Xorg
Default File System	UFS	ZFS
Additional Software	Companion CD	Available in the IPS network repository
CD Path	/cdrom	/media
Upgrade Mechanism	Oracle Solaris Live Upgrade	Image update
Binary Compatibility	Guarantee	Guarantee
Root Login	Allowed	Role-based access control (RBAC). (Can be reverted to a root account if required, but that is not recommended.)

Conclusion

Long considered the mission-critical operating system, Oracle Solaris is the platform of choice for enterprise developers. Oracle Solaris 11 builds on this foundation to deliver an integrated and virtualized environment that facilitates the enterprise application development process. With a wide range of scalability, efficiency, security, and availability enhancements, Oracle Solaris 11 makes it easier than ever to create and deploy reliable application stacks with greater efficiency.

For More Information

For more information on Oracle Solaris 11, including developer documentation on the Oracle Technology Network (OTN), see the references listed in Table 4.

TABLE 4. REFERENCES FOR MORE INFORMATION ABOUT DEVELOPING ON ORACLE SOLARIS 11

Oracle Solaris 11	http://www.oracle.com/solaris
Oracle Solaris 11 information on OTN	http://www.oracle.com/technetwork/server-storage/solaris11/overview/index.html
Oracle Solaris Studio information on OTN	http://www.oracle.com/technetwork/server-storage/solarisstudio/overview/index.html
Oracle Solaris 11 documentation	http://www.oracle.com/technetwork/server-storage/solaris11/documentation/index.html
Oracle Solaris developer resources	http://wwww.oracle.com/technetwork/systems/solaris-developer/index.html
Oracle Solaris Studio documentation	http://www.oracle.com/technetwork/server-storage/solarisstudio/documentation/index.html
Oracle Solaris 11 support	http://www.oracle.com/us/support/software/premier/index.html

ORACLE

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

Oracle Solaris 11: What's New for Application Developers
October 2012, Revision 1.2

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Hardware and Software, Engineered to Work Together