



# Solaris 10

## A Technical Overview ...

Markus Halter  
Technical Consultant  
Sun Microsystems



# Agenda



## Utilization & Management

- N1 Grid Containers or Zones
- Dynamic Resource Pools
- Fair Share Scheduler
- Resource Controls
- Unified Patch/Package DB
- Extended Patch/Package Utilities
- System Management Agent

## Data Path

- ZFS a new approach
- SVM enhancements
- SAN Boot
- Multi-Terabyte support
- NFSv4
- QFS/SAMFS
- IB support

## RAS Features

- Dynamic Tracing
- Fault management
- Service Awareness

## Performance

- SysCall enhancements
- Library enhancements
- Memory Placement
- Dynamic TSB
- SEGMAP
- CMT Support
- Multiple Page Sizes
- SSE2
- AMD64
- Fireengine
- Clearwater

## Security

- Process Rights Management
- Role Based Access Controls
- Solaris Zones
- Basic Auditing and Reporting
- Crypto Framework
- IP Filter
- SASL Library

## Installation & Upgrade

- WAN Boot
- Mirrored Root
- Multiple Network IF's
- Live Upgrade with SVM
- Suninstall on DVD

## Solaris Adoption

- Binary Compatibility Guaranteed
- Compatibility Toolset
  - APPCERT
  - Application Scanner

## Development

- Process Model Unification
- Removal of Static Libraries
- Removal of statically linked Tools
- Event Port Framework
- uDAPL

Once upon a time ...

# Solaris Development & Life Cycle

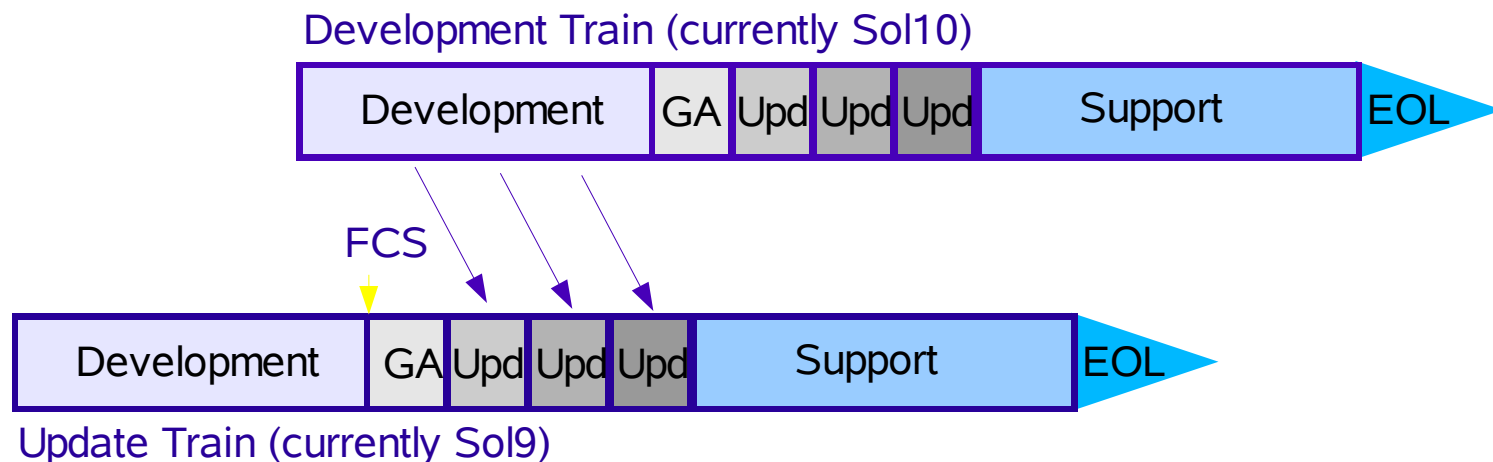
# Solaris Development Model

- Product support lifecycle designed for customers' needs
  - Releases extend, rather than replace
  - Guaranteed compatibility
- Extensive testing
  - Continuous, measurable quality gains
  - Solaris 10 OS already deployed internally



# Solaris Release Roadmap

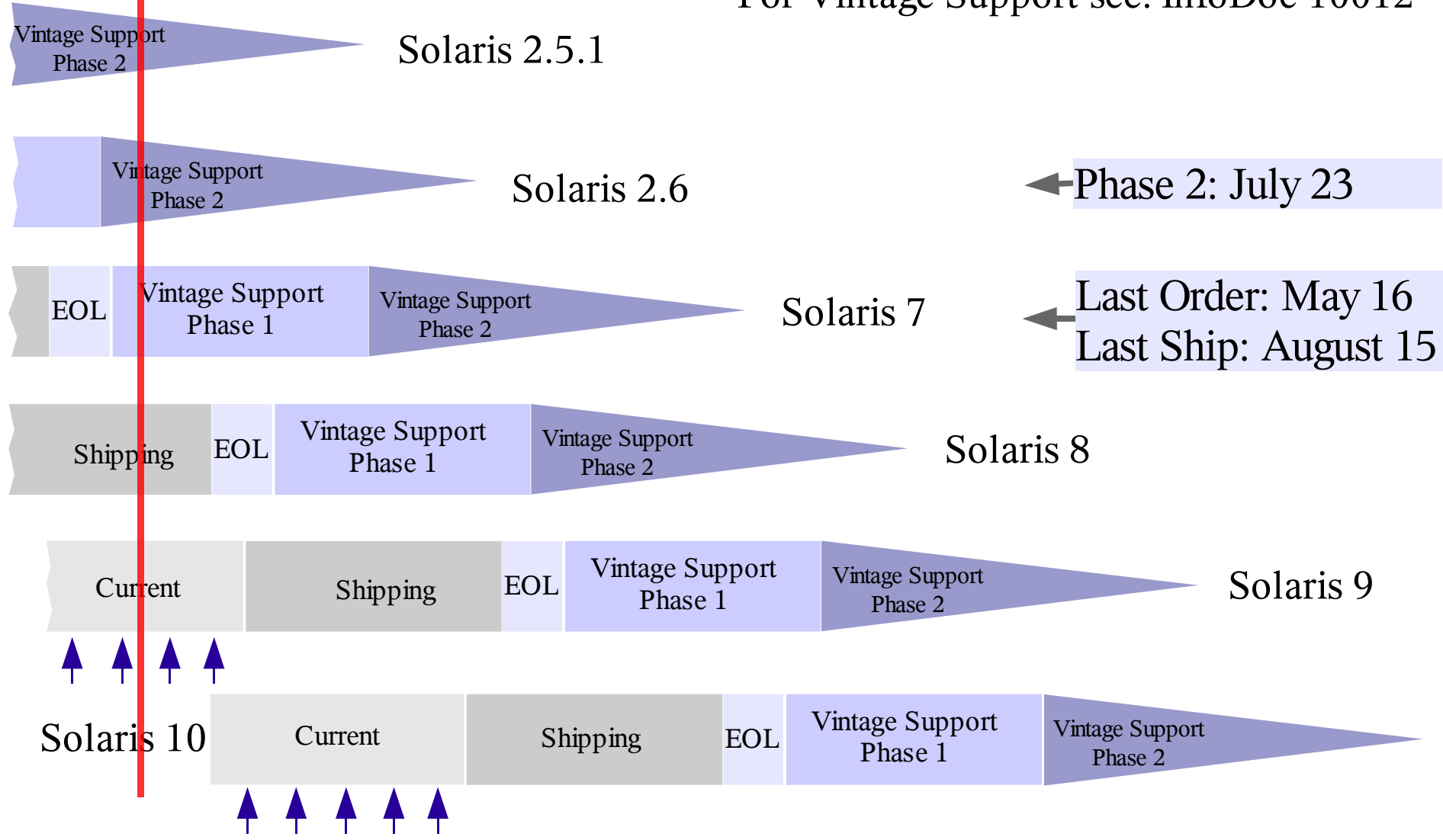
- Named release planned every 2-3 years
- Approximately four updates per year
- Minimum shipping life: 4-1/2 years
- Minimum support life: 9-1/2 years



# Solaris Release Roadmap

*April 2004*

For Vintage Support see: InfoDoc 10012



# Utilization & Management

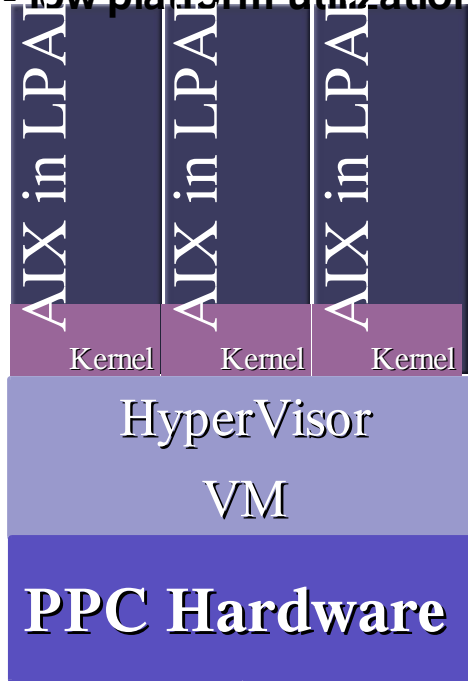
# N1 Grid Containers or Zones

- Virtualization of OS services
  - Zone users have the illusion of exclusively running on their own system
    - Are not able to see other Zone's processes
    - Are not able to observe other Zone's network traffic
    - Have complete node configuration and management capabilities
    - Can be rebooted without affecting other Zone's
- Sharing of all hardware resources
  - Optimal platform for consolidation for optimal system utilization

# Partitioning vs. Zones

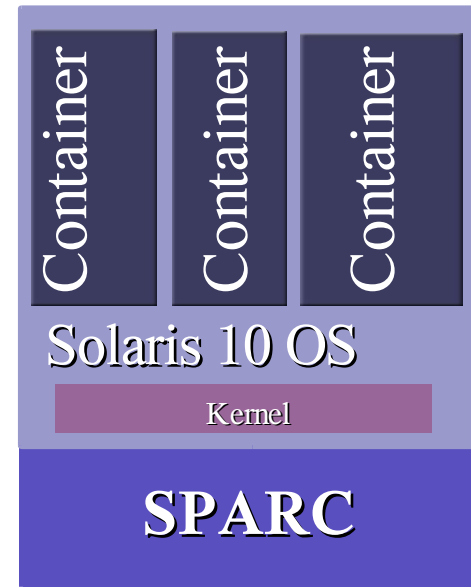
## Many OS instances

- no shared resources
- low platform utilization

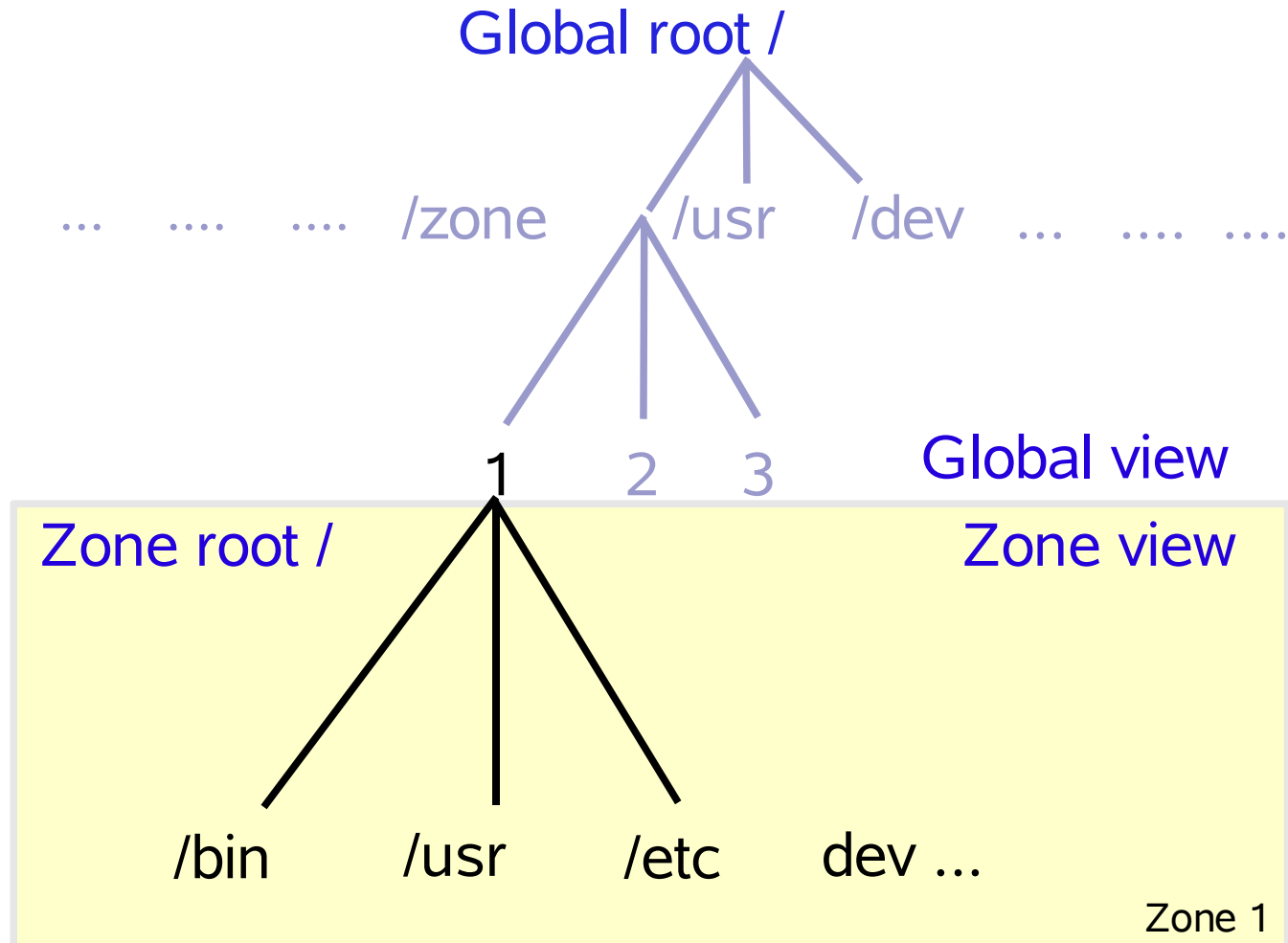


## Single OS instance

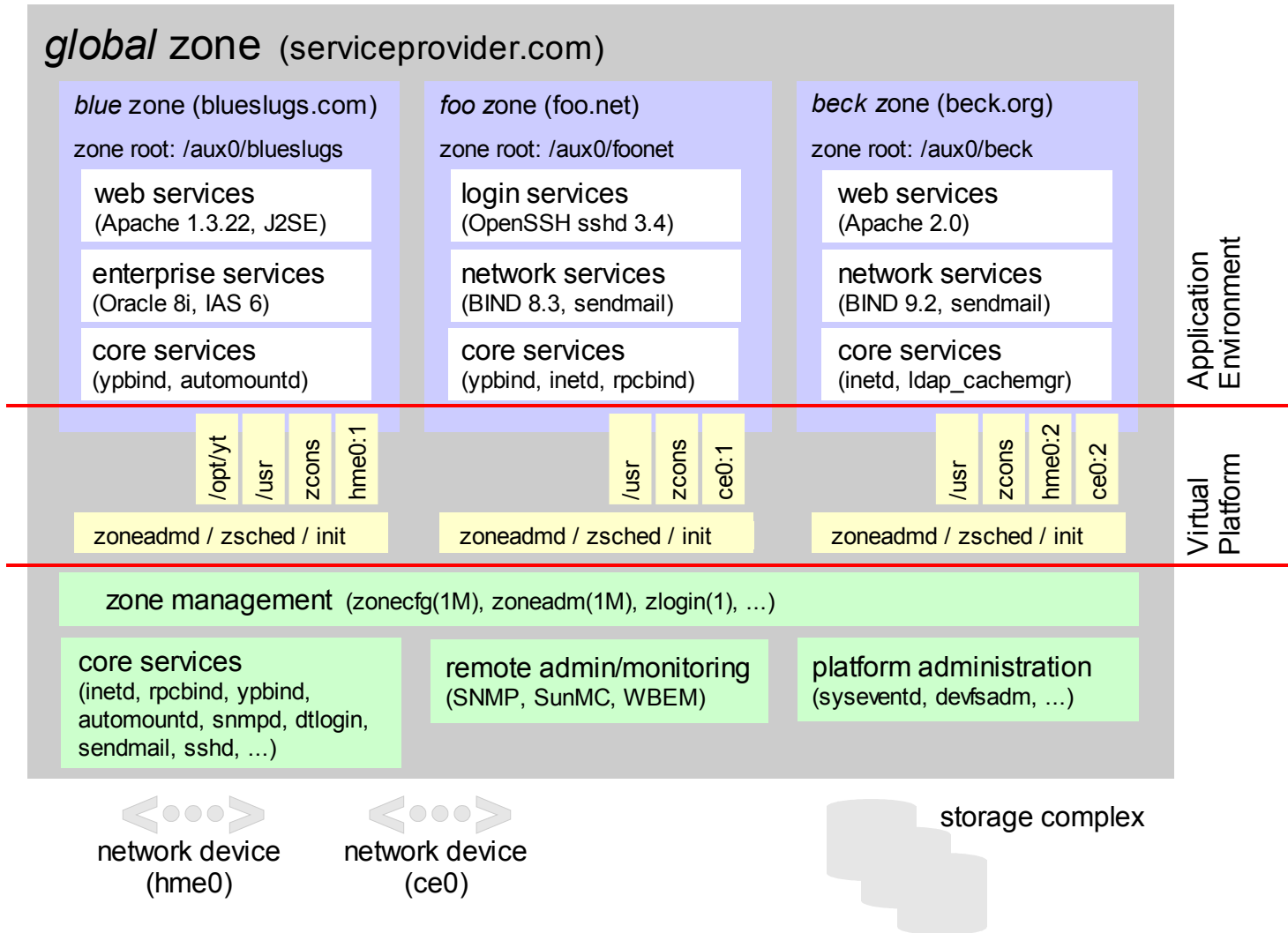
- resources are shared
- managed resource sharing
- high platform utilization



# Zone File-System Isolation



# Zones Block Diagram



# Zones Summary

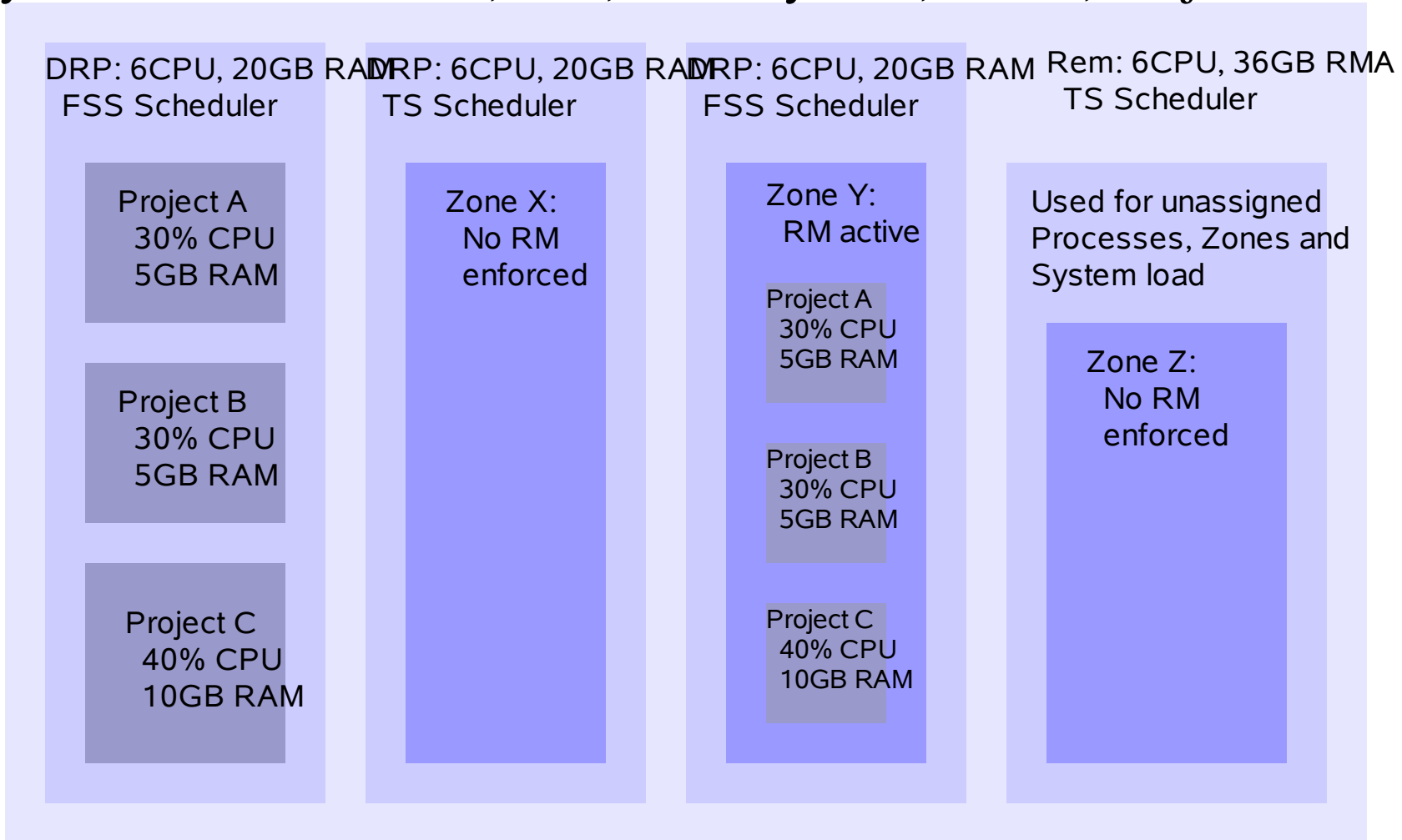
- Optimal platform for consolidation
  - Increase system utilization with minimal overhead
  - Provides almost arbitrary granularity in isolating and sharing resources
- Significant increase in uptime and security
  - Faults and intrusions are isolated
  - Rebooting Zone's is a matter of seconds
- Easy to create, replicate and maintain
  - `zoneadm(1M)`, `zonecfg(1M)`, `zlogin(1)`, `zonename(1)`
- Compatible application environment

# Resource Management

- Manages and controls system resource usage
  - CPU, Memory and Network bandwidth
- Allows for logically splitting a system into multiple Dynamic Resource Pools
  - Dynamically adjusts the number of CPU's based on workload goals on single CPU granularity
- Gives fine granular control over system resources through FSS, IPQoS and RCAPd

# Resource Management

Dynamic Resource Pools, FSS, Memory Set's, RCAP, Projects



System with 24 CPU's & 96GB RAM

# Dynamic Resource Pools

- A collection of system resources
  - A set of CPU's expressed as `min` and `max` values
  - An amount of memory – Memory Set(Solaris 10 update)
  - Manage through: `poolcfg(1M)`, `pooladm(1M)`
- Dynamically adjusts resource allocations in response to system events and load changes to preserve specified performance goals
  - Based on system **Load** or **Utilization**

# FSS and Memory Capping

- Fair share Scheduler
  - Allows for control of CPU resources based on shares
  - Sub-CPU granularity on two levels of control  
Zone -> Project
  - `rctladm(1M)`, `prctl(1M)`, `zonecfg(1M)`
- Memory Resource Capping
  - Allows for physical memory control
  - Violation of defined working set limits starts page-out activity for a particular project

# Resource Controls

- Used to control resource limits

```
rctladm(1M), prct(1M),
project(4)
```

- Limits are managed on different levels
  - Zone, Project, Task and Process
- NEW: SYS V IPC Resource Controls
  - Obsoletes need to tune via `/etc/system`
  - Default values fit most applications

```
process.max-port-events
process.crypto-buffer-limit
process.max-crypto-sessions
process.add-crypto-sessions
process.min-crypto-sessions
process.max-msg-messages
process.max-msg-qbytes
process.max-sem-ops
process.max-sem-nsems
process.max-address-space
process.max-file-descriptor
process.max-core-size
process.max-stack-size
process.max-data-size
process.max-file-size
process.max-cpu-time
task.max-cpu-time
task.max-lwps
project.max-device-locked-memory
project.max-port-ids
project.max-shm-memory
project.max-shm-ids
project.max-msg-ids
project.max-sem-ids
project.cpu-shares
```

# Unified Package / Patch DB

- Unified Package and Patch Database
  - SQLite based relational database
- Improved performance for typical patch work
- Ability to revert back to flat file contents for compatibility
- Extended `pkgchk(1M)` for package content list and query
- New admin command for database management - `pkgadm(1M)`

# Extended Package & Patch Utils.

- Built in support for signed packages & patches
- Ability to download signed packages & patches via HTTP protocol
- HTTP Proxy support
- New admin command for certificate management - `pkgadm(1M)`

# System Management Agents

- SNMP agent based on Open-Source NET-SNMP V5.09
  - See <http://www.netsnmp.org>
- Full framework including development kit
- Supports SNMPv1, SNMPv2C & SNMPv3
  - Authentication, privacy and access control
- Various MIB's including Host-Resource-MIB
- SMA proxy use to enable legacy SEA agent

# Data Path

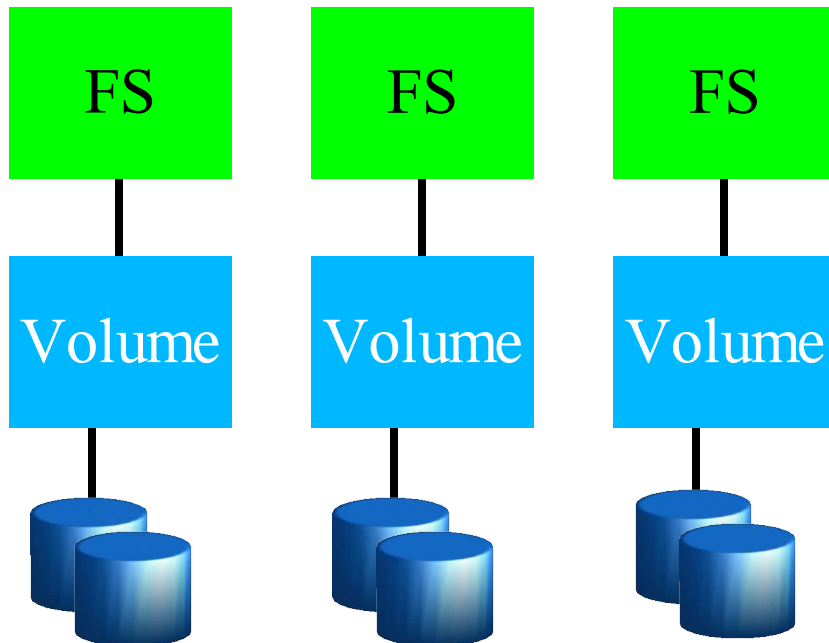
# ZFS – Zetta-Byte File-System

- Immense capacity (128bit)
- Infinite Constant-time snapshots
- Dynamic striping across disks
- All operations are copy-on-write
  - Never overwrite live data
  - On-disk state always valid
  - No need for `fsck(1M)`
  - Write sequentialization
- All data is checksummed
  - No silent data corruption
  - No panics on bad metadata

# ZFS Pooled Storage Model

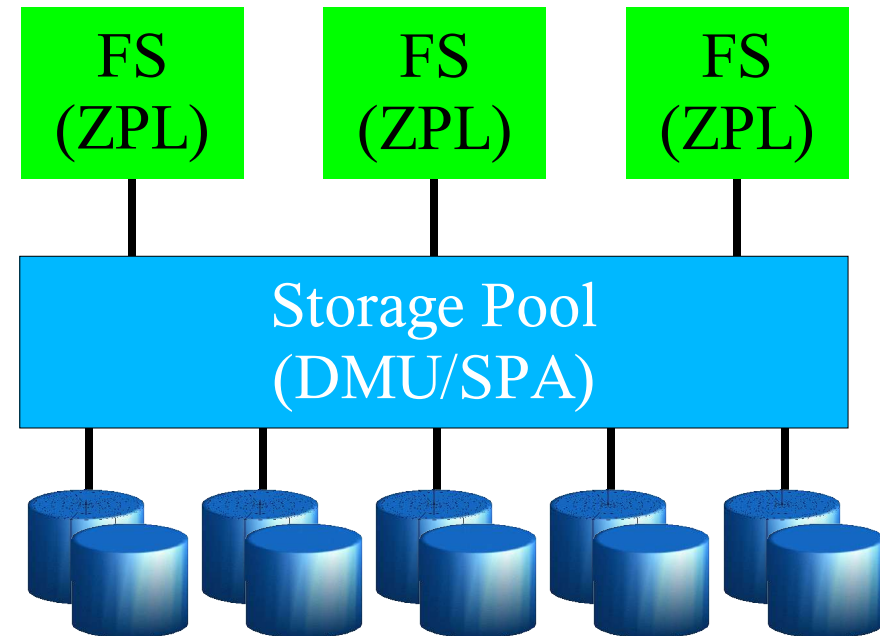
## • Traditional volumes

- Partition per filesystem; painful to manage
- Block-based FS/Volume interface slow, brittle



## • Pooled Storage

- Filesystems share space; easy to manage
- Transactional ZPL/DMU interface fast, robust



# ZFS Management

```
# format
... (long interactive session omitted)

# metadb -a -f disk1:slice0 disk2:slice0

# metainit d10 1 1 disk1:slice1
d10: Concat/Stripe is setup
# metainit d11 1 1 disk2:slice1
d11: Concat/Stripe is setup
# metainit d20 -m d10
d20: Mirror is setup
# metattach d20 d11
d20: submirror d11 is attached

# metainit d12 1 1 disk1:slice2
d12: Concat/Stripe is setup
# metainit d13 1 1 disk2:slice2
d13: Concat/Stripe is setup
# metainit d21 -m d12
d21: Mirror is setup
# metattach d21 d13
d21: submirror d13 is attached

# metainit d14 1 1 disk1:slice3
d14: Concat/Stripe is setup
# metainit d15 1 1 disk2:slice3
d15: Concat/Stripe is setup
# metainit d22 -m d14
d22: Mirror is setup
# metattach d22 d15
d22: submirror d15 is attached
```

```
# newfs /dev/md/rdisk/d20
newfs: construct a new file system /dev/md/rdisk/d20: (y/n)? y
... (many pages of 'superblock backup' output omitted)
# mount /dev/md/dsk/d20 /export/home/ann
# vi /etc/vfstab ... while in 'vi', type this exactly:
/dev/md/dsk/d20 /dev/md/rdisk/d20 /export/home/ann ufs 2 yes -

# newfs /dev/md/rdisk/d21
newfs: construct a new file system /dev/md/rdisk/d21: (y/n)? y
... (many pages of 'superblock backup' output omitted)
# mount /dev/md/dsk/d21 /export/home/ann
# vi /etc/vfstab ... while in 'vi', type this exactly:
/dev/md/dsk/d21 /dev/md/rdisk/d21 /export/home/bob ufs 2 yes -

# newfs /dev/md/rdisk/d22
newfs: construct a new file system /dev/md/rdisk/d22: (y/n)? y
... (many pages of 'superblock backup' output omitted)
# mount /dev/md/dsk/d22 /export/home/sue
# vi /etc/vfstab ... while in 'vi', type this exactly:
/dev/md/dsk/d22 /dev/md/rdisk/d22 /export/home/sue ufs 2 yes -

# format
... (long interactive session omitted)
# metattach d12 disk3:slice1
d12: component is attached
# metattach d13 disk4:slice1
d13: component is attached
# metattach d21
# growfs -M /export/home/bob /dev/md/rdisk/d21
/dev/md/rdisk/d21:
... (many pages of 'superblock backup' output omitted)
```

# ZFS Management

- Create a storage pool named “home”  
`# zpool create "home" mirror(disk1,disk2)`
- Create filesystems “ann”, “bob”, “sue”  
`# zfs mount -c home/ann /export/home/ann`  
`# zfs mount -c home/bob /export/home/bob`  
`# zfs mount -c home/sue /export/home/sue`
- Later, add space to the “home” pool  
`# zpool add "home" mirror(disk3,disk4)`

# Data Technologies

- Multi-Terabyte Support
  - `sd(7d)` / `ssd(7d)` disk drivers, SVM, and UFS all now support 1TB or greater volumes
- UFS logging performance boost on SPECsfs
  - on by default now
- SVM Enhancements
  - Volume Configuration Assistant
  - Disk-Set import
  - Cluster Volume Management
  - Support for Dynamic Reconfiguration

# Data Technologies

- Support for NFSv4 - `nfs(4)`
- General NFS enhancements - `mount_nfs(1M)`
  - RDMA (IB) support eliminates TCP/IP overhead
  - Parallel write support eliminates POSIX writer lock
  - Network transport enhancements – `mss 1MB`
- STMS (MPxIO) integrated into Solaris 10
  - Support for booting off MPxIO devices and Power Management of MPxIO-managed devices  
`stmsboot(1M)`

# Data Technologies

- devfs – Virtual Device File-System
  - Improved boot time through on demand device attachment during FS lookup
  - Device nodes in `/devices` are no longer `specfs` nodes in `ufs`
  - `libdevinfo` & `prtconf` may not show all devices
- USB 2.0 support
  - Audio, Serial, Printers, Hard-Disk, Floppy, Flash-Devices, Card-Readers
  - USB generic driver for “userland device drivers”

# Performance

# General Service Latency

- System Call enhancements
  - Speed up many commonly used system calls, some by as much as 15x, with at least 25% improvement  
`dup`, `fcntl`, `flock`, `getsockname`, `getpeername`,  
`gettimeofday`, `lseek`, `select`, `semop`, `setcontext`,  
`setsockopt`, `sigaction`, `siglongjmp`, `signal`,  
`sigprocmask`, `socket`, `time`, `times`
- Library Call enhancements
  - Significant speedups in many frequently used routines  
`strftime`, `mktime`, `localtime`, `getenv` and Sparc  
`str*`

# General Networking Performance

- **Project FireEngine**
  - new merged TCP/IP stack
  - Improved throughput, better scalability and reduced connection setup times
  - Affected system calls include  
`bind`, `accept`, `socket`, `connection`, `listen`, ...
- **Project Clearwater**
  - Get rid of the STREAMs based networking stack in favor of a procedural implementation
- **Improved UNIX Domain Sockets**

# Platform Specific Performance

- Memory Placement Optimization<sub>SPARC/AMD64</sub>
  - New lgroup functions in `liblgrp(3lib)` for more detailed control by applications on NUMA plat.
- Multiple Page-Size support<sub>SPARC</sub>
  - Enables page-size control for Heap, Stack and Anon segments – `ppgsz(1)`, `pagesize(1)`
- Dynamic TSB<sub>SPARC</sub>
  - Per-process TSB's , to handle 100k+ busy processes

# Platform Specific Performance

- Support for multithreaded CPU's<sub>SPARC/x86</sub>
  - CMP and SMT optimizations – SPARC IV, Niagara, Xeon
- Integrated usage of x86 SSE2 SIMD<sub>AMD64/x86</sub> instructions in `libc(3lib)`
- Support for AMD64 architecture<sub>AMD64</sub>
  - Includes ability to run 32-bit and 64-bit applications simultaneously
  - Smooth migration path from 32-bit to 64-bit

# RAS Features

# Service Management Facility

- Supply a mechanism to formalize relationships between services
- Provide a unified repository for configuration of service startup behavior
- Allow Solaris to start and restart services automatically over the lifetime of a Solaris instance
- New interfaces for administration  
`svcadm(1M)` , `svccfg(1M)` , `svcs(1)`

# Predictive Self Healing

- New software architecture and methodology for fault detection, aggregation and recovery
  - Proactive offline before failure
  - Automatic service restart in cooperation with SMF
  - Simplified error reporting
  - Diagnosis & mitigation in milliseconds, not hours

# Dynamic Tracing

- Today's observation technologies for transient failures are not production ready
  - Post mortem analysis requires system failure
  - Running instrumented binaries requires reboot
- Need to be able to dynamically instrument OS with zero impact if disabled
  - Dynamically instrument the system to record arbitrary data
  - Instrumentation must be completely save

# Dynamic Tracing

- Probes
  - A probe is a point of instrumentation
  - A probe is made available by a provider
- Providers
  - A provider represents a methodology for instrumenting the system
  - Providers: FBT, syscall, lockstat, ...
- Consumers
  - A consumer is a process that interacts with DTrace
  - `dtrace(1M)` is a generic consumer

# Dynamic Tracing

Simple example for aggregation of a systems SysCall ratio

```
bash-2.05b# dtrace -n 'syscall:::entry{ @syscalls[probfunc] = count(); }'
dtrace: description 'syscall:::entry' matched 229 probes
^C
```

fstat	1
schedctl	1
lwp_park	1
.	
.	
.	
pread	206
ioctl	345
write	106836
read	106838

```
bash-2.05b# dtrace -n 'syscall:::entry{ @syscalls[execname] = count(); }'
dtrace: description 'syscall:::entry' matched 229 probes
^C
```

sac	6
ttymon	9
automountd	12
.	
.	
.	
nscd	112
prstat	290
dtrace	393
dd	221041

# Security

# Trusted Solaris Security

- Merged some features of TSol into base Solaris 10
  - Access rights management - `rbac(1)`
  - Process rights management — `ppriv(1)`, `getdevpolicy`
    - Least Privileges reduces the need for setuid applications
- N1 Grid Containers (Zones) to model secure compartments — `zoneadm(1M)`, ...
- IP-Filter built in for secure network access  
`ipf(1M)`, `ipfstat(1M)`, `ipnat(1M)`

# Security enhancements

- Solaris Encryption Framework
  - General crypto provider - `cryptoadm(1M)`
  - HW Acceleration and Key Storage SCA4000
  - Integrated with IKE, IPsec, SSH, GSSAPI, Kerberos
- Basic auditing & Reporting Tool – BART
  - Tripwire like content security - `bart(1M)`
- Password History
  - Definable number of history slots - `passwd(1)`
  - Used within `/etc/passwd` only – LDAP already supports password history

# Installation

# JumpStart

- Install using WAN-Boot
  - Replace RARP, BOOTPARAMS and NFS by secure protocols
  - Add a RAM-disk for loading and booting mini-root  
`ramdiskadm(1M)`
- Patches form JumpStart profile
- Packages form JumpStart profile – also non-OS

# SysID Tool

- SysID Tool extended to support for
  - Mirrored Root installation
  - Multiple network interfaces
  - LDAPv2 profiles
- New Meta-Cluster for minimization –  
SUNWrnet
- WebStart CLI replaced by `suninstall` on  
DVD and install CD

# Development

# Process Model Unification

- Get rid of different process models for single and multi-threaded applications
- Merged `libthread(3lib)` and `libpthread(3lib)` into `libc(3lib)`
- No more special linking required for multi-threaded applications
- EOL of `_lwp_create(2)`, `_lwp_detach(2)`, `_lwp_exit(2)`, `_lwp_getprivate(2)`, `_lwp_makecontext(2)`, `_lwp_setprivate(2)`, `_lwp_wait(2)` system calls

# libumem & Atomic Operations

- Fast, scalable object-caching memory allocation with MT support - `libumem(3lib)`
  - Provides extensive debugging support including
    - detection of memory leaks
    - buffer overruns
    - multiple frees
    - use of uninitialized data
    - use of freed data
- User-Level Atomic Operations
  - Allows for more efficient manipulation of integer objects without mutex protection - `atomic_ops(3C)`  
`atomic_add_*`( ), `atomic_or_*`( ), `atomic_and_*`( )

# Event Port Framework

- Event Port Framework
  - New event completion mechanism - `port_create(3C)`
  - Event queue that multiplexes events from disjoint sources
    - `PORT_SOURCE_AIO`
    - `PORT_SOURCE_FD`
    - `PORT_SOURCE_TIMER`
    - `PORT_SOURCE_USER`
  - Provides scalable, thread-safe event dispatching
  - Handling 20,000 connections is now cheap

# uDAPL 1.1

- User Direct Access Programming Library
  - uDAPL 1.1
    - Transport and Platform independent API for RDMA
    - High speed low latency communication through OS & protocol bypass
    - Access to IB remote shared memory technology  
`libdat(3lib)`, `daplt(7D)`, `tavor(7D)`

# Static linking

- Solaris no longer delivers static versions of system libraries or statically linked applications.
  - `/lib` is now a directory (was symlink), populated with all shared objects needed prior to `/usr` being mounted.

# Adoption

# Solaris Adoption Strategy

- Most applications will run, without modification
- Solaris guarantees forward binary compatibility, if your application uses Solaris public ABI's
  - <http://www.sun.com/software/solaris/programs/guarantee.html>
- SolCAT is provided for testing application compatibility on Solaris
  - appcert: tests ABI conformance

Once upon a time ...

# Platform support

- Solaris Kernel for SPARC platforms is 64-bit only!
  - Sun4m-based platforms are no longer supported
  - SPARCstation-4 SPARCstation-5 SPARCstation-10  
SPARCstation-10,SX SPARCstation-20 SPARCstation-LX  
SPARCstation-LX+ SPARCclassic SPARCclassic-X  
SPARCengine-EC-3
  - Booting a sun4m fails with the following message:  
This hardware platform is not supported by  
this release of Solaris.
  - UltraSPARC-I processor support has been  
removed.
  - Boot UltraSPARC-I fails with the following  
message:  
UltraSPARC-I processors are not supported  
by this release of Solaris

# 64-bit Packaging

- Solaris packages now contain both 32-bit and 64-bit components. Packages retain the names of the 32-bit package; 64-bit packages are no longer delivered.
- ISV packages which have explicit dependencies on 64-bit packages generate installation warnings but will continue to work.

Finally ...

# Road To Solaris 10

## 1. Software Express

- Monthly releases
- Free downloads
- Support available

## 2. Beta Testing

- Begins March 2004
- Big enrollments
- Full Java Enterprise System available

## 3. Shipping Q4 2004

- Simultaneous ship with Java Enterprise System

# Getting ready for Solaris 10

Easy to start planning and testing today via SX  
[sun.com/solarisexpress](http://sun.com/solarisexpress)

Application compatibility guaranteed

Easy adoption toolsets

Solaris 10 adoption services

Request your applications:  
[sun.com/solaris/10](http://sun.com/solaris/10)

# Solaris 10 summary

- Heavy investments in Solaris technology
- Again, massive enhancements in reliability, availability, serviceability, security and performance
- First class platform for server consolidation
- Continued strong compatibility



Solaris 10

[markus.halter@sun.com](mailto:markus.halter@sun.com)

