

D78767GC10

Edition 1.0

March 2013

D81109

ORACLE®

Oracle Solaris 11 Performance Management

Student Guide - Volume II

Author

David Giroux

Technical Contributors & Reviewers

Mike Carew

Benoit Chaffanjon

Glynn Foster

John Hathaway

Dominic Kay

Steve Kirby

Rosemary Martinak

Kristi McNeill

Chad Mynhier

Christian Wolbert

Editors

Malavika Jinka

Vijayalakshmi Narasimhan

Arijit Ghosh

Graphic Designer

Seema Bopaiah

Publishers

Jayanthi Keshavamurthy

Jobi Varghese

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Contents

Preface

1 Introduction

- Overview 1-2
- Goals 1-3
- Course Agenda: Day 1 1-4
- Course Agenda: Day 2 1-6
- Course Agenda: Day 3 1-7
- Course Agenda: Day 4 1-8
- Course Agenda: Day 5 1-9
- Introductions 1-10
- Your Learning Center 1-11

2 Introducing Performance Management

- Objectives 2-2
- Agenda 2-3
- Relevance 2-4
- Performance Management 2-5
- Terminology 2-6
- Performance Graphs 2-9
- Trade-Offs of Performance Tuning 2-10
- Trade-Offs: Example 2-11
- Conceptual Model of Performance 2-12
- Block Functional Diagrams 2-13
- Knowing the Speeds and Feeds 2-14
- Hardware Functional Block Diagram 2-15
- Software Functional Block Diagram 2-16
- Agenda 2-17
- Knowns and Unknowns 2-18
- Drill-Down Analysis Strategy 2-19
- Performance Analysis Approach 2-20
- Agenda 2-22
- Monitoring Tools in Oracle Solaris 11 2-23
- kstat Utility 2-24
- procfs File System 2-25

DTrace	2-26
System Configuration Assessment	2-27
Quiz	2-28
Summary	2-31
Practice 2 Overview: Introducing Performance Management	2-32

3 kstat Monitoring Tools

Objectives	3-2
Relevance	3-3
Identifying Performance Problems Using the kstat Tools	3-4
Agenda	3-5
sar(1)	3-6
Agenda	3-9
vmstat(1)	3-10
vmstat -p	3-11
Agenda	3-12
iostat(M)	3-13
iostat(1M)	3-14
Agenda	3-15
mpstat(1M)	3-16
Agenda	3-17
netstat(1M)	3-18
netstat -i	3-19
Agenda	3-20
nfsstat(1M)	3-21
Agenda	3-23
kstat(1M) Utility	3-24
Quiz	3-29
Summary	3-32
Practice 3 Overview: kstat Monitoring Tools	3-33

4 procfs Monitoring Tools

Objectives	4-2
Relevance	4-3
Agenda	4-4
procfs-Based Tools	4-5
Agenda	4-6
ps Command	4-7
/usr/ucb/ps Command	4-11
Agenda	4-13
prstat(1M) Utility	4-14

- prstat(1M) Utility Options 4-15
- prstat Output: Example 4-17
- Examining Processes 4-19
- Agenda 4-21
- truss(1) Utility 4-22
- Agenda 4-28
- Solaris Studio Performance Analyzer 4-29
- Quiz 4-30
- Summary 4-33
- Practice 4 Overview: Using ps, prstat, the proc Tools, and truss 4-34

5 Introduction to DTrace

- Objectives 5-2
- Relevance 5-3
- Agenda 5-4
- DTrace 5-5
- DTrace Architecture 5-6
- Agenda 5-7
- DTrace Command Syntax 5-8
- DTrace Providers 5-9
- DTrace One-Liners 5-11
- Running a DTrace One-Liner 5-12
- Agenda 5-13
- DTrace Toolkit 5-14
- Agenda 5-16
- DTrace Visualization Tool (DLight) 5-17
- DLight GUI 5-18
- Quiz 5-19
- Summary 5-21
- Practice 5 Overview: Using DTrace 5-22

6 Other Significant Tools

- Objectives 6-2
- Relevance 6-3
- Agenda 6-4
- swap(1M) 6-5
- Agenda 6-7
- CPU Performance Counters 6-8
- cpustat(1M) Utility 6-9
- trapstat(1M) Utility 6-14
- Agenda 6-15

- mdb(1) Utility 6-16
- Agenda 6-28
- Solaris Studio dbx(1) Utility 6-29
- Agenda 6-30
- zonestat Utility 6-31
- zonestat Utility: Examples 6-32
- Agenda 6-35
- GUDDS Script 6-36
- Quiz 6-37
- Summary 6-40
- Practice 6 Overview: Using swap, bus counters, and mdb Tools 6-41

7 Processes and Threads

- Objectives 7-2
- Relevance 7-3
- Agenda 7-4
- Layers of the Operating System 7-5
- Process 7-6
- Life Cycle of a Process 7-7
- Typical Life Cycle of a Process 7-8
- System Processes 7-10
- Process and Address Space 7-11
- pmap(1) 7-12
- Interrupts 7-14
- Interrupt Assignments 7-16
- Interrupt Controls 7-17
- Interprocess Communication (IPC) 7-18
- Process Kernel Structures 7-19
- Process-Related Performance Issues 7-20
- Agenda 7-21
- Threads 7-22
- pmap(1): MT Process 7-23
- Threads 7-24
- Locks 7-25
- Locking Problems 7-26
- lockstat Utility 7-27
- DTrace lockstat Provider 7-33
- adaptive_mutex.d 7-34
- Performance Threads 7-35
- Agenda 7-36
- Process-Related Tunable Parameters 7-37

Showing the Maximum Number of Processes and Limitations	7-39
Agenda	7-44
Process Scheduling	7-45
Scheduling State Diagram	7-47
Thread States	7-48
Scheduling Classes	7-49
Priorities	7-50
Timeshare	7-51
Interactive (IA) Scheduling Class	7-52
Fixed Priority (FX)	7-54
System (SYS)	7-55
Real-Time (RT)	7-56
Fair Share (FSS)	7-57
Quiz	7-59
Summary	7-62
Practice 7 Overview: Processes and Threads	7-63

8 System Caches and Buses

Objectives	8-2
Relevance	8-3
Agenda	8-4
Introducing Caches	8-5
Caches and Buses	8-6
Memory Hierarchy	8-7
Relative Access Times	8-8
System Caches	8-9
Cache Operation	8-11
Replacing Cache Data	8-12
Requesting Data from a Cache	8-13
Factors Affecting the Cache-Hit Rate	8-14
Effects of CPU Cache Misses	8-15
Oracle CPU Caches	8-16
Agenda	8-17
UltraSPARC T-Series Processor Family	8-18
CMT Architecture	8-19
CMT Thread Model	8-20
CMT Pipeline	8-21
Performance Issues and CMT	8-23
pgstat Command	8-26
pginfo Command	8-27
Agenda	8-28

System Buses	8-29
Peripheral Buses	8-30
busstat(1M) Command	8-31
prtdiag(1M) Command	8-32
Diagnosing Bus Problems	8-35
prtconf(1M) Command	8-36
cputrack(1M) Command	8-39
Quiz	8-41
Summary	8-44
Practice 8 Overview: System Caches and Buses	8-45

9 Memory

Objectives	9-2
Agenda	9-3
Main Memory	9-4
Non-Uniform Memory Access (NUMA)	9-5
lgrpinfo Command	9-6
Agenda	9-7
Virtual Memory 2 (VM2)	9-8
Agenda	9-10
MMU	9-11
Translation Lookaside Buffer (TLB)	9-12
Multiple Page Size Support (MPSS)	9-13
Implementing MPSS for a Running Process	9-14
trapstat Utility (SPARC Only)	9-15
Paging and Swapping	9-16
Clock Algorithm	9-17
fastscan and handspread_pages	9-18
Page Scanner Processing	9-19
General Memory-Related Tuning Parameters	9-20
Paging-Related Tuning Parameters	9-22
Additional Paging-Related Tuning Parameters	9-24
Swapping	9-26
Swapping Priorities	9-27
Swap Space	9-28
Swap-Related Tuning Parameters	9-29
Agenda	9-30
Intimate Shared Memory (ISM)	9-31
Dynamic Intimate Shared Memory (DISM)	9-32
Optimized Shared Memory (OSM)	9-34
Agenda	9-35

Memory Monitoring	9-36
Memory Summary	9-37
Memory Consumption	9-38
vmstat(1m) Utility	9-39
Identifying Paging Statistics	9-40
Using sar	9-41
Using the vmstat Command	9-42
Using kstat	9-43
Per-Process Paging Activity	9-44
DTrace Toolkit Scripts	9-45
Swapping Statistics	9-46
Memory Requirements for Applications	9-47
I/Os Queued to a Swap Device	9-48
Measuring the Distribution of Memory	9-49
Solaris Studio discover Utility	9-50
Quiz	9-51
Summary	9-54
Practice 9 Overview: Memory	9-55

10 Disk I/O and the ZFS File System

Objectives	10-2
Agenda	10-3
Disk Performance Considerations	10-4
Storage Data Characteristics	10-5
Disk Bottlenecks	10-6
Disk Utilization	10-7
Disk Saturation	10-8
Disk Configuration	10-9
Disk File Systems	10-10
HDDs Versus SSDs	10-11
Agenda	10-13
iostat Command	10-14
vmstat Command	10-15
fsstat Command	10-17
sar -d Command	10-18
zpool iostat Command	10-19
DTrace Disk I/O One-Liners	10-20
Oracle I/O Numbers Calibration Tool (ORION)	10-21
Agenda	10-22
Oracle Solaris ZFS	10-23
ZFS Design	10-24

ZFS Architecture: Overview	10-25
Interface Layer	10-26
Transactional Object Layer	10-27
Data Management Unit	10-28
ZFS Intent Log (ZIL)	10-29
ZAP Layer	10-30
Dataset and Snapshot Layer (DSL)	10-31
Pooled Storage Layer	10-32
ZFS ARC Cache Introduction	10-33
ZFS Data Structures	10-34
vdev Tree	10-35
vdev Label	10-36
Uberblock	10-38
General ZFS Administration	10-39
ZFS Limitations	10-40
Agenda	10-41
General Storage Tuning	10-42
ZFS Tuning Guidelines	10-43
General Storage Pools Considerations	10-45
Root Pool Considerations	10-46
Non-Root Pool Considerations	10-47
ZFS RAID-Z: Examples	10-48
Network-Attached Storage Considerations	10-49
ZFS Storage Pool: Maintenance and Monitoring	10-50
ZFS File System Considerations	10-52
Agenda	10-53
ZFS ARC Cache	10-54
Viewing ZFS ARC Statistics	10-56
ZFS ARC Cache Tuning Parameters	10-58
Additional ZFS Tuning Parameters	10-59
Viewing ZFS Kernel Parameters	10-61
Quiz	10-62
Summary	10-64
Practice 10 Overview: Disk I/O and the ZFS File System	10-65

11 Solaris 11 Network Tuning

Objectives	11-2
Agenda	11-3
Relevance	11-4
Introducing Network Performance	11-5
Terms Used for Network Analysis	11-6

Packets	11-7
Network Utilization	11-8
Network Errors	11-9
Effects of Misconfigured Components	11-10
Agenda	11-11
Oracle Solaris 11 Networking	11-12
Oracle Solaris 10 Network Model	11-13
Oracle Solaris 11 Network Model	11-14
Agenda	11-15
Data Packet Encapsulation	11-16
Reconfiguring the MTU	11-18
IP Multipathing (IPMP)	11-19
IPMP Configurations	11-20
Configuring IPMP: Active-Active	11-21
Configuring IPMP: Active-Standby	11-22
Link Aggregation	11-23
Configuring Link Aggregation	11-24
Network Virtualization	11-25
Consolidating to Virtual Networking	11-26
Bandwidth Management	11-27
Managing Bandwidth	11-28
Integrated Load Balancer (ILB)	11-29
ILB Example: DSR	11-30
ILB Example: NAT	11-31
ILB Example Using Zones	11-32
ILB Components and Terms	11-33
Load-Balancing Rule	11-35
ILB Algorithms	11-36
ilbadm Utility	11-37
Agenda	11-38
Monitoring Network Performance	11-39
Testing Response Time	11-40
Network Status	11-41
kstat Command (Network)	11-42
ipadm Utility	11-43
dladm Utility	11-44
Introducing the snoop Command	11-45
snoop Command	11-46
wireshark Utility	11-47
flowstat Utility	11-48
flowstat: Examples	11-49

traceroute Utility	11-50
Testing the Reliability of Packet Sizes	11-51
dlstat Utility	11-52
dlstat: Examples	11-53
TCP Statistics from DTrace	11-55
IP Statistics from DTrace	11-56
ICMP Statistics from DTrace	11-57
Agenda	11-58
Displaying and Setting Network Tunable Parameters	11-59
IP Tuning Parameters	11-61
IP Tuning Parameters: ipadm	11-63
TCP Connections	11-64
TCP Tuning Parameters	11-65
TCP Tuning Parameters: ipadm	11-67
NFS Tuning Parameters	11-68
Additional Network Tuning Parameters	11-74
Per-Route Tuning	11-76
Isolating Problems	11-77
Quiz	11-78
Summary	11-81
Practice 11 Overview: Monitoring Network Performance	11-82

12 Resource Management

Objectives	12-2
Agenda	12-3
Resource Management	12-4
When to Use Resource Management	12-5
Agenda	12-7
Projects and Tasks	12-8
Project Database	12-10
Project and Task Commands	12-11
Displaying the Project Database	12-12
Adding a Project to the Project Database	12-13
Agenda	12-14
Resource Management Control Mechanisms	12-15
Resource Control Enforcement	12-17
Resource Control Values, Privileges, and Actions	12-18
Configuring Resource Controls in a Project	12-20
Resource Control Commands	12-22
Agenda	12-23
Fair-Share Scheduler (FSS)	12-24

- Agenda 12-26
- Resource Pools 12-27
- Dynamic Resource Pools 12-28
- Resource Pool Properties 12-29
- Configuration Pool Objectives 12-31
- Resource Pool Commands 12-32
- Configuring Static Resource Pools 12-33
- Configuring Objectives 12-36
- Adding FSS to a Pool 12-37
- poolstat Command 12-38
- Agenda 12-39
- Resource Capping 12-40
- Resource-Capping Commands 12-41
- rcapstat Command 12-42
- Quiz 12-43
- Summary 12-47
- Practice 12: Overview 12-48

13 Oracle Solaris Virtualization Performance Management

- Objectives 13-2
- Agenda 13-3
- How Zones Work 13-4
- Global Zone Characteristics 13-6
- Nonglobal Zone Characteristics 13-7
- Agenda 13-8
- zonestat Utility 13-9
- zonestat Utility: Examples 13-10
- Agenda 13-13
- Zone-Wide Resource Controls 13-14
- rctl Resource Control 13-16
- rctl Resource Properties 13-17
- Zone-Wide Resource Control: Examples 13-18
- Agenda 13-21
- Oracle VM for SPARC 13-22
- CPU Whole Cores and CPU Cap 13-23
- Viewing CPU Whole Cores Configurations 13-24
- CPU Threading Modes and Workloads 13-25
- Viewing CPU Threading Modes 13-26
- Agenda 13-27
- Quiz 13-28
- Summary 13-33
- Practice 13: Overview 13-34

14 Performance Analysis and Testing

- Objectives 14-2
- Relevance 14-3
- Agenda 14-4
- Introduction 14-5
- Maintaining System Performance 14-6
- Performance Analysis Approach 14-7
- Errors 14-8
- Misconfigurations 14-9
- Eliminate Bottlenecks 14-10
- Fine-Tuning 14-11
- Viewing the Values of Tuning Parameter 14-12
- Setting Tuning Parameters 14-13
- Recovering /etc/system File Settings 14-14
- Unknown Bugs 14-15
- Agenda 14-16
- Types of Performance Testing 14-17
- Industry Benchmarks 14-19
- Agenda 14-20
- Performance-Testing Tools 14-21
- Workload Assessment 14-22
- Performance-Testing Tools: File Systems 14-23
- Performance-Testing Tools: Network 14-24
- Performance-Testing Tools: CPU 14-25
- Performance-Testing Tools: Memory 14-26
- Functional Diagram 14-27
- Understand Benchmark Software 14-29
- Sanity Test 14-30
- Double-Check 14-31
- Drive Resources to Saturation 14-32
- Document Your Test System 14-33
- Testing File Servers 14-34
- Ways to Avoid Client Caching 14-35
- Distribute Client Load 14-36
- Disk Matter 14-37
- Check Your Storage Profile 14-38
- Summary 14-39

9

Memory

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORACLE

Objectives

After completing this lesson, you should be able to:

- Describe system memory concepts
- Describe Virtual Memory 2
- Monitor memory utilization

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- **Memory concepts**
- Introduction to Virtual Memory 2
- Swapping, paging, and caching
- Shared memory
- Memory monitoring

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Main Memory

- Fast access
- Relatively inexpensive compared to registers and cache
- Is volatile
- Is organized into tiles, tilelets, chunks, and pages
- Memory Management Unit (MMU) translates virtual addresses to physical addresses

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The propagation delay to get the data off the chip is very low, less than 10 nanoseconds typically, but to get to the DRAM, the memory bus (up to three levels for large systems) has to be traversed. The main delay is in translating the virtual address generated on the CPU by the current thread to a physical address. This may require multiple trips to main memory to do the translation. Some systems may be able to access times of less than 10 ns, but a large server with multiple bus layers may be on the order of a 30–50 ns range or even higher.

Non-Uniform Memory Access (NUMA)

- Non-Uniform Memory Access, or NUMA, is an artifact of modern multiprocessor systems design.
- The non-uniformity refers to the amount of time (latency) required for a thread running on a CPU to get data from memory.
- The time varies based on which CPU and which memory bank(s) the data resides in.
- The Solaris kernel implements Memory Placement Optimization, or MPO, specifically to mitigate the effects of NUMA.
 - Latency group abstraction
 - Attempts to keep threads, the CPUs they run on, and the memory they reference within the same lgrp

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In high-end SMP systems, backplane bus can become a bottleneck. In non-NUMA platforms, all memory access must go through the system bus and it can reduce scalability. NUMA technology was introduced to relieve symptoms of this bottleneck. Every processor has its own local memory and connected by some form of hardware interconnect. A differentiation is also made between local and remote memory in NUMA-based servers. Access to local memory has less latency compared to access to remote memory from a processor set.

In Solaris, NUMA uses the concept of MPO, which attempts to place process resources into latency groups known as *locality groups*. A process running in a locality group (lgrp) has much less latency accessing memory local to that group. As the “remoteness” of memory increases, latency also increases. A lgrp is a hierarchical Directed Acyclic Graph (DAG) representing processor-like and memory-like devices, which are separated from each other by some access-latency upper bound. A node in this graph contains at least one processor and its associated local memory. All the lgrps in the system are enumerated with respect to the root node of the DAG, which is called the root lgrp. Two modes of memory placement are available, *next-touch* and *random*. Next-touch is the default for thread private data, while random is useful for shared memory regions accessed by multiple threads as it can reduce contention.

lgrpinfo Command

```
$ lgrpinfo -Ta
0
|-- 1
|   CPU: 0
|   Memory: installed 2.0G, allocated 1.2G, free 790M
|   Load: 0.274
|   Latency: 56
|-- 2
    CPU: 1
    Memory: installed 2.0G, allocated 1019M, free 1.0G
    Load: 0.937
    Latency: 56
Lgroup latencies:
-----
| 0 1 2
-----
0 | 83 83 83
1 | 83 56 83
2 | 83 83 56
-----
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This example of the `lgrpinfo -Ta` command shows the lgroup topology tree, resources, memory, and CPU information on a two-CPU machine.

Agenda

- Memory concepts
- Introduction to Virtual Memory 2
- Swapping, paging, and caching
- Shared memory
- Memory monitoring

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Virtual Memory 2 (VM2)

- VM1 was designed when a large computer had 16 MB of RAM and one CPU.
- Computer architecture has evolved:
 - Larger machines
 - Large pages
 - NUMA
 - Dynamic reconfiguration
 - FMA

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The old VM system ("VM1") was designed when a large computer had 16 MB of RAM and one CPU. However, computer architecture has evolved: larger machines, large pages, NUMA, dynamic reconfiguration, FMA, and so on. Solaris has adapted, but even the fixes for these changes are showing their age. To keep pace with evolving computer technologies, Solaris 11 introduces Virtual Memory 2 (VM2).

VM2 gives a reliable way to get exclusive access to memory based on *predictors* to coordinate decision making. Features such as physical memory add/remove, memory power management, and enhanced debugging capabilities are supported.

Virtual Memory 2 (VM2)

- VM2 is being developed in three phases.
 - Solaris 11.1 is currently at phase one.
- VM2 phase one introduces the *memory tiles* concept.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

VM2 is being developed in three phases. Solaris 11.1 virtual memory design is currently in phase 1. VM2 phase 1 introduces the *memory tiles* concept. Memory features such as NUMA, kernel cage, large pages are all built-in.

In phase 2, the development focuses on finishing core data structures including:

- VM data structures focused on files, not pages
- New version of Optimized Shared Memory (OSM)
- Hardware Address Translation (HAT), segment driver changes

In phase 3, the development focuses on:

- MAP_PRIVATE mappings
- File system interaction
- Paging and swapping
- Kernel's address space
- Address space layer locking

Agenda

- Memory concepts
- Introduction to Virtual Memory 2
- **Swapping, paging, and caching**
- Shared memory
- Memory monitoring

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

MMU

- Translates virtual addresses to physical addresses
- x86 architectures use table lookup mechanism:
 - Implemented in hardware and firmware
 - Consumes a huge amount of memory for page tables
 - Only two page sizes available
- SPARC architectures use the SFMMU:
 - Implemented in hardware, firmware, and software
 - Complicated
 - Many page sizes available
- `pagesize -a`: The `-a` option lists all the page sizes available for that system.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The way the MMU works for different architectures is a more appropriate topic for the Oracle Solaris 11 Operating System Internals course. However, suffice it to say it consumes much of the system resources in terms of memory and CPU cycles.

The `pagesize` command reports the default base page size for the system.

The `-a` option lists all the page sizes available for that system.

Translation Lookaside Buffer (TLB)

- MMU translations are expensive:
 - Typically four or more accesses to memory
 - Register accesses and MMU logic
- The page is likely to be accessed more than once.
- The Physical Page Number (PPN) is cached in fast access memory called the TLB.
- The TLB is on the CPU chip and uses SRAM (like cache).
- The size of the TLB varies for different processors.
- It keeps track of:
 - PPN
 - Virtual Page Number
 - The process that did the translation
- The `trapstat [-t | T]` command is used to show TLB information.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Note

See http://developers.sun.com/solaris/articles/t1t2_perf_counter.html.

Multiple Page Size Support (MPSS)

- Solaris 9 introduced MPSS.
- Large pages make it possible to track more memory with fewer translations and fewer TLB misses.

```
# pagesize -a
8192
65536
524288
4194304
```

- For amd64 platforms:

```
# pagesize -a
4096
2097152
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Larger page sizes reduce the number of TLB translations needed to address memory. Fewer translations to store reduces the number of TLB cache misses that can occur. Some applications realize a dramatic increase in response time through proper implementation of MPSS.

Note: See “Supporting Multiple Page Sizes in the Solaris™ Operating System” at http://learningsolaris.com/docs/Multiple_page_sizes_bp.pdf.

Implementing MPSS for a Running Process

```
# ppgsz -o heap=512k -p 257
```

- In this example, the preferred page size for the heap segment of process 257 is set to 512 KB.
- To initialize the environment for MPSS before running an application:

```
# LD_PRELOAD=$LD_PRELOAD:mpss.so.1 MPSSHEAP=512K  
# ./newprog
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Solaris 11 OS has support for out-of-the-box large pages (that is, large page OOB). The large page OOB feature turns on MPSS automatically for applications' heap (BSS) and libraries' text segments.

Check for additional information per mapping by using `pmap -xs PID#`. The `-s` option prints HAT page size information.

trapstat Utility (SPARC Only)

- The `trapstat` utility provides cache statistics on TLB activity. The following command shows the output of the `trapstat` utility after a five-second delay:

```
# trapstat -t
```

cpu	m	itlb-miss	%tim	itsb-miss	%tim	dtlb-miss	%tim	dtsb-miss	%tim	%tim
0	u	1	0.0	0	0.0	2171237	45.7	0	0.0	45.7
0	k	2	0.0	1	0.0	3751	0.1	7	0.0	0.1
=====										
ttl		3	0.0	1	0.0	2192238	46.2	7	0.0	46.2

- Statistics are shown for user (u) and kernel (k) space, and `trapstat` shows TLB misses by instruction (itlb) and data (dtlb).
- The percentage of time consumed by TLB misses (%tim) is 46.2 percent in this example.

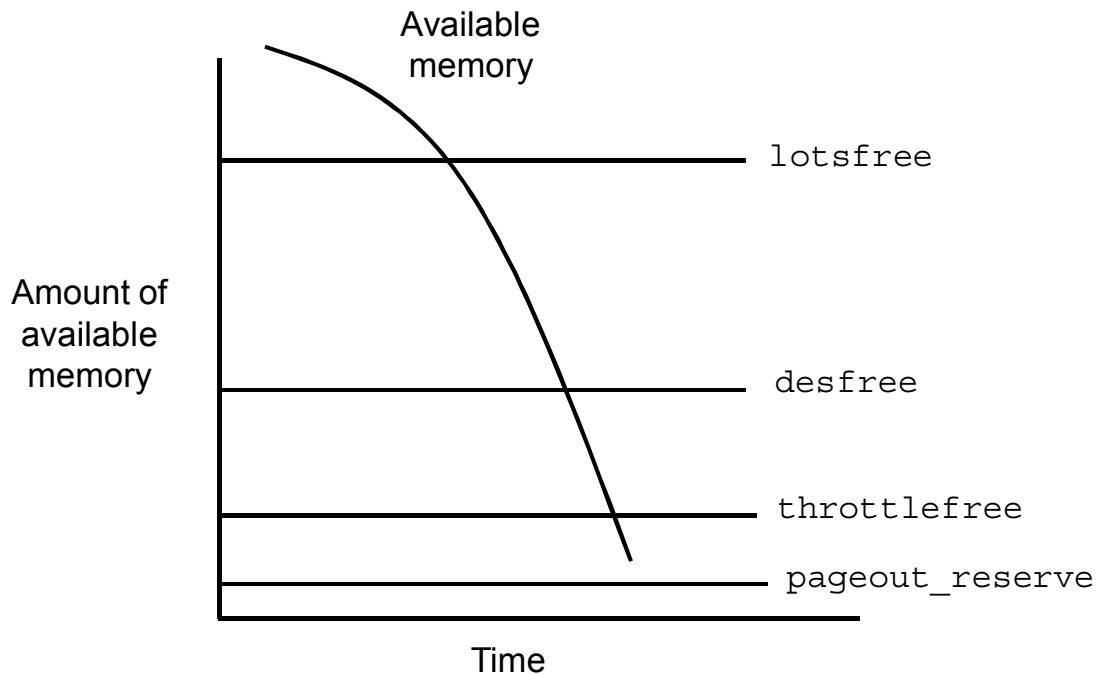
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Note

See “Taming Your Emu to Improve Application Performance” at http://learningsolaris.com/docs/Larger_page_sizes_bp.pdf.

Paging and Swapping



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

- As the amount of available memory drops below a threshold called `lotsfree`, the page daemon begins to page out not recently used pages.
- This is checked every quarter of a second. Because memory could be used up very quickly and a quarter of a second is a long time (in terms of CPU speeds), there is a second threshold called `desfree`.
- If the amount of available memory drops below `desfree`, the page daemon kicks in immediately. It does not wait for the next quarter-of-a-second boundary.
- In addition, if the amount of memory stays below `desfree` for a long time (5 seconds), the swapper begins to “soft swap.” Soft swapping means the swapper will begin to swap out processes that have been sleeping for more than `maxslp` (20) seconds.
- If the amount of available memory drops below `throttlefree`, all user and nonurgent kernel requests for memory are forced to wait until there is more than `throttlefree` memory.
- If the available memory drops below `pageout_reserve`, the only processes that can get pages from `freelist` are the page daemon and the swapper.

fastscan and handspread_pages

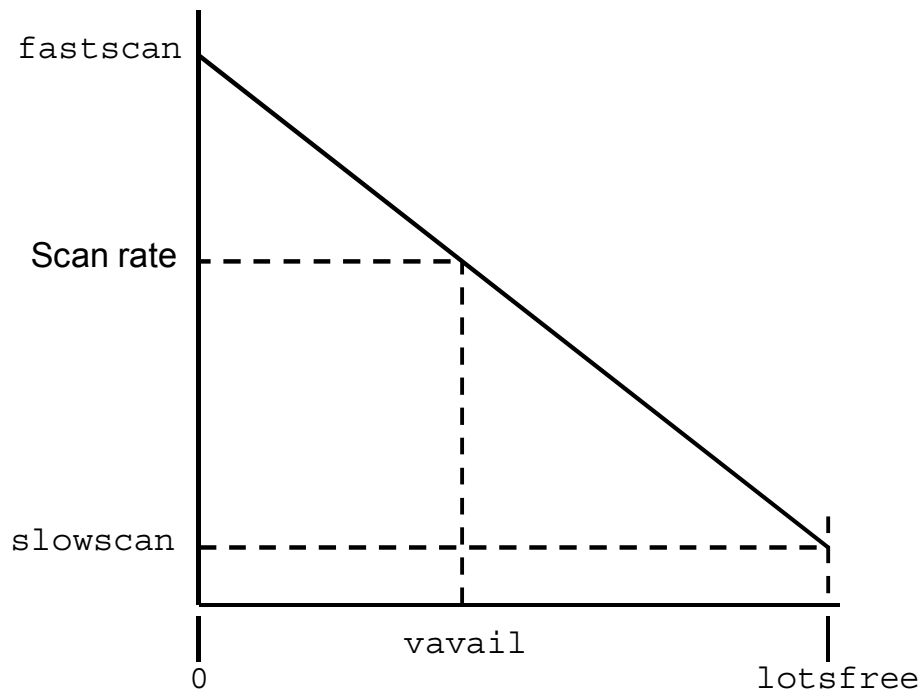
- Before Solaris 9, the default values for `fastscan` and `handspread_pages` could be set too low for large systems.
- The current strategy is to count the number of pages the system is capable of scanning per second at boot time. The result is placed in `pageout_rate`.
- The `fastscan` and `handspread_pages` parameters are reset to `pageout_rate/10` to let the `pageout_scanner` consume up to 10 percent of one CPU.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered within a solid red rectangular bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The resulting values for `handspread_pages` and `fastscan` on systems with large amounts of memory turn out to be hundreds of times greater than the defaults that were being used in Solaris 8.

Page Scanner Processing



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

- **lotsfree**: The number of free pages available, below which the page daemon starts scanning. The default value for **lotsfree** is 1/64 of memory-kernel pages. It is checked every quarter of a second.
- **vavail**: Virtually available memory, set equal to **freemem - deficit** and then clamped between 0 and **lotsfree**
- **deficit**: The amount of memory about to be used by a process that was just swapped in, or a page-in, on-demand process that has just exceeded. It is like a reservation on memory. It is quickly degraded over time back to 0.
- **fastscan**: The maximum scan rate. Used when **vavail** = 0. The default value is 64 MB or half of the **physmem** value, whichever is smaller.
- **slowscan**: The scan rate when **vavail** is used is **lotsfree** memory. The default value is 100 pages/sec.

If the amount of memory is midway between 0 and **lotsfree**, a routine called **schedpaging()** does a linear interpolation to calculate the appropriate number of pages to scan based on the amount of available memory.

For example, if **vavail** is halfway between 0 and **lotsfree** memory ($\text{lotsfree}/2$), the desired scan rate would be half way between **slowscan** and **fastscan**.

General Memory-Related Tuning Parameters

Parameter	Default Value
phymem	Number of usable pages of physical memory available on the system, not counting the memory where the core kernel and data are stored
default_stksize	<ul style="list-style-type: none"> 3 x <code>PAGESIZE</code> on SPARC systems 5 x <code>PAGESIZE</code> on x64 systems
lwp_default_stksize	<ul style="list-style-type: none"> 24,576 for SPARC systems 20,480 for x64 systems
segkpsize	2 GB

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This table shows the tuning parameters related to memory.

- `phymem`: Modifies the system's configuration of the number of physical pages of memory after Solaris and firmware are accounted for. You can change the `phymem` parameter whenever you want to test the effect of running the system with less physical memory. Because this parameter does not take into account the memory used by the core kernel and data, as well as various other data structures allocated early in the startup process, the value of `phymem` should be less than the actual number of pages that represent the smaller amount of memory.
- `default_stksize`: Specifies the default stack size of all threads. No thread can be created with a stack size smaller than `default_stksize`. If `default_stksize` is set, it overrides `lwp_default_stksize`. Increase the value when the system panics because it has run out of stack space.
- `lwp_default_stksize`: Specifies the default value of the stack size to be used when a kernel thread is created, and when the calling routine does not provide an explicit size to be used. Increase the value when the system panics because it has run out of stack space.

- `segkpsize`: Specifies the amount of kernel pageable memory available. This memory is used primarily for kernel thread stacks. Increasing this number allows either larger stacks for the same number of threads or more threads. A system running a 64-bit kernel uses a default stack size of 24 KB. To support large numbers of processes on a system, it is required to change `segkpsize`. The default size is 2 GB, assuming at least 1 GB of physical memory is present. This default size allows the creation of 24-KB stacks for more than 87,000 kernel threads. The size of a stack is the same, whether the process is a 32-bit or 64-bit process. If more than this number is needed, `segkpsize` can be increased, assuming sufficient physical memory exists.

Paging-Related Tuning Parameters

Parameter	Default Value
lotsfree	One-sixty-fourth of looppages (main memory - kernel pages)
desfree	lotsfree/2
minfree	desfree /2
throttlefree	minfree
pageout_reserve	throttlefree/2
pages_pp_maximum	The greater of (tune_t_minarmem+ 100 and [4%ofmemory available at boot time + 4MB]
tune_t_minarmem	25

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This table shows the tuning parameters related to paging.

- **lotsfree:** Serves as the initial trigger for system paging to begin. When this threshold is crossed, the page scanner starts to look for memory pages to reclaim. Consider changing the `lotsfree` parameter when demand for pages is subject to sudden sharp spikes, the memory Algorithm might be unable to keep up with demand. A rule of thumb is to set this parameter to two times what the system needs to allocate in a few seconds. This parameter is workload dependent. A DBMS server can probably work fine with the default settings. However, you might need to adjust this parameter for a system doing heavy file system I/O. For systems with relatively static workloads and large amounts of memory, lower this value. The minimum acceptable value is 512 KB, expressed as pages by using the page size returned by `getpagesize`.
- **desfree:** Specifies the preferred amount of memory to be free at all times on the system. For systems with relatively static workloads and large amounts of memory, lower this value. The minimum acceptable value is 256 KB, expressed as pages by using the page size returned by `getpagesize`.

- `minfree`: Specifies the minimum acceptable memory level. When memory drops below this number, the system biases allocations toward allocations necessary to successfully complete pageout operations or to `swap` processes completely out of memory. Either allocation denies or blocks other allocation requests. The default value is generally adequate. For systems with relatively static workloads and large amounts of memory, lower this value. The minimum acceptable value is 128 KB.
- `throttlefree`: Specifies the memory level at which blocking memory allocation requests are put to sleep, even if the memory is sufficient to satisfy the request. The default value is generally adequate. For systems with relatively static workloads and large amounts of memory, lower this value. The minimum acceptable value is 128 KB.
- `pageout_reserve`: Specifies the number of pages reserved for the exclusive use of the pageout or scheduler threads. When available memory is less than this value, nonblocking allocations are denied for any processes other than pageout or the scheduler. Pageout needs to have a small pool of memory for its use so it can allocate the data structures necessary to do the I/O for writing a page to its backing store. The default value is generally adequate. For systems with relatively static workloads and large amounts of memory, lower this value. The minimum acceptable value is 64 KB.
- `pages_pp_maximum`: Defines the number of pages that must be unlocked. If a request to lock pages would force available memory below this value, that request is refused. Consider changing the `pages_pp_maximum` parameter when memory-locking requests fail or when attaching to a shared memory segment with the `SHARE_MMU` flag fails, yet the amount of memory available seems to be sufficient. Excessively large values can cause memory locking requests (`mlock`, `mlockall`, and `mementl`) to fail unnecessarily.
- `tune_t_minarmem`: Defines the minimum available resident (not swappable) memory to maintain, which is necessary to avoid deadlock. It is used to reserve a portion of memory for use by the core of the OS. Pages restricted in this way are not seen when the OS determines the maximum amount of memory available. The default value is generally adequate. Consider increasing the default value if the system locks up and debugging information indicates that no memory was available.

Additional Paging-Related Tuning Parameters

Parameter	Default Value
<code>fastscan</code>	Set to 64 MB at system boot time. After the system is booted, the value is set to the number of pages that the scanner can scan in one second by using 10 percent of a CPU.
<code>slowscan</code>	100 pages
<code>min_percent_cpu</code>	4
<code>pages_before_pager</code>	200
<code>maxpgio</code>	40 pages
<code>handspreadpages</code>	<code>fastscan</code>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This table shows more tuning parameters related to paging.

- `fastscan`: Defines the maximum number of pages per second that the system looks at when memory pressure is highest. Consider changing the `fastscan` parameter when more aggressive scanning of memory is preferred during periods of memory shortfall, especially when the system is subject to periods of intense memory demand or when performing heavy file I/O.
- `slowscan`: Defines the minimum number of pages per second that the system looks at when attempting to reclaim memory. Consider changing the `slowscan` parameter when more aggressive scanning of memory is preferred during periods of memory shortfall.
- `min_percent_cpu`: Defines the minimum percentage of CPU that pageout can consume. This parameter is used as the starting point for determining the maximum amount of time that can be consumed by the page scanner. Increasing this value on systems with multiple CPUs and lots of memory, which are subject to intense periods of memory demand, enables the pager to spend more time attempting to find memory.

- `pages_before_pager`: Defines part of a system threshold that immediately frees pages after an I/O completes instead of storing the pages for possible reuse. The threshold is `lotsfree + pages_before_pager`. The NFS environment also uses this threshold to curtail its asynchronous activities as memory pressure mounts. You might change this parameter when the majority of I/O is done for pages that are truly read or written once and never referenced again. Setting this variable to a larger amount of memory keeps adding pages to the free list. You might also change this parameter when the system is subject to bursts of severe memory pressure. A larger value here helps maintain a larger cushion against the pressure.
- `maxpgio`: Defines the maximum number of page I/O requests that can be queued by the paging system. This number is divided by 4 to get the actual maximum number used by the paging system. This parameter is used to throttle the number of requests as well as to control process swapping. Increase this parameter to page out memory faster. A larger value might help to recover faster from memory pressure if more than one swap device is configured or if the swap device is a striped device.
- `handspreadpages`: Solaris uses a two-handed clock algorithm to look for pages that are candidates for reclaiming when memory is low. The first hand of the clock walks through memory marking pages as unused. The second hand walks through memory some distance after the first hand, checking to see whether the page is still marked as unused. If so, the page is subject to being reclaimed. The distance between the first hand and the second hand is `handspreadpages`. Increasing the `handspreadpages` value increases the separation between the hands, and therefore, the amount of time before a page can be reclaimed.

Swapping

- *Soft swapping* occurs when free memory is less than `desfree` for 5 seconds or more.
- The swapper, a process called the `sched`, swaps out processes that have been sleeping longer than `maxslp` (20 seconds).
- *Hard swapping* occurs when all the following three conditions are true:
 - At least two processes are on the run queue, waiting for CPU.
 - Free memory is less than the `desfree` parameter on average for more than 30 seconds.
 - The sum of `pageins` and `pageout` operations exceeds `maxpgio`, or `freemem` is less than `minfree`.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Using `mdb -k`, you can check whether a system has been hard swapping or soft swapping by entering:

```
> hardswap/D
hardswap:
hardswap:          0
> softswap/D
softswap:
softswap:          336
> $q
```


Swapping Priorities

- The following formula calculates effective priority for an LWP:

```
epri = swapin_time - rss / (maxpgio / 2) - pri
```

- The LWP with the highest-calculated effective priority is swapped out.
- All the LWPs of a process must be swapped before the address space is swapped out.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `swapin_time` variable actually represents the number of seconds a process has been asleep. The effect is that a process that has not been running recently is more likely to be swapped than an active process, so it is a *least recently used (LRU)* approach.

Swap Space

- Swap space is consumed in two phases: *reservation* and *allocation*.
- The pages that are reserved for swap space include:
 - Stack
 - Heap
 - Modified initialized data pages
- These pages are *anonymous memory* because they are not named objects (files).

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Swap-Related Tuning Parameters

Parameter	Default Value
swapfs_reserve	The smaller of 4 MB and 1/16th of physical memory
swapfs_minfree	The larger of 2 MB and 1/8th of physical memory

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This table shows tuning parameters related to swap.

- **swapfs_reserve**: Defines the amount of system memory that is reserved for use by system (UID= 0) processes. Changing the **swapfs_reserve** is generally not necessary.
- **swapfs_minfree**: Defines the desired amount of physical memory to be kept free for the rest of the system. Attempts to reserve memory for use as swap space by any process that causes the system's perception of available memory to fall below this value are rejected. Pages reserved in this manner can only be used for locked-down allocations by the kernel or by user-level processes. Consider changing the **swapfs_minfree** parameter when processes are failing because of an inability to obtain swap space, yet the system has memory available.

Agenda

- Memory concepts
- Introduction to Virtual Memory 2
- Swapping, paging, and caching
- **Shared memory**
- Memory monitoring

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Intimate Shared Memory (ISM)

- ISM improves the performance of programs that use shared memory.
- ISM allows applications to use large memory pages.
- When Oracle database attaches to an ISM segment:
 - All memory pages are instantiated and locked in memory
 - Shared memory is always instantly available
- Advantages over standard System V shared memory:
 - ISM is automatically locked by the Solaris kernel when the segment is created.
 - It reduces the need for kernel memory and reduces CPU time.
 - Large pages are allocated for ISM segments.
 - No swap space is needed.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ISM is a feature of the Solaris 11 kernel, which improves the performance of programs that use shared memory, and can also improve overall system performance when many processes are accessing the same shared memory (as with the Oracle database). The Solaris ISM facility allows applications to use large pages instead of the default 8-KB pages, thus increasing the reach of the TLB and its applications to access a larger working set without incurring the cost of TLB misses. For example, in SPARC platforms, pages can be as large as 256 MB.

When Oracle database attaches to an ISM segment, all memory pages are instantiated and locked in memory. The result is that shared memory is always instantly available. If, for some reason, the requested amount of shared memory cannot be locked in memory (for example, not enough physical memory is available, or the user does not have permission to allocate that much locked memory), the database instance startup is aborted and an error message is posted in the alert log. ISM offers several performance benefits over standard System V shared memory:

- ISM shared memory is automatically locked by the Oracle Solaris kernel when the segment is created.
- Kernel virtual to physical memory address translation structures are shared between processes that attach to the shared memory, saving kernel memory and CPU time.
- Large pages, supported by the system's Memory Management Unit (MMU), are automatically allocated for ISM segments.
- Because ISM memory is locked, no swap space is needed to back it, thereby saving disk space.

Dynamic Intimate Shared Memory (DISM)

- DISM has the same essential characteristics as ISM, except that it is dynamically resizable.
- Resizing does not require a system reboot.
- Memory locking and unlocking are done by the Oracle database, not Solaris.
- DISM shares the same performance benefits as ISM. In particular:
 - Memory can be locked and unlocked
 - Kernel virtual memory to physical memory address translation structures are shared between processes.
 - Large pages are automatically allocated for DISM segments.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris DISM provides shared memory with the same essential characteristics as ISM, except that it is dynamically resizable. This means that DISM, if configured correctly, offers the performance benefits of ISM, while allowing the shared memory segment to be dynamically resized, both for the sake of performance and to allow dynamic reconfiguration.

With ISM, it is not possible to change the size of an ISM segment once it has been created. To change the size of database buffer caches, databases must be shut down and restarted. This limitation has a negative impact on system availability. For example, if memory is to be removed from a system due to a dynamic reconfiguration event, it may be necessary to first shut down one or more database instances. DISM was designed to overcome this limitation. A large DISM segment can be created when the database boots, and sections of it can be selectively locked or unlocked as memory requirements change. Instead of the kernel automatically locking DISM memory, though, locking and unlocking is done by the Oracle database, providing the flexibility to make adjustments dynamically. DISM is like ISM, except that it is not automatically locked.

DISM shares the same performance benefits as ISM. In particular:

- Memory can be locked, preventing paging and allowing I/O to use fast kernel locking mechanisms. Memory can also be unlocked and freed dynamically and returned to the operating system, allowing it to be used for other purposes.

- Kernel virtual to physical memory address translation structures are shared between processes that attach to the DISM segment, saving kernel memory and CPU time.
- Large pages, supported by the system's Memory Management Unit (MMU), are automatically allocated for DISM segments.

The Oracle database will use DISM instead of ISM if SGA_MAX_SIZE is set larger than the total of the database, the shared pool, the redo buffers, the large pool, the Java pool, and the system global area (SGA) fixed size.

Optimized Shared Memory (OSM)

- OSM is a dynamic, NUMA-optimized, granular shared memory that offers flexibility without compromising performance or functionality.
- OSM allows dynamic resizing of the Oracle Database SGA without having to reserve memory and reboot the Oracle Database.
- Up to eight times faster startup and shutdown of Oracle Database instances.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris 11.1 includes a new OSM interface, which is a dynamic, NUMA-optimized, granular shared memory that offers flexibility without compromising performance or functionality.

OSM allows dynamic resizing of the Oracle Database SGA without having to reserve memory and reboot the Oracle Database. It also allows faster startup of Oracle Database instances by using fewer OSM segments, which can be grown later as needed. The next generation Oracle Database technology is planned to use OSM to dramatically speed up Oracle Database start and stop operations and allow online resizing of the SGA to avoid down time.

Agenda

- Memory concepts
- Introduction to Virtual Memory 2
- Swapping, paging, and caching
- Shared memory
- **Memory monitoring**

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Memory Monitoring

- Memory is consumed by:
 - User processes
 - Kernel operations
 - File caching
 - Shared memory (that is, databases)
- Determine the amount of physical memory with:

```
# prtconf | grep Memory  
Memory size: 512 Megabytes
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Memory Summary

# echo "::memstat" mdb -k				
Page Summary	Pages	MB	%Tot	
-----	-----	-----	-----	-----
Kernel	279487	2183	13%	
ZFS File Data	276173	2157	13%	
Anon	78565	613	4%	
Exec and libs	2772	21	0%	
Page cache	15338	119	1%	
Free (cachelist)	10057	78	0%	
Free (freelist)	1418376	11081	68%	
Total	2080768	16256		

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This is one of the quickest ways to view memory summary.

Memory Consumption

```
# mdb -k
```

```
Loading modules: [ unix genunix specfs dtrace ufs sd mpt pcisch ip hook
neti sctp arp usba fcp fctl nca lofs zfs md cpc random crypto wrsmd fcip
logindmux ptm spps nfs ]
```

```
> ::memstat
```

Page Summary	Pages	MB	%Tot
-----	-----	-----	-----
Kernel	63340	494	3%
ZFS File Data	109681	856	5%
Anon	70883	553	3%
Exec and libs	6166	48	0%
Page cache	27058	211	1%
Free (cachelist)	18473	144	1%
Free (freelist)	1762868	13772	86%
Total	2058469	16081	
Physical	2055438	16058	

On UltraSPARC the page size is 8192 bytes.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This is one of the quickest ways to see what is using memory.

vmstat(1m) Utility

- The `vmstat` command displays free memory on the system along with current paging activity.

```
# vmstat 3 3
kthr      memory          page        disk        faults        cpu
r  b  w     swap    free   re  mf pi po fr de sr dd f0 s1 --   in    sy    cs us sy id
0  0  0  1103256  318920  2 107  1  0  0  0  0  1  0  0  0  402   95   54  0  3 96
0  0  0  1094896  310648  79   7 763  0  0  0  0 151  0  0  0  702 3604  356  7 38 56
0  0  0  1094736  310696  79   0 736  0  0  0  0 144  0  0  0  689 4834  359  9 52 39
```

- The first row of output represents an average since boot time.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The most important columns to check for memory shortage are `sr` and `w`. `sr` stands for scan rate and a nonzero value indicates that the `pageout` scanner is actively searching for pages to free. The `w` column is the number of swapped-out LWPs. If any LWPs are swapped, it indicates that the page daemon could not keep up with the demand for memory.

Identifying Paging Statistics

- The `sar`, `memstat`, and `vmstat` commands
- Using `sar` reports

```
# sar -g 2 3
SunOS texastea 5.10 Generic sun4u      09/08/2005
11:46:52  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
11:46:54      0.00      0.00      0.00      0.00      3.10
11:46:56      0.00      0.00      0.00      0.00      1.35
11:46:58      0.00      0.00      0.00      0.00      0.00
Average      0.00      0.00      0.00      0.00      2.22
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

`pgscan/s` reports the number of pages scanned per second by the `pageout` scanner (same as `sr` in `vmstat`).

Using sar

```
# sar -p 4 4
```

```
SunOS texastea 5.10 Generic sun4u 09/08/2005
```

	atch/s	pgin/s	ppgin/s	pflt/s	vflt/s	slock/s
12:05:15	102.98	3.23	3.23	236.72	279.16	0.00
12:05:23	90.80	8.46	8.46	260.70	308.21	0.00
12:05:27	112.66	0.00	0.00	261.79	309.43	0.00
12:05:31	83.42	0.00	0.00	244.06	288.37	0.00
Average	97.46	2.92	2.92	250.81	296.28	0.00

```
# sar -r 4 4
```

```
SunOS texastea 5.10 Generic sun4u 09/08/2005
```

	freemem	freeswap
12:26:14	32500	2068796
12:26:18	32133	2054924
12:26:22	32018	2058243



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `pgin/s` column is the number of page-in requests per second. `ppgin/s` is the number of pages paged in per second.

`vflt/s` is the number of times there was not a valid translation from the virtual to physical address. It is the same as `mf` in `vmstat`.

Using the `vmstat` Command

```
# vmstat 5 4
```

kthr			memory				page				disk				faults			cpu			
r	b	w	swap	free	re	mf	pi	po	fr	de	sr	dd	f0	s1	--	in	sy	cs	us	sy	id
0	0	0	1102576	317976	3	142	2	0	0	0	0	1	0	0	0	403	167	57	1	4	95
1	0	0	1088992	305528	279	87	0	0	0	0	0	50	0	0	0	500	14957	242	22	70	8
1	0	0	1088040	304456	163	0	0	0	0	0	0	59	0	0	0	518	13984	258	22	73	5
2	0	0	1087712	304160	240	0	0	0	0	0	0	54	0	0	0	509	15108	228	22	75	3
1	0	0	1087408	303912	189	0	0	0	0	0	0	54	0	0	0	508	14462	225	22	73	5


```
# vmstat -p 5 5
```

memory			page				executable				anonymous			filesystem		
swap	free	re	mf	fr	de	sr	epi	epo	epf	api	apo	apf	fpi	fpo	fpf	
1102456	317872	3	155	0	0	0	0	0	0	0	0	0	0	2	0	0
1072848	291200	30	1607	0	0	0	0	0	0	0	0	0	0	24	0	0
1071616	291096	99	1537	0	0	0	0	0	0	0	0	0	0	2	0	0
1071344	290976	72	1534	0	0	0	0	0	0	0	0	0	0	0	0	0
1071720	290800	87	1541	0	0	0	0	0	0	0	0	0	0	0	0	0

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Any anonymous activity, such as a nonzero value for `apo`, indicates that the page daemon or the swapper is active.

Using kstat

```
# kstat -n system_pages
module: unix                               instance: 0
name:   system_pages                       class:   pages
...<output omitted>
    desfree                               16058
    desscan                               25
    econtig                               54657024
    fastscan                              118808
    freemem                               1783178
    kernelbase                            16777216
    lotsfree                               32116
    minfree                                8029
    ...<output omitted>
    slowscan                              100
    snaptime                              2795681.0445757
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `kstat -n system_pages` command is a quick way to check the values for `lotsfree`, `desfree`, and `minfree` to see whether the default values are being used. The `lotsfree` parameter is in pages and defaults to (number of memory pages - the kernel pages) / 64. Then, `desfree` defaults to `lotsfree/2` and `minfree` defaults to `desfree/2`.

Per-Process Paging Activity

```
# dtrace -n 'pgin { @[execname] = count() }'  
dtrace: description 'pgin ' matched 1 probe  
^C  
vold                      1  
bash                      1  
dtrace                    1  
se.sparcv9                2  
df                        4  
find                      9552
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `pgin` probe is part of the `vminfo` provider in DTrace. It is fired every time there is a page-in request. If you want to see the number of pages paged in, you can use the probe name `pgpgin`.

DTrace Toolkit Scripts

- `anonpgpid.d`: Anonymous memory paging information by PID on CPU
- `minfbypid.d`: Minor faults by PID
- `minfbyproc.d`: Minor faults by process name
- `pgpginbypid.d`: Pages paged in by PID
- `pgpginbyproc.d`: Pages paged in by process name
- `swapinfo.d`: Print virtual memory information
- `vmbypid.d`: Virtual memory stats by PID
- `xvmstat`: Extended `vmstat` demonstration by using DTrace

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Swapping Statistics

# vmstat -S 5 5																					
kthr			memory			page				disk				faults			cpu				
r	b	w	swap	free	si	so	pi	po	fr	de	sr	dd	f0	s1	--	in	sy	cs	us	sy	id
0	0	0	1102368	317792	0	0	2	0	0	0	0	1	0	0	0	403	217	59	1	5	94
0	0	0	1093896	310328	0	0	0	0	0	0	0	0	0	0	0	401	31	52	0	1	99
0	0	0	1093896	310328	0	0	0	0	0	0	0	0	0	0	0	401	25	45	0	0	99
0	0	0	1093896	310328	0	0	0	0	0	0	0	0	0	0	0	401	20	48	0	0	100
0	0	0	1093896	310328	0	0	0	0	0	0	0	0	0	0	0	401	26	50	0	0	99
# sar -w 4 4																					
SunOS texastea 5.10 Generic sun4u 09/08/2005																					
16:33:55 swpin/s bswin/s swpot/s bswot/s pswch/s																					
16:33:59 0.00 0.0 0.00 0.0 47																					
16:34:03 0.00 0.0 0.00 0.0 47																					
16:34:07 0.00 0.0 0.00 0.0 41																					
16:34:12 0.00 0.0 0.00 0.0 54																					
Average 0.00 0.0 0.00 0.0 48																					

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Memory Requirements for Applications

```
# pmap -xl 9
9:      /lib/svc/bin/svc.configd
```

Address	Kbytes	RSS	Anon	Locked	Mode	Mapped File
00010000	400	352	-	-	r-x--	svc.configd
00084000	24	24	16	-	rwX--	svc.configd
0008A000	6232	6232	2312	-	rwX--	[heap]
FE0FA000	8	8	8	-	rw--R	[anon]
...						
FF180000	848	848	-	-	r-x--	libc.so.1
FF264000	32	32	24	-	rwX--	libc.so.1
FF26C000	8	8	8	-	rwX--	libc.so.1
...						
FF3B0000	176	176	-	-	r-x--	ld.so.1
FF3EC000	8	8	8	-	rwX--	ld.so.1
FF3EE000	8	8	8	-	rwX--	ld.so.1
FFBFC000	16	16	8	-	rw---	[stack]

total Kb	9264	9160	2576	-		



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

I/Os Queued to a Swap Device

```
# iostat -xPnce 5 2
cpu
us sy wt id
1 5 0 95

          extended device statistics          ---- errors ---
r/s      w/s      kr/s      kw/s wait actv wsvc_t asvc_t  %w  %b s/w h/w trn tot device
1.0      0.2      7.0      0.7  0.0  0.0      8.8    9.0   0   1  0  0  0  0 c0t0d0s0
0.0      0.0      0.0      0.0  0.0  0.0     66.4   36.9   0   0  0  0  0  0 c0t0d0s1
0.0      0.0      0.0      0.0  0.0  0.0      0.0    0.0   0   0  0  0  0  0 c0t0d0s2
0.0      0.0      0.0      0.1  0.0  0.0     54.1   13.1   0   0  0  0  0  0 c0t0d0s4
cpu
us sy wt id
23 73 0 4

          extended device statistics          ---- errors ---
r/s      w/s      kr/s      kw/s wait actv wsvc_t asvc_t  %w  %b s/w h/w trn tot device
57.4     0.0    458.9      0.0  0.0  0.2      0.0    4.3   0  25  0  0  0  0 c0t0d0s0
0.0      0.0      0.0      0.0  0.0  0.0      0.0    0.0   0   0  0  0  0  0 c0t0d0s1
0.0      0.0      0.0      0.0  0.0  0.0      0.0    0.0   0   0  0  0  0  0 c0t0d0s2
0.0      0.0      0.0      0.0  0.0  0.0      0.0    0.0   0   0  0  0  0  0 c0t0d0s4
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Measuring the Distribution of Memory

Output from the `prstat -s rss` command:

PID	USERNAM	SIZE	RSS	STATE	PRI	NICE	TIME	CPU	PROCESS/NLWP
15058	noaccess	227M	150M	sleep	1	0	0:03:44	0.0%	java/18
1266	noaccess	215M	142M	sleep	35	0	0:04:33	0.0%	java/18
12615	noaccess	219M	141M	sleep	1	0	0:03:48	0.0%	java/18
17947	root	93M	30M	sleep	28	4	0:00:22	0.1%	gnome-terminal/2
13383	root	94M	24M	sleep	1	0	0:02:19	0.0%	pooldd/8
542	root	18M	13M	sleep	1	0	0:00:10	0.0%	fmd/23
683	root	14M	13M	sleep	33	0	0:01:03	0.0%	Xvnc/1
7	root	14M	11M	sleep	29	0	0:00:08	0.0%	svc.startd/15
...									

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Processes that have locked pages but have not accessed them appear to have a small resident set size (`rss`). You can check this with the `mem_hog` program used in the practice. The `pmap -x` command used on the program reflects the number of pages it has locked correctly.

Solaris Studio `discover` Utility

The `discover` tool instruments a binary executable. It detects attempts to write pass array bounds or access memory that has been freed.

- It is available for the SPARC platform only.
- The code must be optimized (compiled with `-O` switch).
- Run `discover` on the binary to instrument the code.
- The program reports memory violations that occur.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `discover` utility complements the tools bundled with Solaris Studio. It operates by adding data guard code to an optimized binary, checking memory-based operations as they occur. That is, it can find memory access violations that might go undetected with less intrusive testing.

A command-line sequence (without testing checks) looks like this:

```
# cc -O -o memErr memoryAccessError.c
# discover -w - ./memError
# ./memError
<memory access error output, when applicable>
```


Quiz

Hard swapping occurs when:

- a. At least two processes are on the run queue, waiting for CPU
- b. Free memory is less than the `desfree` parameter, on average, for more than 30 seconds
- c. The sum of `pageins` and `pageout` operations exceeds `maxpgio` or `freemem` is less than `minfree`
- d. All of the above

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: d

Quiz

The following command is a quick way to check the values for `lotsfree`, `desfree`, and `minfree` to see whether the default values are being used:

- a. `kstat -n system_pages`
- b. `prstat -s rss`
- c. `dtrace -n 'pgin { @[execname] = count() }'`
- d. `iostat -xPnce 5 2`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

The Solaris Studio `discover` utility can find memory access violations that might go undetected with less intrusive testing.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Summary

In this lesson, you should have learned how to:

- Describe system memory concepts
- Describe Virtual Memory 2
- Monitor memory utilization

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 9 Overview: Memory

This practice covers the following topics:

- Monitoring Memory Consumption
- Monitoring Memory Consumption with the DTrace Toolkit
- Monitoring Memory Access Error with the Solaris Studio `discover` Utility (Optional)

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

10

Disk I/O and the ZFS File System

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Identify cases that impact disk performance
- Describe disk performance monitoring
- Explain the basic concepts of Oracle Solaris ZFS
- Identify the layers of the ZFS architecture
- List storage pool performance considerations
- Describe ZFS tunable parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Disk I/O
- Disk monitoring
- ZFS and related concepts
- ZFS pool and file system considerations
- ZFS tuning parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Disk Performance Considerations

- Storage data characteristics
- Disk bottlenecks
- Disk utilization
- Disk saturation
- Disk configuration
- Disk file system
- HDDs versus SSDs

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Storage Data Characteristics

- Random or sequential data access

Application	Access Pattern
Operating system	Random, 40% read, 60% write
Mail server	Random, 67% read, 33% write
Database server	Random, 67% read, 33% write
Database log file	Sequential, 100% write
Web server	Random, 100% read
Backup	Sequential, 100% write
Restore	Sequential, 100% write
Video streaming	Sequential, 100% read

- Block size

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Accessing data sequentially is much faster than accessing it randomly because of the way in which traditional (not SSDs) disk hardware works. The seek operation, which occurs when the disk head positions itself at the right disk cylinder to access data requested, takes more time than any other part of the I/O process. Because reading randomly involves a higher number of seek operations than does sequential reading, random reads deliver a lower rate of throughput. The same is true for random writing. You might find it useful to examine your workload to determine whether it accesses data randomly or sequentially. If you find that disk access is predominantly random, you might want to pay particular attention to the activities being done and monitor for the emergence of a bottleneck. This table shows examples of typical data access patterns of various applications.

Data transfer always takes place in blocks during access to a disk subsystem. The size of the transferred data blocks depends on features of the operating system and the application and cannot be influenced by the user. For random access, throughput and response time (latency) increase with increasing block size on an almost linear basis, while the number of transactions decreases on a linear basis. With sequential access, the response time increases on an almost linear basis with an increasing block size, but throughput does not. Which throughput rates can therefore be achieved depends very much on the access pattern that an application generates on the disk subsystem.

Disk Bottlenecks

- A disk bottleneck occurs when data is processed by the remote and local applications faster than the data can be read from (or written to) disk.
- The `vmstat` command

```
# vmstat 5
  kthr    memory          page          disk          faults          cpu
 r  b  w swap  free re mf pi p fr de sr s0  s1  s2  s3 in sy  cs us sy id
12 14 0 11456 4120 1  41 19 1  3  0  2  5 188 12  8 48 112 130 4 14 82
  ^
  |
  |
# mpstat 2 30
CPU minf mjf xcal  intr  ithr  csw  icsw migr  smtx  srw  syscl  usr  sys  wt  idl
  0    0    2    69   399    79  731    99   81   10    0  4039  44    7    0  49
  1    0    3   234   624   502  809    90   67   28    0  7131  43    8    9  41
  2    0    2    32   122    21  962    60   36   28    0  3076  74    4    0  22
                                     ^
                                     |
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A disk bottleneck occurs when data is processed by the remote and local applications faster than the data can be read from disk. As a result, the overall throughput is limited by the disk I/O rate. The application and the network performance are negatively effected.

Disk Utilization

- Any level of disk utilization may degrade system performance because accessing disks is a slow activity.
- Utilization metrics are averages over time.
 - Short bursts of heavy utilization may be difficult to identify if averaged over longer intervals.
- Disk I/O is complex.
 - Utilization is useful as a starting point but additional investigation is often required.
- The `iostat -D` command

```
# iostat -D
      sd2          sd3          sd4          sd5
 rps wps util  rps wps util  rps wps util  rps wps util
  78  40  84   118 101  99   112  77  93   120  97 100
      ^           ^           ^           ^
      |           |           |           |
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Any level of disk utilization may degrade system performance because accessing disks is a slow activity. Typically, disk I/O is measured in milliseconds. Whether a level of disk utilization actually affects system performance greatly depends on how the application uses the disks and how the disk devices respond to requests.

Utilization metrics are averages over time. Often, applications and the OS access the disks in bursts (for example, when reading an entire file, when executing a new command, or when flushing writes). This can cause short bursts of heavy utilization, which may be difficult to identify if averaged over longer intervals.

Note that disk I/O is complex. It typically involves mechanical disk properties, buses, and caching and depends on the way applications use I/O. Condensing this information to a single utilization metric might be over simplifying. The utilization value is useful as a starting point but additional investigation is often required.

Disk Saturation

- A disk at saturation is constantly busy.
- New I/O transactions are unable to preempt the currently active disk operation.
- If `iostat` consistently reports `%w > 5`, the disk subsystem is too busy.

```
# iostat -xc 5 3
```

extended disk statistics										cpu			
disk	r/s	w/s	Kr/s	Kw/s	wait	actv	svc_t	%w	%b	us	sy	wt	id
sd0	3.6	3.0	23.7	22.7	0.1	0.2	53.4	15	10	5	95	21	0
								^					

- The `zfs_vdev_max_pending` parameter improves the synchronous write latency.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A sustained level of disk saturation usually means a performance problem. A disk at saturation is constantly busy, and new transactions are unable to preempt the currently active disk operation in the same way a thread can preempt the CPU. This means that new transactions suffer an unavoidable delay as they queue, waiting their turn.

If `iostat` consistently reports `%w > 5`, the disk subsystem is too busy.

The ZFS tuning parameter `zfs_vdev_max_pending` controls the maximum number of concurrent I/Os pending to each device. In a storage array where LUNs are made of a large number of disk drives (greater than 10 spindles), the ZFS queue can become a limiting factor on read IOPS. This behavior is one of the underlying reasoning for the best practice of presenting as many LUNs as there are backing spindles to the ZFS storage pool. On the other hand, when no separate intent log is in use and the pool is made of JBOD disks, using a small `zfs_vdev_max_pending` value, such as 10, can improve the synchronous write latency as those are competing for the disk resource.

Disk Configuration

To achieve the primary objective of avoiding disk bottlenecks:

- Avoid storing user data and the OS on the same disk.
- Spread disk I/O evenly across all available disks
 - Spread out the I/O traffic across more disks:
 - Stripe the file system
 - Split the data across additional file systems on other disks
 - Split the data across other servers
- Use ZFS because it stripes data across all disks.
- Consider creating UFS file systems on ZFS datasets.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The main objectives of disk I/O optimization is to avoid disk bottlenecks and protect your data. These goals can be achieved by following a few data layout best practices:

- Avoid storing user data and the OS on the same disk.
- Spread disk I/O evenly across all available disks. You need to avoid one disk being fully used up while others are idle.
- Spread out the I/O traffic across more disks. This can be done in hardware if the I/O subsystem includes a RAID controller, or in software by striping the file system (by using Solaris Volume Management/DiskSuite, Veritas Volume Manager, or ZFS), by splitting up the data across additional file systems on other disks, or even splitting the data across other servers.
- Use ZFS because it stripes data across all disks.
- Consider creating UFS file systems on ZFS datasets.

Disk File Systems

- File systems should provide:
 - Strong data protection
 - Good performance behavior
 - Optimal storage utilization
- Consider the application.
 - General purpose computing
 - File servers
 - Database servers
 - Legacy applications
- Combining file systems
- Third-party file systems

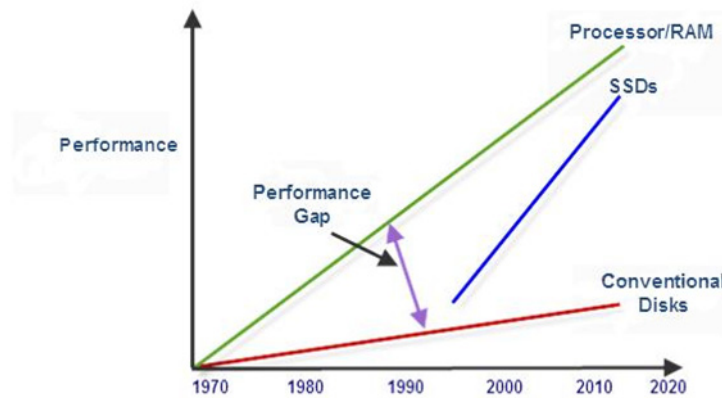
The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Solaris 11 OS supports a wide variety of storage file systems. Each file system has its strengths and weaknesses. A file system should provide strong data protection, good performance behavior, and choosing the best file system depends on your application. For general purpose computing and file servers, ZFS is a good choice. It provides many advanced features that make data storage more reliable and easier to manage. For example, using ZFS you can perform snapshots and roll back in a matter of seconds. It supports RAID so even damaged disks will not be able to damage your data. And it is absolutely space efficient. For an Oracle Database server, ASM is another good choice. For legacy applications, you might require UFS. If so, consider creating the UFS file system on a ZFS dataset. This way, you gain ZFS functionality such as snapshots, clone, and RAID for your legacy applications.

Solaris 11 also supports third-party file systems such as Veritas.

HDDs Versus SSDs

**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Over the past decades, processor and RAM speeds have increased dramatically. At the same time, conventional disk drive access times have marginally improved. This has resulted in a performance gap between processing technology and storage technology. This performance gap is particularly problematic on database servers, which typically carry out far more I/O transactions than other systems. High performance processors and RAM are often wasted as they wait on data access from devices that take several milliseconds to respond. When server processors wait on storage, users wait for their data.

Solid state disk (SSD) technology helps close this performance gap. SSD is any storage device that does not rely on mechanical parts to input and output data. Data is stored directly on memory chips and accessed from them. This generally results in storage speeds far greater than are currently possible with conventional, magnetic storage devices. SSDs are designed to solve the problem of I/O wait time by providing much faster access times.

Typically, SSD access times are less than one millisecond, whereas conventional storage might take seven milliseconds or longer. This results in increased database I/O transactions of up to 100 times that of conventional storage.

Oracle provides a robust Flash-based storage portfolio, from SSDs to high-performance all-Flash arrays with Flash-optimized databases and system software. The Sun Storage F5100 Flash Array delivers record 1.6M IOPS performance. Oracle Database 11g Release 2 is the world's first Flash-optimized database with dynamic Smart Flash Cache technology. Oracle's Sun Flash Accelerator F40 PCIe Card accelerates applications and server performance by reducing storage latencies and increasing I/O throughput for greater productivity and business response. The low latency, high throughput, and IOPS performance of the Sun Flash Accelerator F40 PCIe Card help servers and their applications run faster and more efficiently while reducing space and power.

Agenda

- Disk I/O
- **Disk monitoring**
- ZFS and related concepts
- ZFS pool and file system considerations
- ZFS tuning parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

iostat Command

```
# iostat -xc 5 3
```

	extended disk statistics									cpu			
disk	r/s	w/s	Kr/s	Kw/s	wait	actv	svc_t	%w	%b	us	sy	wt	id
sd0	3.6	3.0	23.7	22.7	0.1	0.2	53.4	15	10	5	95	21	0
sd1	7.2	1.7	36.5	8.0	0.0	0.3	49.7	3	22				
sd2	9.3	1.7	44.1	10.1	0.2	0.3	40.0	5	37				
sd3	13.4	1.6	49.7	14.6	0.2	0.4	31.2	5	41				

```
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `-x` option displays extended I/O status. The `-c` option reports the percentage of time the system has spent in user and system mode waiting for I/O.

The `iostat` command fields have the following meanings:

- `disk`: Name of the disk
- `r/s`: Reads per second
- `w/s`: Writes per second
- `Kr/s`: Kilobytes read per second
- `Kw/s`: Kilobytes written per second
- `wait`: Average number of transactions waiting for service (Q length)
- `actv`: Average number of transactions actively being serviced
- (removed from the queue but not yet completed)
- `%w`: Percent of time there are transactions waiting for service (queue nonempty)
- `%b`: Percent of time the disk is busy (transactions in progress)

vmstat Command

```
# vmstat 5
  kthr    memory          page          disk          faults          cpu
 r   b   w swap  free re  mf  pi  p fr de sr s0 s1 s2 s3  in sy  cs us sy id
12  14   0 11456 4120 1   41 19  1  3  0  2  0  4  0  0  48 112 130 4 14 82
14  18   1 10132 4280 0   44 14  0  0  0  0  0 23  0  0 211 230 144 3 35 62
15  12   1 10132 4616 0    0 20  0  0  0  0  0 19  0  0 150 172 146 3 33 64
17  15   1 10132 5292 0    0  9  0  0  0  0  0 21  0  0 165 105 130 1 21 78
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `vmstat` command fields have the following meanings:

- `kthr/r`: Run queue length
- `kthr/b`: Processes blocked while waiting for I/O
- `kthr/w`: Idle processes that have been swapped
- `memory/swap`: Free, unreserved swap space (Kb)
- `memory/free`: Free memory (Kb)
- `page/re`: Pages reclaimed from the free list
- `page/mf`: Minor faults (page in memory, but not mapped)
- `page/pi`: Paged in from swap (Kb/s)
- `page/po`: Paged out to swap (Kb/s)
- `page/fr`: Freed or destroyed (Kb/s)
- `page/de`: Freed after writes (Kb/s)
- `page/sr`: Scan rate (total number of pages scanned)
- `disk/s#`: Disk activity for disk # (I/Os per second)
- `faults/in`: Interrupts (per second)

- `faults/sy`: System calls (per second)
- `faults/cs`: Context switches (per second)
- `cpu/us`: User CPU time (%)
- `cpu/sy`: Kernel CPU time (%)
- `cpu/id`: Idle + I/O wait CPU time (%)

fsstat Command

```
# fsstat -F
new  name  name  attr  attr  lookup rddir  read read  write write
file remov chng  get   set   ops   ops   ops bytes ops bytes
    0     0     0     0     0     0     0     0     0     0     0 ufs
    0     0     0 42.3M     0 84.6M 2.84K 42.3M 13.9G    68 2.67K proc
    0     0     0     0     0     0     0     0     0     0     0 nfs
42.3M 42.3M   109 2.65G 1.47K 7.02G 84.6M 804M 76.8G 42.4M 106G zfs
    1     0     0 338M     0    59     2    18 3.47K     4    20 lofs
47.4K 45.9K 1.22K 427K    99 42.3M   151 982K 1003M 799K 751M tmpfs
    0     0     0   375     0     0     0    56 13.5K     0     0 mntfs
    0     0     0   121     0   103     6     0     0     0     0 autofs
    0     0     0     0     0     0     0     0     0     0     0 nfs3
    0     0     0     0     0     0     0     0     0     0     0 nfs4

# fsstat -i /u01
read  read  write write  rddir rddir  rwlock rwunlock
ops   bytes ops bytes ops bytes ops      ops
118 54.47K  22  220   31  531    67      67 /u01
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `-F` option reports on all available file system types. The `-i` option reports activity for kernel I/O operations.

The `fsstat` command fields have the following meanings:

- `read ops`: Number of data read operations
- `read bytes`: Bytes transferred data read operations
- `write ops`: Number of data write operations
- `write bytes`: Bytes transferred data write operations
- `rddir ops`: Number of read directory operations
- `rddir bytes`: Bytes transferred read directory operations
- `rwlock ops`: Number of internal file system lock operations
- `rwunlock ops`: Number of internal file system unlock operations

sar -d Command

```
# sar -d
SunOS server1 5.11 11.1 i86pc      011/17/2012

00:00:00   device    %busy    avque    r+w/s    blks/s    await    avserv

00:15:00    sd0         31        0.6       78     16102       4.9       5.3
           sd0,a         0        0.0         0         0       0.0       0.0
           sd0,b        31        0.6       78     16102       1.9       5.3
           sd0,c         0        0.0         0         1       1.6       1.3
...

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

By default, `sar` is not enabled in Solaris 11. You enable `sar` by adding the `sa1` and `sa2` scripts to the `cron` schedule. For example:

```
# crontab -e
0,15,30,45 * * * 0-6 /usr/lib/sa/sa1
55 23 * * 0-6 /usr/lib/sa/sa2 -A
```

The `sar -d` fields are as follows:

- `device`: This is the disk, or disk partition, being measured.
- `%busy`: This is the percentage of time the device is being read from or written to.
- `avque`: This is the average depth of the queue that is used to serialize disk activity. The higher the `avque` value, the more the occurrence of blocking.
- `r+w/s, blks/s`: This is disk activity per second in terms of read or write operations and disk blocks, respectively.
- `await`: This is the average time (in milliseconds) that a disk read or write operation waits before it is performed.
- `avserv`: This is the average time (in milliseconds) that a disk read or write operation takes to execute.

zpool iostat Command

The `zpool iostat` command displays I/O statistics for the given pools.

```
root@server1:~# zpool iostat -v rpool 1 1
```

pool	capacity		operations		bandwidth	
	alloc	free	read	write	read	write
-----	-----	-----	-----	-----	-----	-----
rpool	12.2G	220G	0	0	1.16K	1.12K
c7t0d0s1	12.2G	220G	0	0	1.16K	1.12K
-----	-----	-----	-----	-----	-----	-----

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `zpool iostat` command displays I/O statistics for the given pools. When given an interval, the statistics are printed every interval seconds until `Ctrl-C` is pressed. If no pools are specified, the statistics for every pool in the system is shown. If the count is specified, the command exits after count reports are printed.

DTrace Disk I/O One-Liners

- Show disk I/O size as distribution plots, by process name:

```
dtrace -n 'io:::start { @size[execname] = quantize(args[0]->b_bcount); }'
```

- Identify kernel stacks calling disk I/O:

```
dtrace -n 'io:::start { @[stack()] = count(); }'
```

- Trace errors along with disk and error number:

```
dtrace -n 'io:::done /args[0]->b_flags & B_ERROR/ { printf("%s err: %d",  
args[1]->dev_statname, args[0]->b_error); }'
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Note

For additional information about DTrace on-liners, see
“DTrace: Dynamic Tracing in Oracle Solaris, Mac OS X, and FreeBSD” by Brendan Gregg
and Jim Mauro at www.dtracebook.com.

Oracle I/O Numbers Calibration Tool (ORION)

- ORION is used to access I/O performance.
- ORION is designed to simulate Oracle database workloads.
- Simulated workloads:
 - Small random I/O
 - Large sequential I/O
 - Large random I/O
 - Mixed workloads
- Download URL:

<http://www.oracle.com/technetwork/topics/index-089595.html>

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ORION (Oracle I/O Calibration Tool) is a stand-alone tool for calibrating the I/O performance for storage systems that are intended to be used for Oracle databases. The calibration results are useful for understanding the performance capabilities of a storage system, either to uncover issues that would impact the performance of an Oracle database or to size a new database installation. Because ORION is a stand-alone tool, the user is not required to create and run an Oracle database.

With the goal of closely mimicking the Oracle database, ORION generates a synthetic I/O workload, using the same I/O software stack as Oracle. ORION can be configured to generate a wide range of I/O workloads, including ones that simulate OLTP and data warehouse workloads.

Agenda

- Disk I/O
- Disk monitoring
- **ZFS and related concepts**
- ZFS pool and file system considerations
- ZFS tuning parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris ZFS

ZFS is a complete rewrite of the file system from the ground up with the following goals:

- Provable Data Integrity: Checksums data and metadata
 - 256-bit checksum
 - Choice of algorithm
- Checksum contained in parent block pointer
- Immense capacity: 128-bit block numbers
- Simple administration
- The ZFS Administration Guide:

http://docs.oracle.com/cd/E26502_01/html/E29007/index.html

- Contains a wealth of information including administration, best practices, and tuning advice.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

With ZFS, all data and metadata is checksummed by using a user-selectable algorithm. Traditional file systems that do provide checksumming have performed it on a per-block basis. The traditional design means that certain failure modes, such as writing a complete block to an incorrect location, can result in properly checksummed data that is actually incorrect. ZFS checksums are stored in a way such that these failure modes are detected and can be recovered from gracefully. By storing the checksum in the parent block pointer, it will be in a different locality than the block that is being checksummed, and there is no extra overhead in retrieving the saved checksum.

The file system uses a 128-bit block number, allowing for 256 quadrillion zettabytes of storage. All metadata is allocated dynamically, so there is no need to preallocate on disk inodes at the time the file system is created.

ZFS provides a greatly simplified administration model. By using hierarchical file system layout, property inheritance, and auto management of mount points and NFS share semantics, ZFS makes it easy to create and manage file systems without needing multiple commands or editing configuration files.

ZFS uses the concept of storage pools to manage physical storage.

ZFS Design

ZFS is modeled after the virtual memory subsystem, which allows for:

- The integration of the Volume Manager
- Pooled storage: Elimination of the concept of volumes
- A transactional storage model: Live data, or metadata, is never overwritten, which means:
 - Copy-on-write semantics keep both data and metadata always consistent
 - No constraints on I/O order
 - Ease of taking snapshots

ORACLE

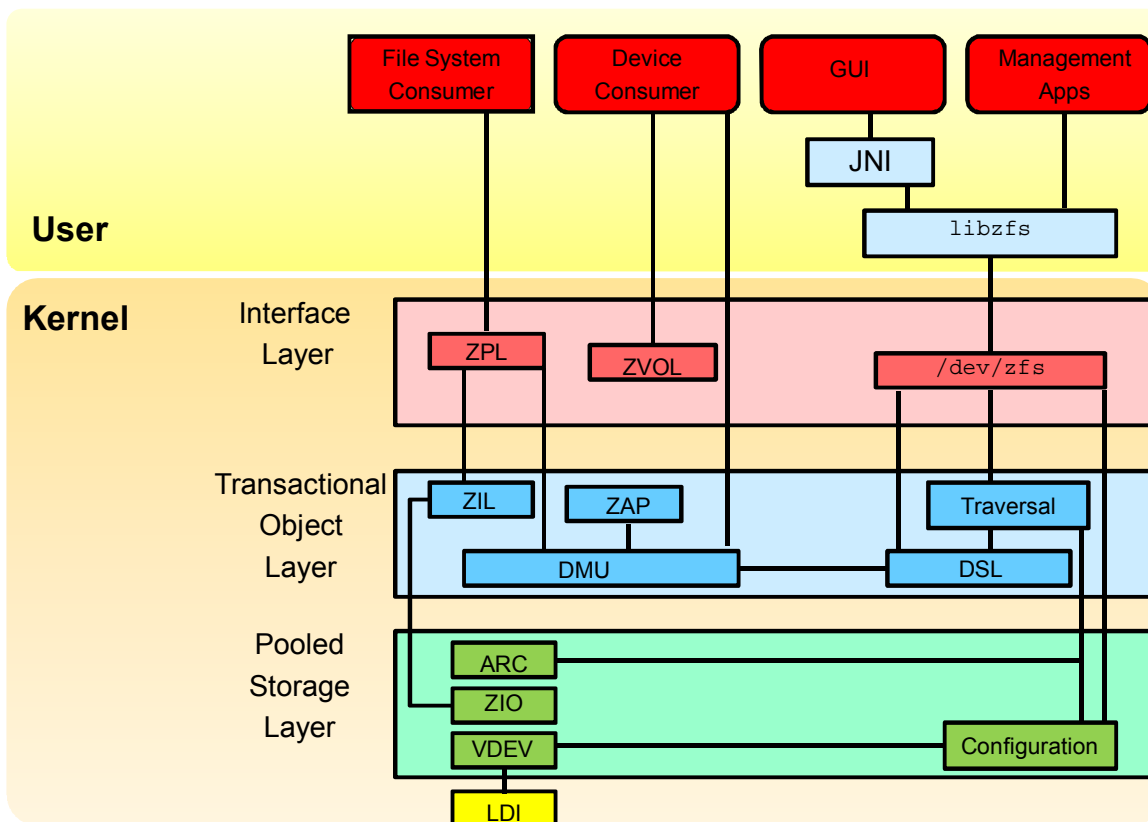
Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Historically, file systems were constructed on top of a single physical device. To address multiple devices and provide for data redundancy, the concept of a volume manager was introduced. This created the image of a single device so that file systems would not have to be modified to take advantage of multiple devices. This design ultimately prevented certain file system advances because the file system had no control over the physical placement of data on the virtualized volumes.

ZFS eliminates the volume management altogether. ZFS aggregates devices into a storage pool. The storage pool describes the physical characteristics of the storage (device layout, data redundancy, and so on) and acts as an arbitrary data store from which file systems can be created. File systems are no longer constrained to individual devices, allowing them to share space with all file systems in the pool.

File systems grow automatically within the space allocated to the storage pool. When new storage is added, all file systems within the pool can immediately use the additional space without additional work. The storage pool acts as a virtual memory system. When a memory DIMM is added to a system, the operating system does not force you to invoke commands to configure the memory and assign it to individual processes. All processes on the system automatically use the additional memory.

ZFS Architecture: Overview



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the diagram, you can see that the three main layers of ZFS consist of a ZFS POSIX Layer, the Data Management Unit, and the Storage Pool Allocator. If you were to draw a vertical line through the center of the diagram, interfaces to access the actual data lie for the most part on the left side and the interfaces that manage the metadata (administrative access) lie on the right.

The diagram illustrates the three main layers of ZFS:

- Interface Layer
- Transactional Object Layer
- Pooled Storage Layer

Interface Layer

The interface layer includes:

- ZPL (ZFS POSIX Layer): Interface between VFS interfaces and the DMU
- ZVOL (ZFS Emulated Volume): For creating logical volumes that are exported as block devices
- `/dev/zfs`: Point of control for `libzfs` (`zfs` and `zpool` commands)

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

These are applications that interact with ZFS solely through the POSIX file system APIs. Virtually every application falls into this category.

ZFS provides a means to create “emulated volumes.” These volumes are backed by storage from a storage pool, but appear as a normal device under `/dev`. The most common consumer is a kernel file system or target driver layered on top of the device and passed through the generic OpenSolaris VFS layer to the ZPL.

These applications are those that manipulate ZFS file system or Storage pools. The two main applications are `zpool(1M)` and `zfs(1M)`.

The `zpool(1M)` command is responsible for creating and managing ZFS storage pools through calls to `libzfs`.

The `zfs(1M)` command is responsible for creating and managing ZFS file systems through calls to `libzfs`.

The `libzfs` library is the primary interface for management applications to interact with the ZFS kernel module, `/dev/zfs`.

Transactional Object Layer

The transactional object layer defines the copy-on-write semantics of ZFS. It consists of the:

- DMU: The Data Management Unit
- ZIL: The ZFS Intent Log
- ZAP: The ZFS Attribute Processor for directories
- Traversal: For disk scrubbing and resilvering
- DSL: The Dataset and Snapshot Layer

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ZFS is a transactional file system, which means that the file system state is always consistent on disk. Traditional file systems overwrite data in place, which means that if the machine loses power, for example, between the time a data block is allocated and when it is linked into a directory, the file system will be left in an inconsistent state. Historically, this problem was solved by using the `fsck` command. More recently, file systems have introduced the concept of journaling. The journaling process records action in a separate journal, which can then be replayed safely if a system crash occurs. This process introduces overhead, because the data must be written twice, and often results in a new set of problems, such as when the journal cannot be replayed properly.

With a transactional file system, data is managed by using copy-on-write semantics. Data is never overwritten, and any sequence of operations is either entirely committed or entirely ignored. This mechanism means that the file system can never be corrupted through accidental loss of power or a system crash. So, no need for a `fsck` equivalent exists. While the most recently written pieces of data might be lost, the file system itself will always be consistent. In addition, synchronous data (written by using the `O_DSINC` flag) is always guaranteed to be written before returning, so it is never lost.

Data Management Unit

The DMU is responsible for presenting a transactional object model.

- DMU consumes blocks and groups them into objects.
- Objects can be grouped by the DMU into object sets.
- Object sets are used in ZFS to group related objects, such as a file system, snapshot, clone, or volume.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ZFS Intent Log (ZIL)

- The majority of data is not written to disk immediately; otherwise, performance would be slow.
- For `fsync`, `O_DSYNC`, or other synchronous requirements:
 - ZIL saves transaction records of system calls with enough information to be able to replay them
 - These are stored in memory until:
 - Transaction group (`txg`) commits to stable pool
 - A `fsync`, `O_DSYNC`, or other synchronous requirement
- In the event of a panic or power failure, the log records (transactions) are replayed.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There is one ZIL per file system. Its on-disk (pool) format consists of three parts:

- ZIL header
- ZIL blocks
- ZIL records

A log record holds a system call transaction. Log blocks can hold many log records and the blocks are chained together. The ZIL header points to the first block in the chain.

ZAP Layer

ZAP is the ZFS Attribute Processor.

- Built on the DMU
- Uses scalable hash algorithms to create (name, object) associations within an objset
- Used to implement directories
- Used extensively throughout the DSL as well

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The ZFS Attribute Processor is most commonly used to implement directories within the ZFS POSIX Layer (ZPL).

Dataset and Snapshot Layer (DSL)

- The DSL aggregates DMU objsets into a hierarchical namespace.
- The DSL manages relationships-between and properties-of object sets.
- Four kinds of object sets are represented in the DSL as a data set:
 - ZFS file system
 - ZFS clone
 - ZFS snapshot
 - ZFS volume

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A data set manages:

- Space consumption statistics for an object set
- Object set location information
- Any snapshot's interdependencies

Data sets are grouped hierarchically into collections called Dataset Directories.

Dataset Directories manage a related grouping of data sets and the properties associated with that grouping. A DSL directory always has exactly one "active data set." All other data sets under the DSL directory are related to the "active" data set through snapshots, clones, or child/parent dependencies.

Pooled Storage Layer

The pooled storage layer forms the basis of treating disks as virtual storage. It consists of the following sections:

- Virtual Devices (VDEV): Method of arranging and accessing devices
- ZFS I/O Pipeline (ZIO): Where data is checksummed and optionally compressed
- Adaptive Replacement Cache (ARC)
- Configuration: Maintains pool configuration information; includes routines to create and destroy pools

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The virtual device subsystem provides a unified method of arranging and accessing devices. There are two types of virtual devices: physical virtual devices (called leaf `vdevs`) and logical virtual devices (called interior `vdevs`). A physical `vdev` is a writable media block device (a disk, for example). A logical `vdev` is a grouping of physical `vdevs` (mirror or RAID-Z).

The ZIO pipeline is where all data must pass when going to or from the disk. It is responsible for translation of Device Virtual Addresses (DVAs) into logical locations on a `vdev`, as well as checksumming and compressing data as necessary.

The configuration portion is the public interface between the ZIO and `vdev` layers. It includes routines to create and destroy pools from their configuration information.

ZFS ARC Cache Introduction

ZFS uses a modified version of an ARC to provide its primary caching needs.

- Cache is layered between the DMU and the SPA
- Allows file systems to share their cached data with their snapshots and clones
- Takes into account both frequency and recency of use

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ZFS uses a modified version of an ARC to provide its primary caching needs. This cache is layered between the DMU and the SPA and so acts at the virtual block level. This allows file systems to share their cached data with their snapshots and clones.

See the ZFS Tuning section of this lesson for references of sizing the ARC.

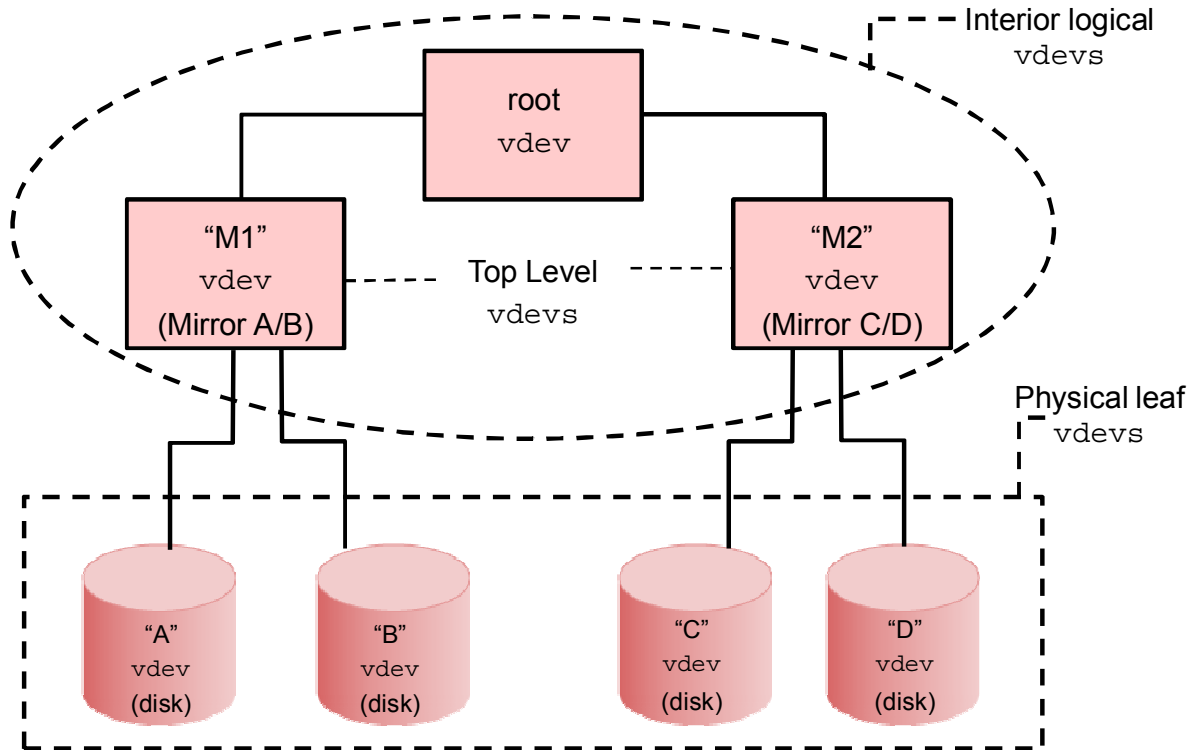
ZFS Data Structures

- The `vdev` tree: Keeps track of devices within the storage pool
- `vdev` label: Contains disk label, uberblocks, description of related `vdevs`
- The `dnode` and `znode`: Equivalent to `inode` in UFS
- The Block Pointer: Keeps block location, checksum, compression, and so on
- The uberblock: Equivalent to the superblock in UFS

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

vdev Tree

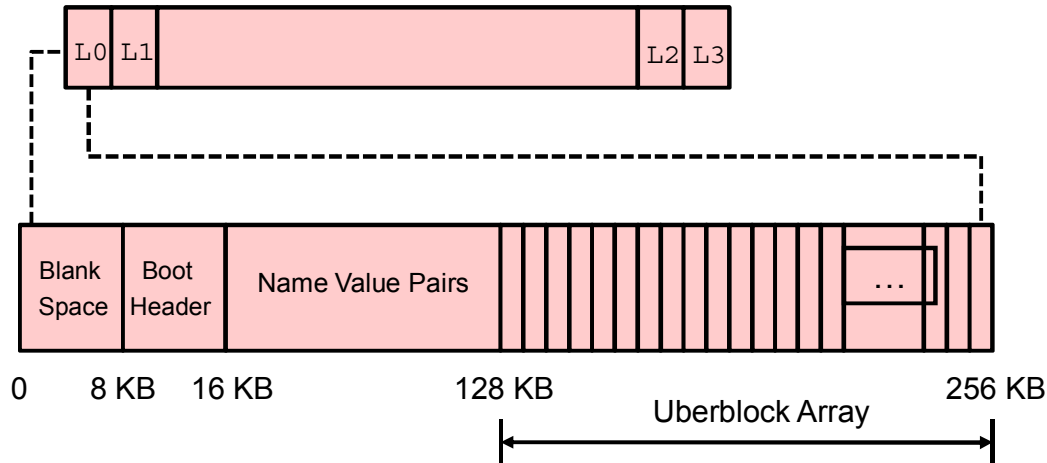


ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Virtual devices form a tree, with a single root vdev.

vdev Label



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Each physical `vdev` within a storage pool contains a 256-KB structure called a `vdev` label. When a device is added to the pool, ZFS places two labels at the front of the device and two labels at the back of the device.

The location of the `vdev` labels is fixed at the time the device is added to the pool. Thus, the `vdev` label does not have copy-on-write semantics like everything else in ZFS. To ensure that ZFS always has access to its labels, a staged approach is used during update.

Each `vdev` label consists of:

- Blank 8-KB space for VTOC or EFI that is placed at the beginning of the physical disk
- An 8-KB boot header
- The Name-Value Pair List, which describes all the related `vdevs` under the same top-level `vdev`
- An array of 128 uberblocks

vdev Label

The information stored in the name-value pairs for related `vdevs` include:

- Transaction group number used to write this label: `"txg"`
- Global unique identifier (guid) for the pool: `"pool_guid"`
- Global unique identifier (guid) for the top-level `vdev`: `"top_guid"`
- Global unique identifier (guid) for this `vdev`: `"guid"`
- Descriptions of other related `vdevs` under this top level `vdev`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered within a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Uberblock

- The uberblock is the portion of the label containing information necessary to access the contents of the pool.
- The uberblock is the equivalent to the superblock in UFS.
- Only one uberblock in the pool is active at any point in time.
- Uberblocks are written in a round-robin fashion across the various `vdevs` with the pool.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Immediately following the `nvpair` lists in the `vdev` label is an array of 128 uberblocks.

General ZFS Administration

- ZFS administration is performed while data is online.
- ZFS file systems are mounted automatically when created.
- ZFS file systems do not have to be mounted by modifying the `/etc/vfstab` file.
- The `zpool(1M)` command is used to create, list, and modify storage pools.
- The `zfs(1M)` command is used to configure a data set (file system) within a storage pool.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Note

For most ZFS administration tasks, see the `zfs.1m` and `zpool.1m` man pages. For more detailed documentation, see the *ZFS Administration Guide*.

You can use "`iostat -En`" to see drives.

ZFS Limitations

- ZFS does not provide a comprehensive backup or restore utility. You can use the `zfs send` and `zfs receive` commands to capture ZFS data streams.
- Capacity expansion is normally achieved by adding groups of disks as a `vdev`.
- It is currently not possible to reduce the pool capacity.
- You cannot mix `vdev` types in a `zpool`. For example, if you have a striped ZFS pool consisting of disks on a SAN, you cannot add the local disks as a mirrored `vdev`.
- Reconfiguring storage requires copying data offline, destroying the pool, and re-creating the pool with the new policy.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Capacity expansion is normally achieved by adding groups of disks as a `vdev` (stripe, RAID-Z, RAID-Z2, or mirrored). Newly written data will dynamically start to use all available `vdevs`. It is also possible to expand the array by iteratively swapping each drive in the array with a bigger drive and waiting for ZFS to heal itself—the heal time will depend on the amount of stored information, not the disk size. The new free space will not be available until all the disks have been swapped.

It is currently not possible to reduce the number of `vdevs` in a pool nor otherwise reduce pool capacity. However, this capability is currently being worked on by the ZFS team.

ZFS is not a native cluster, distributed, or parallel file system and cannot provide concurrent access from multiple hosts, as ZFS is a local file system.

Agenda

- Disk I/O
- Disk monitoring
- ZFS and related concepts
- **ZFS pool and file system considerations**
- ZFS tuning parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

General Storage Tuning

- Tuning is often evil and should rarely be done.
- Default values are set to get the most effect of the tuning on the software that they supply.
- Tuning might help a given workload. It could possibly degrade some other aspects of performance.
- Over time, tuning recommendations might become stale.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered within a solid red rectangular bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Customers are leery of changing a tuning that is in place and the net effect is a worse product than what it could be. Moreover, tuning enabled on a given system might spread to other systems, where it might not be warranted at all.

Nevertheless, it is understood that customers who carefully observe their own system may understand aspects of their workloads that cannot be anticipated by the defaults. In such cases, the following tuning information may be applied, provided that you work to carefully understand its effects.

ZFS Tuning Guidelines

- Keep the system up-to-date with latest Solaris releases and patches.
- Confirm that your controller honors cache flush commands.
- Cap memory to actual system workload requirements.
- Perform regular backups.
- Consider using ZFS RAID over JBOD-mode for storage.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Here are some important general tuning guidelines that impact ZFS performance and data integrity:

- Keep the system up-to-date with latest Solaris releases and patches.
- Confirm that your controller honors cache flush commands so that you know your data is safely written. This is generally not a problem on Oracle hardware, but it is good practice to confirm that your hardware's cache flushing setting is enabled.
- Cap memory to actual system workload requirements.
 - With a known application memory footprint, such as for a database application, you might cap the ARC size so that the application will not need to reclaim its necessary memory from the ZFS cache.
 - Consider deduplication memory requirements.
- Perform regular backups. No disk storage methodology is immune to hardware failures, power failures, or disconnected cables. Make sure you back up your data on a regular basis. If your data is important, it should be backed up.

- Consider using JBOD-mode for storage arrays rather than hardware RAID so that ZFS can manage the storage and the redundancy. ZFS redundancy has many benefits. For production environments, configure ZFS so that it can repair data inconsistencies. Use ZFS redundancy, such as RAID-Z, RAID-Z-2, RAID-Z-3, mirror, regardless of the RAID level implemented on the underlying storage device. With such redundancy, faults in the underlying storage device or its connections to the host can be discovered and repaired by ZFS.

General Storage Pools Considerations

- Use whole disks when creating storage pools.
- Use ZFS redundancy so that ZFS can repair data inconsistencies.
 - Do not create single `vdev` pools.
- Use hot spares to reduce down time due to hardware failures.
- Use similar size disks so that I/O is balanced across devices.
- Do not create RAIDZ pools on 4-K disks.
- Consider creating a small root pool and larger data pools to support faster system recovery.
- Starting with Solaris 11.1, rpool can be on EFI-labeled disks.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Root Pool Considerations

- Do not create a root pools on removable media.
- Do not rename the root pool after it is created by an initial installation.
- The root pool cannot have a separate log device.
- Pool properties can be set during an AI installation.
- A second root pool can be created:
 - Single-disk and mirror configurations are supported.
 - RAID-Z configurations are not supported.
- When creating a second root pool, use disk slices.

```
# zpool create rpool mirror c0t1d0s0 c0t2d0s0
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Non-Root Pool Considerations

- ZFS works best without any additional volume management software.
- Create non-root pools with whole disks.

```
# zpool create data1 c0t1d0
```

- Create redundant pool configurations across multiple controllers with multiple VDEVs.
 - Consider mirrored storage pools with two VDEVs.

```
# zpool create data2 mirror c1t0d0 c2t0d0 \  
mirror c3t0d0 c4t0d0
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ZFS works best without any additional volume management software.

For better performance, use individual disks or at least LUNs made up of just a few disks. By providing ZFS with more visibility into the LUNs setup, ZFS is able to make better I/O scheduling decisions.

Create redundant pool configurations across multiple controllers to reduce down time due to a controller failure.

ZFS RAID-Z: Examples

- RAID-Z configurations
 - Consider RAID-Z storage pools with two VDEVs.

```
# zpool create data3 raidz1 c1t1d0 c2t1d0 \
raidz1 c3t1d0 c4t1d0
```

- Consider RAIDZ-2 storage pools with two VDEVs.

```
# zpool create data4 raidz2 c1t2d0 c2t2d0 c3t2d0 \
c4t2d0 c5t2d0 c6t2d0 raidz2 c1t3d0 c2t3d0 c3t3d0 \
c4t3d0 c5t3d0 c6t3d0
```

- Consider RAIDZ-3 storage pools.

```
# zpool create data5 raidz3 c1t2d0 c2t2d0 c3t2d0 \
c4t2d0 c5t2d0 c6t2d0 c1t3d0 c2t3d0 c3t3d0
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Non-Root Pool Considerations

RAID-Z storage pools can be created with three parity strategies, where parity equals 1 (raidz), 2 (raidz2), or 3 (raidz3). A RAID-Z configuration maximizes disk space and generally performs well when data is written and read in large chunks (128 K or more).

- Consider a single-parity RAID-Z (raidz) configuration with two VDEVs of three disks (2+1) each.
- A RAIDZ-2 configuration offers better data availability, and performs similarly to RAID-Z. RAIDZ-2 has significantly better mean time to data loss (MTTDL) than either RAID-Z or two-way mirrors. This example shows a double-parity RAID-Z (raidz2) configuration at six disks (4+2).
- A RAIDZ-3 configuration maximizes disk space and offers excellent availability because it can withstand three disk failures. This example shows a triple-parity RAID-Z (raidz3) configuration at nine disks (6+3).

Network-Attached Storage Considerations

- Determine whether the network connection meets your data access requirements.
 - Network might not be appropriate for the root pool.
- The disk array must not flush its cache after a flush write cache request is issued by ZFS.
- Use whole disks, not disk slices, as storage pool devices.
- Create one LUN for each physical disk in the array.
- A storage array that uses dynamic provisioning software to implement virtual space allocation is not recommended for Oracle Solaris ZFS.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If you create a pool on SAN devices and the network connection is slow, the pool's devices might be UNAVAIL for a period of time. You need to assess whether the network connection is appropriate for providing your data in a continuous fashion. Also, consider that if you are using SAN devices for your root pool, they might not be available as soon as the system is booted and the root pool's devices might also be UNAVAIL.

Confirm with your array vendor that the disk array is not flushing its cache after a flush write cache request is issued by ZFS.

Use whole disks, not disk slices, as storage pool devices so that Oracle Solaris ZFS activates the local small disk caches, which get flushed at appropriate times.

For best performance, create one LUN for each physical disk in the array. Using only one large LUN can cause ZFS to queue up too few read I/O operations to actually drive the storage to optimal performance. Conversely, using many small LUNs could have the effect of swamping the storage with a large number of pending read I/O operations.

A storage array that uses dynamic (or thin) provisioning software to implement virtual space allocation is not recommended for Oracle Solaris ZFS. When Oracle Solaris ZFS writes the modified data to free space, it writes to the entire LUN. The Oracle Solaris ZFS write process allocates all the virtual space from the storage array's point of view, which negates the benefit of dynamic provisioning.

ZFS Storage Pool: Maintenance and Monitoring

- Make sure that the pool capacity is below 80% for best performance.
- Monitor pool health
 - Redundant pools: Monitor pool with `zpool status` and `fmddump` on a weekly basis
 - Non-redundant pools: Monitor pool with `zpool status` and `fmddump` on a biweekly basis
- Run `zpool scrub` on a regular basis.
- Monitoring pool for device failures with `zpool status` and `fmddump`.
 - Pool device is `UNAVAIL` or `OFFLINE`.
- Use the `smtp-notify` service.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Make sure that the pool capacity is below 80% for best performance. Pool performance can degrade when a pool is full and file systems are updated frequently, such as on a busy mail server. Full pools might cause a performance penalty, but no other issues. If the primary workload is immutable files, keep the pool in the 95-96% utilization range.

Monitor pool health

- Redundant pools: Monitor pool with `zpool status` and `fmddump` on a weekly basis.
- Non-redundant pools: Monitor pool with `zpool status` and `fmddump` on a biweekly basis.
- Run `zpool scrub` on a regular basis to identify data integrity problems.
- If you have consumer-quality drives, consider a weekly scrubbing schedule.
- If you have datacenter-quality drives, consider a monthly scrubbing schedule.

You should also run a scrub before replacing devices or temporarily reducing a pool's redundancy to ensure that all devices are currently operational.

Monitor pool or device failures with `zpool status` and `fmddump` or `fmddump -eV` to see whether any device faults or errors have occurred.

- **Redundant pools:** Monitor pool health with `zpool status` and `fmddump` on a weekly basis
- **Non-redundant pools:** Monitor pool health with `zpool status` and `fmddump` on a biweekly basis

Pool device is `UNAVAIL` or `OFFLINE` : If a pool device is not available, verify that the device is listed in the `format` command output. If the device is not listed in the `format` output, it will not be visible to ZFS. If a pool device has `UNAVAIL` or `OFFLINE`, it generally means that the device has failed or the cable has been disconnected or some other hardware problem, such as a bad cable or bad controller has caused the device to be inaccessible.

ZFS File System Considerations

- Create one file system per user for home directories.
- Consider using file system quotas.
- Consider using user and group quotas.
- Use ZFS property inheritance.
- Oracle Database
 - Match the ZFS `recordsize` property to the Oracle `db_block_size`.
 - Create database table and index file systems in main database pool by using an 8-KB `recordsize` and the default `primarycache` value.
 - Create temp data and undo table space file systems in the main database pool by using default `recordsize` and `primarycache` values.
 - Create archive log file system in the archive pool, enabling compression and the default `recordsize` value and `primarycache` set to metadata.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Disk I/O
- Disk monitoring
- ZFS and related concepts
- ZFS pool and file system considerations
- ZFS tuning parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ZFS ARC Cache

- The ARC is where ZFS caches data from all active storage pools.
- The ARC grows and consumes memory on the principle that no need exists to return data to the system while there is free memory.
- When outside memory pressure exists, the ARC releases its hold on memory.
- The established mechanism works reasonably well and may not warrant tuning.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered within a solid red rectangular bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The ZFS adaptive replacement cache (ARC) tries to use most of a system's available memory to cache file system data. The default is to use all of physical memory except 1 GB. As memory pressure increases, the ARC relinquishes memory.

In general, limiting the ARC is wasteful if the memory that now goes unused by ZFS is also unused by other system components. Note that non-ZFS file systems typically manage to cache data in what is nevertheless reported as free memory by the system.

ZFS ARC Cache

There are some limitations to ARC and these are highlighted during the following situations:

- If a memory requirement is large and well-defined
- Applications use large pages
- If dynamic reconfiguration is needed
- If a system is running another non-ZFS file system

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

For these cases, it can be desirable to limit the ARC. The trade-off is to consider that limiting this memory footprint means that the ARC is unable to cache as much file system data, and this limit could impact performance. There is no easy way to foretell whether limiting the ARC degrades performance.

Viewing ZFS ARC Statistics

```

root@server1:~# echo "::arc" | mdb -k
hits                        =          461679
misses                      =           28644
demand_data_hits           =        109103
demand_data_misses         =           3711
demand_metadata_hits       =        325247
demand_metadata_misses     =           8717
prefetch_data_hits         =           36
prefetch_data_misses       =          1495
prefetch_metadata_hits     =        27293
prefetch_metadata_misses   =        14721
mru_hits                   =        44464
mru_ghost_hits             =              0
mfu_hits                   =        302224
mfu_ghost_hits             =              0
...

```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Viewing ZFS ARC Statistics

```
root@server1:~# kstat -p "zfs:0:arcstats:"
zfs:0:arcstats:buf_size 8786768
zfs:0:arcstats:c          16045658112
zfs:0:arcstats:c_max      16045658112
zfs:0:arcstats:c_min      67108864
zfs:0:arcstats:class      misc
zfs:0:arcstats:crttime    100.367237953
zfs:0:arcstats:data_size   426647360
zfs:0:arcstats:deleted    0
zfs:0:arcstats:demand_data_hits 109189
zfs:0:arcstats:demand_data_misses 3711
zfs:0:arcstats:demand_metadata_hits 325266
zfs:0:arcstats:demand_metadata_misses 8717
zfs:0:arcstats:hash_chain_max 3
zfs:0:arcstats:hash_chains 18
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ZFS ARC Cache Tuning Parameters

- ZFS ARC parameters:
 - `zfs_arc_max`
 - `zfs_arc_min`
- For example, if an application needs 5 GB of memory on a system with 36 GB of memory, you can set the `arc` maximum to 30 GB (0x780000000 or 32212254720 B).
 - Set the `zfs_arc_max` parameter in the `/etc/system` file by using either of the following:

```
set zfs_arc_max=0x780000000
```

```
set zfs_arc_max=32212254720
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `zfs_arc_max` parameter determines the maximum size of the ZFS Adaptive Replacement Cache (ARC). If a future memory requirement is significantly large and well-defined, you might consider reducing the value of this parameter to cap the ARC so that it does not compete with the memory requirement. For example, if you know that a future workload requires 20% of memory, it makes sense to cap the ARC such that it does not consume more than the remaining 80% of memory.

The `zfs_arc_min` parameter determines the minimum size of the ZFS ARC. When a system's workload demand for memory fluctuates, the ZFS ARC caches data at a period of weak demand and then shrinks at a period of strong demand. However, ZFS does not shrink below the value of `zfs_arc_min`. Generally, you do not need to change the default value.

Additional ZFS Tuning Parameters

- ZFS file-level prefetch
 - `zfs_prefetch_disable`
- ZFS device I/O queue depth
 - `zfs_vdev_max_pending`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Here are additional key ZFS tuning parameters:

- The `zfs_prefetch_disable` parameter determines a file-level prefetching mechanism called `zfetch`. This mechanism looks at the patterns of reads to files and anticipates on some reads, thereby reducing application wait times. If the results of `er_kernel` show significant time in `zfetch_*` functions, or if lock profiling with `lockstat` shows contention around `zfetch` locks, disabling file-level prefetching should be considered. The default is enabled (0).
- The `zfs_vdev_max_pending` parameter controls the maximum number of concurrent I/Os pending to each device.

In a storage array where LUNs are made of a large number of disk drives, the ZFS queue can become a limiting factor on read IOPS. This behavior is one of the underlying reasoning for the best practice of presenting as many LUNS as there are backing spindles to the ZFS storage pool. However, when no separate intent log is in use and the pool is made of JBOD disks, using a small `zfs_vdev_max_pending` value, such as 10, can improve the synchronous write latency as those are competing for the disk resource. The default is 10.

Note: Using separate intent log devices can alleviate the need to tune this parameter for loads that are synchronously write intensive since those synchronous writes are not competing with a deep queue of non-synchronous writes.

Additional ZFS Tuning Parameters

- ZFS and cache flushing
 - `zfs_nocacheflush`
- ZFS metadata compression
 - `zfs_mdcomp_disable`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

- The `zfs_nocacheflush` parameter controls (enable or disable) ZFS write cache flushes for the entire system. Cache flush tuning helps some SSD performance when used as log devices. The default is enabled (0).
- The `zfs_mdcomp_disable` parameter controls compression of ZFS metadata (indirect blocks only). ZFS data block compression is controlled by the ZFS compression property that can be set per file system. Metadata compression should be enabled at all times. The default is enabled (0).

Viewing ZFS Kernel Parameters

```
root@server1:~# echo "::zfs_params" | mdb -k
arc_reduce_dnlc_percent = 0x3
zfs_arc_max = 0x0
zfs_arc_min = 0x0
arc_shrink_shift = 0x7
zfs_mdcomp_disable = 0x0
zfs_prefetch_disable = 0x0
zfetech_max_streams = 0x8
zfetech_min_sec_reap = 0x2
zfetech_block_cap = 0x100
zfetech_array_rd_sz = 0x100000
zfs_default_bs = 0x9
zfs_default_ibs = 0xe
metaslab_aliquot = 0x80000
...
```

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to its upper right. The logo is white and is positioned on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Quiz

In ZFS, the equivalent of the `fsk` command is the `zpool` command.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

The command used to configure a data set (file system) within a storage pool:

- a. `zfs`
- b. `zpool`
- c. `tunefs`
- d. None of the above

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Summary

In this lesson, you should have learned how to:

- Identify cases that impact disk performance
- Describe disk performance monitoring
- Explain the basic concepts of Oracle Solaris ZFS
- Identify the layers of the ZFS architecture
- List storage pool performance considerations
- Describe ZFS tunable parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 10 Overview: Disk I/O and the ZFS File System

This practice covers the following topics:

- Benchmarking System Disk I/O
- Monitoring Disk I/O
- Limiting the Size of the ARC in ZFS

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

11

Solaris 11 Network Tuning

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- Describe terms used for network analysis
- Describe network utilization
- Understand the effects of misconfigured components
- Describe the differences between Solaris 10 and Solaris 11 networking
- Describe the benefits of IPMP
- Describe the benefits of link aggregation
- Describe network monitoring commands that are commonly used in Solaris 11
- Describe network tuning parameters that are commonly used in Solaris 11

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Network concepts
- Oracle Solaris 11 networking
- Network configuration
- Monitoring network performance
- Tunable parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Relevance

Discussion: The following questions are relevant to understanding the content of this lesson:

- Why is it important to tune network servers?
- Which are the tunable elements in a network?
- What settings are relevant to network tuning?

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Introducing Network Performance

- The scale of client connection requests to a server resource
- The varying link speeds, delays, and error rates of each client's route to the server
- The tuning of a system's TCP/IP protocol stack, in particular the connection-oriented transport layer

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Network performance may or may not be a primary responsibility for many system administrators. While the overall performance of networked systems relies on workstation hardware and operating system performance, the efficient flow of data between systems relies on proper networking, too. The skills and experience that drive the design and administration of efficient networks differ from the skills needed for administering systems. Many system administrators therefore take contributory roles to administering networks.

The goal is still simple, however—to move the data as fast as possible from the application out to the network interface card (NIC), onto the network, and then to the target machine. To tune the network traffic effectively, several factors must be taken into account, including those shown in the preceding slide.

In this lesson, a general model for how the Transmission Control Protocol (TCP) engine adjusts over time to the often changing conditions of client demands is described. It also describes some key tunable parameters that affect TCP engine behavior.

Terms Used for Network Analysis

- Packets
- Bytes
- Utilization
- Saturation
- Errors
- Link status
- By-process
- TCP
- IP
- ICMP

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The following terms are used as they relate to network analysis.

- **Packets:** Network interface packet counts can be fetched from `netstat -i` and can indicate approximate network activity.
- **Bytes:** Measuring throughput in terms of bytes is useful because interface maximum throughput is measured in comparable terms, bits/second.
- **Utilization:** Heavy network use can degrade application response time.
- **Saturation:** Network applications can experience delays once an interface is saturated.
- **Errors:** Error counts: collisions, input errors, and output errors
- **Link Status:** Used to describe the state of the interface
- **By-process:** Network I/O by-process can be analyzed with DTrace.
- **TCP:** Various TCP statistics are kept for MIB-II, plus additional statistics.
- **IP:** Various IP statistics are kept for MIB-II, plus additional statistics.
- **ICMP:** Commands like `ping` and `traceroute` that make use of ICMP can inform about the network surroundings.

Packets

```
# netstat -i 1
```

input	hme0	output			input	(Total)	output		
packets	errs	packets	errs	colls	packets	errs	packets	errs	colls
13234076	0	4124358	0	0	13415882	0	4306164	0	15
12	0	0	16	0	13	0	0		
20	0	20	0	0	20	0	20	0	0
18	0	11	0	0	18	0	11	0	0
17	0	15	0	0	17	0	15	0	0
17	0	15	0	0	17	0	15	0	0
25	0	24	0	0	29	0	28	0	0
20	0	18	0	0	20	0	18	0	0
18	0	17	0	0	18	0	17	0	0
21	0	19	0	0	21	0	19	0	0
19	0	17	0	0	19	0	17	0	0
19	0	17	0	0	19	0	17	0	0
23	0	19	0	0	23	0	19	0	0
19	0	16	0	0	19	0	16	0	0
18	0	18	0	0	18	0	18	0	0
14	0	14	0	0	14	0	14	0	0

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the above example, you can see that the `hme0` interface has very few errors, which is useful to know. You cannot tell if this interface is at 100 percent utilization or 1 percent utilization; all it tells us is that the traffic is occurring.

Network Utilization

- Network events
- Network cards
- Network speeds
- Overhead
- High utilization

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The following list describes the effects of network utilization:

- Network events, like disk events, are slow. A client application that is heavily network bound will experience delays.
- A network card that is at 100 percent utilization will most likely degrade application performance.
- Dividing the current KB/sec by the speed of the network card can provide a useful measure of network utilization.
- Using only KB/sec in a utilization calculation fails to account for performance overheads.
- Unexpected high utilization may be caused when auto-negotiation has failed by selecting a slower speed.

Network Errors

- output:coll: Collisions, normal in small doses
- input:errs: A frame failed its frame check sequence
- output:errs: Late collisions. A collision occurred after the first 64 bytes were sent.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Errors can occur from network collisions and as such are common. Three types of errors were visible in the previous `netstat -i` output.

The last two types of errors can be caused by bad wiring, faulty cards, auto-negotiation problems, and electromagnetic interference.

Effects of Misconfigured Components

- Poor performance
- No error statistics
- Need to scrutinize network settings
- Need to check all interface settings
- Need to check configuration settings

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Sometimes, poor network performance can be due to misconfigured components. This can be difficult to identify because there are no error statistics to indicate a fault. The misconfiguration might be found only after a meticulous scrutiny of all network settings.

Places to check: All interface settings (`ifconfig -a`), route tables (`netstat -rn`), interface flags (`link_speed/link_mode`), name server configurations (`/etc/nsswitch.conf`), DNS resolvers (`/etc/resolv.conf`, `/var/adm/messages`), FMA faults (`fmadm faulty`, `fmdump`), firewall configurations, and configurable network components (switches, routers, gateways).

Agenda

- Network concepts
- Oracle Solaris 11 networking
- Network configuring
- Monitoring network performance
- Tunable parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris 11 Networking

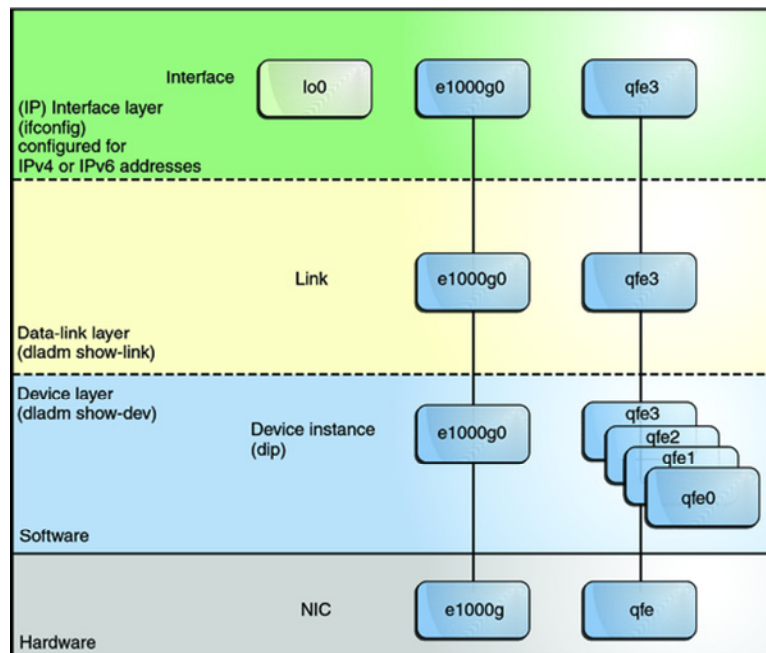
Oracle Solaris 11 introduces an implementation of the network stack.

- The basic relationship between the hardware, datalink, and interface layers remains the same as Solaris 10.
- The software layer is decoupled from the hardware layer.
- This implementation makes network administration more flexible in the following ways:
 - The network configuration is insulated from any changes that might occur in the hardware layer.
 - It allows the use of customized link names in the datalink layer.
 - Multiple networking abstractions such as VLANs, VNICs, physical devices, link aggregations, and IP tunnels are unified into a common administrative entity called datalinks.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris 10 Network Model



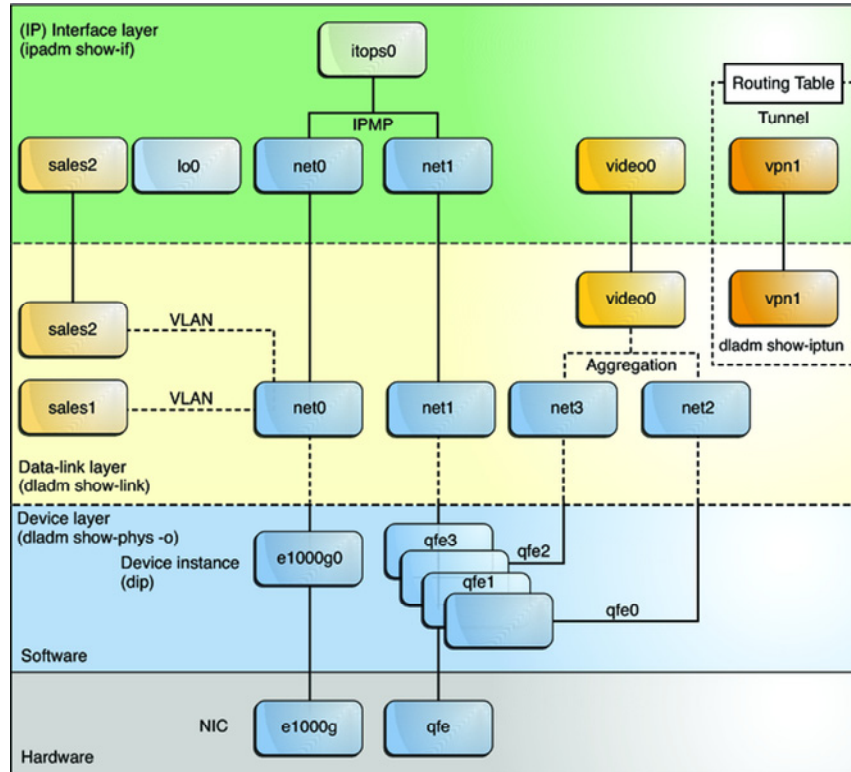
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This figure shows two NICs on the hardware layer: `e1000` with a single device instance `e1000g0`, and `qfe` with multiple device instances, `qfe0` to `qfe3`. The devices `qfe0` through `qfe2` are not used. Devices `e1000g` and `qfe3` are used and have corresponding links `e1000g` and `qfe3` on the datalink layer. In the figure, the IP interfaces are likewise named after their respective underlying hardware, `e1000g` and `qfe3`. These interfaces can be configured with IPv4 or IPv6 addresses to host both types of network traffic. Note also the presence of the loopback interface `lo0` on the interface layer. This interface is used to test, for example, that the IP stack is functioning properly.

Different administrative commands are used at each layer of the stack. For example, hardware devices that are installed on the system are listed by the `dladm show-dev` command. Information about links on the datalink layer is displayed by the `dladm show-link` command. The `ifconfig` command shows the IP interface configuration on the interface layer.

Oracle Solaris 11 Network Model



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This figure also provides a sample of how administratively chosen names can be used in the network setup.

- VLANs are configured on the `net0` link. These VLANs, in turn, are also assigned customized names, such as `sales1` and `sales2`. The VLAN `sales2`'s IP interface is plumbed and operational.
- The device instances `qfe0` and `qfe2` are used to service video traffic. Accordingly, the corresponding links in the datalink layer are assigned the names `subvideo0` and `subvideo1`. These two links are aggregated to host video feed. The link aggregation possesses its own customized name as well, `video0`.
- Two interfaces (`net0` and `net1`) with different underlying hardware (`e1000g` and `qfe`) are grouped together as an IPMP group (`itops0`) to host email traffic.
- Two interfaces have no underlying devices: the tunnel `vpn1`, which is configured for VPN connections and `lo0` for IP loopback operations.

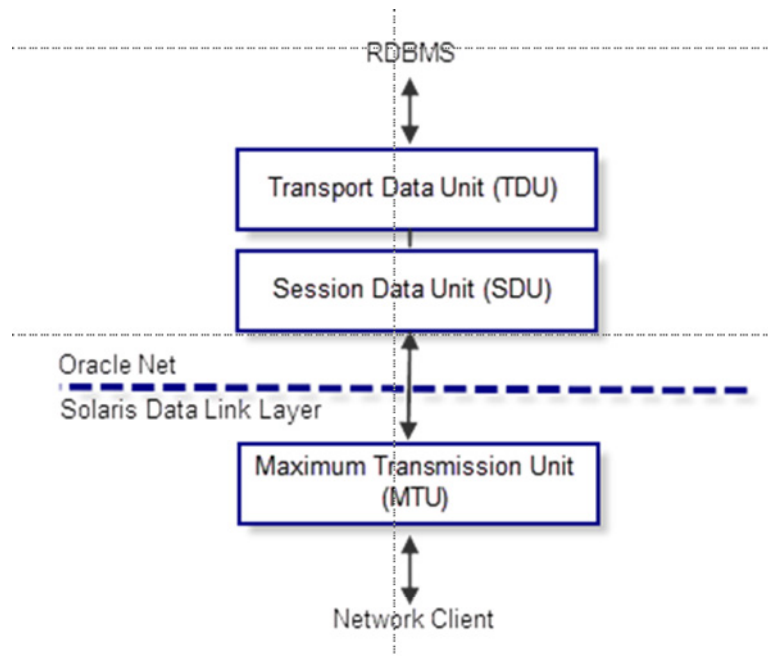
Agenda

- Network concepts
- Oracle Solaris 11 networking
- **Network configuration**
- Monitoring network performance
- Tunable parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Data Packet Encapsulation



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Even if the physical network is performing well, it is possible for clients and servers to not run as efficiently as possible. In this example, Oracle Net encapsulates data into buffers before sending the data across the network. Oracle Net sends each buffer when it is filled, flushed, or when an application tries to read data.

When tuning Oracle data for transmission across the network, you need to consider the following layers of encapsulation:

- **Transport data unit (TDU):** The TDU is the default packet size used within Oracle Net to group data together. This layer will store data up to this value before transferring to the SDU.
- **Session data unit (SDU):** This is the lowest network layer for Oracle Net. Packets sent from Oracle Net to the OS will be no bigger than this size. This layer will store data up to this value before transferring to the MTU.
- **Maximum transmission unit (MTU):** This is the value supported by the Solaris network stack data link layer. Solaris supports a default MTU of 1500 bytes for Ethernet. This layer stores data up to this value before transmitting to the network.

The graphic shows how TDU, SDU, and MTU work together. Assuming that the SDU is set to 3000 and the TDU is set to 15000, Oracle Net will store up to 15000 bytes in a buffer.

The lower network layer (SDU), however, splits this packet up into five packets of 3000 bytes. The 3000-byte packets are sent to the network controller and broken up by the MTU into ten 1500-byte packets.

Note that the Oracle database sets the SDU to 2048 bytes by default. This works well if the data being delivered is less than 2048 bytes. But in cases where the data to be delivered is greater than the SDU, the server breaks this into multiple packets. The Oracle server process incurs a wait event that causes it to context switch and wait until more data is available.

Reconfiguring the MTU

```
# dladm show-phys
LINK      MEDIA      STATE      SPEED  DUPLEX    DEVICE
net0      Ethernet    unknown    0       unknown   e1000g0
# dladm show-link net0
LINK      CLASS      MTU       STATE    OVER
net0      phys        1500      unknown  --
# vi /kernel/drv/e1000g.conf
...
MaxFrameSize=3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
#### 0 is for normal ethernet frames.
#### 1 is for upto 4k size frames.
#### 2 is for upto 8k size frames.
#### 3 is for upto 16k size frames.
...
# dladm set-linkprop -p mtu=9000 net0
# dladm show-link net0
LINK      CLASS      MTU       STATE    OVER
net0      phys        9000      unknown  --
```

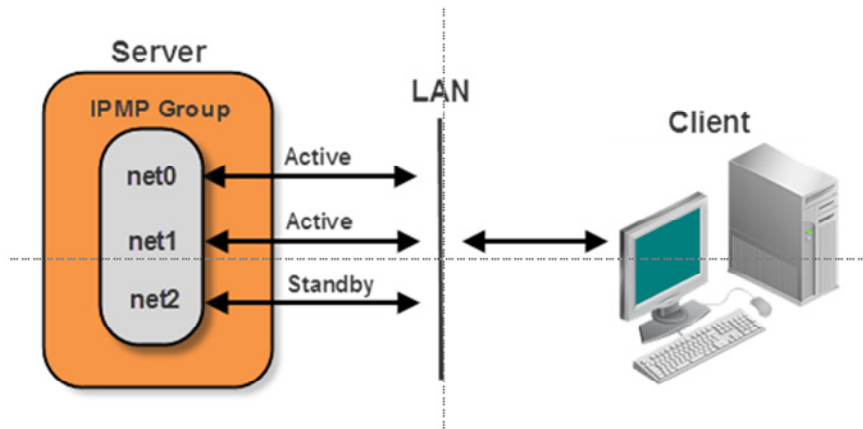
ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Jumbo frame support in Solaris Gigabit Ethernet Network Interface Cards allows the sending and receiving of jumbo frame size packets, which are up to six times the size of standard Ethernet packets. Jumbo frame delivers up to 9216-byte packets instead of a 1522-byte packet for standard Ethernet, which consists of a 1500-byte payload, plus 14 bytes for header, plus VLAN tag 4 bytes, plus CRC 4 bytes.

Enabling jumbo frames in Solaris 11 is dependent on the network being used. For example, the e1000g has a MaxFrameSize setting in /kernel/drv/e1000g.conf, which limits MTU size. In the example shown here, the jumbo frame support of up to 16 kilobytes is enabled on interface e1000g0 (first e1000g instance).

IP Multipathing (IPMP)



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In production environments, it is important to eliminate any single point of failure. IP multipathing (IPMP) provides a mechanism for building redundant network interfaces to guard against failures with network interfaces, cables, switches, or other networking hardware. In addition to eliminating any single point of failure, the IPMP load-spreading feature increases the machine's bandwidth by spreading the outbound load among all the cards in the same IPMP group.

With IPMP, you can assign two or more NICs to a failover group. Each interface is assigned a static test IP address, which is used by Solaris to verify the operational state of the interface. These IP links are used to periodically send an Internet Control Message Protocol (ICMP) echo request to a target system and listen for the response. If no response occurs within a given number of tries, the link is marked as failed. IPMP will fail over all application IP addresses currently configured on that physical interface to another physical interface within the IPMP group. In this way, network outages due to failed network hardware are eliminated.

IPMP Configurations

- Two or more physical interfaces are assigned to an IPMP group.
- IPMP group configurations:
 - Active-active
 - Active-standby

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

An IPMP configuration typically consists of two or more physical interfaces on the same system that are attached to the same LAN. These interfaces can belong to an IPMP group in either of the following configurations:

- **Active-active configuration:** In this configuration, all underlying interfaces are active. An active interface is an IP interface that is currently available for use by the IPMP group. By default, an underlying interface becomes active when you configure the interface to become part of an IPMP group.
- **Active-standby configuration:** In this configuration, at least one interface is administratively configured as a reserve. The reserve interface is called the standby interface. Although idle, the standby IP interface is monitored by the multipathing daemon to track the interface's availability, depending on how the interface is configured. If link-failure notification is supported by the interface, link-based failure detection is used. If the interface is configured with a test address, probe-based failure detection is also used. If an active interface fails, the standby interface is automatically deployed as needed. You can configure as many standby interfaces as you want for an IPMP group.

Configuring IPMP: Active-Active

```
# dladm rename-link net0 link0_ipmp0
# dladm rename-link net1 link1_ipmp0
# ipadm create-ip link0_ipmp0
# ipadm create-ip link1_ipmp0
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i link0_ipmp0 \
  -i link1_ipmp0 ipmp0
# ipadm create-addr -T static \
  -a 192.168.0.112/24 ipmp0/v4add1
# ipadm create-addr -T static \
  -a 192.168.0.142/24 link0_ipmp0/test
# ipadm create-addr -T static \
  -a 192.168.0.143/24 link1_ipmp0/test
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the steps to configure an active-active IPMP configuration with flexible data link. Here, you rename the data links `net0` and `net1` to `link0_ipmp0` and `link1_ipmp0`, respectively. Before these data links can be used by IPMP, you must create an IP interface for each one.

Now you are ready to create the IPMP group. This involves two steps. You first create the IPMP group (`ipmp0` in this example) and then add the underlying interfaces (`link0_ipmp0` and `link1_ipmp0`) to the group. Note that this example shows vanity naming of the network interfaces. You use vanity naming to label network components. This helps you clarify complex network topologies.

Next, assign the data IP addresses to the IPMP interface (`ipmp0`) in the form of IP address objects (`ipmp0/v4add1` and `ipmp0/v4add2`).

Finally, assign the test IP addresses to each underlying interface in the form of IP address objects (`link0_ipmp0/test` and `link1_ipmp0/test`).

Configuring IPMP: Active-Standby

```
# dladm rename-link net0 link0_ipmp0
# dladm rename-link net1 link1_ipmp0
# dladm rename-link net2 link2_ipmp0
# ipadm create-ip link0_ipmp0
# ipadm create-ip link1_ipmp0
# ipadm create-ip link2_ipmp0
# ipadm create-ipmp ipmp0
# ipadm add-ipmp -i link0_ipmp0 -i link1_ipmp0 \
  -i link2_ipmp0 ipmp0
# ipadm set-ifprop -p standby=on -m ip link2_ipmp0
# ipadm create-addr -T static -a 192.168.0.112/24 \
  ipmp0/v4add1
# ipadm create-addr -T static -a 192.168.0.142/24 \
  link0_ipmp0/test
# ipadm create-addr -T static -a 192.168.0.143/24 \
  link1_ipmp0/test
# ipadm create-addr -T static -a 192.168.0.144/24 \
  link2_ipmp0/test
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the steps to configure an active-standby IPMP configuration with flexible data link names.

Here, you rename the data links `net0`, `net1`, and `net2` to `link0_ipmp0`, `link1_ipmp0`, and `link2_ipmp0`, respectively. You then create an IP interface for each one.

Now you create the IPMP group. This involves two steps. You first create the IPMP group (`ipmp0` in this example) and then add the underlying interfaces (`link0_ipmp0`, `link1_ipmp0`, and `link2_ipmp0`) to the group.

Once the IPMP group is created, you set the `standby` property in one of the underlying interfaces (`link2_ipmp0` in this example) to `on`.

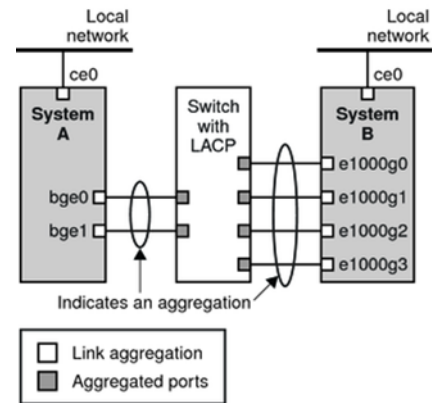
Next, assign the data IP addresses to the IPMP interface (`ipmp0`) in the form of IP address object (`ipmp0/v4add1`).

Finally, assign the test IP addresses to each underlying interface in the form of IP address objects (`link0_ipmp0/test`, `link1_ipmp0/test`, and `link2_ipmp0`).

Link Aggregation

Link aggregation features:

- Increased bandwidth
- Automatic failover/failback
- Load balancing
- Support for redundancy
- Improved administration
- One network address



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris supports the organization of network interfaces IEEE 802.3 link aggregations. A link aggregation consists of several interfaces on a system that are configured together as a single, logical unit. This link aggregation group is then treated as though it were, in fact, a single link.

The following are the features of link aggregations:

- **Increased bandwidth:** The capacity of multiple links is combined into one logical link.
- **Automatic failover/failback:** Traffic from a failed link is failed over to working links in the aggregation.
- **Load balancing:** Both inbound and outbound traffic is distributed according to user-selected load-balancing policies, such as source and destination MAC or IP addresses.
- **Support for redundancy:** Two systems can be configured with parallel aggregations.
- **Improved administration:** All interfaces are administered as a single unit.
- **One network address:** The entire aggregation has less stress on the available IP address pool.

Configuring Link Aggregation

```
# dladm show-link
LINK      CLASS      MTU      STATE      OVER
net0      phys      1500     unknown    --
net1      phys      1500     unknown    --
net2      phys      1500     unknown    --
net3      phys      1500     unknown    --
# dladm create-aggr -l net0 -l net1 -l net2 -l net3 speedway0
root@s11-serv1:~# dladm show-link speedway0
LINK      CLASS      MTU      STATE      OVER
speedway0 aggr      1500     up          net0 net1 net2 net3
# dladm show-aggr
LINK      POLICY  ADDRPOLICY  LACPACTIVITY  LACPTIMER  FLAGS
speedway0 L4       auto        off           short      -----
# ipadm create-ip speedway0
# ipadm create-addr -T static -a 192.168.0.112/24 speedway0/v4
# ipadm show-addr speedway0/v4
ADDROBJ      TYPE      STATE      ADDR
speedway0/v4 static    ok         192.168.0.112/24
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the steps to configure a link aggregation.

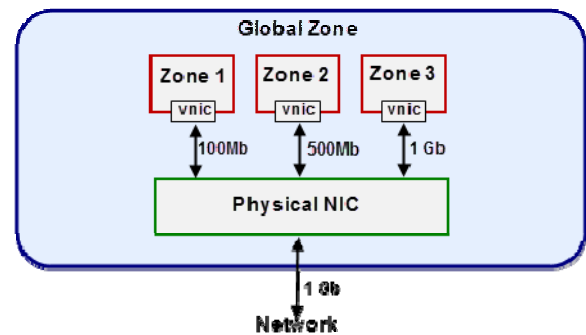
Data links used in a link aggregation cannot be previously configured for other purposes. Here, the `dladm show-link` command shows that none of the data links are currently configured.

Next, the `dladm create-aggr` command is used to assign the available data links to a link aggregation group. This example shows assigning `net0`, `net1`, `net2`, and `net3` links to the `speedway0` link aggregation group.

Next, the `ipadm create-ip` command is used to create a network interface for `speedway0`. After a network interface is created, the `ipadm create-addr` can be used to configure a static IP address.

Network Virtualization

- It is the process of consolidating network resources into a single virtual administrative unit.
- A virtual network consists of one system using zones that are configured over at least one virtual network interface.
- Use network virtualization to:
 - Provide a secure networking environment for your zone workloads
 - Consolidate network hardware and software
 - Control bandwidth limits on selected zones



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

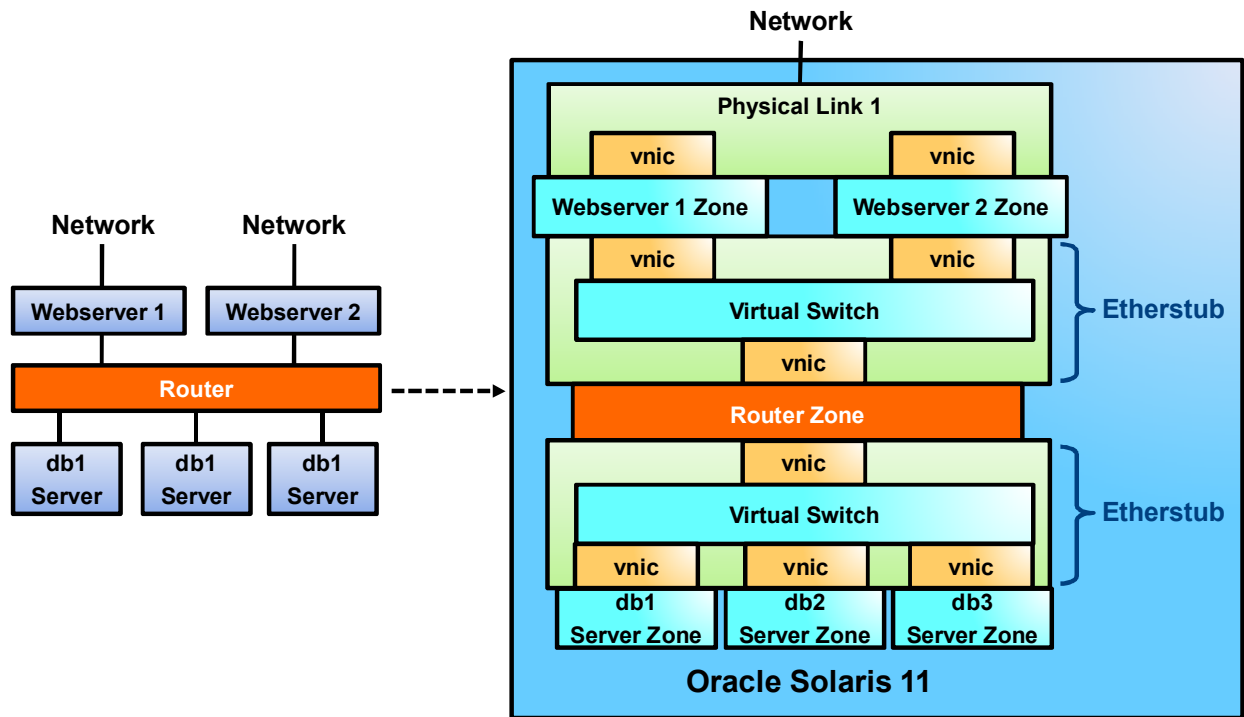
Virtual Networking

Network virtualization is the process of consolidating hardware network resources and software network resources into a single virtual administrative unit. The end product of network virtualization is the *virtual network*. The goal of network virtualization is to provide systems and users with efficient, controlled, and secure sharing of the networking resources.

A virtual network consists of one system using zones that are configured over at least one virtual network interface. The zones can communicate with each other as though on the same local network, providing a virtual network on a single host. The building blocks of the virtual network are virtual network interface cards (virtual NICs or VNICs) and virtual switches.

Virtual networking provides a secure networking environment for your zone workloads. You can use virtual networking to consolidate network hardware and software. You can also use virtual networking to control bandwidth limits on selected zones.

Consolidating to Virtual Networking



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Network virtualization is the process of combining hardware network resources and software network resources into a single administrative unit. The goal of network virtualization is to provide systems and users with efficient, controlled, and secure sharing of the networking resources. The end product of network virtualization is the virtual network.

You can use network virtualization to consolidate your existing network hardware and software. An important part of planning server consolidation is determining how the servers are connected to the network and what the network security requirements for each server are. You can then use virtual networking components to design the appropriate virtual network infrastructure that meets the requirements.

In this illustration, you can see a server network topology before consolidation on the left. On the front side of the topology, two servers (webserver 1 and webserver 2) are connected directly to the network. On the back side of the topology, three database servers (db1, db2, and db3) are installed. Between the web servers and database servers is a router, which provides firewall services for protecting the databases.

The right side of the illustration shows a model of the servers and networks after consolidation that meets the original connection and security requirements.

Bandwidth Management

- Enables assignment of a portion of the available bandwidth of a NIC
- The allocated portion of bandwidth is known as a share.
 - The limit is the maximum allocation of bandwidth that the share can consume.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

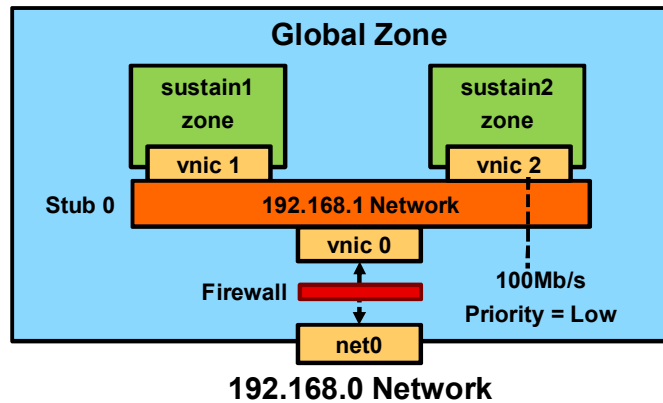
Bandwidth management enables you to assign a portion of the available bandwidth of an NIC to a consumer, such as an application or customer. You can control bandwidth on a per-application, per-port, per-protocol, and per-address basis. Bandwidth management ensures efficient use of the large amount of bandwidth available from the new GLDv3 network interfaces. Resource control features enable you to implement a series of controls on an interface's available bandwidth.

The allocated portion of bandwidth is known as a share. By setting up shares, you can allocate enough bandwidth for applications that cannot function properly without a certain amount of bandwidth. For example, streaming media and Voice-over IP consume a great deal of bandwidth. You can use the resource-control features to guarantee that these two applications have enough bandwidth to run successfully. You can also set a limit on the share. The limit is the maximum allocation of bandwidth that the share can consume. By using limits, you can prevent noncritical services from taking away bandwidth from critical services.

You can prioritize among the various shares allotted to consumers. You can give the highest priority to critical traffic, such as heartbeat packets for a cluster, and lower priority for less critical applications.

You can control bandwidth usage through the management of flows (by using the `flowadm` command) and link utilization (by using the `dladm` command).

Managing Bandwidth



```
# flowadm add-flow -l vnic2 -a transport=tcp,local_port=80 http1
# flowadm set-flowprop -p maxbw=100M http1
# flowadm show-flowprop http1
```

FLOW	PROPERTY	VALUE	DEFAULT	POSSIBLE
http1	maxbw	100	--	--

```
# dladm set-linkprop -p priority=low vnic2
# dladm show-linkprop -p priority vnic2
```

LINK	PROPERTY	PERM	VALUE	DEFAULT	POSSIBLE
vnic2	priority	rw	low	high	low, medium, high

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows you how to restrict flows and lower priority on a VNIC. Flows consist of network packets that are organized according to an attribute. Flows enable you to further allocate network resources.

In this example, a flow named `http1` is created by using the `flowadm` command. This user-designed flow (`http1`) restricts `vnic2` bandwidth to 100 Mb/s and sets the link priority to low.

Integrated Load Balancer (ILB)

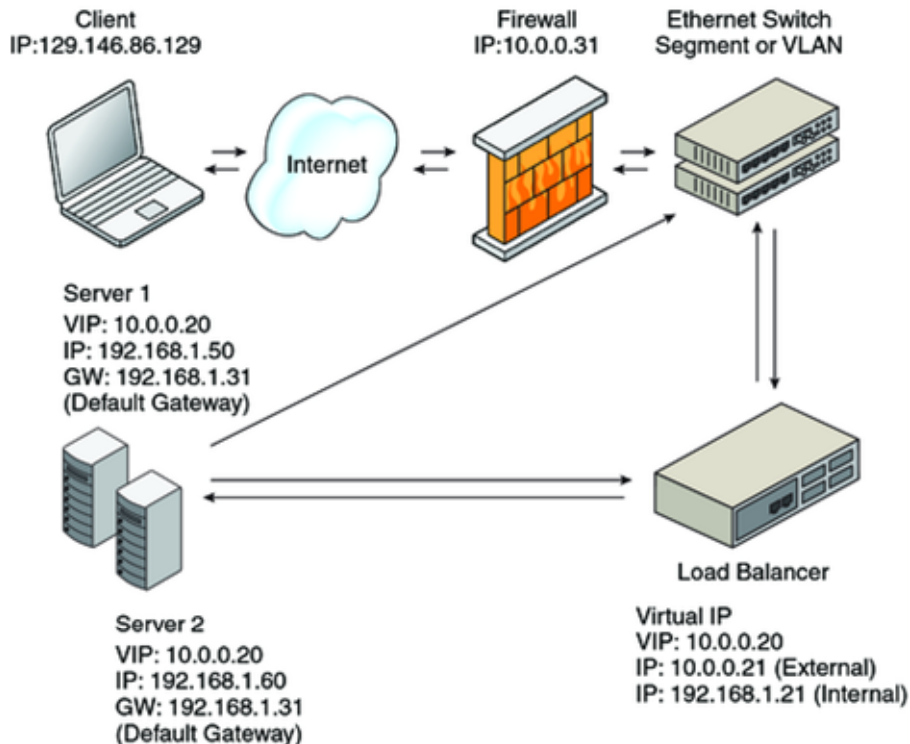
- ILB provides load-balancing capabilities for Layer 3 and Layer 4.
- ILB:
 - Supports the stateless Direct Server Return (DSR) and Network Address Translation (NAT) modes of operation for both IPv4 and IPv6
 - Enables ILB administration through a command-line interface (CLI)
 - Provides server monitoring capabilities through health checks

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ILB provides Layer 3 and Layer 4 load-balancing capabilities. It operates at the network (IP) and transport (TCP/UDP) layers—for the Oracle Solaris operating system installed on SPARC-based and x86-based systems. ILB intercepts incoming requests from clients, decides which back-end server should handle the requests based on load-balancing rules, and then forwards the requests to the selected server. ILB performs optional health checks and provides the data for the load-balancing algorithms to verify whether the selected server can handle the incoming requests.

ILB Example: DSR



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Direct Server Return mode (DSR) refers to load-balancing incoming requests to the back-end servers and letting the return traffic from the servers bypass the load balancer by sending them directly to the client. ILB's current implementation of DSR does not provide TCP connection tracking (meaning that it is stateless).

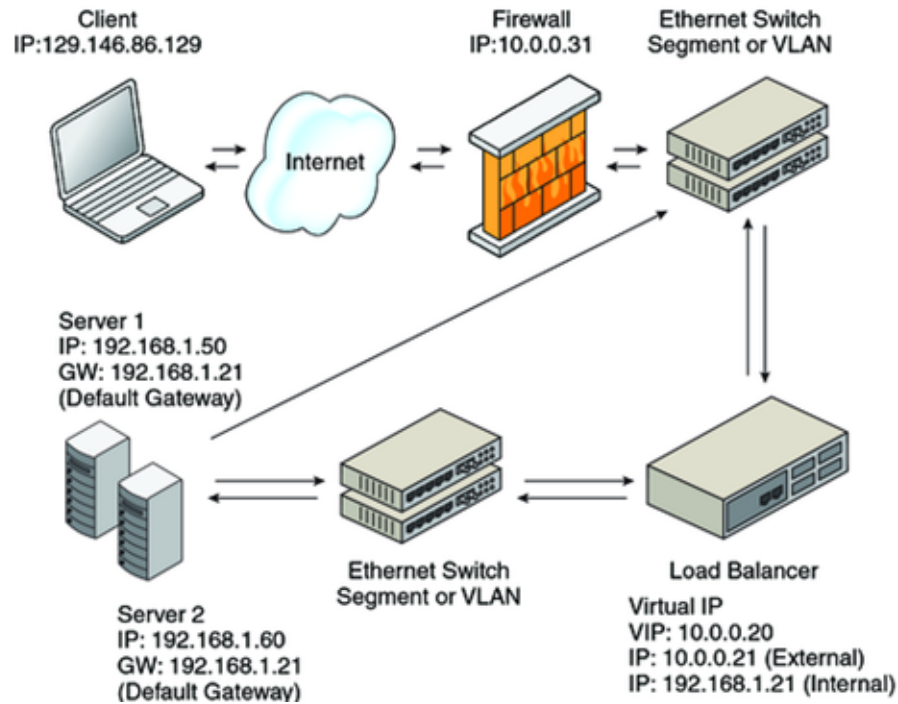
Advantages:

- Better performance than NAT because only the destination MAC address of packets is changed and servers respond directly to clients.
- Full transparency: The servers see a connection directly from the client IP address and reply to the client through the default gateway.

Disadvantages:

- The back-end server must respond to both its own IP address (for health checks) and the virtual IP address (for load balanced traffic).
- Because the load balancer maintains no connection state (meaning that it is stateless), adding or removing servers will cause connection disruption.

ILB Example: NAT



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

NAT-based load-balancing involves rewriting the IP header information, and handles both the request and the response traffic. There are two types of NAT: half-NAT and full-NAT. Both types rewrite the destination IP address. However, full-NAT also rewrites the source IP address, making it appear to the server that all connections are originating from the load balancer. NAT does provide TCP connection tracking (meaning that it is stateful).

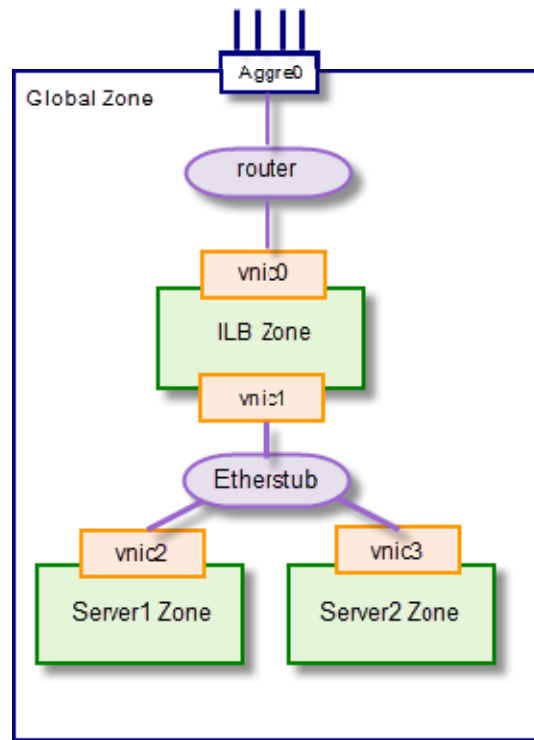
Advantages:

- Works with all back-end servers by changing the default gateway to point to the load balancer.
- Because the load balancer maintains the connection state, adding or removing servers without connection disruption is possible.

Disadvantages:

- Slower performance than DSR because processing involves manipulation of the IP header and servers send responses to the load balancer.
- All the back-end servers must use the load balancer as a default gateway.

ILB Example Using Zones



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This illustration shows combining ILB with Oracle Solaris Zones and virtual networking in a NAT topology.

ILB Components and Terms

- The `ilbadm` utility
- The `libilb` configuration library
- The `ilbd` daemon
- Direct server return (DSR)
- NAT-based load-balancing
- The VIP
- Load-balancing algorithm
- Load-balancing rule

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

- `Ilbadm` utility: You can use this interface to configure load-balancing rules, perform optional health checks, and view statistics.
- `libilb` configuration library: `ilbadm` and other third-party applications can use the functionality implemented in `libilb` for ILB administration.
- `ilbd` daemon: This daemon performs the following tasks:
 - Manages persistent configuration
 - Provides serial access to the ILB kernel module by processing the configuration information and sending it to the ILB kernel module for execution
 - Performs health checks and provides the results to the ILB kernel module so that the load distribution is properly adjusted
- Direct server return (DSR) refers to load-balancing incoming requests to the back-end servers and letting the return traffic from the servers bypass the load balancer by sending them directly to the client. ILB's current implementation of DSR does not provide TCP connection tracking (meaning that it is stateless).

- NAT-based load-balancing involves rewriting the IP header information, and handles both the request and the response traffic. There are two types of NAT: half-NAT and full-NAT. Both types rewrite the destination IP address. However, full-NAT also rewrites the source IP address, making it appear to the server that all connections are originating from the load balancer. NAT does provide TCP connection tracking (meaning that it is stateful).
- The VIP (virtual IP address): The IP address for the ILB virtual service.
- Load-balancing algorithm: The algorithm that ILB uses to select a back-end server from a server group for an incoming request.
- Load-balancing rule: In ILB, a virtual service is represented by a load-balancing rule.

Load-Balancing Rule

- You use `ilbadm` to create, delete, and list the load-balancing rules.
- Create and enable the rule.

```
# ilbadm create-servergroup -s server=60.0.0.109,60.0.0.11 apache1
# ilbadm create-rule -e -i vip=10.0.0.10,port=5000-5009,protocol=tcp \
-m lbalg=rr,type=NAT,proxy-src=60.0.0.101-60.0.0.104 \
-h hc-name=hc1 -o servergroup=apache1 apache1
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In ILB, a virtual service is represented by a load-balancing rule and is defined by the following parameters:

- Virtual IP address
- Transport protocol: TCP or UDP
- Port number (or a port range)
- Load-balancing algorithm
- Type of load-balancing mode (DSR, NAT (full-NAT), or h (half-NAT))
- Server group consisting of a set of back-end servers
- Optional server health checks that can be executed for each server in the server group
- Optional port to use for health checks
- Rule name to represent a virtual service

ILB Algorithms

- Round-robin
- *Source IP* hash
- *Source IP, port* hash
- *Source IP, VIP* hash

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

ILB algorithms control traffic distributions and provide various characteristics for load distribution and server selection. ILB provides the following algorithms for the two modes of operation:

- **Round-robin:** In a round-robin algorithm, the load balancer assigns the requests to a list of the servers on a rotating basis. Once a server is assigned a request, the server is moved to the end of the list.
- **Source IP hash:** In source IP hash method, the load balancer selects a server based on the hash value of the source IP address of the incoming request.
- **Source IP, port hash:** In source IP, port hash method, the load balancer selects a server based on the hash value of the source IP address, and the source port of the incoming request.
- **Source IP, VIP hash:** In source IP, VIP hash method, the load balancer selects a server based on the hash value of the source IP address, and the destination IP address of the incoming request.

ilbadm Utility

ilbadm Command	Description
ilbadm create-rule	Creates a rule name with the given characteristics
ilbadm show-rule	Displays characteristics of specified rules or displays all the rules if no rules are specified
ilbadm show-statistics	Shows ILB statistics
ilbadm create-servergroup	Creates a server group
ilbadm add-server	Adds the specified servers to server groups
ilbadm show-servergroup	Lists a server group or lists all the server groups if no server group is specified
ilbadm create-healthcheck	Sets up health check information that can be used to set up rules
ilbadm show-hc-result	Shows the health check results

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Here is a list of commonly used `ilbadm` commands. For a complete list of `ilbadm` commands refer to the `ilbadm man` page.

Agenda

- Network concepts
- Oracle Solaris 11 networking
- Network configuration
- **Monitoring network performance**
- Tunable parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Monitoring Network Performance

- Check the response time of network hosts.
- Test the reliability of packet sizes.
- Capture data packets and trace the calls from each client to each server.
- Display the network status.
- Display server and client statistics.
- Display the route that packets take to reach the network host.
- Display the number of hops between two nodes on a segmented network.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle Solaris 11 OS provides commands that you use to complete the actions listed in the slide.

Testing Response Time

Physical problems can be caused by any one of the following:

- Loose cables or connectors
- Improper grounding
- Missing termination
- Signal reflection
- Lack of compliance with network standards, such as cables that exceed their specified reach

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You use the `ping` command to check the response time of a host on the network. A response time that is more than the expected response time indicates a physical problem in the network.

The most basic form of the `ping` command displays the state of the client computer. For example, the following `ping` command sends a single packet to the client and prints a message displaying the status of the client:

```
# ping wildcat
wildcat is alive
```

When you use the `ping` command with the `-s` option, one data packet is sent to the specified host per second. The command then prints each response message and the time taken for the round trip. The following is a sample output of the `ping -s` option.

Network Status

```
# netstat -r
```

Routing Table: IPv4					
Destination	Gateway	Flags	Ref	Use	Interface
default	ppp26143.cwiz.com	UG	1	2724	
218.200.163.0	nazteca	U	1	1811	hme0
BASE-ADDRESS.MCAST.NET	nazteca	U	1	0	hme0
localhost	localhost	UH	1	123	lo0

Routing Table: IPv6					
Destination/Mask	Gateway	Flags	Ref	Use	
If					

fe80::/10	fe80::a00:20ff:fee1:3bee	U	1	0	hme0
ff00::/8	fe80::a00:20ff:fee1:3bee	U	1	0	hme0
localhost	localhost	UH	1	21	lo0

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

To view the state of the routing tables on the system, use the `netstat -r` command. A sample output of the `netstat -r` command is shown in the above slide.

The **Flags** field in the preceding output shows the status of the network interface. The **Flags** field can have the following values:

- **U** - Indicates that the route is working
- **G** - Indicates that the route is a gateway
- **H** - Indicates that the route is to a host system
- **D** - Indicates that the route is dynamically created

kstat Command (Network)

```
# kstat -m tcp
module: tcp                instance: 0
name:  tcp                 class:   mib2
      activeOpens          30
      attemptFails         16
      connTableSize        72
      connTableSize6       96
      crtime                255.182981365

# kstat -m ip
module: ip                 instance: 0
name:  icmp                class:   mib2
      crtime                255.180818107
      inAddrMaskReps        0

# kstat -m hme
module: hme                instance: 0
name:  hme0                class:   net
      align_errors          0
      allocbfail            0
      asic_rev              193
      babble                 0
      brdcstrcv             1299372
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Solaris Kernel Statistics framework tracks network usage, and the `kstat` command fetches these details. This command has a variety of options for selecting statistics.

The `-m` option for `kstat` matches on a module name. In the above example, you use it to display available statistics for the networking modules. These commands fetch statistics for `ip`, `tcp`, and `hme` (the Ethernet card in the example system).

ipadm Utility

ipadm show-if

IFNAME	CLASS	STATE	ACTIVE	OVER
lo0	loopback	ok	yes	--
net0	ip	ok	yes	--

ipadm show-addr

ADDROBJ	TYPE	STATE	ADDR
lo0/v4	static	ok	127.0.0.1/8
net0/v4	static	ok	192.168.0.112/24
lo0/v6	static	ok	::1/128
net0/v6	addrconf	ok	fe80::a00:27ff:febb:669c/10

ipadm show-addrprop training1/v4

ADDROBJ	PROPERTY	PERM	CURRENT	PERSISTENT	DEFAULT	POSSIBLE
training1/v4	broadcast	r-	192.168.0.255	--	192.168.0.255	--
training1/v4	deprecated	rw	off	--	off	on,off
training1/v4	prefixlen	rw	24	24	24	1-30,32
training1/v4	private	rw	off	--	off	on,off
training1/v4	reqhost	r-	--	--	--	--
training1/v4	transmit	rw	on	--	on	on,off
training1/v4	zone	rw	global	--	global	--

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

dladm Utility

```
# dladm show-phys
```

LINK	MEDIA	STATE	SPEED	DUPLEX	DEVICE
net0	Ethernet	up	1000	full	e1000g0
net1	Ethernet	unknown	1000	full	e1000g1
net2	Ethernet	unknown	1000	full	e1000g2
net3	Ethernet	unknown	0	unknown	e1000g3

```
# dladm show-phys -L
```

LINK	DEVICE	LOCATION
net0	e1000g0	MB
net1	e1000g1	MB
net2	e1000g2	MB
net3	e1000g3	MB

```
# dladm show-link
```

LINK	CLASS	MTU	STATE	OVER
net0	phys	1500	up	--
net1	phys	1500	unknown	--
net2	phys	1500	unknown	--
net3	phys	1500	unknown	--
zone1/net0	vnuc	1500	up	net0
zone2/net0	vnuc	1500	up	net0

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Introducing the snoop Command

Tip	Use
Capture the output in a binary file.	# snoop -o <i>filename</i>
Capture data packets between two nodes.	# snoop -o <i>filename</i> <i>host1</i> <i>host2</i>
Capture only relevant information, such as the packet header in the first 120 bytes of data.	# snoop -s 120
Capture a specific number of data packets.	# snoop -c <i>number_of_packets</i>
Use filters to capture specific types of network activity.	# snoop broadcast

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You use the `snoop` command to capture data packets and trace the calls that each client makes to a server. The `snoop` command displays the data packets as soon as they are received. However, this might occupy all the space on the disk and impact the performance of the network.

To avoid this situation, you can save the information in a binary file and use it later to evaluate network performance. The table shows various tips that you can use to prevent the output of the `snoop` command from occupying all the disk space.

Note: To display the contents of a binary file, use the `snoop -i filename` command.

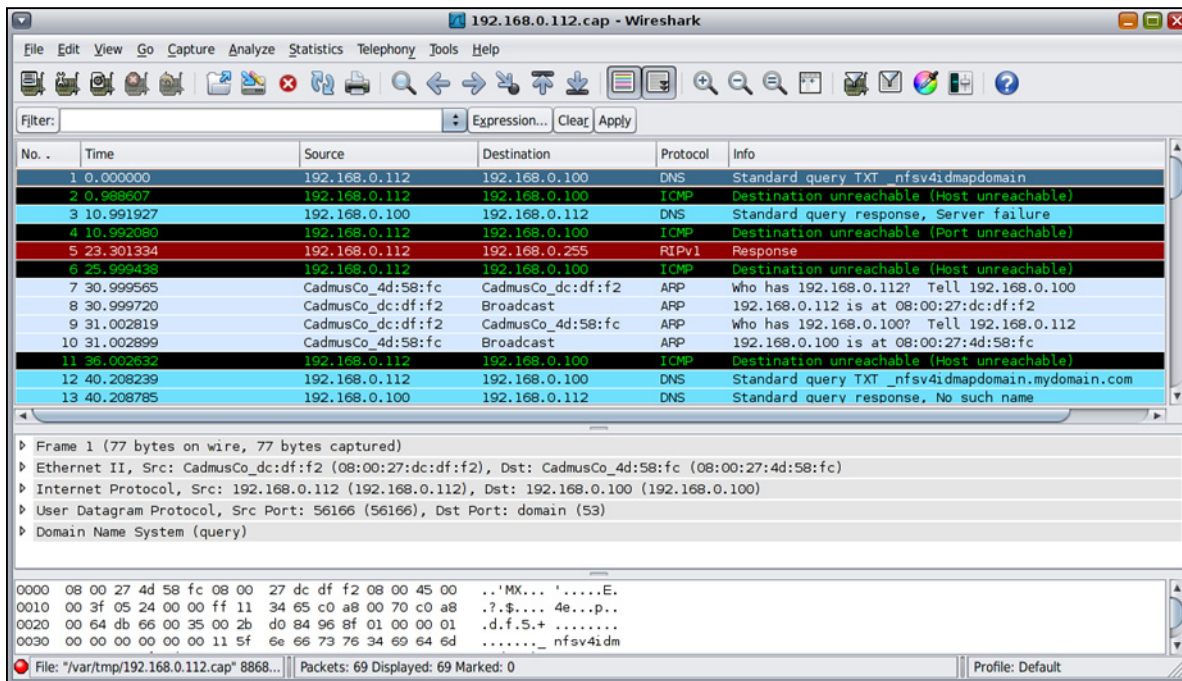
snoop Command

```
# snoop
Using device hme0 (promiscuous mode)
218.200.163.20 -> 218.200.163.213 TCP D=22 S=52666 Ack=4249755803
Seq=2080552721 Len=0 Win=16183
218.200.163.213 -> 218.200.163.20 TCP D=52666 S=22 Push Ack=2080552721
Seq=4249755803 Len=148 Win=49640
218.200.163.20 -> 218.200.163.213 TCP D=22 S=52666 Ack=4249755951
Seq=2080552721 Len=0 Win=16146
218.200.163.213 -> 218.200.163.20 TCP D=52666 S=22 Push Ack=2080552721
Seq=4249755951 Len=148 Win=49640
218.200.163.213 -> 218.200.163.20 TCP D=52666 S=22 Push Ack=2080552721
Seq=4249756099 Len=148 Win=49640
218.200.163.20 -> 218.200.163.213 TCP D=22 S=52666 Ack=4249756247
Seq=2080552721 Len=0 Win=16072
218.200.163.213 -> 218.200.163.20 TCP D=52666 S=22 Push Ack=2080552721
Seq=4249756247 Len=148 Win=49640
218.200.163.213 -> 218.200.163.20 TCP D=52666 S=22 Push Ack=2080552721
Seq=4249756395 Len=148 Win=49640
218.200.163.213 -> 218.200.163.20 TCP D=52666 S=22 Push Ack=2080552721
Seq=4249756543 Len=148 Win=49640
218.200.163.20 -> 218.200.163.213 TCP D=22 S=52666 Ack=4249756691
Seq=2080552721 Len=0 Win=16425
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

wireshark Utility



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Wireshark is a network protocol analyzer. You can use it to capture and interactively browse the traffic running on a computer network. Because of its rich and powerful feature set, system administrators, security experts, developers, and educators around the world use it regularly. It is freely available as open source and is released under the GNU General Public License version 2.

With Wireshark, you can:

- Capture live packet data from a network interface
- Display packets with very detailed protocol information
- Open and save captured packet data
- Import and export packet data from and to many other capture programs
- Filter packets by using many criteria
- Search for packets by using many criteria
- Colorize packet display based on filters
- View various statistics

This slide shows the Wireshark packet analyzer interface.

flowstat Utility

- Enables you to gather runtime statistics on user-defined flows
- Using `flowstat`, you can:
 - Display receive-side statistics only (includes bytes)
 - Display transmit-side statistics only
 - Specify an interval in seconds at which statistics are refreshed. The default interval is one second.
 - Display statistics for all flows on the specified link or statistics for the specified flow

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Flows consist of network packets that are organized according to an attribute. Flows enable you to further allocate network resources. Packets traverse a path when they flow into or out of a system. On a granular level, packets are received and transmitted through receive (Rx) rings and transmit (Tx) rings of an NIC. From these rings, received packets are passed up the network stack for further processing while outbound packets are sent to the network.

The `flowstat` command allows you to gather reports on runtime statistics about user-defined flows.

Using `flowstat`, you can:

- Display receive-side statistics only (includes bytes)
- Display transmit-side statistics only
- Specify an interval in seconds at which statistics are refreshed. The default interval is one second.
- Display statistics for all flows on the specified link or statistics for the specified flow

flowstat: Examples

```
# flowstat -i 1
FLOW    IPKTS    RBYTES    IDROPS    OPKTS    OBYTES    ODROPS
http1   430.45K   910.46M      0    398.22K   44.09M      0

# flowstat -r
FLOW    IPKTS    RBYTES    IDROPS
http1   2.95M    3.44M      0

# flowstat -t
FLOW    OPKTS    OBYTES    ODROPS
http1   17.89M   987.22M      0
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The first example shows information every second about incoming and outgoing traffic on all configured flows on the system.

The second example shows receive-side statistics for all flows.

The third example shows transmit-side statistics for all flows.

traceroute Utility

```
# traceroute www.sun.com
traceroute to www.sun.com (72.5.124.61), 30 hops max, 40 byte packets
1  ppp26143.cwiz.com (218.200.163.1)  1.009 ms  0.748 ms  0.763 ms
2  502.ATM1-1.GW1.DEN4.ALTER.NET (157.130.123.49)  9.576 ms  9.578 ms  9.460 ms
3  144.at-5-0-0.XR1.ATL5.ALTER.NET (152.63.80.146)  9.872 ms  38.178 ms  9.866
ms
4  0.so-1-1-2.XT1.ATL5.ALTER.NET (152.63.85.189)  9.860 ms  24.144 ms  10.005
ms
5  POS6-0.BR2.ATL5.ALTER.NET (152.63.82.197)  9.832 ms  9.799 ms  9.689 ms
6  204.255.168.106 (204.255.168.106)  10.671 ms  10.589 ms  11.093 ms
7  cr1-bundle-pos-1.sanfrancisco.savvis.net (204.70.197.30)  75.405 ms  76.516
ms  75.412 ms
8  er1-7-0-0.SanJoseEquinix.savvis.net (204.70.200.197)  75.238 ms  75.344 ms
75.308 ms
9  208.175.172.10 (208.175.172.10)  77.829 ms  77.906 ms  78.936 ms
10 border2.te7-1-bbnet1.sfo002.pnap.net (63.251.63.17)  77.708 ms  99.068 ms
79.137 ms
11 * * *
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You use the `traceroute` utility to determine the route that a packet takes to reach the destination Internet host. The `traceroute` command is especially useful for determining problems in routing configuration and routine path failures. If a particular host is unreachable, you use the `traceroute` command to identify the path that the packet follows to reach the host and the possible failure areas along that path.

The `traceroute` command requires the name of the destination system. The above is a sample output of the `traceroute` command.

Caution: Note that IP traffic routes are not guaranteed to be the same in both directions. In particular, the `traceroute` command does not function properly if the reverse path is blocked. If outgoing packets reach each router but the inbound packets are blocked by a firewall, the `traceroute` command cannot resolve each hop check.

The output of the `traceroute` command also displays the number of times a packet hops before reaching the destination.

Testing the Reliability of Packet Sizes

```
# spray -c 20 -d 10 -l 1026 wildcat
sending 20 packets of length 1026 to wildcat ...
    10 packets (50.000%) dropped by wildcat
    0 packets/sec, 683 bytes/sec
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You use the `spray` command to test the reliability of packet sizes. The `spray` command displays the number and size of packets that the client drops. The following is the syntax of the `spray` command:

```
spray [-c count -d interval -l packet_size ] hostname
```

- `-c count`: Specifies the number of packets to be sent
- `-d interval`: Specifies the pause time in microseconds between sending packets
- `-l packet_size`: Specifies the size of the packet
- `Hostname`: Specifies the name of the client system

In the example above, the `spray` command sends 20 packets, each containing 1026 B of data, to the wildcat server after a time interval of 10 microseconds.

dlstat Utility

- Reports runtime statistics about data links.
- `dlstat` allows you to:
 - Examine all links and reports statistics
 - Examine a specific link and reports statistics
 - Examine physical network devices and reports statistics
 - Examine link aggregations and reports statistics
 - Specify a sampling interval

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `dlstat` command reports runtime statistics about data links. The output is sorted in the descending order of link utilization. The slide lists what you can do by using `dlstat`.

dlstat: Examples

```
# dlstat
  LINK           IPKTS           RBYTES    OPKTS    OBYTES
  vnic0           222           9.42K     1.50K    118.00K
  vnic1           1.10K          82.73K    168      7.15K
  vnic2           1.10K          82.73K    168      7.15K
  speedway0       8.95K          713.56K   17.69K   20.80M

# dlstat show-phys
  LINK           TYPE    INDEX      PKTS      BYTES
  net0           rx      0         5.25K     464.55K
  net1           rx      0         1.32K     93.89K
  net2           rx      0         1.32K     93.89K
  net3           rx      0         1.32K     93.89K
  speedway0      rx      0         5.25K     464.55K
  speedway0      rx      1         1.32K     93.89K
  speedway0      rx      2         1.32K     93.89K
  speedway0      rx      3         1.32K     93.89K
  speedway0      tx      0         4.86K     3.46M
  speedway0      tx      1          885     831.00K
  speedway0      tx      2         1.79K     1.88M
  speedway0      tx      3        10.21K    14.64M
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The slide shows examples of `dlstat` usage.

In the first example, running `dlstat` without subcommands displays a summary of statistics for all the links. The report shows incoming traffic (`IPKTS` and `RBYTES`) and outgoing traffic (`OPKTS` and `OBYTES`).

In the second example, the `show-phys` subcommand reports network traffic statistics for each physical network device. The `INDEX` field identifies the ring queue associated with a device. The report includes statistics for data received (`rx`) and data transmitted (`tx`). Note that if your link aggregations (`speedway0`) are present, they are also displayed.

dlstat: Examples

```
# dlstat show-link
  LINK TYPE   ID      INDEX  PKTS    BYTES
  vnic0      rx      local   --      114      4.84K
  vnic0      rx      bcast   --      112      4.75K
  vnic0      rx      sw       --       0       0
  vnic0      tx      bcast   --     1.01K    79.68K
  vnic0      tx      sw       --      514     40.38K
...
  speedway0  rx      hw        0      5.22K   458.88K
  speedway0  rx      hw        1      1.28K    87.51K
  speedway0  rx      hw        2      1.28K    87.51K
  speedway0  rx      hw        3      1.28K    87.51K
# dlstat show-aggr
  LINK PORT   IPKTS    RBYTES  OPKTS    OBYTES
  speedway0  --  9.26K   751.05K  17.78K   20.82M
  speedway0  net0    5.28K   466.74K   4.89K    3.46M
  speedway0  net1    1.33K   94.77K    885      831.00K
  speedway0  net2    1.33K   94.77K    1.79K    1.88M
  speedway0  net3    1.33K   94.77K   10.22K   14.64M
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `show-link` subcommand reports network traffic statistics for each network link. In the output, the ID field indicates whether hardware rings are exclusively assigned (indicated by `hw`) or shared (indicated by `sw`) among clients. `rx` rings are shared if other clients, such as VNICS, are configured over the link as well. In the example shown in the slide, sharing is indicated by the `vnic0 sw` value in the ID column.

The `show-aggr` subcommand reports incoming and outgoing network traffic statistics for aggregated links. The PORT field indicates the devices that make up the link aggregation.

TCP Statistics from DTrace

```
# dtrace -ln 'mib:ip::tcp*'
ID    PROVIDER    MODULE    FUNCTION NAME
2990      mib        ip        tcp_find_pktinfo tcpInErrs
2991      mib        ip        ip_mib2_add_ip_stats tcpInErrs
3109      mib        ip        ip_rput_data_v6 tcpIfStatsInErrs
3110      mib        ip        ip_tcp_input tcpIfStatsInErrs
3597      mib        ip        tcp_fuse_output tcpInSegs
3601      mib        ip        tcp_ack_timer tcpOutAckDelayed
3602      mib        ip        tcp_fuse_output_urg tcpOutUrg
3603      mib        ip        tcp_xmit_mp tcpOutUrg
3605      mib        ip        tcp_xmit_early_reset tcpOutRsts
3606      mib        ip        tcp_xmit_ctl tcpOutRsts
3608      mib        ip        tcp_xmit_mp tcpOutControl
3609      mib        ip        tcp_xmit_early_reset tcpOutControl
3610      mib        ip        tcp_xmit_ctl tcpOutControl
3611      mib        ip        tcp_multisend tcpOutControl
3612      mib        ip        tcp_fuse_output tcpOutDataBytes
3613      mib        ip        tcp_send tcpOutDataBytes
3614      mib        ip        tcp_multisend_data tcpOutDataBytes
3615      mib        ip        tcp_output tcpOutDataBytes
3616      mib        ip        tcp_fuse_output tcpOutDataSegs
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The command in the slide lists the TCP MIB statistics from DTrace.

IP Statistics from DTrace

```
# dtrace -ln 'mib:ip::ip*'
ID      PROVIDER  MODULE                                FUNCTION NAME
2920     mib      ip icmp_redirect_v6                  ipv6IfIcmpInBadRedirects
2921     mib      ip icmp_inbound_v6                  ipv6IfIcmpInBadRedirects
2922     mib      ip ip_mib2_add_icmp6_stats          ipv6IfIcmpInBadRedirects
2923     mib      ip ndp_input_solicit                ipv6IfIcmpInBadNeighborSolicitations
2924     mib      ip ip_mib2_add_icmp6_stats          ipv6IfIcmpInBadNeighborSolicitations
2925     mib      ip ndp_input_advert                 ipv6IfIcmpInBadNeighborAdvertisements
2926     mib      ip ip_mib2_add_icmp6_stats          ipv6IfIcmpInBadNeighborAdvertisements
2927     mib      ip ip_fanout_proto_v6              ipv6IfIcmpInOverflows
....
3497     mib      ip icmp_inbound_v6                  ipv6IfIcmpInErrors
3498     mib      ip ip_mib2_add_icmp6_stats          ipv6IfIcmpInErrors
3499     mib      ip mldv2_query_in                   ipv6IfIcmpInErrors
3500     mib      ip mld_input                        ipv6IfIcmpInErrors
3501     mib      ip ip_mib2_add_icmp6_stats          ipv6IfIcmpInGroupMembTotal
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

As with TCP, DTrace can trace IP statistics as they are updated. The command in the slide lists the probes that correspond to IP MIB statistics that begins with “ip” (which is not all of them).

ICMP Statistics from DTrace

```
# dtrace -n 'mib:::icmp* { @[probename] = sum(arg0); }'  
dtrace: description 'mib:::icmp* ' matched 34 probes  
^C  
  
icmpOutDestUnreachs      11  
icmpOutMsgs              11
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The DTrace one-liner in the slide tracks ICMP MIB events.

Agenda

- Network concepts
- Oracle Solaris 11 networking
- Network configuration
- Monitoring network performance
- Tunable parameters

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Displaying and Setting Network Tunable Parameters

- The `/etc/system` file

```
set ip:ipcl_conn_hash_size=40000
```

- Use the `ipadm` command to manage TCP/IP tunable parameters.

```
# ipadm set-prop -p forwarding=on ipv4
# ipadm show-prop -p forwarding ipv4
PROTO  PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
ipv4    forwarding rw     on        on           on        on,off
```

- The `ndd` command

```
# ndd /dev/tcp \? | head -25
tcp_time_wait_interval      (read and write)
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Network parameter tuning can be performed by using one of the following methods:

- The `/etc/system` file
- The `ipadm` command
- The `ndd` command
 - The `/etc/system` file: You can configure network tuning parameter in the `/etc/system` file by using the `set` directive as shown in this example. Use this method only when the tuning parameter is not supported by `ipadm` or `ndd` commands.
- The `ipadm` command: You use this command to display and set TCP/IP tunable parameters. `ipadm` supports all network parameters with the following exceptions, which are managed in the `/etc/system` file:
 - `ipcl_conn_hash_size`
 - `ip_queue_worker_wait`
 - `ip_queue_fanout`

- The `ndd` command: A complete list of TCP tuning parameters is available through the `ndd /dev/tcp \?` command. A partial output is shown in the slide above. Note that while a full listing is available, not all of these parameters are intended for tuning.

Note: In Oracle Solaris 11, the preferred method for managing networking tunable parameters is to use the `ipadm` command.

IP Tuning Parameters

Commonly tuned IP parameters:

- forwarding
- ttl
- hoplimit
- hostmodel

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Tuned IP Parameters

The following list describes some commonly tuned IP parameters:

- `forwarding`: Controls whether IPv4 or IPv6 forwards packets with source IPv4 routing options or IPv6 routing headers
- `ttl`: Controls the time to live (TTL) value in the IPv4 header for the outbound IPv4 packets on an IP association. You can set the value from 1 up to 255 (default). It is highly recommended that you do not adjust this parameter.
- `hoplimit`: Sets the value of the hop limit in the IPv6 header for the outbound IP packets on an IP association. You can set the value from 0 up to 255 (default).
- `hostmodel`: Controls send and receive behavior for IPv4 or IPv6 packets on a multi-homed system. This property can have the following values: weak, strong, and src-priority. The default value is weak.

IP Tuning Parameters

- weak
 - **Outgoing packets:** The source address of the packet going out need not match the address configured on the outgoing interface.
 - **Incoming packets:** The destination address of the incoming packet need not match the address configured on the incoming interface.
- strong
 - **Outgoing packets:** The source address of the packet going out must match the address configured on the outgoing interface.
 - **Incoming packets:** The destination address of the incoming packet must match the address configured on the incoming interface.
Note: If a machine has interfaces that cross strict networking domains (for example, a firewall or a VPN node), set this parameter to strong.
- src-priority
 - **Outgoing packets:** If multiple routes for the IP destination in the packet are available, the system prefers routes where the IP source address in the packet is configured on the outgoing interface. If no such route is available, the system falls back to selecting the best route.
 - **Incoming packets:** The destination address of the incoming packet must be configured on any one of the host's interface.

IP Tuning Parameters: ipadm

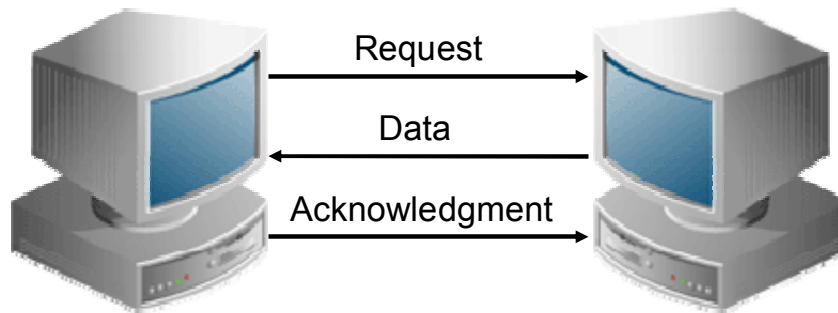
```
# ipadm show-prop ip
```

PROTO	PROPERTY	PERM	CURRENT	PERSISTENT	DEFAULT	POSSIBLE
lpv4	forwarding	rw	on	on	off	on,off
ipv4	ttl	rw	255	--	255	1-255
ipv4	hostmodel	rw	weak	--	weak	strong,src-priority,weak
lpv6	forwarding	rw	off	--	off	on,off
ipv6	hoplimit	rw	255	--	255	1-255
lpv6	hostmodel	rw	weak	--	weak	strong,src-priority,weak

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

TCP Connections

**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Setting up a TCP connection requires one round trip of delay to create a connection. Before data transmission occurs, a three-way handshake must take place, as shown in the slide above.

The reliable or *stateful* nature of TCP connections requires more system resources, including memory, than UDP connections. As a result, applications, such as Simple Network Management Protocol (SNMP) agents, or network environments, such as small local area networks (LANs), can often run UDP-based applications with low error rates and less demand on system resources.

Ideal TCP Tuning

To send data to a destination machine, a source machine first estimates or calculates a size to send. This size is called a window. Window size calculations are based on several runtime factors, including:

- Current link speed from source to destination
- Current average round-trip time (RTT)
- Current congestion and flow statistics

TCP Tuning Parameters

Commonly tuned TCP parameters:

- `ecn`
- `send_buf`
- `recv_buf`
- `max_buf`
- `sack`
- `smallest_anon_port`
- `largest_anon_port`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Commonly Tuned Parameters

The following list describes some commonly tuned parameters:

- `ecn`: Controls Explicit Congestion Notification (ECN) support. ECN can help TCP better handle congestion control. However, there are existing TCP implementations, firewalls, NATs, and other network devices that are confused by this mechanism. If this parameter is set to 0, TCP does not negotiate with a peer that supports the ECN mechanism.
- `send_buf`: Defines the default send window size in bytes. Size can be 4096 up to the current value of `max_buf`.
- `recv_buf`: Defines the default receive window size in bytes. Size can be 2048 up to the current value of `max_buf`.
- `max_buf`: Defines the maximum send and receive buffer size in bytes. This parameter controls how large the send and receive buffers are set to by an application that uses `setsockopt`. If TCP connections are being made in a high-speed network environment, increase the value to match the network link speed.

- `sack`: SACK processing can improve TCP retransmission performance so it should be actively enabled. If set to 2 (active), TCP always sends a SYN segment with the selective acknowledgment (SACK)–permitted option. If TCP receives a SYN segment with a SACK-permitted option and this parameter is set to 1 (passive), TCP responds with a SACK-permitted option. If the parameter is set to 0 (never), TCP does not send a SACK-permitted option, regardless of whether the incoming segment contains the SACK-permitted option. Sometimes, the other side can be confused with the SACK option actively enabled. If this confusion occurs, set the value to 1 so that SACK processing is enabled only when incoming connections allow SACK processing.
- `smallest_anon_port`: This parameter controls the smallest port number TCP can select as an ephemeral port. An application can use an ephemeral port when it creates a connection with a specified protocol and it does not specify a port number. Ephemeral ports are not associated with a specific application. When the connection is closed, the port number can be reused by a different application.
- `largest_anon_port`: This parameter controls the largest port number TCP can select as an ephemeral port. An application can use an ephemeral port when it creates a connection with a specified protocol and it does not specify a port number. Ephemeral ports are not associated with a specific application. When the connection is closed, the port number can be reused by a different application.

TCP Tuning Parameters: `ipadm`

```
# ipadm show-prop tcp
PROTO  PROPERTY          PERM  CURRENT  P  ERSISTENT  DEFAULT  POSSIBLE
tcp    ecn                rw    passive  --          passive  never,passive,active
tcp    extra_priv_ports  rw    2049,4045 --          2049,4045 1-65535
tcp    largest_anon_port rw    65535    --          65535     32768-65535
tcp    max_buf            rw    1048576  --          1048576   128000-1073741824
tcp    recv_buf          rw    128000   --          128000    2048-1048576
tcp    sack              rw    active   --          active     never,passive,active
tcp    send_buf          rw    49152    --          49152     4096-1048576
tcp    smallest_anon_port rw    32768    --          32768     1024-65535
tcp    smallest_nonpriv_port rw 1024     --          1024     1024-32768
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A subset of TCP tuning parameters is available through the `ipadm` command. These are the more commonly tuned TCP parameters.

NFS Tuning Parameters

- Define NFS parameters in the `/etc/system` file.
- Commonly tuned NFS parameters:
 - `nfs:nfs#_pathconf_disable_cache`
 - `nfs:nfs#_cots_timeo`
 - `nfs:nfs#_do_symlink_cache`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Here are some commonly used NFS tuning parameters.

Note: # indicates the NFS version. For example, `nfs3` or `nfs4`.

- `nfs:nfs#_pathconf_disable_cache`: Controls the caching of `pathconf` information for NFS-mounted file systems. The `pathconf` information is cached on a per-file basis. However, if the server can change the information for a specific file dynamically, use this parameter to disable caching. There is no mechanism for the client to validate its cache entry.
- `nfs:nfs#_cots_timeo`: Controls the default RPC timeout for NFS-mounted file systems by using connection-oriented transports such as TCP for the transport protocol. TCP does a good job ensuring requests and responses are delivered appropriately. However, if the round-trip times are very large in a particularly slow network, the NFS client might time out prematurely. Increase this parameter to prevent the client from timing out incorrectly. The range of values is very large, so increasing this value too much might result in situations where a retransmission is not detected for long periods of time.

- `nfs:nfs#_do_symlink_cache`: Controls whether the contents of symbolic link files are cached for NFS-mounted file systems. If a server changes the contents of a symbolic link file without updating the modification time stamp on the file or if the granularity of the time stamp is too large, changes to the contents of the symbolic link file might not be visible on the client for extended periods. In this case, use this parameter to disable the caching of symbolic link contents. Doing so makes the changes immediately visible to applications running on the client.

NFS Tuning Parameters

- `nfs:nfs#_lookup_neg_cache`
- `nfs:nfs#_max_threads`
- `nfs:nfs#_nra`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

- `nfs:nfs#_lookup_neg_cache`: Controls whether a negative name cache is used for NFS-mounted file systems. This negative name cache records file names that were looked up, but were not found. The cache is used to avoid over-the-network look-up requests made for file names that are already known to not exist.
- `nfs:nfs#_max_threads`: Controls the number of kernel threads that perform asynchronous I/O for the NFS client. Because NFS is based on RPC, and RPC is inherently synchronous, separate execution contexts are required to perform NFS operations that are asynchronous from the calling thread. To increase or reduce the number of simultaneous I/O operations that is outstanding at any given time (for example, for a very low bandwidth network), you might want to decrease this value so that the NFS client does not overload the network. Alternately, if the network has a very high bandwidth, and the client and server have sufficient resources, you might want to increase this value. By doing so, you can more effectively use the available network bandwidth and the client and server resources.

- `nfs:nfs#_nra`: Controls the number of read-ahead operations that are queued by the NFS client when sequential access to a file is discovered. These read-ahead operations increase concurrency and read throughput. Each read-ahead request is generally for one logical block of file data. To increase or reduce the number of read-ahead requests that is outstanding for a specific file at any given time (for example, for a very low bandwidth network or on a low memory client), you might want to decrease this value so that the NFS client does not overload the network or the system memory. Alternately, if the network has very high bandwidth, and the client and server have sufficient resources, you might want to increase this value. By doing so, you can more effectively use the available network bandwidth and the client and server resources.

NFS Tuning Parameters

- `nfs:nfs#_bsize`
- `nfs:nacache`
- `nfs:nfs3_jukebox_delay`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

- `nfs:nfs#_bsize`: Controls the logical block size used by the NFS client. This block size represents the amount of data that the client attempts to read from or write to the server when it needs to do an I/O. Examine the value of this parameter when attempting to change the maximum data transfer size. Change this parameter in conjunction with the `nfs:nfs#_max_transfer_size` parameter. If larger transfers are preferred, increase both parameters. If smaller transfers are preferred, just reducing this parameter should suffice.
- `nfs:nacache`: Tunes the number of hash queues that access the file access cache on the NFS client. The file access cache stores file access rights that users have with respect to files that they are trying to access. The cache itself is dynamically allocated. However, the hash queues used to index into the cache are statically allocated. The algorithm assumes that there is one access cache entry per active file and four of these access cache entries per hash bucket. Thus, by default, the value of this parameter is set to the value of then `rnode` parameter. Examine the value of this parameter if the basic assumption of one access cache entry per file would be violated. This violation could occur for systems in a time-sharing mode where multiple users are accessing the same file at about the same time. In this case, it might be helpful to increase the expected size of the access cache so that the hashed access to the cache stays efficient.

- `nfs:nfs3_jukebox_delay`: Controls the duration of time that the NFS version 3 client waits to transmit a new request after receiving the `NFS3ERR_JUKEBOX` error from a previous request. The `NFS3ERR_JUKEBOX` error is generally returned from the server when the file is temporarily unavailable for some reason. This error is generally associated with hierarchical storage, and DVD or tape jukeboxes.

Additional Network Tuning Parameters

- IP Quality of Service (IPQoS)
 - `_policy_mask`
- Network cache and accelerator (NCA)
 - Typically configured on dedicated web servers
 - Example: A system with 4-GB memory
 - `set sq_max_size=0`
 - `set ge:ge_intr_mode=1`
 - `set nca:nca_conn_hash_size=82500`
 - `set nca:nca_conn_req_max_q=100000`
 - `set nca:nca_conn_req_max_q0=100000`
 - `set nca:nca_ppmax=393216`
 - `set nca:nca_vpmax=393216`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Here are some additional important networking tuning parameters.

- IP Quality of Service (IPQoS)

The `_policy_mask` parameter enables or disables IPQoS processing in any of the following callout positions: forward outbound, forward inbound, local outbound, and local inbound. The mask is a 15 (0x0f) bit value. A “1” in any of the position masks or disables IPQoS processing in that particular callout position.

The callout positions are defined as:

- Bit 1 = Local inbound
- Bit 2 = Local outbound
- Bit 3 = Forward inbound
- Bit 4 = Forward outbound
- Bits 5-8 = Not used

For example, a value of 0x01 disables IPQoS processing for all the local inbound packets. By default, IPQoS processing is enabled on all callout positions.

- Network cache and accelerator (NCA)

Typically configured on dedicated web servers. For example, a system with 4-GB memory should have the following parameters set in the `/etc/system` file. Use `pagesize` to determine your system's page size.

```
set sq_max_size=0
set ge:ge_intr_mode=1
set nca:nca_conn_hash_size=82500
set nca:nca_conn_req_max_q=100000
set nca:nca_conn_req_max_q0=100000
set nca:nca_ppmax=393216
set nca:nca_vpmax=393216
```

Per-Route Tuning

- You can associate a tuning property with an entry in the route table.
- Use the `route change` command to make the association.

```
# netstat -rn
Routing Table: IPv4
  Destination      Gateway          Flags   Ref    Use    Interface
-----
default           10.150.36.1      UG      4      805016 net0
10.150.36.0       10.150.36.245    U       6      2346003 net0
10.150.37.0       10.150.37.77     U       6      6384011 net1
127.0.0.1         127.0.0.1        UH      2       996     lo0
# route change -net 10.150.37.0 -recvpipe 512000
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You can perform per-route tuning by associating a tuning property with a IPv4 or IPv6 routing table entry.

In this example, a system has two different network interfaces, `net0` has a 1-gigabit Ethernet interface and `net1` has a 10-gigabit Ethernet interface. By default, the system applies the default `recv_maxbuf` (128,000 bytes) to both interfaces. This default is sufficient for the 1-gigabit Ethernet interface, but may not be sufficient for the 10-gigabit Ethernet interface.

Instead of increasing the system's default for `recv_maxbuf`, you can associate a different default TCP receive window size (512,000) to the 10-gigabit Ethernet interface routing entry. By making this association, all TCP connections going through the route have the increased receive window size.

Isolating Problems

- Tune individual host systems
- Ensure individual systems at an expected level as defined in Service Level Agreement (SLA)
- Tune client-side NFS by implementing CacheFS
- Tune applications on the server
- Tune physical devices on the network

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Before you tune the network, it is important to tune individual host systems properly. Tuning the client and server systems significantly improves the performance of the network.

Network performance depends on the collective performance of individual systems on the network. Therefore, before tuning the network, ensure that the individual systems perform at the expected level, as defined in Service Level Agreement (SLA).

After you tune the individual systems on the network, tune client-side NFS performance by implementing the CacheFS file system or by setting the `actimeo` parameter.

After the individual systems perform at the expected level and the NFS activity is tuned, tune the applications on the server.

After you tune the individual systems, the NFS, and the applications, tune the physical devices on the network. To do this, collect information on network performance. You use the `ping`, `spray`, `snoop`, `netstat`, and `nfsstat` commands for detailed information on the performance of the network.

Quiz

Network interface packet counts can be fetched with the following command:

- a. `netstat -i`
- b. `netstat -r`
- c. `traceroute <name of the destination system>`
- d. None of the above

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

When you use the `ping` command with the `-s` option, one data packet is sent to the specified host per second. The command then prints each response message and the time taken for the round trip.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Which of the following factors degrade TCP performance?

- a. Retransmissions
- b. Duplicate packets
- c. Listen queues
- d. All of the above

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: d

Summary

In this lesson, you should have learned how to:

- Describe terms used for network analysis
- Describe network utilization
- Understand the effects of misconfigured components
- Describe the differences between Solaris 10 and Solaris 11 networking
- Describe the benefits of IPMP
- Describe the benefits of link aggregation
- Describe network monitoring commands commonly used in Solaris 11
- Describe network tuning parameters commonly used in Solaris 11

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 11 Overview: Monitoring Network Performance

This practice covers the following topics:

- Optimizing the network for optimal performance
- Monitoring the network performance

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

12

Resource Management

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe the Oracle Solaris 11 resource management capabilities
- Describe projects
- Describe process scheduling
- Describe resource pool
- Configure resource pool
- Describe memory capping

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

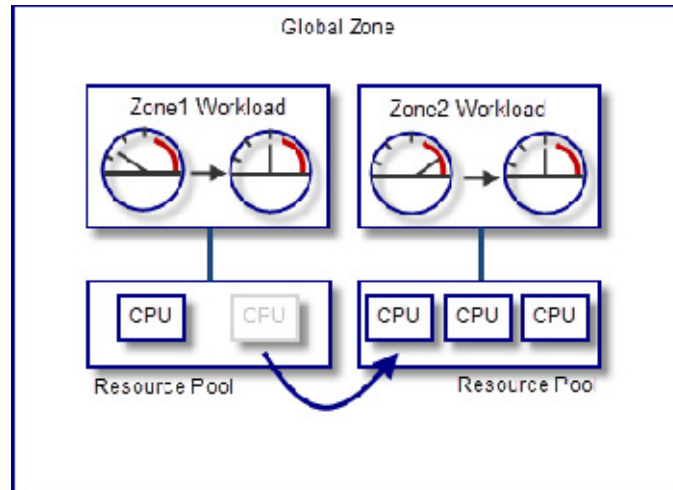
Agenda

- Oracle Solaris 11 resource management
 - Projects and tasks
 - Resource control
 - Process scheduling
 - Resource pools
 - Resource capping

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Resource Management



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris 11 resource management software increases resource availability for users, groups, and applications. It provides the ability to allocate and control major system resources, such as CPU, memory, and process scheduler. After a resource policy is set, the system administrator can rest assured that mission-critical applications will get the resources they demand. Resource management allows you to modify the default behavior of the operating system with respect to different workloads. You can use Oracle Solaris 11 resource management to do the following:

- Deny resources or prefer one application to another for a larger set of allocations than otherwise permitted
- Treat certain allocations collectively instead of through isolated mechanisms

When to Use Resource Management

- Ensure that your applications have the required response times.
- Increase resource utilization.
- Server consolidation
 - Support a large or varied user population.
 - Limit resource utilization for software license compliance.
 - Ensure that customers get only the resources for which they have paid.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Use resource management to ensure that your applications have the required response times. Resource management can also increase resource utilization. By categorizing and prioritizing usage, you can effectively use reserve capacity during off-peak periods, often eliminating the need for additional processing power.

Server Consolidation

Resource management is ideal for environments that consolidate several applications on a single server. If you are providing Internet and application services, you can use resource management to do the following:

- Host multiple web servers on a single machine. You can control the resource consumption for each website and you can protect each site from the potential excesses of other sites.
- Prevent a faulty common gateway interface (CGI) script from exhausting CPU resources.
- Stop an incorrectly behaving application from leaking all available virtual memory.
- Limit resource utilization for software license compliance.
- Ensure that customers get only the resources for which they have paid.

- Ensure that one customer's applications are not affected by those of another customer, although they run at the same site.
- Provide differentiated levels or classes of service on the same machine.
- Obtain accounting information for billing purposes.
- Support a large or varied user population.

Use resource management features in any system that has a large, diverse user base. If you have a mix of workloads, the software can be configured to give priority to specific projects.

Resource management is also ideal for supporting thin-client systems. These platforms provide stateless consoles with frame buffers and input devices, such as smart cards. The actual computation is done on a shared server, resulting in a timesharing type of environment. Use resource management features to isolate the users on the server. Then, a user who generates excess load does not monopolize hardware resources and significantly impact others who use the system.

Agenda

- Oracle Solaris 11 resource management
- **Projects and tasks**
- Resource control
- Process scheduling
- Resource pools
- Resource capping

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Projects and Tasks

- Facilities that can be used to separate and identify workloads:
 - Projects
 - Tasks
- A new task is started in a project:
 - When a new session is opened by a user login
 - By a `cron`, `newtask`, `setproject`, or `su` command
 - When SMF starts a service
- The controls specified in the project are set on the process, task, and project.
- All processes and tasks that are created within the project inherit these controls.
- The project identifier is an administrative identifier that is used to identify related work.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In the Oracle Solaris 11 system, you have two facilities that can be used to separate and identify workloads: the *project* and the *task*. The project provides a network-wide administrative identifier for related work. The task collects a group of processes into a manageable entity that represents a workload component. A new task is started in a project when a new session is opened by a user login or by a `cron`, `newtask`, `setproject`, or `su` command or when SMF starts a service. Each process belongs to only one task, and each task belongs to only one project. Projects and tasks are the basic entities that are used to identify workloads in the Solaris 11 operating system.

The controls specified in the project name service database are set on the process, task, and project. All processes and tasks that are created within the project inherit these controls.

The project identifier is an administrative identifier that is used to identify related work. The project identifier can be thought of as a workload tag equivalent to the user and group identifiers. A user or group can belong to one or more projects. These projects can be used to represent the workloads in which the user (or group of users) is allowed to participate. This membership can then be the basis of chargeback that is based on, for example, usage or initial resource allocations. Although a user must be assigned to a default project, the processes that the user launches can be associated with any of the projects of which that user is a member.

Project Database

- It contains all the information related to the projects.
- Project data can be stored in the local file `/etc/project` or in a naming service.
- Local `/etc/project` file properties:

Property	Description
projname	The name of the project
projid	The project's unique numerical ID (PROJID) within the system
comment	A description of the project
user-list	A comma-separated list of users who are allowed in the project
group-list	A comma-separated list of groups who are allowed in the project
attributes	A semicolon-separated list of name-value pairs, such as resource

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Projects and Tasks

The project database contains all the information related to the projects defined in the system. The project data can be stored in a local `/etc/project` file or in a naming service such as DNS, NIS, or LDAP.

The `/etc/project` file properties are shown in this slide.

Project and Task Commands

Command	Description
projects	Displays project memberships for users. Lists projects from project database.
projadd	Adds a new project entry to the <code>/etc/project</code> file
projmod	Modifies information for a project on the local system
projdel	Deletes a project from the local system
useradd	Adds default project definitions to the local files
userdel	Deletes a user's account from the local file
usermod	Modifies a user's login information on the system
newtask	Executes the user's default shell or specified command, placing the execution command in a new task that is owned by the specified project

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This table lists the commands you use to manage projects and tasks.

Displaying the Project Database

```
root@s11-serv2:~# projects -l
system
    projid : 0
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
user.root
    projid : 1
    comment: ""
    users  : (none)
    groups : (none)
    attribs:
...
root@s11-serv2:~# projects root
user.root default
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This example shows using the `projects -l` command to display the contents of the projects database. Using the `projects` command followed by a username will display a list of projects of which the user is a member.

Adding a Project to the Project Database

```

root@s11-serv2:~# projadd -U mary,tom eng_team1
root@s11-serv2:~# projmod -s \
-K "project.cpu-shares=(privileged,3,none)" \
-K project.pool=eng_FSS eng_team1
root@s11-serv2:~# projects -l
...
eng_team1
    projid : 100
    comment: ""
    users  : mary
            tom
    groups : (none)
    attribs: project.cpu-shares=(privileged,3,none)
            project.pool=eng_FSS
...

```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Displaying the project Database

This example shows using the `projadd` and `projmod` commands to add and configure a project named `eng_team1` in the projects database. Users `mary` and `tom` have been assigned to the project. All tasks performed within the context of this project are restricted to three CPU shares. The task processes are under the control of the fair-share scheduler (FSS).

Agenda

- Oracle Solaris 11 resource management
- Projects and tasks
- **Resource control**
- Process scheduling
- Resource pools
- Resource capping

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Resource Management Control Mechanisms

- Constraint mechanisms
- Scheduling mechanisms
- Partitioning mechanisms

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

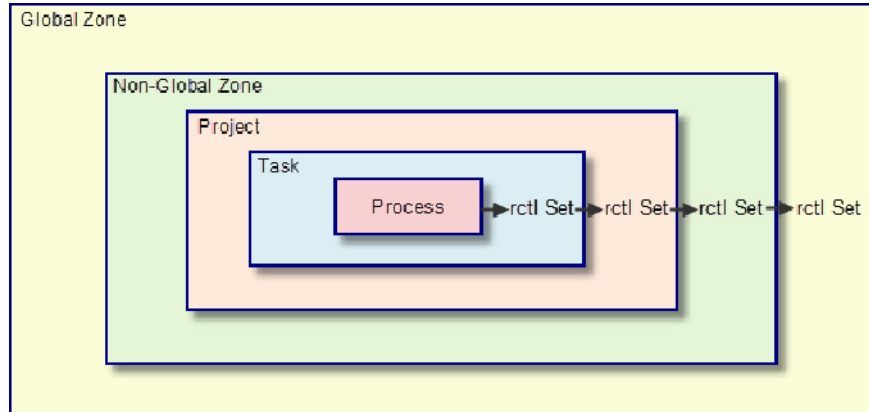
Constraints allow the administrator or application developer to set bounds on the consumption of specific resources for a workload. With known bounds, modeling resource consumption scenarios becomes a simpler process. Bounds can also be used to control ill-behaved applications that would otherwise compromise system performance or availability through unregulated resource requests.

Constraints do present complications for the application. The relationship between the application and the system can be modified to the point that the application is no longer able to function. One approach that can mitigate this risk is to gradually narrow the constraints on applications with unknown resource behavior.

Scheduling refers to making a sequence of allocation decisions at specific intervals. The decision that is made is based on a predictable algorithm. An application that does not need its current allocation leaves the resource available for another application's use. Scheduling-based resource management enables full utilization of an under-committed configuration, while providing controlled allocations in a critically committed or overcommitted scenario. The underlying algorithm defines how the term "controlled" is interpreted. In some instances, the scheduling algorithm might guarantee that all applications have some access to the resource.

Partitioning is used to bind a workload to a subset of the system's available resources. This binding guarantees that a known amount of resources is always available to the workload. The resource pools functionality enables you to limit workloads to specific subsets of the machine.

Resource Control Enforcement

**ORACLE**

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

More than one resource control can exist on a resource. A resource control can exist at each containment level in the process model. If resource controls are active on the same resource at different containment levels, the smallest container's control is enforced first. For example, if a resource control is set at the project level and the resource control is set at the nonglobal zone level, the project level is enforced first.

Resource Control Values, Privileges, and Actions

- The value is the threshold (or cap) on the resource control.
- Each threshold value is associated with a privilege:
 - `basic`
 - `privilege`
 - `system`
- For each threshold value, you associate one or more actions.
 - `none`
 - `deny`
 - `signal=`

The Oracle logo, consisting of the word "ORACLE" in white, uppercase letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Resource controls have three properties: value, privilege, and action.

The value is the threshold (or cap) on the resource control. This is the enforcement point where local actions are triggered. Each threshold value is associated with a privilege level:

- **Basic:** This can be modified by the owner of the calling process.
- **Privileged:** This can be modified only by privileged (root) callers.
- **System:** This is fixed for the duration of the operating system instance.

For each threshold value, you associate one or more actions. Following are the actions you can take when a threshold is exceeded:

- **none:** No action is taken on resource requests for an amount that is greater than the threshold.
- **deny:** You can deny resource requests for an amount that is greater than the threshold.
- **signal=:** You can enable a global signal message action when the resource control is exceeded.

For example:

```
# rctladm project.max-sem-ids
```

```
project.max-sem-ids      syslog=off      [ no-basic deny count ]
```

Local actions are taken on a process that attempts to exceed the control value.

```
# prctl -n project.max-sem-ids $$
```

```
process: 1512: sh
```

NAME	PRIVILEGE	VALUE	FLAG	ACTION	RECIPIENT
project.max-sem-ids					
	privileged	128	-	deny	-
	system	16.8M	max	deny	-

Configuring Resource Controls in a Project

- Resource controls can be configured by using the project database.
 - The `attrs` field
- Example:

```
root@s11-serv2:~# projects -l
...
oracle
  projid : 100
  comment: ""
  users  : oracle
  groups : (none)
  attrs: project.max-sem-ids=(priv,100,deny)
        project.max-sem-nsems=(priv,256,deny)
        project.max-shm-ids=(priv,100,deny)
        project.max-shm-memory=(priv,4294967296,deny)
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The resource controls can be configured through the project database by using the `attrs` field. For example:

```
root@s11-serv2:~# projects -l
...
oracle
  projid : 100
  comment: ""
  users  : oracle
  groups : (none)
  attrs: project.max-sem-ids=(priv,100,deny)
        project.max-sem-nsems=(priv,256,deny)
        project.max-shm-ids=(priv,100,deny)
        project.max-shm-memory=(priv,4294967296,deny)
```

Configuring Resource Controls

This example shows project oracle. The resource controls are:

- `project.max-sem-ids` – This sets a cap on the maximum number of message queue IDs allowed for this project. The field in parentheses (`priv,100,deny`) indicates that this is a privileged operation, the message queue cap is 100, and any attempt to exceed the cap will be denied.
- `project.max-sem-nsems` – This sets a cap on the maximum number of semaphores per semaphore set allowed for this project. The field in parentheses (`priv,256,deny`) indicates that this is a privileged operation, the semaphore cap is 256, and any attempt to exceed the cap will be denied.

This example shows project oracle. The resource controls are:

- `project.max-shm-ids` – This sets a cap on the maximum number of shared memory IDs allowed for this project. The field in parentheses (`priv,100,deny`) indicates that this is a privileged operation, the shared memory ID cap is 100, and any attempt to exceed the cap will be denied.
- `project.max-shm-memory` – This sets a cap on the total amount of shared memory allowed for this project. The field in parentheses (`priv,4294967296,deny`) indicates that this is a privileged operation, the shared memory cap is 4 GB, and any attempt to exceed the cap will be denied.

Resource Control Commands

Command	Description
rctladm	Allows you to make runtime interrogations of and modifications to the resource controls facility, with global scope
prctl	Allows you to make runtime interrogations of and modifications to the resource controls facility, with local scope
ipcs	Allows you to observe which IPC objects are contributing to a project's usage

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This table lists commands used to view and modify system resource controls.

Agenda

- Oracle Solaris 11 resource management
- Projects and tasks
- Resource control
- **Process scheduling**
- Resource pools
- Resource capping

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Fair-Share Scheduler (FSS)

- FSS controls the allocation of available CPU resources among workloads, based on their importance.
- FSS is integrated into the project framework.
- Workload importance is expressed by the number of shares of CPU resources that you assign to each workload.
- FSS guarantees a fair dispersion of CPU resources among projects, which is based on allocated share.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

You use the fair-share scheduler (FSS) to control the allocation of available CPU resources among workloads, based on their importance. The fair-share scheduler allows more flexible process priority management that integrates with the project framework.

Workload importance is expressed by the number of shares of CPU resources that you assign to each workload. You give each project CPU shares to control the project's entitlement to CPU resources. The FSS guarantees a fair dispersion of CPU resources among projects, which is based on allocated shares, independent of the number of processes that are attached to a project. The FSS achieves fairness by reducing a project's entitlement for heavy CPU usage and increasing its entitlement for light usage, in accordance with other projects.

Fair-Share Scheduler (FSS)

- Each project is allocated several CPU shares by using the `project.cpu-shares` resource control.
 - Each project is allocated CPU time, based on its `cpu-shares` value divided by the sum of the `cpu-shares` values for all active projects.

- Temporarily change the number of shares:

```
# prctl -r -n project.cpu-shares -v 3 -i project oracle
```

- Persistent change across reboots:

```
# projmod -sK "project.cpu-shares=(privileged,3,none)" oracle
```

- Making FSS the default scheduler:

```
# dispadmin -d FSS
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

How the FSS works:

Each project is allocated a certain number of CPU shares using the `project.cpu-shares` resource control. Each project is allocated CPU time, based on its `cpu-shares` value divided by the sum of the `cpu-shares` values for all active project.

If you want to temporarily change the number of shares assigned to a project without altering the project's attributes in the project database, use the `prctl` command. For example, to change the value of the Oracle project `project.cpu-shares` resource control to 3 while processes associated with that project are running, type the following:

```
# prctl -r -n project.cpu-shares -v 3 -i project oracle
```

To make the same change persistent across reboots, use the `projmod` command:

```
# projmod -sK "project.cpu-shares=(privileged,3,none)" oracle
```

To make FSS the default scheduling class for the entire Solaris 11 system, use the `dispadmin` command:

```
# dispadmin -d FSS
```

Agenda

- Oracle Solaris 11 resource management
- Projects and tasks
- Resource control
- Process scheduling
- **Resource pools**
- Resource capping

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Resource Pools

- Resource pools separate workloads so that workload consumption of resources does not overlap.
- Resource pools provide a persistent configuration mechanism for:
 - Processor set (`pset`) configuration
 - Scheduling class assignment (optional)
- Resource pools are persistently enabled by using the SMF service `svc:/system/pools:default`.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

By using resource pools, you can separate workloads so that workload consumption of certain resources does not overlap. This resource reservation helps to achieve predictable performance on systems with mixed workloads. Resource pools provide a persistent configuration mechanism for processor set (`pset`) configuration and, optionally, scheduling class assignment.

Resource pools are persistently enabled by using the SMF service

`svc:/system/pools:default`.

Dynamic resource pools (DRPs) provide a mechanism for dynamically adjusting each pool's resource allocation in response to system events and application load changes. DRPs simplify and reduce the number of decisions required from an administrator. Adjustments are automatically made to preserve the system performance goals specified by an administrator.

Dynamic resource pools are persistently enabled by using the SMF service

`svc:/system/pools/dynamic`.

Dynamic Resource Pools

- DRPs adjust each pool's resource allocation in response to system events and application load changes.
- Adjustments are automatically made to preserve the system performance goals.
- DRP is control by the `poolld` daemon.
- DRPs are persistently enabled by using the SMF service `svc:/system/pools/dynamic`.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

DRPs provide a mechanism for dynamically adjusting each pool's resource allocation in response to system events and application load changes. DRPs simplify and reduce the number of decisions required from an administrator. Adjustments are automatically made to preserve the system performance goals specified by an administrator.

DRPs are controlled by the `poolld` daemon. Periodically, `poolld` examines the load on the system and determines whether intervention is required to enable the system to maintain optimal performance with respect to resource consumption.

DRPs are persistently enabled by using the SMF service `svc:/system/pools/dynamic`.

Resource Pool Properties

Pool properties fall into two categories:

- Configuration constraints
 - Maximum and minimum
- Objective
 - Workload-dependent
 - Workload-independent

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Each pool property falls into the following categories:

- **Configuration:** Defines resource pool logging and monitoring attributes
- **Constraints:** Defines boundaries of a property. The typical constraints are the maximum and minimum allocations.
- **Objective:** Changes the resource assignments of the current configuration to generate new candidate configurations that observe the established constraints. An objective has the following categories:
 - Workload-dependent: A workload-dependent objective varies according to the conditions imposed by the workload.
 - Workload-independent: A workload-independent objective does not vary according to the conditions imposed by the workload.

Resource Pool Properties

Property	Description
pset.max	Maximum number of CPUs for this processor set
pset.min	Minimum number of CPUs for this processor set
cpu.pinned	CPU dedicated to this processor set
system.poold.objectives	System objective type
pset.poold.objectives	Processor set objective type
pool.importance	User-assigned importance
system.poold.log-level	Logging level
system.poold.log-location	Logging location
system.poold.monitor-interval	Monitoring sample interval
system.poold.history-file	Decision history location

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Configuration Pool Objectives

Objective	Description
wt-load	The <code>wt-load</code> objective favors configurations that match resource allocations to resource utilizations. A resource set that uses more resources is given more resources when this objective is active. <code>wt-load</code> means <i>weighted load</i> .
locality	The <code>locality</code> objective is used to influence the impact of latency (or time) between resources. The values are: tight - Maximize resource locality is favored. loose - Minimize resource locality is favored. none - Not influenced by resource locality This objective falls into the workload-independent category.
utilization	This favors configurations that allocate resources to partitions that are not meeting the specified utilization objective. This falls into the workload-dependent category.

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, with a registered trademark symbol (®) to the upper right.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Resource Pool Commands

Command	Description
poolcfg	Creates and modifies resource pool configuration files
pooladm	Activates and deactivates the resource pools facility
poolstat	Reports active pool statistics
poolbind	Binds processes, tasks, or projects to resource pools
poold	Automated resource pools partitioning daemon

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This table lists the resource pool commands.

Configuring Static Resource Pools

- The `/etc/pooladm.conf` file contains the static pools configuration.
- Steps for configuring a static resource pool:
 1. Create the `/etc/pooladm.conf` file.
 2. Create a processor set (`pset`).
 3. Create a resource pool.
 4. Associate the resource pool with the processor set.
 5. Commit the static pool configuration.

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Configuring Resource Pools

The `/etc/pooladm.conf` configuration file describes the static pools configuration. A static configuration represents the way in which an administrator would like a system to be configured with respect to resource pools functionality. An alternate file name can be specified. When the service management facility (SMF) or the `pooladm -e` command is used to enable the resource pools framework, then, if an `/etc/pooladm.conf` file exists, the configuration contained in the file is applied to the system.

If the `/etc/pooladm.conf` configuration file does not exist, you create it by using the `poolcfg -c discover` command.

Configuring Static Resource Pools

```
root@s11-serv2:~# ls /etc/pooladm.conf
/etc/pooladm.conf: No such file or directory
root@s11-serv2:~# poolcfg -c discover
root@s11-serv1:~# ls /etc/pooladm.conf
/etc/pooladm.conf
root@s11-serv2:~# poolcfg -c 'create pset eng_pset1 \
(uint pset.min = 1; uint pset.max = 2)'
root@s11-serv2:~# poolcfg -c 'create pool eng_pool'
root@s11-serv2:~# poolcfg -c 'associate pool eng_pool \
(pset eng_pset1)'
root@s11-serv2:~# pooladm -c
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows you how to configure a static resource pool. The example starts out with no `/etc/pooladm.conf` file present. You use the `poolcfg -c discover` command to create the pool configuration file. Next, you create a processor set named `eng_pset1` that consists of a maximum of one CPU and a maximum of two CPUs. You create a pool named `eng_pool` and associate the `eng_pool` pool with the `eng_pset1` processor set. Finally, you commit the pool configuration by using the `pooladm -c` command.

Configuring Static Resource Pools

```
root@s11-serv2:~# pooladm
...
pool eng_pool
    int      pool.sys_id 1
    boolean  pool.active true
    boolean  pool.default false
    int      pool.importance 1
    string   pool.comment
    pset     eng_pset1
pset eng_pset1
    int      pset.sys_id 1
    boolean  pset.default false
    uint     pset.min 1
    uint     pset.max 2
    string   pset.units population
    uint     pset.load 0
    uint     pset.size 1
    string   pset.comment
...
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows you the resulting entries in the pool configuration file.

Configuring Objectives

```
root@s11-serv2:~# poolcfg -c 'modify s11-serv2 \  
(string system.poold.objectives="wt-load")'  
root@s11-serv2:~# poolcfg -c info  
system s11-serv2  
    string system.comment  
    int system.version 1  
    boolean system.bind-default true  
    int system.poold.pid 4679  
    string system.poold.objectives wt-load  
...
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows you how to add an objective to the pool configuration. In this example, the `wt-load` objective is added to the system `s11-serv2`.

Adding FSS to a Pool

```
root@s11-serv2:~# # poolcfg -c 'modify pool eng_pool \  
  (string pool.scheduler="FSS")'  
root@s11-serv2:~# pooladm  
...  
pool eng_pool  
    int      pool.sys_id 1  
    boolean  pool.active true  
    boolean  pool.default false  
    int      pool.importance 1  
    string   pool.comment  
    string   pool.scheduler FSS  
    pset     eng_pset1  
...  

```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows you how to add the FSS scheduler to the pool. In this example, FSS is added to the `eng_pool`.

poolstat Command

```
root@s11-serv2:~# poolstat -r pset -p eng_FSS 5 5
id pool      type rid rset      min  max size used load
  1 eng_FSS  pset   1 eng_pset1   1    2   1 0.00 0.99
...
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Configuring Objectives

The `poolstat` command iteratively examines all active pools on the system. It reports statistics based on the selected output mode. `poolstat` provides options to examine only specified pools and report resource set-specific statistics. Without options, `poolstat` examines all pools, reports basic statistics for their resource sets, and exits. The command output fields are defined as follows:

- **id** - Pool ID
- **pool** - Pool name
- **Rid** - Resource set id
- **rset** - Resource set name
- **type** - Resource set type
- **min** - Minimum resource set size
- **max** - Maximum resource set size
- **size** - Current resource set size
- **used** - The measure of how much of the resource set is currently in use
- **load** - The absolute representation of the load that is put on the resource set

Agenda

- Oracle Solaris 11 resource management
- Projects and tasks
- Resource control
- Process scheduling
- Resource pools
- Resource capping

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Resource Capping

- A resource cap is an upper bound placed on the consumption of a resource.
- Caps can be defined in the project database.
 - The `rcap.max-rss` attribute

```
root@s11-serv1:~# projmod -a -K rcap.max-rss=4GB oracle
```

- Enable resource capping:
 - The `svc:/system/rcap:default` SMF service
- When enabled, the `rcapd` daemon:
 - Repeatedly samples the resource utilization of projects that have physical memory caps
 - Reduces the resource consumption of projects

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

A resource cap is an upper bound placed on the consumption of a resource, such as physical memory. Per-project physical memory caps are supported. The resource-capping daemon `rcapd` and associated utilities provide mechanisms for physical memory resource cap enforcement and administration. The resource caps are defined by using attributes of project entries in the project database. To define a physical memory resource cap for a project, establish a resident set size (RSS) cap by adding the `rcap.max-rss` attribute to the project database entry. RSS is the total amount of physical memory, in bytes, that is available to processes in the project.

You can enable resource capping with the SMF service `svc:/system/rcap:default`.

The `rcapd` daemon repeatedly samples the resource utilization of projects that have physical memory caps. When the system's physical memory utilization exceeds the threshold for cap enforcement, the daemon takes action to reduce the resource consumption of projects with memory caps to levels at or below the caps.

Resource-Capping Commands

Objective	Description
rcapadm	Configures the resource-capping daemon, displays the current status of the resource-capping daemon if it has been configured, and enables or disables resource capping. It is also used to set a temporary memory cap.
rcapstat	Monitors the resource utilization of capped projects
rcapd	The resource-capping daemon

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, is positioned on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Configuration Objectives

This table lists the resource-capping commands.

rcapstat Command

```
root@s11-serv2:~# rcapstat
```

id	project	nproc	vm	rss	cap	at	avgat	pg	avgpg
100	eng_team1	1	148K	280K	50K	1412K	1412K	68K	68K
101	eng_team2	-	148K	372K	5120K	0K	0K	0K	0K
102	eng_team3	-	148K	372K	5120K	0K	0K	0K	0K
103	eng_team4	1	148K	280K	50K	1412K	1412K	68K	68K



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows the output of the `rcapstat` command.

The `rcapstat` command fields are:

- `id`: The project ID of the capped project
- `project`: The project name
- `nproc`: The number of processes in the project
- `vm`: The total amount of virtual memory size used by processes in the project
- `rss`: The amount of the total resident set size (RSS) of the processes in the project
- `cap`: The RSS cap defined for the project
- `at`: The total amount of memory that `rcapd` attempted to page out since the last `rcapstat` sample
- `avgat`: The average amount of memory that `rcapd` attempted to page out during each sample cycle that occurred since the last `rcapstat` sample
- `pg`: The total amount of memory that `rcapd` successfully paged out since the last `rcapstat` sample
- `avgpg`: An estimate of the average amount of memory that `rcapd` successfully paged out during each sample cycle that occurred since the last `rcapstat` sample

Quiz

In the Oracle Solaris 11 system, you have two facilities that can be used to separate and identify workloads. These facilities are:

- a. Processes and tasks
- b. Profiles and tasks
- c. Projects and profiles
- d. Projects and tasks

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: d

Quiz

Which project database property identifies the users associated with a project?

- a. users
- b. project-users
- c. user-list
- d. attributes

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Which command is used to list projects from the project database?

- a. projects
- b. projstat
- c. projlist
- d. project-list

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Resource management constraint mechanisms allow the administrator or application developer to set bounds on the consumption of specific resources for a workload.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Summary

In this lesson, you should have learned how to:

- Describe the Oracle Solaris 11 resource management capabilities
- Describe projects
- Describe process scheduling
- Describe resource pool
- Configure resource pool
- Describe memory capping

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 12: Overview

This practice covers managing resources.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle Solaris Virtualization Performance Management

13

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to:

- Describe Oracle Solaris Zones
- Monitor zone resource consumption
- Control zone resources
- Describe VM for SPARC
- Control Oracle VM for SPARC resources

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Oracle Solaris Zones
- Monitoring Oracle Solaris zones
- Controlling zone resources
- Controlling Oracle VM for SPARC resources

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

How Zones Work

- Two types of zones: global and nonglobal
- Every Oracle Solaris system contains a *global* zone.
 - Default zone for the system
 - Used for systemwide administrative control
 - The only zone bootable from the system hardware
- *Nonglobal* zones isolate software applications or services by using flexible, software-defined boundaries.
- A process running in a nonglobal zone:
 - Can manipulate, monitor, and directly communicate with other processes that are assigned to the same zone
 - Cannot perform these functions with processes that are assigned to other zones

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Zones are ideal for environments that consolidate several applications on a single server. The cost and complexity of managing numerous machines make it advantageous to consolidate several applications on larger, more scalable servers.

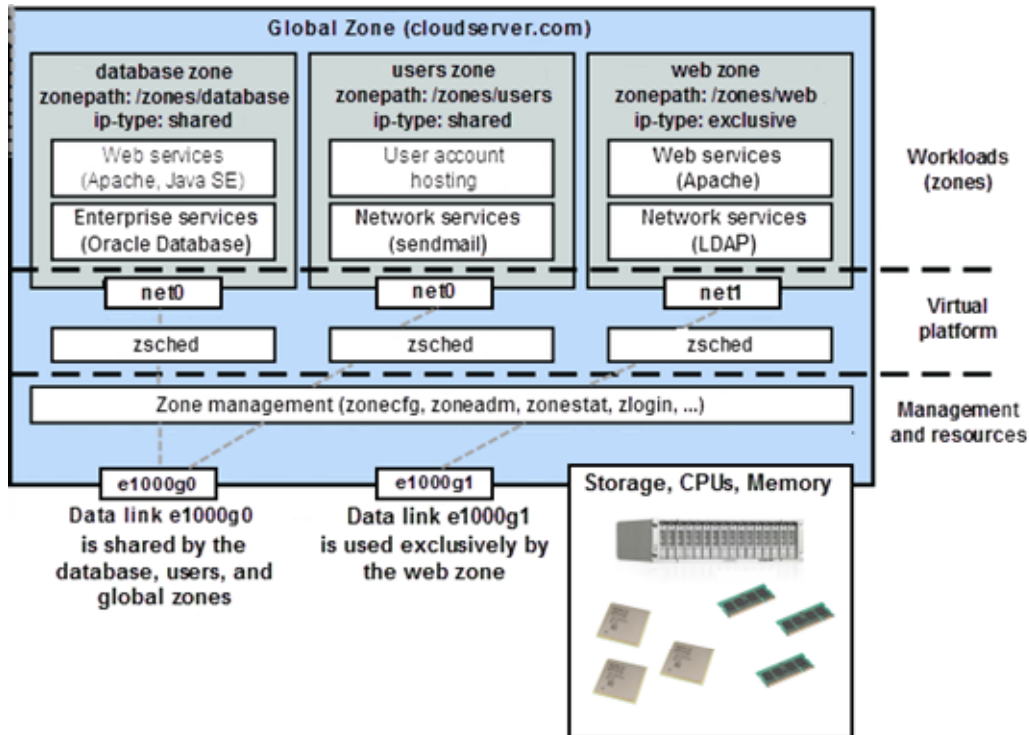
There are two types of zones: global zone and nonglobal zones.

Every Oracle Solaris system contains a *global* zone. The global zone is both the default zone for the system and the zone used for systemwide administrative control. The global zone is the only zone from which a nonglobal zone can be configured, installed, managed, or uninstalled. Only the global zone is bootable from the system hardware.

A *nonglobal* zone functions as a container. One or more applications can run in this container without interacting with the rest of the system. Zones isolate software applications or services by using flexible, software-defined boundaries. Applications that are running in the same instance of the Oracle Solaris operating system can then be managed independently of one other. Thus, different versions of the same application can be run in different zones, to match the requirements of your configuration.

A process assigned to a zone can manipulate, monitor, and directly communicate with other processes that are assigned to the same zone. A process cannot perform these functions with processes that are assigned to other zones in the system or with processes that are not assigned to a zone. Processes that are assigned to different zones are only able to communicate through network APIs.

How Zones Work



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Each zone, including the global zone, is assigned a zone name. The global zone always has the name global. Each zone is also given a unique numeric identifier, which is assigned by the system when the zone is booted. The global zone is always mapped to ID 0.

The figure in this slide shows a system with three zones (database, users, and web). Each of the zones is running a workload unrelated to the workloads of the other zones. This example illustrates that different applications can be run without negative consequences in different zones, to match the consolidation requirements. Each zone can provide a customized set of services.

Each zone workload can be allocated system resources such as networks, storage, CPUs, and memory to meet performance requirements.

Global Zone Characteristics

Type of Zone	Characteristics
Global	Is assigned ID 0 by the system
	Provides the single instance of the Oracle Solaris kernel that is bootable and running on the system
	Contains a complete installation of the Oracle Solaris system software packages
	Can contain additional software packages or additional software, directories, files, and other data not installed through packages
	Holds configuration information specific to the global zone only, such as the global zone host name and network configuration
	Is the only zone that is aware of all devices and all file systems
	Is the only zone with knowledge of nonglobal zone existence and configuration
	Is the only zone from which a nonglobal zone can be configured, installed, managed, or uninstalled

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide lists the global zone characteristics.

Nonglobal Zone Characteristics

Type of Zone	Characteristics
Nonglobal	Is assigned a zone ID by the system when the zone is booted
	Shares operation under the Oracle Solaris kernel booted from the global zone
	Contains an installed subset of the complete Oracle Solaris operating system software packages
	Can contain additional installed software packages
	Can contain additional software, directories, files, and other data created on the nonglobal zone that are not installed through packages
	Is not aware of the existence of any other zones
	Cannot install, manage, or uninstall other zones, including itself

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide lists the nonglobal zone characteristics.

Note that it is possible to have very different software installed in the global zone and nonglobal zones. Unlike Solaris 10, Solaris 11 allows the global zone to be minimized while having “fat” zones or to have minimized zones and a full desktop environment in the global zone.

Agenda

- Oracle Solaris Zones
- **Monitoring Oracle Solaris Zones**
- Controlling zone resources
- Controlling Oracle VM for SPARC resources

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

zonestat Utility

- The `zonestat` utility reports the utilization statistics on currently running zones.
 - CPUs
 - Memory
 - Networking
 - Resource controls
- When run from within a nonglobal zone:
 - Only processor sets visible to that zone are reported
 - All of the memory resources are reported
- The `zonestatted` daemon:
 - Starts when the zone boots up
 - Has no configurable components

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `zonestat` utility reports on the CPU, memory, networking, and resource control utilization of the currently running zones. Each zone's utilization is reported both as a percentage of system resources and the zone's configured limits. The `zonestat` utility prints a series of interval reports at the specified interval. It optionally also prints one or more summary reports at a specified interval.

When run from within a nonglobal zone, only processor sets visible to that zone are reported.

The nonglobal zone output includes all of the memory resources and the limits resource.

The `zonestatted` system daemon is started during system boot. The daemon monitors the utilization of system resources by zones, as well as zone and system configuration information such as `psrset` processor sets, pool processor sets, and resource control settings. There are no configurable components.

zonestat Utility: Examples

- Display a summary of web zone memory utilization every five seconds.

```
root@s11-serv1:~# zonestat -z web -r physical-memory 5
...
Interval: 2, Duration: 0:00:10
PHYSICAL-MEMORY      SYSTEM MEMORY
mem_default          3499M
                        ZONE  USED  %USED  CAP   %CAP
                        [total] 982M 28.0%   -     -
                        [system] 362M 10.3%   -     -
                        web 46.8M 1.33% 50.0M 93.7%
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows memory utilization by the web zone at five-second intervals.

The fields displayed are:

- **SYSTEM-MEMORY:** The total amount of memory available on the physical host
- **ZONE:** The zone using the resource
- **[total]:** The total quantity of resource used systemwide
- **[system]:** The quantity of resource used by the kernel or in a manner not associated with any particular zone
- **USED:** The amount of resource used
- **%USED:** The amount of resource used as a percent of the total resource
- **CAP:** If a zone is configured to have a cap on the given resource, the cap is displayed in this column.
- **%CAP:** The amount of resource used as a percent of zone's configured cap

zonestat Utility: Examples

- Report on the processor sets (psets) once a second for one minute.

```

root@s11-serv1:~# zonestat -r psets 1 1m
...
PROCESSOR_SET      TYPE  ONLINE/CPUS  MIN/MAX
cloud_pset1    pool-pset      1/1      1/1
                ZONE    USED %USED      CAP  %CAP    SHRS  %SHR  %SHRU
                [total] 0.00 0.84%      -    -      -    -    -
                [system] 0.00 0.20%      -    -      -    -    -
                storage 0.00 0.32%    0.30 1.07%      -    -    -
                web    0.00 0.31%    0.25 1.27%      -    -    -
...

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this example, zonestat reports on the processor sets (psets) once a second for one minute.

The fields displayed are:

- ZONE: The zone using the resource
- [total]: The total quantity of resource used systemwide
- [system]: The quantity of resource used by the kernel or in a manner not associated with any particular zone
- USED: The amount of resource used
- %USED: The amount of resource used as a percent of the total resource
- CAP: If a zone is configured to have a cap on the given resource, the cap is displayed in this column.
- %CAP: The amount of resource used as a percent of zone's configured cap
- SHRS: The number of shares allocated to the zone
- %SHRS: The fraction of the total shares allocated to the zone
- %SHRU: Of the share allocated to the zone, the percentage of CPUs used

zonestat Utility: Examples

- Report on the high utilizations.

```

root@s11-serv1:~# zonestat -q -R high 5s 20s
Report: High Usage
  Start: Wed May  2 16:55:54 MDT 2012
  End:   Wed May  2 16:56:14 MDT 2012
  Intervals: 4, Duration: 0:00:20
SUMMARY          Cpus/Online: 2/2   PhysMem: 3499M  VirtMem: 4523M
  ---CPU---  --PhysMem-- --VirtMem-- --PhysNet--
    ZONE  USED %PART  USED %USED  USED %USED  PBYTE %PUSE
  [total] 0.18 9.20%  965M 27.5% 1432M 31.6%   538 0.00%
  [system] 0.04 2.48%  344M 9.84%  937M 20.7%    -  -
    global 0.13 13.1%  476M 13.6%  361M 7.98%   538 0.00%
  database 0.00 0.05%  24.1M 0.69%  21.4M 0.47%    0 0.00%
  storage  0.00 0.06%  39.6M 1.13%  32.3M 0.71%    0 0.00%
    users  0.00 0.10%  34.0M 0.97%  46.4M 1.02%    0 0.00%
    web    0.00 0.07%  45.5M 1.30%  33.5M 0.74%    0 0.00%

```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this example, `zonestat` monitors silently at five-second intervals for 20 seconds and then produces a report on the high utilizations.

The fields displayed are:

- **ZONE:** The zone using the resource
- **[total]:** The total quantity of resource used systemwide
- **[system]:** The quantity of resource used by the kernel or in a manner not associated with any particular zone
- **USED:** The amount of resource used
- **%PART:** The amount of CPU used as a percentage of the total CPU in a processor set to which the zone is bound
- **%USED:** The amount of resource used as a percent of the total resource
- **PBYTES:** The number of transmitted bytes that consumes physical bandwidth
- **%PUSE:** The sum of PRBYTES and POBYTES as a percent of the total available physical bandwidth

Agenda

- Oracle Solaris Zones
- Monitoring Oracle Solaris Zones
- **Controlling zone resources**
- Controlling Oracle VM for SPARC resources

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Zone-Wide Resource Controls

- Zone-wide resource controls limit the total resource usage of all process entities within a zone.
 - Pools as zone-wide resource control
 - FSS as a zone-wide resource control
 - CPU caps as zone-wide resource control
 - Memory caps as zone-wide resource control
- Zone-wide resource control commands:
 - `zonecfg set pool`
 - `zonecfg set scheduling-class`
 - `zonecfg add capped-cpu`
 - `zonecfg add capped-memory`
 - `zonecfg add rctl`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Zone-wide resource controls limit the total resource usage of all process entities within a zone. Any of the resource controls used in the global zone can be used to control resources in the nonglobal zones. The resource controls and attributes used in a zone to control projects, tasks, and processes within that zone are subject to the additional requirements regarding pools and the zone-wide resource controls.

Zone-wide resource controls do not take effect when they are set in the project file. A zone-wide resource control is set through the `zonecfg` utility.

Pools as a Zone-Wide Resource Control

A non-global zone can be associated with one resource pool, although the pool need not be exclusively assigned to a particular zone. Multiple nonglobal zones can share the resources of one pool. Processes in the global zone, however, can be bound by a sufficiently privileged process to any pool. The resource controller `poold` daemon runs only in the global zone, where there is more than one pool for it to operate on. The `poolstat` utility run in a nonglobal zone displays only information about the pool associated with the zone. The `pooladm` command run without arguments in a nonglobal zone displays only information about the pool associated with the zone.

FSS as a Zone-Wide Resource Control

FSS CPU shares for a zone are hierarchical. The shares for the global and nonglobal zones are set by the global administrator through the zone-wide resource control `zone.cpu-shares`. The `project.cpu-shares` resource control can then be defined for each project within that zone to further subdivide the shares set through the zone-wide control. You can use `zone.cpu-shares` to assign FSS shares in the global and the nonglobal zones. If FSS is the default scheduler on your system and shares are not assigned, each zone is given one share by default. If you have one nonglobal zone on your system and you give this zone two shares through `zone.cpu-shares`, that defines the proportion of CPU that the nonglobal zone will receive in relation to the global zone. The ratio of CPU between the two zones is 2:1.

CPU Caps as a Zone-Wide Resource Control

CPU caps provide absolute fine-grained limits on the amount of CPU resources that can be consumed by a project or a zone. CPU caps allow workloads to run on any CPUs while limiting their CPU usage and provide fine-grained (specified in fractions of a CPU) limit. In contrast to FSS, CPU caps provide absolute usage limit that does not depend on other workloads running in the system. CPU caps can be used with CPU binding, processor sets, and FSS. When used in conjunction with processor sets, CPU caps limit CPU usage within a set. When used with FSS, CPU resources defined by caps are further subdivided using FSS shares.

Memory Caps as a Zone-Wide Resource Control

You can control the amount of physical memory, physical locked memory, and swap size when configuring a zone. The `capped-memory` resource sets limits for physical, swap, and locked memory. Each limit is optional, but at least one must be set. To use the `capped-memory` resource, the resource-cap package must be installed in the global zone.

Zone-wide resource controls limit the total resource usage of all process entities within a zone. You configure zone-wide resource controls by using the `zonecfg` command:

- `zonecfg set pool`
- `zonecfg set scheduling-class`
- `zonecfg add capped-cpu`
- `zonecfg add capped-memory`
- `zonecfg add rctl`

rctl Resource Control

- Zone-wide resource controls limit the total resource usage of all process entities within a zone.
- The `zonecfg add rctl` command.

```
root@s11-serv2:~# zonecfg -z workzone
zonecfg:workzone> add rctl
zonecfg:workzone:rctl> set name=zone.cpu-shares
zonecfg:workzone:rctl> add value (priv=privileged,limit=5,action=deny)
zonecfg:workzone:rctl> end
...
```

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Zone-wide resource controls limit the total resource usage of all process entities within a zone. You configure zone-wide resource controls by using the `zonecfg add rctl` command. The `rctl` properties are `name` and `value`. The available name properties are shown in the following slide. The value property consists of privilege, threshold, and action. For example:

```
root@s11-serv2:~# zonecfg -z workzone
zonecfg:workzone> add rctl
zonecfg:workzone:rctl> set name=zone.cpu-shares
zonecfg:workzone:rctl> add value
(priv=privileged,limit=5,action=deny)
zonecfg:workzone:rctl> end
```

rctl Resource Properties

Property	Description
zone.cpu.caps	Maximum number of CPUs
zone.cpu-shares	Number of FSS CPU shares for this zone
zone.cpu-lofi	Maximum number of lofi devices
zone.max-lwps	Maximum number of LWPs
zone.max-msg-ids	Maximum number of message queue IDs
zone.max-processes	Maximum number of simultaneous processes
zone.max-sem-ids	Maximum number of semaphore IDs
zone.max.shm.memory	Total amount of System V shared memory
zone.max-swap	Total amount of swap that can be consumed by user processes and tmpfs mounts
zone.max-locked-memory	Total amount of physical locked

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This table lists the available zone-wide resource properties supported by the `rctl` resource.

Zone-Wide Resource Control: Examples

- Assign a resource pool to a zone.

```
root@s11-serv1:~# zonecfg -z oracle_11g
zonecfg:oracle_11g> set pool=ora_pool
zonecfg:oracle_11g> verify
zonecfg:oracle_11g> commit
zonecfg:oracle_11g> exit
```

- Make FSS the default scheduler for a zone.

```
root@s11-serv1:~# zonecfg -z oracle_11g
zonecfg:oracle_11g> set scheduling-class=FSS
zonecfg:oracle_11g> set cpu-shares=50
zonecfg:oracle_11g> verify
zonecfg:oracle_11g> commit
zonecfg:oracle_11g> exit
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows adding resource controls to a zone. In the first example, the `ora_pool` is assigned to the `oracle_11g` zone. In the second example, the default scheduling class for the `oracle_11g` zone is set to FSS. The `cpu-shares` for this zone is 50.

Zone-Wide Resource Control: Examples

- Cap the number of CPUs used by a zone.

```
root@s11-serv1:~# zonecfg -z oracle_11g
zonecfg:oracle_11g> add capped-cpu
zonecfg:oracle_11g:capped-cpu> set ncpu=4
zonecfg:oracle_11g:capped-cpu> end
zonecfg:oracle_11g> verify
zonecfg:oracle_11g> commit
zonecfg:oracle_11g> exit
```

- Dedicate CPUs to a zone.

```
root@s11-serv1:~# zonecfg -z oracle_11g
zonecfg:oracle_11g> add dedicated-cpu
zonecfg:oracle_11g: dedicated-cpu> set ncpu=2
zonecfg:oracle_11g: dedicated-cpu > end
zonecfg:oracle_11g> verify
zonecfg:oracle_11g> commit
zonecfg:oracle_11g> exit
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This slide shows adding resource controls to a zone. In the first example, the `oracle_11g` zone is capped (limited) to four CPUs. In the second example, two CPUs are dedicated to the `oracle_11g` zone.

Zone-Wide Resource Control: Examples

- Add a physical memory cap for a zone.

```
root@sl1-serv1:~# zonecfg -z oracle_11g
zonecfg:oracle_11g> add capped-memory
zonecfg:oracle_11g:capped-memory> set physical=4g
zonecfg:oracle_11g:capped-memory> set swap=8g
zonecfg:oracle_11g:capped-memory> end
zonecfg:oracle_11g> verify
zonecfg:oracle_11g> commit
zonecfg:oracle_11g> exit
```

- Add cap for light-weight processes in a zone.

```
root@sl1-serv1:~# zonecfg -z oracle_11g
zonecfg:oracle_11g> add rctl
zonecfg:oracle_11g:rctl> set name=zone.max-lwps
zonecfg:oracle_11g:rctl> add value (priv=privileged,limit=200,action=deny)
zonecfg:oracle_11g:rctl> end
zonecfg:oracle_11g> verify
zonecfg:oracle_11g> commit
zonecfg:oracle_11g> exit
```

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Many experienced Solaris administrators have the idea that they "don't want to swap" and as such they set a physical memory cap and don't bother with swap. This causes exactly the opposite of the desired behavior. Administrators that are trying to avoid paging to swap devices are rightly doing so to avoid a huge performance hit that affects the entire system. The use of a physical memory cap causes `rcapd` to monitor the RSS of processes and when it detects that too much is used, it forces memory to be paged out. That is, a physical memory cap does not prevent allocation, it causes `rcapd` to induce paging. To strictly limit the amount of memory that a zone can use, use the swap cap. Note that the term swap means multiple things. In this context, it is the amount of virtual memory that a zone can use. If the zone tries to allocate more memory than the swap limit, the allocation will fail. Much like running a global zone without swap devices configured, this can have undesirable effects, particularly when processes are consuming a lot of memory.

This slide shows capping the zone's physical memory to 4 GB and the zone's swap space to 8 GB. In the second example, the resource control `zone.max-lwps` is added to the `oracle_11g` zone. The control is privileged (only `root` can change it), the light-weight processes are capped at 200, and any attempt to exceed the cap is denied.

Agenda

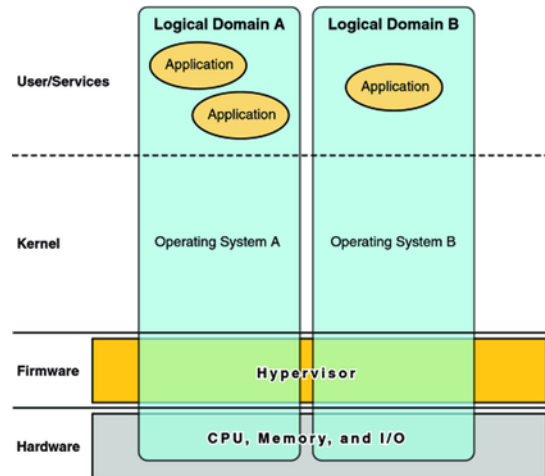
- Oracle Solaris Zones
- Monitoring Oracle Solaris Zones
- Controlling zone resources
- Controlling Oracle VM for SPARC resources

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle VM for SPARC

- Consolidation of up to 128 virtual machines on one SPARC T-Series server.
- Live migration between hosts.
- Fully dynamic resource management of CPU, memory, virtual I/O, and crypto accelerators.
- Automatic CPU dynamic resource management places resources where they are needed the most.
- Redundant virtual networks and disks for higher availability.



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

CPU Whole Cores and CPU Cap

- By default, Oracle VM Server for SPARC domains are configured with CPU threads (strands).
- Domains CPUs can be partitioned and capped by using whole CPU cores.
- CPU whole core configuration:
 - `ldm set-core number-of-cpu-cores domain`
 - # `ldm set-core 2 primary`
 - `ldm set-domain max-cores=max-number-of-cpu-cores domain`
 - # `ldm set-domain max-cores=2 primary`

The Oracle logo, consisting of the word "ORACLE" in white capital letters on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The Oracle VM Server for SPARC software runs on Oracle's SPARC T-Series servers, which use SPARC T-Series processors. The SPARC T-Series processors have multiple CPU cores, and each CPU core has multiple CPU threads called strands. By default, domains that are created with the Oracle VM Server for SPARC software are configured with CPU threads.

Beginning with the Oracle VM Server for SPARC 2.0 release, hard partitioning is enforced by using CPU whole-core configurations. In such a case, domains are configured with CPU whole cores, instead of the default of individual CPU threads (virtual CPUs). When binding such a domain, the system provisions the specified number of CPU cores and all its CPU threads to the domain. Using a whole-core configuration and setting a CPU cap also limits the number of CPU cores that can be dynamically assigned to a bound or active domain, and live migration is not permitted under the terms of the hard partitioning license.

The commands in this example configure a domain to use two CPU whole cores and set the maximum number of two CPU cores (CPU cap) for the domain.

Viewing CPU Whole Cores Configurations

```
# ldm list ldom1
NAME      STATE   FLAGS   CONS  VCPU  MEMORY UTIL  UPTIME
ldom1     active  -n----  5000  16    2G      0.4%   5d 17h 49m
# ldm list -o resmgt ldom1
NAME
ldom1
CONSTRAINT
  whole-core
  max-cores=2
# ldm list -o core ldom1
NAME
ldom1
CORE
CID  PCPUSET
1   (8, 9, 10, 11, 12, 13, 14, 15)
2   (16, 17, 18, 19, 20, 21, 22, 23)
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `ldm list` command shows 16 virtual CPUs assigned to domain `ldom1`. The `ldm list -o resmgt` command shows a CPU cap of two cores. The `ldm list -o core` command shows cores 1 and 2 assigned to domain `ldom1`.

CPU Threading Modes and Workloads

- Dynamic CPU threading controls to optimize workload performance on SPARC T4 systems.
- Two CPU threading modes:
 - Maximizing for throughput (max-throughput)
 - Maximizing for IPC (max-ipc)
- Configure CPU threading mode of domain:
 - `ldm add-domain[threading=max-throughput|max-ipc] ldom`
 - `ldm set-domain [threading=max-throughput|max-ipc] ldom`

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

On SPARC T4 systems, you can optimize CPU performance by specifying the CPU threading mode. The threading mode can be set dynamically and independently for each domain on the system. A reboot is not required to change the threading mode, and the set mode is maintained across domain reboots or platform power cycles.

By selecting the appropriate CPU threading mode, you can improve the performance of applications and workloads that are running on a domain. You can select a threading mode that either maximizes throughput or maximizes the number of instructions per cycle:

- **Maximizing for throughput (max-throughput):** Workloads that benefit most from high throughput run a lot of software and perform a lot of I/O operations. When you optimize for maximum throughput, you enable CPU cores to concurrently run a maximum number of hardware threads. This mode is best for running heavily threaded workloads, such as those performed by web servers, database servers, and file servers. This mode is used by default and is also used on older SPARC T-series platforms, such as SPARC T3 platforms.
- **Maximizing for IPC (max-ipc):** Workloads that benefit most from high IPC are CPU intensive, such as systems that run intensive arithmetic computations. When you optimize for maximum IPC, you enable a CPU thread to execute more instructions per CPU cycle. This optimization is achieved by reducing the number of CPU threads that are concurrently active on the same CPU core.

Viewing CPU Threading Modes

```
# ldm list -o resmgt ldom1
```

```
NAME
```

```
ldom1
```

```
CONSTRAINT
```

```
whole-core
```

```
max-cores=3
```

```
threading=max-ipc
```

```
# ldm list -o cpu ldom1
```

```
NAME
```

```
ldom1
```

```
VCPU
```

```
VID PID CID UTIL STRAND
```

```
0 8 1 0.3% 100%
```

```
1 9 1 0% 100%
```

```
2 10 1 0% 100%
```

```
3 11 1 0% 100%
```

```
4 12 1 0% 100%
```

```
...
```

```
8 24 1 0.4% 100%
```



Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The `ldm list -o resmgt` command shows the constraints. The `ldm list -o cpu` command shows the deactivated virtual CPUs by specifying a value of 0 in the `UTIL` column. The bold text in this max-ipc example shows that only one thread is activated per CPU.

Agenda

- Oracle Solaris Zones
- Monitoring Oracle Solaris Zones
- Controlling zone resources
- Controlling Oracle VM for SPARC resources

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Quiz

Which `rctl` resource property is used to set the number of FSS CPU shares for a zone?

- a. `zone.fss-shares`
- b. `zone.cpu-shares`
- c. `zone.cpu.cap`
- d. `zone.fss.cap`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: b

Quiz

In the Oracle Solaris 11 system, you have two facilities that can be used to separate and identify workloads. These facilities are:

- a. Processes and tasks
- b. Profiles and tasks
- c. Projects and profiles
- d. Projects and tasks

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: d

Quiz

Which project database property identifies the users associated with a project?

- a. users
- b. project-users
- c. user-list
- d. attributes

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: c

Quiz

Which command is used to list projects from the project database?

- a. projects
- b. projstat
- c. projlist
- d. project-list

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Quiz

Resource management constraint mechanisms allow the administrator or application developer to set bounds on the consumption of specific resources for a workload.

- a. True
- b. False

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Answer: a

Summary

In this lesson, you should have learned how to:

- Describe Oracle Solaris Zones
- Monitor zone resource consumption
- Control zone resources
- Describe VM for SPARC
- Control Oracle VM for SPARC resources

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Practice 13: Overview

This practice covers managing zone-wide resources and controls.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

14

Performance Analysis and Testing

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Objectives

After completing this lesson, you should be able to do the following:

- Describe steps for maintaining system performance
- Describe common utilities for measuring system performance and setting tunable parameters
- Describe specific types of bottlenecks and methods for reducing them
- List the basic steps for the performance analysis approach
- Use the Performance Analysis Checklist to identify problems
- Plan to conduct performance testing
- Describe common pitfalls

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Relevance

Discussion: The question “How can I compare the performance of different product offerings?” is relevant to understanding the Oracle Sun Storage System.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Performance analysis
- Types of performance testing and benchmarks
- Performance testing tools

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Introduction

- Performance testing is the study of performance by applying specific workloads and taking measurements.
- It is conducted by:
 - Engineers, during development of products to identify performance issues
 - Customers, to evaluate products for production environments, and to provide data for capacity planning
- Performance testing can be deceptively complex.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Maintaining System Performance

- Monitoring the workload
- Collecting long-term usage information
- Documenting system changes (software patches, hardware upgrades)
- Analyzing the data for trends
- Isolating performance bottlenecks
- When necessary, adjusting system parameters to match the demands of the workload

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Performance review should start with comparing measurements between current and previous usages, the baseline. This baseline, along with any notes on changes in workload or system configuration, helps to identify impact on performance in the current system.

Performance Analysis Approach

1. Errors
2. Mis-configs
 - Checking software/firmware versions so that you do not experience known bugs—Oracle Explorer
3. Load
 - Knowing what is happening
 - Eliminate unnecessary work—for example, 10x win
4. Eliminate Bottlenecks
 - Upgrading saturated resource—for example, 100 percent improvement
5. Fine-tuning
 - *Solaris Tunable Parameters Reference Manual*
 - Apache `httpd.conf`
 - MySQL `my.cnf` and so on
 - Tune one thing at a time.
6. Unknown bugs

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, centered within a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

In this lesson, we review the list of steps that you can take to respond to a performance problem. As we examine each step, we will review the various tools we have covered in the course that will be of assistance.

We will also cover some of the resource management tools that enable you to isolate various workloads from each other, and bound their resource consumption.

Errors

- Check at system level
 - syslog, syslog-ng, /var/adm/messages
 - iostat -En
 - DTrace Toolkit
 - errinfo
 - opensnoop -e
 - execsnoop
- Check key applications
 - truss(1)
 - DTT
 - DTrace

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Errors can account for tremendous resource consumption in terms of hardware and system overhead in dealing with faults and conditions. Eliminate the obvious in the messages files, network status tools, and I/O status tools. Then, focus on which applications are causing the activity by using the DTrace Toolkit applications that can aggregate system activity and identify the executables (by execname) that are causing the error activity. Then, use the per-process tools in the DTrace Toolkit (DTT) and truss to look at those applications. If necessary, use customized DTrace scripts to collect more information.

Misconfigurations

Check software and firmware versions so that you do not experience known bugs.

- Oracle Explorer—to gather current information
- Review all software patch README files and release notes.
 - Latest versions contain bug fixes.
 - Latest versions contain performance enhancements.
- Document the hardware limits of the system.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle customer support can help analyze Explorer output to identify misconfigurations, known bugs, and patched revisions of supported products.

Eliminate Bottlenecks

- Memory
- CPU
- I/O
 - Disk
 - Network
 - Other devices
- Locks
- Balance the load

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered on a red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

When eliminating bottlenecks, start with memory. If your system is short on memory, that will cause an increase in disk I/O, possibly network I/O, and even CPU consumption. Remember to check whether memory pressure is causing the ZFS ARC to do reclaims on its buffer cache. Also, remember to check the memory bus by measuring the cycles per instruction by using `cpustat`.

Check the CPU for lack of idle time and saturation (run queue size). Check the distribution of load between system and user time and compare to your baseline. If system time has gone up, find out why. It should be system calls, exception or error handling, interrupts, or kernel lock contention.

For Disk I/O, use `iostat`, `iosnoop`, the `fsinfo` provider in DTrace, `sar`, or vendor-supplied tools for third-party file systems and storage. For network activity, use `kstat`, `netstat`, and DTrace tools.

High system time may mean lock contention in the kernel. Check `smtx` and `srw` values in `mpstat`. Use `lockstat` and DTrace to isolate the locks and code causing the problem.

Use the provided checklist to make sure you covered everything.

Fine-Tuning

- *Oracle Solaris Tunable Parameters Reference Manual*
<http://download.oracle.com/docs/cd/E19253-01/817-0404/index.html>
- Use custom commands
- Configuration files
- `/etc/system`

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Consult the *Oracle Solaris Tunable Parameters Reference Manual*, especially under the “Oracle Solaris Kernel Tunable Parameters” section. Check the description, range of values, and the default value. Always check `/etc/system` and `mdb -k` to see whether the default value is being used. If the tunable parameter is not set to the default value, try using the default value. Check the “When to Change” section of the *Oracle Solaris Tunable Parameters Reference Manual* for any given tunable parameter for guidelines.

If you do change a tunable parameter, use the `/etc/system` file, or preferably the custom configuration tool, or the configuration file intended for that parameter. Do not change the live system with `mdb` unless that is the only way documented to make the change. In that case, proceed with extreme caution.

The system might crash if you use the `mdb` utility to change the values of parameters. An improper setting could cause the kernel to panic.

Document your changes. Otherwise, old tuning changes will inevitably show up in later versions of the OS where the change is not necessary and may degrade performance.

Viewing the Values of Tuning Parameter

Function	Description
Boot	<code>/etc/system</code> : Configuration file for customizing various parameters in the kernel. This file is read-only at boot time.
Network	<code>ndd</code> : Used for temporarily setting exposed parameters in drivers such as TCP/IP settings
Configuration Files	<code>/etc/default</code> : Contains configuration files for many Solaris services. There are also individual configuration files for drivers in <code>/kernel/drv</code> , <code>/usr/kernel/drv</code> , and under <code>/platform</code> .
Custom Commands	Family of task-based control commands including: <code>dladm(1M)</code> , <code>prctl(1)</code> , <code>routeadm(1M)</code> , <code>pooladm(1M)</code> , <code>ifconfig(1M)</code> , <code>tunefs(1M)</code>
<code>mdb</code>	Enables you to view, and in rare cases, modify kernel parameters while the system is running
Dynamic	DTrace allows you to view, but not change, parameters in the kernel by using the <code>backquote</code> scoping operator.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Setting Tuning Parameters

Changing tuning parameters in the `/etc/system` file:

- To set the value of a tuning parameter for the kernel, you use the following syntax:
 - `set variable_name=value`
- To set the value of a tuning parameter for a kernel module, you use the following syntax:
 - `set [modulename:] variable_name=value`
- Restart the system after modifying the `/etc/system` file for the changes to take effect.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Make a copy of the existing `/etc/system` file before modifying it.

The `/etc/system` file contains the values of tuning parameters. You modify these values to set tuning parameters for the kernel and kernel modules.

- To set the value of a tuning parameter for the kernel, you use the following syntax:
`set variable_name=value`
- To set the value of a tuning parameter for a kernel module, you use the following syntax:
`set [modulename:] variable_name=value`

For example, the `set ncsiz=10000` directive overwrites the default `ncsiz` value. Improper settings in the `/etc/system` file can cause the kernel to crash, repeatedly.

Recovering /etc/system File Settings

- Make a copy of the existing `/etc/system` file before modifying it.
- To start the system with the previous version:
 1. `ok boot -a`: Prompts for files that it uses during startup
 2. Provide the name of the previous version of the `/etc/system` file. For example:

```
Name of system file [etc/system]:  
/etc/system.oldfile
```

where `system.oldfile` is the name of the file in which you saved the earlier `/etc/system` file.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

If there are no important settings you need in the old version of the `/etc/system` file, you can use the `/dev/null` file for this step.

Caution: Make a copy of the existing `/etc/system` file before modifying it. If there are important settings in the `/etc/system` file and you use the `/dev/null` file to replace it, you could do permanent damage.

Unknown Bugs

- When all else fails, you may be dealing with an unknown and therefore undocumented bug.
- To eliminate what is not causing the problem, a checklist may be useful.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

The rest of this lesson consists of an example checklist. A copy is provided in the lab directory.

Agenda

- Performance analysis
- Types of performance testing and benchmarks
- Performance testing tools

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Types of Performance Testing

- Real-World Testing
 - The best test of performance is of the system in production, with the intended workload.
- Simulated Load
 - Next to real-world testing, the most effective test of performance is with a workload that simulates the intended environment as closely as possible.
 - HP LoadRunner
 - Sun FileBench

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

- **Microbenchmarks**
 - Open-source packages to test component performance using simple workloads are often called microbenchmarks.
 - Microbenchmarks picked to test those characteristics provide a reasonable estimation of performance.
 - The `dd (1M)` utility may be used as a microbenchmark for single-threaded sequential I/O.
- It is important to understand which benchmark is applicable to your target workload.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Industry Benchmarks

- Industry-standard benchmarks are well documented and results are often published. Two popular groups of industry benchmarks are:
 - Standard Performance Evaluation Corporation (SPEC)
<http://www.spec.org>
 - Transaction Processing Performance Council (TPC)
<http://www.tpc.org>
- It is important to:
 - Understand exactly what the benchmark measures
 - Understand how it relates to your target workload

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Agenda

- Performance analysis
- Types of performance testing and benchmarks
- Performance testing tools

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Performance-Testing Tools

- Numerous tools exist:
 - Research the tools available in the area of interest
 - Double-check their results from independent software
- For example, run `iostat` when using disk benchmarking tools to check that actual disk performance.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Workload Assessment

- Identify the applications that use the most resources
 - `prstat` and `ps`
 - DTrace Toolkit
- Identify key applications, isolate and distribute the load
 - Resource management
 - Virtualization
- View overall activity from the system level and drill down
 - The `truss` utility
 - `pmap -x`, `pstack`, `ptree`, and other `proc` tools
 - The `dtrace` utility

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is centered on a solid red rectangular background.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Any time you can identify unnecessary load, and eliminate it, can be a huge performance enhancement.

In this section, the tools that can be used to isolate, distribute, and confine the various workloads on a given server are discussed.

Performance-Testing Tools: File Systems

Tool	Description
FileBench	File system workload simulator, http://www.solarisinternals.com/wiki/index.php/FileBench
vdbench	File system or disk benchmark, http://sourceforge.net/projects/vdbench
iozone	File system benchmark, http://www.iozone.org
iometer	File system benchmark, http://www.iometer.org
bonnie++	File system benchmark, http://sourceforge.net/projects/bonnie
dd	dd (1M) , possible single-threaded streaming load generator

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

These tools usually test application I/O to a local or remote file system. Some can be applied directly to disk devices.

Performance-Testing Tools: Network

Tool	Description
uperf	Network workload simulator, http://www.uperf.org
netperf	Network benchmark, http://www.netperf.org
iperf	Network benchmark, http://sourceforge.net/projects/iperf
ttcp	Test TCP, simple network benchmark, available from multiple locations and available in Java
ping(1M)	<code>ping -s</code> : A simple network latency test

The Oracle logo, consisting of the word "ORACLE" in a bold, sans-serif font, is positioned on the right side of a red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Network benchmark tools are also helpful for troubleshooting network environment bottlenecks.

Performance-Testing Tools: CPU

Tool	Description
whetstone	Synthetic cpu benchmark (floating point), http://www.netlib.org/benchmark
dhystone	Synthetic cpu benchmark (integer), http://www.netlib.org/benchmark
iperf	cpu benchmark (floating point), http://www.netlib.org/benchmark
SPEC	See the SPECint results on http://www.spec.org .

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

CPU benchmarking can be performed by tools such as the ones shown in the slide.

There are various benchmarks that exist for testing CPU performance. Some of these focus on running small kernels. Small kernels have the problem that they may not be representative of actual performance of a user's application. Other benchmarks such as the SPEC CPU suites use real applications and workloads to provide performance metrics that may be more representative of those experienced by users.

Performance-Testing Tools: Memory

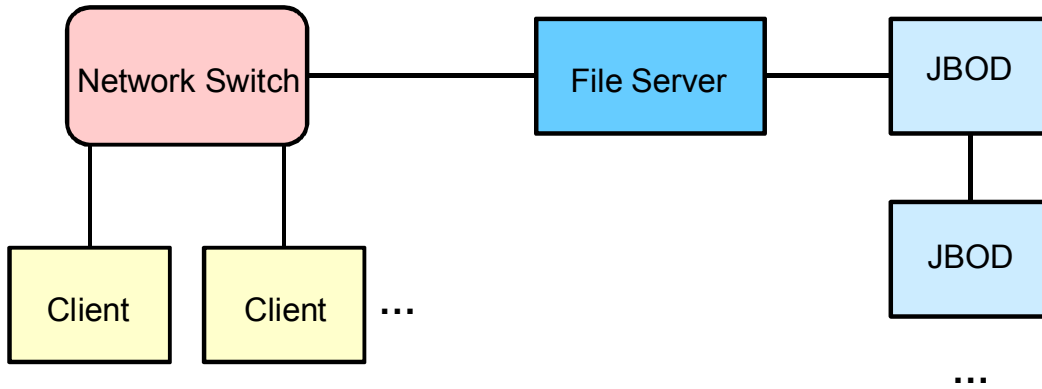
Tool	Description
stream	Memory throughput benchmark, http://www.streambench.org

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Memory bus throughput can be an important factor for many workloads.

Functional Diagram



Example functional diagram of a file server system

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Consider a block diagram of the system to test all components in the data path: clients, routers, switches, and so on. Test the performance of as many of these components as possible.

Functional Diagram

For each component in the block diagram, check for:

- Utilization
- Saturation
- Errors

The Oracle logo, consisting of the word "ORACLE" in a white, sans-serif font, is positioned on the right side of a solid red horizontal bar.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Even if that information is not available, it is a useful exercise to understand what is known and what is not known.

Tests of a Sun Storage 7410 system, considering the entire data path and examining each component, found that there were no bottlenecks, except for the network switch, which did not provide statistics for utilization. The switch was replaced with a more powerful model, and the performance improved by 30 percent.

Understand Benchmark Software

- What the software does.
- What its default settings are:
 - Defaults have been historically significant, but need to be reconsidered.
 - `iozone`, for instance, defaults to 512 MB or less file size, which modern systems can cache in memory.
- Look out for:
 - File size used
 - Number of threads

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Sanity Test

- Find ways to sanity-test the numbers.
 - A gigabit network interface has a theoretical maximum of about 120 MB/sec (converting 1 GbE to bytes).
 - Invalid results such as 300 MB/sec and faster can be rejected.
- IOPS can be checked in a similar way: 20,000 x 8 KB read ops/sec would require about 156 MB/sec of network throughput, plus protocol headers—too much for a 1-GbE link.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Double-Check

- Use an alternate method to confirm your results.
- If testing network throughput, validate results in the data path:
 - Switches
 - Routers
 - The origin and destination
- A result can appear sane but still be wrong. Check whether the numbers add up, end to end.
- An excellent tool for double-checking results is DTrace.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Drive Resources to Saturation

- To find peak performance, apply load until the target has reached saturation. This resource may be:
 - CPUs
 - System buses
 - Available disks
 - Network interfaces
- Once that resource is 100 percent used (saturated), the system is likely to be running at its peak.
- In many systems, the limiter is measured as CPU utilization, because the system can generally go no faster.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

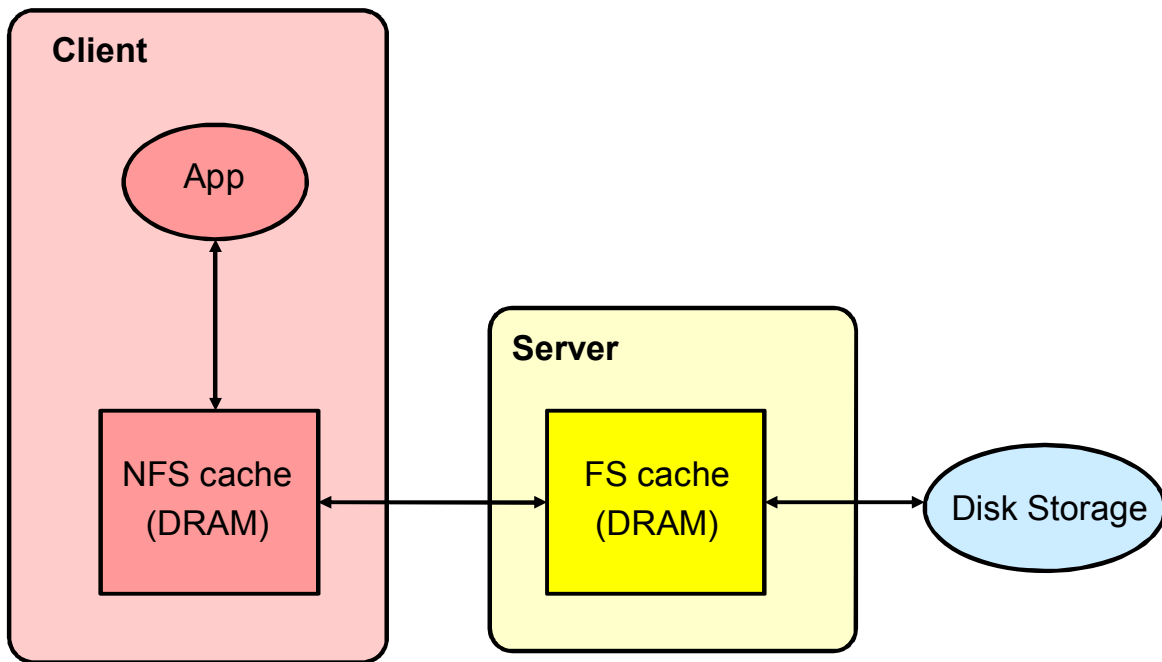
Document Your Test System

- Performance results should be accompanied by details on the:
 - Target system
 - Clients
 - Network devices
- Describe how the tested product compared to the maximum configuration is available.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Testing File Servers



ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

File servers are tricky to test. They involve applying load from clients, which may become the bottleneck.

Beware of Client Caching. Many file access protocols cache data on the client, which is performance tested instead of the file server.

Ways to Avoid Client Caching

Per-Client File Size Used	Client DRAM	Server DRAM	Result
1 GB	2 GB	128 GB	Client caching (server not tested)
10 GB	2 GB	128 GB	Server caching
1 TB	2 GB	128 GB	Server disk access

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Ways to avoid client caching include:

- Use a file size (working set) that is twice the client's DRAM size.
- Set the `physmem` tunable parameter to limit client memory.
- Remount between runs.
- Mount with the `forcedirectio` option for read tests.
- Use multiple clients (at least 10).

Distribute Client Load

- Modern file servers, such as the Oracle Sun Storage 7410, can saturate a 10-GbE interface, but it is difficult for a single client to do the same.
- A file server's optimized kernel can respond to requests quicker than client-side software can generate them.
- It takes twice the CPU to drive load as it does to accept it.
- Network bandwidth can also be a bottleneck: It takes at least 10 clients with 1-GbE interfaces to max out a 10-GbE server interface.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Disk Matter

- File servers make use of many GB of DRAM cache to improve performance. Some use Solid State Disks (SSDs) as a secondary cache such as the L2ARC in the Oracle Sun Storage 7000 series.
- While these caches can greatly improve performance, data must eventually be written to (and read from) disk.
- Check disk utilization.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Check Your Storage Profile

- Evaluate the desired redundancy profile against mirroring.
- Double-parity RAID optimizes price/GB with lower performance than mirroring, particularly random reads. Compare price/performance and price/GB to understand the trade-off made.

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Plan to conduct performance testing
- Describe common pitfalls

ORACLE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

There is no practice for this lesson.

