Solaris 10
New Features

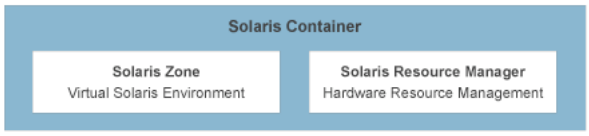
Andreas Polze

Virtualization Technology

- Solaris Containers
  - The software partitioning function provides multiple independent fine-grain OS environments in the server.
  - Up to 8192 partitions even without hardware or firmware based partitioning features.
- Solaris Container provides security and problem isolation between Containers.
  - This helps you to safely consolidate servers and more efficiently utilize system resources including CPUs, to provide improved cost reductions throughout the life of your system.

AP 06/08

Solaris Containers



- Software partitioning technology
  - partitions one OS space into multiple OS partitions.
  - Consisting of Solaris Zones and Solaris Resource Manager.
- Solaris Zones is software technology
  - Provides up to 8,000+ entities per Solaris OS instanc
- Solaris Resource Manager
  - enables flexible distribution of hardware resources (CPU, memory) across the virtual Solaris environments.
  - prioritizes the allocation of resources to operations.
  - even single CPU servers can be partitioned into multiple virtual Solaris environments.

AP 06/08

Container Characteristics

- Fully separated virtual Solaris environments
  - You are able to construct secure systems due to the access prohibitions between the various virtual Solaris environments.
  - Each virtual Solaris environment is fully separate and one environment can't access the processes of any other.
- Flexible hardware resources allotment
  - Server resources utilization can be changed according to workload.
  - Hardware resources such as CPUs and memory can be allotted flexibly to each virtualized Solaris environment and changed as your resource utilization requirements change.
  - For example, your resource allotment policy can be switched such that resources are focused on online operations during the day and on batch operations overnight.
- Speedy construction of Solaris environments
  - The flexibility of Solaris containers, means you also don't miss business opportunities due to restrictions on fast system construction.
  - Time-consuming procedures of new system acquisition can be removed.
  - You just use Solaris Containers, by changing system parameters, to quickly prepare the new Solaris environment.

AP 06/08

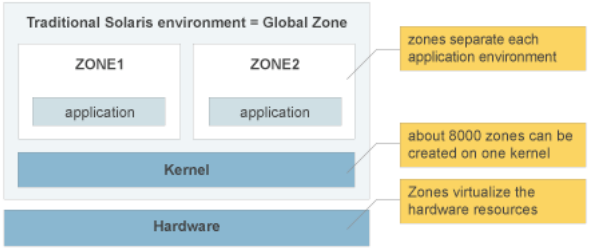
Solaris Zones

- A Solaris Zone is a partitioned virtual OS environment working in a Solaris OS space.
- There are two types of Solaris Zones:
  - Global Zones and Non-Global Zones.
- Global Zone is the traditional OS environment and is where Solaris OS is installed.
  - All system operations such as installations, startups and shut-downs are done in the Global Zone.
  - However the 8000+ Non-Global Zone entities work as virtual Solaris OS environments within the Global Zone.
- Only Disk and Network interfaces defined in the Global Zone can be used in the Non-Global Zones.
  - Definition can only be done in the Global Zone.
  - Non-Global Zones are more simply referred to as Zones.

AP 06/08

Zone construction

- Most OS parameters succeed from Solaris OS to each Zone.
  - Network configurations such as Host names and IP addresses must be configured in each Zone.
  - Plus, to ensure the system security at Zone level, root privileges are placed in each Zone so that each Zone can be managed as an independent system.
  - System startup and shut-down control is performed in the Global-Zone by the zoneadm(1M) command.

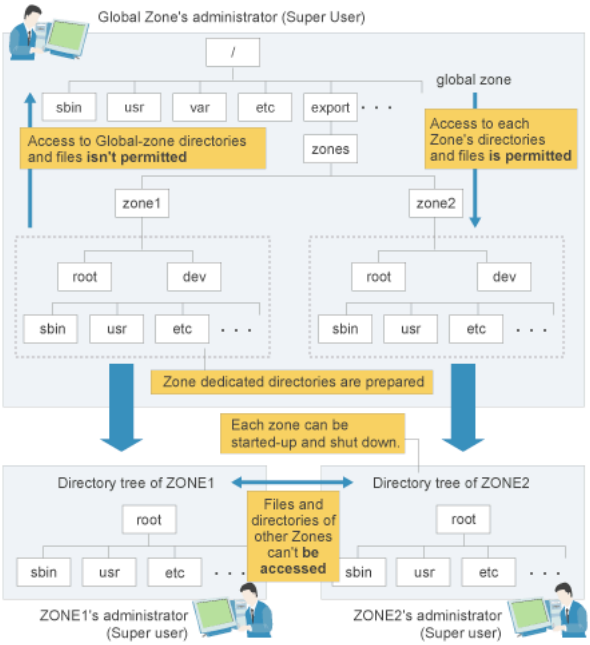


AP 06/08

File system mechanism

- Each Zone has a dedicated directory under the Global Zone's file system.
- Zone installation then constructs a dedicated directory.
  - In the global Zone, access to all files or directories under each dedicated Zone directory is permitted.
  - But in each Zone, files or directories not included in the dedicated Zone directory can't be accessed.
  - Directories in Global Zone can be mounted to a Zone's dedicated directory .
  - You can also choose the access mode to the mounted directories, enabling either read-only or read-write

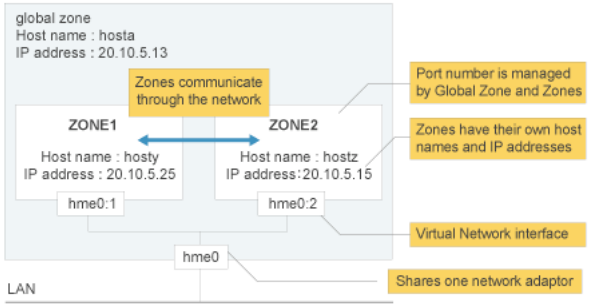
AP 06/08



AP 06/08

Network mechanism

- Each Zone has a virtual network interface for communication with other systems on the network.
  - Zones in the same server can also communicate through the network.
- Global Zones and Zones have their own specific host names and IP addresses.
  - Port numbers are also managed independently by Global Zone and the other Zones.



'08

Solaris Resource Manager

- Maximizes your hardware utilization without planned down time, by scheduling hardware resource allotment to each Zone.
- Solaris Resource Manager enables you to schedule the hardware resource allotment to each Zone.
  - Now you can maximize your hardware utilization without having to schedule any planned down time.
- When you want to run multiple applications, for instance ERP and CRM in the same system, you can create separate Zones for each application unit.
  - Solaris Resource Manager allots the hardware resources to each Zone so that peak workloads in one Zone has no effect on the Zone environments.
- Moreover, resource allotment policy can be re-configured dynamically.
  - This lets you change resource allotments according to resource workloads without shutting down applications or requiring reboot operations.

AP 06/08

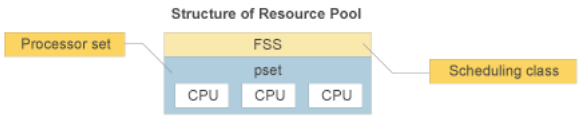
Resource Pools

- A Resource pool is a CPU allotment scheduling function that operates between specific Zones.
  - Each Zone is assigned to just one Resource Pool and that one Resource Pool can shared by the multiple Zones.
  - These Zones share a defined number of CPUs(processor set) under a defined CPU scheduling policy(Scheduling class).
- processor set(pset)
  - A processor set is assigned a specified number of CPUs.
- Scheduling class (FSS and TS)
  - The Resource Pool allots CPU resources to processes working in a Zone by reference to the Resource Pool definition according to a specified scheduling class.
  - Typical scheduling classes are TS and FSS.

AP 06/08

Scheduler

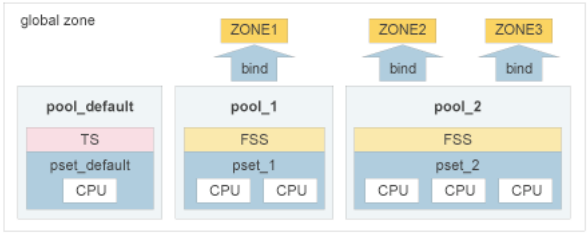
- FSS (Fair Share Scheduler) allots CPUs to processes according to the CPU utilization ratios of Zones referenced from the Resource Pool definition.
  - For instance, if the CPU utilization ratios of Zone A and B are defined as 4:1, Zone A can use 80% and Zone B can use 20% of the usable CPU resources
- TS(Time Share Scheduler) is the default scheduler with Solaris OS.
  - It fairly allots CPU resources to every processes, and does not concentrate CPU resources to any one specific process.



AP 06/08

Zone and Resource Pool

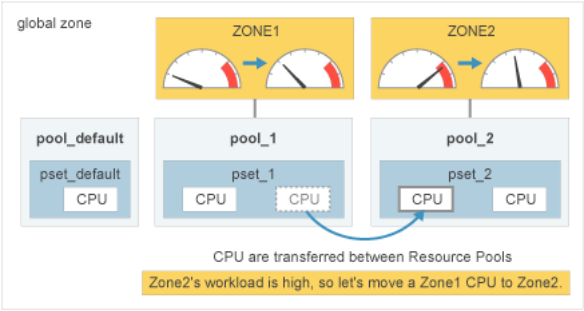
- A Zone is assigned (bound) one Resource Pool for system start.
  - One Resource Pool can be bound to more than one Zone, however each Zone can only be bound to one Resource pool .
  - If FSS is set in the scheduling class parameter, CPUs are scheduled based on the CPU resource ratio set in the scheduling class parameter.
- Resource Pool has the default definition (pool_default).
  - Global Zone is bound to the pool_default.
  - When a Resource Pool is not specified at Zone start-up, the Zone is automatically bound to the pool default.



AP 06/08

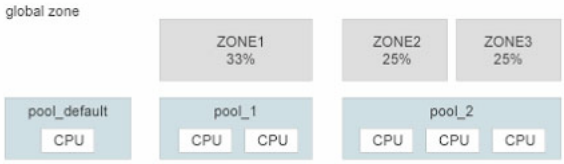
Dynamic Resource Pool

- CPUs can be transferred between different Resource Pools based on CPU utilization.
  - CPU allotments can also be changed.
  - Such CPU re-allotment is done either by command line or a threshold definition



AP 06/08

Solaris Container Demonstration

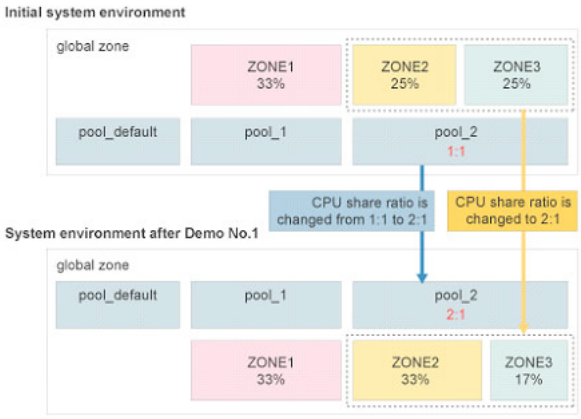


- Initial environment
  - The global zone contains three non-global zones (ZONE1, ZONE2, ZONE3)
  - One CPU is allotted to a resource pool named pool_default, two CPUs to pool_1 and three CPUs to pool_2
  - A zone named ZONE1 is allotted to pool_1, with ZONE2 and ZONE3 allotted to pool_2
  - CPU share ratio between ZONE2 and ZONE3 is 1:1
  - A workload program is running in each zone
- CPU share ratio in the initial environment
  - CPU share ratio of all six CPUs totals 100%
  - CPU share ratio of ZONE1 is around 33% equal to one third of the total six CPUs
  - CPU share ratio of ZONE2 is 25% because CPUs allotted to pool_2 are equally shared with ZONE2 and ZONE3
  - CPU share ratio of ZONE3 is 25%; the same ratio as ZONE2

AP 06/08

Demonstration 1

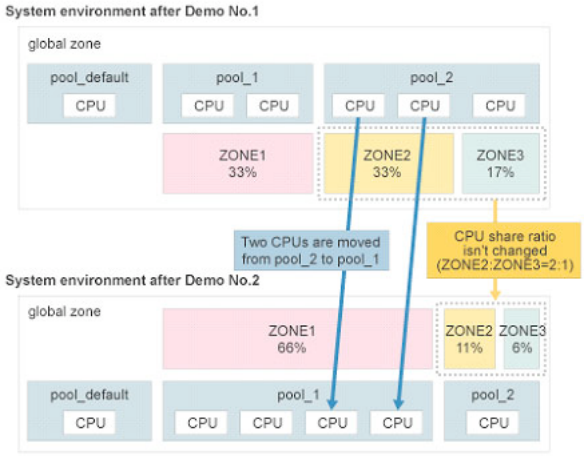
- CPU share ratio of ZONE2 and ZONE3 (both in pool_2) is changed from 1:1 to 2:1
- CPU allotment tuning will change CPU share ratio of ZONE2, and of ZONE3



)6/08

Demonstration 2

- Two CPUs are transferred from pool_2 to pool_1
  - In the demonstration1 environment two CPUs are moved from pool_2 to pool_1. As a result the CPU share ratio of ZONE1 increases.
  - The CPU share ratios of ZONE2, and ZONE3 go down, but the proportion between the two zones doesn't change (2:1).



AP 06/08

Virtualization Technology

- Solaris ZFS
  - 128 bit address space enables you to handle large scale file systems with simplified management and data protection mechanisms
  - almost unlimited data capacity:
    A file system in ZFS can contain 256 quadrillion zettabytes of storage
  - simple data management and a high-profile data protection mechanism.

AP 06/08

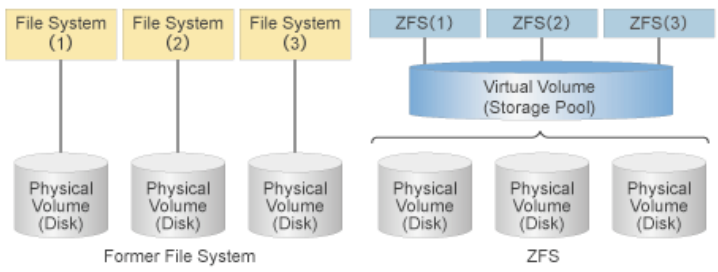
ZFS storage pool

- storage management for multiple disk devices.
  - All file systems constructed in a storage pool are extendable by simple operation.
  - This is because file systems in ZFS are independent of physical volumes and file system capacity can be increased without physical volume definition.
- historically: file system on specific physical volumes
  - So, file system capacity expansion needs time-consuming procedures including physical volume re-configuring and, perhaps new disk installation.
  - Such complexity is caused by the explicit relationship between the file system, physical volume and disk device.

AP 06/08

ZFS storage pool

- File systems are no longer bound to disk devices.
- Storage pool manages multiple disk devices and provides a virtual storage interface to file systems.
- Any file system can be easily extended, even while the system is operational.



AP 06/08

Web-based management tool for ZFS

Main features:

- Create a storage pool
- Add capacity to a storage pool
- Move (export) or import a storage pool
- View information on a storage pool
- Create a file system/volume
- Take s snapshot of a file system/volume
- Roll back a snapshot of a file system/volume

AP 06/08

Data Protection mechanism

- ZFS improves business continuity
  - data correction and data mirroring
    - Data, if corrupted, is detected and corrected using a 64 bit checksum mechanism
    - Moreover, checksum information is stored in an area separate from the data itself, preventing checksum information corruption even during data corruption.
- mirror data repair function
  - replaces any corrupted data with correct data.
    - ZFS mirror maintains a copy of all data on a different location from the original.
    - When the checksum mechanism detects a data corruption, the corrupted data is repaired by replacing it with correct data from the mirror.

AP 06/08

Predictive Self-Healing

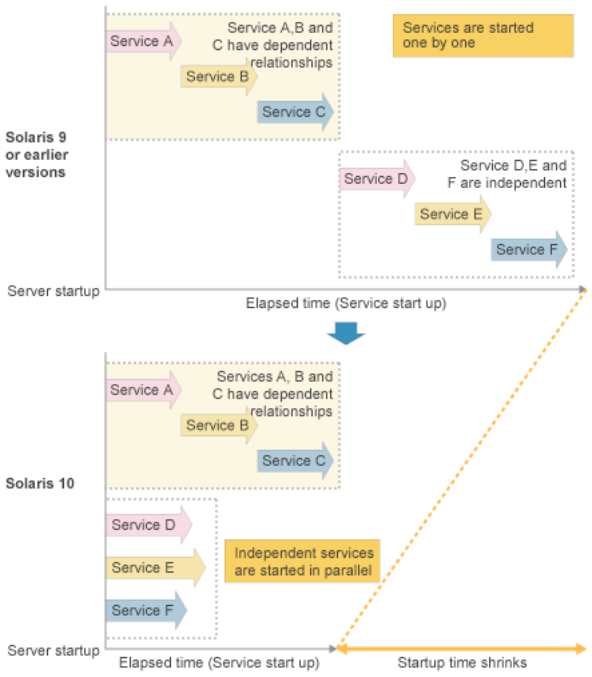
- Predictive Self-Healing is a set of automated management functions that reduce the work of system management.
  - They restart services swiftly, assist in pinpointing hardware errors and provide a rich array of analysis and problem isolation features.
  - Two major new functions, Solaris Service Manager and Solaris Fault Manager have been included in the latest version.
- Solaris Service Manager unifies service management functions.
  - This enables much swifter server start up and more rapid halt of the system following service failure.
- Solaris Fault Manager takes remedial action following server hardware error,
  - providing the right prescriptive action to the system administrator

AP 06/08

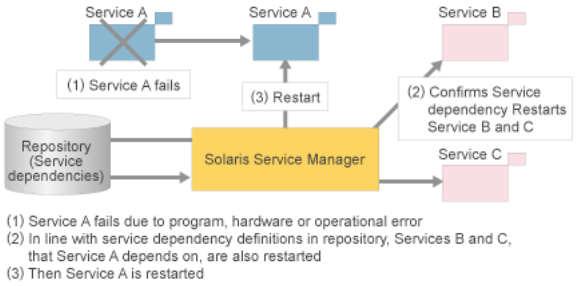
Solaris Service Manager

- Solaris Service Manager unifies service control by managing the interdependency between services, ensuring that they are started (or restarted following service failure) in the appropriate order
- Solaris Service Manager shrinks service startup time by starting independent services in parallel.
  - With Solaris 9, services are started one by one following a service start script.
  - This startup method is simple and adequate when handling small-scale systems.
- However, with UNIX servers being used for more and more large-scale systems with many more services, it takes a long time to start them all in sequence.
  - Now with Solaris 10, non-dependent services are started in parallel based on a stored services relationship configuration.
  - This new feature enables much quicker system startup and recovery, leading to greater business continuity and service availability.

AP 06/08



AP 06/08



(1) Service A fails due to program, hardware or operational error
(2) In line with service dependency definitions in repository, Services B and C, that Service A depends on, are also restarted
(3) Then Service A is restarted

Automatic service restarting

- With Solaris 9 OS, service management required a detailed knowledge of services.
  - If a service failed, the system administrator had to not only restart the failed service, but all other services that service depended on.
  - Such careful operation had to be undertaken manually and required the administrator to have an intimate knowledge of the service dependencies.
- With Solaris10, the system administrator no longer needs to know the interdependency of services.
  - Solaris Service Manager holds the service interdependencies allowing it to automatically restart all necessary services without administrator intervention.
  - Solaris Service Manager constantly watches server status and automatically restarts any failed service.

AP 06/08

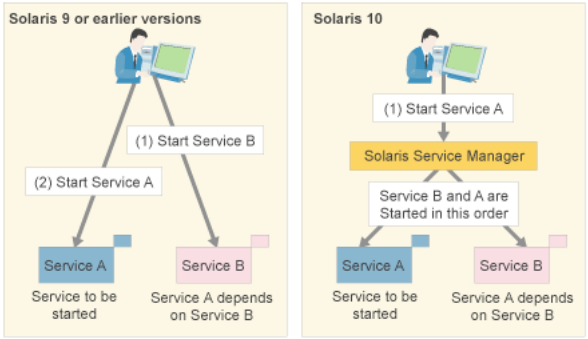
Service management details

- svcs(1) command
  - System administrators can more easily monitor services using Solaris Service Manager's service status information and service activation/ deactivation interfaces based on the commands (svcs(1), svcadm(1) etc).
  - Until Solaris 9, it was a complicated procedure to understand service status. Service level information was not provided and system administrators have to assume service status from their own analysis of kernel level information. A slow and error prone process.
- svcadm(1M) command
  - Services and the services on which they depend are started in their appropriate order using the Solaris Service Manager svcadm(1) command.
  - System administrators are longer required to run complicated service startup operations.
- With Solaris 10 kill(1) or pkill(1) commands are no longer available.
  - This is because, once stopped, Solaris Service Manager will automatically restart them.
  - So a new command, svcadm (1M) is now used for stopping services.

AP 06/08

Service Manager Example

- Service A depends on Service B
  - previously the system administrator needed to start the services paying close attention to their dependent relationship.
  - Now with Solaris10, they only have to start Service A.
  - Solaris Service Manager automatically detects that Service B needs to be started, and starts the services in the right sequence.



System administrator checks dependent Relationship and starts services in a Definite order by operational commands

Solaris Service Manager checks all necessary services and automatically starts then in the correct order

AP 06/08

Solaris Fault Manager

- Solaris Fault Manager reduces service downtime by automatically handling hardware errors, including error detection, error analysis and failed part isolation.
  - The result is a major improvement in server process availability.
- This new function provides enhancement to the PRIMEPOWER machine management function called Enhanced Support Facility (ESF - Fujitsu Siemens detail)
  - Combined with PRIMEPOWER's own hardware monitoring technology it provides even higher-levels of server failure management.

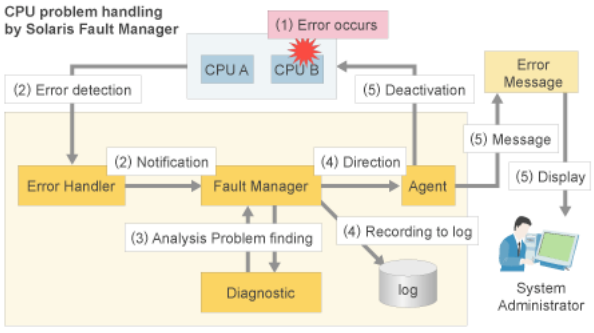
AP 06/08

Hardware error diagnostic mechanism

- In Solaris10 the hardware monitoring mechanism has has three main components.
  - The Error Handler detects errors.
  - The Fault Manager receives error information from the Error Handler.
  - The Diagnostic Engine provides the cause and effect resulting from the error information.
- Fault Manager records the error to a log file and directs an agent to deactivate any failed component.
  - Finally, an error message is sent to the system console screen providing details to the administrator for remedial action to be taken

AP 06/08

Fault Manager Example



(1) Error occurs in CPU B
(2) Error handler detects error and notifies Fault Manager
(3) Diagnostic Engine analyses the error and decides CPU B has an error
(4) Fault Manager directs Agent to deactivate CPU B
(5) Agent deactivates CPU B and create error message
(6) System administrator checks details of the error

AP 06/08

DTrace (Dynamic Trace)

- If performance deteriorates or other server system problems occur Dynamic Trace (DTrace) helps you resolve the situation.
  - DTrace lets you monitor and understand the operational state, detailed system behavior and system problems, on your server.
- Winner of the Gold award in Wall Street Journal's Technological Innovation Award in 2006
  - DTrace was evaluated top in break-through technology against more than 600 applications.
  - Its minimal workload and around 40 thousand tracing points imbedded in the Solaris™ 10 Operating System, provide powerful trace filtering functions.
  - These allow you to monitor system behavior in detail - obtaining just the right information anytime you need.

AP 06/08

Before DTrace

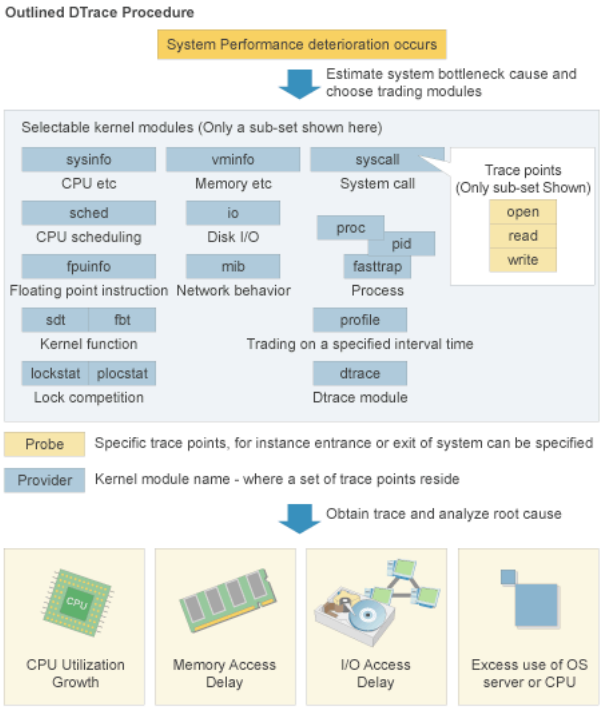
- Traditional system analysis tools could in themselves affect system performance.
  - This meant that system administrators often had insufficient opportunity to obtain useful analysis information.
  - As a result they had to presume the cause and then confirm their presumptions by trial and error and their own experience.
- Traditional system analysis tools were also primarily aimed at the debugging of single applications.
  - This often made them inadequate at investigating the behavior of an entire server.
  - For instance, kernel information tools such as process status lists or system call traces, although able to provide important clues, don't directly lead to resolution due to the coarseness of information provided.
  - With intermittent faults, memory snapshot, which is often the investigator's course of last resort, is hard to use effectively so takes a significant time to reach a conclusion.

AP 06/08

DTrace Technical Outline

- DTrace, creates an environment in which approximately 40 thousand trace points are automatically populated around the Solaris 10 system.
- The DTrace environment spans the Solaris 10 system.
  - Each trace point provides focus and fine grained information on system performance, process deterioration and other system problems.
- Trace actions are controlled by D Script.
  - This lets you select target kernel modules, call name information and the output format.
  - It also lets you focus on specific information, for example invalid data, delayed processing time or repeated system calls.

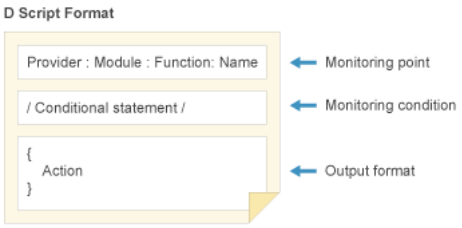
AP 06/08



AP 06/08

D Script format

- D Script has its own simple command definition language (D Language).
  - System administrators will find D language easy to use as it has similar syntax to the programming language C.
  - Just a few simple rules need to be learnt.
- Monitoring point, monitoring condition and output format are the three elements of the D language.



AP 06/08

Monitoring point

- The top line of D Script, (the monitoring point definition) specifies the trace points and necessary information.
- It consists of just four comma separated fields: Provider, Module, Function, Probe.
  - Provider: Is the target kernel module set or library to which the trace points belong.
  - Module: Is the target module name (This parameter can be omitted)
  - Function: Is the name of the program functions such as system call name (This parameter can be omitted)
  - Probe: Specifies the condition or point were tracing information will be obtained. (Typically this represents a location in a function)

Example	Explanation
syscall : : exec : return Provide Module Function Prove	Just befor exec system call was finished and returning to caller

AP 06/08

D Script (contd.)

- Monitoring condition
  - This statement specifies the condition for obtaining traced information. Statement is bounded by forward slashes ("/").

Example	Explanation
/ pid = = 123 /	Trace information is obtained only when process id is 123

- Output format
  - This statement specifies a output format of traced information. The statement is between parentheses ("{" and "}").

Example	Explanation

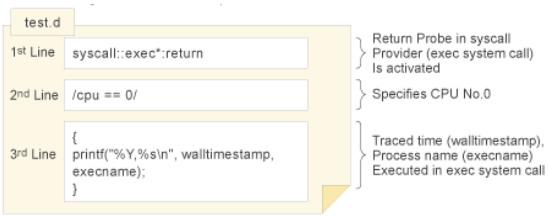
{
printf ("%s", execname)
}
	Executed program name by exec system call is put on screen or file

- D Script can be also specified in command format.

DTrace Command format

```
# dtrace -n Provider : module : function : name ' /condition/{action}
```

AP 06/08



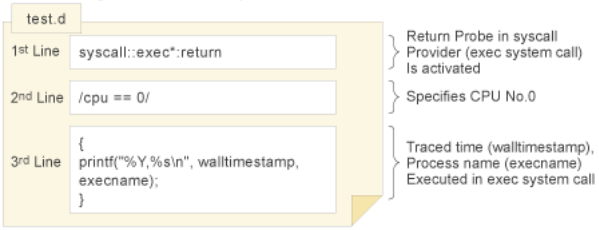
DTrace example

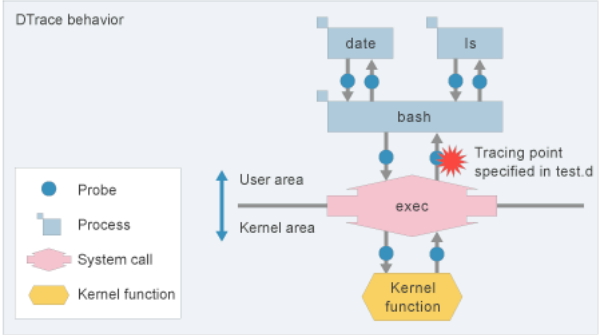
- Let's look at an example that analyses system performance.
  - The following example shows that a process executing on CPU No. 0 issues a system call, exec(2).
- Let's prepare the D Script.
  - On the first line, you specify the monitoring point.
  - To check status just after issuing an exec system call, you specify Provider field as Syscall, Function field as exec, and Probe field as return.
- On the second line, you specify the conditional statement.
  - you specify CPU No. as 0.
- On the third line, you specify the output format.
  - You specify timestamp (walltimestamp) and the process name that exec(2) system call was executing.

AP 06/08

D Script

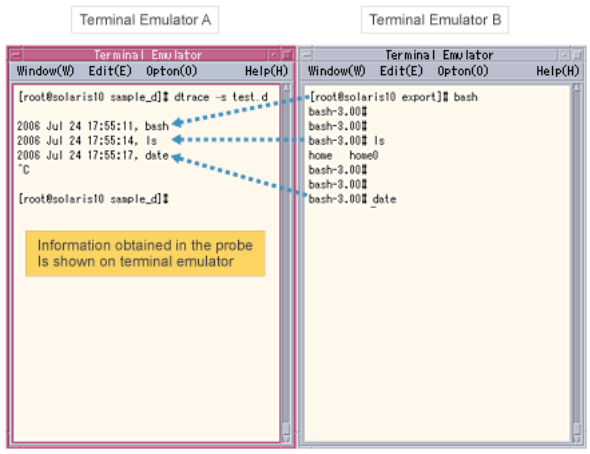
The program name called from the exec(2) system call and the finish time are stamped.
Plus the tracing condition is that the process was executed on CPU No.0





AP 06/08

DTrace execution



- Following D language definition, you only have to execute dtrace(1M) command.
- Executing a Unix command such as date(1) or ls(1) on another screen, will let you confirm those commands were executed.

AP 06/08

DTrace characteristics

- DTrace, lets you obtain both kernel information and library information enabling you to fully investigate from both OS and application viewpoints.
- The 40 thousand trace points in the OS mean you don't need to write your own debug programs.
  - They also let you obtain relevant records close to the fault site.
  - This ensures faster cause detection and swift resolution of application and system bottlenecks.
  - Plus because you can filter out irrelevant trace points you can maintain good system performance.
  - Now you can analyze your system at your site, without any need to build a clone system for troubleshooting.

AP 06/08

Addl. improvements

- Security-Process Privilege Administration
  - High-level access control, controlling accesses to system resources in a detailed level, can improve entire system stability.
- Performance improvement
  - Renewed TCP/IP control program provides high network performance.
  - Just an upgrade to Solaris 10 will improve your system performance.

AP 06/08

References

- Solaris 10 at Fujitsu Siemens: http://www.fujitsu.com/global/services/computing/server/unix/os/solaris10/
- Predictive Self-Healing wins InfoWorld 200 Innovator Award Innovation Awards: The winners are... (The Wall Street Journal Online)
- Solaris Dynamic Tracing Guide http://docs.sun.com/app/docs/doc/817-6223
- Solaris 10 Software Developer Collection http://docs.sun.com/app/docs/coll/45.20

AP 06/08