Oracle Solaris Virtualization Technologies

● Akitoshi Ozawa ● Katsuyuki Suzuki ● Masami Taoda

There is an urgent need to reduce system optimization costs in the information and communications technology (ICT) infrastructure, and virtualization technology is widely expected to be key to more efficient use of system resources. A standard installation of the Oracle Solaris operating system supports server virtualization with Solaris Containers and Oracle VM Server for SPARC (VM: virtual machine) and storage virtualization with Solaris ZFS (Solaris Zettabyte File System). Solaris ZFS is a highly reliable file system that uses storage pool technology and snapshot/cloning technology to increase disk usage efficiency. This paper discusses the domain management, resource management, and domain migration features of Oracle VM Server for SPARC and the zone management, resource management, and zone migration features of Solaris Containers. The ZFS storage pool and ZFS file system of Solaris ZFS are also introduced.

1. Introduction

The information and communications technology (ICT) infrastructure has expanded the scale of systems so that business can be conducted faster and more efficiently. But as the number of servers increases, it becomes more important to reduce their operating and administration costs and power consumption.

Server virtualization is a technique for efficiently allocating system resources such as central processing units (CPUs) and memory according to the workload on each server so as to reduce up-front costs and enable large numbers of servers to be integrated economically. It is possible to create virtual servers instantly without having to wait for new hardware to be delivered and to discard them when no longer needed. Server virtualization is the first step towards system optimization.

The Oracle Solaris Operating System (Solaris OS)[1] installed in Fujitsu's SPARC (scalable processor architecture) Enterprise UNIX server features virtualization functions that become more powerful with every update release. Among the virtualization functions provided as standard in SPARC Enterprise, this paper introduces the Oracle VM Server for SPARC (VM: virtual machine) and Solaris Containers used for server virtualization[2), 3)] and the Solaris ZFS (Solaris Zettabyte File System) functions for storage virtualization.[4), 5)]

2. SPARC Enterprise virtualization functions

SPARC Enterprise provides three server virtualization functions as standard through hierarchical partitioning of the virtual server (**Figure 1**).

1) Hardware partition

A partition running an independent Solaris OS can be configured by logically partitioning the physical system board. The CPU and memory can be dynamically modified in response to requests for business expansion, the addition

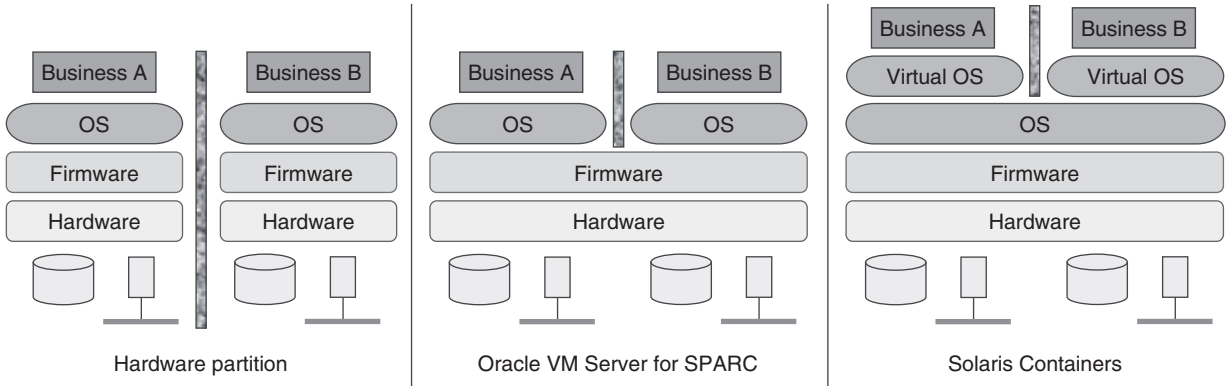

Figure 1
SPARC Enterprise virtualization functions.

of new business, and so on without the system operation being halted.

2) Oracle VM Server for SPARC

Oracle VM Server for SPARC (previously called Sun Logical Domains)[6] can configure logical domains running separate instances of Solaris OS by using the SPARC hypervisor in the firmware layer to partition the physical server into virtual servers. The CPU, memory, and input/output (I/O) devices are flexibly allocated by the Domain Manager.

3) Solaris Containers

Solaris Containers allows the Solaris OS to be virtually partitioned into zones that constitute independent virtual OS environments. CPUs and memory are flexibly allocated according to the zone's operating conditions. I/O devices are allocated when the zone is configured.

Solaris OS also provides Solaris ZFS for storage virtualization as a standard function. The ZFS file system manages multiple physical disks as a storage pool. Virtualized volumes can be created by allocating the necessary space from the storage pool. The ZFS file system is not only durable and scalable but also easy to administer. The virtualization functions of Solaris OS are described below.

2.1 Oracle VM Server for SPARC

A logical domain is a virtual server in which the CPU, memory, and I/O devices are logically grouped and an independent Solaris OS is running in each logical domain. Each logical domain can be started up and shut down independently, and up to 128 domains can be created. Logical domains can communicate with each other via hypervisor logical domain channels (LDCs). Virtual devices such as disks and networks access the physical devices by using LDCs to communicate with virtual services.

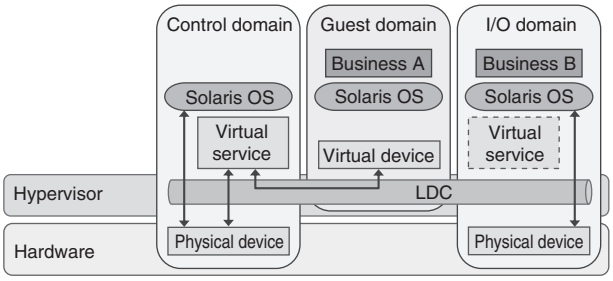

Figure 2
Roles of logical domains.

2.1.1 Role of logical domains

Logical domains have three roles (**Figure 2**).

1) Control domain

A control domain is a domain operated by the Domain Manager to create and administer other logical domains and allocate virtual resources to them. It also provides virtual device services such as virtual disk servers and virtual

switches.

2) I/O domain

An I/O domain can access physical I/O devices directly. Like a control domain, it can also provide virtual device services.

3) Guest domain

A guest domain receives virtual device services from a control or I/O domain, enabling itself to use virtual I/O devices such as virtual disks and virtual networks.

2.1.2 Resource management

Resources such as virtual CPUs, memory, and virtual I/O devices can be dynamically reconfigured while a logical domain is still operating. Furthermore, the number of virtual CPUs of logical domains can be automatically increased or decreased according to a dynamic resource management policy. The dynamic resource management policy is drawn up by combining various factors such as the number of CPUs that are used, the usage ratios, upper/lower limits, and time slots. In CPU power supply management, power savings can be achieved by stopping the power supply to CPUs that fall idle as the workload changes.

2.1.3 Domain replication

An existing domain can be replicated to enable virtual servers for new domains to be rapidly provided (provisioned) in response to user requests. If a startup disk image for a guest domain is stored in the ZFS file system, then it can be instantly replicated as a ZFS clone, as described below. Allocating the replicated startup disk image to another guest domain makes it unnecessary to go through the Solaris OS installation procedure.

2.1.4 Migration

With changes in workload, incremental growth in the number of servers and so on, it is possible to migrate a guest domain to another physical server. The guest domain is temporarily halted, and its memory contents are compressed and transmitted at high speed. When servers are consolidated, a physical server can be migrated to a logical domain by using a physical-to-virtual (P2V) migration tool. The configuration information of the physical server is collected to create a file system image. A logical domain is created on the basis of this configuration information, and the file system image is restored on the virtual disk.

2.2 Solaris Containers

Solaris Containers consists of a Solaris Zone function for virtually partitioning a single OS space to make it appear as if multiple OSs are running and a Solaris Resource Manager that flexibly allocates hardware resources such as CPUs and memory. A Solaris Zone is a virtualized OS environment that implements a safe isolated environment suitable for running applications. Processes running in each zone are isolated and unable to affect other zones.

2.2.1 Solaris Zones

A Solaris system has just one global zone, which is responsible for managing the entire system (**Figure 3**). Tasks such as the creation and administration of non-global zones and

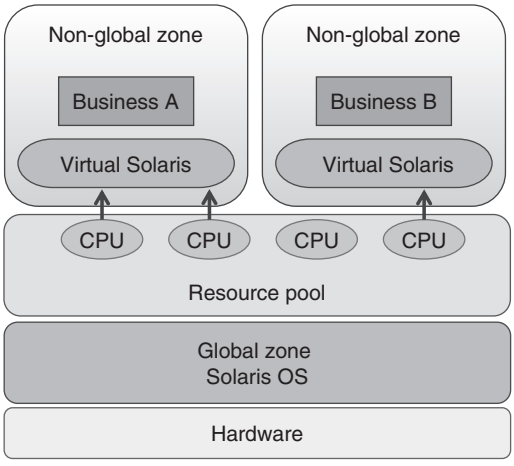

Figure 3
Solaris Containers configuration.

the allocation of physical I/O devices can be performed only in the global zone.

A non-global zone is a software partition of a virtual Solaris environment, in which applications can run without affecting other zones. Up to 8191 zones can be created, each of which can use only its permitted file system and permitted physical I/O devices.

The constituent system files of a non-global zone are copied from the global zone when the zone is created. When the global zone is patched to modify these files, all the non-global zone files are also synchronously updated.

2.2.2 Solaris Resource Manager

The Solaris Resource Manager periodically monitors resource usage and automatically allocates resources within a specified range without stopping the zone. The resources allocated to a zone are managed by a resource capping daemon that controls the memory and resource pool controlled by the CPU. Furthermore, the resource pool is configured from a processor set comprising a group of CPUs and a scheduling class (time-sharing, fair share) that controls CPU usage allocation.

2.2.3 Zone replication

A new zone can easily be created by copying an existing zone. When a zone exists in a ZFS file system, it can be replicated instantly by using the ZFS cloning function described below, and it is possible to save disk capacity usage.

2.2.4 Zone migration

When resources such as CPUs and memory become exhausted or the combination of zones is changed, the zones can be migrated to another server. This involves detaching zones from their original server and attaching them to the destination server. Even if the two servers have different environments (e.g., different package configurations), the system files that configure the zone during attachment are synchronized with the global zone of the destination server.

Furthermore, an active Solaris 10 OS system can use the P2V function to perform migration by unpacking a flash archive into a zone. The Solaris 8/9 OS system can use Solaris 8/9 Containers to perform migration to a Solaris 10 OS zone without modifying the applications.

2.3 Solaris ZFS

Solaris ZFS is a 128-bit file system that can manage a practically unlimited data capacity. The metadata used for the administration of a ZFS file system is dynamically allocated as required, so there is no limit to the number of file systems or the number of files. In a conventional file system, the file system size is limited to the physical device size. However, Solaris ZFS is not limited to specific physical devices because the physical devices are hidden by the ZFS storage pool. The ZFS file system can create file system hierarchies easily without initialization, and it automatically expands within the range of the disk capacity allocated to the ZFS storage pool.

2.3.1 Storage pool

The ZFS storage pool is a mechanism that collectively manages physical disks and can select non-redundant, mirroring, RAID-Z (single parity), RAID-Z2 (double parity), or RAID-Z3 (triple parity) (RAID: redundant array of independent disks) as redundancy configurations. When data is written to the storage pool, it is dynamically striped across all the available devices. If a defective data block is detected in a redundantly configured storage pool, the correct data is automatically retrieved from a redundant copy.

2.3.2 Storage pool management

The storage pool allows disks to be added and replaced without going offline, so when systems are introduced there is no need to provide them with the entire disk capacity needed to meet their projected future requirements (**Figure 4**).

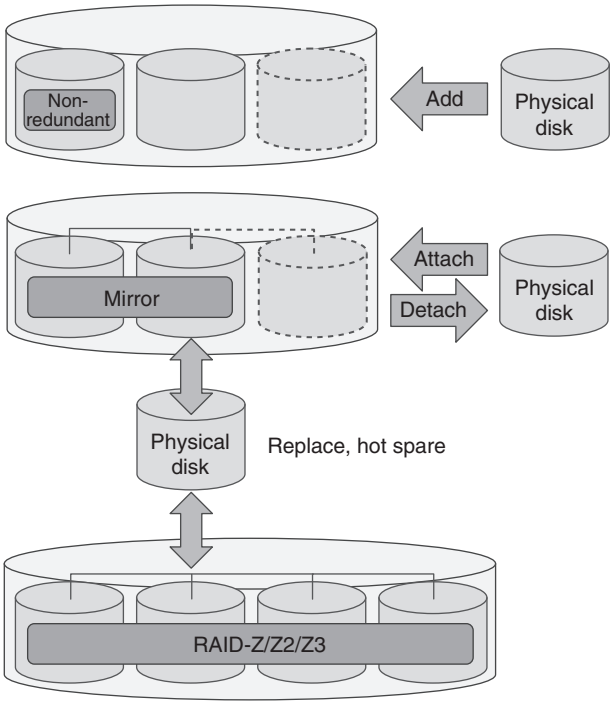

Figure 4
ZFS storage pool configuration.

ZFS storage pool management has three key functions.

1) Attaching and detaching

Mirroring is achieved by attaching disks to an existing mirrored or non-mirrored storage pool. When a disk is added, resynchronization begins immediately. The mirror configuration can be modified by detaching a disk from a mirrored storage pool. When an individual disk is offline, it can be temporarily disconnected. When the disk is online, its data is resynchronized.

2) Replacement

In a redundant storage pool, disks can be replaced dynamically. If a hot spare disk is provided, then a disk that has failed or generated errors in the storage pool can be replaced automatically. Hot spare disks can be shared among storage pools.

3) Migration

Storage pools can be migrated to other servers. When a storage pool is exported, all of the unprocessed data is written to the disks. The constituent disks of the storage pool are all attached to the destination server and can be made available simply by importing the storage pool.

2.3.3 Storage pool performance improvement

To improve performance and reduce power consumption, it is possible to configure a ZFS hybrid storage pool from a combination of random access memory, solid-state drives (SSDs), and hard disk drives. The ZFS Intent Log (ZIL) usually uses the regions inside the storage pool, but improves the synchronous write performance by allocating high-speed devices such as SSDs to individual log devices. The cache device (L2ARC: level 2 adaptive replacement cache) improves the random read performance for static data by adding high-speed devices such as SSDs as a cache between the memory and disks.

2.3.4 File system

The ZFS file system is a transaction-based file system. Data is written not by overwriting the existing data but by updating a copy of the original data (copy-on-write). After a sequence of data update processes has been completed, the pointers to the new and old data are swapped so that the integrity of the file system is always maintained. Even if the server is suddenly powered off, the file system is not damaged in any way.

When a file system is created, a mount point is automatically generated and mounted. Since it is automatically mounted when the server is started up, no administrative intervention is needed to mount the file system. ZFS volumes are identified as block devices. When a volume is created, space is reserved for the initial size, and the volume is automatically expanded according to usage.

2.3.5 File system reliability

The ZFS file system uses end-to-end checksums for the metadata of all user data

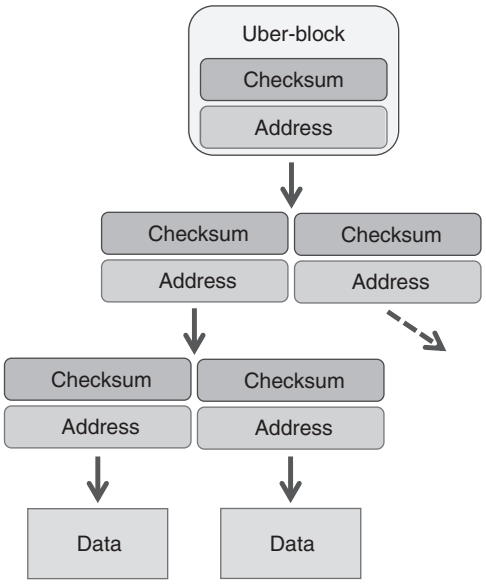

Figure 5
ZFS end-to-end checksums.

and management information. Data block checksums are stored in the parent block, and this continues up to the top management block (uber-block) so that the entire data tree can be self-validated (**Figure 5**). If an error is detected, the data is restored from a redundant copy. For greater reliability, the ZFS file system metadata is automatically stored a number of times across different disks (ditto blocks). The ZFS file system can also store multiple copies of user data. In cases where redundancy across multiple disks is not possible, recovery from a disk block read failure is still possible.

2.3.6 Snapshots and clones

A ZFS snapshot is a read-only copy of a file system or volume that can be instantly created without consuming disk space. A snapshot cannot be referenced directly, but can be used for operations such as rollback, cloning, and backups (**Figure 6**). When the original file system or original volume is updated, disk space is consumed only by the parts where data was changed.

When a rollback is performed, the changes

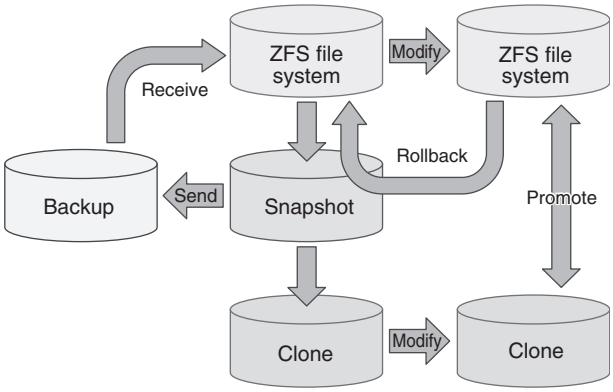

Figure 6
ZFS snapshots and ZFS clones.

made to the data since the creation of a snapshot are deleted, and the file system is returned to the state when the snapshot was created.

A ZFS clone is a writeable copy of a file system or volume that can be created from a snapshot. Like a snapshot, it can be created instantly without consuming disk space, and disk space is consumed only by parts where the data is changed. A file system can be replaced with its clone by promotion. ZFS cloning makes it easy to replicate zones, virtual disks in guest domains, and Solaris Live Upgrade boot environments.

A file system backup can be made by creating a stream from a snapshot. This stream is sent to another storage pool, where the file system is recreated. It can be sent as an incremental snapshot containing only data that has changed. Moreover, instead of being expanded in another storage pool, a backup can be saved as an archive in various formats.

3. Conclusion

This paper introduced the virtualization functions in Solaris OS: server virtualization implemented in Oracle VM Server for SPARC and Solaris Containers and Solaris ZFS storage virtualization. We intend to continue enhancing the virtualization capabilities of Solaris OS, and we will contribute to system optimization.[7)]

References

1) Fujitsu: Oracle Solaris 10. *http://www.fujitsu.com/global/services/computing/server/sparcenterprise/products/software/solaris10/*
2) M. Yuhara: Prospects for Virtual Machine Technology. (in Japanese), *FUJITSU*, Vol. 60, No. 3, pp. 221–227 (2009).
3) Y. Oguchi et al.: Server Virtualization Technology and Its Latest Trends. *Fujitsu Sci. Tech. J*, Vol. 44, No. 1, pp. 46–52 (2008).
4) T. Kumazawa: Overview of Storage Virtualization. (in Japanese), *FUJITSU*, Vol. 60, No. 3, pp. 241–246 (2009).
5) T. Akasaka et al.: Virtualization of ETERNUS Disk-array Subsystem. (in Japanese), *FUJITSU*, Vol. 60, No. 3, pp. 253–257 (2009).
6) Fujitsu: Logical Domains Manager Software. *http://www.fujitsu.com/global/services/computing/server/sparcenterprise/products/software/ldoms/*
7) Fujitsu: Solaris Technical Park. (in Japanese). *http://primeserver.fujitsu.com/sparcenterprise/technical/*

Akitoshi Ozawa
Fujitsu Ltd.
Mr. Ozawa is engaged in the planning and development of virtualization software.

Katsuyuki Suzuki
Fujitsu Ltd.
Mr. Suzuki is engaged in the planning and development of virtualization software.

Masami Taoda
Fujitsu Ltd.
Mr. Taoda is engaged in the planning and development of virtualization software.