FUJITSU

shaping tomorrow with you

ZFS Overview and Design Guide

December 2016 (Edition 1.0)

Fujitsu Limited

Preface 1/2

Purpose

- This document presents an overview of ZFS (Zettabyte File System), which is the standard file system of Oracle Solaris 11.

Audience

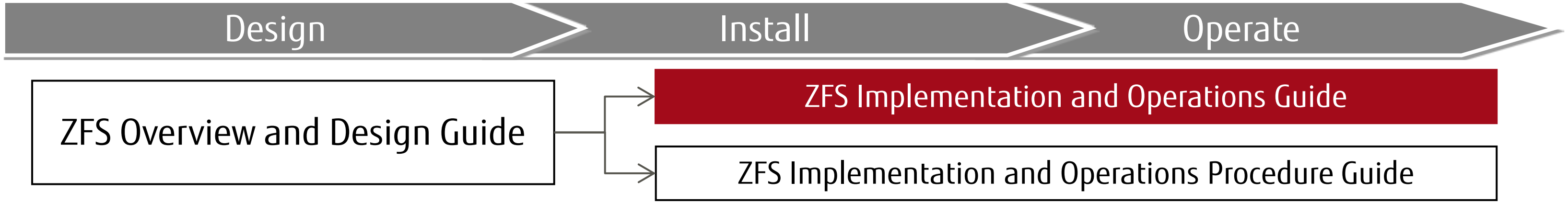
- People who have a basic knowledge of Oracle Solaris and RAID

Notes

- The contents of this document are based on Oracle Solaris 11.3. For the latest information on Oracle Solaris 11, see the manuals from Oracle.
  - Oracle Solaris 11 Documentation
    http://www.oracle.com/technetwork/documentation/solaris-11-192991.html
- Fujitsu M10 is sold as SPARC M10 Systems by Fujitsu in Japan. Fujitsu M10 and SPARC M10 Systems are identical products.

Positioning of documents

- ZFS
  http://www.fujitsu.com/global/products/computing/servers/unix/sparc/downloads/documents/


Design
Install
Operate
ZFS Overview and Design Guide
ZFS Implementation and Operations Guide
ZFS Implementation and Operations Procedure Guide

Preface 2/2

- Descriptions in this document
  - The section numbers of commands are omitted.

    Example:
    - ls(1) => ls command
    - shutdown(1M) => shutdown command
  - The following table lists terms that may be abbreviated.

Abbreviation	Formal Name
Solaris	Oracle Solaris

Contents

1. Overview of ZFS
2. ZFS Features
3. ZFS Reliability and Availability
4. Storage Pool Configuration
5. File System Management
6. ZFS Functions
7. ZFS Operations

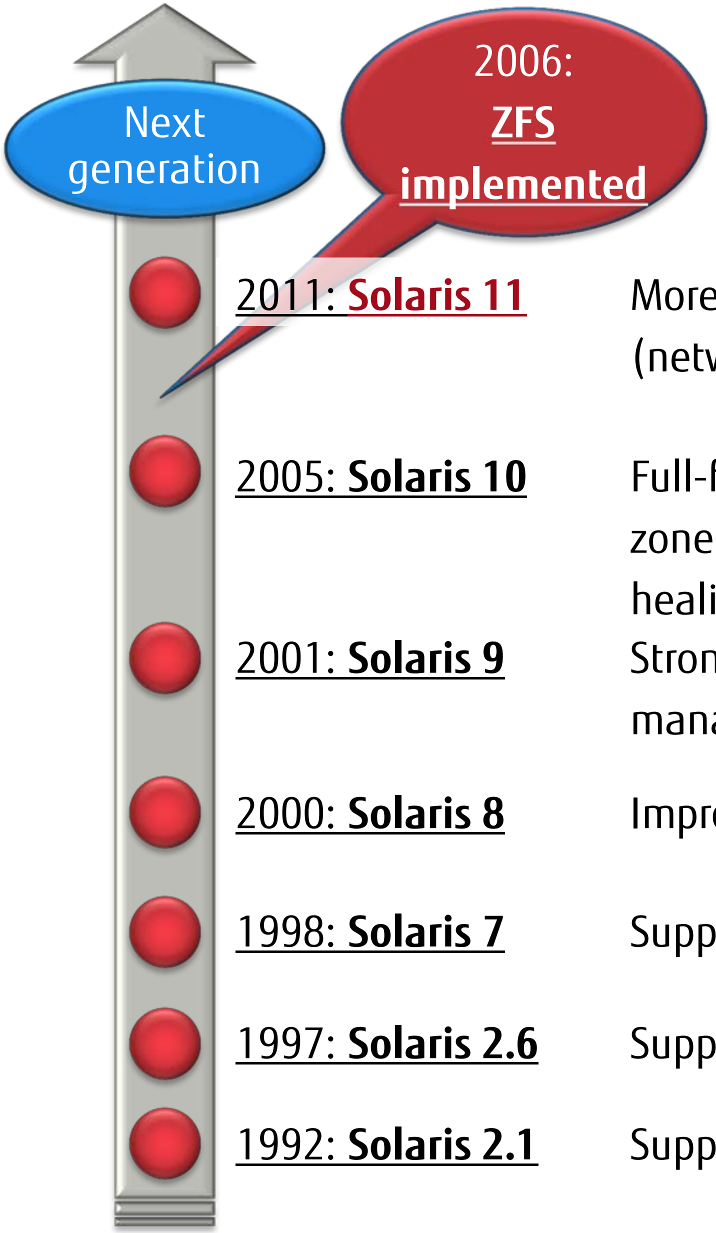
- Appendix

1. Overview of ZFS

This chapter provides an overview of the ZFS file system and describes the file system mechanism and the environment for virtualization made possible by ZFS.

Oracle Solaris History and ZFS

- ✓ **Over 20 years of binary compatibility (reliable protection of customer assets)**
- ✓ **Continuously evolving to meet the needs of the times**



Next generation

2006: **ZFS implemented**

2011: **Solaris 11** — More strongly supported virtualization and operability for the cloud (network virtualization function, IPS, Boot Environment)

2005: **Solaris 10** — Full-fledged implementation of the virtualization function (Solaris zone), improved availability and manageability (predictive self-healing, SMF)

2001: **Solaris 9** — Stronger resource management (Java multithread support, resource manager), expanded Linux compatible tools and desktop tools

2000: **Solaris 8** — Improved security and availability (RBAC, multipath I/O)

1998: **Solaris 7** — Support of 64-bit machines and the Java language

1997: **Solaris 2.6** — Support of large-scale files (1 TB)

1992: **Solaris 2.1** — Support of the SPARC architecture

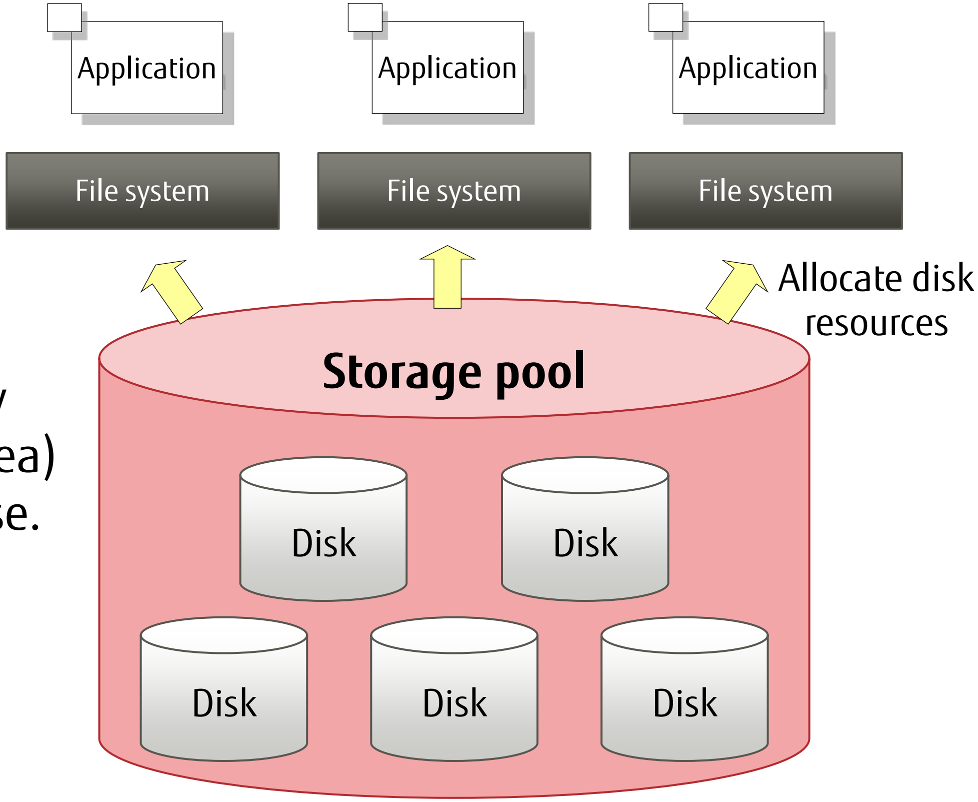
Overview and Mechanism of ZFS

Overview of ZFS

- ZFS is the next-generation standard file system of Solaris 11.
- ZFS has excellent characteristics and functions in terms of operation management and reliability.

Mechanism of ZFS

- The function called "storage pool" collectively manages and pools multiple disks.
- With the storage pool, you can freely create a file system (data storage area) for each application and intended use.
- You can easily expand the capacity of the storage pool by adding a disk.



Server and Storage Virtualization Realized in Oracle Solaris

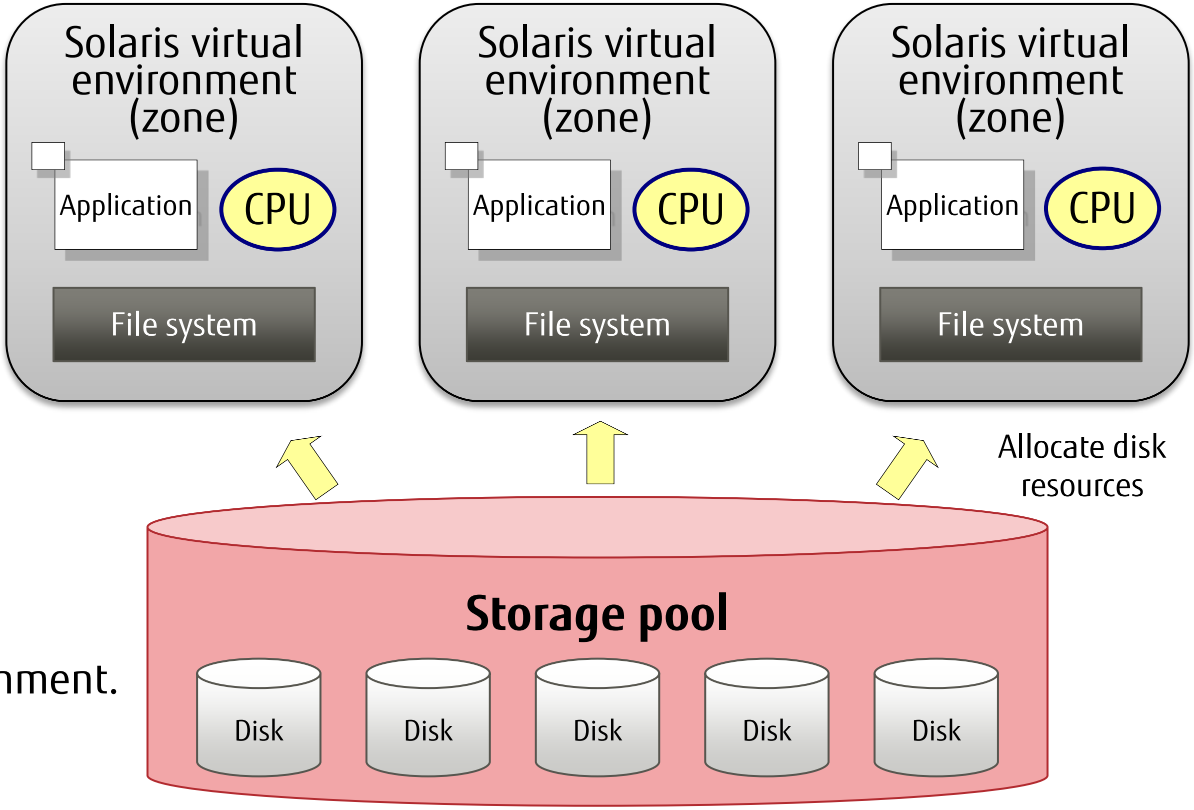
ZFS also has the aspect of a "storage virtualization function" for easily managing and allocating many large-capacity disk resources.
By combining ZFS and the server virtualization function called Solaris zones, you can build a virtualization infrastructure with servers and storage together.

Solaris zone

- Build multiple virtual OS environments.
- Allocate CPU resources flexibly to each virtual environment.

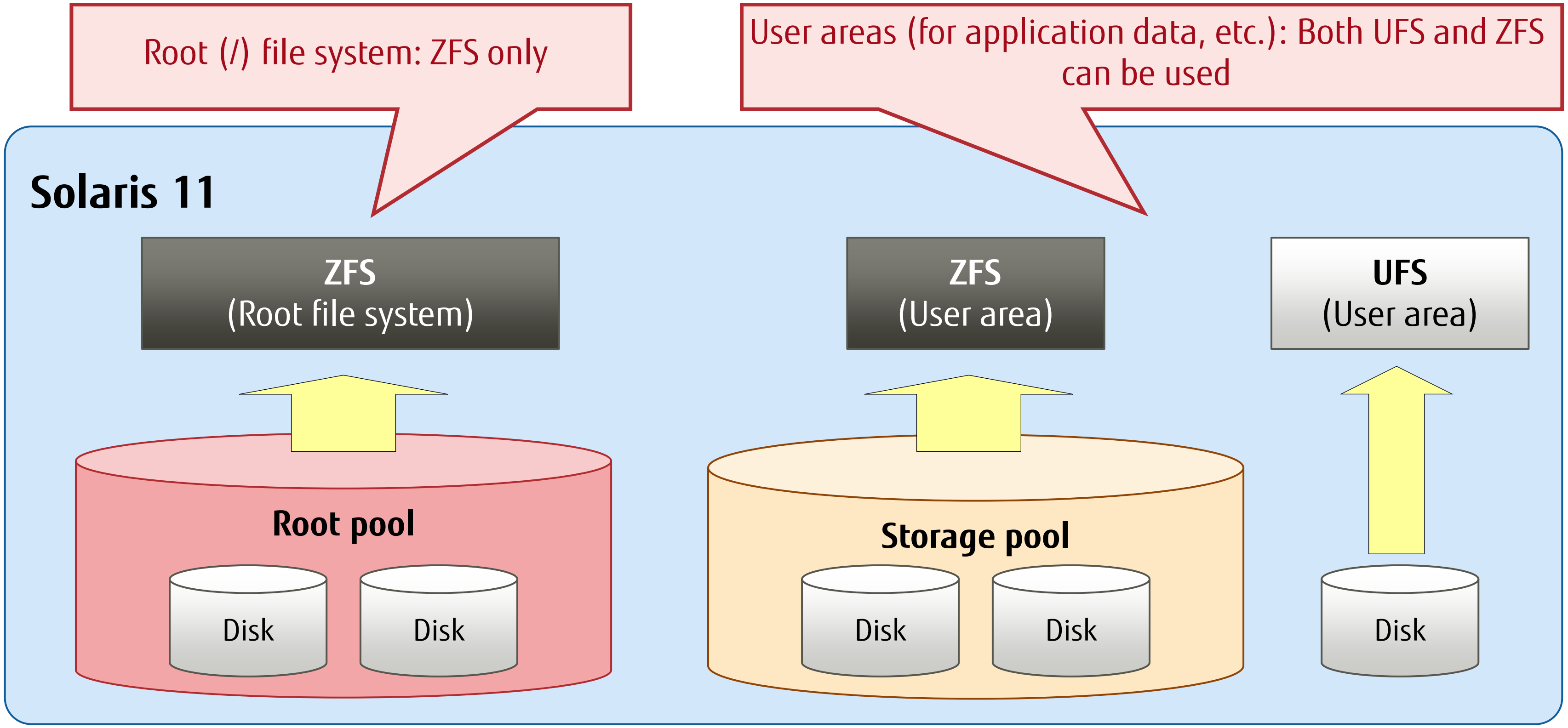
ZFS

- Centrally manage all disks from the storage pool.
- Freely allocate disk resources (capacity) to each virtual environment.



Standard Use of ZFS in Oracle Solaris 11

- Solaris 11 uses ZFS as the standard file system.
  - Only ZFS is supported for the system area (root file system) of Solaris 11.
  - The existing UFS file system can be used only for user areas.
  - The storage pool of the system area is called the "root pool."



2. ZFS Features

This chapter describes the ZFS features, which are scalability, manageability, and data robustness.

ZFS Features

- Scalability
  - World's first 128-bit file system
  - File system builds said to have virtually infinite capacity
- Manageability
  - Simple management commands
  - Volume management software not necessary
  - Central management of disks from a storage pool
- Data robustness
  - Data redundancy by a RAID configuration
  - Transaction file system model using the Copy-On-Write (COW) method
  - Detection and self-correction of invalid data by checksum

Scalability

ZFS is a 128-bit file system, and can manage data with virtually no limit on capacity and quantity.

	UFS	ZFS
Upper limit on file system size	16 TB	256 x 10^{15} ZB (256,000,000,000,000,000,000,000,000 TB)
Number of files that can be created in file system	Number of nodes (*)	No upper limit

* It varies depending on the file system size. If the size is 16 TB, the number is 15 x 10^{6}.

Manageability 1/2

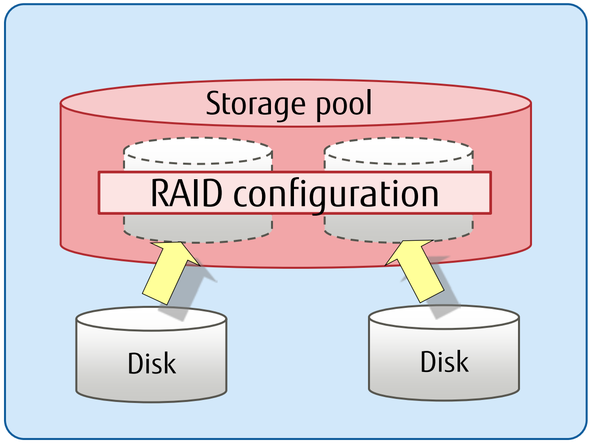
- Only two commands needed for ZFS management
  - zpool command

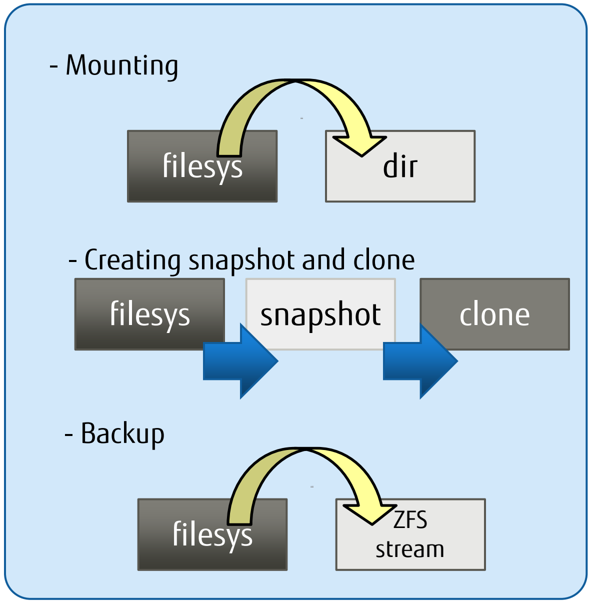
    <Main tasks>

    - Creating a storage pool from one or more disks
    - Setting up a redundant configuration of disks (RAID)
    - Checking data integrity (Scrub)
  - zfs command

    <Main tasks>

    - Creating or mounting a file system on the storage pool
    - Creating snapshots and clones
    - Backing up or restoring a file system

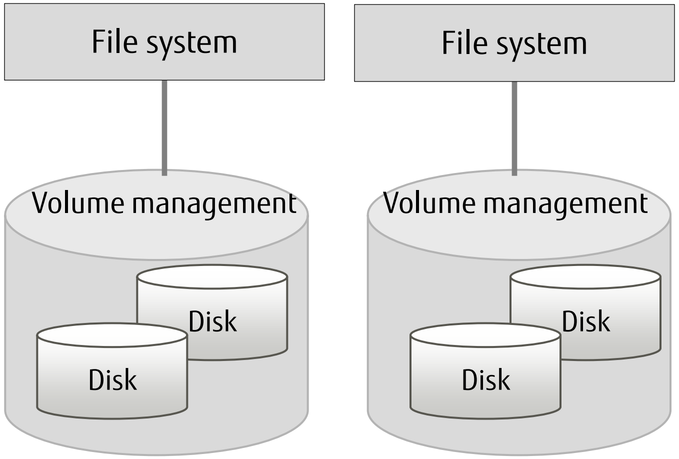




Manageability 2/2

Differences between UFS and ZFS file system configurations

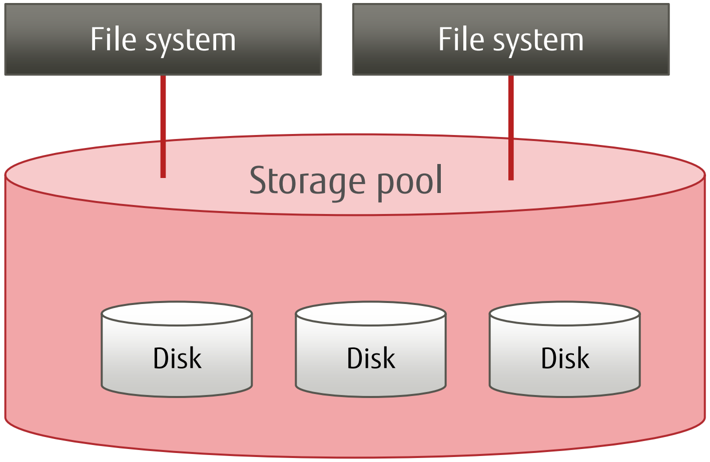
UFS



- **Settings by volume management software (SVM) are required** for each file system.
- To change the file system size, you need to **stop the OS** to perform backup/restore.
- The partition configuration and the file system size **must be determined** at the design time.
- A sudden system failure **may cause data inconsistency**.

-> For details on creating a UFS system, see "<<Reference>> Creating a UFS System."

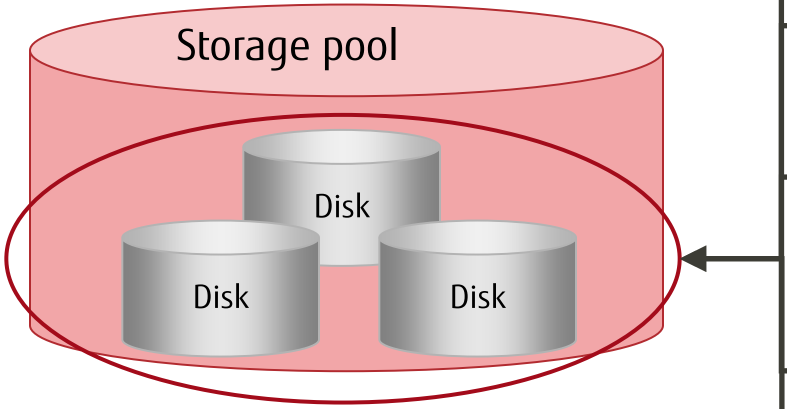
ZFS



- The storage pool centrally manages disks. **Volume management software is not necessary**.
- Disks can be expanded **with the system kept online**.
- The partition configuration and the file system size **do not have to be determined** at the design time.
- Even a sudden system failure **does not cause data consistency**.

Data Robustness - ZFS RAID Configuration -

- Possible to establish a redundant configuration of disks (RAID configuration) with the standard functions



RAID configuration specified when creating storage pool from multiple disks

RAID configurations supported by ZFS

RAID 0 (striping)
Non-redundant configuration but can be combined with RAID 1 (RAID 1+0)

RAID 1 (mirroring)
Can be mirror configuration with 3 or more disks

RAID-Z
Single parity
(similar to RAID 5, handles failure of 1 disk)

RAID-Z2
Double parity
(similar to RAID 6, handles failure of up to 2 disks)

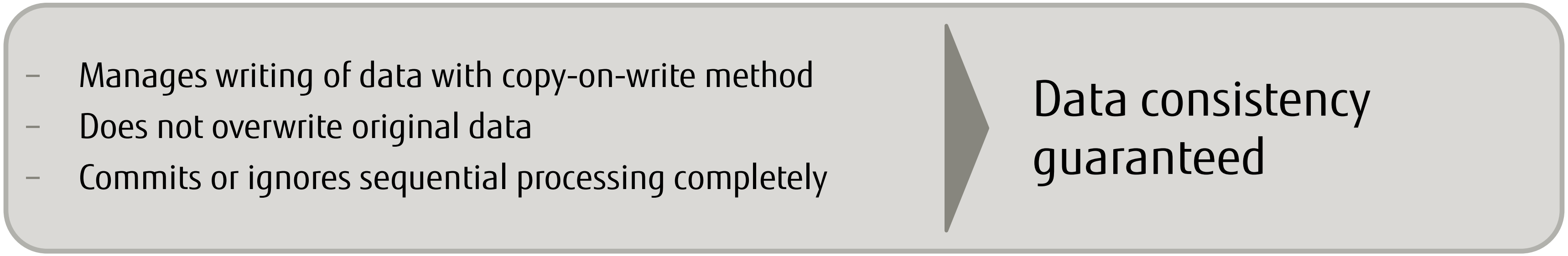
RAID-Z3
Triple parity
(handles failure of up to 3 disks)

- In conventional RAID 5 and RAID 6, if a failure occurs during the writing of parity code, an inconsistency may occur between the parity code and data. (**Write hole problem**)
- The above inconsistency does not occur in RAID-Z/RAID-Z2/RAID- Z3 of ZFS, enabling highly reliable RAID configurations.

Data Robustness - ZFS Mechanism -

- Mechanisms of a very robust file system

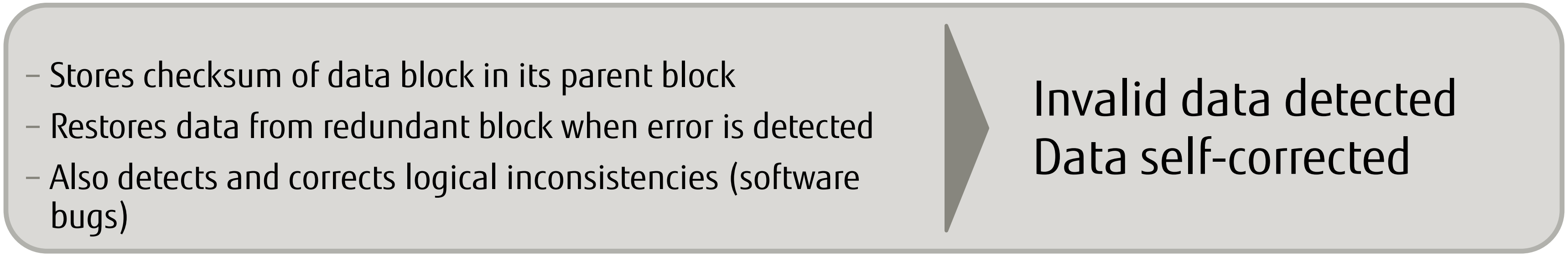
Transaction file system



- Manages writing of data with copy-on-write method
- Does not overwrite original data
- Commits or ignores sequential processing completely

Data consistency guaranteed

End-to-end checksum



- Stores checksum of data block in its parent block
- Restores data from redundant block when error is detected
- Also detects and corrects logical inconsistencies (software bugs)

Invalid data detected
Data self-corrected

<<Reference>> Creating a UFS System

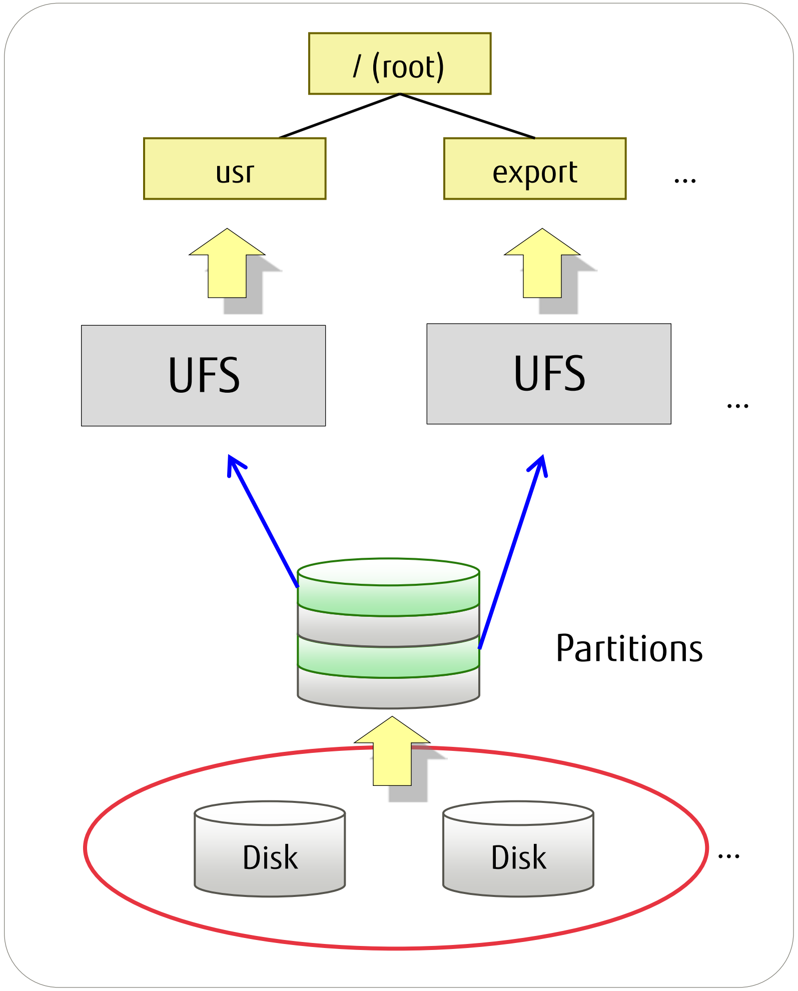
1-to-1 correspondence of UFS with the physical disk area

4. Mount file system (mount)
   - Assign an appropriate mount point.

3. Create file system (newfs, fsck)
   - Create a file system per partition (slice), and check the consistency there.

2. Create partition information (format)
   - Define partitions (capacity) used on the disks.

1. Add and recognize physical disks
   - Create a redundant configuration and logical disks using volume management software.



3. ZFS Reliability and Availability

This chapter describes the mechanisms and functions that support ZFS reliability and availability.

ZFS Reliability and Availability

- Mechanisms that support ZFS reliability and availability
  - Transaction file system
    - Mechanism using the copy-on-write method to ensure consistency during data update
  - Checksum
    - Mechanism of highly reliable data protection using end-to-end checksums
  - Scrub
    - Function that checks the integrity of all data in a storage pool

Transaction File System 1/2 - Ensuring Data Consistency at Data Update -

ZFS treats sequential write operations as one update unit (transaction).

Advantages of the transaction file system

No data inconsistency

- The copy-on-write method (*1) ensures that data updates are either a "complete success" or "complete failure."
  - *1 This method does not overwrite the source data when writing to a file.
    For details see "Transaction File System 2/2 - Copy-on-Write Method -."

Consistency check not required for the file system (Required for UFS)

- The file system is protected even if the system suddenly crashes.

Points to consider about the transaction file system

Asynchronous writing to disks

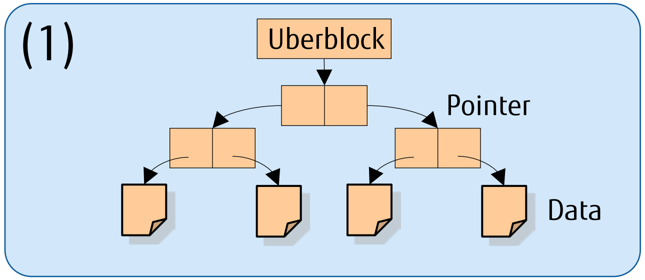
- The system writes to the disk (I/O) after the end of sequential processing, so a check of the actual disk capacity (by a command such as df or du) may find that it is different.

Transaction File System 2/2 - Copy-on-Write Method -

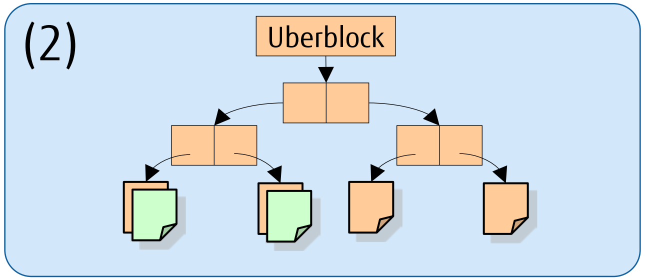
ZFS uses a method called "copy-on-write" to update data. With this method, a mechanism (transaction file system) that does not cause inconsistencies in the file system or the files themselves has been realized.

How data is updated in the copy-on-write method

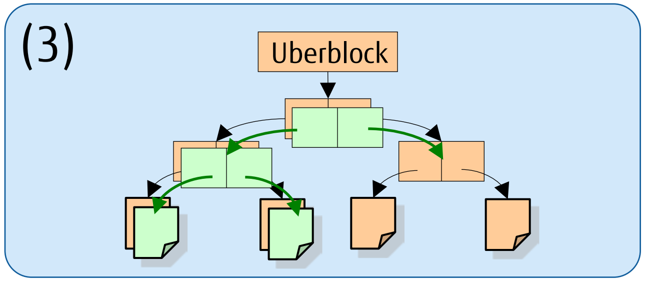
- When changing data written in a block, this method first creates a copy of the original data and then updates the copy (2). After updating the data, it updates data block links at an appropriate time according to the system (3), and replaces the uberblock (*1) with a new uberblock (4).



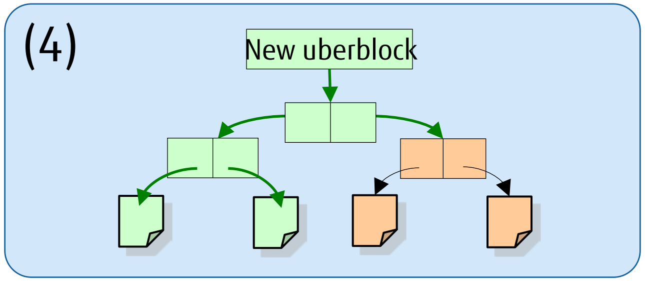
Initial block structure



Copying blocks and updating data



Copying blocks that contain pointers and linking to the new data



Replacing the uberblock

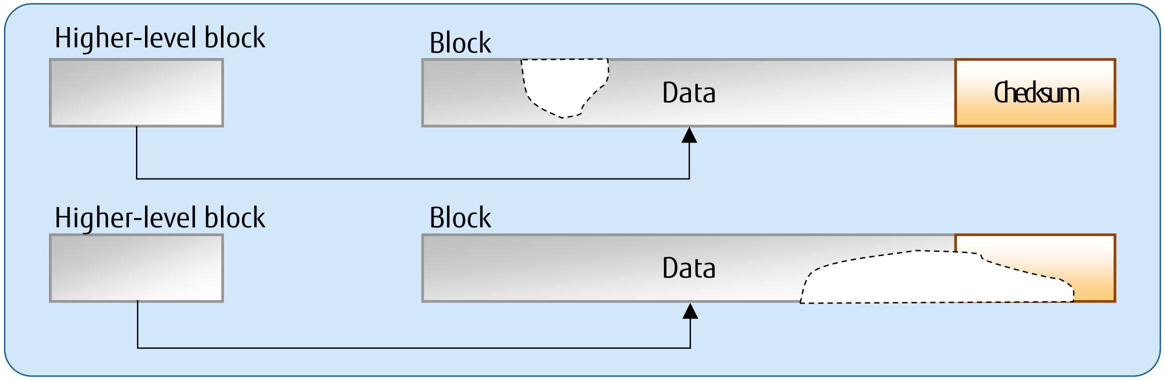
*1 The uberblock is the administrative area that stores the important setting information for the file system. Each file system has one uberblock.
The uberblock is equivalent to the superblock in UFS.

Checksum - Invalid Data Detection and Automatic Recovery -

ZFS improves the reliability of the checksum itself by saving the data and checksum in different blocks.

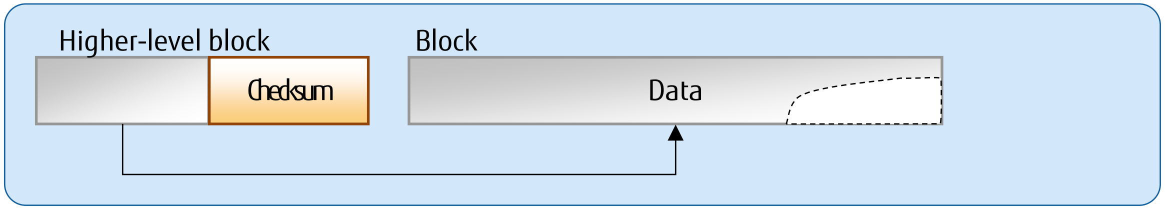
Data recovery method using checksum

Conventional file system



Data cannot be recovered when the data corruption range includes the checksum.

ZFS



The data and checksum are physically separated and read individually. The checksum itself can be recovered from the higher-level block.

- ZFS not only detects the read errors caused by a disk fault, but also detects and corrects logical inconsistencies caused by a software bug.
- When read or written, invalid data is detected by the checksum. When invalid data is detected, if the system is in a redundant configuration (RAID 1 (mirroring), RAID-Z, RAID-Z2, or RAID-Z3), the data is automatically recovered (self-healing) from replicated data.

Scrub - Explicit Data Inspection -

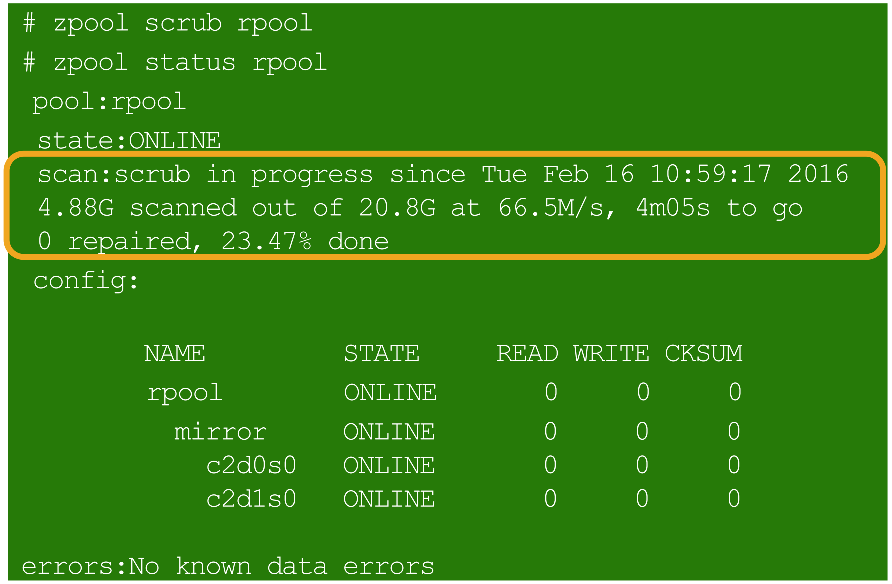
Basically, in ZFS, checksum is used for automatic recovery (self-healing) from corrupted data. However, there is also a scrub function for manual inspections of the file system.

Features of scrub

- It can detect and prevent errors before a hardware or software failure occurs.
- You can regularly check the disk integrity on the storage pool by executing scrub periodically.
- The priority of scrub is lower when there is I/O during scrubbing. (There is no effect on business.)

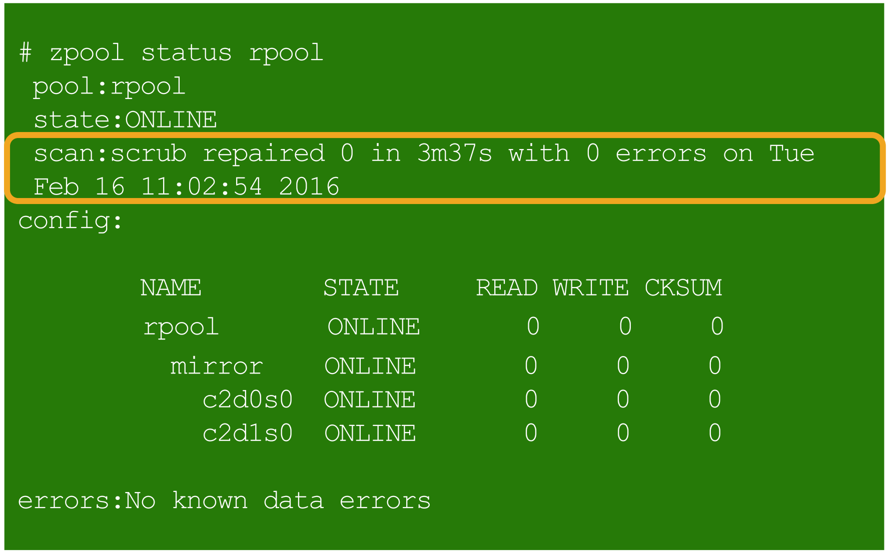
Scrub execution

✓ Immediately after scrub is executed



```
# zpool scrub rpool
# zpool status rpool
 pool:rpool
 state:ONLINE
 scan:scrub in progress since Tue Feb 16 10:59:17 2016
 4.88G scanned out of 20.8G at 66.5M/s, 4m05s to go
 0 repaired, 23.47% done
 config:

        NAME           STATE      READ WRITE CKSUM
        rpool          ONLINE        0     0     0
          mirror       ONLINE        0     0     0
            c2d0s0     ONLINE        0     0     0
            c2d1s0     ONLINE        0     0     0

errors:No known data errors
```

✓ After scrub is completed



```
# zpool status rpool
 pool:rpool
 state:ONLINE
 scan:scrub repaired 0 in 3m37s with 0 errors on Tue
 Feb 16 11:02:54 2016
config:

        NAME           STATE      READ WRITE CKSUM
        rpool          ONLINE        0     0     0
          mirror       ONLINE        0     0     0
            c2d0s0     ONLINE        0     0     0
            c2d1s0     ONLINE        0     0     0

errors:No known data errors
```

* Scrubbing goes through all data blocks in sequence to confirm that all the blocks can be read.

4. Storage Pool Configuration

This chapter describes the configurations of the storage pool called "root pool" and the storage pool simply called "storage pool." The root pool has the system area, and the storage pool has a user area.

Types of Storage Pool

- **Root pool**
  - The storage pool for the system area (for OS startup) is the root pool.
  - It stores the OS startup-related files and directories.
    (/, /bin, /sbin, /dev, /kernel, /etc, swap area, dump area, etc.)
- **Storage pool**
  - The storage pool for a user area (data area) is just called a "storage pool."
  - It mainly stores the data of user applications, etc.

Pool Name (Set Name in OS)	Intended Use	Configuration Timing	Configurable RAID Level	Dynamic Expansion
Root pool (rpool)	System area	At OS installation (*1)	RAID 1	Not possible
Storage pool (Can set any name)	User area	As required	RAID 0, RAID 1, RAID 1+0, RAID-Z, RAID-Z2, RAID-Z3	Possible

*1 The root pool is created during OS installation. However, redundancy (RAID 1) is configured manually after OS installation.

- Even though a part of a partition or a data file can be used as one of the devices composing a storage pool, from the perspective of configuration and management, we recommend using a whole disk.

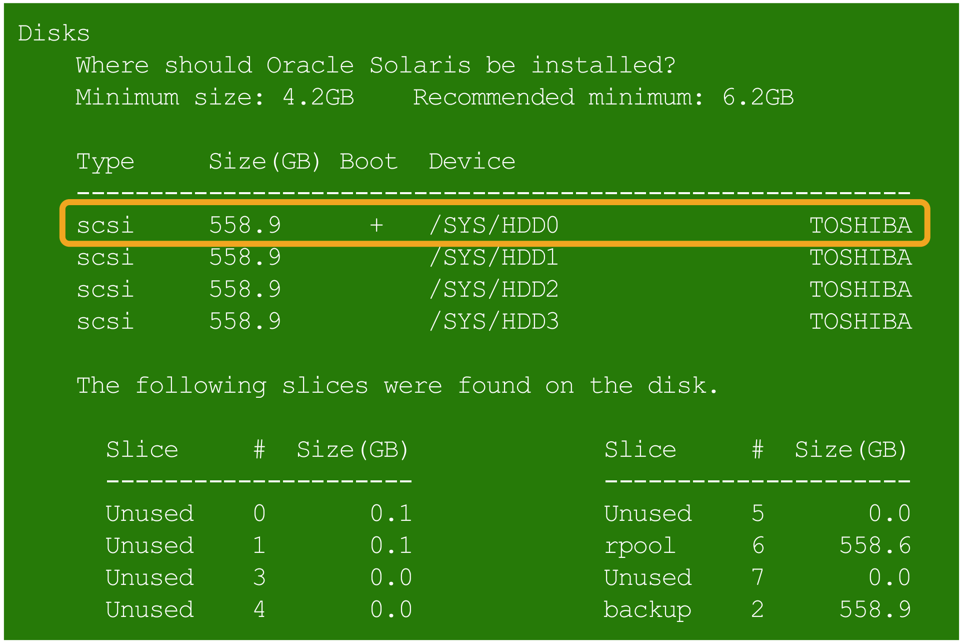
Root Pool Configuration 1/7

Root pool configuration at OS installation

- The default (at OS installation) configuration of the root pool in Solaris 11 has only one disk.
- Disk mirroring of the root pool is manually configured after OS installation.

  -> For the differences from Solaris 10, see "<<Reference>> Root Pool Configuration at OS Installation."

✓ Screen for OS installation



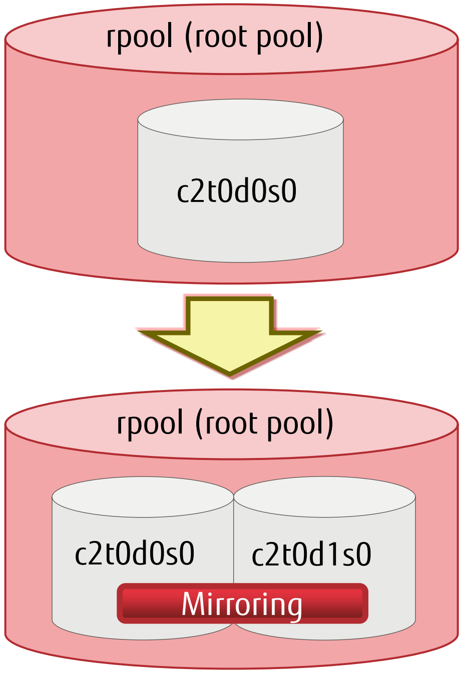
```
Disks
    Where should Oracle Solaris be installed?
    Minimum size: 4.2GB    Recommended minimum: 6.2GB

    Type      Size(GB)  Boot   Device
    ---------------------------------------------------------
    scsi      558.9      +     /SYS/HDD0                 TOSHIBA
    scsi      558.9            /SYS/HDD1                 TOSHIBA
    scsi      558.9            /SYS/HDD2                 TOSHIBA
    scsi      558.9            /SYS/HDD3                 TOSHIBA

    The following slices were found on the disk.

      Slice      #   Size(GB)             Slice      #   Size(GB)
      ---------------------               ---------------------
      Unused     0        0.1             Unused     5        0.0
      Unused     1        0.1             rpool      6      558.6
      Unused     3        0.0             Unused     7        0.0
      Unused     4        0.0             backup     2      558.9
```

At installation

Root pool configuration with single disk

After installation

Root pool configuration with multiple disks

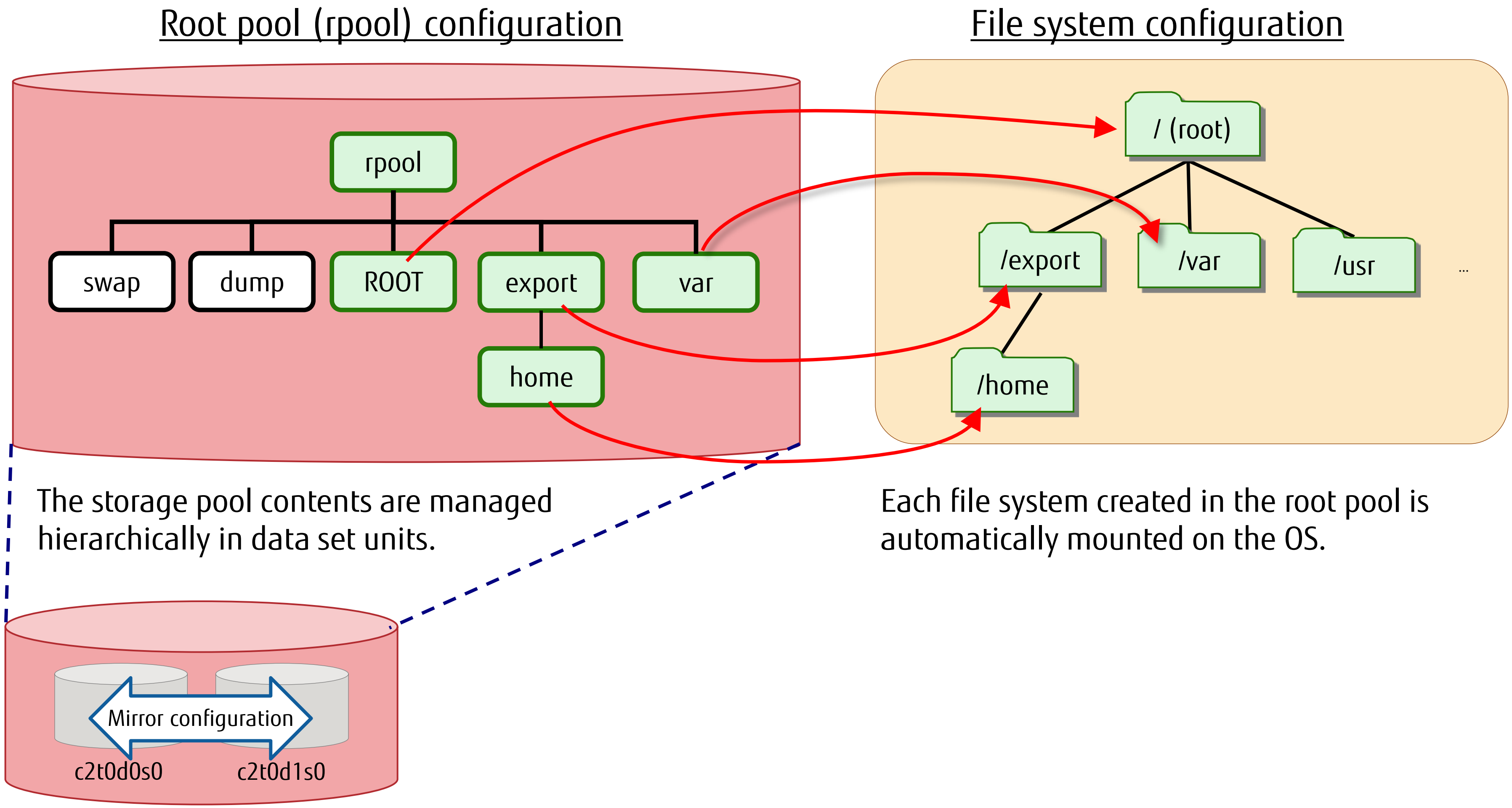


* A mirror configuration can have three or more disks.

- In consideration of failure occurrence, the recommended mirror configuration consists of disks under multiple controllers.

Root Pool Configuration 2/7

Root file system structure



The storage pool contents are managed hierarchically in data set units.

Each file system created in the root pool is automatically mounted on the OS.

– Data set:
It is a general term for a file system, volume, snapshot, or clone created in a storage pool.

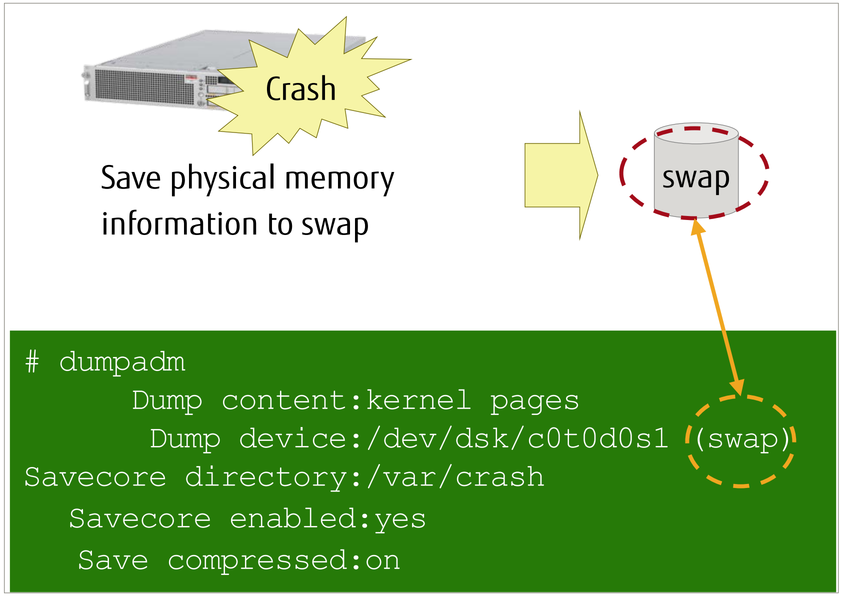
Root Pool Configuration 3/7

Swap area and dump area

- The swap and dump areas on ZFS are created in separate areas on a ZFS volume.

UFS root file system

Estimates of the swap and /var areas according to the dump level are required at installation.
(For a full dump: Area of at least the physical memory)

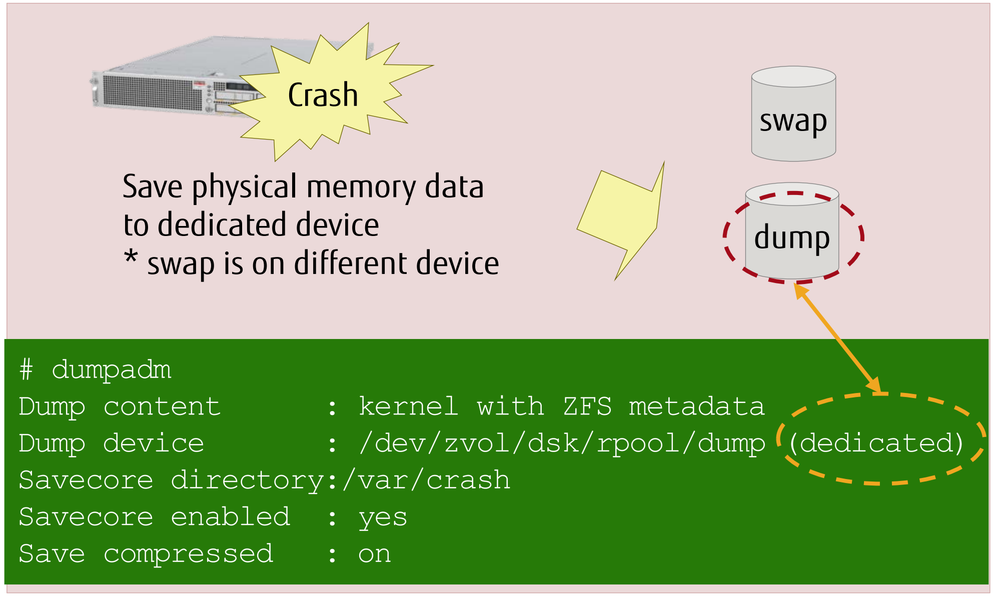


Save physical memory information to swap

```
# dumpadm
        Dump content:kernel pages
         Dump device:/dev/dsk/c0t0d0s1 (swap)
Savecore directory:/var/crash
  Savecore enabled:yes
   Save compressed:on
```

ZFS root file system

The swap and /var area sizes can easily be changed even if the dump level has been changed after installation.



Save physical memory data
to dedicated device
* swap is on different device

```
# dumpadm
Dump content       : kernel with ZFS metadata
Dump device        : /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory:/var/crash
Savecore enabled   : yes
Save compressed    : on
```

- ZFS volume:
  It is a data set on the storage pool used as a block device.
  The device file is located under /dev/zvol.

Root Pool Configuration 4/7

Default sizes of the swap and dump areas

The kernel calculates the swap and dump area sizes based on the physical memory size at initial OS installation. You can change these sizes after OS installation.

- Swap area
  - It is calculated as 1/4 of the physical memory size.
- Dump area
  - It is calculated as 1/2 of the physical memory size.

Swap and dump area sizes calculated by the kernel

System Type	Swap Area Size	Dump Area Size
Mounted physical memory of about 4 GB	1 GB	2 GB
Mounted physical memory of about 8 GB	2 GB	4 GB
Mounted physical memory of about 16 to 128 GB	4 GB	8 to 64 GB
Mounted physical memory of over 128 GB	1/4 of physical memory size	1/2 of physical memory size

- For the swap volume sizes and dump volume sizes calculated by the kernel, see the following:
  Managing File Systems in Oracle® Solaris 11.3 (Oracle)
  http://docs.oracle.com/cd/E53394_01/html/E54785/index.html

Root Pool Configuration 5/7

FUJITSU

Changing the swap and dump area sizes

You can easily and dynamically change the ZFS swap and dump area sizes by just changing the value of the volsize property.

Swap area

Delete the current swap device, and change the volsize property value. Then, rebuild the swap device.

```
# swap -d /dev/zvol/dsk/rpool/swap     <- Delete
# zfs set volsize=3G rpool/swap        <- Change
# /sbin/swapadd                        <- Rebuild
# swap -l                              <- Check
swapfile                  dev      swaplo   blocks     free
/dev/zvol/dsk/rpool/swap 303,1         16  6291440  6291440
```

You can add a swap device like in the conventional system (UFS).

```
# zfs create -V 2G rpool/swap1          <- Create area
# swap -a /dev/zvol/dsk/rpool/swap1     <- Add
# swap -l                               <- Check
swapfile                   dev      swaplo   blocks     free
/dev/zvol/dsk/rpool/swap   303,1        16  6291440  6291440
/dev/zvol/dsk/rpool/swap1  303,3        16  4194288  4194288
```

Dump area

You can change the size by just changing the volsize property value.

```
# zfs set volsize=2G rpool/dump    <- Change
# zfs get volsize rpool/dump       <- Check
NAME         PROPERTY   VALUE   SOURCE
rpool/dump   volsize    2G      local
```

The dump device capacity is checked when the dump level is changed, so if the dump device value is inadequate, change it.

Root Pool Configuration 6/7

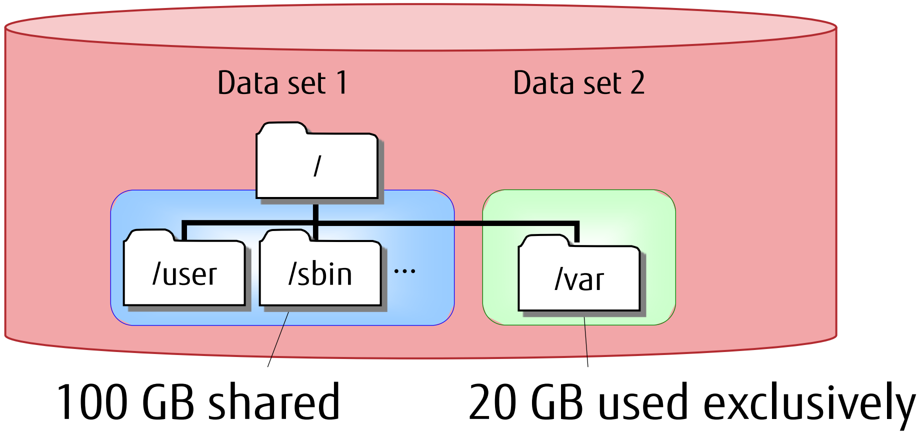
/var area setting

- At OS installation, the default setting creates the /var area in a data set different from the root file system.

Advantages of placing the /var area in a separate data set

- You can create a snapshot in units of the /var area data set.
- You can set an upper limit on use of the area for the /var area data set. (A separate property can be set.)
- It can be used for a log operation policy (on log collection for the /var area, centralized backup, etc.).

/var area setting example



- In Solaris 10, the configuration of a separate data set must be selected at installation, whereas the default settings since Solaris 11 use a separate data set for the /var area.

Root Pool Configuration 7/7

FUJITSU

rpool/VARSHARE file system

- Installation of Solaris 11 automatically creates the rpool/VARSHARE file system under the root pool (rpool) and mounts it to /var/share.

Role of VARSHARE

- Multiple BEs that share the /VARSHARE file system will reduce the capacity of the /var directory required by all BEs.

✓ Check the file system.

```
# zfs list
NAME             USED   AVAIL  REFER  MOUNTPOINT
-- <<Omitted>> --
rpool/VARSHARE   2.54M  467G   2.45M  /var/share
```

✓ Check /var/share.

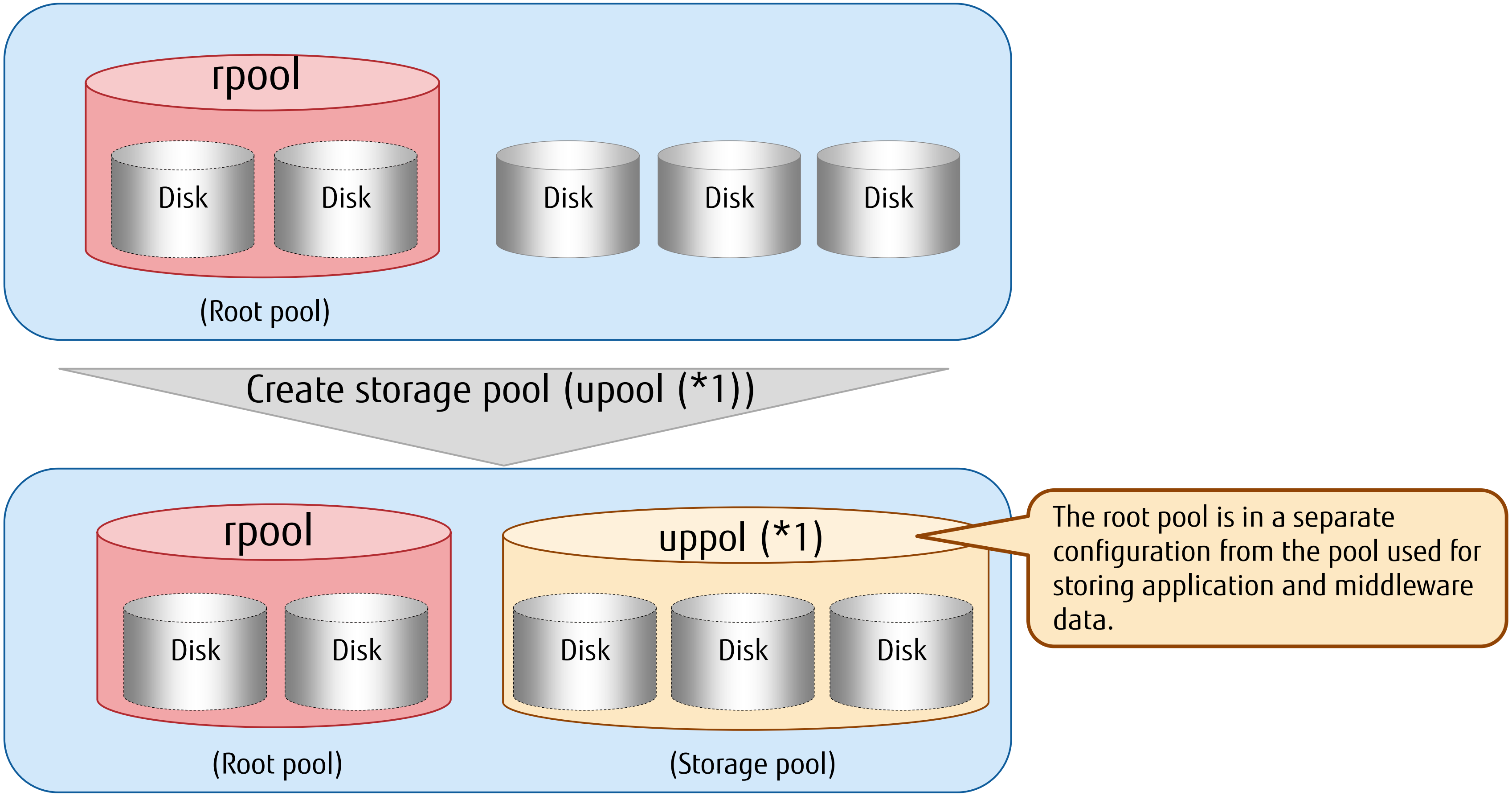
```
# ls -l /var/share
total 27
drwxr-xr-x   2 root     root          5 Nov 14 20:47 audit
drwxr-xr-x   3 root     root          3 Nov 14 03:51 compliance
drwxr-xr-x   2 root     sys           2 Nov 14 03:50 cores
drwx------   2 root     sys           2 Nov 14 03:50 crash
drwx------   4 root     root          9 Nov 14 20:47
ldomsmanager
drwxrwxrwt   3 root     mail          4 Nov 14 22:59 mail
drwxr-xr-x   2 root     bin           2 Nov 14 03:50 nfs
drwxr-xr-x   3 root     root          3 Nov 14 03:51 pkg
drwxr-xr-x   4 daemon   daemon        5 Nov 14 21:17 statmon
```

- BE is the abbreviation for Boot Environment, which is a new function of Solaris 11.
  For details on BE, see the *Oracle Solaris 11* manuals.

Storage Pool Configuration 1/2

Creating a storage pool (user area)

For the storage pool created for a user area to store application and other data, we recommend using disks separate from those in the root pool.


rpool
Disk
Disk
Disk
Disk
Disk
(Root pool)
Create storage pool (upool (*1))
rpool
Disk
Disk
(Root pool)
uppol (*1)
Disk
Disk
Disk
(Storage pool)
The root pool is in a separate configuration from the pool used for storing application and middleware data.

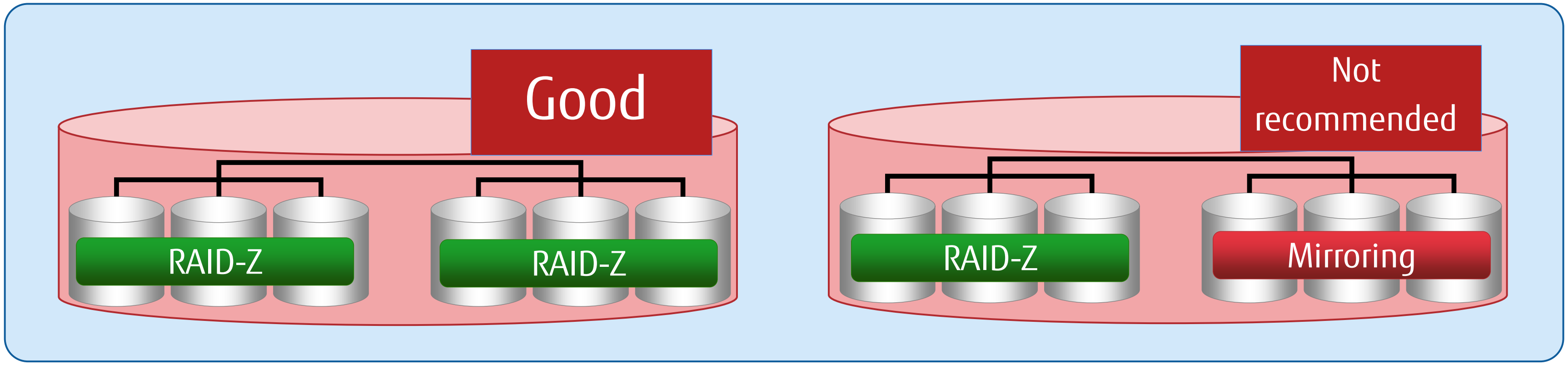
*1 The root pool with the system area (for OS startup) has a fixed name, "rpool", but you can set any name for the storage pool of a user area.

Storage Pool Configuration 2/2

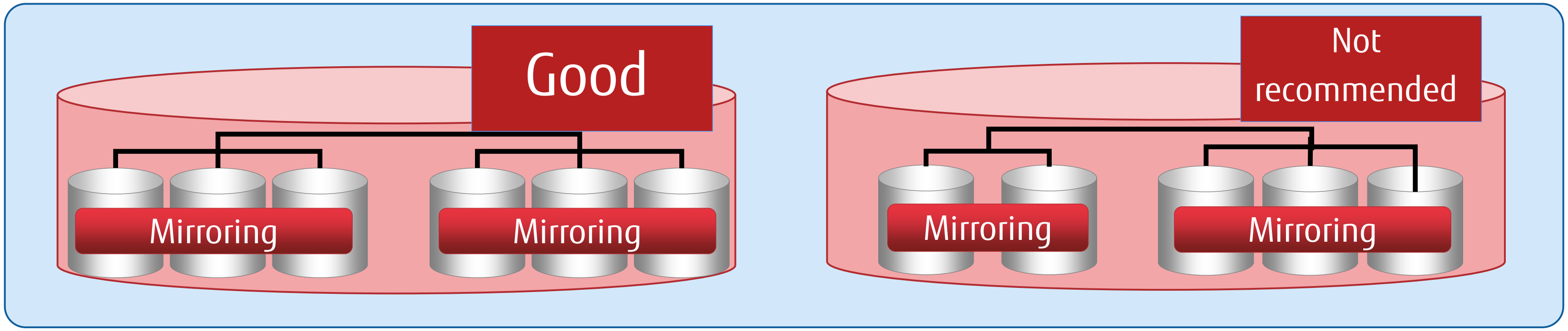
Recommended storage pool configurations

A storage pool can consist of multiple disk groups. In consideration of reliability and performance, we recommend using the same redundancy level for the disk groups.

For the same redundancy level within the storage pool



For the same disk group configuration (capacity of each disk, number of disks, etc.)

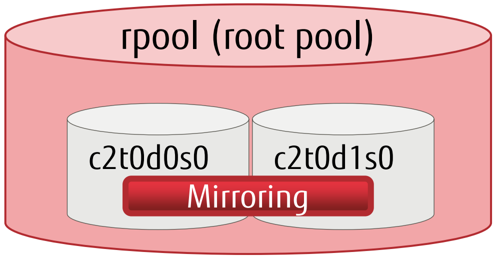


<<Reference>> Root Pool Configuration at OS Installation

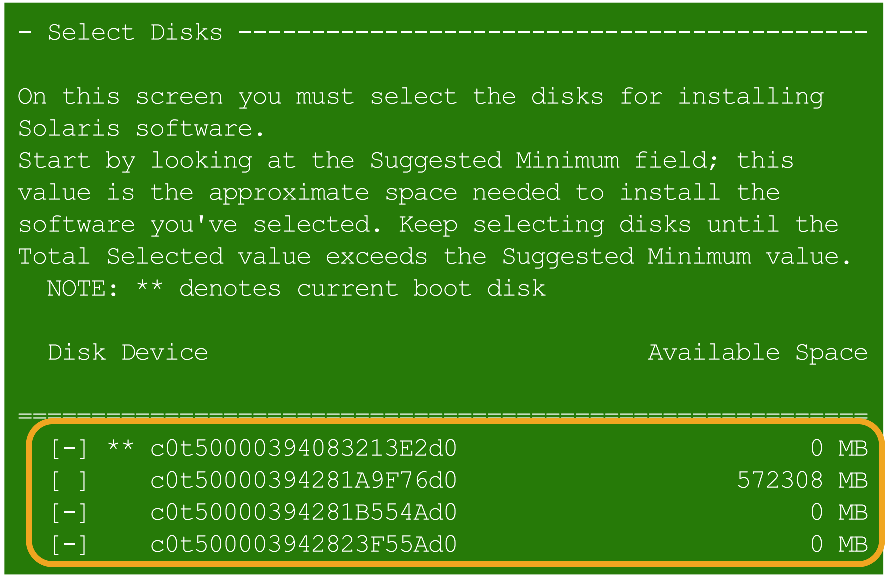
The initial root pool configuration (at OS installation) is different between Solaris 10 and Solaris 11.

Solaris 10

- ✓ Root pool configuration with multiple disks



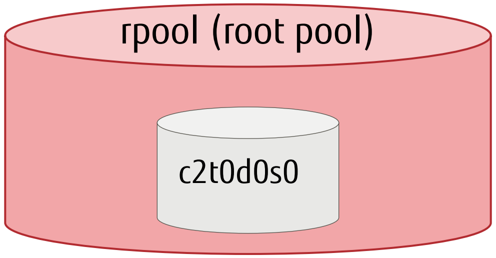
The mirror configuration is available at OS installation.



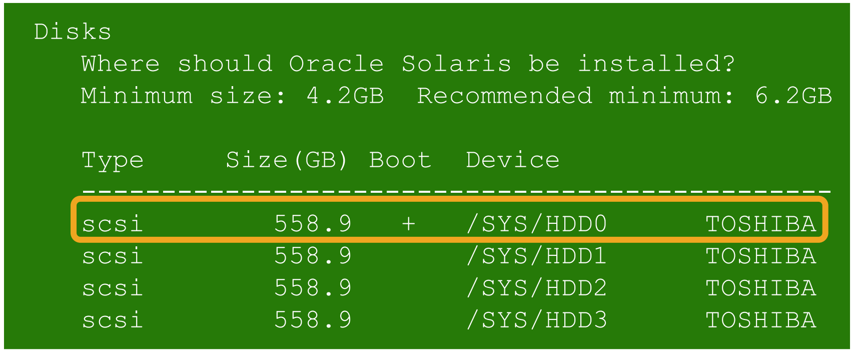
```
- Select Disks ---------------------------------------------

On this screen you must select the disks for installing
Solaris software.
Start by looking at the Suggested Minimum field; this
value is the approximate space needed to install the
software you've selected. Keep selecting disks until the
Total Selected value exceeds the Suggested Minimum value.
  NOTE: ** denotes current boot disk

  Disk Device                                Available Space

==========================================================
  [-] ** c0t50000394083213E2d0                          0 MB
  [ ]    c0t50000394281A9F76d0                     572308 MB
  [-]    c0t50000394281B554Ad0                          0 MB
  [-]    c0t500003942823F55Ad0                          0 MB
```

Solaris 11

- ✓ Root pool configuration with single disk



The mirror configuration is unavailable at OS installation, so it needs to be set up as required after installation.



```
Disks
   Where should Oracle Solaris be installed?
   Minimum size: 4.2GB  Recommended minimum: 6.2GB

   Type      Size(GB)  Boot  Device
   -----------------------------------------------
   scsi          558.9    +   /SYS/HDD0        TOSHIBA
   scsi          558.9        /SYS/HDD1        TOSHIBA
   scsi          558.9        /SYS/HDD2        TOSHIBA
   scsi          558.9        /SYS/HDD3        TOSHIBA
```

5. File System Management

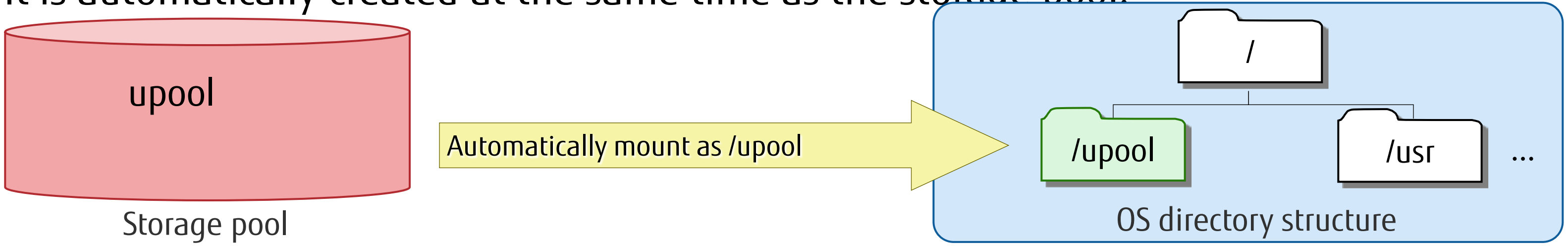
This chapter describes how a file system is created and mounted and the functions of properties that control file system behavior.

Creating a File System

A file system is managed hierarchically.

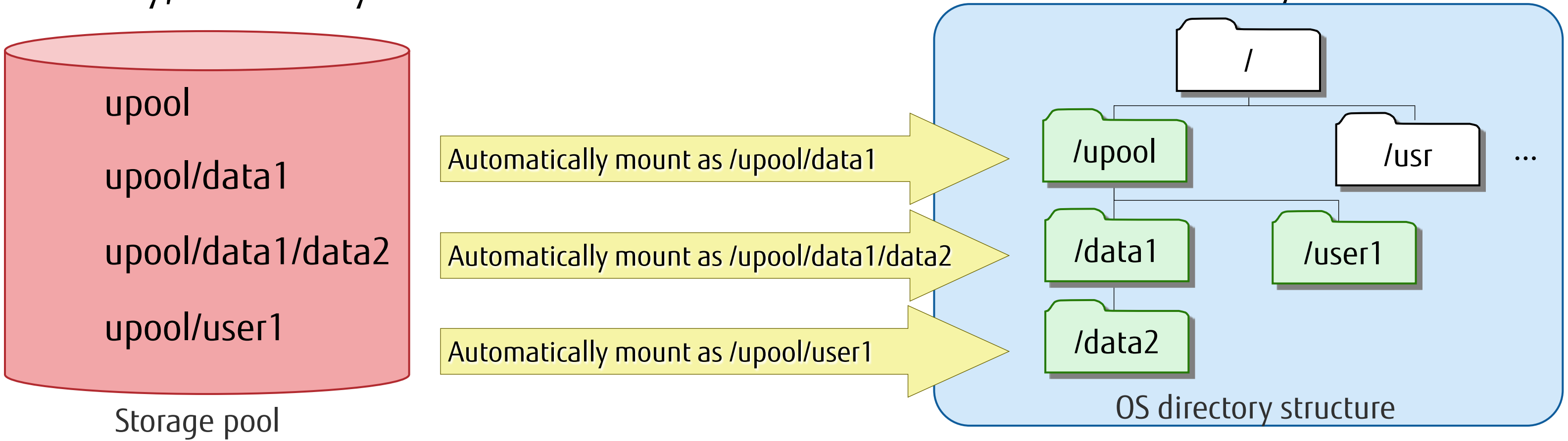
Top-level file system

- It is automatically created at the same time as the storage pool.



Lower-level file system

- Unlike the top-level file system, it is created manually.
- Normally, a directory is created with the same name as the created file system.


upool
upool/data1
upool/data1/data2
upool/user1
Storage pool
Automatically mount as /upool/data1
Automatically mount as /upool/data1/data2
Automatically mount as /upool/user1
/
/upool
/usr
…
/data1
/user1
/data2
OS directory structure

– Each file system has a shared disk area with its higher-level file system.

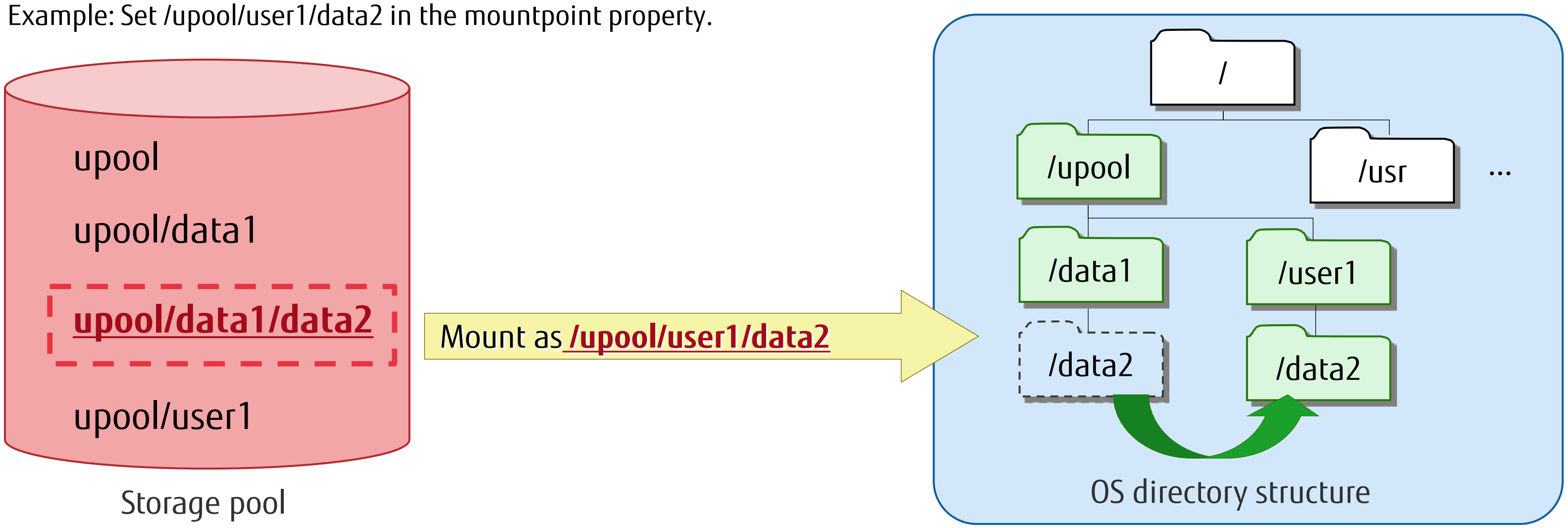
Mounting a File System

Basic behavior of ZFS mounting

- Creating a file system also creates a mount point on the path with the same file system name, and the file system is automatically mounted there.
- Deleting a file system automatically unmounts it.
- You can change the mount point by changing the value of the mountpoint property.
- Mounting can also be done in the conventional (UFS) manner (legacy mount).

Changing a mount point with the mountpoint property

Example: Set /upool/user1/data2 in the mountpoint property.



- In the NFS environment, when a new file system is created on an NFS server, an NFS client automatically recognizes that the file system has been created under a mounted file system. The new system is automatically mounted (by ZFS mirror mount).
-> For details on ZFS mirror mount, see "<<Reference>> ZFS Mirror Mount."

File System Properties 1/2

- ZFS properties
  - You can control file system behavior, such as for mounting, configuring sharing, and limiting the disk usage capacity, by setting various property values in ZFS.
- Types of properties
  - Native properties
    - Native properties control ZFS behavior. They are categorized into the following two types:
      - Read-only native properties (not inherited)
      - Configurable native properties (inherited except for assignment limit [quota] and reservation [reservation])
  - User properties
    - These properties do not affect ZFS behavior.
      The user can set arbitrary values to describe intended uses and annotations of the file system.

> – This document does not cover user properties in detail. The word "property" in the descriptions refers to a native property.

File System Properties 2/2

Inheritance of properties

- A lower-level file system inherits the values of configurable native properties of the parent file system.
  However, the quota property (assignment limit) and reservation property (reservation) are exceptions and not inherited.
- Properties are inherited at the following timing:
  - When a property value for the parent file system is set
  - When a new file system is created under a file system that has the set properties
- You can check whether each property of a data set is inherited.

* The following example checks the compression property under upool. You can check the inheritance status of a property by looking at the SOURCE column values in the command output.

```
# zfs get -r compression upool
NAME               PROPERTY       VALUE    SOURCE
upool              compression    off      default
upool/test         compression    gzip     local
upool/test/data    compression    gzip     inherited from upool/test
```

Value of SOURCE	Description
local	Shows that the property has been explicitly set for the data set.
inherited from *data_name*	Shows the parent [*data_name*] from which the property has been inherited.
default	Shows that the property has not been set (a state other than the above).

Representative Properties 1/7

The next pages present the functions of the following properties representative of native properties.

- Changing a mount point
  - mountpoint property
- Configuring NFS sharing
  - share.nfs property
- File system assignment limit and reservation
  - quota property
  - reservation property
- User/Group assignment limit
  - defaultuserquota property
  - defaultgroupquota property
  - userquota@user property
  - groupquota@group property
- Detection and elimination of data duplication
  - dedup property
- Data encryption
  - encryption property

Representative Properties 2/7

mountpoint property (changing a mount point)

You can change a mount point. You can also make the setting using legacy mount (set in the same way as in UFS).

Setting example

- Specify a mount point.

```
# zfs set mountpoint=/test upool/test
```

- Use legacy mount.

```
# zfs set mountpoint=legacy upool/test
```

Legacy mount method

- Use it to manage mounting of the ZFS file system in the same way as the UFS file system.
- Commands (mount/unmount) to manage mounting are needed because automatic mounting by ZFS is disabled.
  - Example: Mount rz2pool/data2 to /zfs/fs2.

(1) Set the mountpoint property for legacy mount.
(2) Create a mount point.

```
# mkdir /zfs/fs2
```

(3) Execute the mount command.

```
# mount -F zfs rz2pool/data2 /zfs/fs2
```

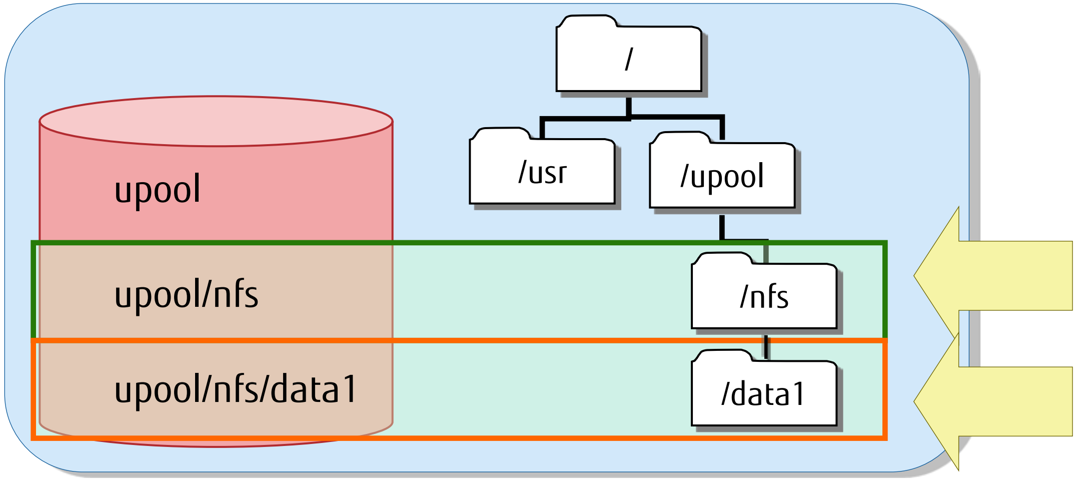
* You can enable automatic mounting for the OS startup time by editing the /etc/vfstab file.
(Example)

```
#device         device     mount      FS     fsck    mount      mount
#to mount       to fsck    point      type   pass    at boot    options
rz2pool/data2   -          /zfs/fs2    zfs    -        yes        -
```

Representative Properties 3/7

share.nfs property (configuring NFS sharing)

- With sharing configured, NFS can share file system data with another server.
- The share setting of a file system configured for sharing is even inherited by its lower-level file systems.



Share property setting

Property inheritance

* In Solaris 11.1 and later, the share.nfs property is used instead of the sharenfs property.
* In Solaris 11 11/11, the method of enabling the share setting is to create a shared file system by using the zfs set share command and set the sharenfs property.

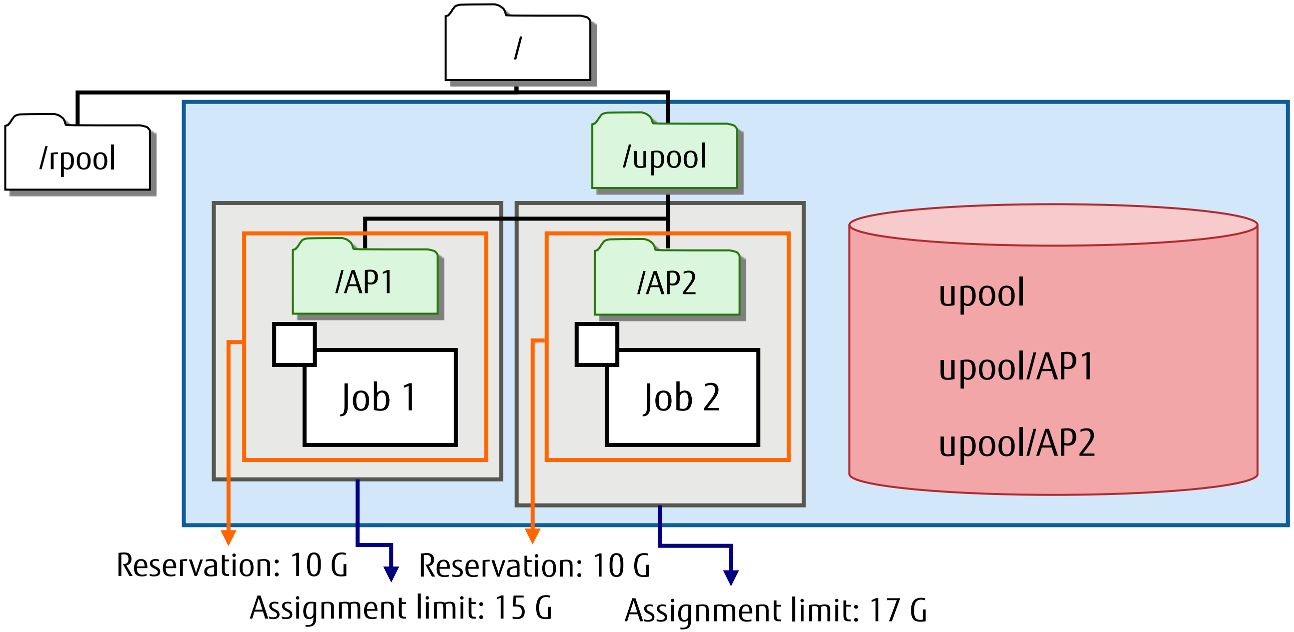
Notes

- You do not need to edit the conventional share configuration file (/etc/dfs/dfstab).
- The share setting is automatically canceled when the file system is deleted.
- Use NFS version v4 (NFSv4).

Representative Properties 4/7

- quota property and reservation property (file system assignment limit and reservation)
  - With the quota property, you can set the maximum capacity (assignment limit) available in the file system
  - With the reservation property, you can secure (reserve) the capacity always available to the file system.

* These properties are not set by default. There is neither a limit on the capacity available in each file system nor a reserved capacity.



– Unlike with UFS partitions, in ZFS, you can easily change the available capacity by changing the assignment limit and reservation values.

Representative Properties 5/7

defaultuserquota property, defaultgroupquota property, userquota@user property, and groupquota@group property (user/group assignment limit)

- You can set the default upper limit on usage capacity to apply to users/groups, and an upper limit on usage capacity for a specific user/group.

Property	Description
defaultuserquota=size | none	Default assignment limit on users
defaultgroupquota=size | none	Default assignment limit on groups
userquota@user= size | none | default	Assignment limit on specific user
groupquota@group= size | none | default	Assignment limit on specific group

* "none" (no assignment limit) releases the assignment limit.
* "default" sets the default value for the assignment limit.

Setting example

- Set the default upper limit on the capacity available to one user to 25 GB.

```
# zfs set defaultuserquota=25gb upool/nfs
```

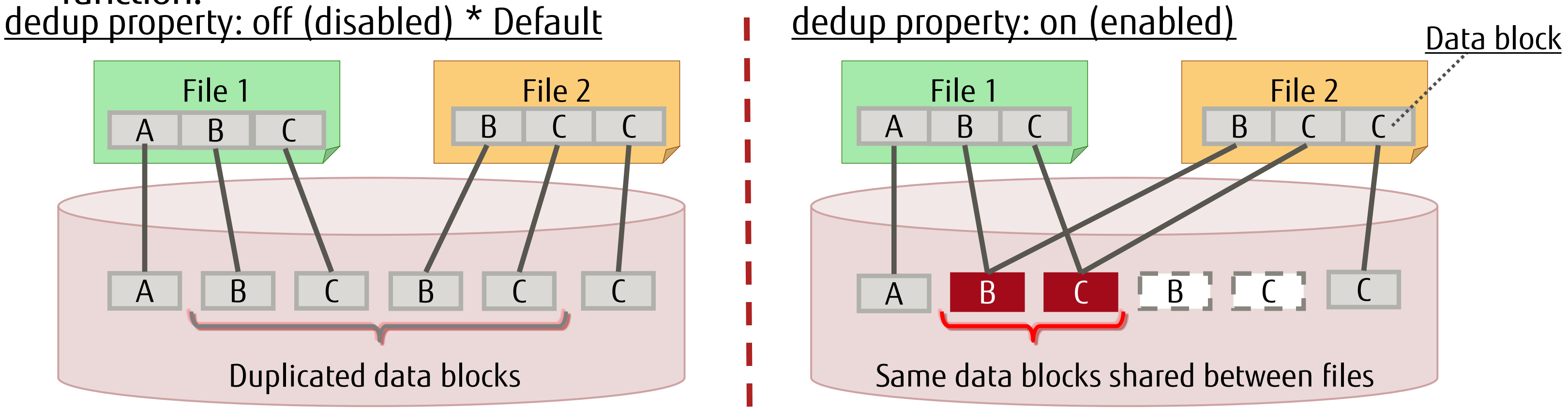
- Set the upper limit on the capacity available to the user user1 to 50 GB.

```
# zfs set userquota@user1=50gb upool/nfs
```

Representative Properties 6/7

dedup property (detection and elimination of data duplication)

- Use it to detect and eliminate duplicated data areas (data blocks) among files.
- The "on" setting as the dedup property value enables the duplication detection and elimination function.



dedup property: off (disabled) * Default
File 1
A
B
C
File 2
B
C
C
A
B
C
B
C
C
Duplicated data blocks
dedup property: on (enabled)
Data block
File 1
A
B
C
File 2
B
C
C
A
B
C
B
C
C
Same data blocks shared between files

* The per-block duplication detection and elimination function eliminates, while online (on the fly), a block in a file if it has the same contents as a block stored in another file.

Setting example

- Enable the duplication detection and elimination function in tank/home.

```
# zfs set dedup=on tank/home
```

Notes

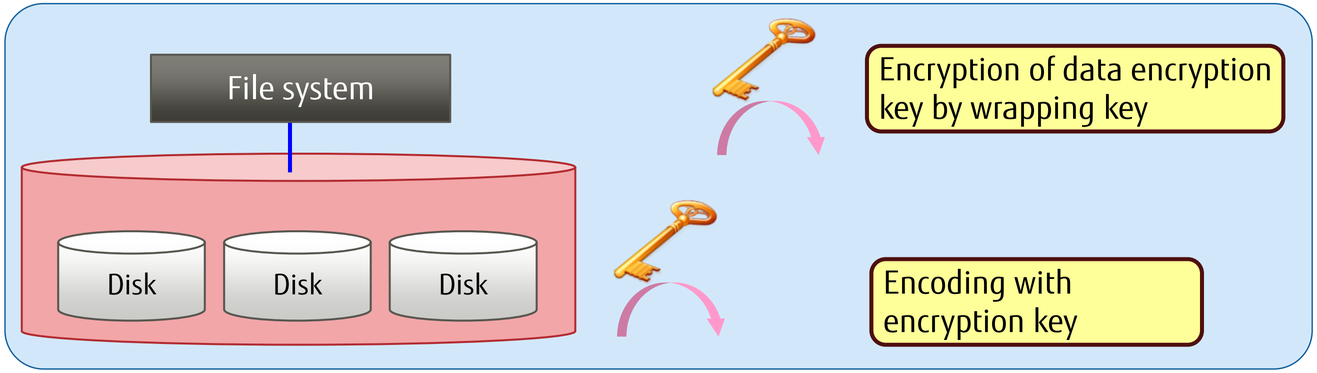
- Duplication is determined by checksum (SHA256).
- You can use this function at the same time as compression or encryption.

- This property enables effective utilization of the disk capacity at virtualization or server backup, and can improve the efficiency of storage use.

Representative Properties 7/7

encryption property (data encryption)

- Use it to encode the file system with an encryption key to protect the data contents.



File system
Disk
Disk
Disk
Encryption of data encryption key by wrapping key
Encoding with encryption key

* An encryption policy is set for each ZFS data set.

Setting example

- Encrypt fj/home/user. Enter an encryption key of eight characters or more.

```
# zfs create -o encryption=on fj/home/user
Enter passphrase for 'fj/home/user' : XXXXXXXX (<- 8 characters or more)
Enter again : XXXXXXXX
```

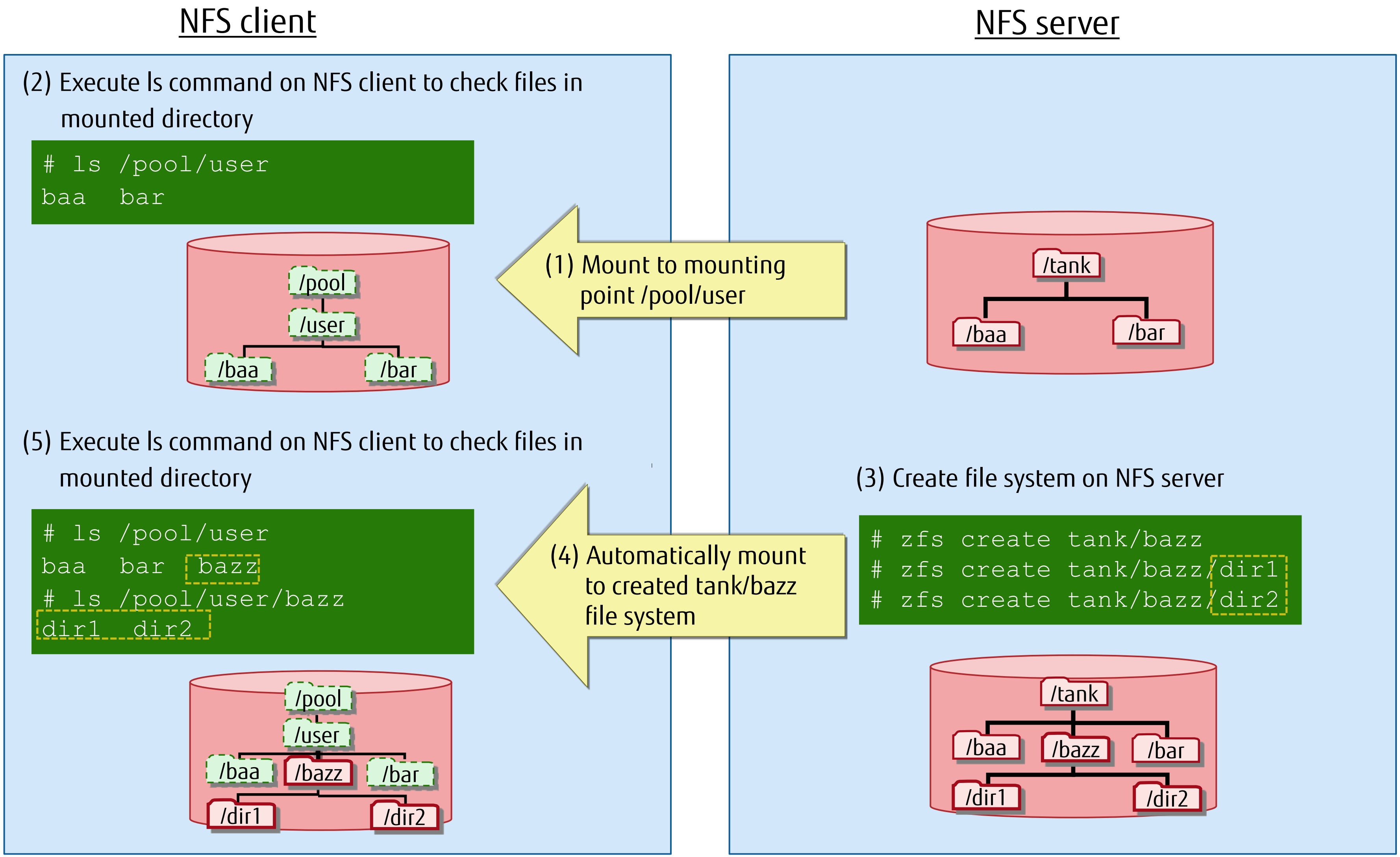
Notes

- Specify an encryption attribute value when creating a data set.
- AES-128/192/256 is supported.
- Encryption can be set when a ZFS file system is created, but the encryption setting cannot be changed.

- ZFS encrypts the encryption key with a wrapping key in addition to the encryption of the file system, so its security function is stronger than that of UFS.

<<Reference>> ZFS Mirror Mount

When a new file system is created on an NFS server, an NFS client automatically recognizes that the file system has been created under a mounted file system. Then, this function automatically mounts the new file system.



NFS client

NFS server

(2) Execute ls command on NFS client to check files in mounted directory

```
# ls /pool/user
baa  bar
```

(1) Mount to mounting point /pool/user

(5) Execute ls command on NFS client to check files in mounted directory

```
# ls /pool/user
baa  bar  bazz
# ls /pool/user/bazz
dir1  dir2
```

(4) Automatically mount to created tank/bazz file system

(3) Create file system on NFS server

```
# zfs create tank/bazz
# zfs create tank/bazz/dir1
# zfs create tank/bazz/dir2
```

6. ZFS Functions

This chapter describes the snapshot, backup/restore, and other functions used for ZFS operation management.

List of Functions Presented Here

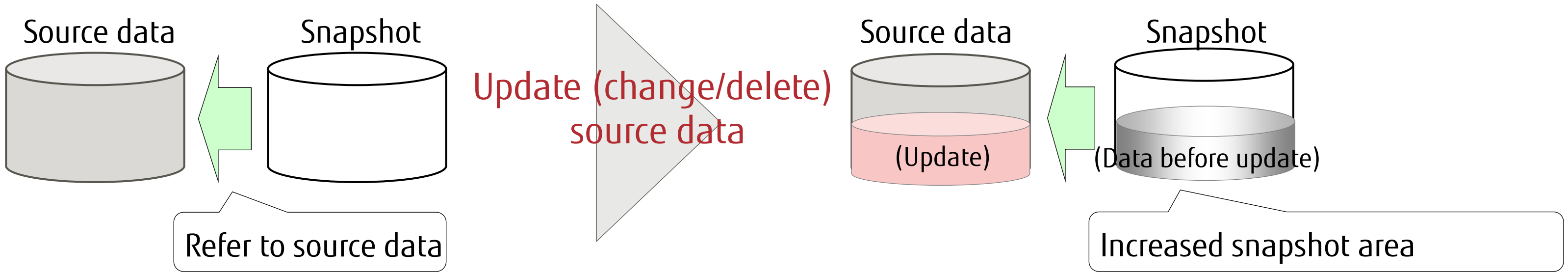
The next pages present the ZFS basic functions used for operations.

- Snapshot
  - The function makes a read-only copy of a data set.
- Rollback
  - This file system restore function uses snapshots.
- Clone
  - The function makes a data set copy utilized by the snapshot function.
- Backup/Restore
  - The function can back up/restore a file system while online.
- Release mirror disk
  - The function releases one disk from a storage pool to create a storage pool that holds the same data.
- Migrate data from UFS to ZFS
  - The function can migrate a UFS file system to a ZFS environment.

Snapshot 1/5

What is a snapshot?

- A snapshot is a read-only copy of a data set (file system or volume). A snapshot is created instantly, and immediately after being created, it does not use the area in the storage pool.
- When data in the source data set of a snapshot is updated (changed/deleted), the snapshot area increases by the volume of the pre-update data.



* Note that, for a snapshot that has been created, the total disk area does not decline even when the source data is deleted.

Notes

- Theoretically, snapshots can be created for up to 2^{64} generations.
- A snapshot uses the disk area of the storage pool of the source file system.
- You can create a snapshot of a higher-level data set and a snapshot of a lower-level data set simultaneously.
- If the source for creating a snapshot contains the ZFS volume, the available capacity must satisfy the following formula:

  Available capacity of the parent pool > Total usage of the ZFS volume contained in the snapshot source

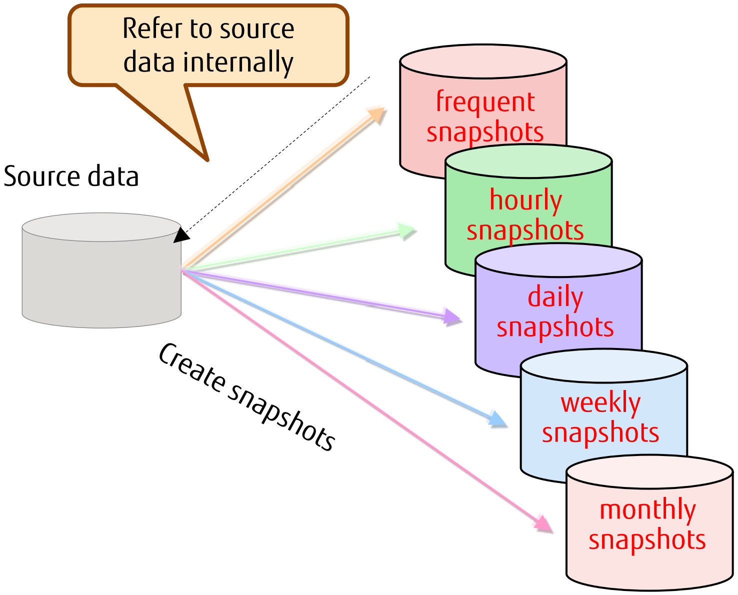
Snapshot 2/5

- Functions that work together with snapshots
  - Rollback to the snapshot creation time
    - You can restore a file system to the state at the time that a snapshot was created.
    - All changes made after the snapshot was created are canceled.
  - Creating a clone
    - For use in a test or other purposes, you can make an environment that has the same conditions as when a snapshot was created.
    - Like a snapshot, a clone is created instantly. When created, a clone does not consume storage pool area.
  - Backup/Restore
    - You can create a stream from a snapshot and send it to a file (backup).
    - Upon receiving a sent stream, you can restore the file system from the stream (restore).

Snapshot 3/5

Automatic snapshot

- You can automatically create a snapshot by using the time slider service.
- You can also view the automatically created snapshots and perform restore.



Criteria of creating snapshot

	Creation Timing	Quantity That Can be Retained
frequent snapshots	Every 15 minutes	4
hourly snapshots	Every hour	24
daily snapshots	Every day	31
weekly snapshots	Every week	7
monthly snapshots	Every month	12

Notes

- From the older automatically created snapshots, snapshots are deleted based on a preset percentage of file system usage.
- The time slider service is disabled by default.

– We recommend enabling the time slider service for a file system that has frequently updated data.

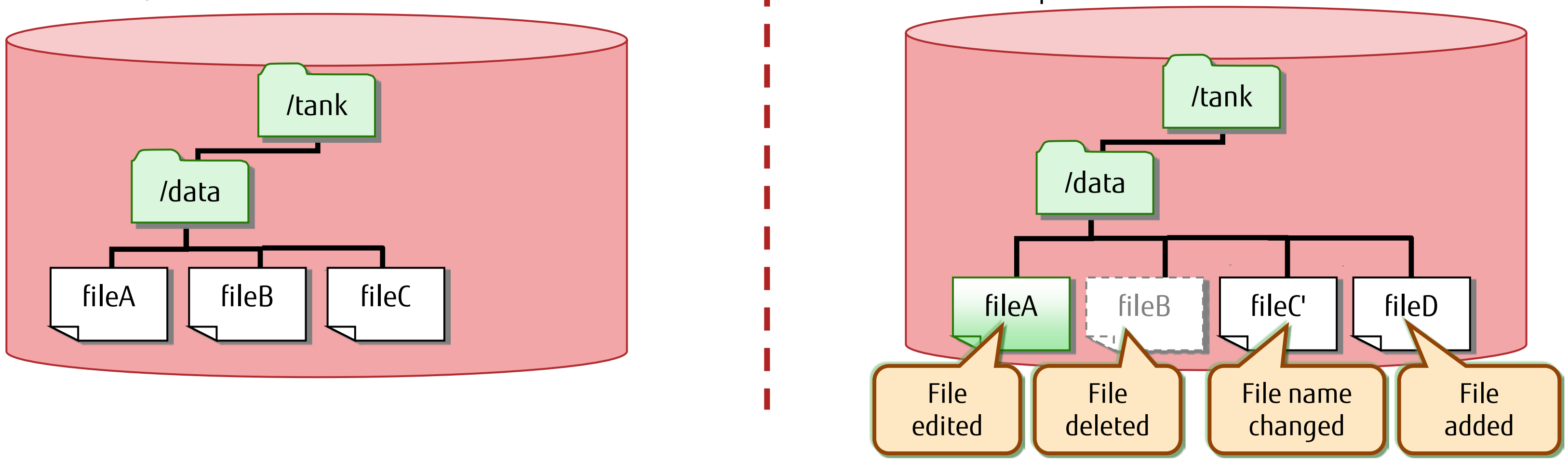
Snapshot 4/5

Snapshot differential display (zfs diff) 1/2

– You can display the differences between two snapshots.

Snapshot A (before data update)
tank/data@snap1

Snapshot B (after data update)
tank/data@snap2



✓ zfs diff command execution

```
$  zfs diff tank/data@snap1 tank/data@snap2
M  /tank/data/
M  /tank/data/fileA
-  /tank/data/fileB
R  /tank/data/fileC  ->  /tank/data/fileC’
+  /tank/data/fileD
```

✓ Snapshot differential indicators

M: Indicates a change of a file or a directory.

R: Indicates that a file name or directory name has changed.

-: Indicates that a file or directory resides in an old snapshot but not in a new snapshot.

+: Indicates that a file or directory resides in a new snapshot but not in an old snapshot.

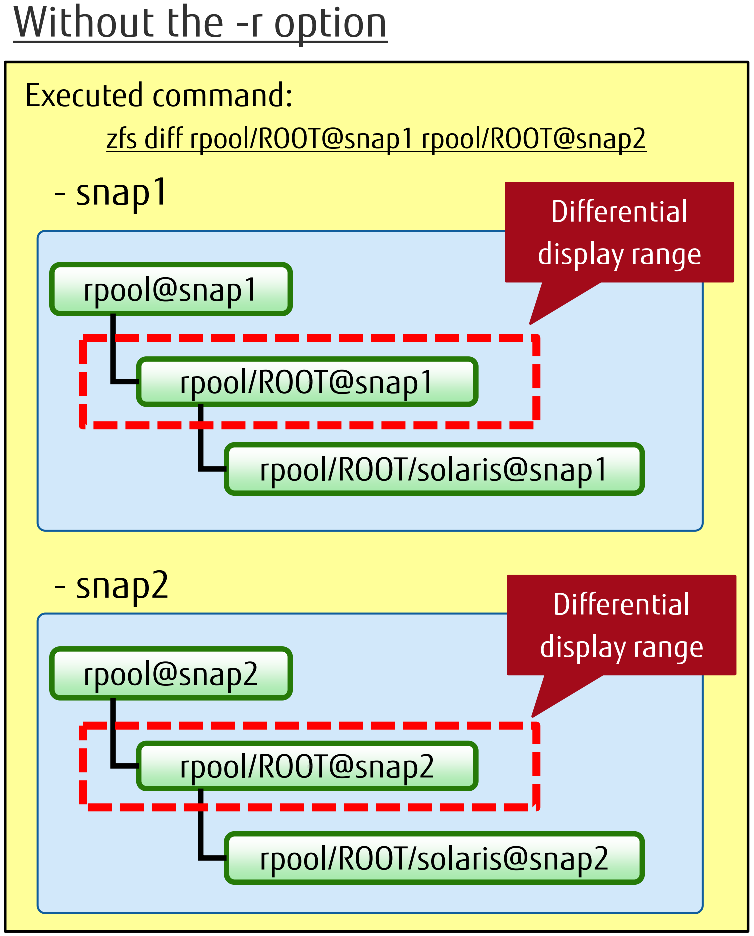
– If the zfs diff command is executed with only one snapshot specified, the command displays the differences between the snapshot and the current file system.

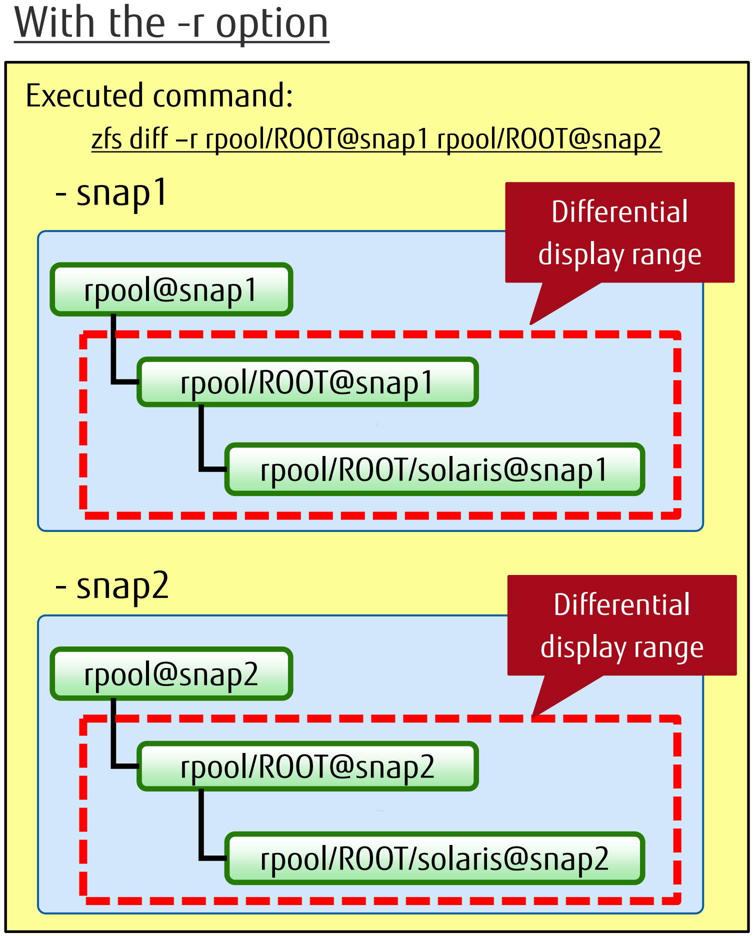
Snapshot 5/5

Snapshot differential display (zfs diff) 2/2

- By specifying the -r option, you can (recursively) display the differences between the specified snapshot and all child snapshots.

  * This function has been available since Solaris 11.3.

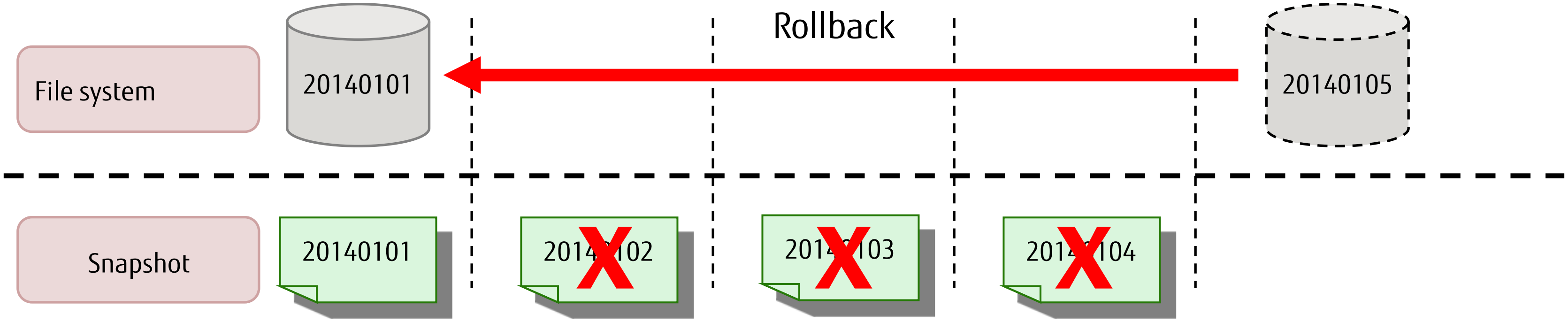




Rollback

What is rollback?

- Rollback cancels all the changes made since the creation of a specific snapshot, and restores the file system to the state at the time that the snapshot was created. By creating a snapshot before changing data, you can restore the data to the unchanged state even if you deleted data by mistake.



Notes

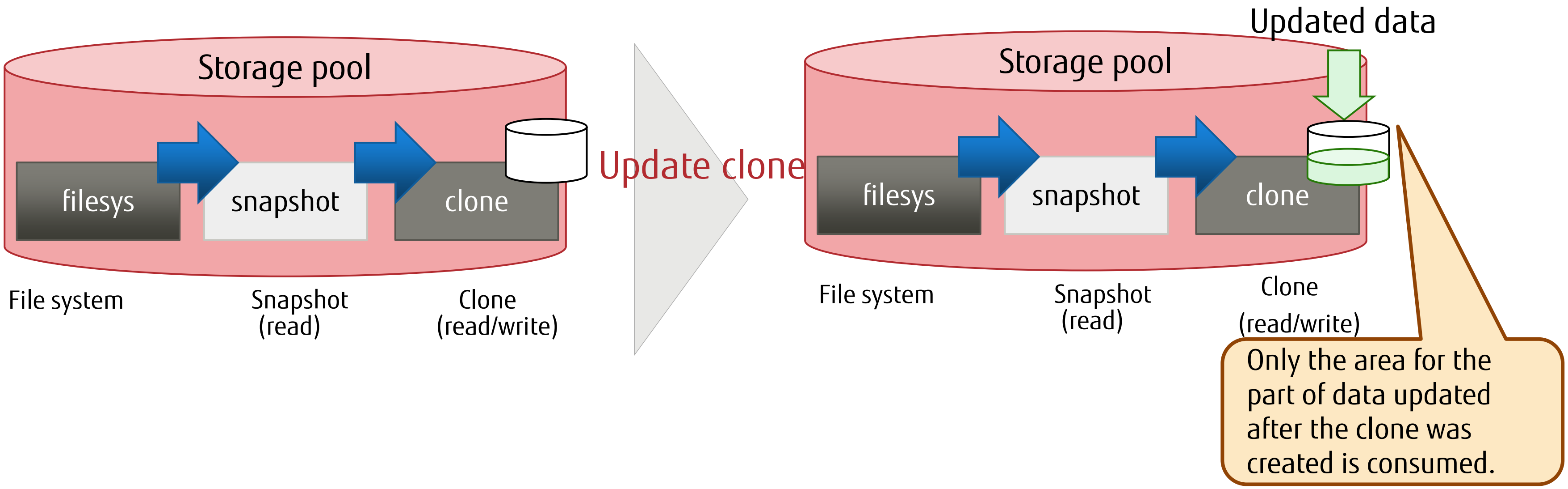
- Rollback can use only the latest snapshot.
- For rollback using a specific snapshot, delete intermediate snapshots so that the target snapshot is the latest one.

> – If there is a clone of an intermediate snapshot, the clone must be destroyed too.

Clone 1/3

What is a clone?

- A clone is a copy of a file system or volume, which can then be easily created.
- Unlike a simple data copy, creating a clone does not consume any disk area at all. However, if the data is updated after the clone is created, disk area is consumed.
- A clone is created from a snapshot of the source file system or volume.



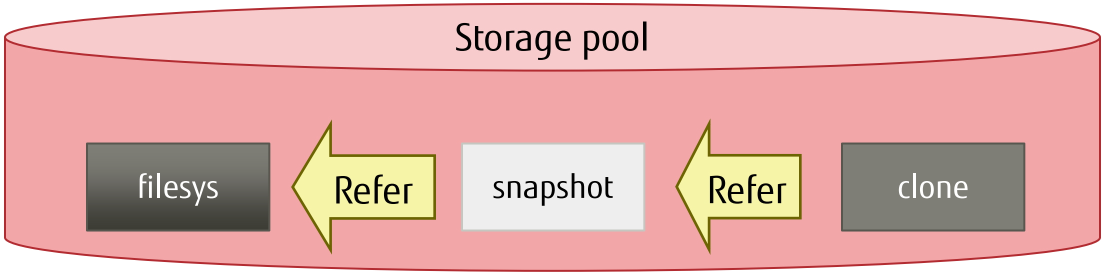
Only the area for the part of data updated after the clone was created is consumed.

- A clone is created only from a snapshot. You can create a snapshot of a clone. However, the properties of the source data set are not inherited.
- To replicate a Solaris zone environment, a ZFS clone is used. Therefore, it can be replicated instantly. Only the data update portion consumes disk area.

Clone 2/3

Dependency

- There are dependencies among the source file system, a snapshot, and a clone.
- If a clone exists, the source snapshot of the clone cannot be deleted. To delete the snapshot, the clone must be deleted first.



How to check dependencies

- You can check dependencies by using the origin property.

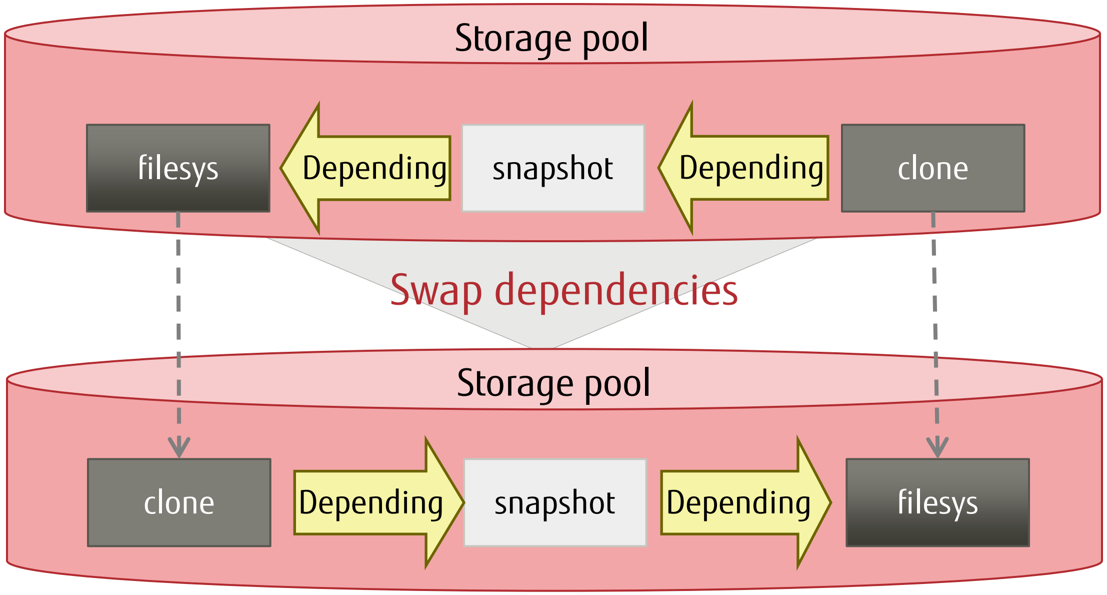
```
# zfs get origin upool/file
NAME          PROPERTY     VALUE                  SOURCE
upool/file    origin       upool/data@now         -
```

* The dependency between the file system upool/file and the snapshot upool/data@now can be confirmed.

Clone 3/3

Swapping the master and a clone

- You can invert the dependency between a clone and the source file system (master) so that the clone becomes the master.
- You can have the master reflect the changes made in the clone environment, by swapping the master and the clone.



- Before editing data that affects the system, create a clone. Edit the clone data, and then if there is no problem, you can safely edit the data by swapping the master and the clone.

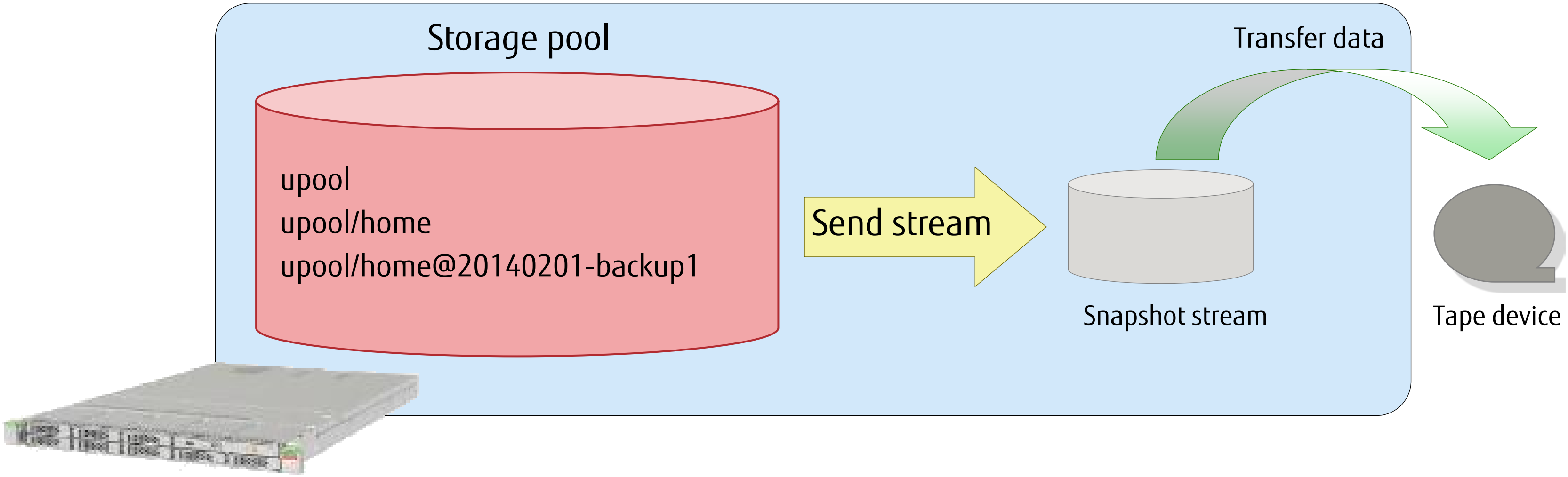
Backup/Restore 1/2

Backup (sending a stream)

- ZFS backs up the file system by generating and sending a stream from a snapshot. You do not have to stop the OS when creating or sending the stream.
  - * A stream is multiple continuous pieces of data treated as one piece of data.

Types of backup

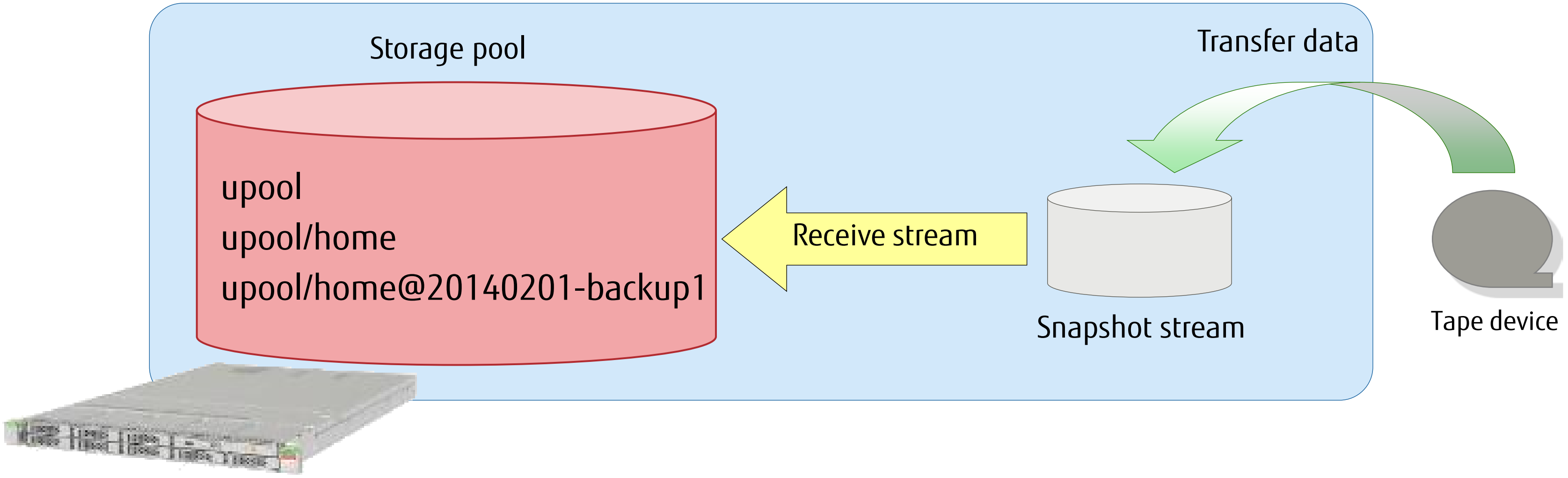
- Full backup (sending a whole snapshot)
- Differential backup (sending snapshot differences)



Backup/Restore 2/2

Restore (receiving a stream)

- ZFS restores the file system by receiving the stream that was generated and sent as a backup.
  Like when sending a stream, you do not have to stop the OS.

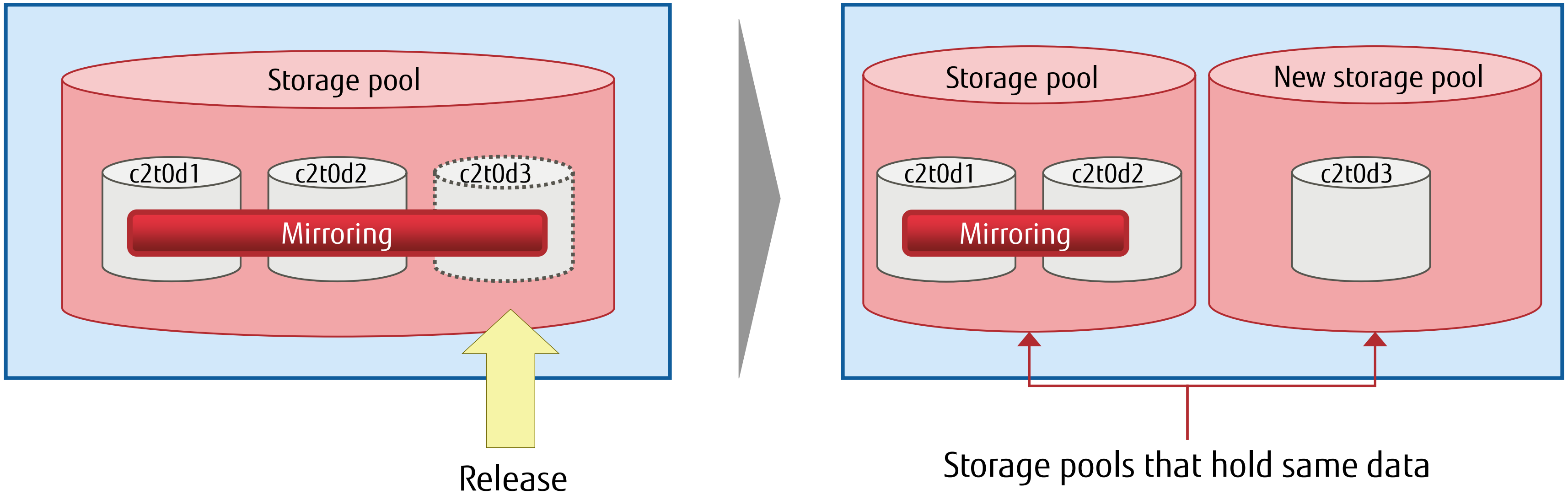


- The target file system and the file systems under it are inaccessible during restore (while receiving a stream).
- You can specify a specific snapshot to restore a file system. Unlike rollback, you do not have to delete intermediate snapshots so that the target snapshot becomes the latest one.

Release Mirror Disk 1/5

Releasing a mirror disk with the zpool split command

- You can release a disk from a storage pool in a mirror configuration to create a new storage pool that holds the same file systems and data as in the original storage pool.



- By importing a released disk, you can create a new storage pool that holds the same file systems and data as in the original ZFS storage pool. The new storage pool can be used as a backup.
- To maintain data redundancy, we recommend performing this operation from a storage pool consisting of at least three mirror disks.

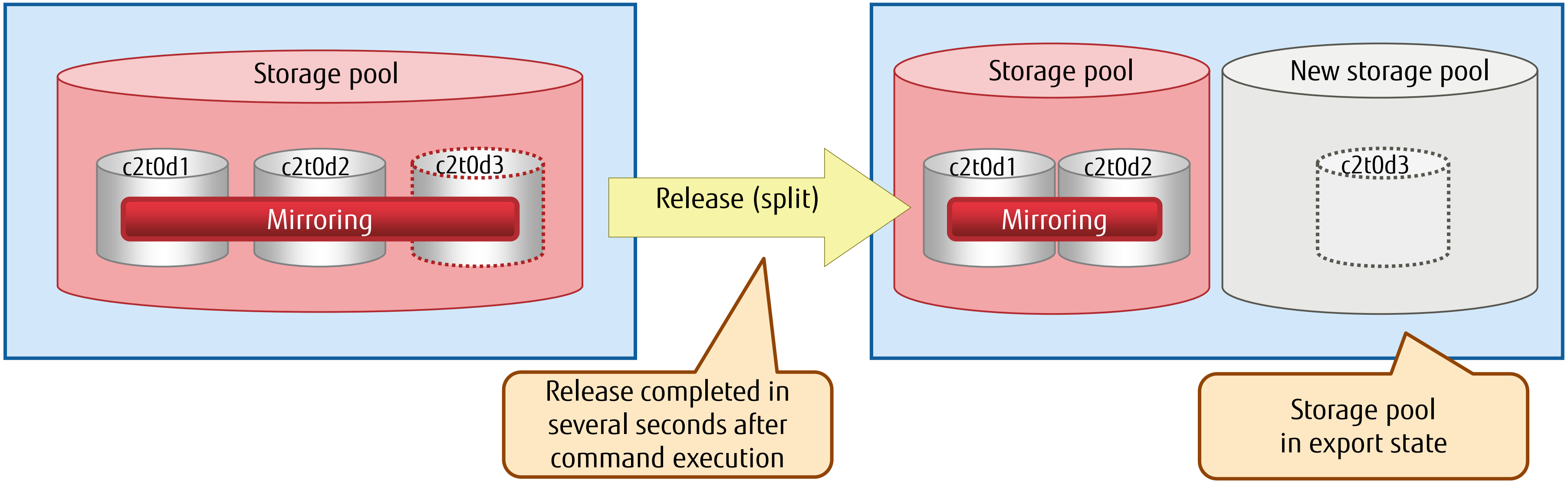
Release Mirror Disk 2/5

- Features of the zpool split command
  - Easy to release a disk
    - You can release a disk from a storage pool in a mirror configuration in several seconds without stopping and restarting the OS.
  - Created storage pool holds the same data
    - You can create a new storage pool that holds the same file systems and data as in the original storage pool.
  - Migrating data to other servers
    - You can easily migrate the data of a storage pool by connecting the released disk to another server.

Release Mirror Disk 3/5

Easy to release a disk

- Execute the zpool split command. Several seconds later, it releases a disk in a mirror configuration and creates a new storage pool.
- After the release, the created storage pool enters the export state, so it is not recognized by the OS.



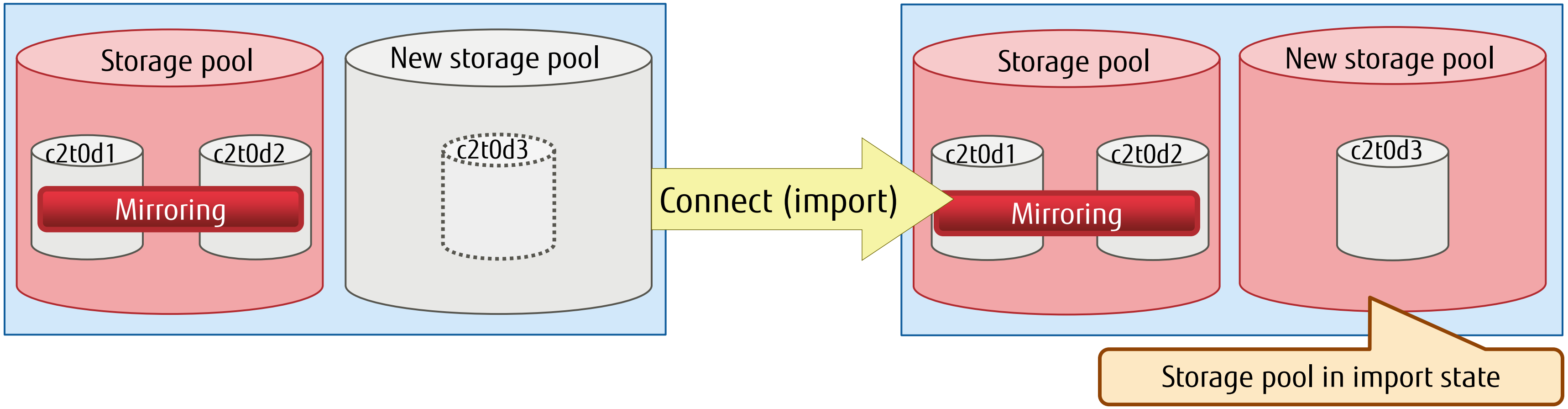
Release completed in several seconds after command execution

Storage pool in export state

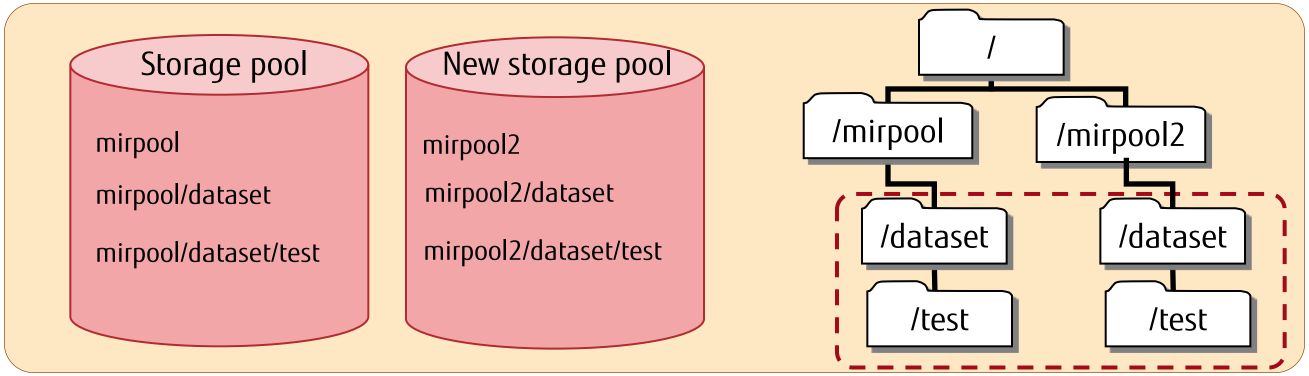
Release Mirror Disk 4/5

Created storage pool holds the same data

- The new storage pool created with a released disk from a mirror configuration is in the export state. When the import command is executed, it is recognized by the OS .



- The new storage pool holds the same file systems and data as in the original storage pool.

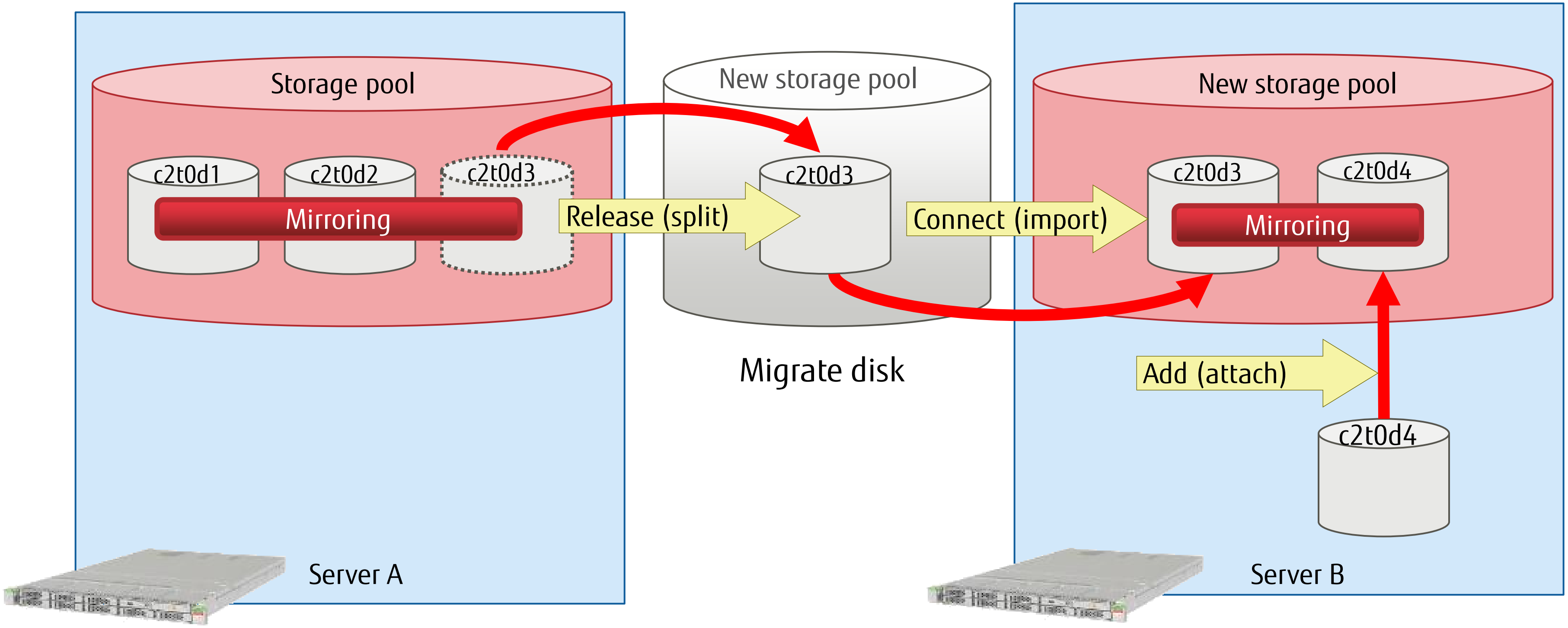


* The new storage pool and the original storage pool cannot have the same name.

Release Mirror Disk 5/5

Migrating data to other servers

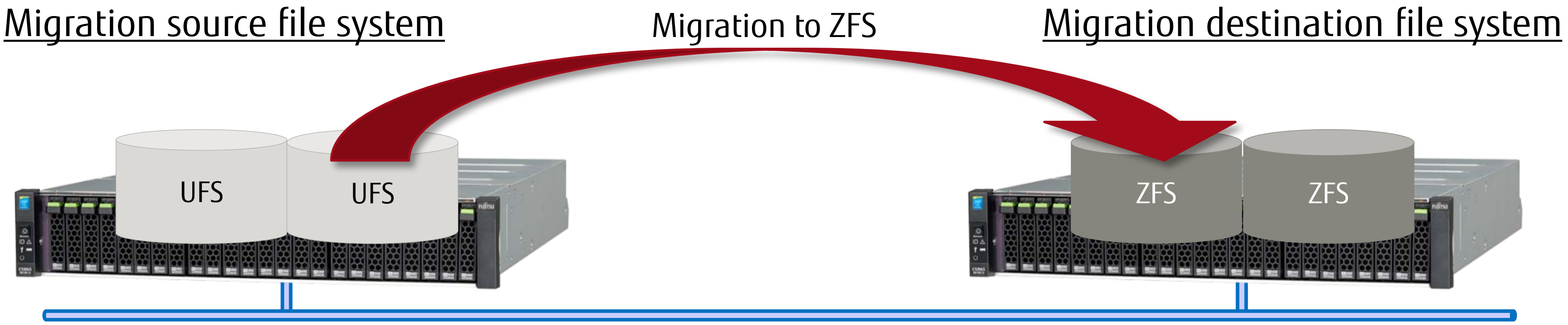
- You can easily migrate a storage pool by connecting the released disk to another server.



– Since the storage pool with the released (split) disk is in the export state, you can just migrate the disk as is by connecting it (import) to another server.

Migrate Data From UFS to ZFS

Using ZFS Shadow Migration, you can migrate a file system from UFS to ZFS without stopping the OS.


Migration source file system
Migration to ZFS
Migration destination file system
UFS
UFS
ZFS
ZFS

✓ Setting on the migration source file system

- Before migrating the file system, set it to read-only.

✓ Setting on the migration destination file system

- Create a new file system, and set the name of the migration source file system in the property for migration (shadow property).

- You can migrate an existing file system (UFS, NFS, or ZFS) to ZFS.
- For NFS, the time required for migration depends on the network bandwidth.
- Depending on whether the migration source file system is the local file system or NFS, the method of setting the shadow property of the destination file system differs as follows:

shadow=file://path	- Syntax for migrating the local file system
shadow=nfs://host:path	- Syntax for migrating NFS

7. ZFS Operations

This chapter presents, by purpose, processes that can be done as required during operation.

Operations Listed by Purpose

The next pages present, by purpose, processes that can be done during operation.

- Expanding the storage pool area
- Configuring/Managing a hot spare
- Replacing a disk
- Migrating a storage pool
- Monitoring ZFS operation

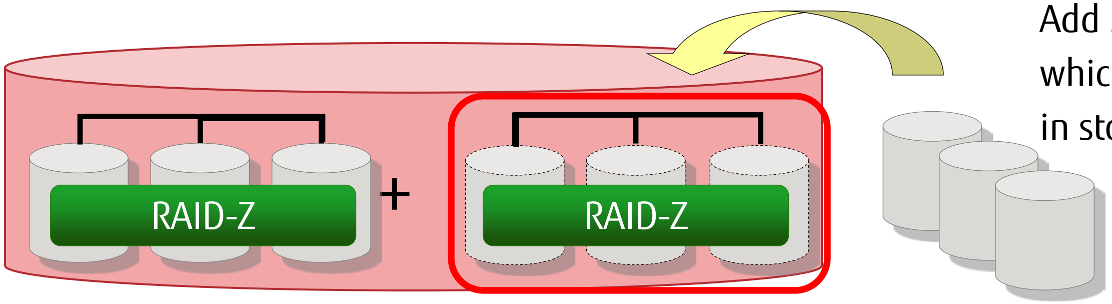
Expanding the Storage Pool Area

Ability to expand the storage pool area

- You can dynamically add disk area to a storage pool. The added area can be used immediately.

Adding a disk

- Expand the disk area in units of disk groups with the same RAID configuration.



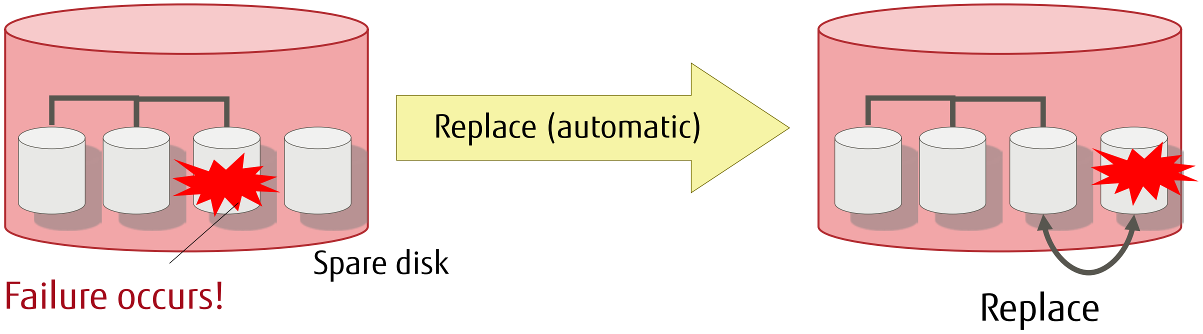
Add 3 disks in RAID-Z configuration,
which is same as
in storage pool

⚠ – The area cannot be reduced. Also, the RAID configuration of any added disk group cannot be changed.

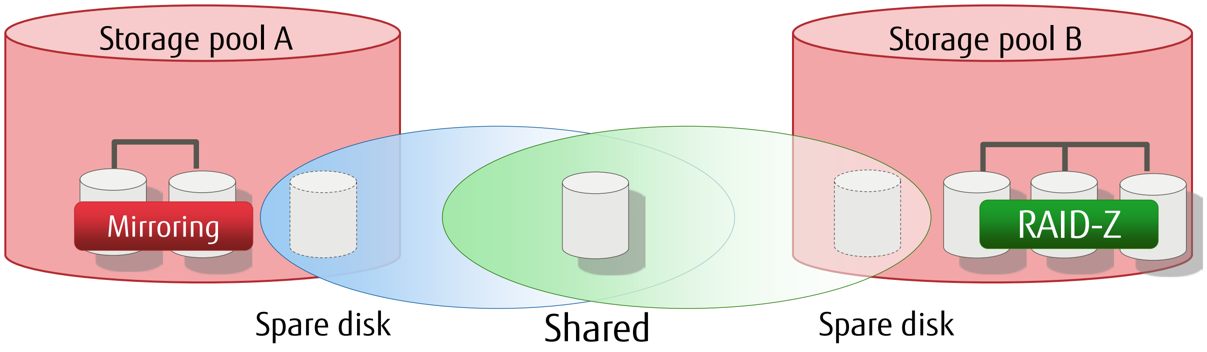
💡 – We recommend that the added disk group and the existing disk group have the same redundancy level.

Configuring/Managing a Hot Spare

- Ability to configure/manage a hot spare
  - In a storage pool configured with a spare disk, a disk that fails is automatically replaced.



  - Multiple storage pools can share the disk specified as a spare disk.



- We recommend using a hot spare when RAID-Z, RAID-Z2, or RAID-Z3 is used.
- The disk used as a hot spare must be the same size as or larger than the disk to be replaced.

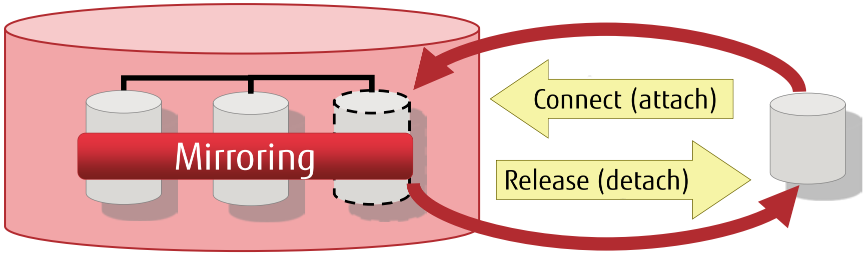
Replacing a Disk

Ability to replace a disk

- You can replace a disk by either the "release (detach)/connect (attach)" or "replace (replace)" method, depending on the mirror configuration.

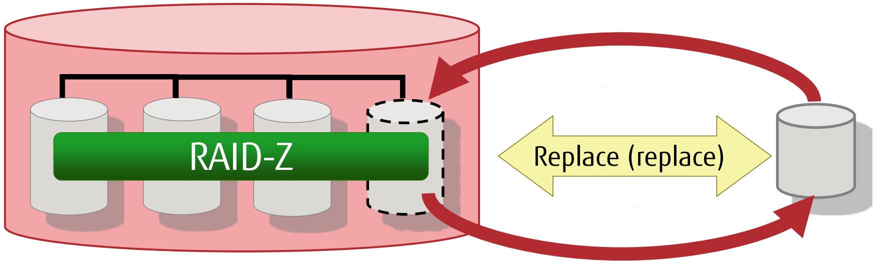
Release (detach) / connect (attach)

* RAID 1 (mirroring) only



Replace (replace)

* RAID 1 (mirroring)/RAID-Z/RAID-Z2/RAID-Z3

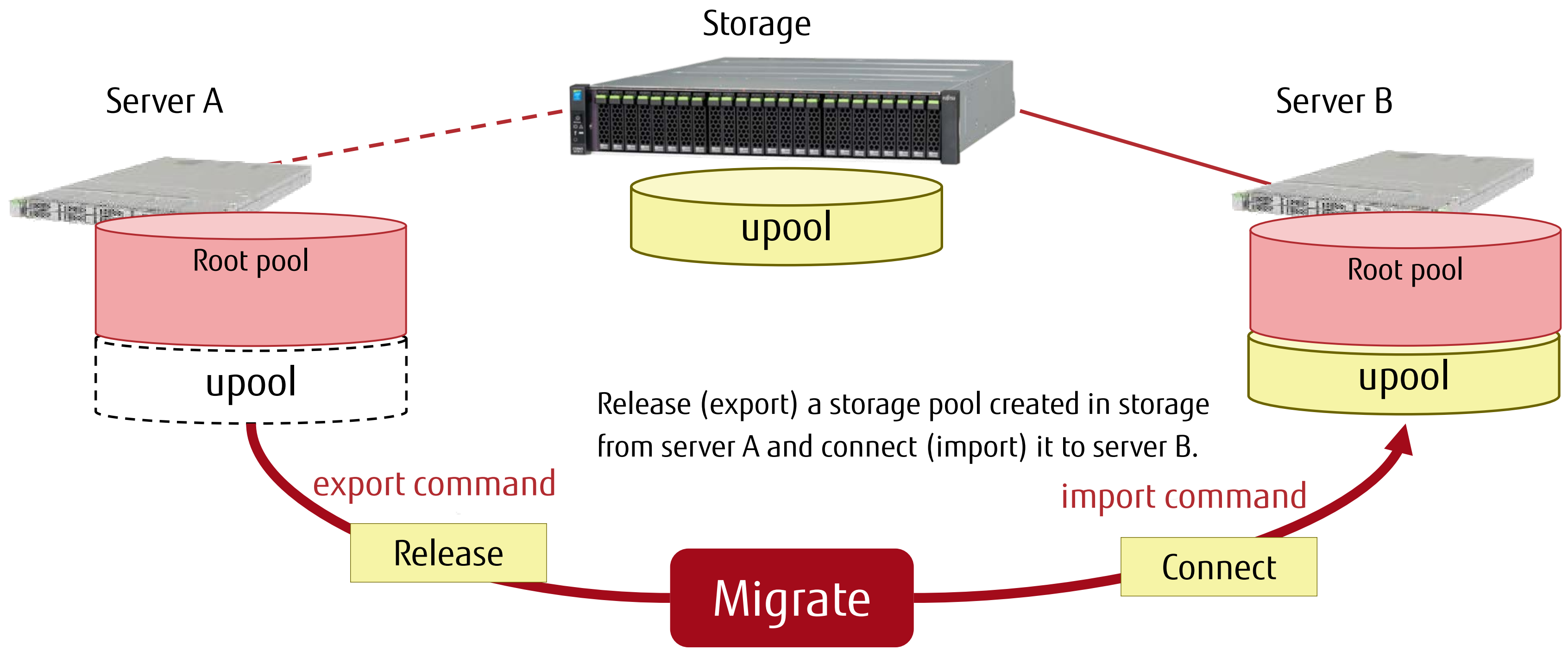


- Synchronization begins immediately when a new disk is connected.
- When replacing a disk, use a disk that is the same size as or larger than the disk to be replaced.

Migrating a Storage Pool

Ability to migrate a storage pool

- You can release (export) a storage pool and connect (import) it to another server. You can easily migrate a storage pool by connecting a disk in a storage pool to another server.



Release (export) a storage pool created in storage from server A and connect (import) it to server B.

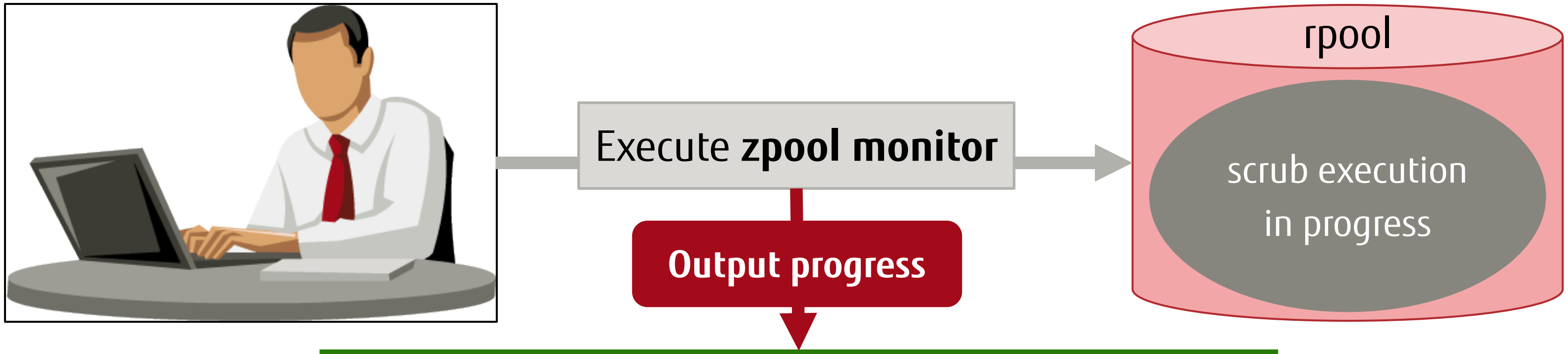
- When released (export), a storage pool is temporarily not recognized by the system. When connected (import), the storage pool is recognized again.
- The root pool cannot be migrated.

Monitoring ZFS Operation

Ability to monitor ZFS operation

– You can monitor the operations on the file systems and storage pools.

* This function has been available since Solaris 11.3.



Execute **zpool monitor**

Output progress

rpool

scrub execution in progress

```
# zpool monitor -t scrub 5
POOL              PROVIDER  PCTDONE  TOTAL  SPEED
TIMELEFT
rpool             scrub       6.8    20.8G  506M   39s
rpool             scrub      26.9    20.8G  737M   21s
rpool             scrub      63.0    20.8G  1.03G  7s
  :
```

* Output continues until Ctrl+C is input or processing ends. In the example on the left, the progress of scrub processing is displayed every five seconds.

Operations where progress can be monitored

File System Operation (zfs Command)	Storage Pool Operation (zpool Command)
- Send/Receive ZFS data (send/receive)	- Verify file system (scrub)
- Destroy snapshot (destroy)	- Resynchronize pool (resilver)

-> For details on verifying a file system (scrub) see "Scrub - Explicit Data Inspection -."

Appendix

Related Documents

Managing ZFS File Systems in Oracle® Solaris 11.3 (Oracle)
  https://docs.oracle.com/cd/E53394_01/pdf/E54801.pdf

Revision History

Edition	Date	Description
First	December 2016	First edition created

Terms of Use and Trademarks

Terms of Use

Copyrights, trademarks, and other intellectual property rights

- The contents (text, images, audio, etc.) are protected by copyrights, trademarks, and other intellectual property rights. The contents can be printed or downloaded for personal use only. The contents may not be otherwise used (reused on a personal webpage, uploaded to another server, etc.) without the prior permission of Fujitsu or the rights holder.

Limitation of warranties

- Fujitsu does not warrant the accuracy, merchantability, fitness of use, etc. of the contents. Fujitsu assumes no liability for any damages arising from use of the contents. The contents may be modified or deleted without any prior notice.

Export or provision to another party

- Before exporting this product or providing it to another party, check the applicable export control regulations such as the Foreign Exchange and Foreign Trade Act of Japan and the Export Administration Regulations of the United States, and follow the necessary procedures.

Trademarks

- UNIX is a registered trademark of The Open Group in the United States and other countries
- SPARC Enterprise, SPARC64, SPARC64 logo, and all other SPARC trademarks are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries and used under license.
- Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates in the United States and other countries.
- All other product names mentioned herein may be product names, trademarks, or registered trademarks of their respective owners.

FUJITSU

shaping tomorrow with you