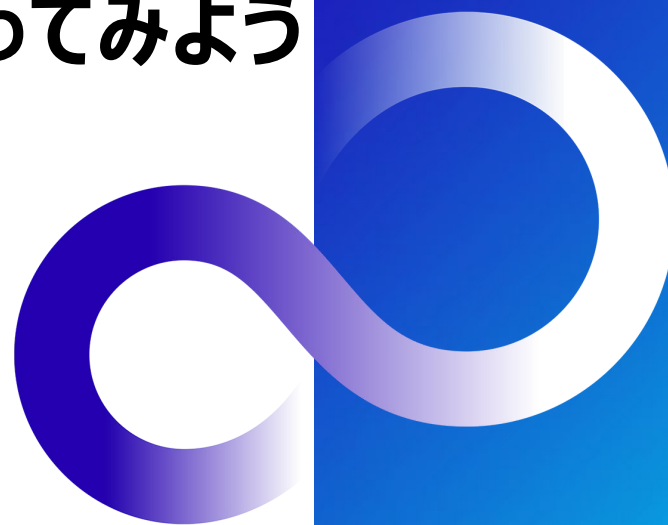


Oracle Solaris 11 ZFSを使ってみよう (概要・設計ガイド)

2019年6月（第5.0版）

富士通株式会社



■ 目的

- Oracle Solaris 11の標準ファイルシステムであるZFS（Zettabyte(ゼタバイト)File System）の概要を紹介します。

■ 対象読者

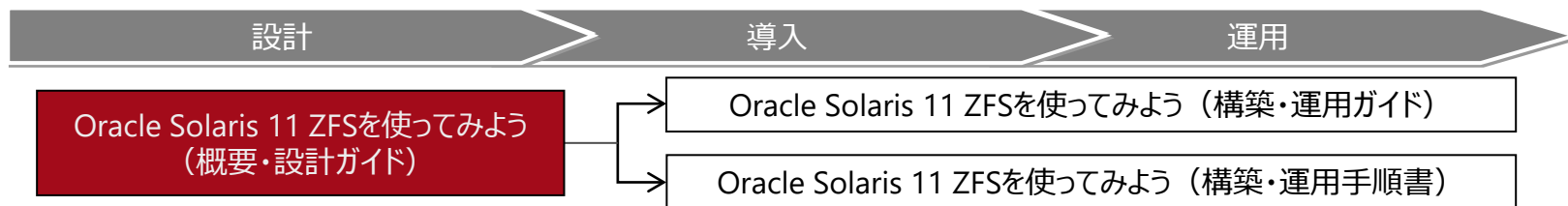
- Oracle SolarisとRAIDの基礎知識を有している方

■ 留意事項

- 本書の内容は、Oracle Solaris 11.4に基づいています。
Oracle Solaris 11に関する最新情報は、Oracle社のマニュアルをご参照ください。
 - Oracle Solaris Documentation
<https://docs.oracle.com/en/operating-systems/solaris.html>

■ ドキュメントの位置づけ

- Oracle Solaris 11 ZFSを使ってみよう
<https://www.fujitsu.com/jp/sparc-technical/document/solaris/#solaris11zfs>



■ 本書での表記

- コマンドのセクション番号は省略しています。

例：

- ls(1) ⇒ lsコマンド
- shutdown(1M) ⇒ shutdownコマンド

- 以下の用語は略称を用いて表記する場合があります。

略称	正式名称
Solaris	Oracle Solaris
Solarisゾーン	Oracle Solarisゾーン
GDS	PRIMECLUSTER GD
SVM	Solaris Volume Manager

1. ZFSの概要
2. ZFSの特長
3. ZFSの信頼性・可用性
4. ストレージプールの構成
5. ファイルシステムの管理
6. ZFSの機能
7. ZFSの運用
- 付録

1.ZFSの概要

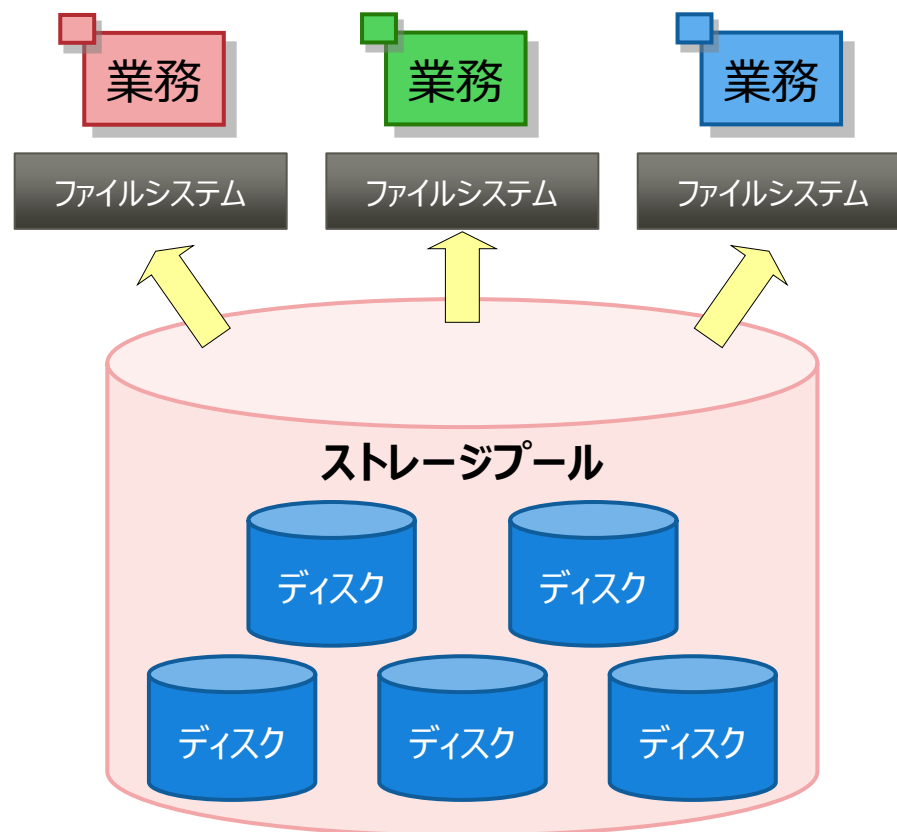
ZFSファイルシステムの概要や仕組み、およびZFSによって実現できる仮想化環境を説明します。

■ ZFSの概要

- Solaris 11標準の次世代ファイルシステムです。
- 「容量の拡張性」、「管理の容易性」、および「データの堅牢性」という優れた特性と機能を有しています。
→詳細は、「[2.ZFSの特長](#)」を参照してください。

■ ZFSの仕組み

- 「ストレージプール」という機能により、複数のディスクを一括で管理・プール化します。
- ストレージプールにより、業務や用途ごとにファイルシステム（データの格納領域）を自由に作成できます。
- ストレージプールにディスクを追加することで、簡単に容量を拡張できます。



Oracle Solarisで実現する仮想化基盤の構築

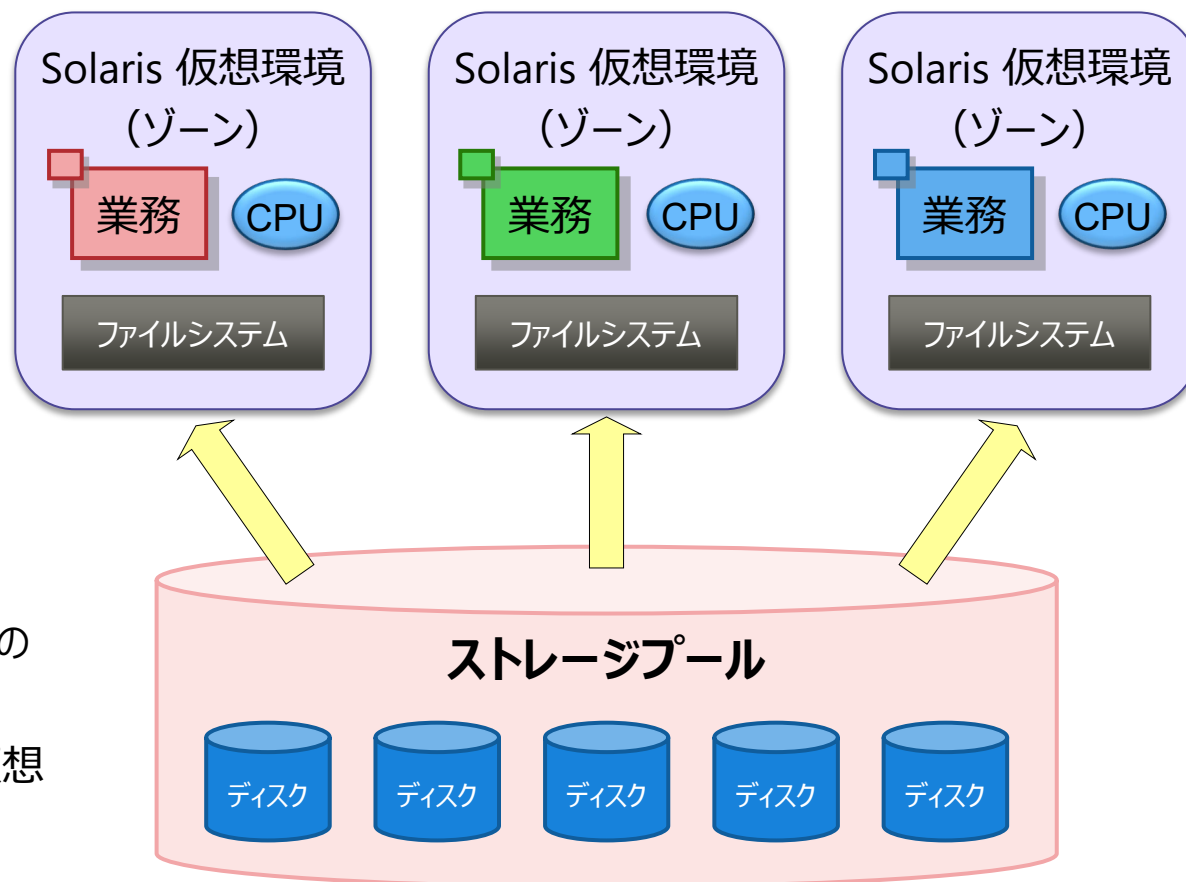
ZFSは、多数・大容量のディスクリソースを簡単に管理・配分する「ストレージ仮想化機能」の側面も持ち合わせています。ZFSとサーバ仮想化機能「Solarisゾーン」を組み合わせることで、サーバもストレージもまとめた仮想化基盤を構築できます。

Solarisゾーン

- 複数の仮想OS環境を構築
- 各仮想環境に対して、柔軟にCPUリソースを配分

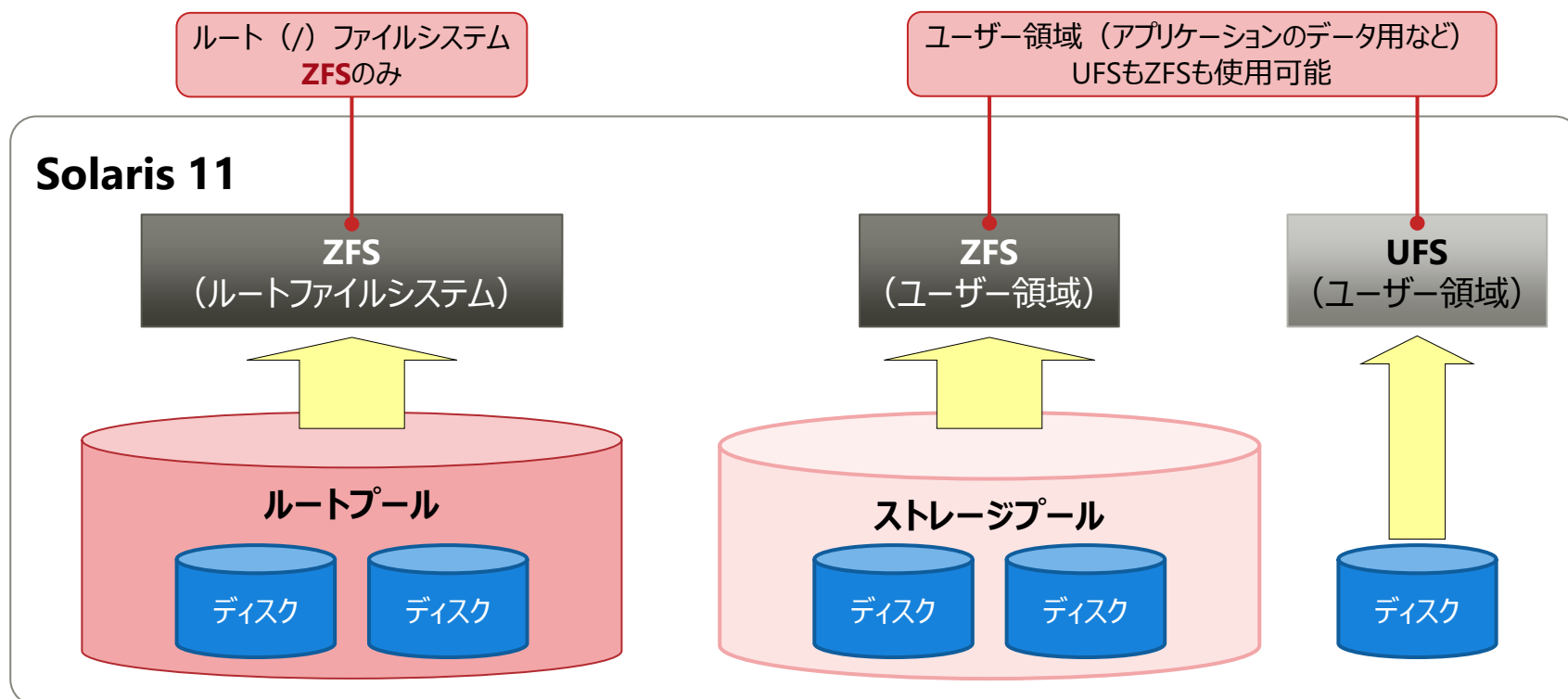
ZFS

- ストレージプールによって、すべてのディスクを一元管理
- ディスクリソース（容量）を各仮想環境に自由に配分



■ Solaris 11では、ZFSが標準ファイルシステムとなります。

- Solaris 11のシステム領域（ルートファイルシステム）でサポートするのは**ZFS**のみです。
- 従来のUFSファイルシステムは、ユーザー領域のみで使用できます。
- システム領域のストレージプールは、「ルートプール」と呼ばれます。



2.ZFSの特長

ZFSの特長である、「容量の拡張性」、「管理の容易性」、および「データの堅牢性」について説明します。

■ 容量の拡張性

- 世界初の128ビットのファイルシステム
- 事実上無限大の容量といえるファイルシステムの構築

■ 管理の容易性

- シンプルな管理コマンド
- ボリューム管理ソフト不要
- ストレージプールによるディスクの一元管理

■ データの堅牢性

- RAID構成によるディスク冗長化
- コピーオンライト（Copy-On-Write）方式（以降、「コピーオンライト方式」と表記）によるトランザクションファイルシステム
- チェックサムによる不正データの検出と自己修復

128ビットのファイルシステムによって、ZFSはほぼ無限大の容量とデータを管理できます。

UFSとZFSの比較を表に示します。

	UFS	ZFS
ファイルシステムサイズの上限	16 TB	25京6000兆ZB (256,000,000,000,000,000,000,000,000 TB)
ファイルシステム内に作成可能な ファイル数	ノードの数分 (※)	上限なし

※ ファイルシステムのサイズにより異なります。16 TBの場合、 15×10^6 個です。

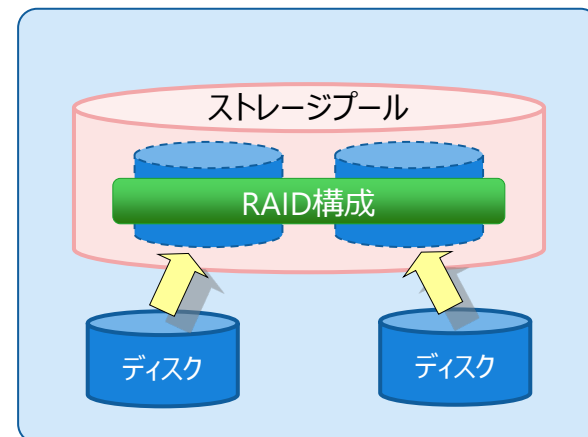
■ ZFSの管理に必要なコマンドは2種類のみ

■ zpoolコマンド

ストレージプールを管理するコマンドです。

<主な操作>

- 単一または複数のディスクからストレージプールを作成
- ディスクの冗長構成（RAID）を設定
- データ整合性チェック（スクラブ）

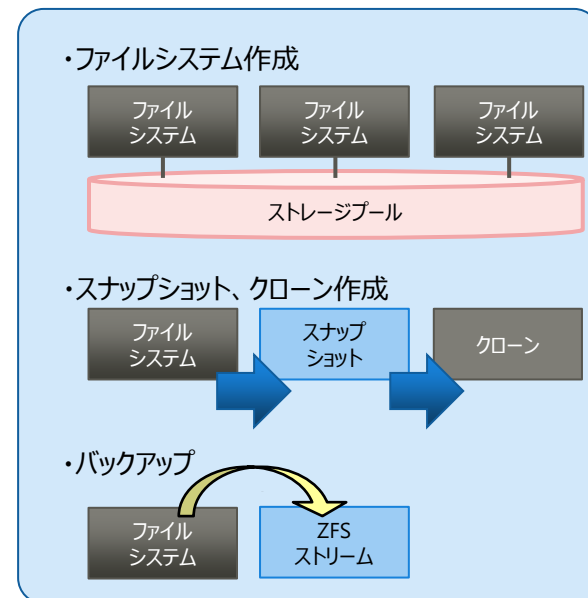


■ zfsコマンド

ZFSファイルシステムを管理するコマンドです。

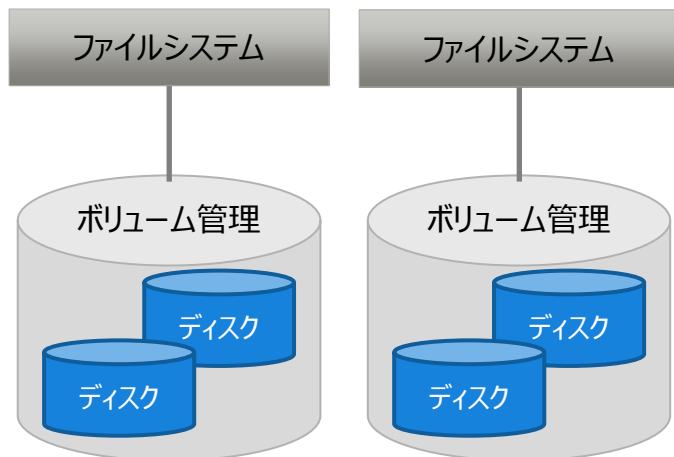
<主な操作>

- ストレージプール上にファイルシステム作成、マウント
- スナップショットやクローンの作成
- ファイルシステムのバックアップ、リストア



■ UFSとZFSのファイルシステム構成の比較

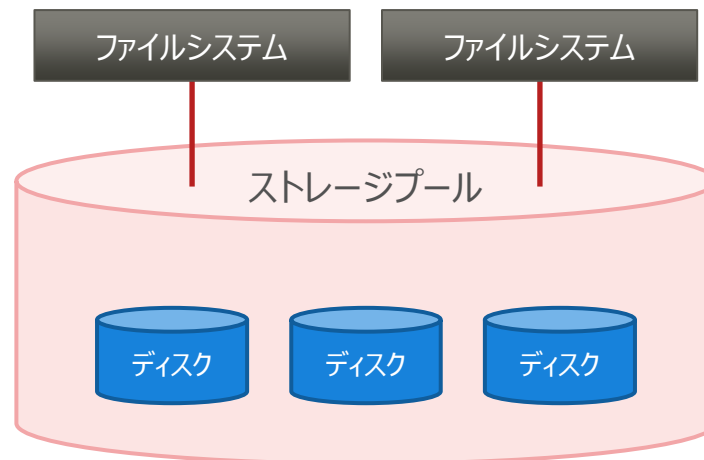
■ UFS



- ・ ファイルシステムごとに、**ボリューム管理ソフト（GDS、SVMなど）による設定が必要。**
- ・ ファイルシステムのサイズを変更する場合、**OSを停止してバックアップ／リストアが必要。**
- ・ 設計時に、パーティション構成やファイルシステムサイズを**決める必要がある。**
- ・ 急なシステムダウンにより、**データの不整合が発生することがある。**

→ UFSの作成については、「[《参考》UFSの作成](#)」を参照してください。

■ ZFS



- ・ ストレージプールによるディスク一元管理。**ボリューム管理ソフトは不要。**
- ・ **オンラインのままディスク増設が可能。**
- ・ 設計時に、パーティション構成やファイルシステムサイズを**決める必要はない。**
- ・ 急なシステムダウンでも、**データの不整合は発生しない。**

■ Oracle Solaris 11の標準機能でディスクの冗長化（RAID構成）を実現

ZFSでサポートするRAID構成

RAID 0（ストライピング）

非冗長構成。ただしRAID 1と併用も可能（RAID 1+0）

RAID 1（ミラー）

3本以上のディスクによるミラー構成も可能

RAID-Z

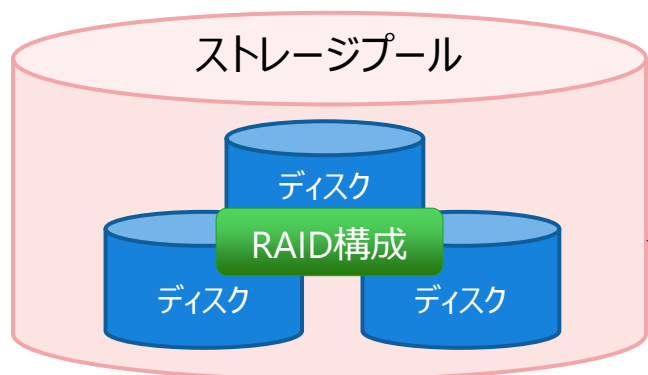
シングルパリティ
（RAID 5に類似、1本のディスク障害に対応）

RAID-Z2

ダブルパリティ
（RAID 6に類似、2本までのディスク障害に対応）

RAID-Z3

トリプルパリティ
（3本までのディスク障害に対応）



複数のディスクからストレージプールを作成するときに、RAID構成を指定



- 従来のRAID 5やRAID 6において、パリティの書き込み中に障害が発生した場合に、パリティとデータの不整合が発生することがありました（**書き込みホール問題**）。
- ZFSのRAID-Z/Z2/Z3では上記の不整合が発生しないため、信頼性の高いRAID構成を使用できます。

■ 堅牢性に優れたファイルシステムの仕組み

トランザクションファイルシステム

- データの書き込みは「コピーオンライト方式」で管理
- 元のデータが上書きされることはない
- 一連の処理が完全に確定するか、完全に無視される

 データの整合性を保証

End-to-Endのチェックサム

- データブロックのチェックサムは親ブロックに格納
- 異常検出時は冗長化したブロックからデータを回復
- 論理的な不整合（ソフトウェアのバグ）でも検出・訂正

 不正データの検出
データの自己修復

■ UFSは物理ディスクの領域と1対1に対応

4.ファイルシステムのマウント (mount)

- 適切なマウントポイントに割り当てます。

3.ファイルシステム作成 (newfs、fsck)

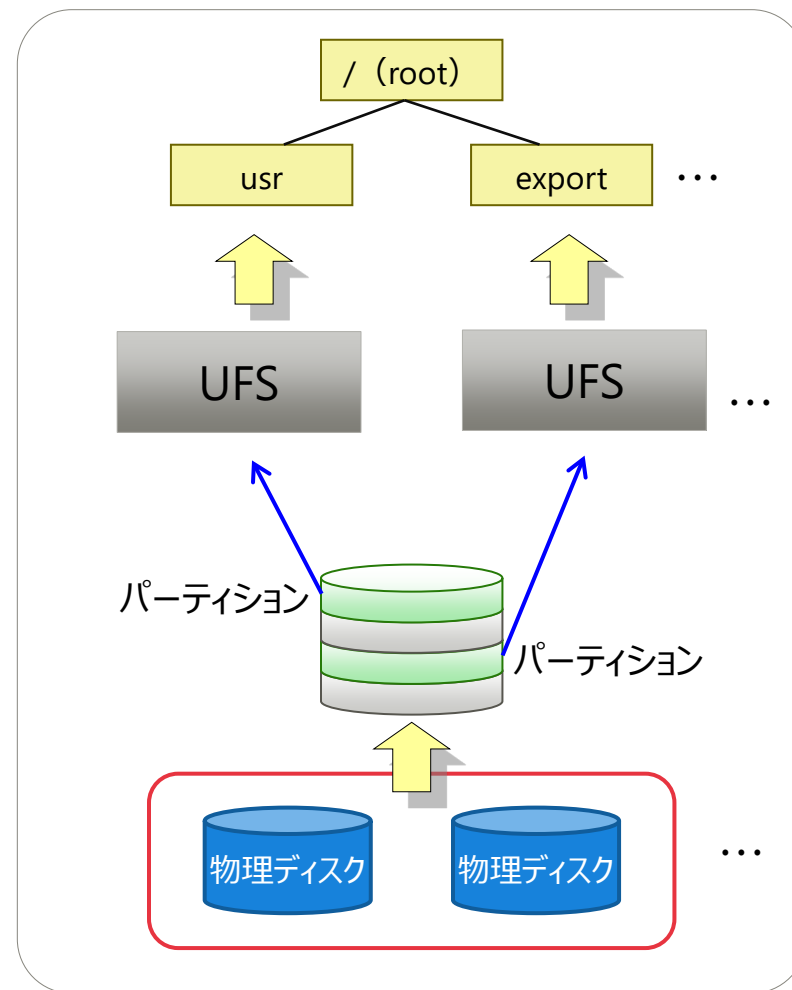
- パーティション (スライス) 単位でファイルシステムを作成し、整合性チェックを行います。

2.パーティション情報作成 (format)

- ディスク上の使用する区画 (容量) を決めます。

1.物理ディスクの追加・認識

- ボリューム管理ソフトによる冗長構成や論理ディスク作成を行います。



3.ZFSの信頼性・可用性

ZFSの信頼性・可用性を支える仕組みや機能などを説明します。

■ ZFSの信頼性と可用性を支える仕組み

■ トランザクションファイルシステム

- コピーオンライト方式によるデータ更新時の整合性を確保する仕組み

■ チェックサム

- End-to-Endのチェックサムによる信頼性の高いデータ保護の仕組み

■ スクラブ

- ストレージプール内の全データの完全性をチェックする機能

ZFSでは、一連の書き込み動作を1つの更新単位（トランザクション）として扱います。

■ トランザクションファイルシステムのメリット

■ データの不整合が発生しない

- コピーオンライト方式（※1）によって、データ更新が「すべて成功」か「すべて失敗」のどちらかであることが保証されます。

※1：ファイルへの書き込み時に元データを上書きしない方式です。

→ 詳しくは「[トランザクションファイルシステム 2/2 -コピーオンライト方式-](#)」を参照してください。

■ UFSでは必須の「ファイルシステムの整合性チェック」が不要

- 不意にシステムがクラッシュしても、ファイルシステムが保護されます。

■ トランザクションファイルシステムの留意点

■ ディスクへの書き込みが非同期である

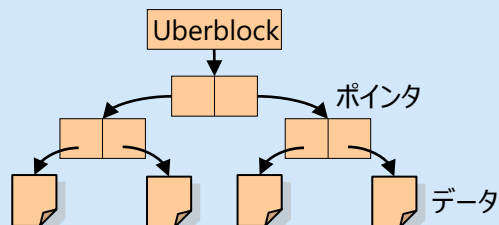
- 一連の処理が終了してからディスクへ書き込みが行われる（I/Oが発生する）ため、実際のディスク容量を確認したときに差異が発生することがあります（dfコマンド、duコマンドなど）。

ZFSのデータの更新方法であるコピーオンライト方式により、ファイルシステムやファイル自体に矛盾が発生しない仕組み（トランザクションファイルシステム）を実現しています。

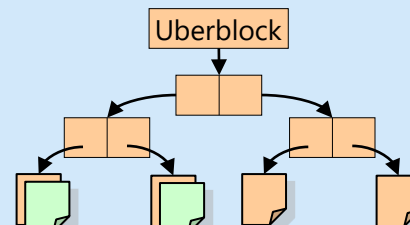
■ コピーオンライト方式のデータ更新方法

- 書き込んだデータ（①初期状態のブロック構造）を変更するときは、まず元のデータのコピーを作成し、コピーしたデータを更新します（②）。データ更新終了後、システムにとって区切りのよい時点でデータブロックのリンクを更新し（③）、新しいUberblock（※1）に置換します（④）。

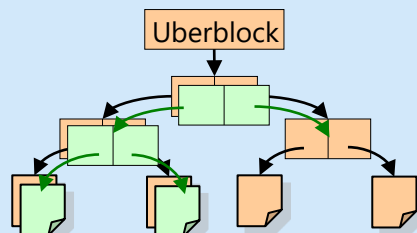
①初期状態のブロック構造



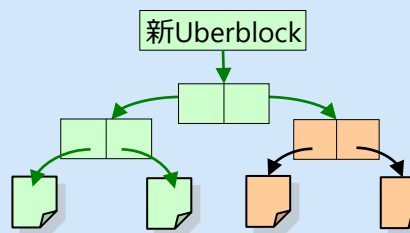
②ブロックをコピーし、コピーを更新する



③ポインタを含むブロックをコピーし、新しいデータにリンクする



④Uberblockを置換する



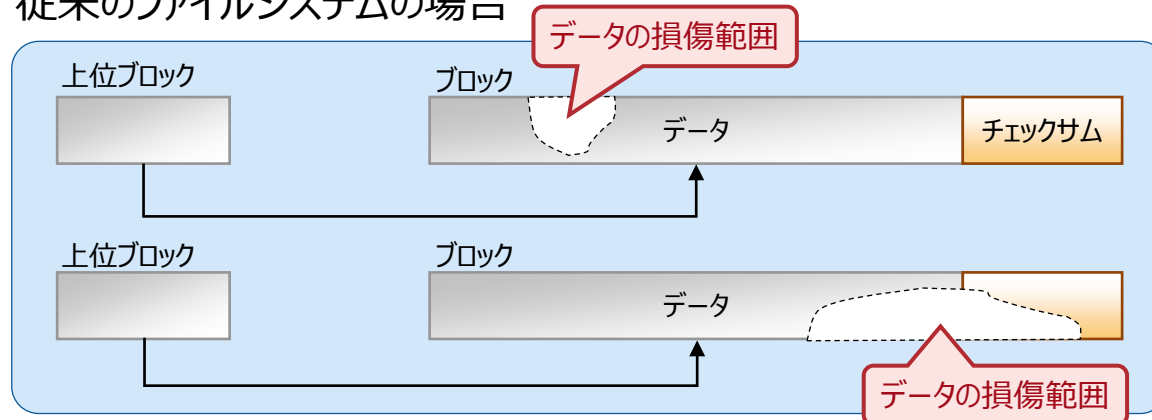
※1 : Uberblockは、ファイルシステムの重要な設定情報が保存される管理領域です。ファイルシステムごとに1つ存在します。UFSのスーパーブロックに相当します。

ZFSでは、チェックサムによって不正データを検出し、データを自動修復します。

また、チェックサムをデータとは異なるブロックに保存することで、チェックサム自身の信頼性を向上しています。

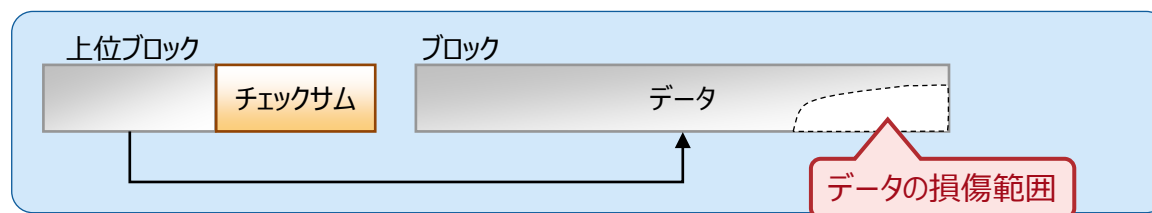
■ チェックサムによるデータ修復方法

✓ 従来のファイルシステムの場合



データの損傷範囲がチェックサムにまで及ぶと、データを修復できません。

✓ ZFSの場合



データとチェックサムが物理的に離れているため、別々に読み込まれます。チェックサム自身も、上位ブロックによって修復できます。

- 💡 ディスク故障による読み込みエラーを検出するだけでなく、ソフトウェアのバグなどによる論理的な不整合が発生しても、検出と訂正が可能です。
- データの読み書き時は、チェックサムによって不正データが検出されます。不正データが検出された場合、冗長構成（RAID 1（ミラー）、RAID-Z、RAID-Z2、RAID-Z3）であれば、複製データから自動的にデータを修復します。

スクラブは、手動でファイルシステムを検査する機能です。

すべてのデータを順番に検査し、すべてのブロックが読み取り可能であることを確認します。

■ スクラブの特長

- ハードウェアまたはソフトウェア障害が発生する前に、エラーを検出および回避できます。
- 定期的にスクラブを実行することで、ストレージプール上のディスクの完全性を継続的にチェックできます。
- スクラブ実行中にI/Oが発生すると、スクラブの優先順位は低くなります（業務への影響はありません）。

■ スクラブの実行

✓ スクラブ実行直後

```
# zpool scrub rpool
# zpool status rpool
pool: rpool
state: ONLINE
scan: scrub in progress since Mon Jan 21 16:45:11 2019
      405M scanned out of 73.6G at 41.8M/s, 29m52s to go
      0 repaired, 0.54% done
config:
```

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

errors: No known data errors



✓ スクラブ完了後

```
# zpool status rpool
pool: rpool
state: ONLINE
scan: scrub repaired 0 in 9m26s with 0 errors on Mon Jan 21
      16:54:37 2019
```

config:

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

errors: No known data errors

4.ストレージプールの構成

ストレージプールは、「ルートプール」（システム領域）と「ストレージプール」（ユーザー領域）で構成されます。ここでは、ルートプールとストレージプールそれぞれの構成を説明します。

■ ルートプール

- ・システム領域（OS起動用）のストレージプールです。
- ・OSの起動に関するファイルとディレクトリが格納されます。
（/、/bin、/sbin、/dev、/kernel、/etc、swap領域、dump領域など）

■ ストレージプール

- ・ユーザー領域（データ領域）用のストレージプールです。
- ・主にユーザーアプリケーションなどのデータを格納します。

プール名 (OSで設定する名称)	用途	構成タイミング	構成可能なRAIDレベル	動的な拡張
ルートプール (rpool)	システム領域	OSインストール時（※1）	RAID 1	不可
ストレージプール (任意の名称を設定可能)	ユーザー領域	適時	RAID 0, RAID 1, RAID 1+0, RAID-Z, RAID-Z2, RAID-Z3	可能

※1：ルートプールは、OSのインストール時に構成されます。ただし、冗長化（RAID 1）の設定は、OSインストール後に手動で実施します。



・ストレージプールを構成するデバイスには、パーティションの一部やデータファイルも利用可能ですが、構成や管理の面を考慮し、ディスク全体を使用することを推奨します。

■ OSインストール時のルートプールの構成

- Solaris 11におけるデフォルト（OSインストール時）のルートプールの構成は、単一のディスクです。
- ルートプールの名称は「rpool」固定です。
- ルートプールのディスクミラーは、OSインストール後に手動で設定します。
→ Solaris 10からの変更点については、「[《参考》OSインストール時のルートプールの構成](#)」を参照してください。

✓ OSインストール時の画面

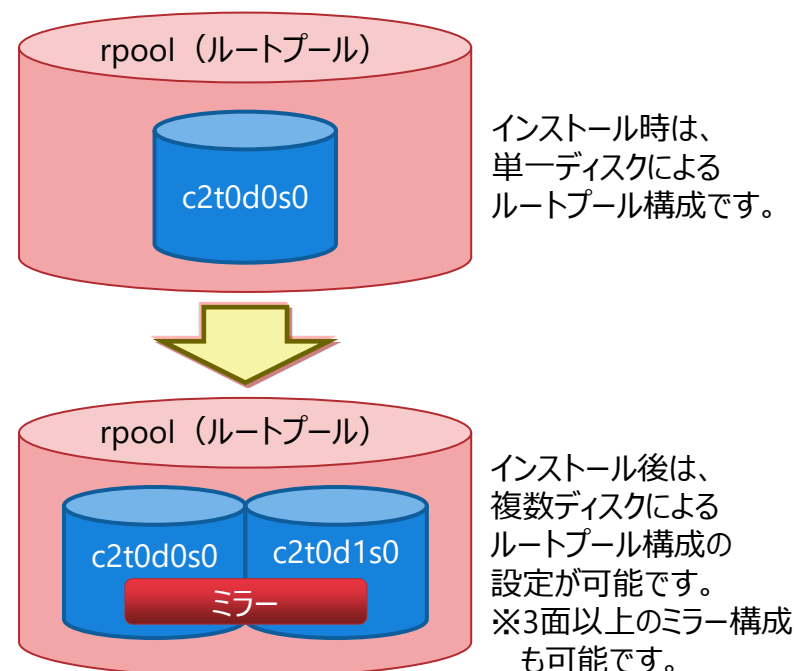
ディスク

Oracle Solaris をどこにインストールしますか？
最小サイズ: 4.1GB 推奨最小: 6.1GB

タイプ	サイズ (GB)	ブート	デバイス
unknown	29.9	+	c2t0d0
unknown	29.9		c2t0d1
unknown	19.9		c2t0d2
unknown	29.9		c2t0d4

ディスク上に次のスライスが見つかりました。

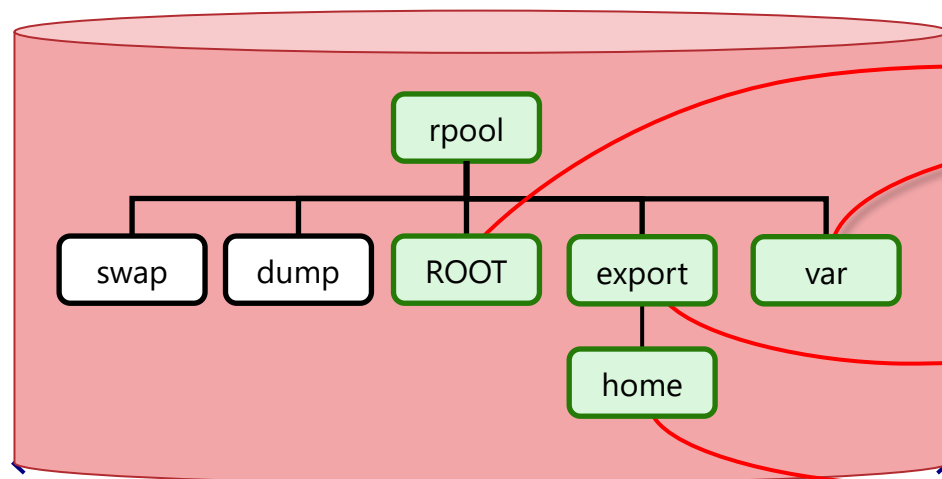
スライス	#	サイズ (GB)	スライス	#	サイズ (GB)
Unused	0	0.1	Unused	5	0.0
Unused	1	0.1	rpool	6	29.6
Unused	3	0.0	Unused	7	0.0
Unused	4	0.0	backup	2	29.9



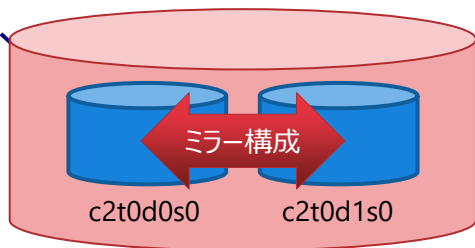
• ミラー構成にするディスクは、障害発生時を考慮して複数のコントローラーにまたがる構成を推奨します。

■ ルートファイルシステムの構成

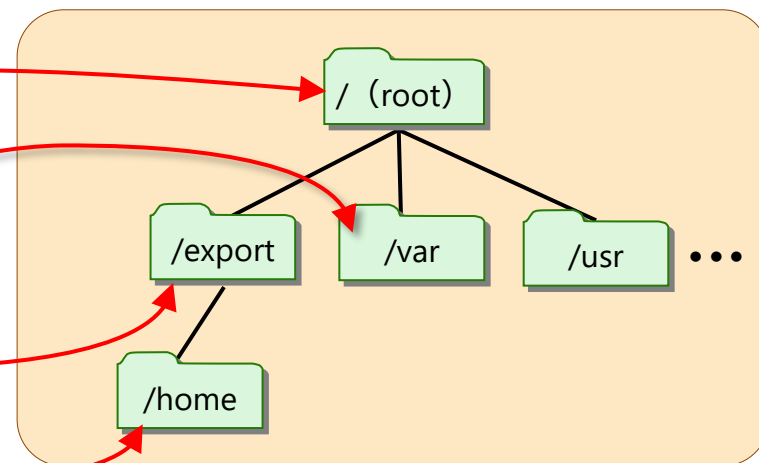
✓ ルートプール (rpool) の構成



ストレージプール内では、データセットを単位として階層構造で管理されます。



✓ ファイルシステムの構成



ルートプールに作成された各ファイルシステムは、自動的にOS上にマウントされます。



• データセットは、ストレージプール内に作成されるファイルシステム、ボリューム、スナップショット、クローンの総称です。

■ スワップ (swap) とダンプ (dump) の領域

- ZFS上では、ZFSボリュームの個別の領域にスワップ (swap) とダンプ (dump) を作成します。

✓ UFSルートファイルシステム

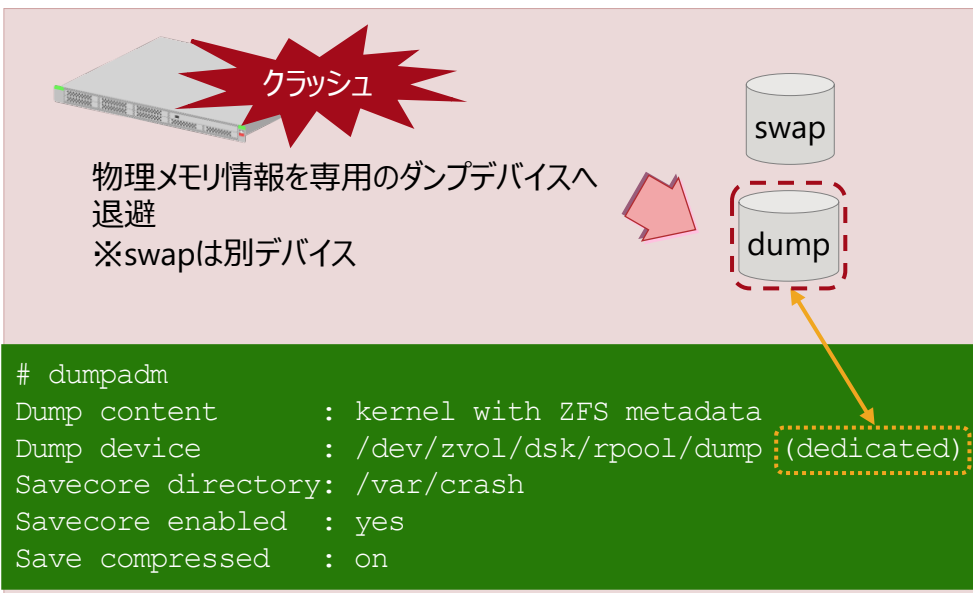
インストール時にダンプレベルに応じて、swapや/var領域の見積りが必要です。

(フルダンプの場合：物理メモリ以上の領域)



✓ ZFSルートファイルシステム

インストール後にダンプレベルを変更しても、swapや/var領域のサイズは容易に変更できます。



- ZFSボリュームは、ブロックデバイスとして使用するストレージプール上のデータセットのことです。
デバイスファイルは、/dev/zvol 配下に存在します。

■ スワップ領域とダンプ領域のデフォルトサイズ

OSの初期インストール時、スワップ領域とダンプ領域のサイズは、物理メモリのサイズに基づいてカーネルにより計算されます。これらのサイズは、OSのインストール後に変更できます。

- スワップ領域のデフォルトサイズ
 - 物理メモリサイズの1/4です。
- ダンプ領域のデフォルトサイズ
 - 物理メモリサイズの約1/2です。

✓ カーネルにより計算されるスワップ領域とダンプ領域のサイズ

システムタイプ	スワップ領域のサイズ	ダンプ領域のサイズ
約4 GBの物理メモリを搭載	1 GB	2 GB
約8 GBの物理メモリを搭載	2 GB	4 GB
約16-128 GBの物理メモリを搭載	4-32 GB	8-64 GB
128 GBを超える物理メモリを搭載	物理メモリサイズの1/4	物理メモリサイズの1/2



• カーネルにより計算されるスワップ領域とダンプ領域のデフォルトサイズの詳細については、以下を参照してください。
『Managing File Systems in Oracle Solaris 11.4 (English)』 (Oracle社)
https://docs.oracle.com/cd/E37838_01/pdf/E61016.pdf

■ スワップ領域とダンプ領域のサイズ変更

ZFSのスワップ領域とダンプ領域のサイズは、volsizeプロパティの値を変更するだけで簡単かつ動的に変更できます。

■ スワップ領域

- ✓ 現在のスワップデバイスを削除し、volsizeプロパティの値を変更した後、スワップデバイスを再構築します。

```
# swap -d /dev/zvol/dsk/rpool/swap
# zfs set volsize=3G rpool/swap
# /sbin/swappadd
# swap -l
```

swapfile	dev	swaplo	blocks	free
/dev/zvol/dsk/rpool/swap	248,1	16	6291440	6291440

←削除
←変更
←再構築
←確認

- ✓ 従来のUFSのように、スワップデバイスを追加することもできます。

```
# zfs create -V 2G rpool/swap1
# swap -a /dev/zvol/dsk/rpool/swap1
# swap -l
```

swapfile	dev	swaplo	blocks	free
/dev/zvol/dsk/rpool/swap	248,1	16	6291440	6291440
/dev/zvol/dsk/rpool/swap1	248,6	16	4194288	4194288

←領域作成
←追加
←確認

■ ダンプ領域

- ✓ volsizeプロパティの値を変更するだけで、サイズを変更できます。

```
# zfs set volsize=2G rpool/dump
# zfs get volsize rpool/dump
```

NAME	PROPERTY	VALUE	SOURCE
rpool/dump	volsize	2G	local

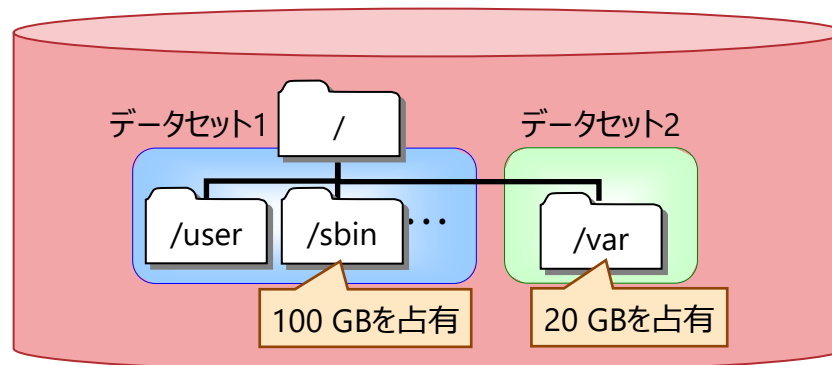
←変更
←確認

ダンプレベルを変更すると、ダンプデバイスの容量がチェックされます。容量が不足している場合は、容量に合わせてダンプデバイスの値を適宜変更してください。

■ /var領域の設定

- OSのインストール時、デフォルト設定で/var領域をルートファイルシステムとは異なるデータセットに作成します。
- ✓ /var領域を別データセットにすることのメリット
 - /var領域のデータセット単位でスナップショットを作成できます。
 - /var領域のデータセットに対して、領域の使用上限を設定できます（プロパティの個別設定が可能）。
 - ログ運用のポリシーに利用できます（/var領域にログを集約して集中バックアップなど）。

/var領域の設定例



• Solaris 10では、インストール時に別データセット構成を選択する必要がありました。Solaris 11からは、デフォルト設定で/var領域が別データセットとなります。

■ rpool/VARSHAREファイルシステム

- Solaris 11をインストールすると、ルートプール（rpool）配下にrpool/VARSHAREファイルシステムが自動的に作成され、/var/shareにマウントされます。

■ VARSHAREの役割

- 複数のBEで/VARSHAREファイルシステムを共有することにより、すべてのBEで必要となる/varディレクトリ内の容量を減らします。

✓ ファイルシステムの確認

```
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
~ (省略) ~
rpool/VARSHARE      192M   472G   6.66M   /var/share
```

✓ /var/shareの確認

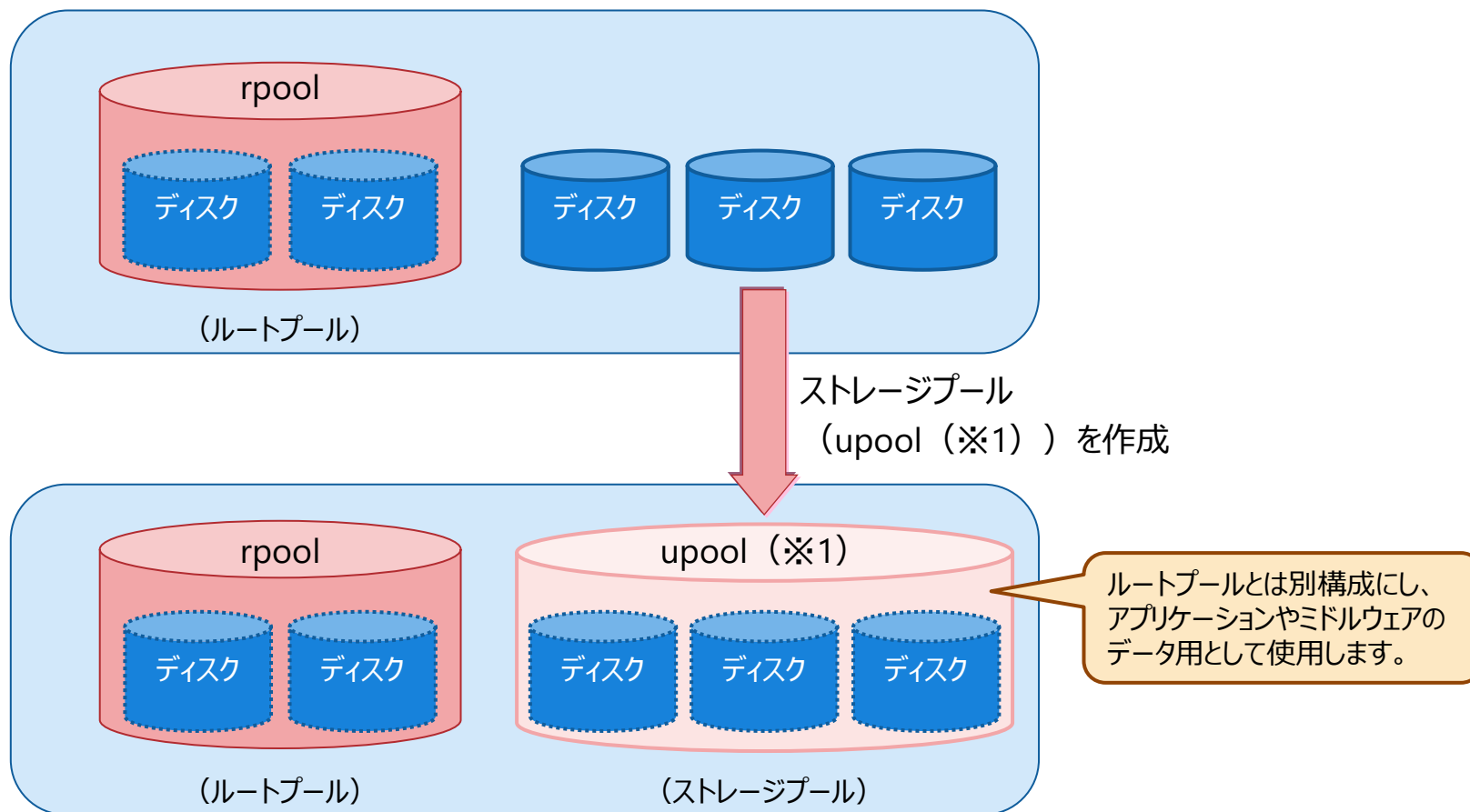
```
# ls -l /var/share
total 54
drwxrwxr-x  2 root    sys      4 Oct 19 12:02 adm
drwxr-xr-x  2 root    root     30 Jan 31 14:18 audit
drwxr-xr-x  2 root    root      2 Jan 31 14:15 beadm
drwxr-xr-x  3 root    root      3 Oct 19 11:58 compliance
drwx----- 2 root    sys      2 Oct 19 11:57 cores
drwx----- 2 root    sys      2 Oct 19 11:57 crash
drwx----- 3 root    root      3 Oct 19 11:59 diag
drwxr-xr-x  2 root    root      2 Oct 19 11:57 kvol
drwx----- 6 root    root     13 Jan 31 14:18 ldomsmanager
drwxrwxrwt  3 root    mail      4 Jan 10 13:43 mail
drwxr-xr-x  2 root    bin       2 Oct 19 11:57 nfs
drwxr-xr-x  3 root    root      3 Oct 19 11:58 pkg
drwxr-xr-x  2 root    sys       2 Oct 19 11:57 smb
drwxr-xr-x  3 root    root      3 Oct 19 11:58 sstore
drwxr-xr-x  2 daemon  daemon    2 Oct 19 11:57 statmon
lrwxrwxrwx  1 root    root      6 Oct 19 11:57 tmp -> ../tmp
drwxr-xr-x  5 root    sys       8 Oct 19 13:12 user
drwxr-xr-x  2 root    sys       3 Jan 10 19:56 zones
```



- BEは、Solaris 11の新機能であるBoot Environmentのことです。
→BEの詳細については、『[Oracle Solaris 11を使ってみよう](#)』を参照してください。

■ ストレージプール（ユーザー領域）の作成

アプリケーションなどのデータを保管するために、ルートプールとは別のディスクを使用して、ユーザー領域のストレージプールを作成することを推奨します。

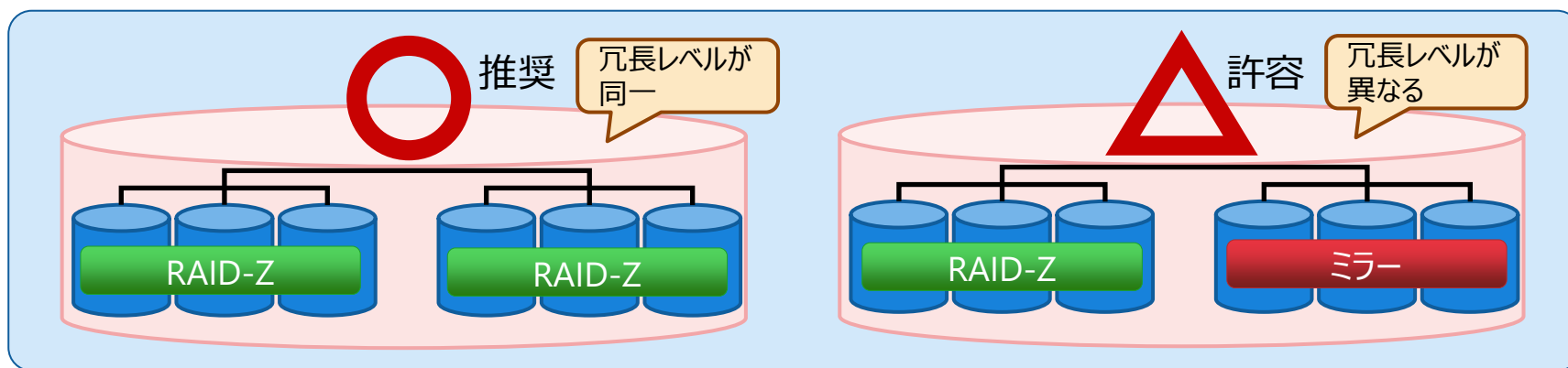


※1：システム領域（OS起動用）のルートプールの名称は「rpool」固定ですが、ユーザー領域のストレージプールは、任意の名称を設定できます。

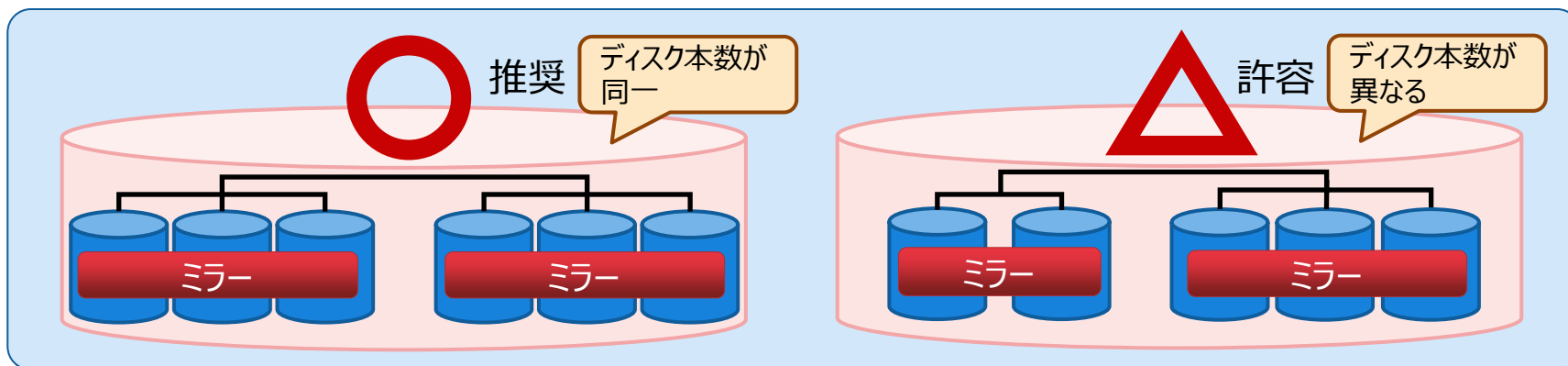
■ ストレージプールの推奨構成

ストレージプールは複数のディスクグループで構成可能ですが、信頼性やパフォーマンスを考慮し、ディスクグループの冗長レベルは同一にすることを推奨します。

- ✓ ストレージプール内は冗長レベルを同一にする



- ✓ ディスクグループの構成（各ディスクの容量や本数など）を同一にする

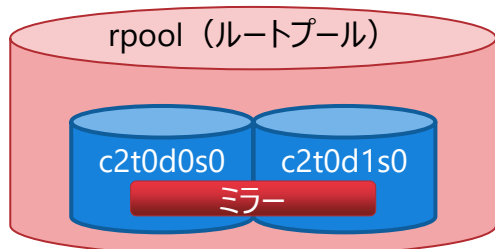


《参考》OSインストール時のルートプールの構成

Solaris 10とSolaris 11では、初期状態（OSインストール時）のルートプールの構成が異なります。

Solaris 10

- ✓ 複数ディスクによるルートプール構成



OSのインストール時にミラー構成が可能です。

- ディスクの選択 -

この画面では、Solaris ソフトウェアをインストールするディスクを選択します。

まず「推奨する最小値」フィールドの値を確認してください。

これは、選択したソフトウェアをインストールするために必要な容量の概算値です。

ZFSの場合は複数のディスクがミラーとして構成されるため、選択したディスクまたは ディスク内のスライスが「推奨する最小値」の値を超えていなければなりません。

注: ** は現在のブートディスクを表します

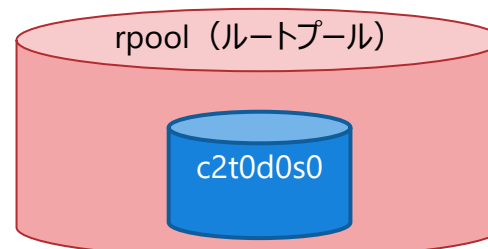
ディスクデバイス

空き領域

[X]	c0t0d0	69994 MB
[X]	c1t0d0	69994 MB

Solaris 11

- ✓ 単一ディスクによるルートプール構成



OSのインストール時のミラー構成は不可のため、必要に応じてインストール後に実施します。

ディスク

Oracle Solaris をどこにインストールしますか?
最小サイズ: 4.1GB 推奨最小: 6.1GB

タイプ サイズ (GB) ブート デバイス

unknown	29.9	+	c2t0d0
unknown	29.9		c2t0d1
unknown	19.9		c2t0d2
unknown	29.9		c2t0d4

5.ファイルシステムの管理

ファイルシステムの作成とマウント、およびファイルシステムの動作を制御するプロパティの機能を説明します。

ファイルシステムは、階層構造で管理されます。

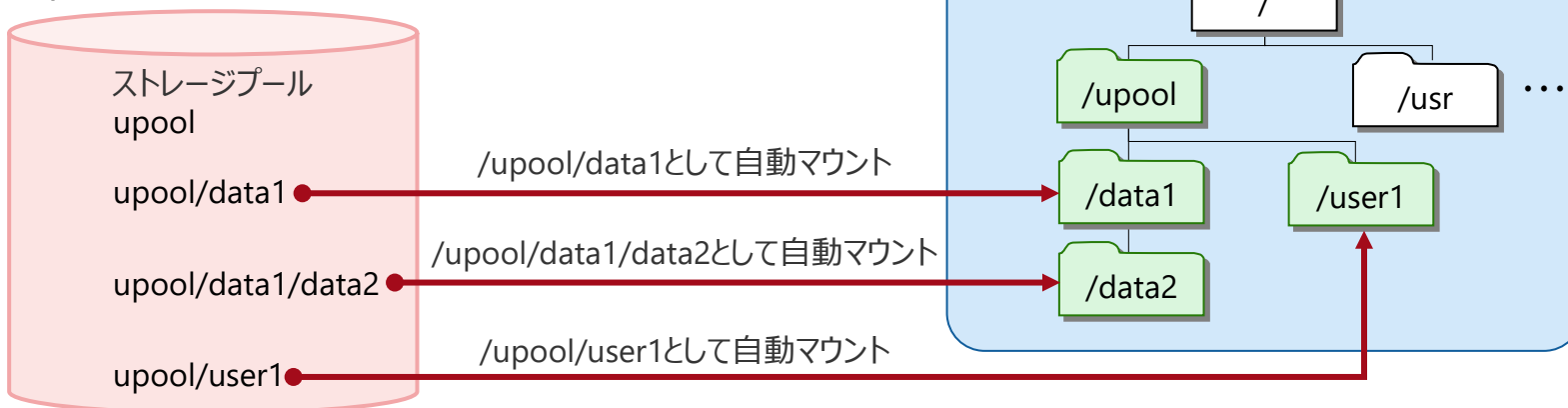
■ 最上位のファイルシステム

- ストレージプールを作成すると、最上位のファイルシステムが自動で作成されます。



■ 下位のファイルシステム

- 最上位のファイルシステムと異なり、手動で作成します。
- 通常、作成したファイルシステムと同じ名称のディレクトリが作成されます。



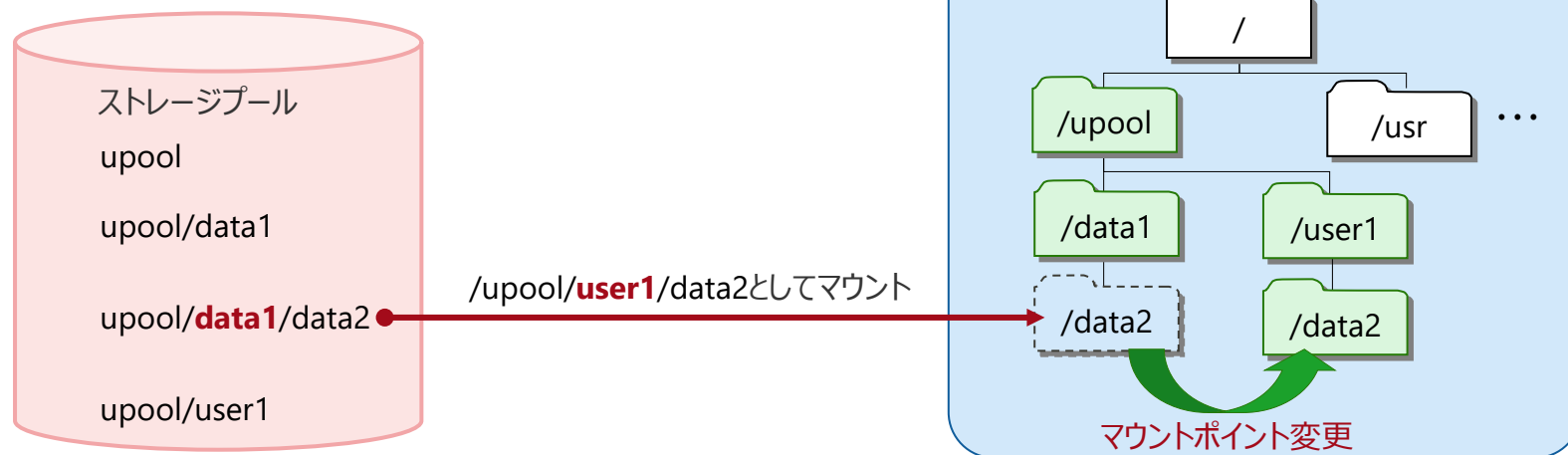
• 各ファイルシステムは、上位のファイルシステムとディスク領域を共有します。

■ ZFSのマウント基本動作

- ファイルシステムを作成すると、ファイルシステム名と同じパス上にマウントポイントが作成され、自動でマウントされます。
- ファイルシステムを削除すると、自動でアンマウントされます。
- mountpointプロパティの値を変更すると、マウントポイントを変更できます。
- 従来のUFSの方法でマウントすることもできます（レガシーマウント）。

✓ mountpointプロパティによるマウントポイントの変更

例：mountpointプロパティに/upool/**user1**/data2を設定



💡 • NFS共有した領域に対して、NFSサーバが新規にファイルシステムを作成した場合、NFSクライアントは自動的にファイルシステムを認識（マウント）します。
→詳しくは、「[《参考》ZFSミラーマウント](#)」を参照してください。

■ ZFSのプロパティ

- ZFSの持つ各種プロパティの値を設定することで、マウント、共有設定、および使用容量の制限など、ファイルシステムの動作を制御できます。

■ プロパティの種類

■ ネイティブプロパティ

- ZFSの動作を制御するプロパティです。以下の2種類があります。
 - 読み取り専用ネイティブプロパティ（プロパティは継承されません）
 - 設定可能ネイティブプロパティ（割り当て制限[quota]と予約[reservation]以外は継承されます）

■ ユーザープロパティ

- ZFSの動作には影響しないプロパティです。
ファイルシステムの用途や注釈などを残しておくために、ユーザーが任意に設定できます。



- 本書では、ユーザープロパティの詳細を説明していません。
- 以降の説明内の「プロパティ」は、「ネイティブプロパティ」を意味します。

■ プロパティの継承

- 親のファイルシステムで設定されたネイティブプロパティの値が、下位のファイルシステムに引き継がれます。

ただし、quotaプロパティ（割り当て制限）とreservationプロパティ（予約）、および読み取り専用ネイティブプロパティは引き継がれません。

- 以下のタイミングで、プロパティの継承が行われます。
 - 親のファイルシステムにプロパティの値が設定されたとき
 - プロパティが設定されているファイルシステム配下に新たにファイルシステムを作成したとき

- データセットのプロパティごとに、継承の有無を確認できます。

※ 以下のコマンド実行結果は、upool配下のcompressionプロパティを確認する例です。コマンド実行結果のSOURCEの値によって、プロパティの継承の状態を確認できます。

```
# 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
```

SOURCEの値	説明
local	プロパティが明示的にそのデータセットに設定されていることを示します。
inherited from data-name	プロパティがどの親[data-name]から継承されているかを示します。
default	プロパティが設定されていない状態を示します（上記以外の状態）。

ネイティブプロパティのうち、以下の代表的なプロパティの機能を紹介します。

- マウントポイントの変更
 - mountpointプロパティ
- NFS共有設定
 - share.nfsプロパティ
- ファイルシステムの割り当て制限と予約
 - quotaプロパティ
 - reservationプロパティ
- ユーザー／グループ割り当て制限
 - defaultuserquotaプロパティ
 - defaultgroupquotaプロパティ
 - userquota@userプロパティ
 - groupquota@groupプロパティ
- データ重複の検知・排除
 - dedupプロパティ
- データ暗号化
 - encryptionプロパティ

■ mountpointプロパティ（マウントポイントの変更）

マウントポイントを変更できます。また、レガシーマウント（UFSと同様の設定方法）の利用設定も行えます。

■ 設定例

- マウントポイントを指定します。

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

- レガシーマウントを利用します。

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

■ レガシーマウントの方法

- UFSファイルシステムと同じ方法でZFSファイルシステムのマウントを管理します。
- ZFSによる自動マウントは実行されないため、コマンド（mount/umount）で管理する必要があります。
upool/testを/zfs/fsへマウントする場合の例を、以下に示します。

- ①レガシーマウント用にmountpointプロパティを設定します。
- ②マウントポイントを作成します。

```
# mkdir /zfs/fs
```

- ③mountコマンドを実行します。

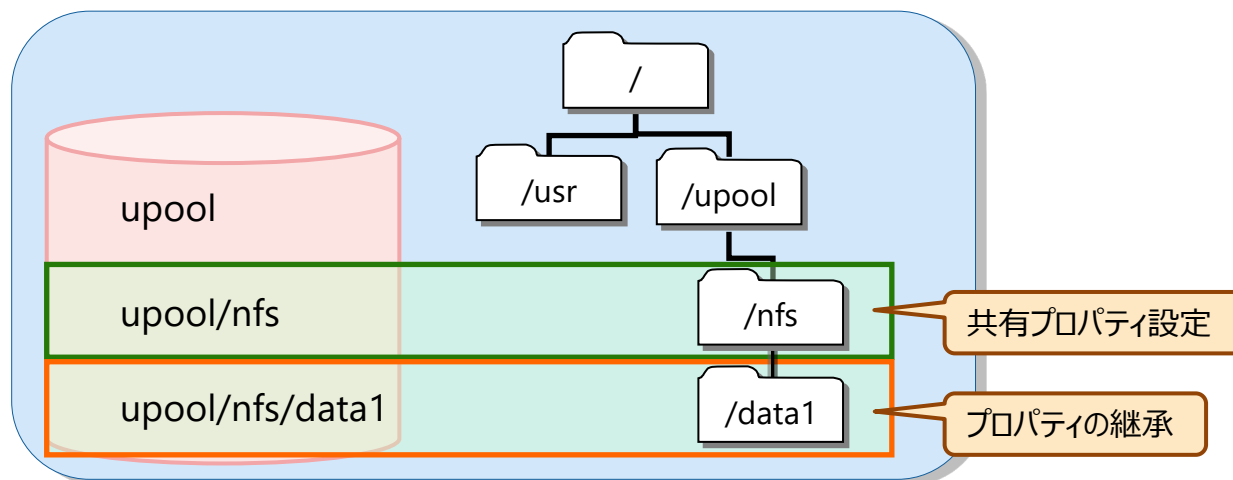
```
# mount -F zfs upool/test /zfs/fs
```

※/etc/vfstabファイルを編集し、OS起動時に自動マウントさせることもできます。実行例を以下に示します。

#device	device	mount	FS	fsck	mount	mount
#to mount	to fsck	point	type	pass	at boot	options
#						
upool/test	-	/zfs/fs	zfs	-	yes	-

■ share.nfsプロパティ（NFS共有設定）

- 共有設定によって、NFSでファイルシステムのデータをほかのサーバと共有させることができます。
- 共有設定を行ったファイルシステムの下位のファイルシステムにも、共有設定が継承されます。



※ Solaris 11.1以降で、sharenfsプロパティはshare.nfsプロパティに変更されました。

※ Solaris 11 11/11では、zfs set shareコマンドを使用して共有ファイルシステムを作成し、sharenfsプロパティを設定することで共有設定を有効にする方式でした。

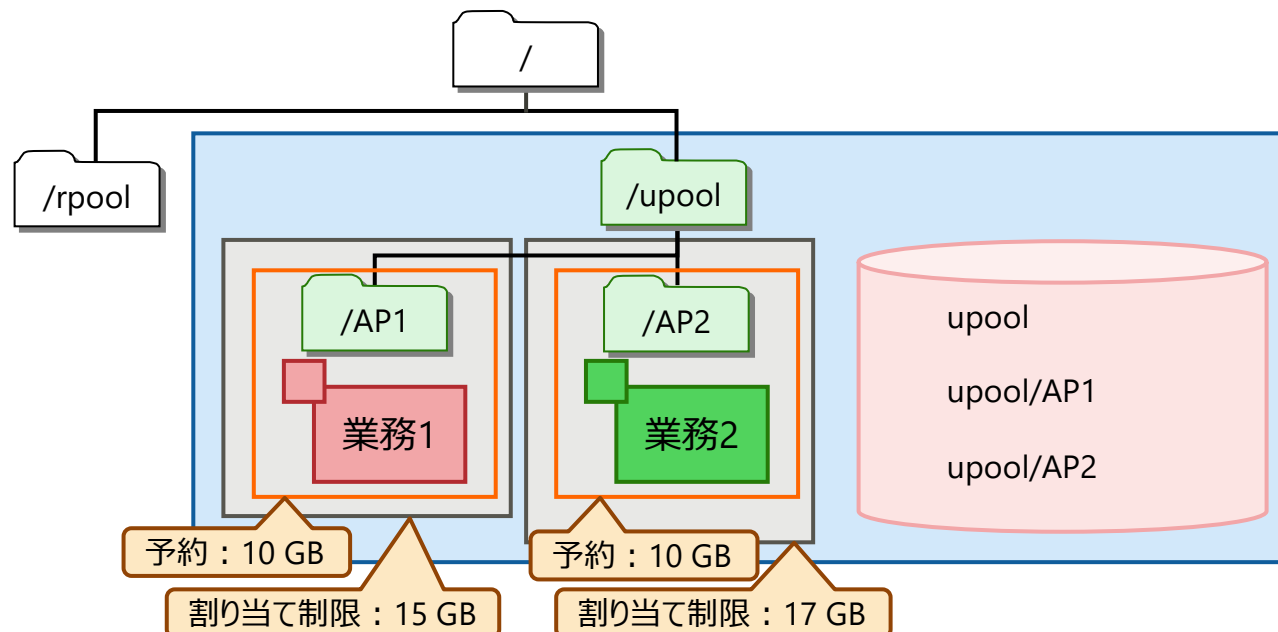
■ 留意事項

- 従来の共有設定ファイル（/etc/dfs/dfstab）を編集する必要はありません。
- ファイルシステムの削除時に、共有設定は自動的に解除されます。
- NFSのバージョンは 4.1（NFSv4.1）を使用します。
※Solaris 11.3 までのNFSのバージョンは 4（NFSv4）です。

■ quotaプロパティ、reservationプロパティ（ファイルシステムの割り当て制限と予約）

- quotaプロパティで、ファイルシステムの使用可能な最大容量（割り当て制限）を設定できます。
- reservationプロパティで、ファイルシステムが必ず使用できる容量（予約）を確保できます。

※ デフォルトでは、上記のプロパティは設定されていません。各ファイルシステムの使用可能な容量に制限はなく、予約されている容量もありません。



• UFSのパーティションとは異なり、ZFSでは「割り当て制限」と「予約」の値を変更することで、使用できる容量を容易に変更できます。

■ defaultuserquotaプロパティ、defaultgroupquotaプロパティ、userquota@userプロパティ、groupquota@groupプロパティ（ユーザー／グループ割り当て制限）

- ユーザー／グループに適用するデフォルトの使用容量の上限、および特定のユーザー／グループごとの使用容量の上限を設定できます。

プロパティ	説明
defaultuserquota=size none	デフォルトのユーザー割り当て制限
defaultgroupquota=size none	デフォルトのグループ割り当て制限
userquota@user= size none default	特定のユーザー割り当て制限
groupquota@group= size none default	特定のグループ割り当て制限

※ 上記の表のsizeは、容量の数値を指定することを意味します。下記の設定例を参照してください。

※ none は、割り当て制限を解除し、割り当て制限なしにします。

※ default は、割り当て制限のデフォルト値に設定します。

■ 設定例

- 1ユーザーが使用できる容量の上限を、デフォルトで25 GBに設定します。

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

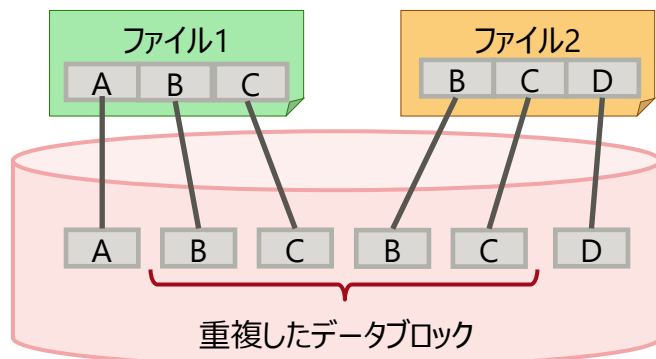
- ユーザー「user01」が使用できる容量の上限を50 GBに設定します。

```
# zfs set userquota@user01=50GB upool/nfs
```

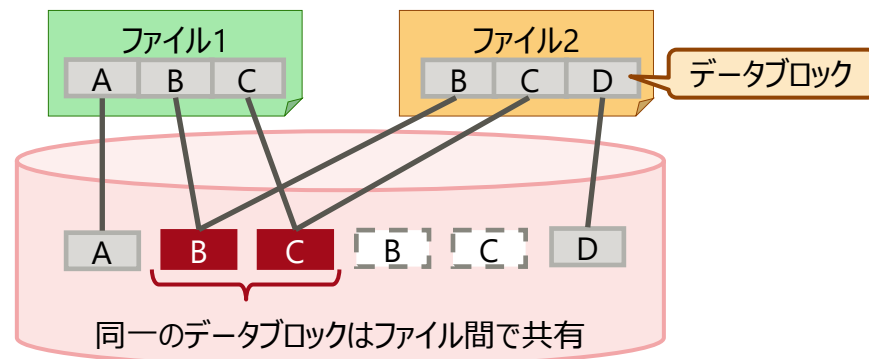
■ dedupプロパティ（データ重複の検知と排除）

- ファイル間で重複したデータ領域（データブロック）を検知して排除します。
- dedupプロパティの値を「on」に設定すると、重複検知・排除機能が有効になります。

✓ dedupプロパティoff（無効）※デフォルト



✓ dedupプロパティon（有効）



※ ブロック単位の重複検知・排除機能により、格納されているブロックと同一内容のブロックはオンライン（オンザフライ）で除去されます。

■ 設定例

- upool/data1内の重複検知・排除機能を有効にします。

```
# zfs set dedup=on upool/data1
```

■ 留意事項

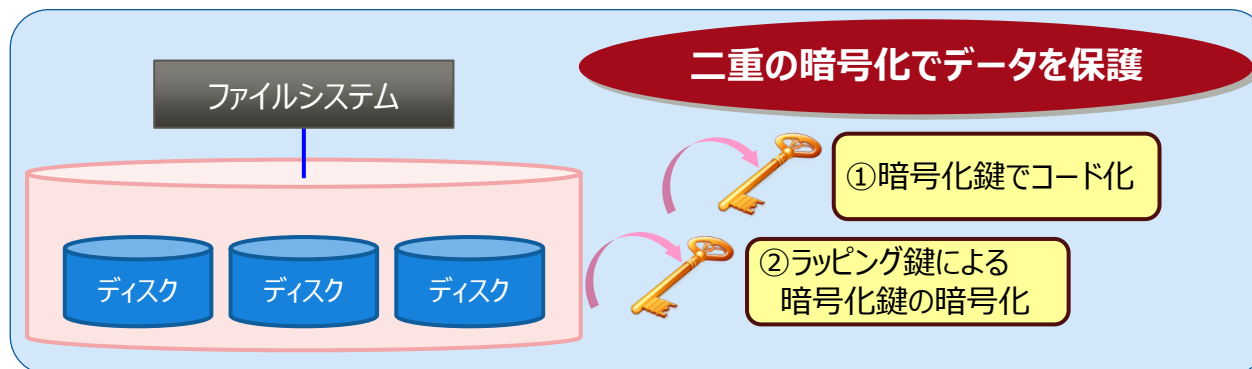
- dedupプロパティは、Oracle社のマニュアルを参照し、事前に考慮事項を確認してから有効にしてください。
『Oracle Solaris 11.4 での ZFS ファイルシステムの管理』（Oracle社）
https://docs.oracle.com/cd/E75431_01/html/E75197/index.html
「第7章 Oracle Solaris ZFS ファイルシステムの管理」-「ZFS のプロパティの概要」-「設定可能な ZFS ネイティブ プロパティ」-「dedup プロパティ」



• 仮想化やサーババックアップ時にディスク容量を有効活用でき、ストレージ利用効率の向上を実現できます。

■ encryptionプロパティ（データ暗号化）

- データのコンテンツを保護するため、ファイルシステムを暗号化鍵でコード化します。



※ ZFSのデータセット単位で、暗号化ポリシーが設定されます。

■ 設定例

- fj/home/userを暗号化します。暗号化鍵を8文字以上で入力します。

```
# zfs create -o encryption=on upool/data1
Enter passphrase for 'upool/data1':***** ←8文字以上
Enter again:*****
```

■ 留意事項

- 暗号化属性値は、データセット作成時に指定します。
- 鍵の長さが 128、192、および 256のAES（Advanced Encryption Standard）を使用して暗号化されます。
- 暗号化はZFSファイルシステム作成時に設定できますが、暗号化設定は変更できません。

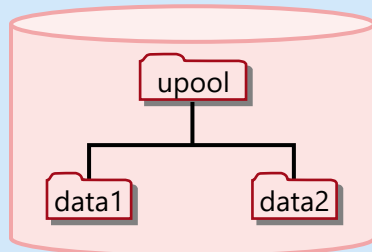


• ZFSでは、ファイルシステムの暗号化に加え、暗号化鍵をラッピング鍵で暗号化することにより、UFSに比べてセキュリティ機能が強化されました。

《参考》ZFSミラーマウント

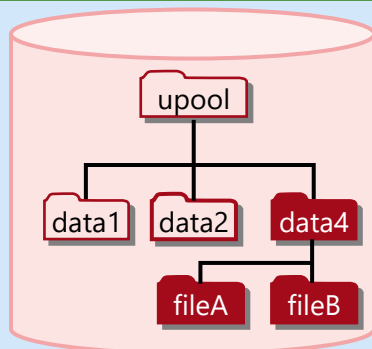
NFSサーバが新規にファイルシステムを作成すると、NFSクライアントが自動的にファイルシステムを認識（マウント）する機能です。

NFSサーバ



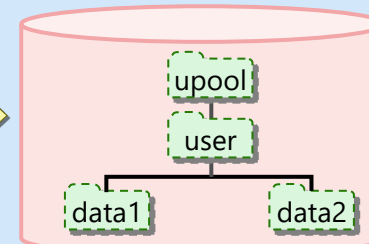
③ NFSサーバで新規にファイルシステムを作成

```
# zfs create upool/data4
# zfs create upool/data4/fileA
# zfs create upool/data4/fileB
```



NFSクライアント

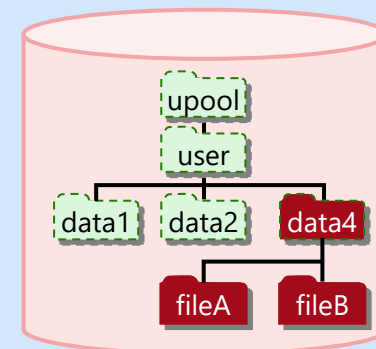
① マウントポイント/upool/userにマウント



② NFSクライアント側でlsコマンドを実行し、マウントしたディレクトリ配下のファイルを確認

```
# ls /upool/user
data1 data2
```

④ NFSサーバで作成されたupool/data4ファイルシステムに対して自動マウント



⑤ NFSクライアント側でlsコマンドを実行し、マウントしているディレクトリ配下のファイルを確認

```
# ls /upool/user
data1 data2 data4
# ls /upool/user/data4
fileA fileB
```

6.ZFSの機能

「スナップショット」や「バックアップ／リストア」など、ZFSの運用管理に使用する機能を説明します。

運用時に利用する、ZFSの基本機能を紹介します。

■ スナップショット

- データセットの読み取り専用コピーです。

■ ロールバック

- スナップショットを使用したファイルシステムの復元機能です。

■ クローン

- スナップショット機能を活用したデータセットのコピーです。

■ バックアップ／リストア

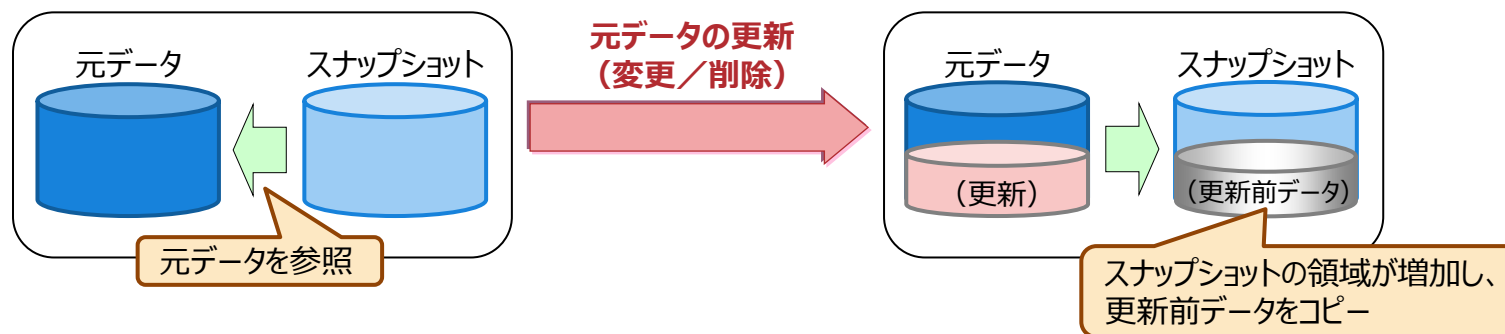
- オンラインでファイルシステムのバックアップ／リストアを実行できます。

■ ミラーディスクの分割

- ストレージプールからミラーディスクを分割し、同一のデータを保持したストレージプールを作成します。

■ スナップショットとは

- データセット（ファイルシステムまたはボリューム）の読み取り専用コピーのことです。スナップショットは瞬時に作成され、スナップショット作成直後はストレージプール内の領域を使用しません。
- スナップショットの作成元の、データセット内のデータが更新（変更／削除）されると、スナップショットの領域に更新前のデータ分の容量が増加します。



※ スナップショットを作成している場合、元データを削除しても全体のディスク領域は減少しないため、注意が必要です。

■ 留意事項

- 作成できるスナップショットの理論上の最大数は、 2^{64} です。
- スナップショットは、作成元のファイルシステムと同じストレージプールのディスク領域を使用します。
- スナップショット作成対象にZFSボリュームが含まれる場合、以下の式を満たす空き容量が必要です。
親プールの空き容量 > スナップショット作成対象に含まれるZFSボリュームの使用量の合計

■ スナップショットと連携する機能

■ スナップショット作成時へのロールバック

- ファイルシステムを、スナップショット作成時の状態に戻すことができます。
- スナップショット作成後に加えられたすべての変更が破棄されます。

■ クローンの作成

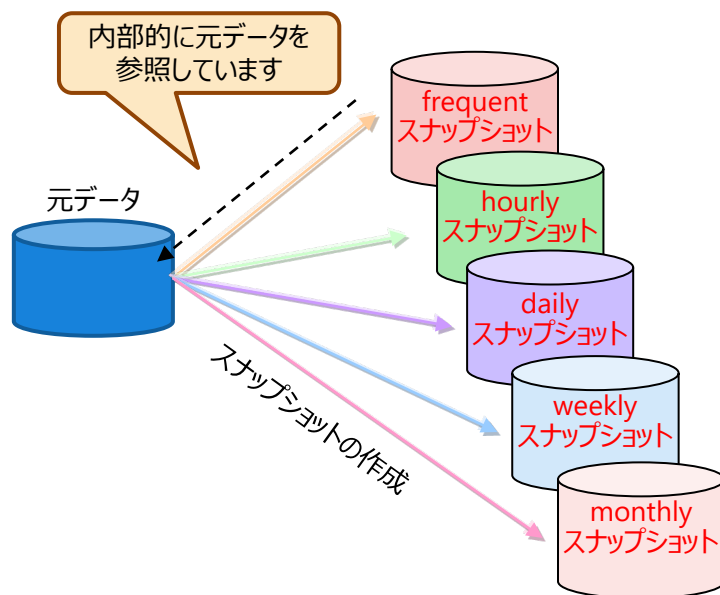
- スナップショット作成時の状態と同一の環境を作り、テストなどに使用できます。
- スナップショットの場合と同様に、クローンは瞬時に作成されます。作成時は、ストレージプールの領域を消費しません。

■ バックアップ／リストア

- スナップショットからストリームを作成し、ファイルへ送信できます（バックアップ）。
- 送信したストリームを受信することで、ストリームからファイルシステムを復元できます（リストア）。

■ 自動スナップショット

- ・ タイムスライダーサービスを使用して、自動的にスナップショットを作成できます。
- ・ 自動的に作成したスナップショットを閲覧し、リストアすることもできます。



✓ スナップショットの作成基準

	作成タイミング	保持可能な数
frequentスナップショット	15分ごと	4
hourlyスナップショット	毎時	24
dailyスナップショット	毎日	31
weeklyスナップショット	毎週	7
monthlyスナップショット	毎月	12

■ 留意事項

- ・ 自動的に作成されたスナップショットは、事前に設定されたファイルシステム使用量の割合に基づいて、古いものから削除されます。
- ・ タイムスライダーサービスは、デフォルト設定では無効です。



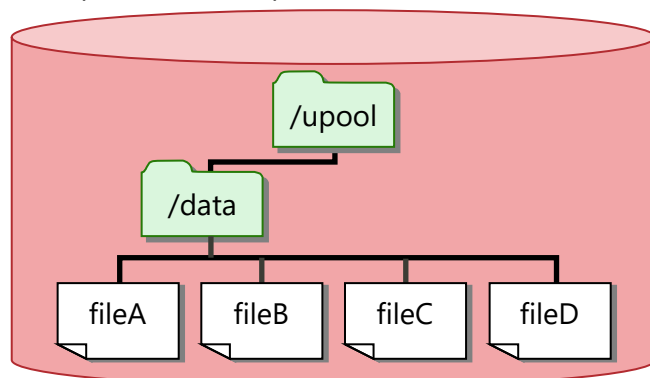
・ データの更新が頻繁なファイルシステムに対して、タイムスライダーサービスを有効化することを推奨します。

■ スナップショット差分表示 (zfs diff)

- 2つのスナップショットの差分を表示できます。

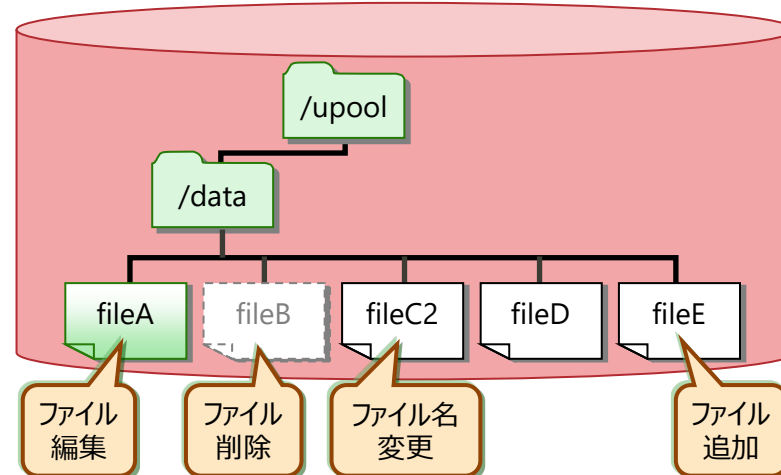
✓ スナップショットA (データ更新前)

upool/data@snap1



✓ スナップショットB (データ更新後)

upool/data@snap2



✓ zfs diffコマンドの実行

```
# zfs diff upool/data@snap1 upool/data@snap2
M      /upool/data/
M      /upool/data/fileA
-      /upool/data/fileB
R      /upool/data/fileC -> /upool/data/fileC2
+      /upool/data/fileE
```

✓ スナップショットの差分表記

- M : ファイルまたはディレクトリの変更を示します。
- R : ファイルまたはディレクトリの名前が変更されたことを示します。
- : 古いスナップショットには存在し、新しいスナップショットには存在しないことを示します。
- + : 新しいスナップショットには存在し、古いスナップショットには存在しないことを示します。



• zfs diffコマンド実行時にスナップショットを1つだけ指定した場合、ファイルシステムの現在の状態との差分が表示されます。

■ スナップショット差分表示 (zfs diff) (前ページからの続き)

- -r オプションを指定すると、指定したスナップショットとすべての子スナップショットの差分を表示（再帰処理）できます。

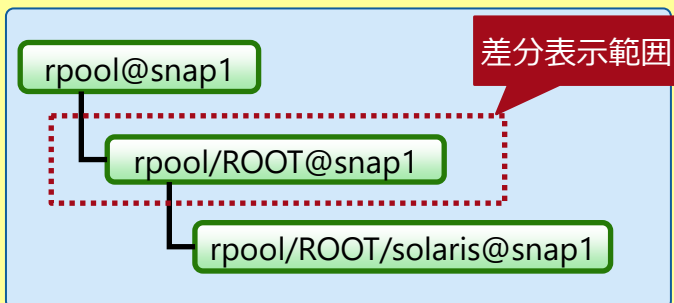
※ Solaris 11.3 から利用できる機能です。

✓ -r オプションなし

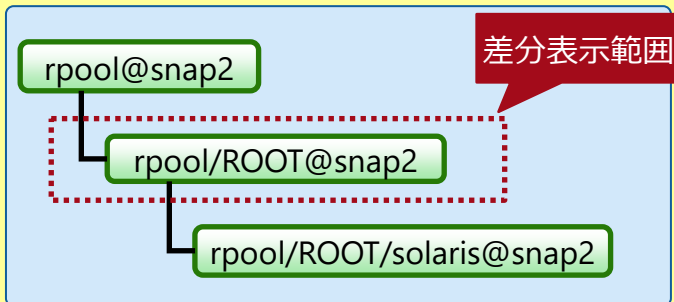
実行コマンド :

```
zfs diff rpool/ROOT@snap1 rpool/ROOT@snap2
```

•snap1



•snap2

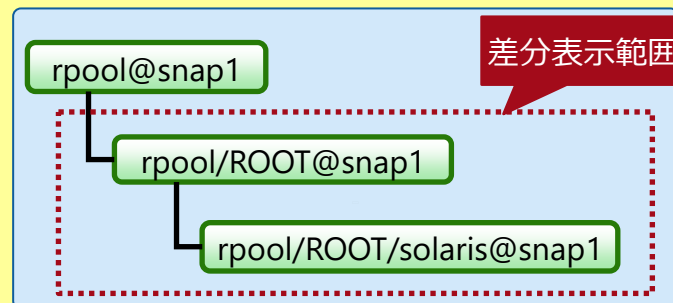


✓ -r オプションあり

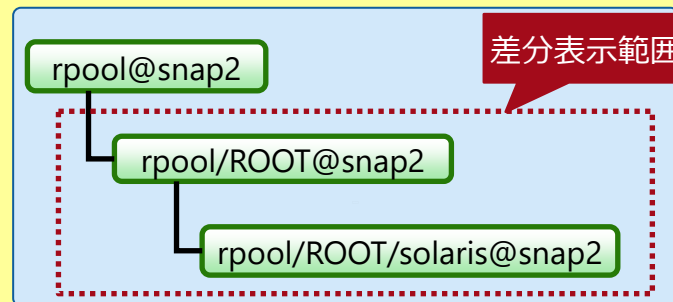
実行コマンド :

```
zfs diff -r rpool/ROOT@snap1 rpool/ROOT@snap2
```

•snap1



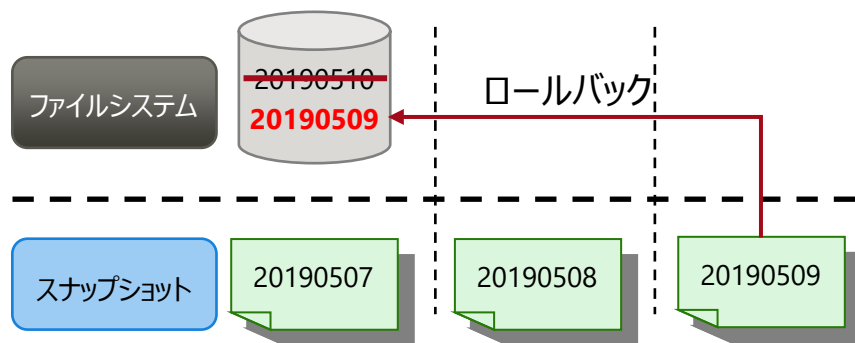
•snap2



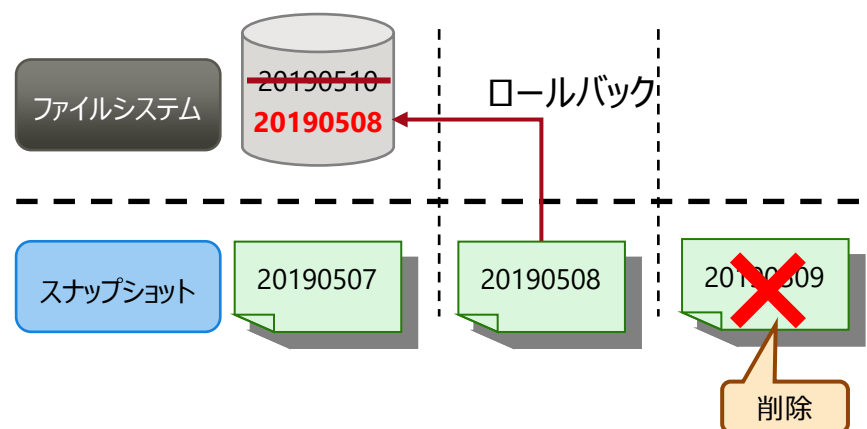
■ ロールバックとは

- 特定のスナップショットを作成した以降に加えられたすべての変更を破棄し、ファイルシステムをスナップショット作成時の状態に戻せます。データを変更する前にスナップショットを作成しておくことで、誤ってデータを削除した場合でも、変更前の状態に戻すことができます。

✓ 最新のスナップショットでロールバック



✓ 特定のスナップショットでロールバック



■ 留意事項

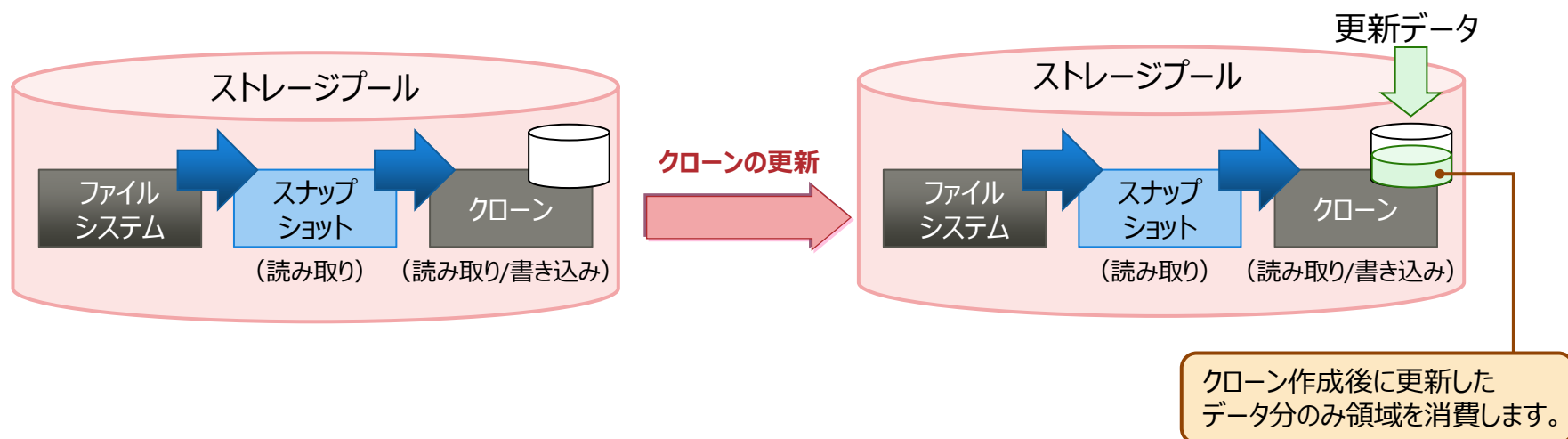
- デフォルトでは、最新のスナップショット以外にはロールバックできません。
- 最新のスナップショット以外にロールバックするには、対象のスナップショットが最新の状態になるように、対象のスナップショットより新しいスナップショットをすべて破棄する必要があります。
- 対象のスナップショットより新しいスナップショットのクローンが存在する場合は、クローンも破棄する必要があります。



• ロールバック実行時に「-r」オプションを指定することで、対象のスナップショットより新しいスナップショットを削除できます。

■ クローンは

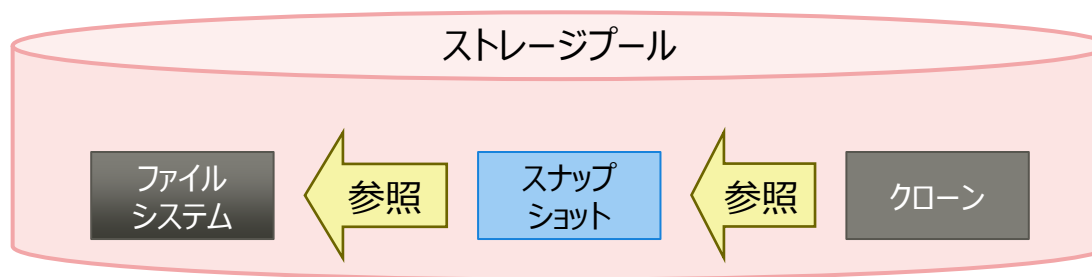
- 簡易に作成できるファイルシステムやボリュームのコピーです。
- 単純なデータのコピーとは異なり、スナップショットと同様に瞬時に作成され、作成直後はディスク領域を消費しません。
- クローンは、作成元のファイルシステム、またはボリュームのスナップショットから作成します。



- クローンはスナップショットからのみ作成します。また、クローンのスナップショットを作成することもできます。ただし、作成元のデータセットのプロパティは継承されません。
- Solarisゾーン環境を複製する場合、ZFSのクローンを使用するため、瞬時に複製できます。データ更新部分のみ、ディスクを消費します。

■ 依存関係

- 元のファイルシステム、スナップショット、およびクローンには依存関係があります。
- クローンが存在する間は、クローンの作成元のスナップショットを削除できません。スナップショットを削除する場合、先にクローンを削除する必要があります。



■ 依存関係の確認方法

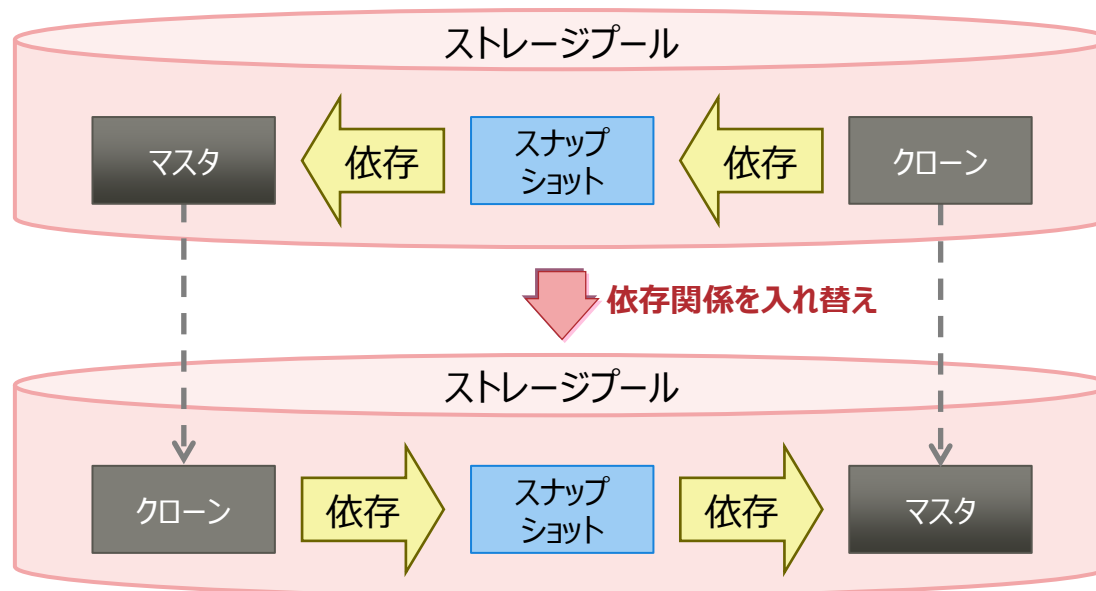
- originプロパティを使用して依存関係を確認できます。

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

※ ファイルシステムupool/fileとスナップショットupool/data@nowとの間に、依存関係があることが確認できます。

■ マスタとクローンの入れ替え

- クローンと元のファイルシステム（マスタ）の依存関係を逆にして、クローンをマスタにすることができます。
- マスタとクローンを入れ替えることで、クローン環境で加えた変更をマスタに反映させることができます。



• システムに影響があるデータを編集する際に、事前にクローンを作成し、クローンのデータを編集後に問題がなければ、マスタとクローンと入れ替えることで、安全にデータを編集できます。

■ バックアップ（ストリームの送信）

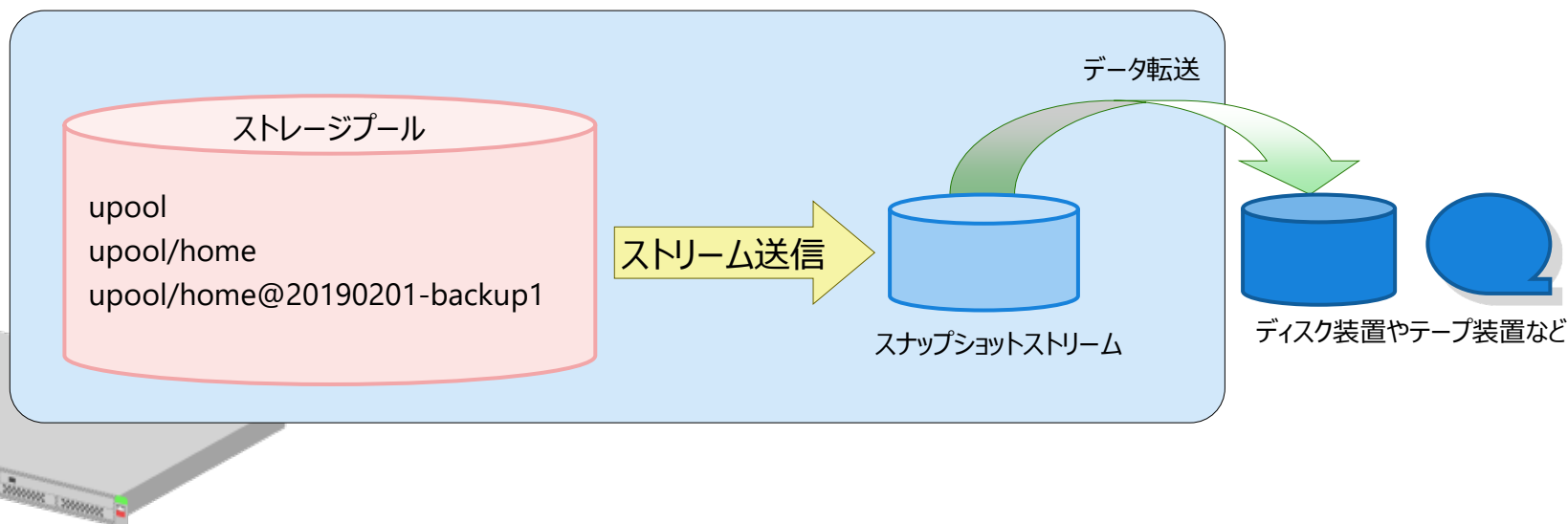
- ZFSでは、スナップショットからストリームを生成して送信することで、ファイルシステムのバックアップを行います。

なお、ストリームの生成時や送信時に、OSを停止する必要はありません。

※ ストリームとは、複数のデータを連続する1つのデータとして扱うためのものです。

■ バックアップの種類

- フルバックアップ（スナップショット全体を送信）
- 増分バックアップ（スナップショットの増分を送信）

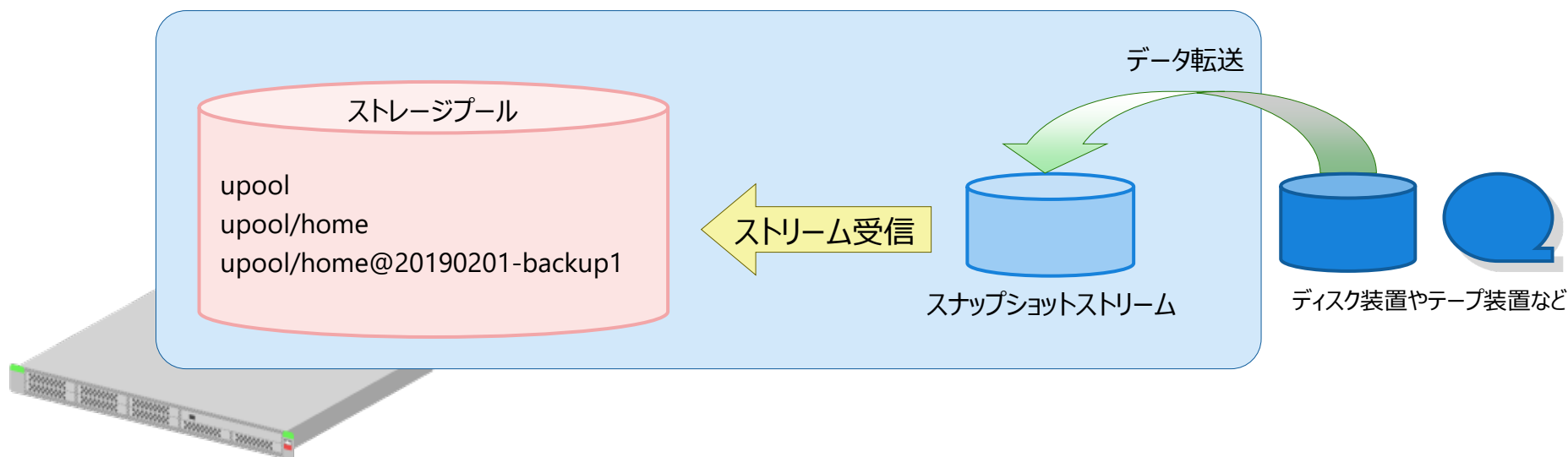


■ リストア（ストリームの受信）

- ZFSでは、バックアップで生成・送信されたストリームを受信することで、ファイルシステムのリストアを行います。

ストリーム送信時と同様に、OSを停止する必要はありません。

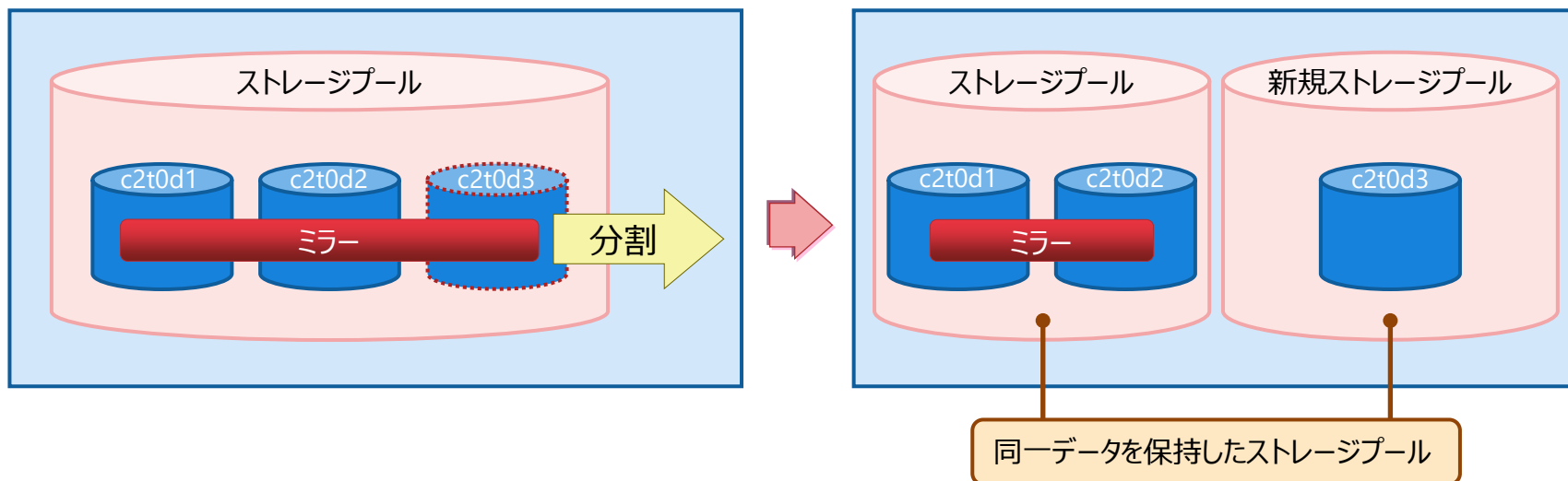
※システム領域（ルートプール）をリストアする際は、OSの停止が必要です。



- 💡 • リストア中（ストリーム受信中）は、対象のファイルシステムとその配下のファイルシステムにアクセスできません。
- 特定のスナップショットストリームを指定してリストアすることができます。ロールバックとは異なり、対象のスナップショットが最新の状態になるように、対象よりも新しいスナップショットを削除する必要はありません。

■ zpool splitコマンドによるミラーディスクの分割

- ミラー構成のストレージプールからディスクを分割し、分割元と同一のファイルシステムやデータを保持した新規ストレージプールを作成できます。



- 分割したディスクをインポートすることで、元のZFSストレージプールと同一のファイルシステムとデータを保持した新規ストレージプールを作成することができるため、バックアップとして利用できます。
- 分割元のデータの冗長性を保つため、3本以上のミラーディスクで構成されているストレージプールで実施することを推奨します。

■ zpool splitコマンドの特長

■ 容易なディスクの分割

- OSの停止や再起動を行うことなく、ミラー構成のストレージプールから数秒でディスクを分割します。

■ 同一データを保持したストレージプールの作成

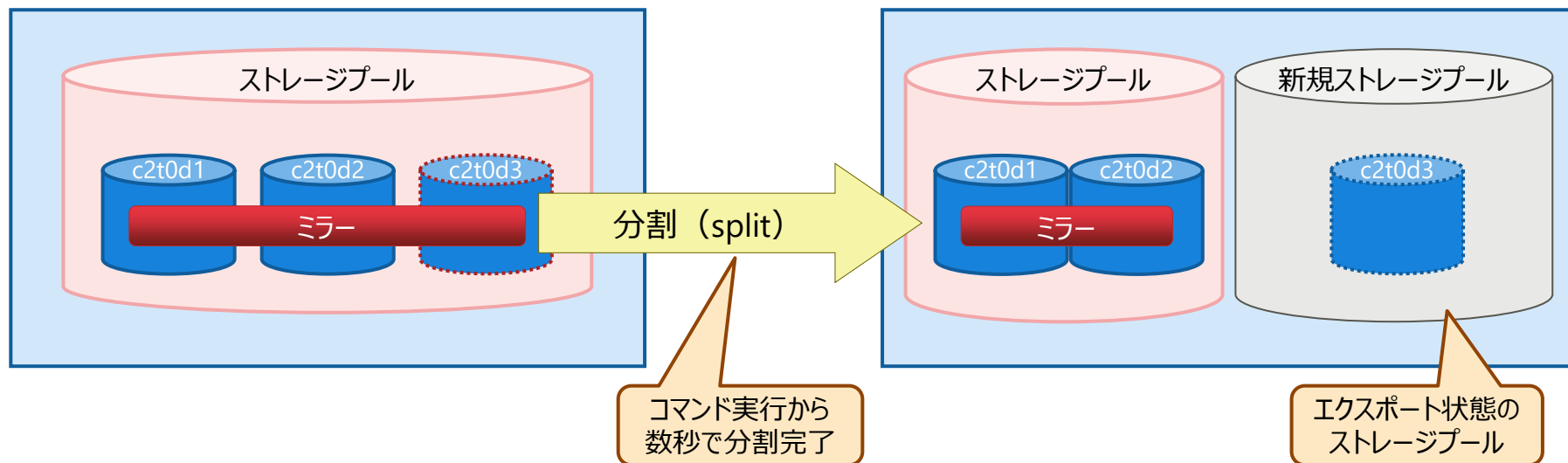
- 分割元のストレージプールと同一のファイルシステムとデータを保持した新規ストレージプールが作成できます。

■ 別サーバへのデータ移行

- 分割したディスクを別サーバへ接続することで、ストレージプールのデータ移行が容易に行えます。

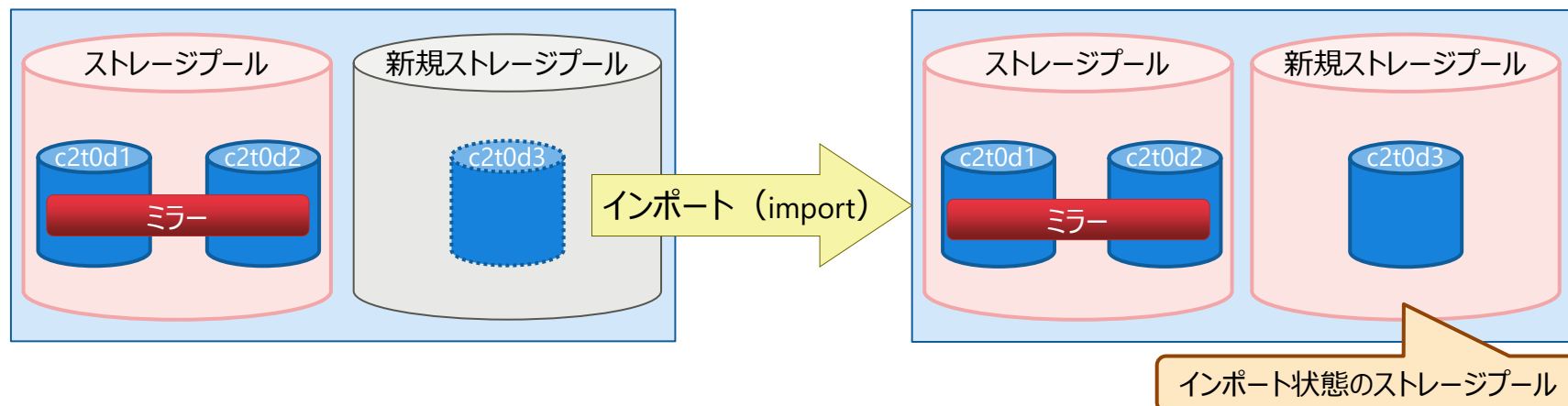
■ 容易なディスクの分割

- zpool split コマンド実行後、ミラー構成のディスクを数秒で分割すると同時に、新規ストレージプールが作成されます。
- 作成されたストレージプールはエクスポート状態となるため、分割後はOSから認識されません。



■ 同一データを保持したストレージプールの作成

- ミラー構成のディスクを分割して作成されたエクスポート状態の新規ストレージプールは、importコマンドを実行することでOSから認識されます。



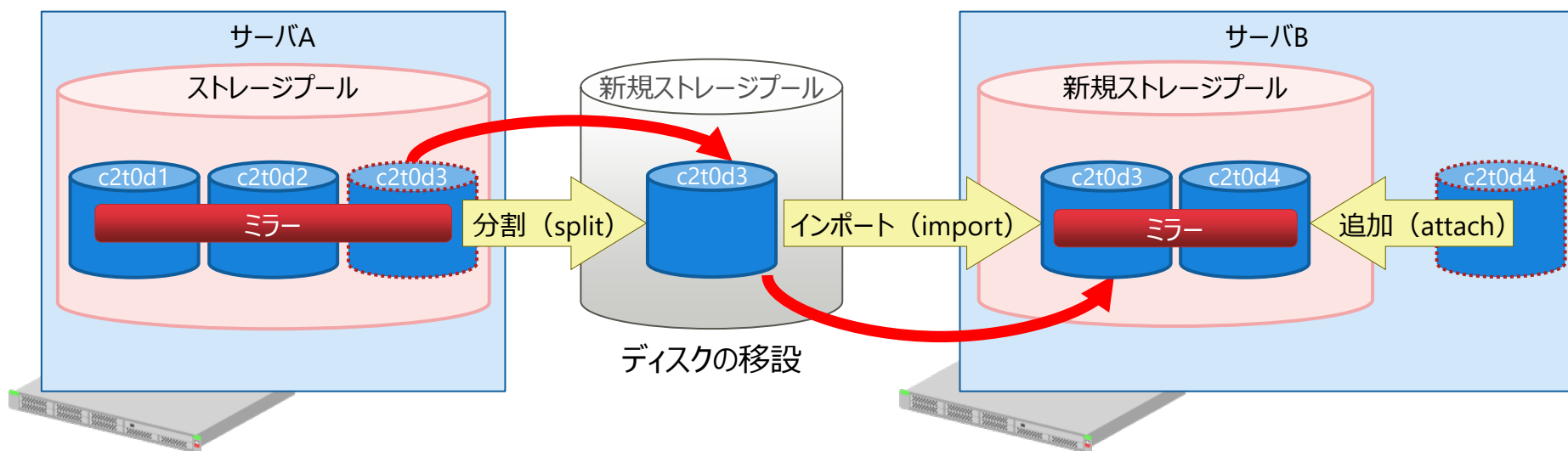
- 新規ストレージプールは、元のストレージプールと同一のファイルシステムとデータを保持しています。



※ ただし、ストレージプール名は、元のストレージプール名と同一にはできません。

■ 別サーバへのデータ移行

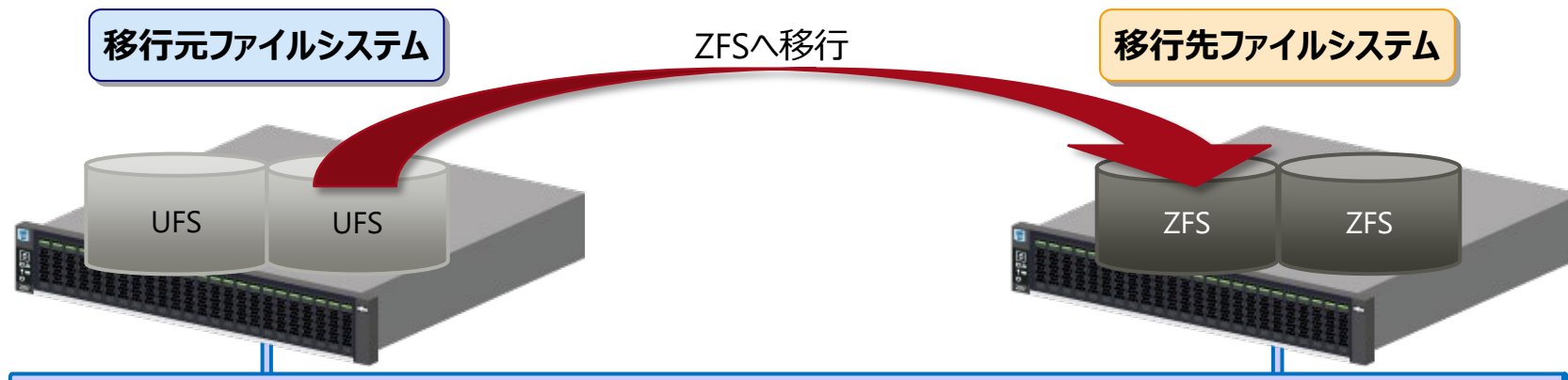
- 分割したディスクを別サーバへインポートすることで、ストレージプールの移行が容易に行えます。



• 分割 (split) 後のストレージプールはエクスポート状態のため、そのままディスクを移設し、別サーバ上でインポートできます。

《参考》UFSからZFSへのデータ移行

ZFS Shadow Migrationを利用して、OSを停止することなくUFSからZFSへ移行できます。



✓ 移行元ファイルシステム側の設定

- ・ 事前に、移行するファイルシステムを読み取り専用（read-only）に設定します。

✓ 移行先ファイルシステム側の設定

- ・ 新規ファイルシステムを作成し、移行用のプロパティ（shadowプロパティ）に移行元のファイルシステム名を設定します。

- ・ 既存のファイルシステム（UFS、NFS、ZFS）をZFSに移行できます。
- ・ NFSの場合の移行時間は、ネットワーク帯域幅に依存します。
- ・ 移行元ファイルシステムがローカルファイルシステム、またはNFSの場合、移行先ファイルシステムのshadowプロパティの設定方法は以下のように異なります。

shadow=file://path - ローカルファイルシステムを移行する構文
shadow=nfs://host:path - NFSを移行する構文



- ・ ZFSへの移行について詳しくは、『Oracle Solaris 11 ZFS移行手順書』を参照してください。

<https://www.fujitsu.com/jp/sparc-technical/document/solaris/#solaris11zfs>

7.ZFSの運用

ZFSを運用するために、必要に応じて実施する処理を目的別で説明します。

目的別のZFSの運用手順を紹介します。

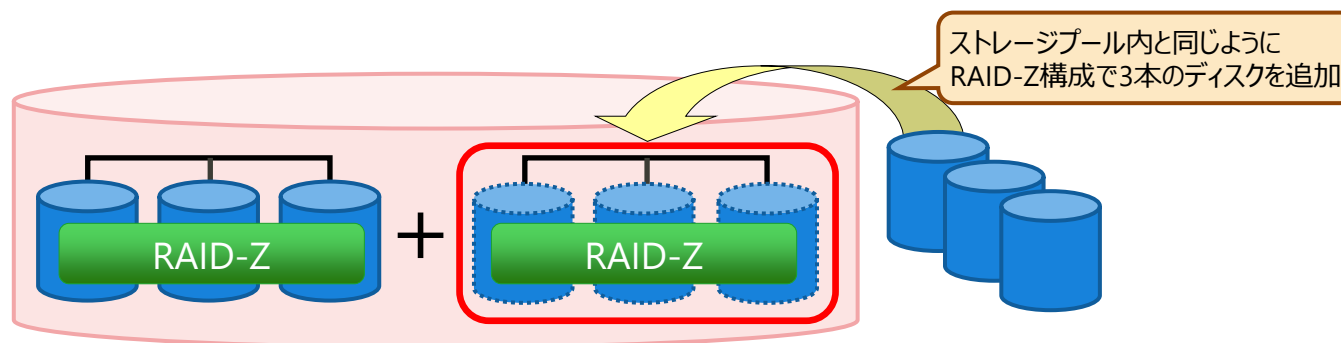
- ストレージプールの領域拡張
- ホットスペアの設定／管理
- ディスクの交換
- ストレージプールの移行
- ZFS操作の監視

■ ストレージプールの領域拡張でできること

- ストレージプールへ動的にディスク領域を追加できます。
追加した領域は、すぐに使用できます。

■ ディスクの追加

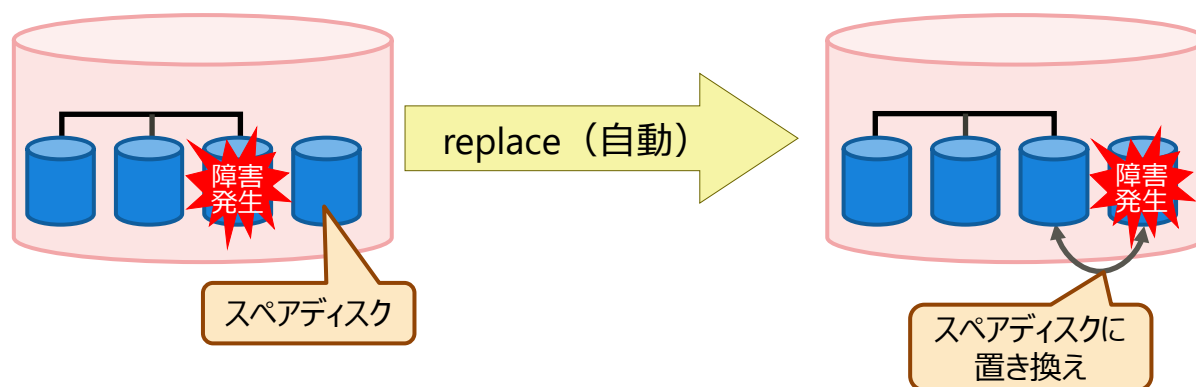
- ディスクグループ単位に、同一RAID構成の領域を拡張します。



- 領域の縮小はできません。また、一度追加したディスクグループのRAID構成は変更できません。
- 追加するディスクグループは、既存のディスクグループと同等の冗長レベルにすることを推奨します。

■ ホットスペアの設定／管理でできること

- スペアディスクを設定しておくことで、ストレージプール内のディスクに障害が発生した際に自動で置き換えられます。



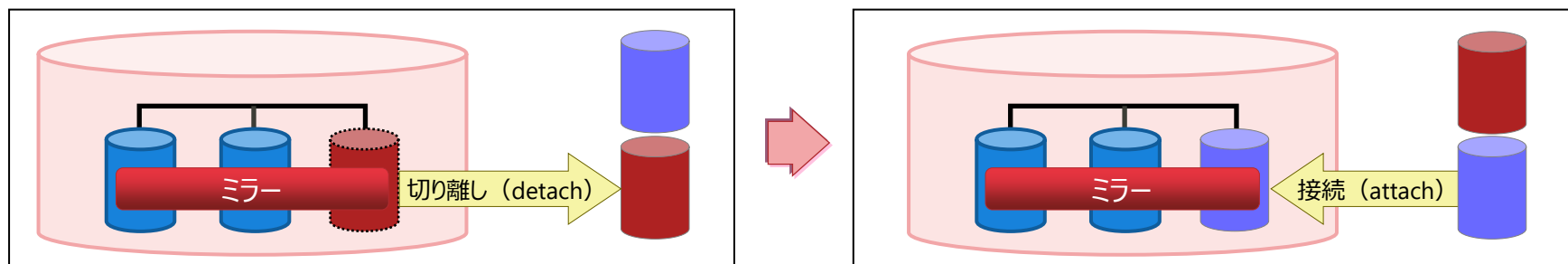
- RAID-Z、RAID-Z2、またはRAID-Z3を使用する場合は、ホットスペアを使用することを推奨します。
- ホットスペアに使用するディスクのサイズは、置き換わるディスク以上のサイズが必要です。

■ ディスクの交換でできること

- ミラー構成に応じて、「切り離し (detach) / 接続 (attach) 」と「置き換え (replace) 」のどちらかの方法でディスクを交換できます。

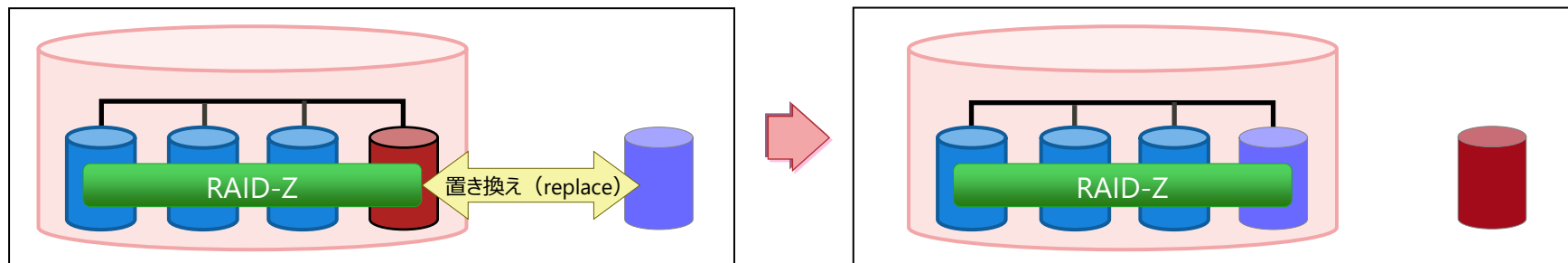
■ 切り離し (detach) / 接続 (attach)

※ RAID 1 (ミラー) のみ



■ 置き換え (replace)

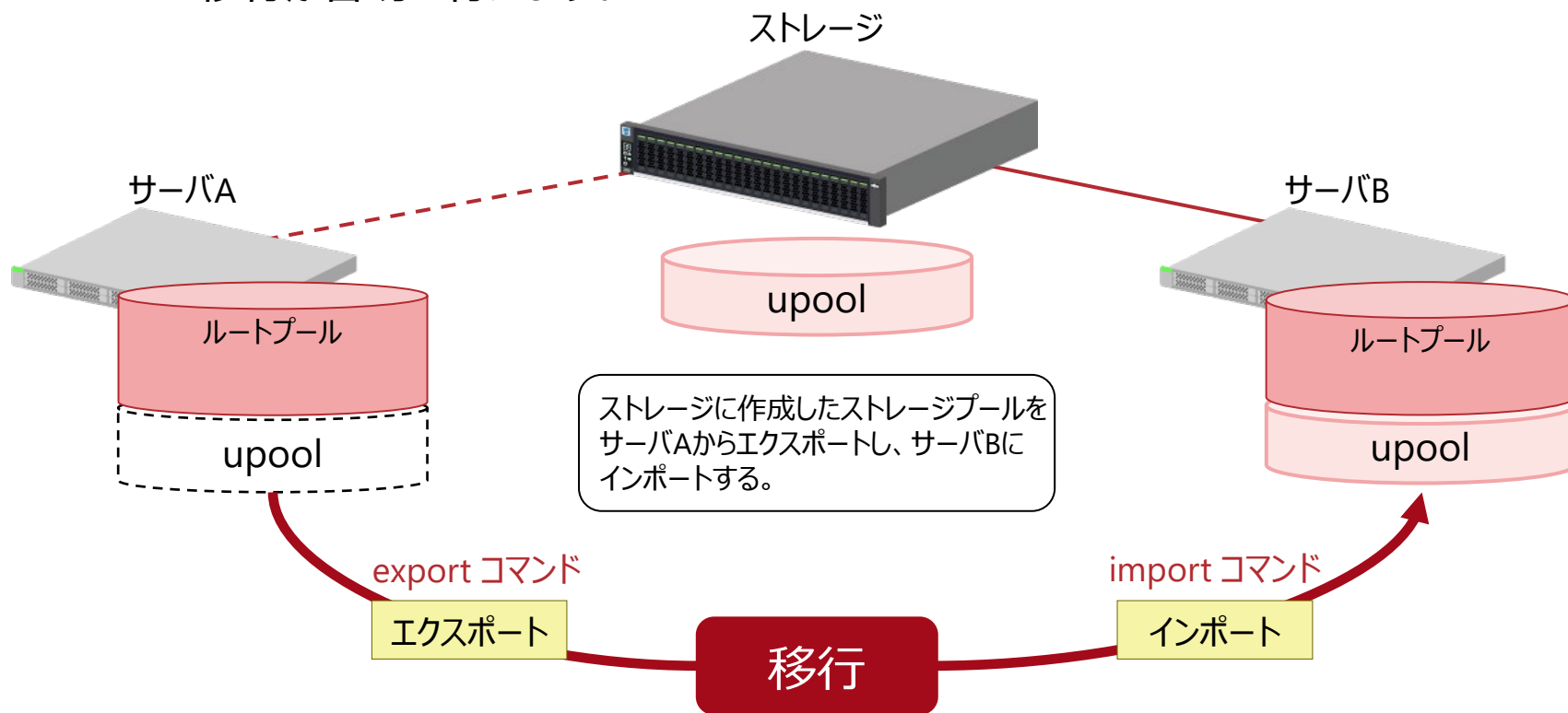
※ RAID 1 (ミラー) / RAID-Z / RAID-Z2 / RAID-Z3



- 新しいディスクを接続すると、すぐに同期が開始されます。
- ディスクを置き換えるときは、元のディスクと同等以上のサイズのディスクを使用してください。

■ ストレージプールの移行でできること

- ストレージプールをエクスポート（export）し、別のサーバにインポート（import）できます。ストレージプールを構成しているディスクを別サーバへインポートすることで、ストレージプールの移行が容易に行えます。



- ストレージプールをエクスポートすると、ストレージプールは一時的にシステムから認識されなくなります。インポートすることで、ストレージプールが再認識されます。
- ルートプールの移行はできません。

■ ZFS操作の監視でできること

- ファイルシステム、およびストレージプールの操作を監視できます。

※ Solaris 11.3 から利用できる機能です。



※ Ctrl+Cを入力するか、該当の処理が完了するまで出力し続けます。
左記の例では、5秒間隔でスクラブ処理の進捗が表示されます。

✓ 進捗状況を監視できる操作

ファイルシステムの操作 (zfsコマンド)	ストレージプールの操作 (zpoolコマンド)
• ZFSデータ送受信 (send/receive)	• ファイルシステム検証 (scrub)
• スナップショット破棄 (destroy)	• プール再同期 (resilver)

→ ファイルシステム検証 (スクラブ) の詳細については、「[スクラブ -明示的なデータ検査-](#)」を参照してください。

付録

『Oracle Solaris 11.4 での ZFS ファイルシステムの管理』 (Oracle社)

https://docs.oracle.com/cd/E75431_01/pdf/E75197.pdf

『Managing File Systems in Oracle Solaris 11.4 (English)』 (Oracle社)

https://docs.oracle.com/cd/E37838_01/pdf/E61016.pdf

SPARCサーバ／Oracle Solarisの技術情報を掲載



今すぐアクセス!!!

<https://www.fujitsu.com/jp/sparc-technical/>

版数	更新日時	更新内容
初版	2012年8月	新規作成
第2.0版	2013年4月	SPARC M10 および、Oracle Solaris 11.1 に対応
第3.0版	2015年6月	Oracle Solaris 11.2 に対応
第4.0版	2016年5月	レイアウトデザインと構成を更新 Oracle Solaris 11.3 に対応
第5.0版	2019年6月	SPARC M12 および、Oracle Solaris 11.4 に対応

■ 使用条件

■ 著作権・商標権・その他の知的財産権について

- コンテンツ（文書・画像・音声等）は、著作権・商標権・その他の知的財産権で保護されています。本コンテンツは、個人的に使用する範囲でプリントアウトまたはダウンロードできます。ただし、これ以外の利用（ご自分のページへの再利用や他のサーバへのアップロード等）については、当社または権利者の許諾が必要となります。

■ 保証の制限

- 本コンテンツについて、当社は、その正確性、商品性、ご利用目的への適合性等に関して保証するものではなく、そのご利用により生じた損害について、当社は法律上のいかなる責任も負いかねます。本コンテンツは、予告なく変更・廃止されることがあります。

■ 輸出または提供

- 本製品を輸出又は提供する場合は、外国為替及び外国貿易法及び米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとり下さい。

■ 商標

- UNIXは、米国およびその他の国におけるオープン・グループの登録商標です。
- SPARC Enterprise、SPARC64、SPARC64 ロゴおよびすべてのSPARC商標は、米国SPARC International, Inc.のライセンスを受けて使用している、同社の米国およびその他の国における商標または登録商標です。
- OracleとJavaは、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
- その他各種製品名は、各社の製品名称、商標または登録商標です。

