

ZFSを試してみよう（基本編）

2010年3月
第1版

富士通株式会社



本テキストについて

■ 目的

Solaris10 OSの標準機能として実装されている、新しいファイルシステムであるZFS[Zettabyte(ゼタバイト)File System]について学び、実践的なスキルの向上を目指します。

■ 前提

本テキストの内容は、Solaris10 OS 10/09で提供される機能を前提にしています。

ZFS機能を使用する際には、Solaris 10 最新情報をご確認ください。

Solaris 10 最新情報は、[マニュアル](#)(Sun社webサイトへリンク)をご参照ください。

ZFSの機能詳細は、[Solaris ZFS 管理ガイド](#)(Sun社webサイトへリンク)をご覧ください。

目次

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

1. ZFSの概要

Solaris10の主な機能

ZFSは、ストレージ仮想化機能です。

資源を無駄なく

- **世界初の128bitファイルシステム (ZFS)**
事実上無限大といえるサイズのファイルシステムを構築可能

ストレージの
仮想化

- **ソフトウェアパーティショニング (Solarisコンテナ、LDoms)**
1つのサーバ上に複数のSolaris環境を構築
システムのイベントや負荷の状況に応じたシステム資源割り当てが可能

OSの
仮想化

- **性能向上 (TCP/IPスタックチューニング)**
TCP/IPスタック改良によるパフォーマンス向上

業務停止を短く

- **SW/HW障害の自動的な診断・修復 (予測的セルフヒーリング)**
あらかじめ予測されるシステム障害を自動的に検知、可用性を向上させる自己修復機能

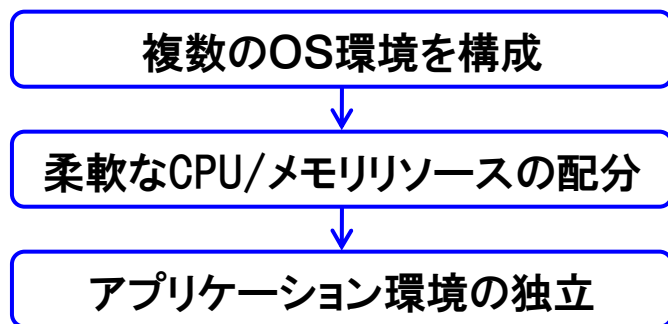
- **ボトルネックの早期究明 (Dtrace:ダイナミックトレース)**
システム障害や性能低下の原因解明を簡単におこなえるシステム診断ツール

- **堅牢なセキュリティ環境を構築 (Trusted Solaris)**
「Trusted Solaris」として提供されていた高度なセキュリティ管理機能を標準搭載

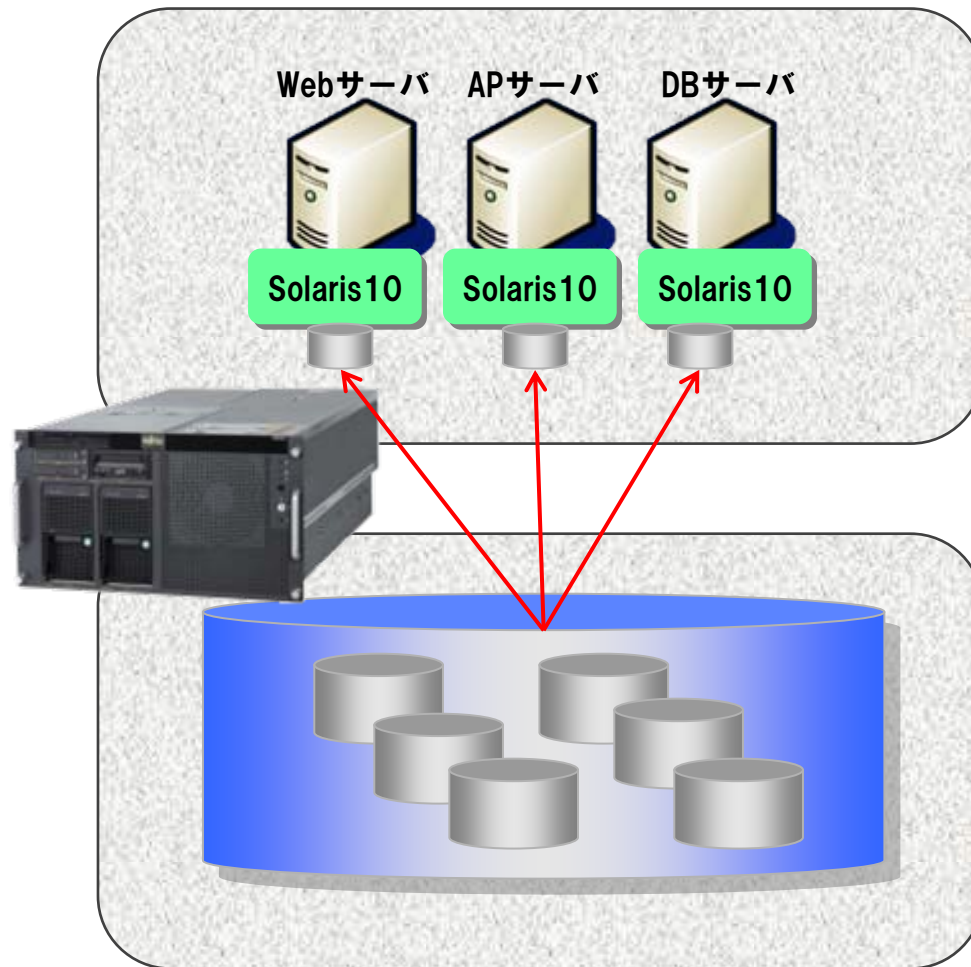
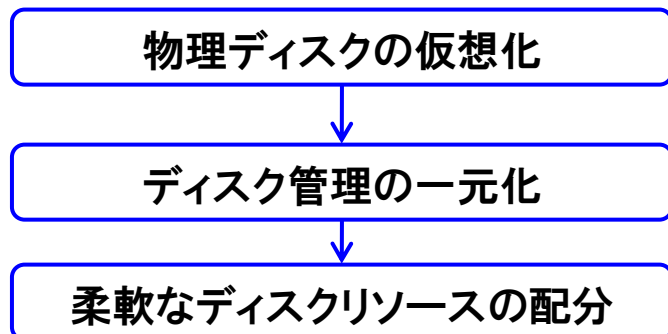
サーバ・ストレージの仮想化

ストレージの仮想化により、柔軟な仮想サーバの構成/運用が可能に！

■サーバ仮想化



■ストレージ仮想化



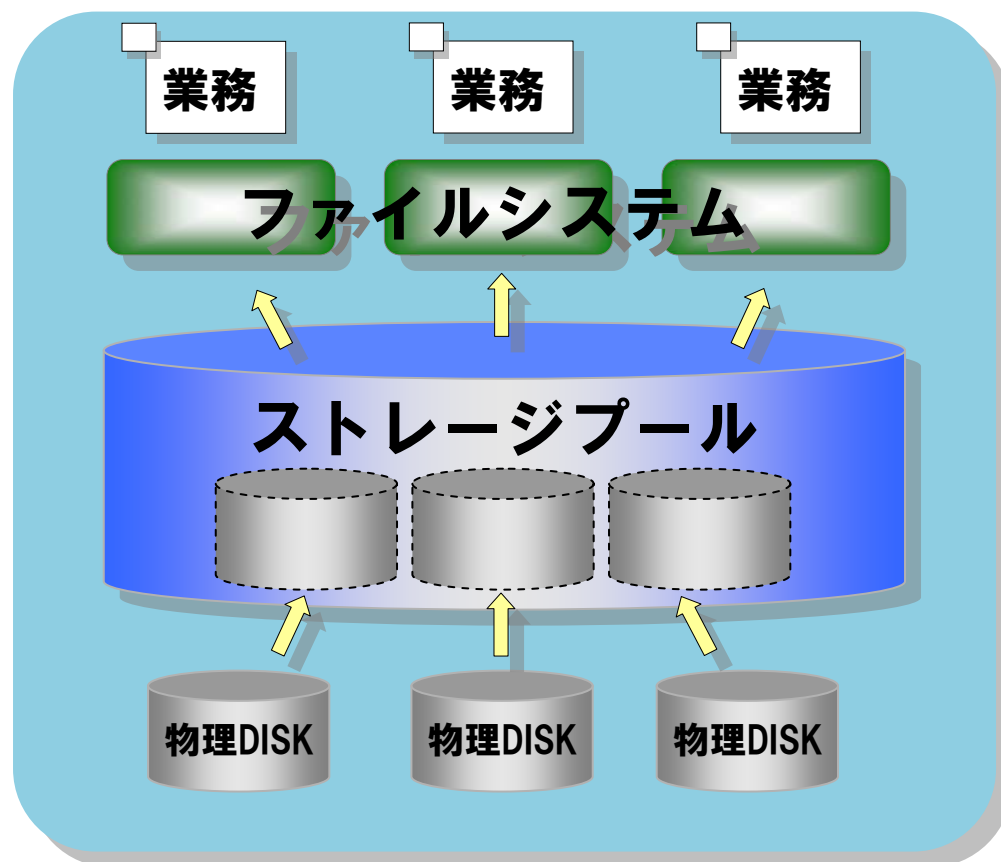
サーバの仮想化 ～SPARC Enterprise～

機能名 (適用モデル)	XPAR (SPARC Enterprise Mxxxx)	LDoms (SPARC Enterprise Txxxx)	Solaris コンテナ (SPARC Enterprise全て)
分割階層	ハードウェア層	ハイパーバイザー層	OS層
構成図			
特長	障害隔離性: 高 柔軟性: 高		
耐障害性	アプリケーションの独立性		
	OSの独立性		
構成自由度	ハードウェアの独立性		
	稼働中に1チップ/4 or 8DIMM / 2PCIeスロット単位で追加/削除/移動ができる	稼働中に1スロット単位で移動できる	ハードウェアに依存せず、稼働中に業務毎のCPU/メモリ比率を変更できる

ストレージの仮想化 ～ZFS～

■ ストレージプールにより物理的なデバイス構成から完全に独立

- ✓ 物理ディスク、ディスクパーティション、実ファイルなどをまとめて、1つの仮想的なストレージデバイスである「ストレージプール」を構成



業務からはストレージプールから切り出されたファイルシステムを利用する

物理ディスクを意識する必要はなく、ストレージプールには物理デバイスの動的追加が可能

2. ZFSの特長

ZFSの特長

◆優れたスケーラビリティ

- 世界初の128bitのファイルシステム
- 事実上無限大といえるファイルシステムの構築

◆簡素化された管理

- 管理コマンドの簡素化
- ストレージプールによるデバイスの隠蔽とストレージ管理の容易化

◆データの堅牢性

- RAID構成によるデータ冗長化
- Copy-On-Write (COW) トランザクションファイルシステムモデル
- チェックサムとデータの自己修復

優れたスケーラビリティ

■ ZFS… Zetta byte File System

↳ 10^{21} (1,000,000,000,000,000,000,000) byte (70bit)



128bit はさらに大きなサイズ!! (3.4×10^{38})

	UFS	ZFS
ファイルシステムサイズの上限	16TB	25京6000兆ZB (256,000,000,000,000,000,000,000TB)
ファイルシステム内に作成可能なファイル数	i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る。16TBの場合、 15×10^6 個

➤ **実際のオペレーションにおいては無限と考えてよい**

参考: キロ (10^3)、メガ (10^6)、ギガ (10^9)、テラ (10^{12})、ペタ (10^{15})、エクサ (10^{18})、ゼタ (10^{21})、ヨタ (10^{24})

簡素化された管理(1)

ZFSの管理に必要なコマンドはわずか2つ

zfs コマンド

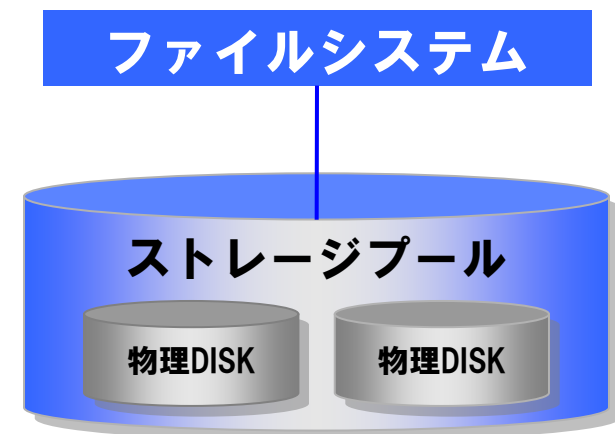
<主な操作>

- ・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

zpool コマンド

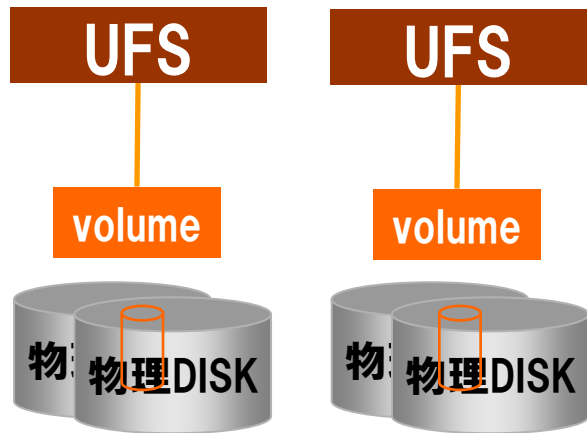
<主な操作>

- ・物理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(ディスクスクラブ)

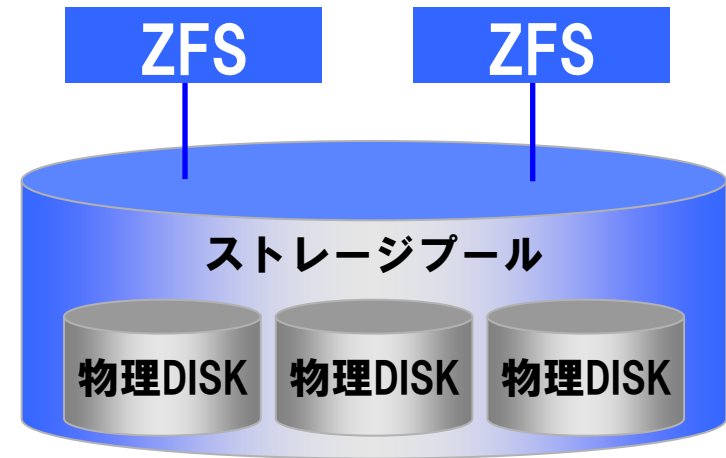


簡素化された管理(2)

UFSとZFSのファイルシステム構成の違い



- 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変更は、システムを停止しバックアップ/リストア。
- 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。



- 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。

＜参考＞UFSファイルシステムの作成

■ UFSファイルシステム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る

3. ファイルシステム作成(newfs、fsck)

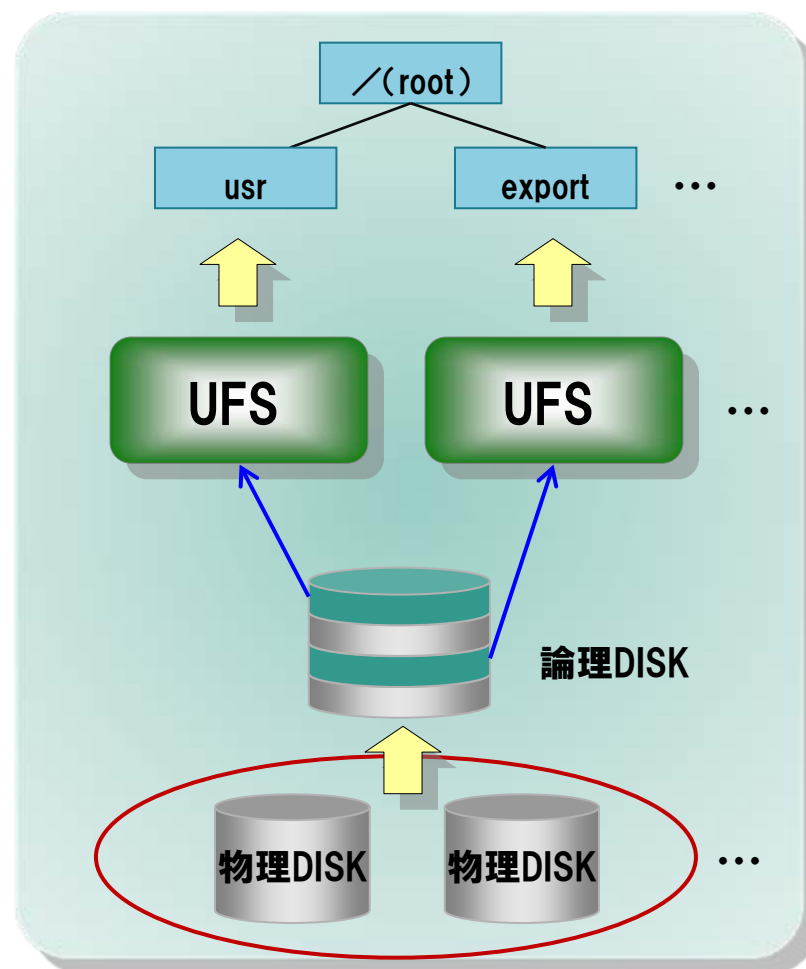
- ・パーティション単位にファイルシステムを作成し、整合性チェックを行う

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)を決める

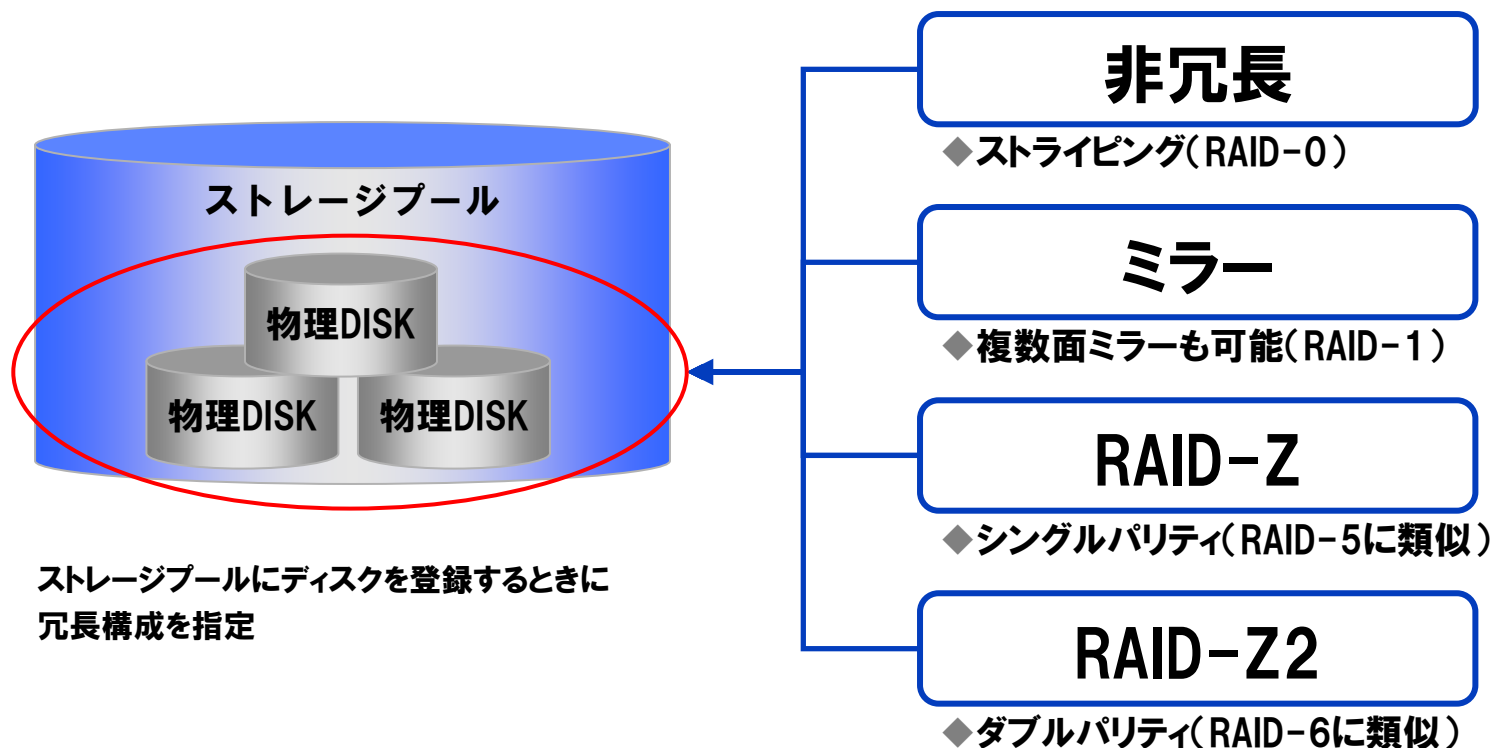
1. 物理ディスクの追加・認識

- ・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う



データの堅牢性(1)

ZFSでサポートするディスクの冗長構成



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(ミラー、RAID-Z、RAID-Z2)であれば複製データから自動的にデータを修復します(Self-Healing)

データの堅牢性(2)

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」の方式で管理
- ✓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

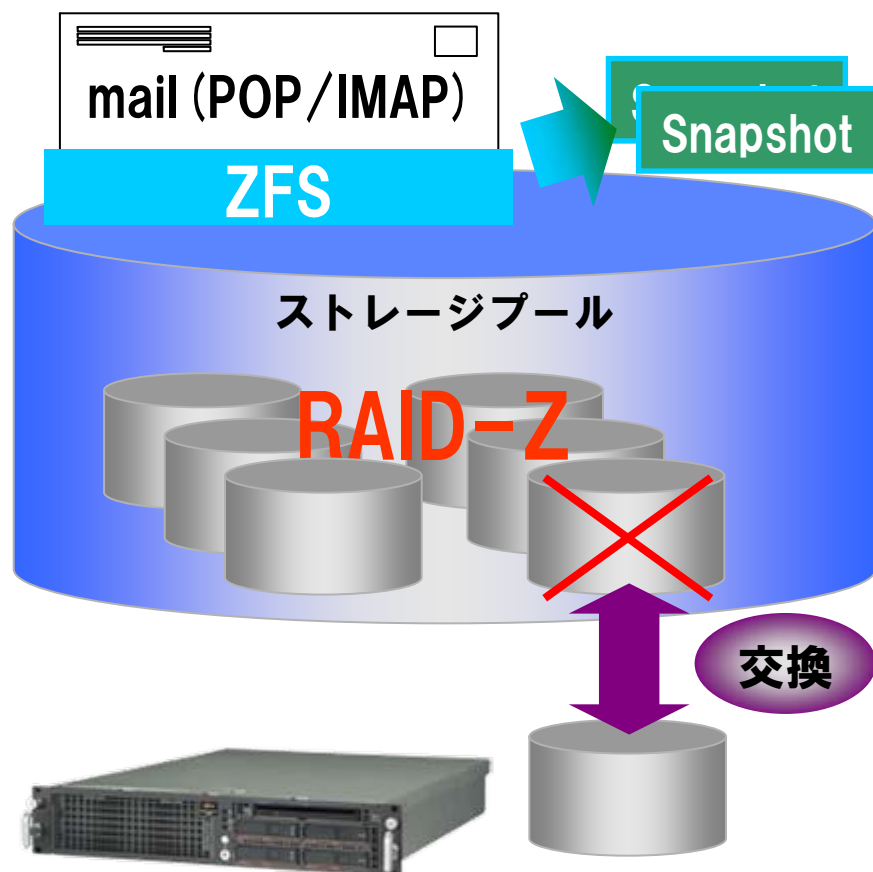
End-to-Endのチェックサム

- ✓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

適用事例：弊社部門メールサーバ

OS標準機能のみで高信頼なメールサーバを構築



<特長>

- RAID-Zでストレージプールを作成し、信頼性とディスク容量を確保
- スナップショットによる世代管理
→作成時間を感じないほど短時間

<運用時>

- ディスク障害時は縮退して業務を継続
- オンラインでディスク交換を実施

3. ZFSの信頼性/可用性

ZFSの信頼性・可用性

■ZFSの信頼性と可用性を支える仕組み

✓コピーオンライト

ファイルへの書き込み時に元データを上書きしない仕組み

✓トランザクションファイルシステム

データの更新時に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

✓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

✓RAID-Z RAID-Z2

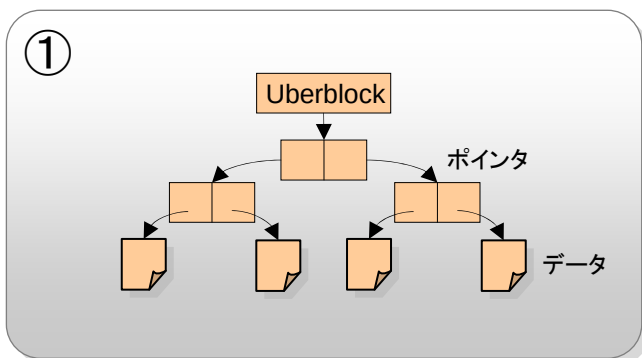
物理ディスクの損傷時にデータを守るZFS特有のRAID構成

コピーオンライト

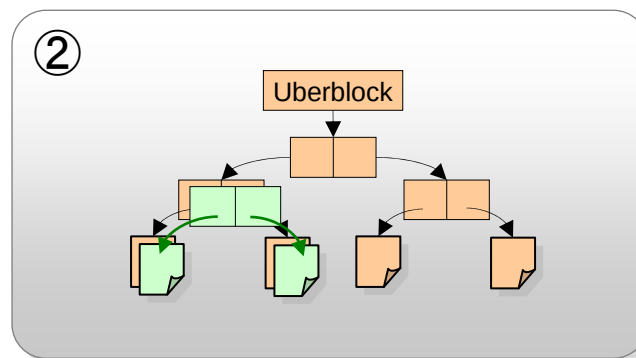
■コピーオンライトによるデータ更新

ZFSは、コピーオンライト(Copy On Write)というデータの更新方法をとっており、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になっています。

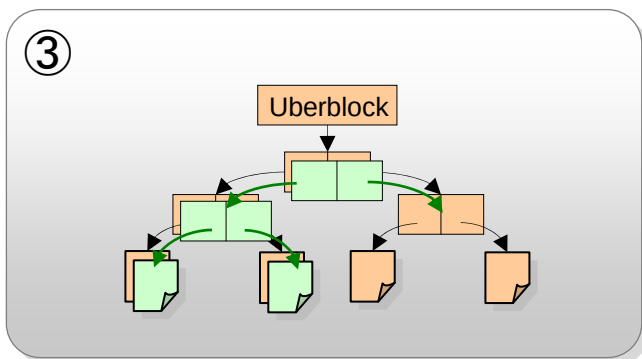
コピーオンライトでは、いったん書き込んだデータを変更するとき、まず元のデータのコピーを作成しそのコピーの方を変更します。その後、システムにとって区切りの良い時点でデータブロックのリンクを更新します。



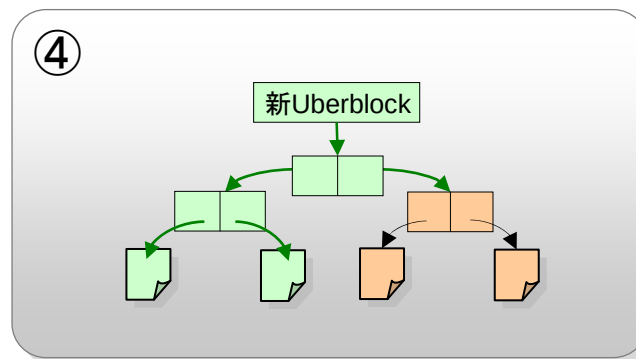
初期状態のブロック構造



データを含むブロックを複製し、更新する



ポインタを含むブロックを複製し、新しいデータにリンクする



Uberblockを更新する

トランザクションファイルシステム

■ ZFSはトランザクションファイルシステムを採用

ZFSファイルシステムは、一連の書き込み動作を一つの更新単位(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利点

- ✓データの不整合が発生しない
データ更新が「全て成功」か「全て失敗」のどちらかであることが保障される
- ✓ファイルシステムの整合性チェックが不要(UFSファイルシステムでは必須)
不意にシステムがクラッシュしてもファイルシステムを保護します

トランザクションファイルシステムの留意点

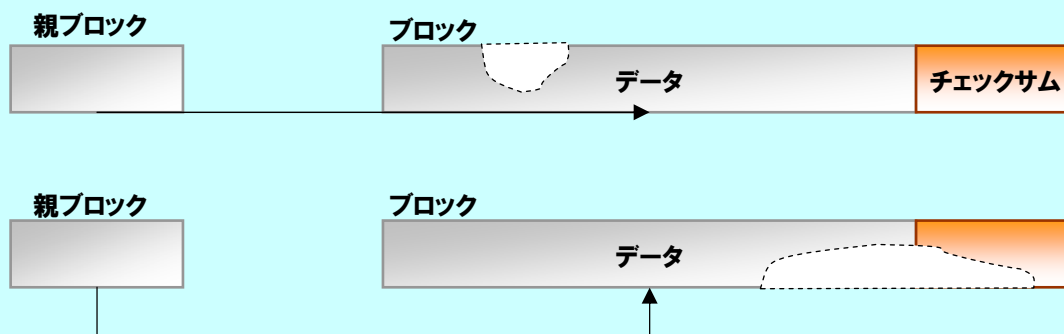
- ✓ディスクへの書き込みが非同期である
一連の処理が終了してからディスクへ書き込み(I/O発生)が行われるため、実際のディスク容量を確認したときに差異が発生することがあります(dfコマンド、duコマンドなど)

チェックサムとデータの自己修復

■End-to-Endのチェックサムによりファイルシステムの保護機能を強化

データとチェックサムを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。

従来のファイルシステム



データの損傷範囲がチェックサムにまで及ぶとデータ修復ができない。

ZFSファイルシステム



データとチェックサムが物理的に離れて別々に読み込まれる。チェックサム自身も上位ブロックによって修復可能。

- ディスク故障による読み出し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(ミラー、RAID-Z、RAID-Z2)であれば複製データから自動的にデータを修復します(Self-Healing)

スクラブ

■ZFSファイルシステムのデータの完全性を明示的にチェックする機能

基本的にZFSではチェックサムによってデータの損傷は自動修復(Self-Healing)されますが、明示的にファイルシステムを検査する場合には「スクラブ」を実行します。

スクラブは全てのデータを順にたどってみて、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

- ✓スクラブによってハードウェア障害やディスク障害に起因する非表示のエラーも検出します
- ✓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させることができます。
- ✓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(業務へ影響を与えません)。

●スクラブ実行直後

```
# zpool scrub rpool
# zpool status rpool
pool: rpool
state: ONI INF
scrub: scrub in progress for 0h0m, 0.02% done, 11h58m to go
config:
```

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c0d0s0	ONLINE	0	0	0
c0d1s0	ONLINE	0	0	0

errors: No known data errors

●スクラブ完了後

```
# zpool status rpool
pool: rpool
state: ONI INF
scrub: scrub completed after 0h13m with 0 errors on Thu Feb  4 15:52:42
2010
config:
```

NAME	STATE	READ	WRITE	CKSUM
rpool	ONLINE	0	0	0
mirror	ONLINE	0	0	0
c0d0s0	ONLINE	0	0	0
c0d1s0	ONLINE	0	0	0

errors: No known data errors

4. ストレージプールの構成

ストレージプールの種類

■2種類のストレージプール

●ストレージプール

- ✓ 仮想的なストレージデバイス
- ✓ 物理ディスクやパーティション、データファイルで構成

●ルートプール

- ✓ OS起動用のストレージプール
- ✓ 従来のシステムボリュームを構成(/、swap、/usr、/var、/opt、/sbinなど)

プール名	用途	構成タイミング	構成可能なRAIDレベル	備考
ストレージプール	ユーザ領域の管理	適時	RAID0・1・Z・Z2	動的な拡張が可能
ルートプール	ブート領域の管理	OSインストール時	RAID0・1	動的な拡張が不可

※RAID1+0も構成可能。

※ルートプールはSolaris10 10/08以降でサポートされます。Solaris10 10/08以前のZFSはブート領域には対応していません。

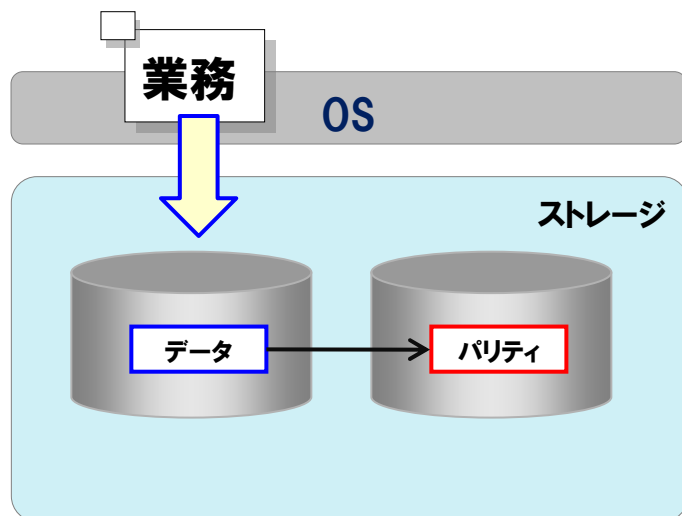
ストレージプールを構成するデバイスは、物理ディスク全体を使用することを推奨します。

ZFS特有のRAID構成 ～RAID-Z,RAID-Z2～

■「RAID5書き込みホール」問題をソフトウェアで解決する世界初のソリューション

- RAID-Z : RAID5(シングルパリティ)に相当。ディスク2本以上で構成。
- RAID-Z2 : RAID6(ダブルパリティ)に相当。ディスク3本以上で構成。

パリティを利用する従来の仕組み(RAID5)



データディスクが故障してもパリティからデータを復元することが可能

停電など、不慮の事故でパリティの書き込みが途中で不完全に終わるとデータの不整合が起こる(書き込みホール問題)

1) データの書き込み 2) パリティを計算して書き込み

ZFSファイルシステムは、データ書き込みとパリティ書き込みが同時に完了するため、データの不整合は発生しない仕組みになっています(トランザクションファイルシステム)。

ルートプールの作成

■システムボリュームにZFSを選択することが可能（Solaris10 10/08以降）

- ✓OSインストール時にファイルシステムタイプに「ZFS」を選択するだけです
- ✓複数のディスクを選択すると、自動的にミラー構成のファイルシステムを作成します（推奨）

<OSインストール時の選択画面>

q ファイルシステムタイプの選択 qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
Solaris インストールに使用するファイルシステムを選択します

[] UFS
[X] ZFS

qq
Esc-2_継続 F3_前に戻る F5_終了 F6_ヘルプ

q ディスクの選択 qqq

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

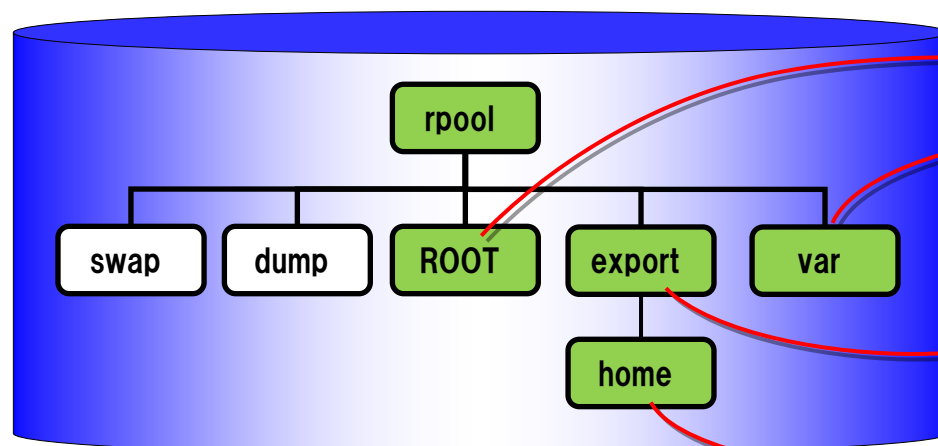
※複数ディスクを選択することで3面以上のミラー構成も可能です。

- OSインストール後にミラー構成にすることも可能ですが、bootblock情報のコピーや代替ルートディスクの追加設定が必要となります。
- ミラー構成にするディスクは、複数のコントローラにまたがる構成を推奨します。

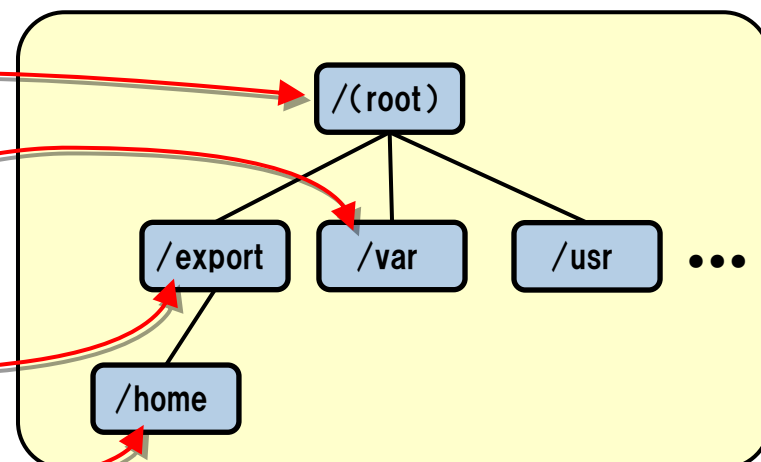
ルートプールの構成

■ルートプールはシステムボリューム専用のストレージプール

＜ルートプール (rpool) の構成＞

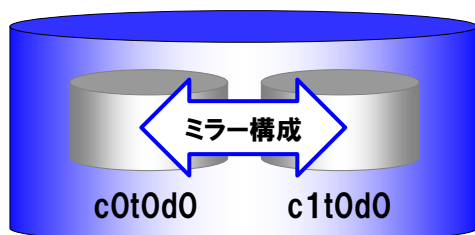


＜ファイルシステムの構成＞



✓ストレージプールでは、データセットを階層構造で管理しています

✓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マウントされます



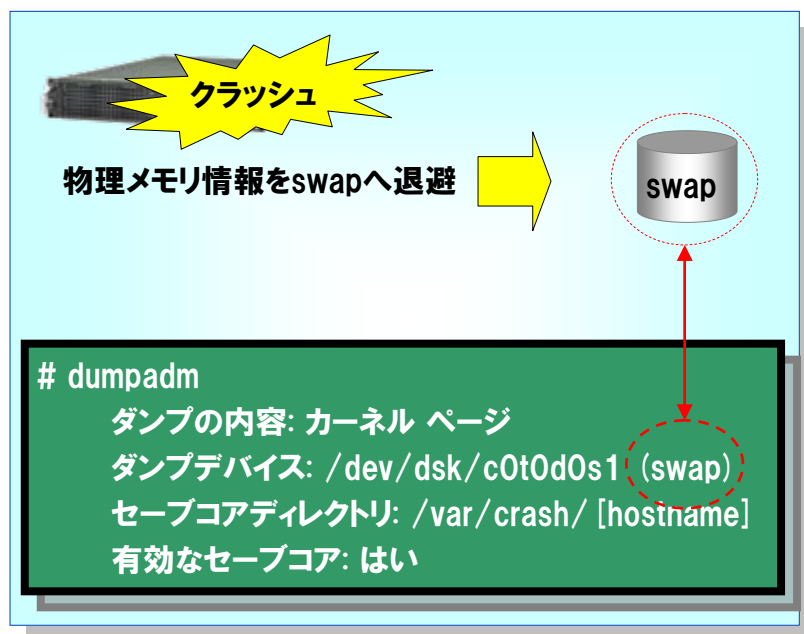
データセット: ファイルシステム、ボリューム、スナップショット、クローンなどストレージプール上での総称

ZFS上のスワップとダンプの扱い

■ZFSではスワップ領域とダンプ領域はZFSボリュームで作成された個別の領域で作成される

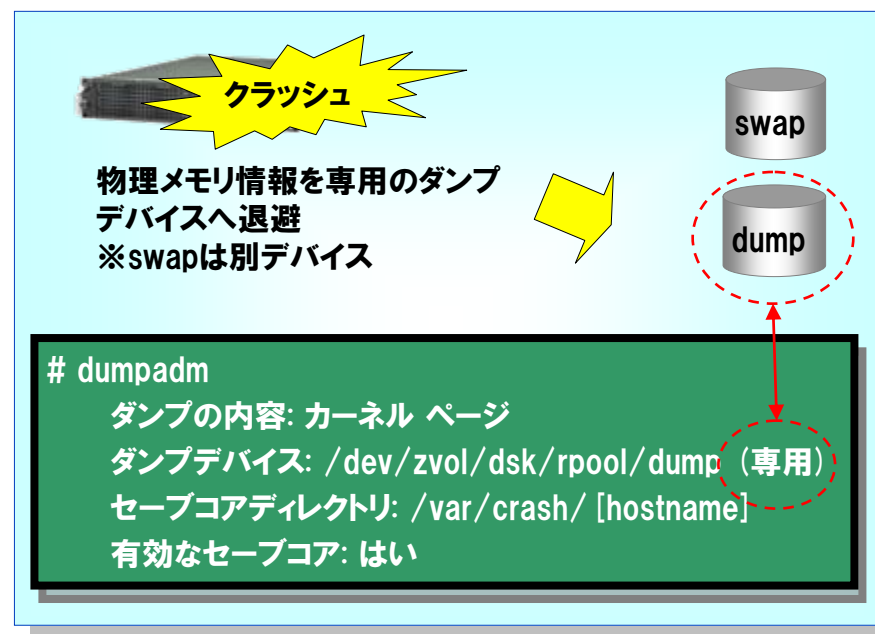
UFSルートファイルシステム

- 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(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可能



ZFSボリューム : 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こと。
OS上のデバイスは/dev/zvol 配下で管理されます。

スワップデバイスとダンプデバイス

■スワップデバイスとダンプデバイスの領域はインストール後に変更可能

✓スワップデバイスのデフォルトサイズ

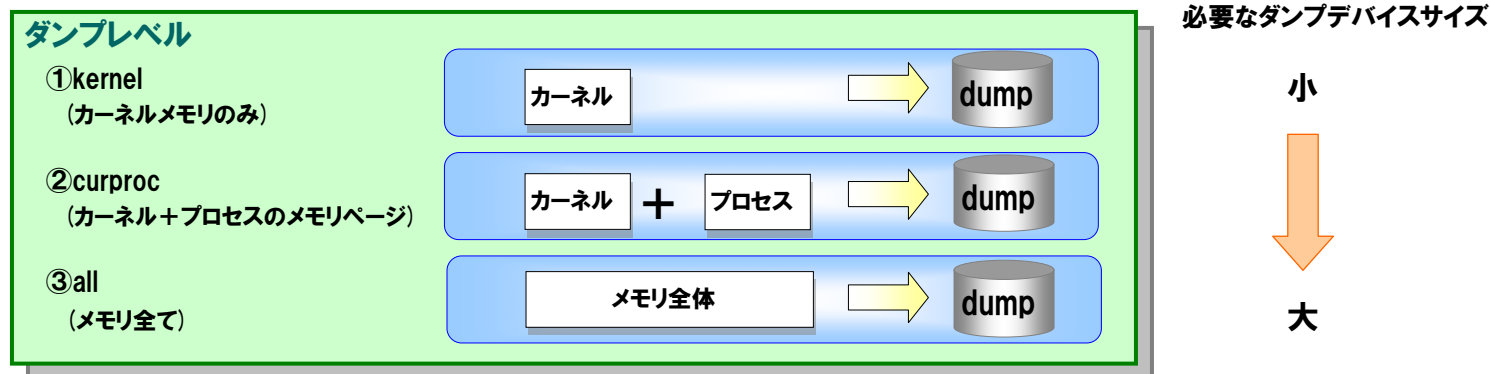
Solarisの初期インストール時に、物理メモリの半分で計算されます。(但し、512MB～2GB の範囲で設定されます。)

✓ダンプデバイスのデフォルトサイズ

物理メモリのサイズとダンプレベルの情報に基づいて、カーネルによって計算されます。

ZFSにおけるダンプデバイスの見積もり

取得するダンプレベルに合ったサイズのダンプデバイスの確保が必要です。



dumpadmコマンドでダンプレベルを変更時に、ダンプデバイスのサイズが小さい場合は、異常メッセージが出力され、ダンプレベルを変更できません。

```
# dumpadm -c all
dumpadm: ダンプデバイス /dev/zvol/dsk/rpool/dump はシステムダンプを格納するには
小さすぎますダンプサイズは 5261225984 バイト、デバイスサイズは 3221225472 バイトです
```

インストール時にはダンプレベルを変更できないため、ダンプレベル変更後に必要なデバイスサイズを確認し、ダンプデバイスのサイズを変更します。

スワップデバイスとダンプデバイスの領域変更

■インストール後の領域変更が可能

ZFSのスワップデバイスとダンプデバイスのサイズは、「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。

swapデバイス

現在のswapデバイスを削除し、volsizeプロパティ値を変更後にスワップデバイスを再構築する

```
# swap -d /dev/zvol/dsk/rpool/swap
# zfs set volsize=3G rpool/swap
# /sbin/swapadd
# swap -l
swap ファイル      デバイス  swaplo ブロック 空き容量
/dev/zvol/dsk/rpool/swap 256,2    16 6291440 6291440
```

従来のようにswapデバイスの追加も可能

```
# zfs create -V 2G rpool/swap1
# swap -a /dev/zvol/dsk/rpool/swap1
# swap -l
swap ファイル  デバイス  swaplo ブロック 空き容量
/dev/zvol/dsk/rpool/swap 256,2    16 4194288 4194288
/dev/zvol/dsk/rpool/swap1 256,3    16 4194288 4194288
```

dumpデバイス

volsizeプロパティの値を変更するのみでOK

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

ダンプレベル変更時にdumpデバイスの容量がチェックされるため、不足していた場合はdumpデバイスの値を変更します。

/var領域の設定

■インストール時に/varをルートファイルシステムと別のデータセットにすることが可能

/varを別データセットにする利点

- ✓/varデータセット単位でスナップショット作成が可能になる
- ✓/varデータセットに対して領域の使用上限を設定することが可能になる(プロパティの個別設定)

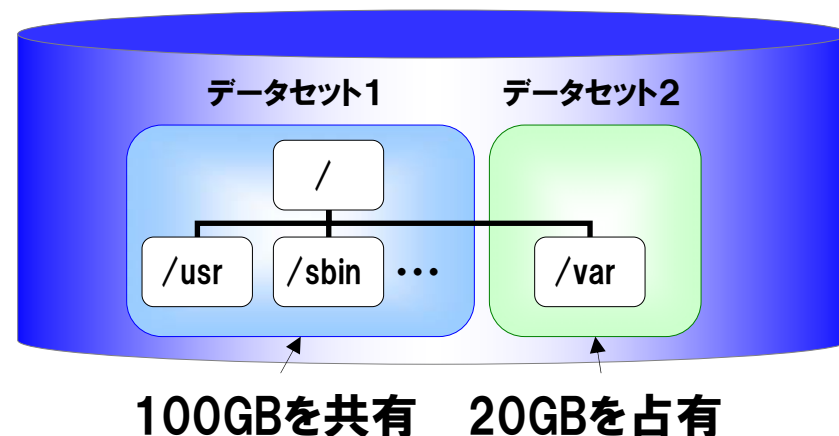
q ZFS 設定の構成 q

選択したディスクから作成するプールの名前を指定します。また、ファイルシステムのルートディレクトリとして使用するプール内に作成するデータセットの名前を指します。

ZFS プール名: rpool
 ZFS ルートデータセット名: s10s_u8wos_08a
 ZFS プールサイズ (MB 単位): 12204
 スワップ領域のサイズ (MB): 2048
 ダンプ領域のサイズ (MB): 1024
 (プールサイズは 6726 MB から 12204 MB の範囲でなければなりません)

- ☐ / と /var を統合したままにする
☒ /var を別のデータセットに配置する

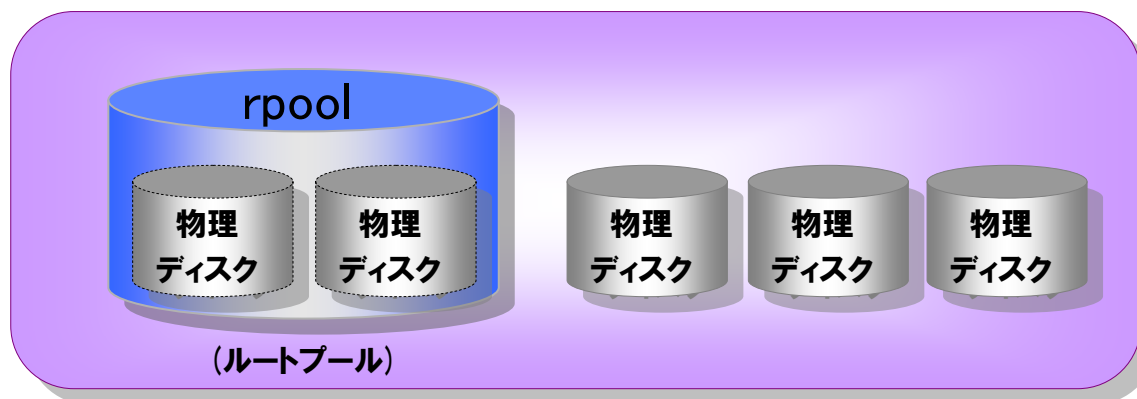
例えば...



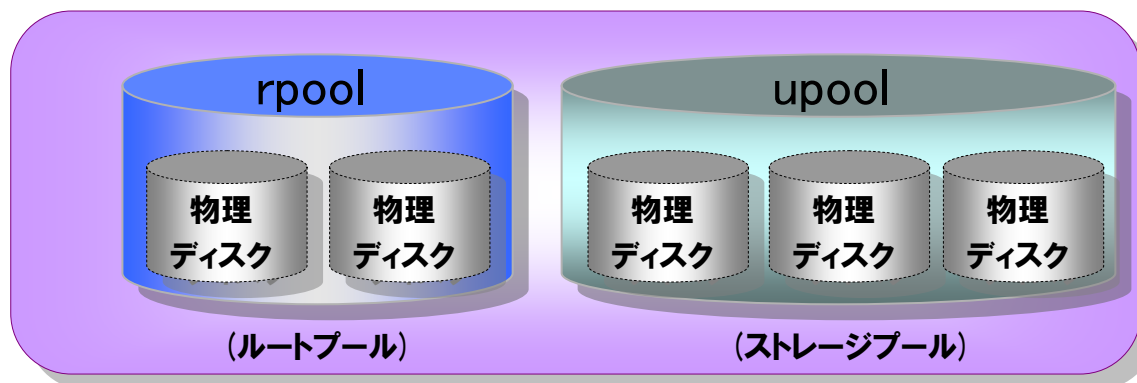
インストール後に別データセット構成にすることはできません。別データセット構成にしても特にデメリットはありませんのでインストール時に設定しておくことを勧めます。

ストレージプールの作成

■ルートプールとは別の物理ディスクを使用して、ユーザ用ストレージプールを作成



ストレージプール(upool)を作成

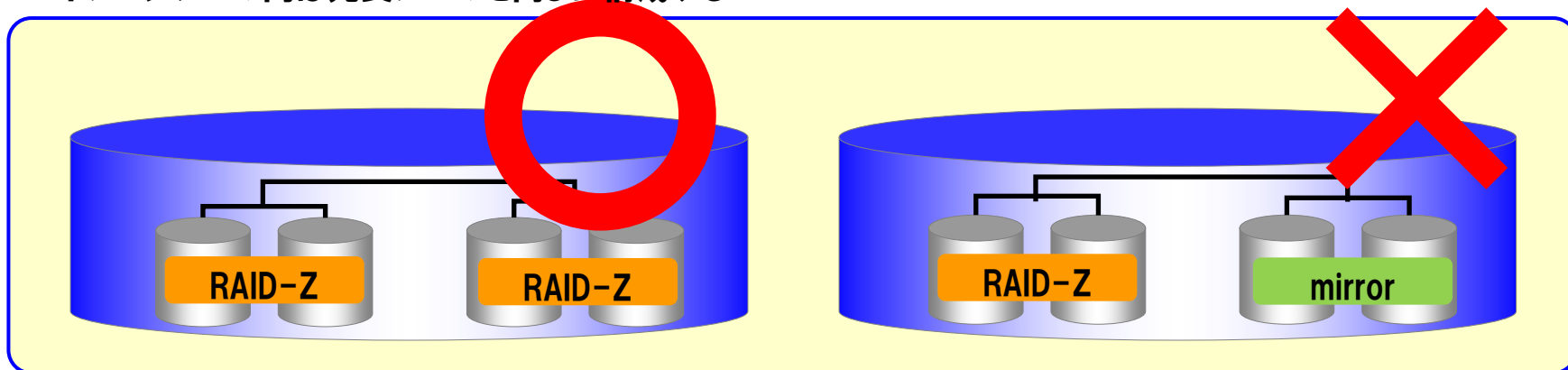


ルートプールとは別構成にして、アプリケーションやミドルウェアのデータ用として使用

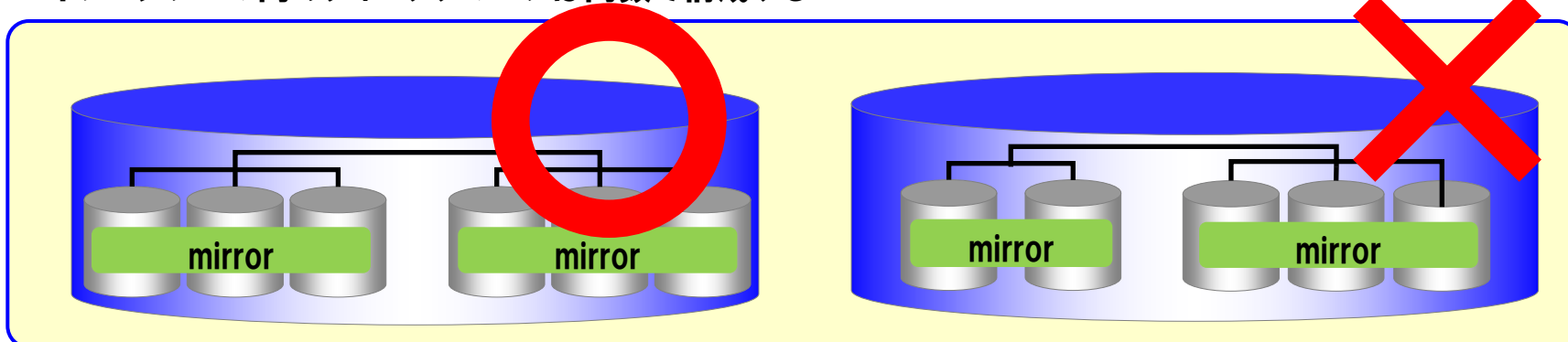
推奨されるストレージプールの構成

- 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だが、信頼性/パフォーマンス面で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じに構成する



ストレージプール内のディスクグループは同数で構成する



<参考>UFSからZFSへの移行

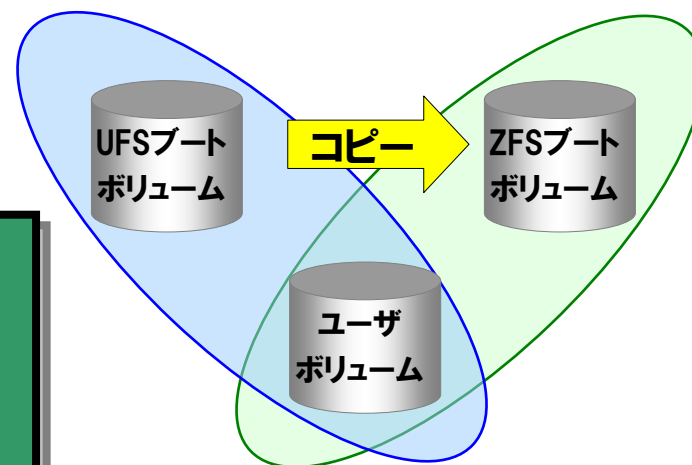
■ファイルシステムをUFSからZFSに移行することが可能

<移行手順>

- ✓ZFSルートファイルシステムとなるスライスの準備
- ✓Solaris Live Upgrade機能を使用してブート環境を複製
- ✓ブート環境を変更してOS再起動

```
# lustatus
```

Boot Environment Name	Is Complete	Active Now	Active On Reboot	Can Delete	Copy Status
bootenv-UFS	yes	no	no	yes	-
bootenv-ZFS	yes	yes	yes	no	-



■Solaris Live Upgrade とは・・・

- ✓システム稼働中にブート環境を複製し、その環境にパッチ適用やOSアップグレードを実施することができる
- ✓複製したブート環境はOS再起動で切り替えることが可能
- ✓複製した環境はシステム情報(設定ファイルやユーザ情報など)を元の環境から引き継ぐので再設定は不要
- ✓元の環境に戻したい場合も、ブート環境を変更してOS再起動することで可能

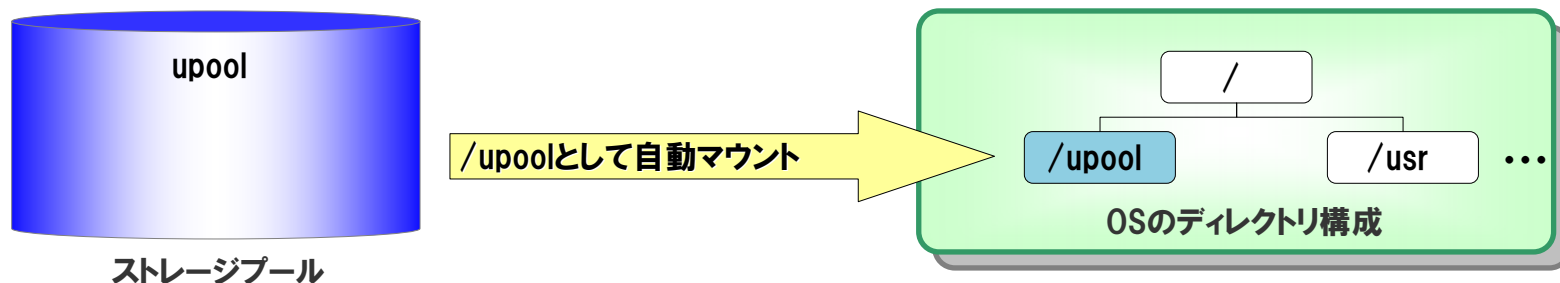
➡ Solaris10 10/08からZFSブート環境への切り替えがサポートされています

5. ファイルシステムの管理

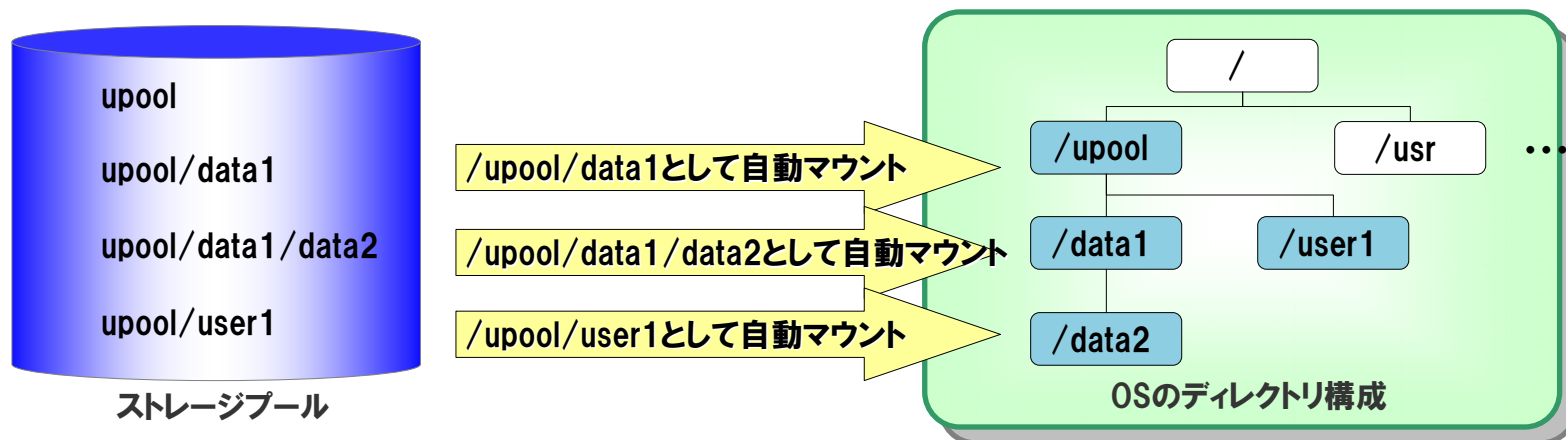
ファイルシステムの作成

■ZFSファイルシステムは階層構造で管理されます

✓最上位のZFSファイルシステムは、ストレージプールと同時に自動作成されます



✓上位のZFSファイルシステムを親として下位のZFSファイルシステムを作成します



各ファイルシステムは親のファイルシステムと同じディスク領域を共有します。

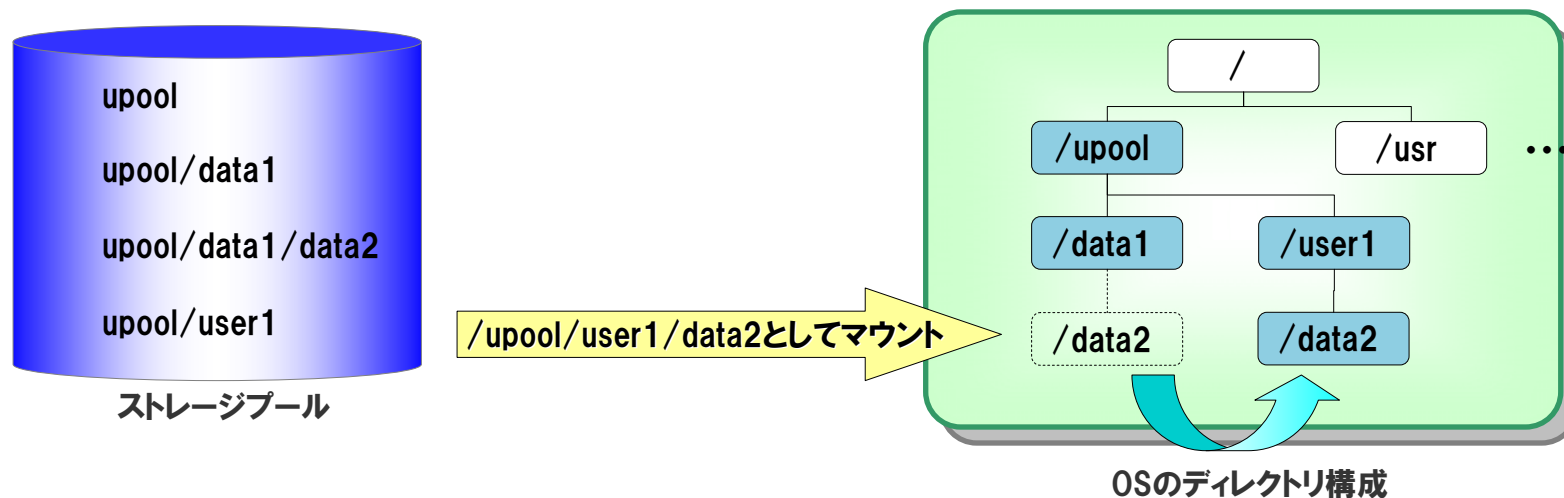
ZFSファイルシステムのマウント

■ZFSファイルシステム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を作成し、自動でマウントします。ファイルシステムを削除すると自動でアンマウントします。
- ✓マウントポイントの変更は、mountpointプロパティでマウントポイントを指定します。
- ✓従来(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



ファイルシステムのプロパティ(1)

■ZFSファイルシステムのプロパティ

ファイルシステムの特性のほとんどが、簡単なプロパティを使って制御します。
ファイルシステムのマウントポイント、割り当て制限などがプロパティによって制御されます。

■プロパティの種類

ファイルシステムのプロパティには大きく2種類あります。

ネイティブプロパティ

ZFSファイルシステムの動作を制御します。

✓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 (reservation) と予約 (quota) を除いて継承されます

ユーザプロパティ

ZFSの動作には影響しない任意に設定可能なプロパティ

ファイルシステムのプロパティ(2)

■プロパティの継承とは

親ファイルシステムのプロパティ値が子孫のファイルシステムにもプロパティ値が引継がれることです。継承できるプロパティについては、割り当て制限と予約を除いて、すべての設定可能なプロパティは、親から値を継承します。

■継承可能なプロパティの**SOURCE**は、プロパティがどのように取得されたかを表示します。

```
# 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
```

種類	説明
local	・プロパティが明示的に設定されたことを示します。
Inherited from <i>data-name</i>	・プロパティがどの親から継承されたかを示します。
default	・上記以外の状態を示します。

mountpointプロパティ

■mountpointプロパティの設定方法

1)マウントポイントを指定する

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

2)レガシーマウントを利用する

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

✓ZFSのレガシーマウント(/etc/vfstabを利用)

UFS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。

例: rz2pool/data2を/zfs/fs2へマウントする場合

① /etc/vfstabの編集

```
rz2pool/data2 - /zfs/fs2 zfs - yes -
```

② マウントポイントの作成

③ mountコマンドの実行

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

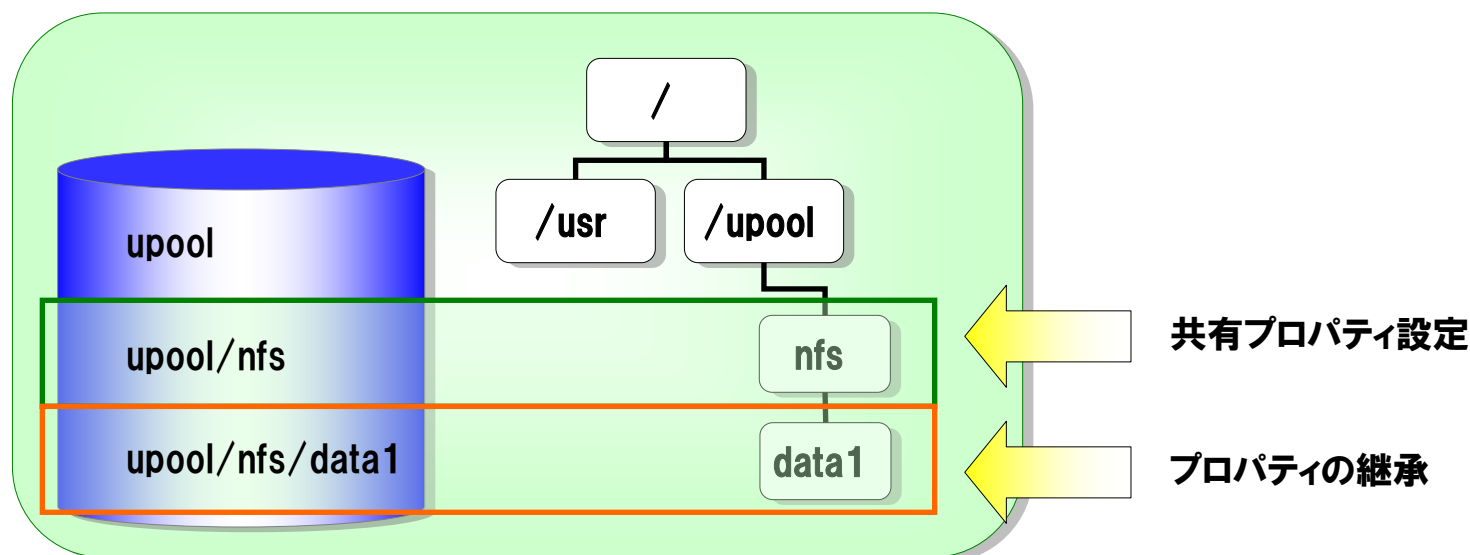
share nfs プロパティ

■ZFSのNFS共有設定

ZFS ではshare nfs プロパティを設定すると自動的にファイルシステムを共有できます。

ZFSの自動共有の特長

- ✓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v4(NFSv4)を使用します。

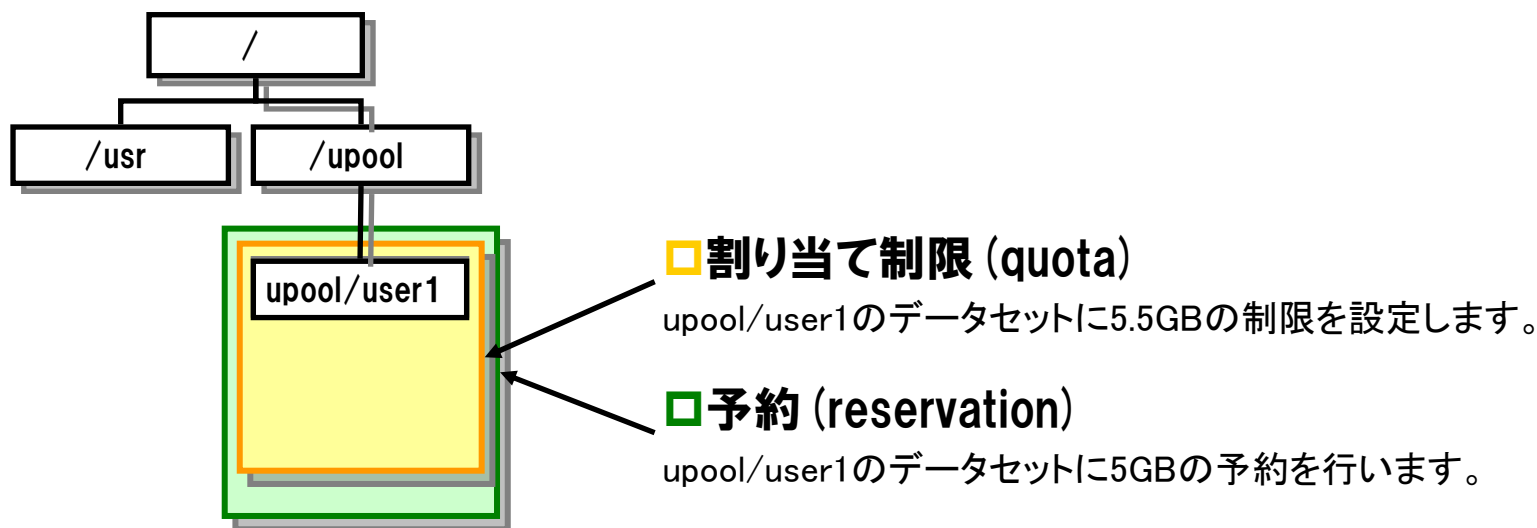


quota・reservationプロパティ

■ファイルシステムの割り当て制限と予約

ファイルシステム毎に「割り当て制限」を行い使用できる領域を制限することができます。
 ファイルシステムが使用できる領域を使用済みとして「予約」(確保)できます。
 これら2つのプロパティは設定したファイルシステムとその子孫に適用されます。

例:user1に対して5GBの領域を予約し、5.5GB以上の領域使用を制限する。



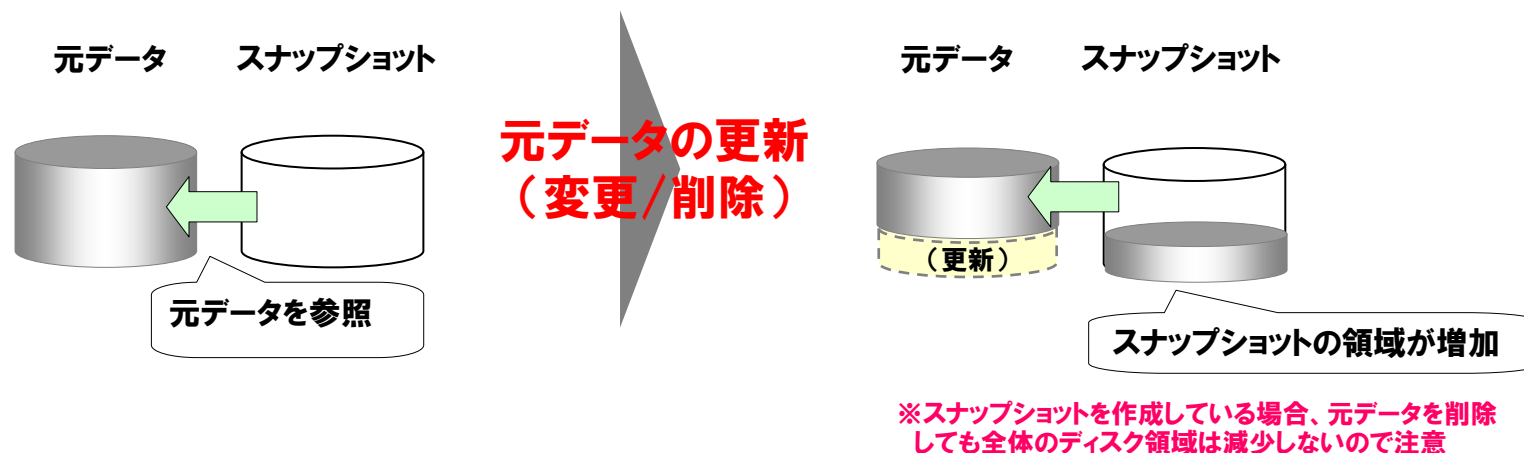
6. ZFSの機能

ZFSスナップショット

■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、スナップショット取得直後はプール内の領域を使用しません。

スナップショットの取得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元データ分の容量が増加します。



スナップショットの特長

- ✓作成可能なスナップショットの世代数は理論上 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。
- ✓上位のデータセットのスナップショットと下位のデータセットのスナップショットを同時に取得することもできます。

スナップショットのメリット

■スナップショットと連携する機能

✓スナップショット取得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。
スナップショットより後に加えられたすべての変更を破棄します。

✓クローンの作成

スナップショット取得時の状態と同一の環境を作り、テストなどに使用することができます。
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、プールの領域を消費しません。

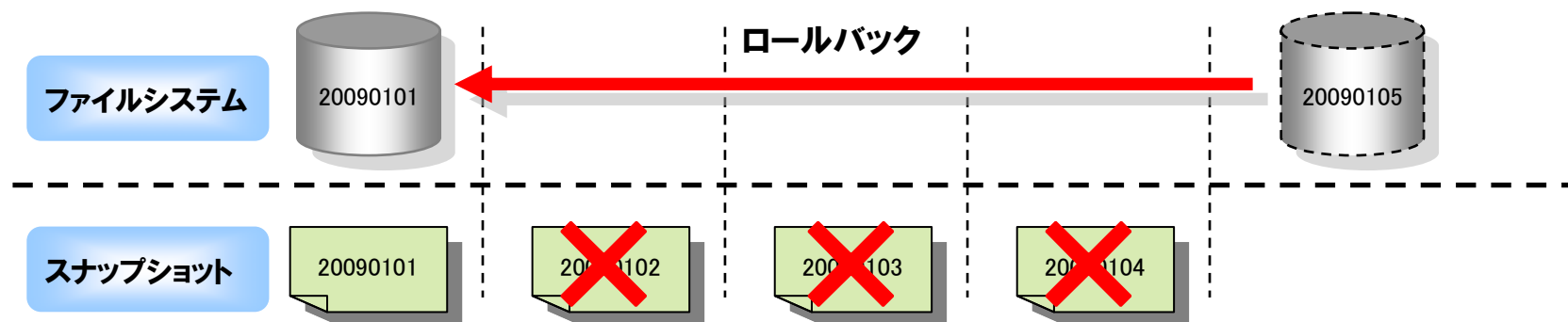
✓ストリームの送信/受信 (バックアップ/リストア)

スナップショットからストリームを作成してファイルへ送信することができます。(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することができます。(リストア)

ロールバック

■ロールバックの概要

特定のスナップショットより後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。データ変更する前にスナップショットを取得しておくことで、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。



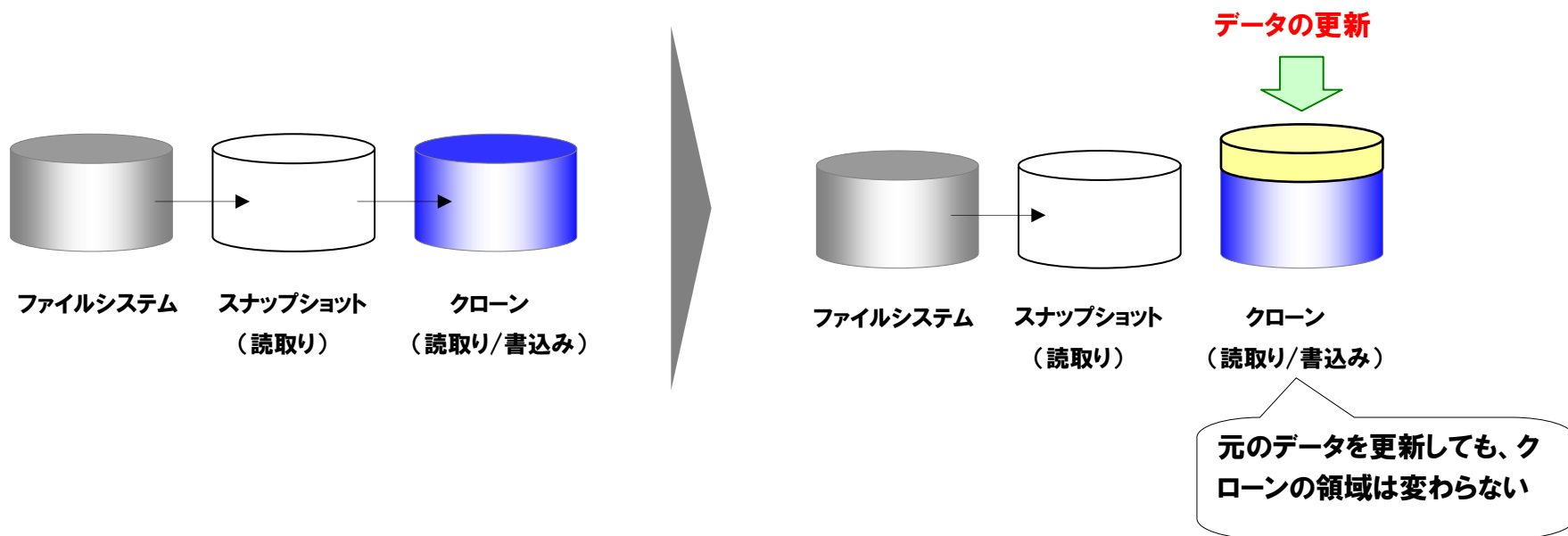
- ✓最新のスナップショットのみロールバックすることができます。
- ✓特定のスナップショットをロールバックする場合は、ロールバック対象のスナップショットを最新の状態になるように、中間のスナップショットを削除します。

中間にあるスナップショットのクローンが存在する場合も、クローンも破棄する必要があります。

クローン

■クローンの概要

書き込み可能なファイルシステムまたはボリュームのことで、取得直後はスナップショットと同じです。スナップショットの場合と同様に瞬時に作成され、クローン作成直後はプール内の領域を使用しません。クローンの領域は元のスナップショットと共有されます。

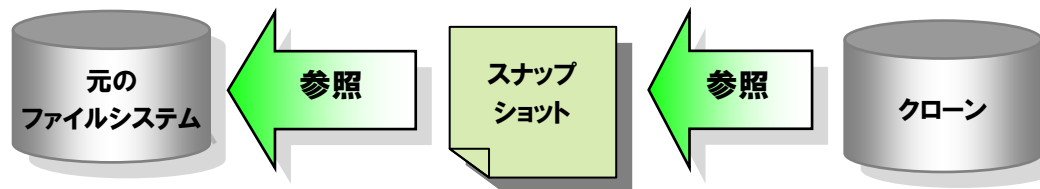


クローンはスナップショットからのみ作成します。また、クローンのスナップショットを作成することもできます。ただし、作成元のデータセットのプロパティは継承されません。

クローンとの依存関係

■依存関係について

クローンを作成すると元のスナップショットとの間に依存関係が作成されるため、クローンが存在する間は、スナップショットを削除することは出来ません。スナップショットを削除する場合、先にクローンの削除が必要です。



◆依存関係の確認

依存関係は、originプロパティで確認することが出来ます。

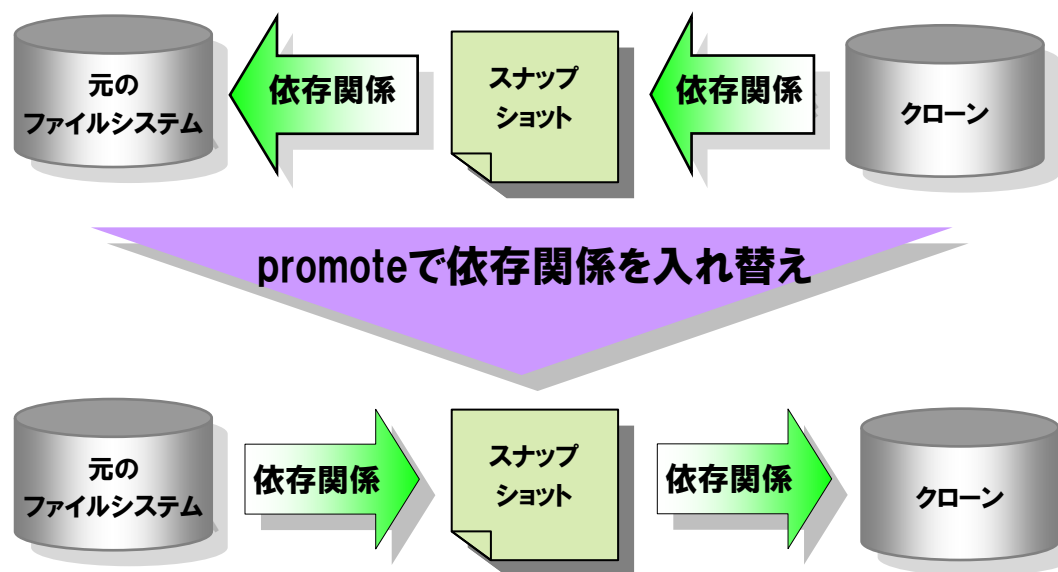
```
# zfs get origin upool/file
NAME      PROPERTY  VALUE                SOURCE
upool/file origin    upool/data@now       -
```

originプロパティによってファイルシステム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。

クローンとの置き換え

■クローンとの置き換え概要

作成したクローンは元のファイルシステムと依存関係を置き換えることで、クローン環境で加えた変更を反映させることができます。スナップショットも含めてクローンの依存関係を置き換えます。



ストリームの送信

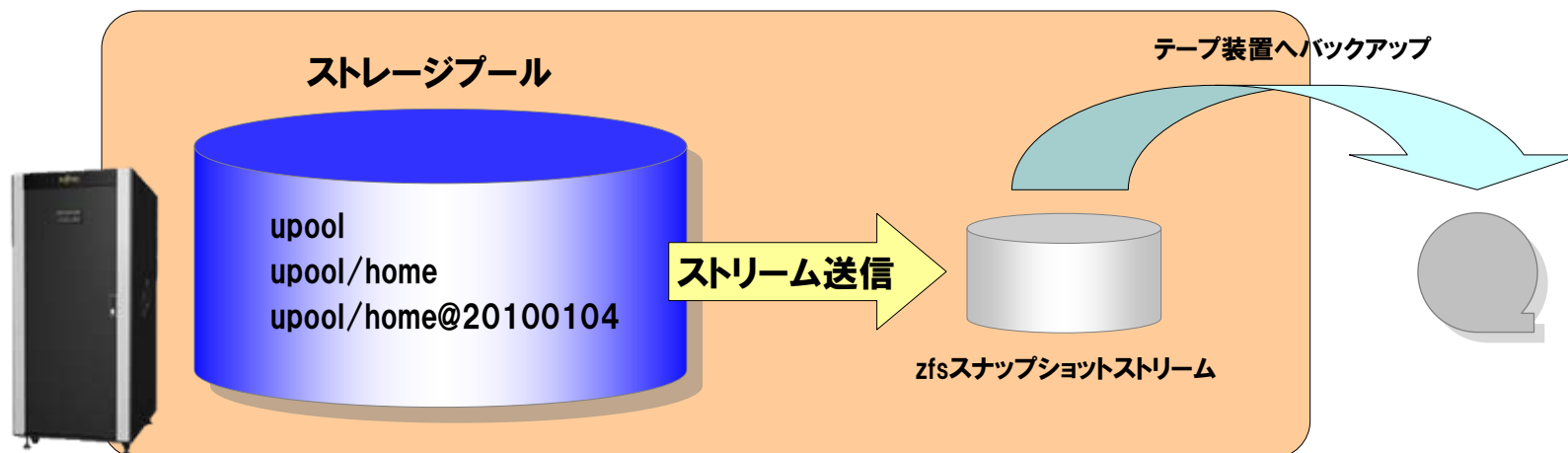
- ストリームとは複数のデータを連続する一つのデータとして扱うことであり、ZFSにおけるバックアップ/リストアを実現します

✓ ストリームの送信(バックアップ)

ZFSファイルシステムでは、スナップショットからストリームの生成および送信を行うことで、ZFSファイルシステムのバックアップとします。尚、ストリームの生成時や送信時の操作にはシステム停止は必要ありません。

ストリームの送信の種類

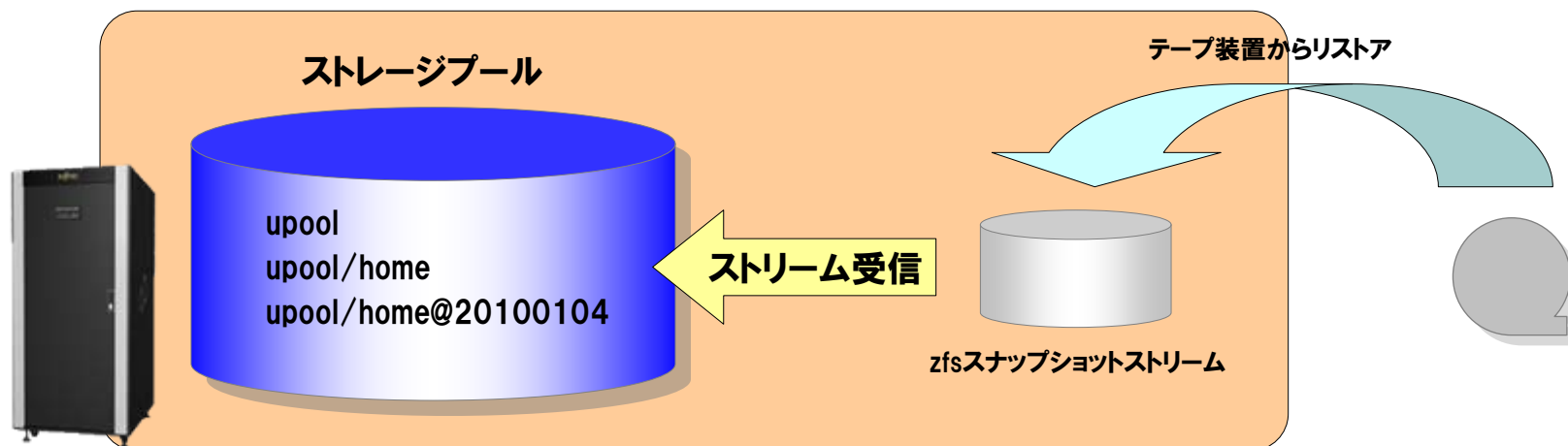
- スナップショット全体の送信 (フルバックアップ)
- スナップショットの差分送信 (差分バックアップ)



ストリームの受信

✓ストリームの受信(リストア)

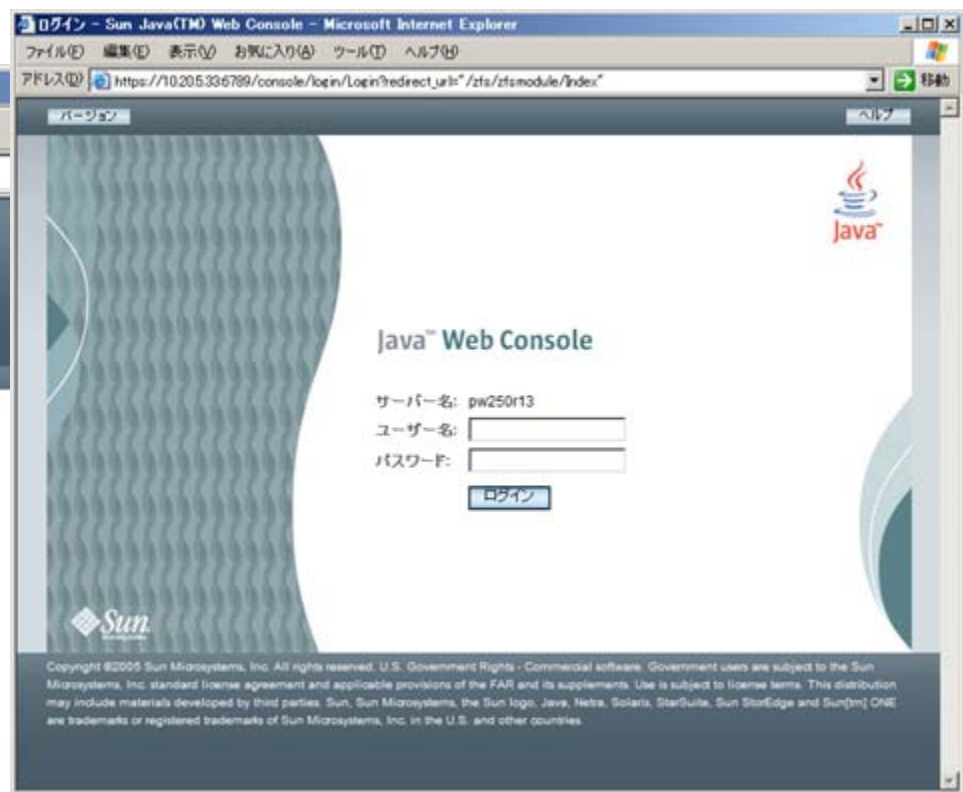
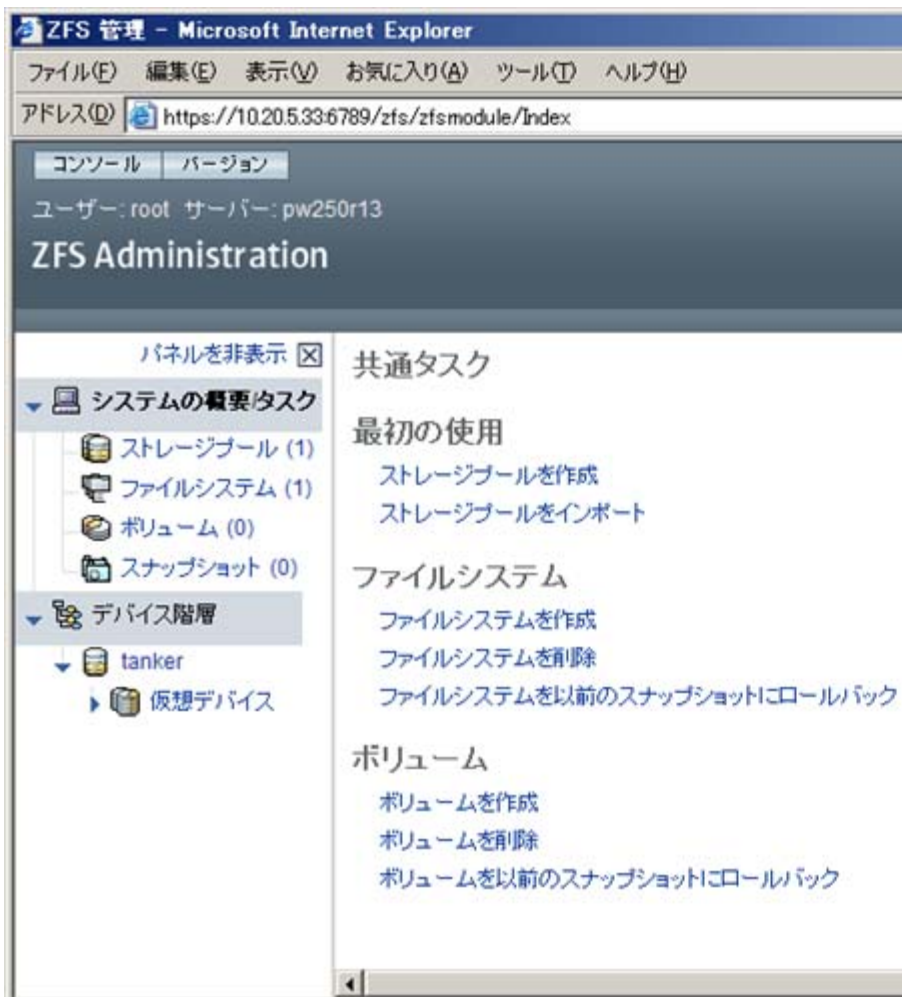
ZFSファイルシステムでは、ストリームの受信を行うことでリストアを行います。
ストリーム送信時と同様にシステム停止は必要ありません。



・ストリーム受信(リストア)中はファイルシステムとその子孫のファイルシステムにアクセスできません。

＜参考＞Web管理コンソール Sun Java Web Console

■GUI画面からもZFSファイルシステムの作成・管理が可能



スナップショットを作成
スナップショットを削除
ファイルシステムのスナップショットからクローンを作成
ボリュームのスナップショットからクローンを作成

7. ZFSの運用

ZFSファイルシステムの運用

■ZFSファイルシステムの運用

ストレージプールの領域拡張

ホットスペアの設定/管理

物理ディスクの交換

ストレージプールの別サーバへの移行

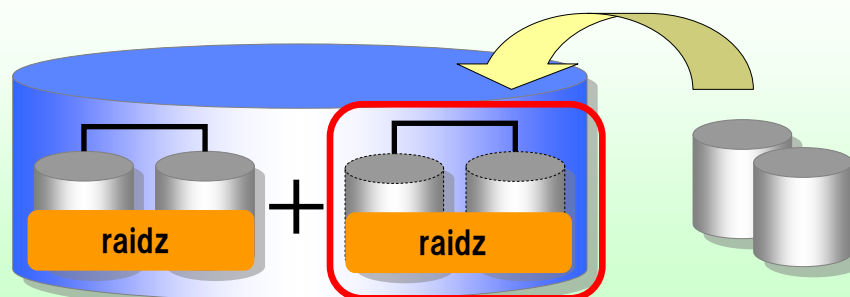
ストレージプールの領域拡張

■領域の拡張

ストレージプールへ動的にディスク領域の追加が可能です。追加した領域は、即座に使用することができます。

◆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



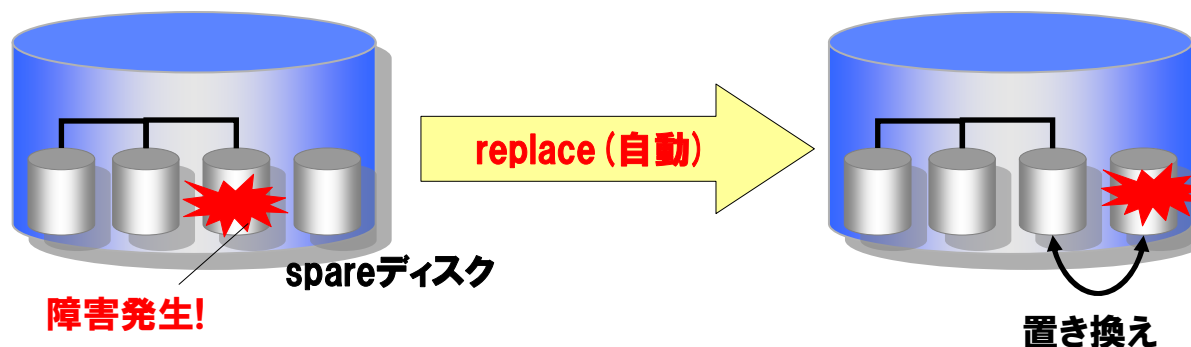
注意！

領域の縮小は行えないため、領域拡張を実施する際は注意が必要です。
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。

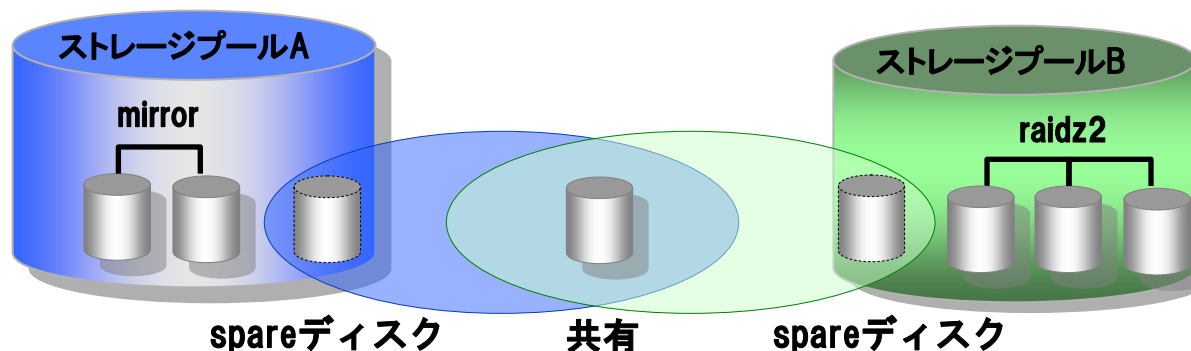
・追加するディスクグループは既存ディスクグループと同等の冗長レベルにすることを推奨します。

ホットスペアの設定/管理

- ◆ホットスペアを設定すると、ストレージプール内の物理ディスクに障害が発生した際に自動で置き換えます



- ◆ホットスペアに指定したデバイスは複数のストレージプール間で共有が可能です

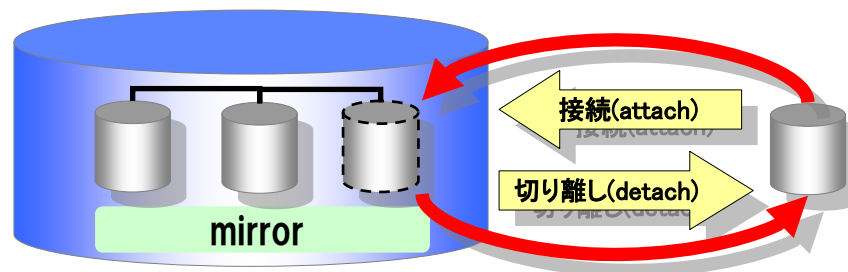


RAID-ZとRAID-Z2を使用する場合は、ホットスペアを使用することを推奨します。
 ホットスペアに使用する物理ディスクのサイズは置き換わる物理ディスク以上のサイズが必要です。

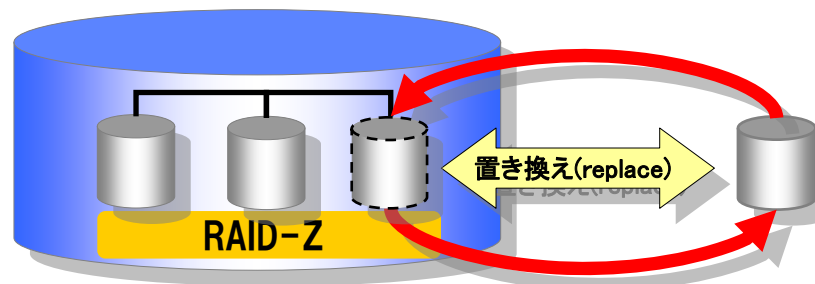
物理ディスクの交換

■物理ディスクの交換には「切り離し(detach)/接続(attach)」と「置き換え(replace)」の2つの方法があります

◆「切り離し(detach) / 接続(attach)」
※ミラー構成のみ



◆「置き換え(replace)」
※ミラー構成/RAID-Z/RAID-Z2

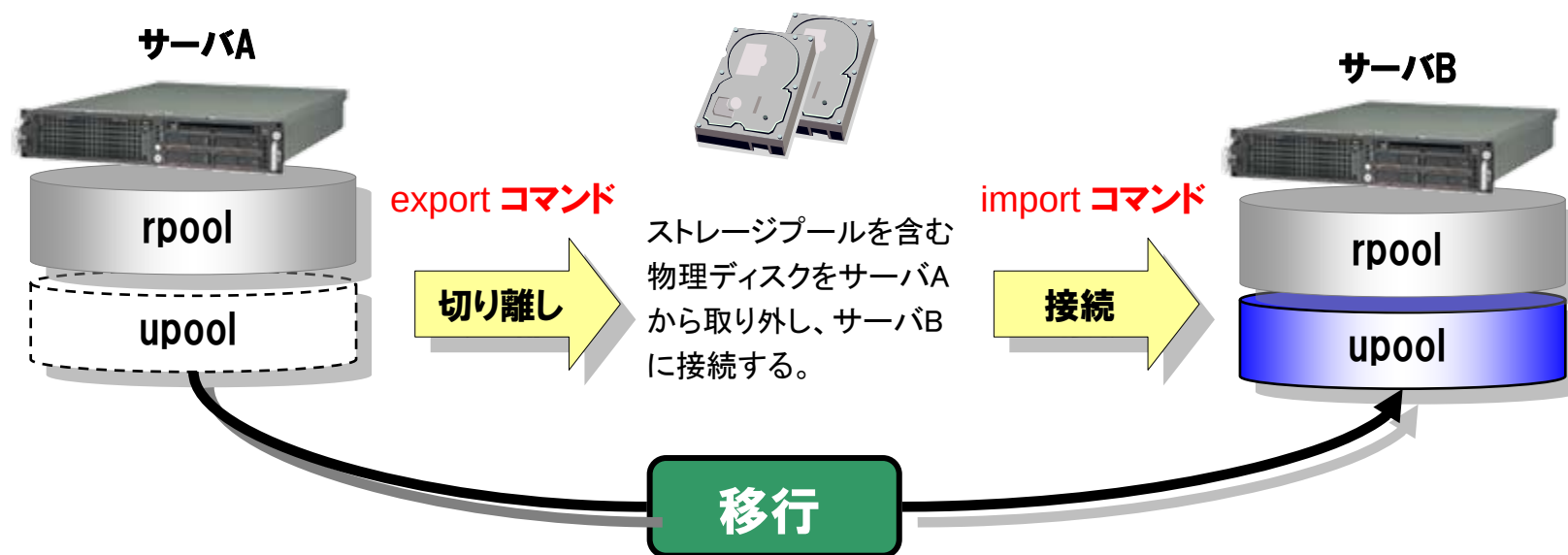


- 新しいディスクを接続すると、即座に同期化が開始されます。
- ディスクを置き換えるときは、元のディスクと同等かそれ以上のサイズのディスクを使用してください。

ストレージプールの移行

■ストレージプールの移行

ストレージプールを構成している物理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移動が容易に行えます。



- 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。インポートを実行することで、ストレージプールが復元されます。
- ルートプールの移行はできません。

＜参考＞ZFSのメモリキャッシュチューニング

■ZFSのプライマリキャッシュであるARCが使用するメモリ

ARCが使用するメモリの最大値は、搭載している物理メモリの3/4もしくは、1GBを残した全部のいずれか大きい方です。また、最小値は、64MBかARCの最大値の8分の1のいずれか大きい方です。

```
# mdb -k
Loading modules: [ unix genunix specfs dtrace zfs ldc ip hook neti sctp arp usba fctl
nca lofs md cpc random crypto fcip logindmux ptm ufs sPPP nfs ]
> ::arc
p          = 5009 MB
c          = 10019 MB
c_min      = 1252 MB
c_max      = 10019 MB
size       = 211 MB
```

p・・・MRUとMFUのキャッシュサイズ
c・・・キャッシュとして使用可能サイズ
c_min・・・ARCが使用する物理メモリの最小値
c_max・・・ARCが使用する物理メモリの最大値
size・・・現在使用されているARCサイズ

✓ZFSのメモリチューニングは/etc/systemに設定

設定例

ARCが使用する物理メモリの最大値を10GBで設定

```
# vi /etc/system
set zfs:zfs_arc_max=[10737418240/0x280000000]
```

10進数 16進数

サイズの指定は10進数と16進数
どちらも指定可能です。

商標について

- SPARC Enterprise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。
- 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。
- Sun、Sun Microsystems、Sunロゴ、SolarisおよびすべてのSolarisに関連する商標及びロゴは、米国およびその他の国における米国Sun Microsystems, Inc.の商標または登録商標であり、同社のライセンスを受けて使用しています。
- すべての SPARC 商標は、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登録商標です。SPARC 商標が付いた製品は、Sun Microsystems, Inc. が開発したアーキテクチャーに基づくものです。
- SPARC64 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登録商標です。
- ORACLE, SQL * Plus, SQL * Forms, SQL * Net, Pro * C, Pro * FORTRAN, Pro * COBOLは、ORACLE Corporationの登録商標もしくは商標です。
- 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。

留意事項

- 本書の内容は、改善のため事前連絡なしに変更することがあります。
- 本書の内容は、細心の注意を払って制作致しましたが、本書中の誤字、情報の抜け、本書情報の使用に起因する運用結果に関しましては、責任を負いかねますので予めご了承ください。
- 本書に記載されたデータの使用に起因する第三者の特許権およびその他の権利の侵害については、当社はその責を負いません。
- 無断転載を禁じます。

