複数のボリューム監理システムのための共通 API の開発

10G473 高石諒 (最所研究室)

はじめに

- 近年，大容量のストレージが必要とされる場面が増えている
  - 扱うデータの巨大化 (1 つのディスクに収まらないデータ)
  - クラウド上の巨大ストレージ
- 複数ディスクでストレージを構成する必要がある
- 大容量ストレージを構築するには?

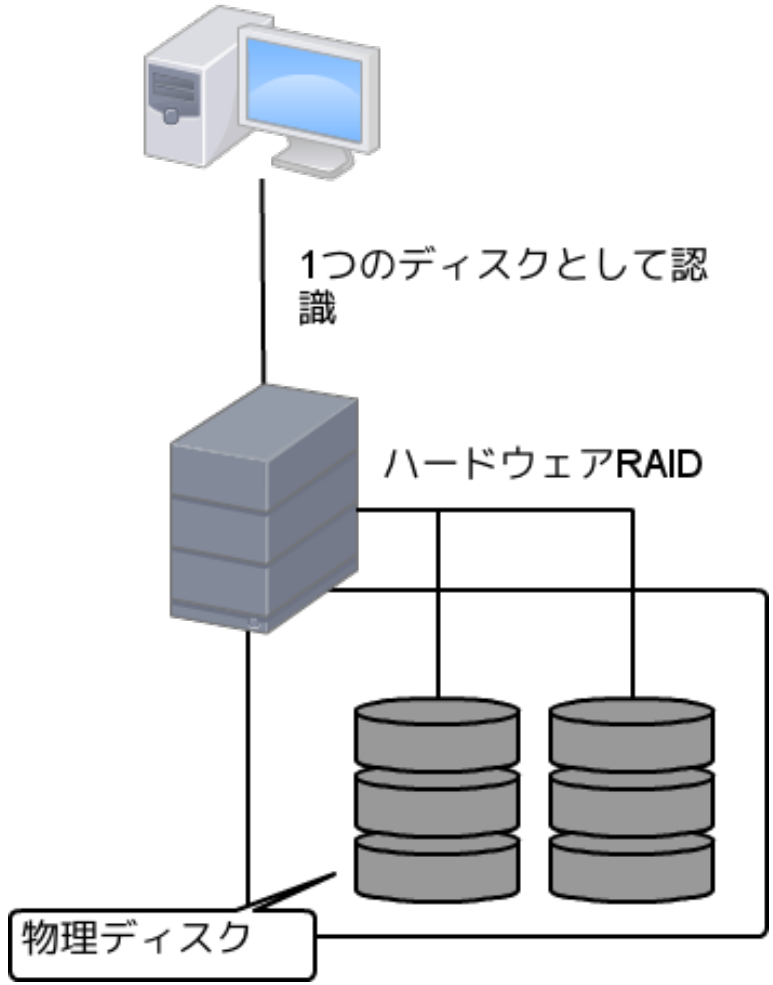
大容量ストレージの構築

ハードウェア RAID を用いる

- 単一デバイスとして表示される
- 内部には複数ディスクを持つ

ハードウェア RAID の欠点

- 専用ハードが必要
- 大容量ディスクが大量に必要だと高コスト



ボリューム管理システムを用いたストレージ構築

ボリューム管理システム

- 複数の物理ディスクを 1 つに見せる
- LVM，BtrFS，ZFS
- それぞれの操作方法，API が異なる
  - ユーザや開発者から使いにくい

共通 API の開発

- どのボリューム管理システム同じように操作できる

ボリューム管理システムについて

- 複数の物理ディスクを１つの論理的なディスクに見せる
- 物理ディスクと論理ディスクのマッピングを行っている
- LVM
  - 代表的なボリューム管理システム
  - Linux や UNIX で使われている
- ZFS，BtrFS
  - ファイルシステムにボリューム管理機能が内蔵

LVM について

- 物理ボリューム (PV)
  ⇒ 物理的なディスク
- ボリュームグループ (VG)
  ⇒ PV をまとめたもの
- 論理ボリューム (LV)
  ⇒ VG から必要な容量を切り出したもの
- LV のサイズは動的に変更可能

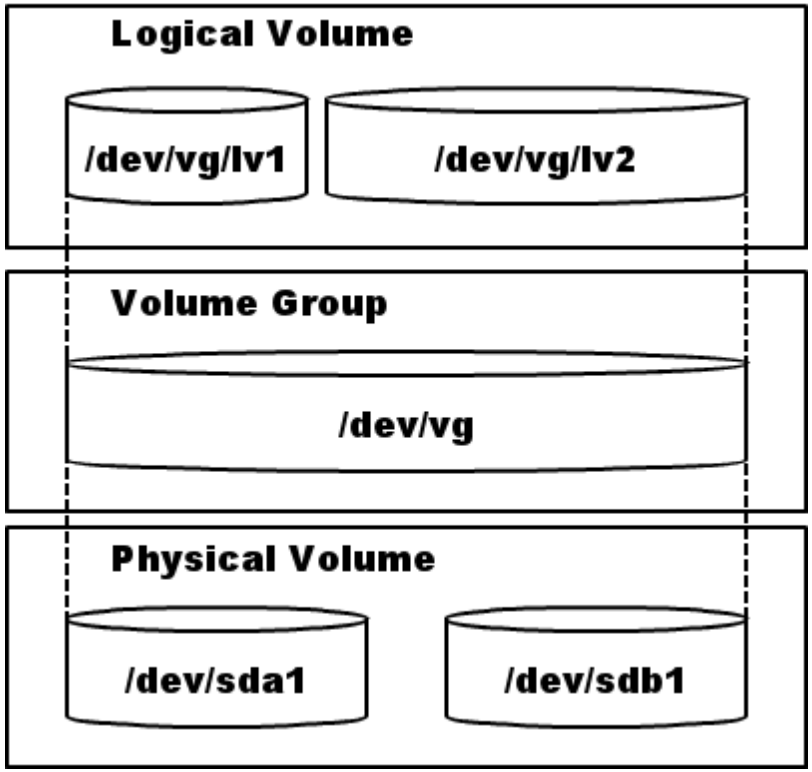


Figure: LVM の概要

BtrFS，ZFS

- ファイルシステム
- ボリュームとファイルシステムのサイズ変更を同時に行うことができる
- ボリュームマネージャ機能は LVM と似たような構造
  - 複数ディスクをまとめ，必要な容量を切り出して使用する

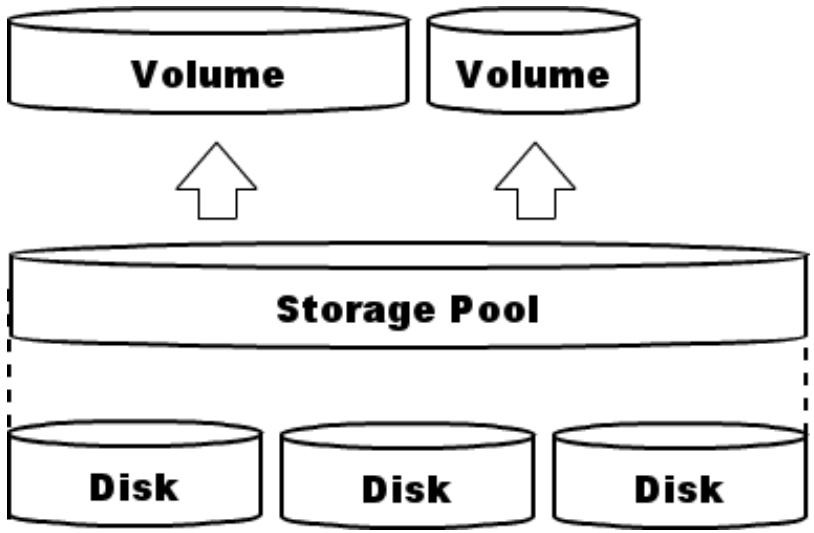


Figure: ZFS の概要

LVM と BtrFS，ZFS の違い

LVM

- ボリューム管理のみを行う
- ファイルシステムを自由に選択することが可能
- 論理ボリュームのサイズ変更時，ファイルシステムのサイズ変更を別に行う必要がある
- ファイルシステムがサイズの変更に対応している必要がある

BtrFS，ZFS

- ボリュームの拡張とファイルシステムの拡張を同時に行う
- ファイルシステムを選ぶことはできない

分散ファイルシステム (参考)

- 分散ファイルシステムにおいてもボリューム管理機能が用いられている
- GlusterFS
  - 各サーバのストレージをまとめてストレージプールとする
  - プールから必要な容量のボリュームを切り出す

LVM と同様の構造

- 各サーバのストレージ = 物理ボリューム
- ストレージプール = ボリュームグループ
- 切り出したボリューム = 論理ボリューム

ボリューム管理システムの多様化

様々なものが登場

- LVM だけでなく，ファイルシステムが独自で実装
- 仮想ボリューム管理機能持つファイルシステムの登場
- それぞれのボリューム管理システムのコマンドや API が異なる

API が異なることによる問題

1. 学習コストの増加
   ⇒ LVM から ZFS に移行する際に，操作方法が異なると再学習する必要がある
2. システム開発におけるコストの増加
   ⇒ 各ボリューム管理システム対応する必要が生じる

共通 API の開発

- 共通 API を開発することで，問題の解決を行う

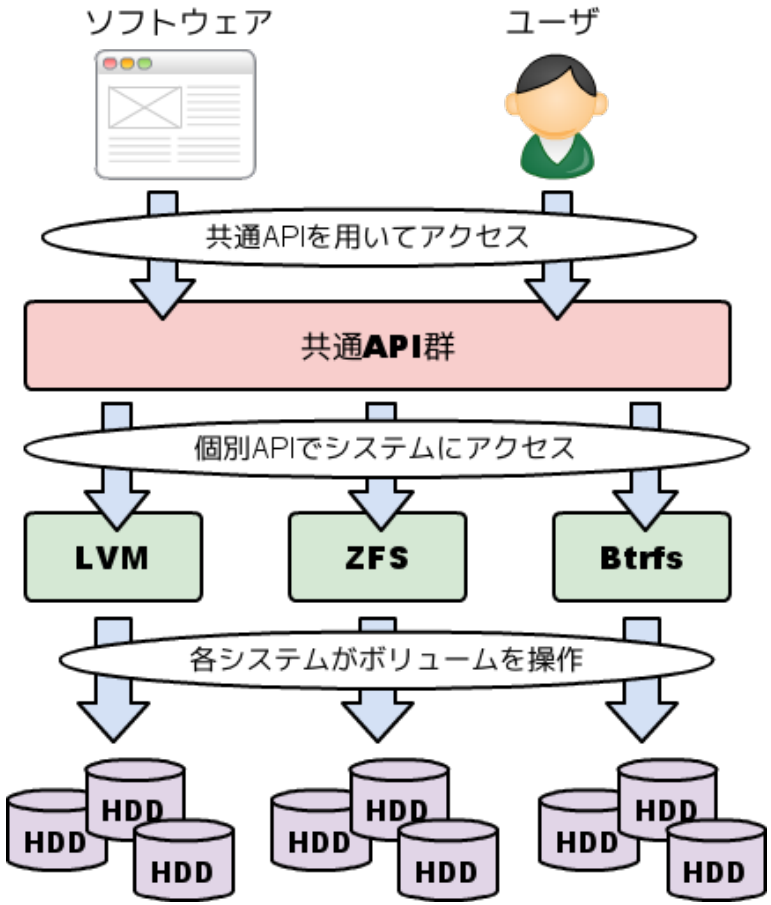


Figure: 共通 API

メリット・デメリット

共通 API を利用するメリット

- ボリューム管理システム変わってもそれ以前と同じように操作することができる
- ボリューム管理システムを操作するソフトウェアを作成する場合，個別に対応しなくても共通 API のみ対応すれば全て利用できる

デメリット

- ボリューム管理システム固有の機能が使えなくなる可能性がある

ボリュームに関する用語の定義

ボリューム管理システム用語が異なるため，以下のように統一する．

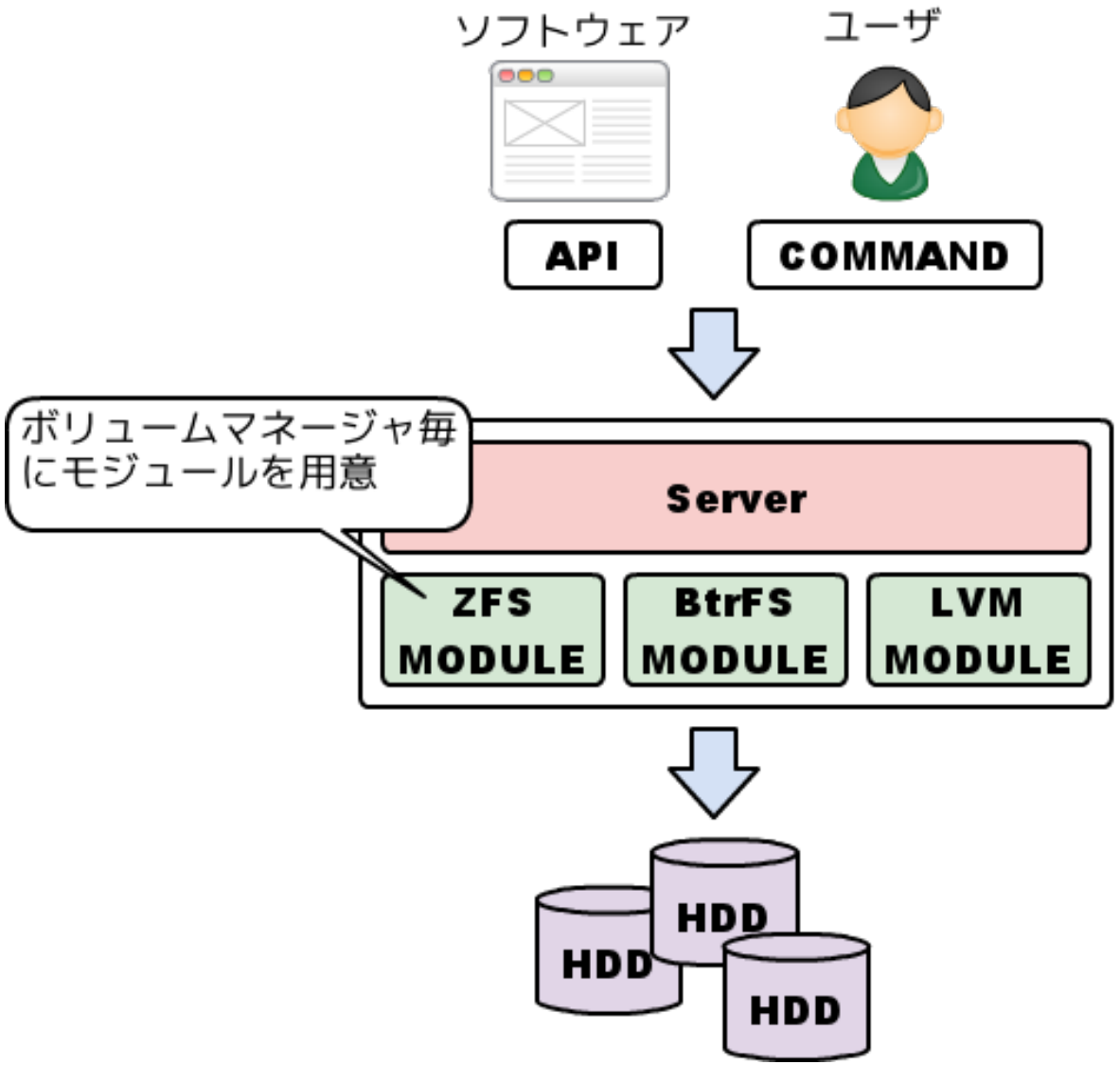
- ボリュームマネージャ
  ボリューム管理機能の総称．LVM そのもの．
- ブロックデバイス
  ハードディスクを指す．LVM における物理ボリューム．
- ストレージプール
  ブロックデバイスをまとめたもの．LVM におけるボリュームグループ．
- ボリューム
  ストレージプールから必要な容量を切り出したもの．LVM における論理ボリューム．

システム概要

共通 API は，以下の要素で構成される．

- ボリューム管理サーバ
  ブロックデバイス・ストレージプール・ボリュームの管理を行う
- ボリュームマネージャモジュール
  ボリューム管理システムに対応するためのモジュール．
- API，コマンド
  ユーザ・ソフトウェアが共通 API を操作する

システム概要


ソフトウェア
ユーザ
API
COMMAND
ボリュームマネージャ毎にモジュールを用意
Server
ZFS MODULE
BtrFS MODULE
LVM MODULE
HDD
HDD
HDD

共通 API の対象機能

ストレージプールの操作

- ストレージプールの作成
- プールへのデバイスの追加・削除

ボリュームの操作

- プールからボリュームを切り出す
- ボリュームのサイズ変更

スナップショット

- ストレージのあるタイミングの状態を記録する機能
- 各ボリュームマネージャが持っている

例 1: ストレージプールの作成

- 共通 API
  make-pool -t type poolvolume-name device-path [device-path. . .]
- LVM
  pvcreate /dev/sda1 /dev/sdb1
  vgcreate VOLUME-NAME /dev/sda1 /dev/sdb1
- BtrFS
  mkfs.btrfs /dev/sda1 /dev/sdb1
- ZFS
  zpool create VOLUME-NAME /dev/sda1 /dev/sdb1
- GlusterFS(参考)
  gluster peer probe server1
  gluster peer probe server2

例 2:ボリュームの作成

- 共通 API
  make-volume volume-name -s size subvolume-name
- LVM
  lvcreate -L size -n DATA VOLUME-NAME
  mkfs /dev/VOLUME-NAME/DATA
  mount /dev/VOLUME-NAME/DATA /data
- BtrFS
  mkdir /data mount -t btrfs /dev/sda1 /data
- ZFS
  zfs set mountpoint=/data VOUME-NAME
- GlusterFS(参考)
  gluster volume create NEW-VOLUME transport tcp
  server1:/exp1 server2:/exp2
  gluster volume set VOLUME

今後の予定

共通 API の開発

- LVM と BtrFS に対応したものを開発
- GUI を用いたボリューム操作
  - (共通 API を用いたシステム)

評価

- 同じ操作で扱うことができることの確認