

# 異なるボリューム管理システムを透過的に扱うための共通 API とそれを用いたボリューム監視システムの開発

10G473 高石諒 (最所研究室)

LVM や ZFS のようなボリューム管理システムを透過的に扱う共通 API と必要に応じて領域を拡大することでストレージの余剰領域を削減するボリューム監視システムの設計と評価について述べる。

## 1 はじめに

近年、クラウドコンピューティングのみならず一般過程でも、1台のディスク容量を越えるストレージを要求することが多くなっている。この要求に対し、複数のハードディスクを連結し1つのハードディスクに見せかけるとソフトウェアとして LVM や ZFS, Btrfs といったボリューム管理システムが開発されている。ボリューム管理システムは、ブロックデバイスと呼ぶ物理的なディスクをまとめたストレージプールという単位で管理し、そこから必要な容量のボリュームを切り出して使用する。ボリューム管理システムのコマンドや API はそれぞれ異なるため ユーザの学習コストの増加や、システム開発におけるコストの増加という問題が発生している。このような問題を解決するため、本研究では複数のボリューム管理システムを透過的に操作ができる共通 API を提案する。

また、ストレージのリソース管理は重要な課題である。例えば、ディスクをパーティションに分けて運用する場合、どのように分割するのがよいかを事前に判断するのは困難である。また、オンラインストレージのようなシステムでは、容量を多く使うユーザやほとんど使わないユーザに分かれ、使われていない領域は余剰領域となってしまう。これを解決するために、本研究では共通 API を用いて必要に応じてボリュームを拡大することでストレージリソースを効率良く運用するためのシステムであるボリューム監視システムの開発を行う。

## 2 共通 API

共通 API の概要を図 1 に示す。アプリケーションやユーザは、共通 API を経由することで、どのボリューム管理システムでも操作できるようになる。共通 API と各ボリューム管理システム対応付けは、モジュールによって実現する。モジュール化することで、新しい

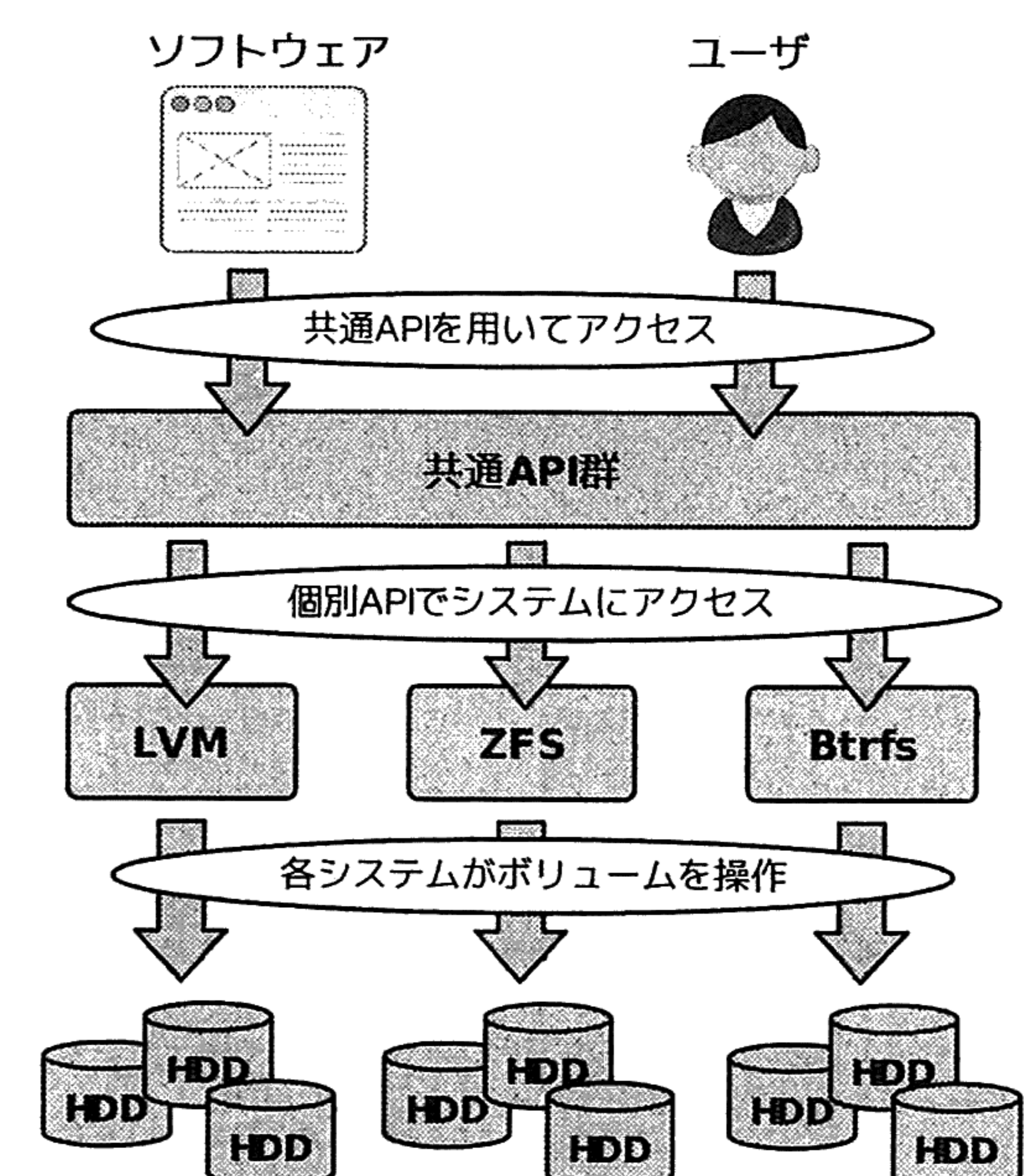


図 1: 共通 API

ボリューム開発された際にもすぐに対応することができる。今回は、ブロックデバイス操作 (一覧・作成・削除)、ストレージプール操作 (一覧・作成・削除、プールへのデバイスの追加・削除)、ボリューム操作 (一覧・作成・削除、リサイズ) に対応することとした。共通 API, LVM と Btrfs 用のモジュール、共通 API を用いてボリューム操作を行うためのコマンド群を実装し、コマンドから LVM と Btrfs を操作できることを確認した。

## 3 ボリューム監視システム

図 2 にボリューム監視システムの動作概要を示す。ボリューム監視システムは定期的にボリューム使用量を監視し、ボリュームの空き容量が条件を満たしたら拡大を行う。これにより、常に使用量に応じた領域を割り当てることができ、ストレージの余剰領域を減らし、効率的に運用することが可能になる。拡大の条件には、指定した容量以下かどうかで判断する単純比較や、最小二乗法や指数平滑法を用いた使用量の予測値が指定容量以下かどうかで判断する手法が考えられる。シミュレーションを用いて 3 つの手法とこれらを組み合わせ合わせたものの比較を行った結果、単純比較と最小二

乗法を併用する手法を採用することとした。必要に応じて領域を割り当てる類似システムとして、クラウド向けのストレージ構築システムである AoDI[1] がある。LVM をベースに独自のボリューム管理システムを作っており、共通 API を用いて複数のボリューム管理システムで使用できる事を目指す本研究とは方向性が異なると言える。

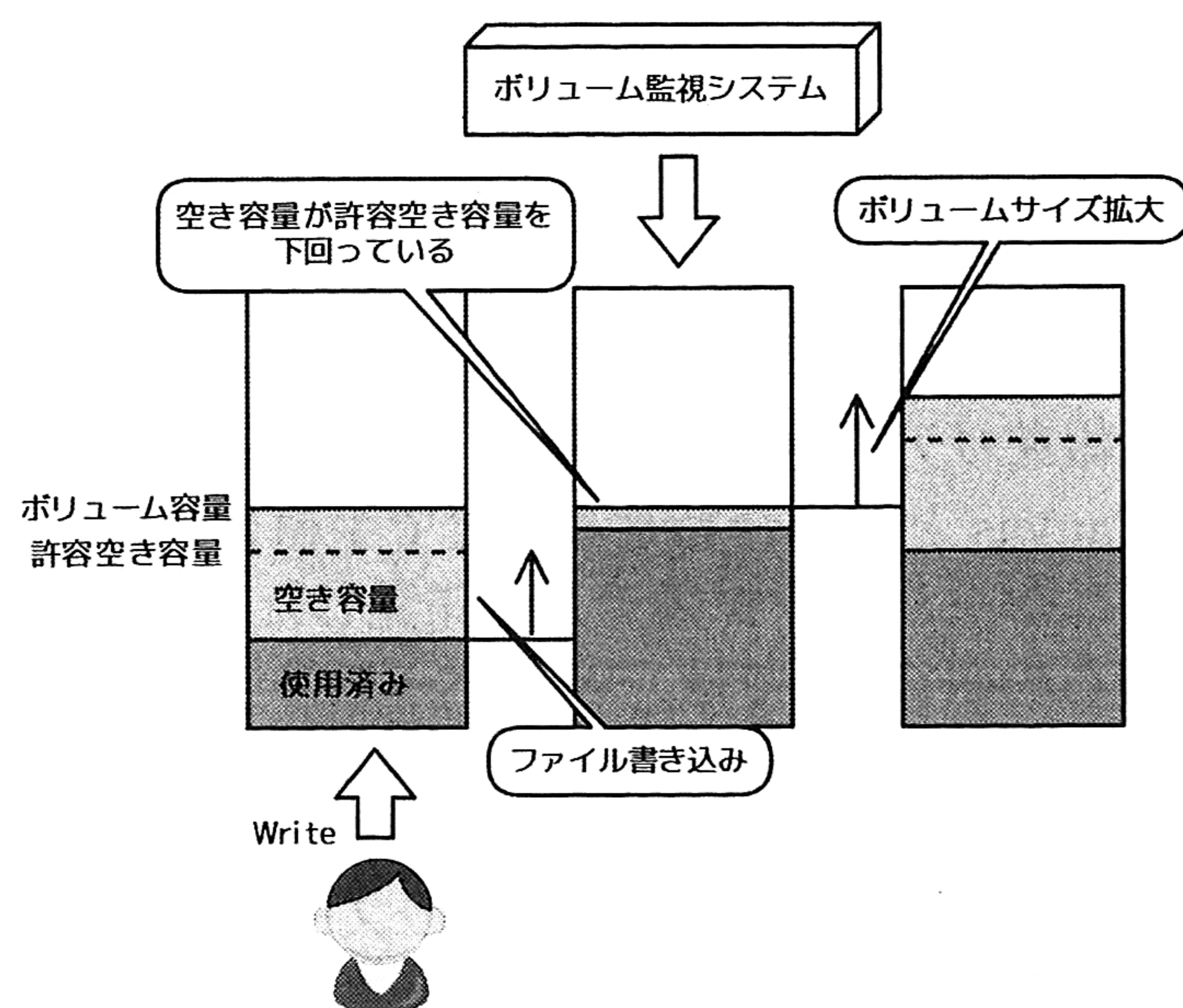


図 2: ボリューム監視システムの動作概要

### 3.1 機能評価

実装したシステムの機能評価として、実際にボリュームの監視を行い、空き容量の変化を調べた。実験の期間は1週間、実験開始時のボリュームサイズは250MB、使用量の増加量は8KB/s、許容空き容量は25MB、監視間隔は20分、ボリュームの拡大サイズは150MBとした。図3と表1に空き容量の変化について実機による実験の結果を示す。空き容量が許容空き容量である25MB付近になると、ボリュームの拡大が行われて、空き容量が増えていることが確認できる。許容空き容量を越えた時に拡大している場合と、越えていなくても拡大している場合があるが、これから単純比較による拡大決定と最小二乗法を用いた予測による拡大決定の両方が動作していることが分かる。

表 1: 空き容量が0以下・上限・下限を越えた回数

0 以下	下限	上限	合計
0	13	122	135

## 4 まとめ

異なるボリューム管理システムを透過的に扱うための共通 API を提案した。また、ボリューム管理シス

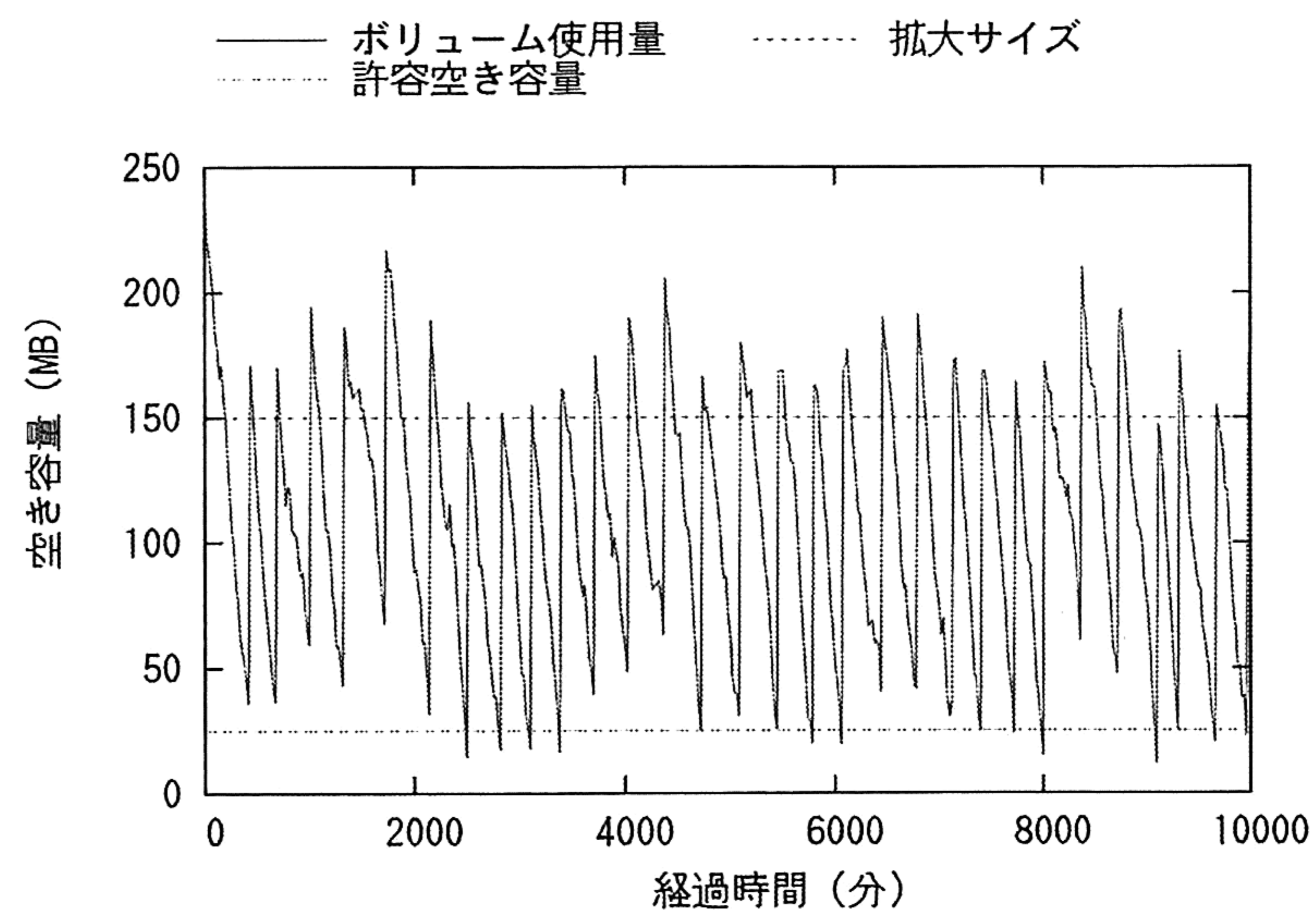


図 3: 空き容量変化の比較

テムを用いてストレージの過剰空き領域を減らし、リソースの効率化を行うためのボリューム監視システムの開発を提案した。共通 API はモジュールを用いて各ボリューム管理システムの操作方法・API の違いを吸収する。ボリューム監視システムは、定期的にボリュームの空き容量を測定し、その値か、過去の測定値も用いた最小二乗法による予測値が一定以下になっていればボリュームの拡大を行う。共通 API, ボリューム監視システム共に機能評価を行い、実用性について確認することができた。共通 API とボリューム監視システムの今後の主な課題を以下に述べる。

- 共通 API
  - ZFS, GlusterFS 用の共通 API モジュールの開発
  - より汎用的な共通 PAI の提供  
今回は Ruby で実装したが、より多くの言語から使いやすくするために C 言語で実装する必要がある。
- ボリューム管理システム
  - ボリューム使用量の増加度に応じた拡大機構の実装
  - 急激な容量増加への対応

## 参考文献

[1] R. Cao, C. Zhen, Y. Gao, G. Xu, X. Liu, G. Wang, and G. Xie. "aodi: an allocation-on-demand incremental volume based on lvm". In *Proc. the 2011 ACM Symposium on Applied Computing*, pp 133–138, 2011.