B-031

# オブジェクトベースストレージデバイスでのオブジェクト保存手法の評価

## Evaluation of A Method to Store Objects on the Object-based Storage Device

藤田 智成† 森田 和孝† 盛合 敏†
Tomonori Fujita Kazutaka Morita Satoshi Moriai

### 1．はじめに

ここ数年，企業が扱わなければいけないデータ容量は爆発的に増加しているが．その要求を満たすストレージ技術の一つとして，ハイパフォーマンスコンピューティング分野で使われている並列ファイルシステム技術が注目を集めている．並列ファイルシステムは，複数の計算機から構成され，各計算機に搭載されたディスクを束ねて利用することで，大容量のストレージシステムを実現する．

多くの並列ファイルシステムが，クライアントとデータを格納する計算機間とのプロトコルとして，オブジェクトベースストレージデバイスプロトコル[1]を用いている．本稿では，オブジェクトベースストレージデバイスにおける，オブジェクト保存手法の性能評価結果を報告する．

### 2．オブジェクトベースストレージデバイス

図 1 に，並列ファイルシステムの一般的なアーキテクチャを示す．並列ファイルシステムは，多数のストレージサーバと一台または少数のメタデータサーバから構成される．並列ファイルシステムのクライアントは，それぞれが独立に，ストレージデバイスに直接アクセスし，データを読み書きすることができる．ディレクトリ操作などのメタデータ操作は，メタデータサーバを介して実行される．並列ファイルシステムは，メタデータ操作とデータ操作を分離することで，データ操作を並列化し，クライアント数に対する高いスケーラビリティ，高性能を実現している．

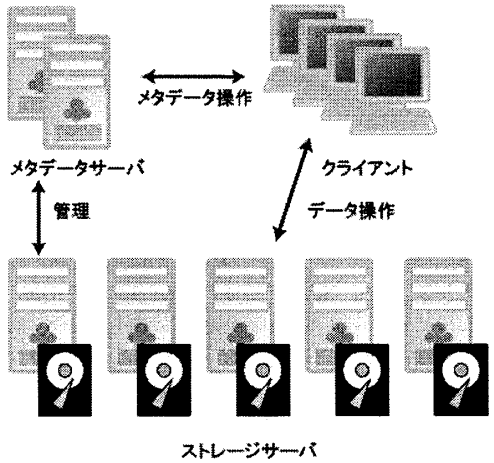


図 1 並列ファイルシステムの一般的なアーキテクチャ

† 日本電信電話株式会社 NTT サイバースペース研究所
NTT Cyber Space Laboratories, NTT Corporation

Lustre[2]，pNFS[3]，Panasas Parallel File System[4]，Ceph[5]など，多くの並列ファイルシステムが，データサーバとして SCSI デバイスの一種である，オブジェクトベースストレージデバイス（Object-based Storage Device）を用いている．

従来の SCSI ブロックデバイス（ディスク）はセクタと呼ばれる固定長のデータ単位で，クライアントはデータを読み書きできる．一方，OSD はオブジェクトと呼ばれる可変長のデータ単位でのアクセスインターフェイスをクライアントに提供する．クライアントは，オブジェクト識別子を使って，指定したオブジェクトを読み書きできる．

OSD は，従来のファイルシステムが実装していたストレージデバイスにおけるデータの物理レイアウトの管理機能を提供し，ファイルシステムの実装を簡素化する．

### 3．OSD におけるオブジェクト保存手法

OSD は，クライアントに対してはオブジェクトベースのアクセスインターフェイスを提供するが，内部では，従来のブロックデバイスにオブジェクトをデータとして保存する．

OSD のオブジェクト保存手法は，オブジェクト識別子とオブジェクトの組への高速なアクセスを実現する機能に加え，複数のオブジェクトをアトミックに更新する機能，停電などの予期せぬ障害に対しても矛盾した状態になることを防ぐ，などの機能を持たなければならない．

Linux と汎用 PC を使って OSD を実現するために，上記の要求を満たすオブジェクト保存手法に利用できるソフトウェアとして，Linux の汎用ファイルシステム Btrfs と Oracle 社の Berkeley DB（BDB）に着目し，その性能特性を評価する．

### 4．評価

#### 4.1 評価環境

Btrfs は，次世代の Linux の標準ファイルシステムとして活発に開発が進められており，スナップショット，トランザクション，データ整合性保護などの先進的な機能を備えている．

我々は，オブジェクト識別子をファイル名として管理することで，識別子とオブジェクトの管理を実現した．オブジェクトの大きさに関係なく，1 つのオブジェクトを 1 つのファイルとして管理した．

BDB は，キー・バリュー型のインターフェイスを提供するオープンソースのデータベースソフトウェアである．

我々は，オブジェクト識別子をキー，オブジェクトをバリューとして扱うことで，識別子とオブジェクトの管理を実現した．

OSD プロトコルは，書き込みリクエストにおけるキャッシュ管理を，電源が切れてもデータが失われない状態になってからクライアントにリクエストの完了を返すポリシーを必須，データがメモリに保存された時点でリクエストを返すポリシーをオプションと定めている．本評価では，前者のポリシーを実現するために，Btrfs，BDB のディスクへのデータ書き込みが完了してからリクエストを完了とするようにした．

評価対象ソフトウェアには，Btrfs， BDB に加えて，OSD の要求は満たさないが，単純なデータ保存を実現するためのオーバヘッドの参考値として，Linux の標準ファイルシステムである Ext3 を加えた．

測定に用いたハードウェア構成および，ソフトウェアのバージョンを表 1 にまとめた．

測定は，ファイルシステムが使うオペレーティングシステムのページキャッシュやデータベースのキャッシュにデータが保存されていない状態から開始した．測定結果は，三回の試行の平均値である．

表 1 ハードウェア構成とソフトウェアのバージョン

| | |
|---|---|
| CPU | Xeon X5365 3.0 GHz × 2 |
| Memory | 16 GB |
| Hard disk | SCSI SAS 10,000 rpm |
| Linux kernel | 2.6.30 |
| Berkeley DB | 4.6.21 |

## 4.2 評価結果

図 2 は，10,000 個のオブジェクト作成の測定結果である．オブジェクトサイズが 64 KB 以下の場合，BDB の性能は，2 つのファイルシステムの性能よりも良い．特に，4 KB では，BDB の性能は，ファイルシステムの約 10 倍である．一方，オブジェクトサイズが大きくなるにつれて，ファイルシステムの性能は向上するが，BDB の性能はほとんど向上せず，オブジェクトサイズが 512 KB では性能の低下が確認された．Ext3 と Btrfs の性能差は 2 倍弱であった．

図 3 は，10,000 個のオブジェクトの読み込み性能結果である．オブジェクト作成性能と同様，オブジェクトサイズが小さい場合，BDB はファイルシステムよりも良い性能であり，Btrfs と比較すると，256 KB 以下では，2 倍以上の性能があった．BDB の性能は，オブジェクトサイズが 256 KB から 512 KB に増加すると，極端に低下した（71.2 MB/sec から 4.5 MB/sec）．オブジェクトサイズを 1 MB にして，BDB の性能を測定したところ，512 KB と同様に約 4 MB/sec であり，オブジェクトサイズが大きい場合，BDB の読み込み性能が低下することが分かった．Ext3 と Btrfs の性能差は，最大で約 3.7 倍となっており，オブジェクト作成での性能差よりも大きかった．

## 5．まとめ

並列ファイルシステムで用いられるオブジェクトベースストレージデバイスにおけるオブジェクトの保存手法と利用できるソフトウェアとして，Btrfs と BDB に着目し，その評価結果を報告した．オブジェクトのサイズが小さい場合は BDB，大きい場合は Btrfs の性能が良く，その性能差は 2 倍程度であることが分かった．また，オブジェクトサイズが 256 KB を超えると，BDB の読み込み性能が極端に低下することが分かった．Btrfs を単純なデータ保存方法である Ext3 の性能と比較すると，読み込み性能の差が大きく，オブジェクトサイズが 32 KB では，Ext3 は Btrfs の 3.7 倍の性能であった．

今後は，Btrfs と BDB の性能差の原因を調査し，その改善方法を検討していく．

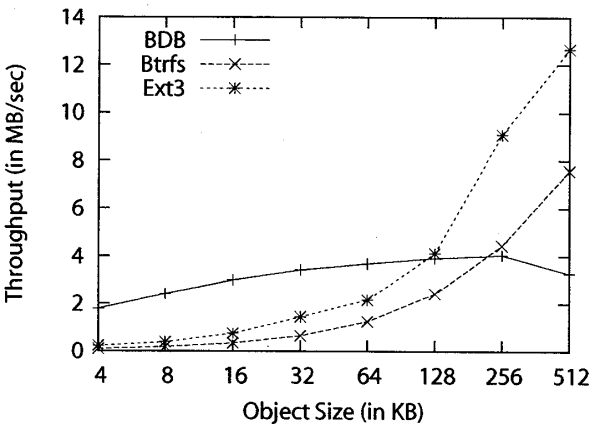


図 2 オブジェクト作成性能

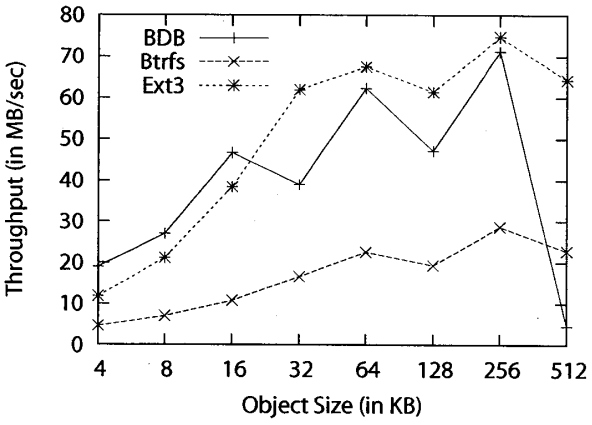


図 3 オブジェクト読み込み性能

## 参考文献

[1] Technical Committee T10. http://t10.org/.

[2] Lustre. http://www.lustre.org/.

[3] Parallel NFS. http://www.pnfs.com/.

[4] Brent Welch, Marc Unangst, Zainul Abbasi, Garth Gibson, Brian Mueller, Jason Small, Jim Zelenka, and Bin Zhou. Scalable Performance of the Panasas Parallel File System. In Proceedings of the 6th USENIX Conference on File and Storage Technologies, pp. 1-17, 2008.

[5] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D.E. Long, and Carlos Maltzahn. Ceph: A Scalabe, High-Performance Distributed File System. In Proceedings of the 7th Conference Operating Systems Design and Implementation, pp. 307-320, 2006.