2A-2

ファイルシステムのジャーナルを利用したデータ同期機構の設計と実装

湯山 圭一† 伊藤 雅典† 山中 顕次郎† 村上 明彦†
株式会社 NTT データ 技術開発本部†

1. はじめに

災害時にクラウドシステムのディザスタリカバリを行い、被災していないデータセンタでサービスを引き継ぐことが行われるようになってきている。サービスを引き継ぐためには、データベースに格納されたデータ等のユーザデータの同期が必要となる。クラウドシステム上では様々なユーザが多様なアプリケーションを動作させるので、アプリケーションにデータ同期をできるだけ意識させない方式が望ましい。また、災害の影響を避けるために離れたデータセンタへデータを同期させる必要があるので、ネットワークの遅延が大きい環境でデータ同期を行わなければならない。しかし、ネットワークの遅延が大きい環境において同期通信方式でデータ同期を行うと、同期速度が上がらないという問題がある。

著者らは、ファイルシステムレイヤでデータ同期を行う機構を設計し、プロトタイプを実装した。また、通信方式によりどの程度同期速度が向上するかを評価した。

2. 想定環境

東京―大阪程度に離れたデータセンタ間でデータ同期を行うことを想定する。ネットワークの遅延は RTT20ms とする。また、データセンタ間のネットワークの帯域は 1Gbps 程度使用できることを想定する。

クラウドシステムのディザスタリカバリを行うためには、クラウドシステム同士を制御するクラウド連携マネージャ[1]が必要になる。

3. 関連研究

データ同期機構は様々なレイヤで実装されている。以下に例を示す。

データベースレイヤでのデータ同期機構は、さまざまなデータベースマネジメントシステム(以下: DBMS)で既に実装されている。この方式は、アプリケーションが明示的に commit を行うことから、整合性を取りやすい。しかし、特定の DBMS に依存してしまうという問題がある。

ブロックデバイスレイヤでデータ同期機構を実装すると、アプリケーションを変更する必要がないという利点がある。このレイヤの実装として、たとえば DRBD[2]がある。DRBD の同期通信方式では、ネットワークの遅延が大きい環境で同期速度が上がらないという問題がある。

4. 設計と実装

本研究では、ファイルシステムレイヤにデータ同期機構を実装することで、アプリケーションの変更なしにデータ同期を実現する。またファイルシステムのジャーナルのみを送信することで、ファイル単位や一括方式のデータ同期に比べ同期速度を上げる。構成を図 1 に示す。あらかじめ同期元の同期対象のディスクのスナップショットを同期先へ転送しておく。本稿ではクラウドシステムでの利用を想定しているので、パーティションイメージを同期先へコピーする。

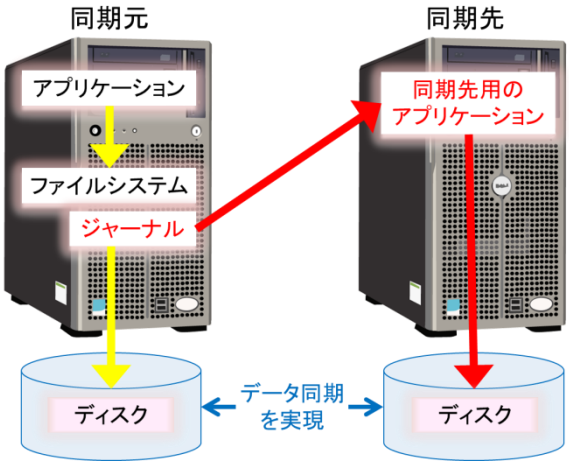

図 1 データ同期機構の構成

同期元

ディスクに書き込む際に、本来ディスクに書き込まれるファイルシステムのジャーナルを、

The design and implementation of data replication which use journal of file system.
Keiichi Yuyama†, Masanori Itoh†, Kenjiro Yamanaka† and Akihiko Murakami†
†Research and Development Headquarters, NTT DATA CORPORATION

表 1　同期速度の評価

	bonnie: Sequential Write (Per Char)	bonnie: Sequential Write (block)	pgbench
オーバラップなし 1 コネクション	144 KB/s	0.190 MB/s	4.30 tps
オーバラップなし 10 コネクション	455 KB/s	1.77 MB/s	15.1 tps
オーバラップあり 10 コネクション	641 KB/s	100 MB/s	269 tps
同期機構なし	713 KB/s	158 MB/s	770 tps

同期先にも送信する。同期通信方式では、同期元のディスクへの書き込みが終わり、同期先の書き込み完了通知を受け取ったのち、書き込み完了とする。

ファイルシステムのジャーナルには一般的に、ディスクにメタデータのみを書き込む方式と、メタデータとデータの差分を書き込む方式がある。ジャーナルを使用して同期先でデータを復元するために、同期先へはメタデータとデータの差分双方を送信する。また、データの差分をディスクに書き出すと性能が低下してしまうので、同期元のディスクにはメタデータのみを書き込む。

なお、ジャーナルのディスク書き込みと同期先への送信を直列に行うと、双方の完了を待つので同期速度が低下する。これを改善するために、ディスク書き込みと同期先への送信をオーバラップして実行する。

本研究では、Linux Kernel 2.6.37-2.fc15 を元に実装を行った。同期対象のファイルシステムとして、ext4 を使用した。

4.1.　同期先

受信したジャーナルはファイルに保存しておき、ある程度ジャーナルが蓄積したらディスクのスナップショットに適用する。これは、ジャーナル適用の負荷が大きいため、受信したジャーナルを即座に適用すると、同期通信方式では同期速度が低下するからである。

同期通信方式では、ジャーナルをディスクに書き出した後に、同期元に書き込み完了通知を送信する。

4.2.　複数の通信コネクション

ネットワークの遅延が高い環境で同期通信を行うと、応答確認で待たされる時間が長くなるため、通信コネクション当たりの同期速度が上がらない。同期速度を向上させるために、通信コネクションを複数本使用する。

5.　評価

遅延環境を再現するために、中間ノードを経由してデータ同期を行う構成とした。中間ノードで Linux の netem を実行し、20ms の遅延を発生させた。

この環境で、一般的なファイル書き込みやデータベースを動作させることを想定し、bonnie++ 1.96 と PostgreSQL 9.1.0 に付属の pgbench で計測を行った。結果を表 1 に掲載した。ジャーナルのディスク書き込みと同期先への送信をオーバラップさせることで、小さいデータの書き出し処理であれば、1 割程度の性能低下で同期通信によるバックアップを行うことができた。

6.　まとめと今後の課題

ファイルシステムレイヤにデータ同期機構を実装することで、アプリケーションの変更なしにデータ同期を実現した。同期元にてディスク書き込みと同期先への送信をオーバラップして実行し、また同期用の通信コネクションを複数用意することで、同期速度を向上させた。しかし、比較的大きいデータの Sequential Write や pgbench の計測結果は依然として遅いことから、これらの理由を明らかにした上で性能を改善することが今後の課題である。

謝辞

本研究は総務省「クラウドサービスを支える高信頼・省電力ネットワーク制御技術の研究開発（高信頼クラウドサービス制御基盤技術）」委託研究による研究成果です。

参考文献

[1] 武田 健太郎、伊藤 雅典、山中 顕次郎、村上 明彦：“複数クラウド間でスケールアウトやディザスタリカバリを実現するクラウド連携マネージャの設計と実装”、情報処理学会、第72回全国大会講演論文集、″3-35″-″3-36″、2010。

[2] DRBD.jp by Thirdware inc.、http://www.drbd.jp/。