3P-2

REST ベースのオンラインストレージへのアプリケーションのデータ移行

平井徹也† 福田洋治‡ 毛利公美†† 白石善明†
名古屋工業大学† 愛知教育大学‡ 岐阜大学††

1. はじめに

大規模なデータの紛失を伴う情報システムの破壊が自然災害で起きることが広く認識されつつある．クラウドサービスで新しい業務環境の提供はできるが，ローカルに構築しているシステムが被害を受けたときに，環境を従前通りに復元するのは容易ではない．業務を速やかに再開するには，アプリケーションサーバやクライアントの設定ファイル，データを安全に管理し，システム環境の復旧に備えておく必要がある．

従来のシステム環境を復旧するサービスに「瞬快」[1]，「NetSHAKER for School」[2]などがある．これらは再起動時に自動で，あるいは 1 台のホストから手動でリモート操作することでシステム環境の復旧を行う．これらのサービスは，システム環境の復旧のためにローカルストレージにシステムファイルのバックアップファイルを作成する．つまり，バックアップファイルが紛失した場合はシステム環境の復元ができない．

本研究では，設定ファイルやデータをはじめとするファイルの安全な保護の手段として，オンラインストレージに注目する．オンラインストレージでは，アプリケーション開発のために REST（Representational State Transfer）ベースの API が提供されているものがある[3][4]．REST では HTTP と URI によって動作が定義できる．

システムで使用しているローカルなファイルやデータを REST ベースのオンラインストレージへ移行するとき，ローカルストレージへの処理要求を HTTP 要求に書き換えなければならない．本稿では，オンラインストレージへのアプリケーションのデータ移行を支援するための，ローカルストレージへの処理要求を HTTP リクエストへ変換する手法を提案する．

変換手法の一実現法として，ローカルストレージでのファイルの読み込み／書き込みをオンラインストレージに対する操作に変更するソフトウェアを実装する．変更されたソフトウェアは，ネットワークを介したデータ操作を行うので，変更前よりもアクセス遅延が生じる．そこで，ローカルストレージにデータを保存し，操作を行うキャッシュ機能を実装する．キャッシュ機能の有無により，オンラインストレージでデータ管理を行うアプリケーションの遅延がどの程度軽減されるかを検討する．

2. REST ベースのオンラインストレージ

オンラインストレージは，インターネット上でファイル保管用のディスクスペースを貸し出すサービスである．オンラインストレージは，組織の規模に応じたストレージ容量の契約ができる，ハードディスクの故障と関係なくバックアップがとれるといった特徴がある．オンラインストレージを用いたアプリケーション開発のために，REST ベースの API が提供されているサービスがある[3][4]．

REST ベースのオンラインストレージでは，HTTP メソッドを GET に指定した HTTP リクエストを送信し，返ってきた HTTP レスポンスのボディ部分に付加されたデータを取得することでファイルの読み込みを行う．HTTP メソッドを POST に指定し，HTTP リクエストのボディ部分にデータを付加し送信することで，ファイルの書き込みを行う．

3. 提案手法

3.1. データ操作方法の変更

一般に，サービス利用者のアプリケーションクライアントやサービス提供者のサービスサーバの設定ファイルはクライアントやサーバのソフトウェアの動作と密接に関連しており，図 1 に示すようにサービス利用者とサービス提供者の各々のローカルストレージでデータを管理している．REST ベースのオンラインストレージでデータ管理を行う場合，図 2 に示すようにオンラインストレージを管理するサーバに HTTP リクエストを送り，HTTP レスポンスを受け取ることでデータ操作を行う．このようにローカルストレージとオンラインストレージでは，データの操作方法が異なるため，データ移行をするにあたり，ローカルストレージへの処理要求を HTTP リクエストに書き換える必要がある．

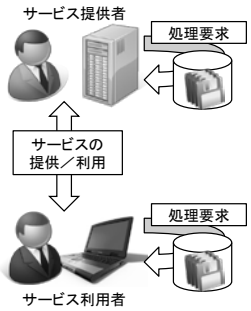

図 1 ローカルストレージでのデータ利用

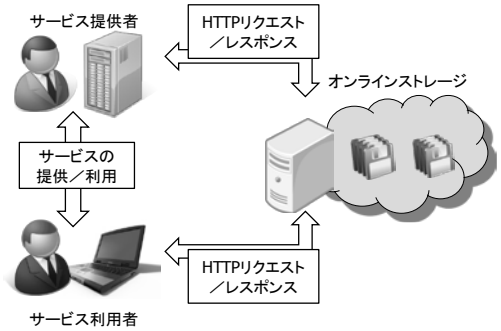

図 2 REST ベースのオンラインストレージでのデータ利用

3.2. 変換手法

ローカルストレージからオンラインストレージへのファイル読み込み／書き込みの操作方法の変換は以下の手順で行う．

Step1. 書き換えを行うプログラムを読み込み，ファイル読み込み／ファイル書き込みを行っている部分を検出
Step2. 操作するファイルの対象をローカルストレージからオンラインストレージに変更するために，ファイルパスを URL に変更
Step3. ファイル読み込みの場合は HTTP メソッドを GET に，ファイル書き込みの場合は HTTP メソッドを POST に指定
Step4. 指定した URL と HTTP メソッドで，オンラインストレージに対してリクエストを送信

4. 既存プログラムを書き換えるソフトウェアの実装

ローカルストレージでデータ操作を行う既存プログラムからオンラインストレージでデータ操作を行うアプリケーションへの書き換えが容易にできれば開発者の支援につながる．変換手法の変換を支援する方法の一つとして，ローカルストレージでファイル読み込み／書き込みを行う Java プログラムから REST ベースのオンラインストレージでファイル読み込み／書き込み

A Data Migration of Application to REST-Based Online File Storage Service
† Tetsuya HIRAI and Yoshiaki SHIRAISHI・Nagoya Institute of Technology
‡ Youji FUKUTA・Aichi University of Education
†† Masami MOHRI・Gifu University

を行う Java プログラムへ変換するソフトウェアを実装した．ソフトウェアの動作は次のようになる．

Step1. 書き換えるプログラムを読み込み，ファイル走査をし，FileReader クラスのオブジェクト，FileWriter クラスのオブジェクトを検出

Step2. FileReader クラスのコンストラクタ／FileWriter クラスのコンストラクタの引数となっているファイルのファイルパスを取得

Step3. “オンラインストレージの URL”を指し示す URL クラスのオブジェクトを作成するコードを追加

Step4. URL が参照するリモートオブジェクトへの接続を表す HttpURLConnection クラスのオブジェクトを作成するコードを追加

Step5. HTTP メソッドを読み込みの場合は GET，書き込みの場合は POST に指定するコードを追加

Step6. 通信リンクを確立するメソッドを呼び出すコードを追加

Step7. 読み込みの場合， FileReader クラスの入力ストリームから HttpURLConnection クラスの入力ストリームに変換．書き込みの場合， FileWriter クラスの出力ストリームから HttpURLConnection の出力ストリームに変換

Step8. 通信リンクを切断するメソッドを呼び出すコードを追加

Step9. ソースコードの追加，変換を行ったプログラムを出力

ローカルストレージでデータ管理をする Java プログラムにソースコードの追加，変更を行い，REST ベースのオンラインストレージでデータ管理をする Java プログラムへの変換を実現した．複数のファイルを操作する際は，スレッドにより複数の HTTP リクエストを送信する．

5. キャッシュ機能の組み込み方法

オンラインストレージでのデータ操作は通信を行うので，ローカルストレージでのデータ操作に比べて遅延が生じる． データの送受信に伴う遅延を軽減するためにローカルストレージをキャッシュとして利用する機能を実装した．キャッシュされたデータの更新，キャッシュのデータをオンラインストレージに同期する機能を以下に示す．

[データの更新] キャッシュされたデータの更新には，HTTP の If -Modified-Since リクエストヘッダフィールドを使用する．もしリクエストされたデータがこのフィールドにて指定された時刻以降に更新されていなければ，サーバはデータを返す代わりに，HTTP レスポンス 304（not modified）をメッセージボディ無しで返す[5]．オンラインストレージへファイル読み込みの HTTP リクエストを送信するにあたり，If -Modified-Since リクエストヘッダフィールドにキャッシュ内のファイルの最終更新時刻を指定し，更新があったとき，キャッシュ内のファイルをリロードする．

キャッシュ機能を用いることで，ファイルが存在していない場合，またはキャッシュ内のファイルの最終更新時刻以降に更新があった場合はメッセージボディ部にファイルが付加された HTTP レスポンスを受信し，キャッシュ内のファイルをリロード．ファイルの更新がなかった場合，HTTP レスポンス 304 を受信する．

更新がなかった場合，データの受信を行わないので，データ移行に伴い生じる遅延を軽減できる．

[データの同期] アプリケーション終了時，または一定時間ごとにキャッシュ内で更新があったファイルが存在するかを確認し，更新されたファイルがあった場合オンラインストレージにファイルを同期する．このようにすることでオンラインストレージへのデータ送信を行う頻度を減らすことができる．

6. キャッシュ機能の評価

キャッシュ機能を用いることで軽減されるファイル読み込み時の遅延をキャッシュ機能の有無による応答時間の比較を行い，評価する．応答時間の測定環境を表 1 に示す．ストレージサーバの実装に用いた JSR 311 は，Java プラットファームにおいて REST スタイルの Web アプリケーションを開発するための API である．読み込みを行うファイル数を 10 個，ファイルサイズを

表 1 測定環境

	クライアント	ストレージサーバ
OS	Windows 7 Professional	Windows 7 Professional
CPU	Intel(R) Core(TM) 2 Quad CPU Q9550 2.83GHz	Intel(R) Core(TM) i5 CPU 2.67GHz
メモリ	4.00GB	4.00GB
Java ランタイム	jre1.6.0_24	jre1.6.0_24
Java API		JSR 311

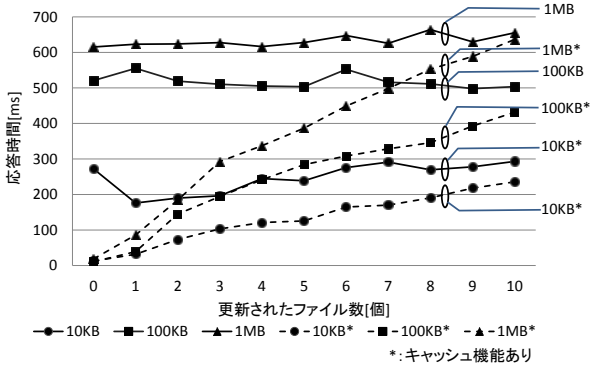

図 3 測定結果

10KB，100KB，1MB の 3 パターンとする．更新されたファイル数を変更し，100 回ずつ測定を行った．応答時間の測定結果は図 3 に示す．

キャッシュ機能を用いることで，更新されたファイル数が少ないときにはファイル読み込みの応答時間を減らすことができている．また，すべてのファイルが更新されていた場合でも応答時間の上限は，キャッシュ機能なしの場合を超えないことを確認した．すべてのファイルが更新されていない場合，20ms 以内でファイル読み込みを完了できた．

7. おわりに

本研究は，データを安全に管理し，紛失を未然に防ぐことだけでなく，サーバやクライアントのハードウェアが破壊・損傷などの被害を受けても元のシステム環境に速やかに復元する方法の確立を目的としている．本稿では，オンラインストレージという技術に注目し，ローカルストレージから REST ベースのオンラインストレージへのデータ移行を支援するため，ローカルストレージへの処理要求を HTTP リクエストへ変換する手法を提案した．

オンラインストレージでデータ管理をするにあたって遅延が生じることから，遅延を軽減するキャッシュ機能を実装した．ファイル読み込み時は，応答時間の増加を軽減し，ファイル書き込み時は，データ送信の頻度を減らすことが可能となった．

参考文献

[1] 瞬快，http://jp.fujitsu.com/group/shikoku/services/packages/shunkai/ （参照 2012-01-10）

[2] NetSHAKER for School，http://www.ysknet.co.jp/product/type/appliance/nsk_dots/index.html （参照 2012-01-10）

[3] Amazon Simple Storage Service（Amazon S3），http://aws.amazon.com/jp/s3/ （参照 2012-01-10）

[4] IIJ GIO ストレージサービス，http://www.iij.ad.jp/GIO/service/storage/（参照 2012-01-10）

[5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, ``Hypertext Transfer Protocol -- HTTP/1.1”, RFC2616, June 1999.