

MyCloud: 複数ベンダのクラウドを用いて構成する 高速で高信頼な分散ストレージ

堀内 公平

本研究では複数ベンダのクラウドを用いて構成する高速で高信頼な分散ストレージ MyCloud を提案する。従来のクラウドコンピューティングでは、単一のクラウドベンダに依存することによる、転送遅延問題、可用性問題、プライバシー問題が指摘されている。これらの問題を解決するため、本研究では次の 3 点を実現した。(1)高速性: 並列分散処理によりダウンロード速度を最大で 13 倍程度高速化した。(2)高可用性: データを冗長分散し、システムの一部であるクラウドサービスが一定数以下停止しても、継続して利用できるシステムを構成した。(3)強秘匿性: 暗号化と分散配置により、クラウドからデータが漏洩、あるいはベンダがユーザのデータを無断利用しようとしても、有為な情報を読み取れないシステムを実現した。

This paper is to propose "MyCloud", A High-Performance, Secure Distributed Storage System. In recent years, Cloud Computing plays an ever-increasing role in the whole world. Although the service is useful, there are such 4 problems about latency problem, non-availability problem and privacy problem in due to vendor lock-in. MyCloud constructs A High-Performance, Secure Storage System Using some vendors' Cloud Computing in parallel. MyCloud has 3 contributions. First, MyCloud can construct high availability storage. For example, failure rate is 10e-9% using 3 public clouds. Second, MyCloud strongly preserves privacy to encrypt and distributed to several vendors' cloud. Third, MyCloud translates 13 times faster than normal cloud storage service. In this paper, we talk about how to development of MyCloud and Experimental result for estimating Availability and Download-Speed of MyCloud. MyCloud make future that there is secure, available and useful Cloud Computing.

1. はじめに

1.1. 背景

近年、アプリケーション、システム、ハードウェアをネットワーク経由で利用するサービスの形態であるクラウドコンピューティングが注目されている。ユーザの立場からのクラウドコンピューティングのメリットとして、アクセス透過性、導入の容易さ、リソースが無制限である点などが挙げられる。一方、ユーザ視点からのクラウドコンピューティングの課題として、データ転送時の遅延問題や、クラウドコンピューティングサービス提供者（以降ベンダと記述）に対する信頼性の問題、プライバシー問題、ベンダが企業戦略の一環としてユーザを囲い込むロックイン問題などが指摘されている[1]。

MyCloud: A High-Performance, Secure Distributed Storage System Combining Several Vendors' Cloud Storage Services.

HORIUCHI Kohei, 電気通信大学 総合情報学専攻, Graduate School of Informatics, University of Electro-Communications.

1.2. 問題

クラウドにはそれぞれ以下のような問題がある。

- 転送遅延問題:
クラウドの特性として、ベンダ毎や時間帯によってデータ転送の速度差の変動が大きいということが言える。また、アジアの国々ではそれほど高いデータ転送速度が出ない傾向がある。この理由には諸説あるが、一つの見解として、クラウドのデータセンタが主として欧米に設置される傾向があるため、物理的な距離による転送遅延が大きな要因とみられている。
- 可用性問題:
クラウドストレージの稼働率は従来のオンラインストレージサービスや個人の管理するファイルサーバなどと比べても遜色の無い程度には高い。しかしユーザは自分の管理下に無いサービスが停止する可能性に対して敏感になる傾向がある。そのためクラウドがダウンすると、必要以上に大きなニュースとなり、ユーザの不安をさらに煽る。
- プライバシー問題:

いまや PC やモバイル端末と生活を完全に密着しており、それら情報端末の中には個人情報や機密情報がひしめいているのが常である。しかしモバイル端末の遺失届件数は都内だけで年間 189,000 件発生しており[2], 我々は常に重要な情報の盗難・紛失被害の危険にさらされている。重要なデータをローカルに保存せず、全てオンライン (クラウド) に保存しておくのは紛失・盗難に対して有効な対策であるが、誰にも見られたくないデータ・誰にも見せてはいけないデータをクラウドに預けることができるか、といった問題が生じる。

1.3. 目的

本稿では、これらの課題に対処する 1 つの方法として、複数ベンダのクラウドを用いて構成する高速で高信頼なストレージ MyCloud[3]を提案する。

本提案の目的は、(1) 一般のクラウドストレージサービスと比較してデータ転送を高速化すること (高速性)、(2) システムの一部であるクラウドサービスが一定数以下停止しても、継続して利用できるシステムを構成すること (高可用性)、(3) クラウドからデータが漏洩、あるいはベンダがクラウド内のデータの無断利用を試みても、意味ある情報を読み取れないこと (強秘匿性) とする。

2. 関連研究・関連技術

関連研究として、Abu-Libdeh の RACS: A Case for Cloud Storage Diversity[4]が挙げられる。RACS は複数のクラウドを用いて RAID システムを構築することによってベンダからのロックインを逃れることを目的とした提案である。本提案との差異性は、RACS が Amazon S3[5]のみを用いてシミュレーションを行っているのに対して、MyCloud は実際に複数ベンダのクラウドを用いて実験を行っている点にある。また、RACS の提案の主眼はベンダ乗り換え時のコストの低減にあるが、MyCloud はクラウドストレージサービスを複数並列に用いた場合の高速化に主眼の一つを置いている点異なる。

また、低信頼なストレージを用いて高信頼なストレージを構築するという研究は何点かある[6][7]。

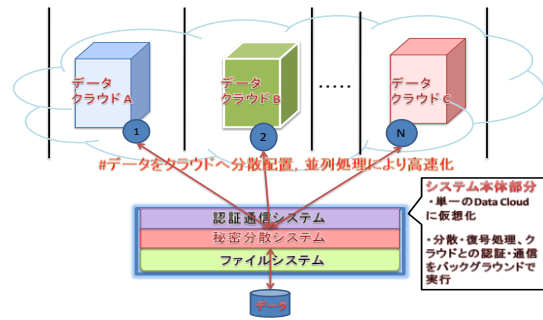


図 3.1 MyCloud の概要図

例えば井口氏による信頼性を考慮したグリッド向け自律分散ストレージシステム[8]では、ストレージノードの信頼性 (稼働率) を常に監視し、その信頼性に応じてリード・ソロモン符号[9]の冗長度を動的に操作する手法が提案されている。これらに対して、本提案では自前のストレージの代わりに複数ベンダのクラウドストレージを用いる点、コンピュータリソースの増減が非常に柔軟である点、NAT などの存在を全く意識することなく機能を利用できる点において異なる。また要素技術としては、ベンダに意味のあるデータを渡さないための分散手法や、クラウドとの通信を管理するための機構などが必要となる。

また関連技術として日本ユニシスグループの真性乱数を使用した秘密分散データストアサービス[10]が挙げられる。これは顧客データを秘密分散して複数のデータセンタに置くサービスであり、秘密分散によってクラウド上のデータの安全性を高めるという点で本提案と関連性がある。しかしこのサービスはあくまで単ベンダで完結しており、ユーザはベンダからのロックインを逃れることはできない。

3. MyCloud とは

3.1. 提案概要

本稿では、複数ベンダのクラウドを用いて構成する秘密分散ストレージ MyCloud を提案する (図 3.1)。MyCloud はクラウドの中でもクラウドストレージサービスを対象とした提案である。MyCloud のフロントエンドは、ユーザ (ファイル操作を伴

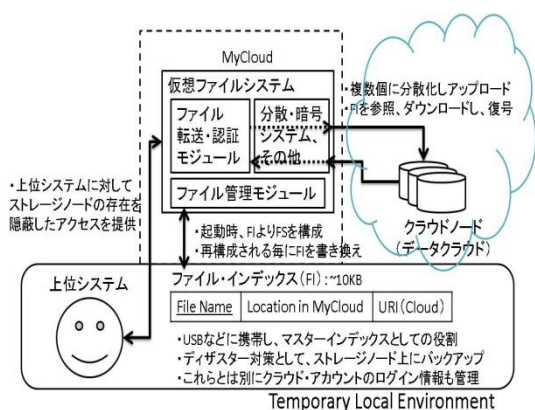


図 3.2 MyCloud の構成図

上位システム) に対して複数のクラウドストレージサービスの存在を隠蔽し、あたかも単一のストレージのように見える。一方バックエンドでは、ユーザのデータに対して分散化・復号化の処理を施し、複数のクラウドとデータを分散化したまま通信し、ユーザデータの管理を行う。複数のクラウドとの通信を並列分散することにより、データ転送の総計時間が短縮される。またクラウド上ではデータは分散化した状態でしか存在しないため、何れかのクラウドからデータが漏洩しても個人情報情報が漏れることがない。また分散手法については冗長性を持つ手法を用いることにより、幾つかのクラウドがサービスを停止しても他のクラウドからデータを収集し元のデータを復号することができる。つまり MyCloud を使うことのメリットは、ユーザが何れのベンダにも依存せずに済むと同時に、クラウドの通常利用時よりも速い転送速度と高可用性、高秘匿性を獲得できることにある。

3.2. 詳細設計

図 3.2 は MyCloud 全体の構成を表した図である。以下より各要素に説明を加える。

3.2.1. MyCloud 本体

今回、MyCloud の本体部分は、proxy サーバなどの Web サービスや web アプリケーションとしてではなく、ローカルで実行される単一のアプリケーションとして実装した。これは MyCloud に何らかのブラックボックス部分が残ってしまい、ベンダ

のログインから解放されたユーザが翻って MyCloud にログインされるのではないかとこの心配を払拭するため、ローカルで動くオープンソース・ソフトウェアとして実装することを目指したためである。MyCloud の本体部分は以下に示すモジュールで構成されている。

- **ファイル転送・認証モジュール：**
各ベンダのクラウドノードとの認証及び、分散データやクラウド上のファイルリストなどのデータのやり取りを実行する。
- **分散・暗号システム、その他：**
上位システムから渡されたデータを分散する。またクラウドノードから得たデータを復号化する。なお分散手法には、リード・ソロモン符号などの誤り訂正符号化方式や、データの機密性に応じて分散する個数 k と復元に必要な個数 n を容易に操作できる閾値秘密分散共有法[11]なども考えられるがこれらは提案の主眼ではなく計算量も非常に大きいため、今回は実装の容易さや計算量の軽さなどの理由から AES 暗号化と分割を組み合わせた方法を用いる。
- **ファイル管理モジュール：**
クラウド上からダウンロードしたファイルリストを併せたファイル・インデックス情報から MyCloud のファイル構造を再現する。
- **インタフェース：**
ファイル管理モジュールの再現したファイル構造から上位システム向けのインタフェースを提供する。インタフェースの提供形態は2種類用意した。ひとつはドラッグアンドドロップベースのクライアントソフトであり、インストールレスのスタンドアロンの軽量なアプリケーションとして実装されている。もう一方は単一のドライブとしてマウントして利用する仮想ファイルシステムであり、他のあらゆるアプリケーションやシステムソフトに対してファイルアクセスを提供する。また仮想ファイルシステムの開発には OS 上に仮想的なファイルシステムを構築するためのライブラリである Dokan[12]と Fuse[13]を用いた。

また今回実装したものは別にスマートフォン向けのアプリケーションも実装中である。

3.2.2. ファイル・インデックス

MyCloud の起動時、認証通信モジュールを通してアカウント情報を持つ全てのクラウド上にある全ファイルリストを取得する。MyCloud に関連するファイルには特定のための特殊な拡張子が付いている。それらの拡張子を元に MyCloud のファイル・インデックスを作成し、ファイル・インデックスからインタフェース上にファイル構造を再現する。

3.2.3. クラウドノード

分散したデータを預けるクラウドストレージ群である。対象としては、一般にサービス展開しているクラウドストレージサービスを想定しており、本稿では以下の 3 つのクラウドを利用した。

- Amazon S3
- SkyDrive[14]
- Nirvanix[15]

Amazon S3 については利用するデータセンタを選択することができる。今回は US Standard, California, Ireland, Singapore の 4 つの地域のデータセンタを冗長化のために使い分けた。

3.3. 機能説明

3.3.1. 基本機能

MyCloud は Put, Get, List, Delete の 4 つの基本機能を提供する。

Put は MyCloud へのデータのアップロード処理である。フロントエンドでは、ユーザは上位システムから MyCloud にアップロードしたいデータを渡し、データの機密性に応じて分散する冗長度 R を決定する。バックエンドの処理として、システムは受け取ったデータをローカル環境で冗長度 R に分散し、それぞれの分散データをネットワーク上のクラウドノードにアップロードする。

Get は MyCloud からのデータのダウンロード処理である。フロントエンドでは、ユーザは MyCloud

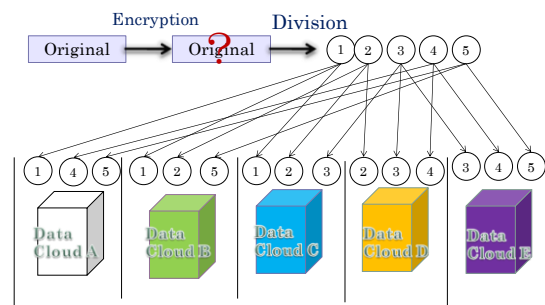


図 3.3 冗長配置の例

のインタフェース上でダウンロードしたいファイルのキーを指定する。バックエンドの処理として、2.3.3. の高速化アルゴリズムに沿って分散データをダウンロードし、ローカル環境でそれらを復号する。

List は MyCloud からのファイルリストのダウンロード処理である。アカウント情報を保有している全てのクラウドノードからファイルリストをダウンロードする。

Delete は MyCloud 上のデータのリムーブ処理である。フロントエンドでは、ユーザは MyCloud のインタフェースで削除したいファイルのキーを選択する。バックグラウンドでは、全てのクラウドノード上からそのキーに関連する全データを削除する。

3.3.2. ディザスタリカバリ

MyCloud のクライアントソフトがクラッシュした場合を想定する。クライアントソフトに保管されているユーザ情報はクラウドノードのアカウント情報だけである。ゆえにユーザは新たに MyCloud のクライアントソフトをダウンロードし、各クラウドのアカウント情報を入力し直すだけで容易に MyCloud のファイル構造を再現できる。

3.3.3. 冗長化手法

図 3.3 は MyCloud での冗長配置の例である。データを暗号化したのち、任意の大きさの適当な大きさへ分割し、各々のデータをクラウドへ分散配置する。この際、任意の離散的な冗長度の値を設定して、冗長配置することによって、クラウドの

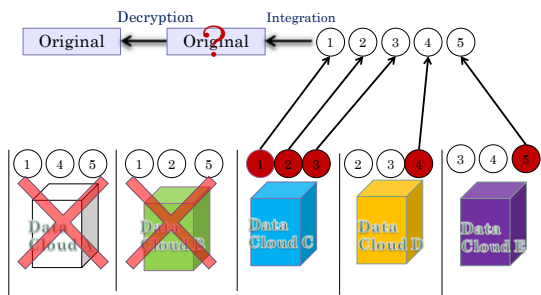


図 3.4 障害発生時の復号化の例

障害に対して耐性が生まれる。この例では冗長度 3 で分散配置を行っている。以上の条件のもと、図 3.4 のように 2 つのクラウドノードがダウンした場合、MyCloud は残り 3 つのクラウドノードから分散したデータを全て収集し、滞りなく復号化することができる。

MyCloud の大きな利点はデータの重要度に応じて冗長度を自由に設定できる点にある。冗長度はクラウドノード数を上限とした自然数の範囲内で決定できる。冗長度とコストは、MyCloud を無料のクラウドストレージサービスを用いて構成しない限り、トレードオフの関係になるが、絶対に消えては困る大事なデータには高い冗長度を設定し、重要度の低いデータには低い冗長度を設定するということが可能となる。

3.3.4. ダウンロード高速化手法

MyCloud では基本的にデータを冗長配置している。冗長配置のメリットとして、ダウンロードするクラウドノードをある程度選択できるということが挙げられる。クラウドストレージは、ベンダや時間帯によってデータ転送の速度差が大きく変化する為、どこからデータをダウンロードするかというのはスループットに大きく影響する。

MyCloud では Get 処理が発生する度に、最適なダウンロードの組み合わせを動的に探索する (図 3.5)。分散データのダウンロードのタスクをキュー構造で保持し、各クラウドノードに振り分ける。ダウンロードが早く終わったノードに対して新たにタスクを振り分ける。

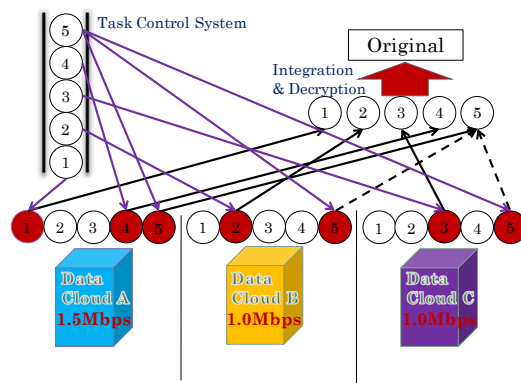


図 3.5 ダウンロード高速化手法

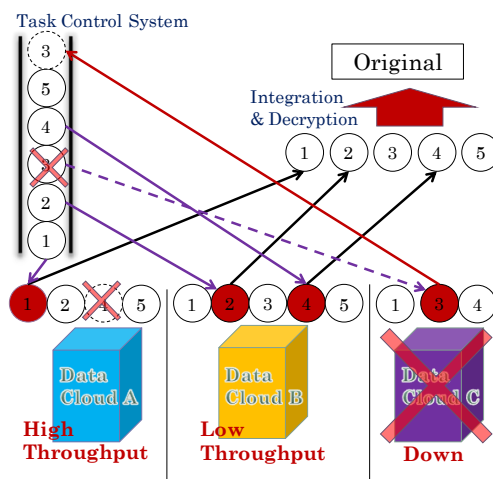


図 3.6 障害発生時のダウンロード

そして、タスクの残りが少なくなった場合は複数のクラウドノードから同一の断片ファイルを同時にダウンロードし、ダウンロードが最も早く終わったファイルを用いて復号処理を開始する。これは転送速度の遅いクラウドノードに全体を引っ張られないための工夫である。また、クラウドノードがダウン、或いはいつまでたっても断片ファイルのダウンロードが終わらないなどの障害が発生する可能性がある (図 3.6)。そのような場合、MyCloud のダウンロードタスクを一度タスク管理用のキューに回収し、新たに振り分け直す。こうすることで障害発生時もダウンロードのタスクが均等に振り分けられる。

表 4.1 クラウドノード数と稼働率の関係

Cloud Nodes	Availability [%]	Failure Time [/year]
1 node (Public Cloud)	99.9	526 min
3 nodes	99.9999999	31*10e-3 sec
5 nodes	100 - (10e -11)	31*10e-9 sec

4. 実験と評価

実装した MyCloud の可用性と高速性について、実験し評価を行った。

4.1. 可用性評価実験

4.1.1. 可用性評価実験の概要

MyCloud のクラウドノード数と稼働率の関係性について、クラウドストレージサービスを単体で使ったケースとの比較により評価を行った。

4.1.2. 可用性評価実験の環境構成

MyCloud で用いているクラウドノードのうちで稼働率が公表されているのは Amazon S3 のみである。近年世界で最も使われているクラウドストレージは Amazon S3 である。よって今回クラウドストレージ 1 台あたりの稼働率は 99.9% とし、冗長度はクラウドノード数と同値として計算した。

また MyCloud 自体がクラッシュする可能性については稼働率の試算の際に考慮しなかった。非常時には MyCloud のクライアントソフトを別の PC 上で動作させアカウント情報を再度入力することによって容易に環境を再現できるためである。

4.1.3. 可用性評価実験の結果

表 4.1 はクラウドノード数と稼働率の関係を示した表である。クラウドストレージを単体で使うと稼働率は 99.9%、停止時間の目安が 1 年間におよそ 526 分という値だが、5 つのクラウドノードを用いて MyCloud を構築した場合、稼働率は 100 - (10e -11) % 停止時間の目安が 1 年間におよそ 31 ナノ秒という値まで改善される。

データのアップロード時、クラウドノード数を

表 4.2 利用クラウドノード一覧

Cloud Node	Kinds of Cloud
1 node	• SkyDrive
3 nodes	• SkyDrive, • Amazon S3 - US Standard - Ireland
5 nodes	• SkyDrive • Amazon S3 - US Standard (US) - California - Ireland (EU) - Singapore

上限とした自然数の中から冗長度を自由に選択できるため、データの重要度に応じて可用性をコントロールすることができる。

4.2. 高速性評価実験

4.2.1. 高速性評価実験の概要

クラウドノード数を変えながら MyCloud の転送速度を計測し、いくつかのクラウドストレージサービスの転送速度と比較した。

4.2.2. 高速性評価実験の環境構成

比較対象のクラウドストレージサービスとして SkyDrive, Amazon S3 を用いた。Amazon S3 については US Standard (US) と Ireland (EU) の両データセンタをそれぞれ計測した。MyCloud については、クラウドノード数を 1 node, 3 nodes, 5 nodes と変動させて計測した。それぞれで使用したクラウドノードの種類は表 4.2 の通りである。冗長度については、1 node 時は冗長度 1, 3 nodes 時は冗長度 2, 5 nodes 時は冗長度 3 と設定した。

転送対象のデータとして 10MB のファイルを用いた。それぞれの転送先に対して 6 時間おきに 5 回ずつ計 20 回のアップロード処理とダウンロード処理を試行し、平均値を算出した。

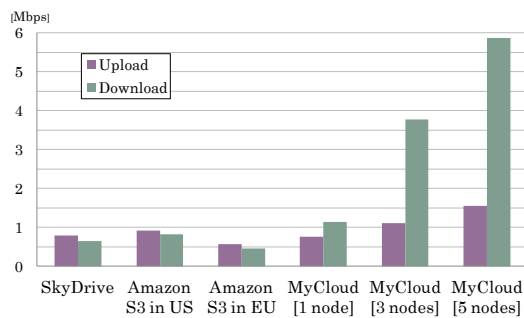


図 4.1 速度比較結果

4.2.3. 高速性評価実験の結果

図 4.2 は速度比較実験の結果である。縦軸は転送速度である。各項目の左側の棒グラフがアップロード速度、右側の棒グラフがダウンロード速度を表している。

ダウンロード速度について、SkyDrive は 0.649Mbps, Amazon S3 in US は 0.823Mbps, Amazon S3 in EU は 0.453Mbps であった。それらに対して、1 node 時の MyCloud のダウンロード速度は 1.14Mbps であり SkyDrive の約 1.76 倍となった。3 nodes 時の MyCloud のダウンロード速度は 3.78Mbps であり、SkyDrive の 5.82 倍、Amazon S3 in US の 4.59 倍、Amazon S3 in EU の 8.34 倍、3 クラウドノードの平均ダウンロード速度と比較すると 5.89 倍となった。5 nodes 時の MyCloud のダウンロード速度は 5.87Mbps であり SkyDrive の 9.04 倍、Amazon S3 in US の 7.13 倍、Amazon S3 in EU の 13.0 倍となった。

5. 考察

5.1. 可用性評価実験の考察

クラウドストレージサービスの稼働率は個人所有や社内でも利用されていたファイルサーバなどと比べても遜色の無い程度には高い。しかしユーザは自分の管理下に無いサービスが停止する可能性に対して敏感になる傾向がある。そのため MyCloud ではユーザのニーズに応じて自由に可用性をコントロールできる仕組みを導入した。当然のことだが、クラウドノード数を増やせば増やすほど MyCloud 上で選択できる冗長度の幅は広がる。

このときクラウドストレージサービスの料金形態が従量課金制であれば、クラウドノード数を増やしたとしてもコストはノード数に比例して増えない。また、SkyDrive のような無料のサービスを優先的に使うことによってコストを下げるができる。アカウント管理の煩雑さは問題となる可能性があるが、それ以外の点においては従来のクラウドストレージサービスのデメリットを消しつつメリットを包括していると言える。

5.2. 高速性評価実験の考察

クラウドストレージサービスを複数使うことにより最大で 13 倍程度のダウンロード高速化を実現した。これは元来ボトルネックとなっているクラウド側のネットワークの負荷を複数のクラウドに分散させたことにより、帯域を有効活用できたためである。クラウドノードを増やしていった場合、ユーザ周囲の細かい帯域部分が初めのボトルネックとなると思われるが、5 クラウドノードではまだその兆候は見られなかった。データ転送の最初のボトルネックの検証とその改善が今後の課題である。

1 ノードで構築した MyCloud と SkyDrive でダウンロード速度に差があるという結果が生じた。MyCloud はクラウドノードが 1 つの場合でもデータを分割する。SkyDrive が 1 アカウントあたりに 2 つのセッションで並列にデータ転送することを許可していたためにこのような結果となった。今回用いなかったクラウドストレージにも SkyDrive と同じように公開されていない特性がある可能性があるが、それらの特性に合わせて自動で最適化することによって更なる高速化が見込める。

アップロードに関しては冗長化を行っているため、転送速度はそれほど向上しなかった。しかし、ダウンロード処理にはリアルタイム性が求められるが、アップロードに関してはバックグラウンドで実行することで使用上問題は無いと考えたため、ダウンロードの高速化のみに注力した。

6. まとめと今後の予定

本論文では複数ベンダのクラウドストレージを用いて構成する高速で高信頼な分散ストレージの

詳細と、その評価について述べた。本提案の新規性は、複数ベンダのクラウドストレージを用いて分散ストレージを構築した点と、世界で広く使われているクラウドストレージを組み合わせてどれだけの転送速度が出るかを明らかにした点にある。

可用性について、MyCloud 上で多くのベンダを利用するほど稼働率の選択の幅が広がることは自明である。今後はますます MyCloud で利用できるクラウドストレージサービスが増えていくと予想される。高速性に関しては、複数ベンダのクラウドの帯域を有効に活用することによって、ダウンロード速度が最大で 13 倍程度向上した。冗長配置を行わずに複数のベンダに分散した場合、コストは殆ど増加しない。そのため、本提案の高速化手法の需要はクラウドストレージの充実とともに今後さらに拡大していくと思われる。秘匿性については、AES 暗号化と分散配置によって、単一のベンダから現実時間内に全情報が漏洩する危険性は殆どなくなった。

今後の予定としては、対応するクラウドストレージサービスを増やしていくこと、より柔軟な冗長化方式による実装を行うこと、アップロード時の一貫性保持のための機能を組み込むこと、ユーザビリティ向上のためにユーザインタフェース・キャッシュシステムの実装を進めていくこと、また MyCloud をデスクトップクライアントソフト及びスマートフォン向けアプリとして製品化する予定なので、コードの公開部分を整理してオープンソース化すること、などが挙げられる。

謝辞

本提案の一部は独立行政法人情報処理推進機構により、未踏 IT 人材発掘・育成事業 2009 年度上期 未踏ユースに採択され、この支援を受けて開発を行いました。

参考文献

- [1] Armbrust, M., Fox, A., Griffith, Rean., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, Gunho., Patterson, D.

A., Rabkin, A., Stoica, Ion. and Zaharia, Matei. : Above the Cloud: A Berkley View of Cloud Computing,” Technical Report, no.UCB/EECS-2009-28, 2009.

[2] 警視庁: 平成 21 年中 遺失物取扱状況,
http://www.keishicho.metro.tokyo.jp/toukei/kaikei/kaikai_21.htm

[3] 堀内公平: 複数ベンダのクラウドを用いた秘密分散ストレージ「MyCloud」の開発, 2009 年度上期未踏 IT 人材発掘・育成事業 採択案件, 2009.

[4] Abu-Libdeh, H., Princehouse, L. and Weatherpoon, H.: RACS: A Case for Cloud Storage Diversity, Proc. SoCC' 10, ACM, 2010, pp. 229-240.

[5] Amazon S3, <http://aws.amazon.com/s3/>

[6] Chang, F., Dean, D., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., Gruber, R. E.: Big Table: A Distributed Storage System for Structured Data, ACM, TOCS, Vol.26, Issue.2, 2008.

[7] 市川俊一, 豊田真智子, 高橋克巳: 高信頼 PC クラスタ・ストレージ実現のための連鎖ネットワーク RAID の設計と実装, IEICE, DEWS2007, 2007.

[8] 井口寧, 渡辺浩二, 松澤照男: 信頼性を考慮したグリッド向け自律分散ストレージシステム, 情報処理学会論文誌, コンピューティングシステム, Vol.47, No.SIG 7, pp.219-230, 2006.

[9] リード・ソロモン符号,
<http://www.siglead.com/technology.html>

[10] 真性乱数を使用した秘密分散データストアサービス,
http://www.unisys.co.jp/news/nr_091007_cloud.html

[11] 尾形わかは, 黒沢馨: 秘密分散共有法とその応用, 電子情報通信学会誌, Vol.82, No.12, pp.1228-1236, 1999.

[12] Dokan, <http://dokan-dev.net/>

[13] Fuse, <http://fuse.sourceforge.net/>

[14] SkyDrive, <http://skydrive.live.com/>

[15] Nirvanix, <http://www.nirvanix.com/>