情報処理学会論文誌 Vol.55 No.3 1140–1150 (Mar. 2014)

広域分散仮想化環境のための分散ストレージシステムの提案と評価

柏崎 礼生[1,a)] 北口 善明[2,b)] 近堂 徹[3,c)] 楠田 友彦[4,d)] 大沼 善朗[4,e)]
中川 郁夫[1,4,f)] 阿部 俊二[5,g)] 横山 重俊[5,h)] 下條 真司[1,i)]

受付日 2013年6月30日, 採録日 2013年12月4日

概要：大規模災害による危機意識の高まりから災害回復（Disaster Recover: DR）を実現するための技術として遠隔地データセンタでのバックアップや分散ストレージに注目が集まっている．現在我々は金沢大学，広島大学，国立情報学研究所を中心として広域分散型のストレージ環境の構築を進めている．この環境は，ランダムアクセス性能の高さに特徴があり，かつ各拠点から同じデータにアクセスできるため，ライブマイグレーションにより他拠点においてサービスを継続することが可能となる．本稿では本環境における I/O 性能を評価し，広域分散ストレージを用いたライブマイグレーション実験について述べ，有用性を示す．

キーワード：仮想化基盤，クラウドコンピューティング，障害回復（DR），広域分散，広域ライブマイグレーション

A Proposal and Evaluations of a Distributed Storage System for a Widely Distributed Virtualization Infrastructure

Hiroki Kashiwazaki[1,a)] Yoshiaki Kitaguchi[2,b)] Tohru Kondo[3,c)]
Tomohiko Kusuda[4,d)] Yoshirou Onuma[4,e)] Ikuo Nakagawa[1,4,f)]
Shunji Abe[5,g)] Shigetoshi Yokoyama[5,h)] Shinji Shimojo[1,i)]

Received: June 30, 2013, Accepted: December 4, 2013

Abstract: This paper focuses on distributed storage technologies and backing up data onto remote data centers because of growing sense of disaster recovery for large scale disaster. Now the authors are constructing widely distributed storage environment among Kanazawa University, Hiroshima University, National Institute of Informatics (NII). This environment is characterized by high performance of random I/O access. Because every site can access to same data, IT services can be continued with live migration technology. In this paper, the authors evaluate performances of storage I/O and wide area live migration on this storage environment.

Keywords: virtualization infrastracture, cloud computing, disaster recovery, widely distributed, wide area live migration

1. はじめに

仮想化技術の成熟とともに組織内の情報システムを稼働させる物理マシンを仮想化環境へと移行し，さらにはパブリッククラウド事業者が提供する IaaS へと移行しよう

[1] 大阪大学サイバーメディアセンター
Cybermedia Center, Osaka University, Ibaraki, Osaka 567–0047, Japan
[2] 金沢大学総合メディア基盤センター
Information Media Center, Kanazawa University, Kanazawa, Ishikawa 920–1192, Japan
[3] 広島大学情報メディア教育研究センター
Information Media Center, Hiroshima University, Higashi-hiroshima, Hiroshima 739–8511, Japan
[4] 株式会社インテック
Intec Inc., Takaoka, Toyama 933–8777, Japan
[5] 国立情報学研究所
National Institute of Informatics, Chiyoda, Tokyo 101–8340, Japan
[a)] reo@cmc.osaka-u.ac.jp
[b)] kitaguchi@imc.kanazawa-u.ac.jp
[c)] tkondo@hiroshima-u.ac.jp
[d)] kusuda_tomohiko@cloud.intec.co.jp
[e)] onuma_yoshiro@cloud.intec.co.jp
[f)] ikuo@inetcore.com
[g)] abe@nii.ac.jp
[h)] yoko@nii.ac.jp
[i)] shimojo@cmc.osaka-u.ac.jp

としている [1]．組織外部のクラウドサービスを使うだけでなく国内の教育・研究機関の情報センタや研究科でパブリッククラウド，あるいはプライベートクラウドが構築されている．静岡大学はクラウドコンピューティングを全面採用した情報基盤システムを構築した [2]．北陸先端科学技術大学院大学（JAIST）では仮想デスクトップサービスを提供するためにプライベートクラウドを構築している [3], [4]．佐賀大学は専用線で接続された外注先にプライベートクラウドを構築し，メールサービスを提供している [5]．一方で，東京工業大学の TSUBAME2 に代表されるクラウド型（スケールアウト型）HPCI や北海道大学アカデミッククラウド [6] など計算能力の高さに重点をおいたパブリックサービスも提供されている．そもそもクラウドコンピューティングという言葉は，2006 年に開催された Search Engine Strategies Conference で Google の CEO（当時）だった Eric Emerson Schmidt 氏が Danny Sullivan 氏との対談で使ったのが初めてとされており[*1]，クラウドコンピュータの定義は Gartner，UC Berkeley，そして NIST による定義が引用されることが多いが [7], [8], [9]，本稿では「仮想化技術を用いて実現されるスケールアウト可能な基盤の上に構築された，規模を収縮可能なサービス」の意味で用いることとする．

日本では 2011 年 3 月 11 日に発生した東日本大震災以来，自然災害による機器の損壊，回線の切断などを要因とするサービスの中断に対応することが切実な問題として表面化したことにより，災害回復（Disaster Recovery: DR）や事業継続計画（Business Continuity Plan: BCP）を実現する手法が求められている．この手法として遠隔地データセンタの利用と一部システムあるいは基幹システムすべての移行というアプローチを京都教育大学や京都大学が採用している[*2,*3]．組織の本拠点とデータセンタが同時に 1 つの自然災害により損壊する確率は低いが，本拠点もデータセンタも人的災害や各種要因によりサービスの中断が発生することがあるため，他一拠点にデータの複製やバックアップを確保することは十分な対策とはいえない．その一方で複数拠点のデータセンタを利用することはコストの面で困難が生じる．

データセンタを利用した DR において，組織の本拠点と同じ構成のシステムをデータセンタ側でも稼働させホットスタンバイ方式で稼働させる場合，本拠点からデータセンタまでの遅延による影響を受けるためストレージのパフォーマンスが距離に応じて低下する．プライベートクラウドの構築にあたって性能向上のボトルネックとなるのは CPU やメモリ資源ではなくストレージであることが指摘されており [10]，この方式による DR の実現には費用対効果の困難さがある．仮想化基盤においては，仮想マシン（VirtualMachine: VM）で稼働する OS やサービスを停止させることなく他のハイパーバイザサーバ上で稼働させるライブマイグレーションが利用される．ライブマイグレーションを利用するためには複数のハイパーバイザサーバが共有するストレージが必要となるが，広域環境で共有ストレージを利用すると前述のホットスタンバイ方式での問題同様，遅延の影響を受けストレージへの I/O パフォーマンスが劣化する．一方，共有ストレージを利用せずに VM イメージを拠点間で移動させるストレージマイグレーションも利用されているが，共有ストレージを利用したライブマイグレーションに比べてサービス断時間が長くなるという問題を解決しなければならない [11]．

広域分散型のストレージとして Gfarm [12], 分散ファイルシステムとしては Google の GFS [13]，および HDFS[*4]が広く利用されている．Gfarm ではデータの保存はファイル単位であり，ファイルの任意の位置の修正においてもファイル全体へのアクセスが必要となってしまう．一方，GFS（HDFS）はファイルをブロック分割して保存するものの，Write-Once-Read-Many（書き込みは 1 度で読み出しを何度も行う）モデルに基づいたデータアクセスを前提とした設計であるため，POSIX の要件を緩和しており，ファイルの任意の位置の修正や複数の単一ファイルへの同時書き込みはできない．以上のことより，シーケンシャルアクセスに対しては十分な性能を発揮する一方で，ファイルの部分的な更新といったランダムアクセス性能については十分な性能を提供することが困難である．現在広く用いられている複数の仮想化ハイパーバイザの実装は POSIX 準拠のファイルシステムに対応している．また，仮想化ハイパーバイザは VM のイメージファイルに対してランダムアクセスする．これまでの POSIX 準拠の広域分散型のストレージはランダムアクセス性能がローカルストレージに比べて低いため，仮想化基盤のためのストレージとして利用することが困難である．そのため仮想化基盤のためのストレージは POSIX 準拠であり，かつ広域分散型であってもローカルストレージと同程度のランダムアクセス性能を示す必要がある．そこで本研究では，スケールアウト型の分散ストレージを地理的に広域に分散した複数拠点に配備し，広域分散型の仮想化基盤を実現するための広域分散ストレージ構築手法について提案する．本稿では国内 3 拠点で広域分散ストレージ環境を構築し，その I/O 性能を評価するとともに，拠点間ライブマイグレーションの評価実験を通して，本提案手法が広域分散仮想化基盤の実現に有効であることを示す．

*1 http://www.google.com/press/podium/ses2006.html
*2 http://pr.fujitsu.com/jp/news/2011/06/28.html
*3 http://pr.fujitsu.com/jp/news/2013/01/10.html
*4 http://hadoop.apache.org/docs/hdfs/current/hdfs_user_guide.html

2. ストレージアーキテクチャ

2.1 EXAGE/Storage のアーキテクチャ

本研究では，分散ストレージ技術として株式会社インテックが開発した EXAGE/Storage を用いる．同ストレージは，ユーザデータであるファイルコンテンツを複数のブロックと呼ばれる小さな単位に分割し，スケールアウト型のオブジェクトストレージ上に保存する [14], [15]．部分的な更新が可能なファイル構造を有しており，ランダムアクセスファイルにも対応している．また，POSIX 準拠の標準的なファイルシステムインタフェースを持ち，NFS や CIFS などの一般的なプロトコルでファイルやディレクトリにアクセスできる．同ストレージでは，ディレクトリやファイル属性などのメタデータを Jobcast と呼ばれる並列分散処理フレームワーク上に実装している [16]．Jobcast はクラウドコンピューティングスタイルの KVS（Key Value Store）データベースであり [17], [18]，Key と Value を分散させるだけでなく，Job と呼ばれる処理ロジックも分散させることにより，メタデータの参照・更新においても，ボトルネックが存在せず，高いスケーラビリティを実現している．

EXAGE/Storage を構成する機器，および基本的な処理の流れを**図 1** に示す．同ストレージはフロントエンドであるアクセスサーバとバックエンドであるコアサーバから構成される．アクセスサーバ，コアサーバとも多数のサーバが接続されることを前提としており，台数に応じてストレージ容量や性能が向上する，スケールアウト型のアーキテクチャを採用している．アクセスサーバはクライアントに対するインタフェースプロトコルを提供する．インタフェースプロトコルとしては NFS や CIFS などが利用可能である．クライアントはアクセスサーバに対してファイルの読み出しや書き込みなどの処理要求を送信する．ファイルに対する処理要求を受け取ったアクセスサーバは，その要求をブロック単位に分割し，コアサーバに処理要求を送信する．アクセスサーバは処理要求をブロック単位に分割し，多数のコアサーバに対して並列分散処理モデルで処理することで，大きなファイルに対する読み書きのトータルスループットを大幅に向上させることができる．

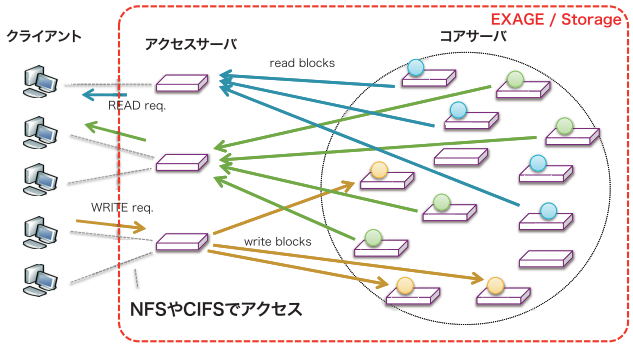

図 1 EXAGE/Storage 概念図
Fig. 1 EXAGE/Storage conceptual diagram.

2.2 広域分散環境への対応

本研究では，コアサーバを広域分散環境で接続可能にすることを提案する．地理的に離れた N カ所の拠点（サイト，デンターセンタ）にコアサーバを多数設置し，拠点間を高速なネットワークで接続する．これらの分散環境に設置されたコアサーバすべてを用いて EXAGE/Storage のバックエンドを構成する．ある拠点でアクセスサーバを設置し EXAGE/Storage を利用することにより，より大規模なストレージ空間を実現しつつ，地理的に離れた場所でデータの冗長化をとることにより，障害・災害に強いストレージ基盤を実現することが可能になる．

EXAGE/Storage を広域分散環境で構築する場合，拠点間の通信遅延がストレージ性能に大きく影響を与えることが懸念される．オリジナルの EXAGE/Storage は単一のデータセンタ内での利用を想定して設計されている．既存の利用例では，クライアントとアクセスサーバは 10 Gbps で接続され，アクセスサーバは 10 Gbps で多数のコアサーバと通信する．単一データセンタ内での利用では，アクセスサーバとコアサーバの通信遅延は 1 msec 以下である．一方で，コアサーバが地理的に離れた拠点に分散されている場合，特に大きなファイルを読み書きする際のスループットで性能劣化が起こりうる．たとえば，ファイルの書き込み時には以下のような課題がある．

従来の EXAGE/Storage では，アクセスサーバがコアサーバにブロックの書き込み要求した場合，コアサーバにブロックを作成し，かつコアサーバ間で必要な多重度（複製の数）が満たされるまで，コアサーバ間で冗長化した後に，アクセスサーバが処理を完了する（クライアントに ACK を返す）．このようなアルゴリズムでは，拠点間の通信遅延が大きくなるに従い，クライアントが ACK を受け取るまでの待ち時間が大幅に増加することになる．

本研究では，拠点（サイト，データセンタ）を個々に区別する．アクセスサーバはブロックを作成する際に，同一拠点内のコアサーバにブロックの作成を要求するが，その際に以下のようなアルゴリズムを用いることによって，地理的に離れた拠点間で冗長化しつつ，クライアントからみたスループットを劣化させない仕組みを実現した．各拠点はネットワークアドレスとサブネットによって区別される（**図 2**）．なお，以下では多重度を n（$n \leq N$）とする．

(1) 新たに作成するブロックのためのユニークな ID を割り振る．ID は拠点情報を含む．アクセスサーバが接続する拠点の ID として該当 ID を割り振るものとする．
(2) 自拠点内の物理的に異なる 2 台のコアサーバに該当ブロックのコピーを作る．
(3) 上記 (2) が完了した時点でアクセスサーバがクライア

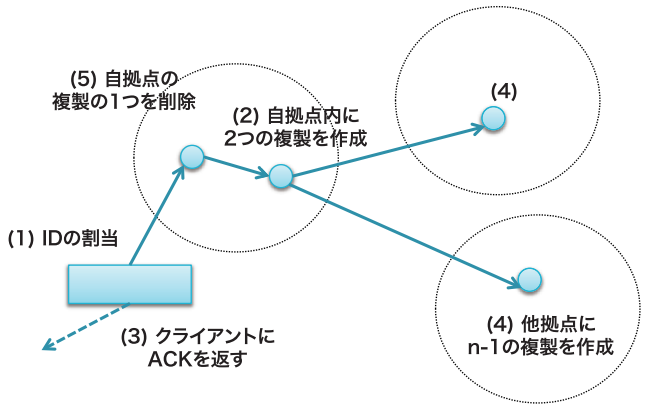

図 2 EXAGE/Storage における分散型の複製管理
Fig. 2 Distributed replication management on EXAGE/Storage.

ントに対して ACK を返す．

(4) 他の $n-1$ 拠点のコアサーバ上に該当のブロックのコピーを作成する．

(5) 上記 (4) が完了した時点で，自拠点にある 2 つのコピーのうちの一方を削除する．

上記のアルゴリズムでは，クライアントは自拠点内の処理が完了した時点で ACK を受け取るため，拠点間の通信遅延はスループットに直接的な影響を与えることはない．また，クライアントは少なくとも物理的に異なるコアサーバ上で冗長化してから ACK を受け取るため，単一サーバの障害でデータを失うことはない．厳密には，拠点そのものの障害があった場合にはデータにアクセスできなくなる可能性があるが，本手法は，スループットとリスクのトレードオフを考慮し，ある程度の信頼性を確保しつつスループットを実現するための仕組みであると考えられる．

上記の仕組みはコアサーバ（およびコアサーバ上で動作するブロックマネージャ）上のアルゴリズムの変更のみによって実現される．本提案では，広域分散環境対応については，アクセスサーバやクライアントに特別な仕組みを必要としない．なお，上記の処理は該当ブロックの複製を管理するブロックマネージャによって制御される．ブロックマネージャはブロックの多重度（コピーの数）に応じてコアサーバに複製を指示するほか，障害時には複製先の切替えや複製のやり直しを指示する．ブロックマネージャはコアサーバ上で動作する機能モジュールの 1 つで．他の機能同様，冗長化やスケールアウトの仕組みを有する．

3. 評価実験

3.1 広域分散ストレージ環境の構成

現在構築を進めている広域分散ストレージ環境の構成図を**図 3** に示す．原稿執筆時点では，広島大学，金沢大学，国立情報学研究所（以下，NII）の 3 拠点の接続が完了している．拠点間は NII が提供する学術情報ネットワーク SINET4 を利用して 10 Gbps で接続し，用途に応じた 3 つの VPN サービス（L2VPN サービス × 2，L3VPN サービス × 1）を利用している．以下に，それぞれについて説明する．

EXAGE-LAN（L3VPN）は，分散ストレージ内部の分散処理用セグメントである．このセグメントは各拠点がそれぞれ独立した L3 ネットワークで構成され，各 L3 ネットワークが SINET4 の L3VPN サービスで相互接続されている．これは前章でも述べたとおり，分散ストレージのアーキテクチャ上，ブロックの配置アルゴリズムがネットワーク単位で決まるためである．

管理 LAN（L2VPN）と MIGRATION-LAN（L2VPN）は，本ストレージをデータストアとする仮想計算機モニタ（VMM）のためのセグメントである．管理 LAN は仮想計算機モニタの管理用セグメントとなり，MIGRATION-LAN は仮想計算機モニタ上で動作する仮想マシン（VM）が接続するセグメントである．このセグメントに接続される VM は，本分散ストレージを OS イメージのデータストアとして利用する．各拠点には，拠点内のコアサーバ（CS），アクセスサーバ（AS），および仮想計算機モニタの死活監視と統計情報を収集するヒントサーバ（HS）を設置する．

広島大学を例に拠点内ネットワーク構成を説明する．**図 4** は，SINET アクセスポイント配下の広島大学拠点の構成を示したものである．各拠点ではアクセスサーバが広域分散ストレージのインタフェースとなる．利用するクライアントは，アクセスサーバに対して NFS マウントすることで POSIX 準拠のファイルシステムとして参照することができる．アクセスサーバは 10 Gbps および 1 Gbps × 4 のリンクアグリゲーション，コアサーバは 1 Gbps × 3 のリンクアグリゲーションにより集約スイッチに接続し，ヒントサーバは仮想マシンで用意している．また，アクセスサーバを NFS マウントする VMM は 1 Gbps × 2 のリンクアグリゲーションで集約スイッチと接続する構成としている．各拠点の機器構成を**表 1** に示す．

3.2 ストレージ性能評価

EXAGE/Storage の I/O 性能を評価するために iozone [*5] を用いて計測した（**図 5**）．広島大学の拠点に設置した VMM は Intel Xeon（E5-2640）を 2 基，64 GB のメモリを搭載し，CentOS 6.3 がインストールされている．この VMM 上で iozone を実行し，従来方式と広域分散対応方式の両方で性能を評価した．EXAGE/Storage のインタフェースプロトコルは NFS とし，close コールを含めた時間を計測する．検証環境の NFS クライアントの実装はキャッシュを保持している．NFS の write 時はキャッシュに対して行われ，fsync によりキャッシュが書き出される．また read 時はキャッシュ上のファイルと NFS サーバ上に

[*5] http://www.iozone.org

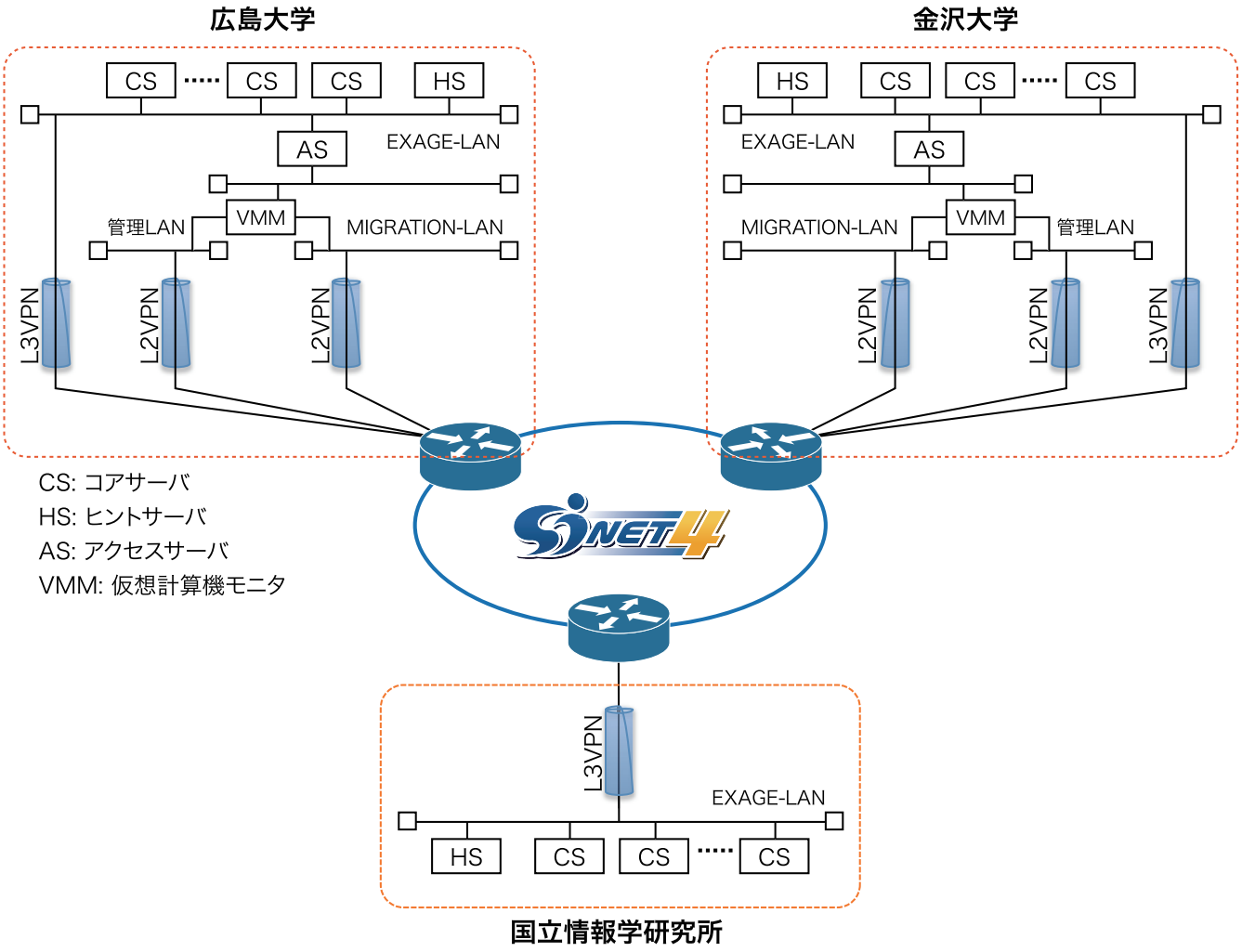

図 3 拠点間構成図

Fig. 3 Participating institutions diagram.

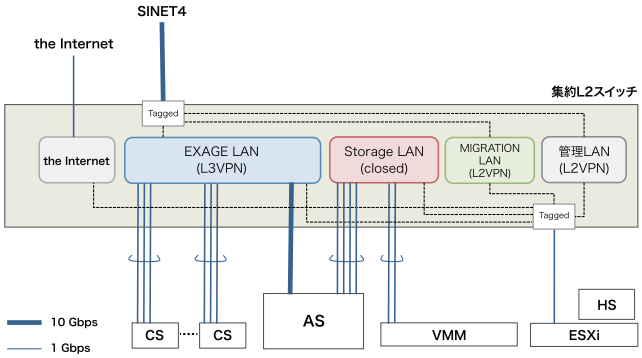

図 4 広島大学のネットワーク構成図

Fig. 4 Network diagram of Hiroshima university.

表 1 各拠点の機器構成

Table 1 Equipment configuration on each facility.

拠点名	サーバの種類	台数
広島大学	アクセスサーバ	1 台
	ヒントサーバ	1 台
	コアサーバ	4 台
金沢大学	アクセスサーバ	1 台
	ヒントサーバ	1 台
	コアサーバ	8 台
NII	ヒントサーバ	1 台
	コアサーバ	4 台

あるファイルの mtime およびファイルサイズを比較し，同一である場合にはキャッシュ上にあるデータを返す．そのため flush（fsync コール）に要する時間を含めた処理時間を計測することで，キャッシュによる性能への影響を排除し，ストレージの性能を直接的に評価する．また Direct IO を利用し，open システムコールがカーネル空間のページキャッシュを利用しないように指定する．アクセスパターンは write，rewrite，read，reread，random read，random write，bkwd read，record rewrite，stride read，fwrite および fread を指定する．ブロックサイズは 4 MB とし，4 MB から 32 GB までのファイルサイズでスループットを計測する．

すべてのアクセスパターンとファイルサイズの組合せに対するスループットのヒストグラムを示す（**図 6**）．従来方式では 30〜40 MB/sec にピークが存在し，平均スループッ

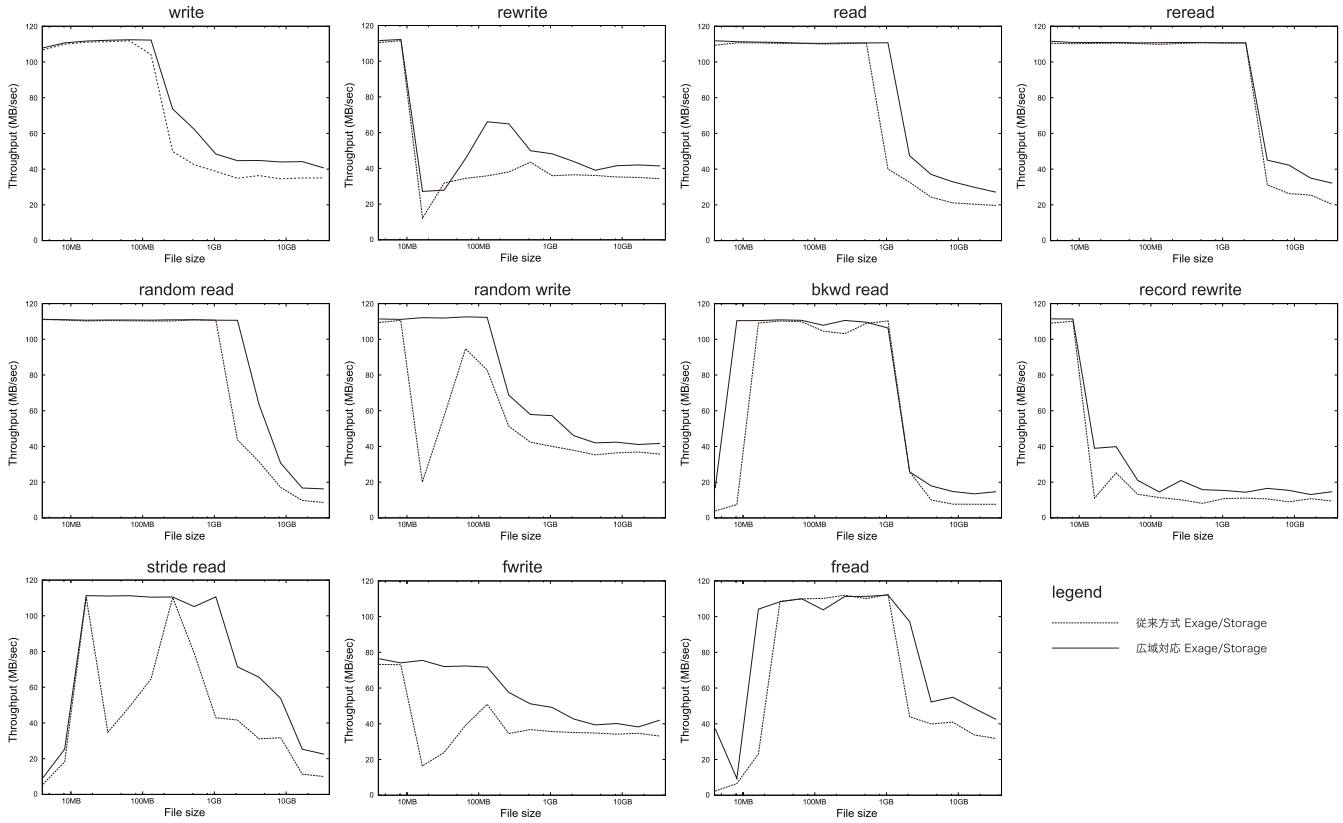

図 5 EXAGE/Storage の Read/Write パフォーマンス
Fig. 5 EXAGE/Storage Read/Write performance.

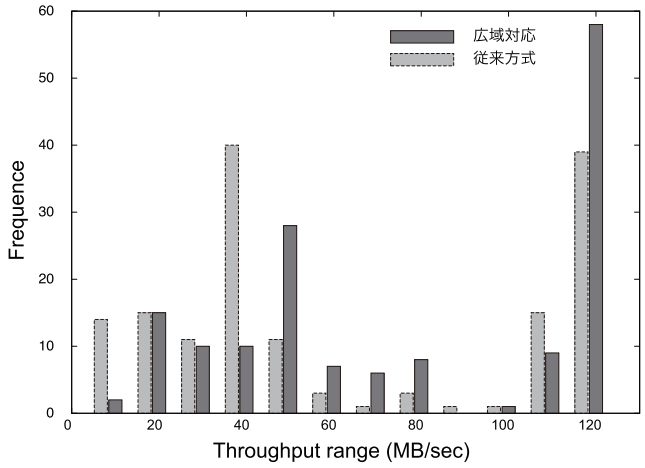

図 6 EXAGE/Storage の Read/Write パフォーマンスのヒストグラム
Fig. 6 Histgram of EXAGE/Storage Read/Write performance.

トは 58.5 MB/sec である．一方，提案する広域分散対応方式では 30〜40 MB/sec のピークが 40〜50 MB/sec に移動し，また 110〜120 MB/sec の頻度は 48.7%増大している．平均スループットは 71.2 MB/sec であり，従来方式より 21.7%の性能向上を実現している．この結果はこの VMM サーバと同一セグメントに配置された同スペックのサーバが持つローカルストレージへの NFS によるアクセスと同等のパフォーマンスを示している．

図 7 に広島拠点の SINET4 L3VPN で観測されたトラフィックの推移を示す．横軸は期間で 1 週間分データ，縦軸はトラフィック量（単位は bps）である．“in” は広島拠点への入力トラフィック，“out” は広島拠点からの出力トラフィックを表しており，いずれも 60 sec 間隔の計測値である．この結果から，広島拠点からの出力トラフィックでは最大で 423 Mbps が観測されていることが分かる．これは，多重度を 3 としているため，広島拠点のアクセスサーバに対して書き込まれたデータについて，コアサーバで複製を作成し，NII および金沢大学に対して SINET4 L3VPN を通してブロックのコピーが行われるためである．複製の通信はユニキャストで行われるため多重度を上げるとそれだけ帯域を占有することとなる．本手法はコアサーバなしのアクセスサーバのみでも拠点として成立するが，コアサーバを設置した拠点については多重化の設定によっては 1 Gbps の帯域が逼迫することが懸念される．

3.3 ライブマイグレーション評価

DR および BCP を実現する際に最もよく利用される手法として，VM のライブマイグレーションによるサービス継続手法が考えられる．一般的に，VM をライブマイグレーションする場合には，VMM 間で共有ストレージを持つ必要がある．そのため，拠点をまたがるような場合では共有ストレージをどのように準備し配置するかが鍵となる．ま

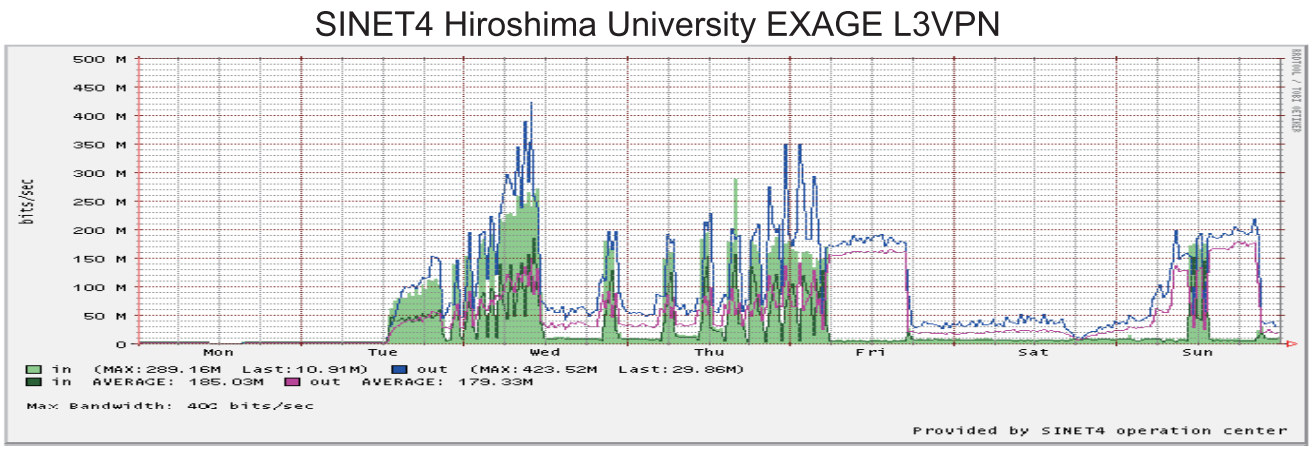

図 7 SINET4 L3VPN のトラフィック状況
Fig. 7 Traffic condition of L3VPN on SINET4.

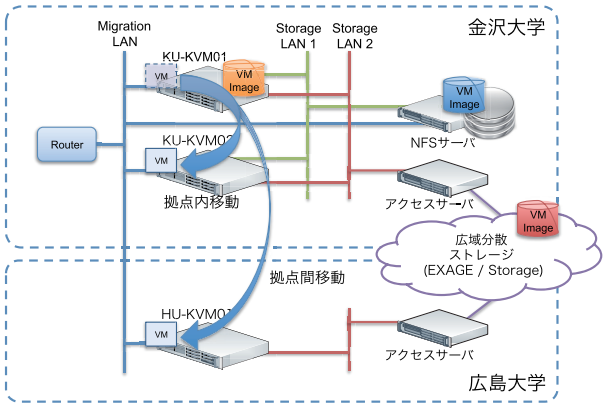

図 8 ライブマイグレーション評価実験の構成図
Fig. 8 Network diagram of live migration evaluation.

た最近の仮想化実装では，共有ストレージを持たない構成におけるマイグレーションとしてストレージマイグレーションも実現されている．

本節では，広域分散ストレージを用いたライブマイグレーションの評価として，次の観点での評価実験を行った．

(1) 自組織内マイグレーションとの比較
(2) ローカル NFS ストレージとの比較
(3) ストレージマイグレーションとの比較

ライブマイグレーション評価に用いる VMM として，今回は KVM[*6]を採用した．評価実験のネットワーク構成図を**図 8** に示す．KVM ホストサーバを金沢大学に 2 台（KU-KVM01，KU-KVM02），広島大学に 1 台（HU-KVM01）用意し，金沢大学内に設置したローカル NFS サーバと EXAGE/Storage による広域分散ストレージを NFSv3 でそれぞれマウントしている．HU-KVM01 ではマイグレーションで利用する L2VPN（MIGRATION-LAN）経由で金沢大学のローカル NFS をマウントしている．また，各サーバ間の接続は 2 Gbps 以上の接続としており，ライブマイグレーションにおけるデータ転送に影響しない構成としている．

[*6] http://linux-kvm.org

表 2 利用した VMM のスペック
Table 2 Specification of VMMs.

	KU-KVM01, 02	HU-KVM01
CPU	Xeon X5760 2CPUs (2.93 GHz)	Xeon E5-2640 2CPUs (2.5 GHz)
メモリ	48 GB	64 GB
HDD	Fujitsu MBD2147RC Rev 5204 (RAID1)	DELL PERC H710 Rev 3.13 (RAID5)
OS	CentOS 6.4 (2.6.32-358.6.2.el6)	
KVM	qemu-kvm-0.12.1.2-2	

今回利用した VMM のハードウェアスペックはそれぞれの拠点で異なるものを利用した（**表 2**）．ただし，KVM を動作させる OS と KVM アプリケーションのバージョンは揃えている．利用する VM は仮想 CPU を 1 つ割り当て，メモリを 1 GB とし，OS には Ubuntu 12.04 を用いている．VM イメージファイルは qcow2 形式を用い，100 GB で作成している．この VM イメージを EXAGE/Storage と NFS サーバ，VMM 上のローカルストレージにそれぞれ配置し，ライブマイグレーションを実施した．ローカルストレージ上の VM に関してはストレージマイグレーションを利用し，比較的移行時間の短い増分コピー（virsh コマンドにおける --copy-storage-inc オプション）を利用した．そのため，評価実験開始時に VMM のローカルストレージに同じ VM イメージファイルを配置している．また，各 VM は同時に起動せず，実験中 1 台の VM しか起動していない状況を保ち，VMM における負荷の影響を最小限とした．

なお，ライブマイグレーションを実施した金沢大学と広島大学の拠点間通信遅延（Round Trip Time: RTT）は約 18 msec であり，vmware 社の vMotion でサポートしている 10 msec[*7]よりも大きな値となっている．

[*7] http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&externalId=2005202

3.4 ライブマイグレーションコストの評価

ライブマイグレーションの性能評価の指標として，VMの移動にかかる時間（処理時間）と外部からの通信が途絶える時間（途絶時間）を利用する．処理時間はライブマイグレーションを実行するvirshコマンドの処理時間とし，timeコマンドを利用して求めている．途絶時間は同一セグメントノードからのICMP応答が途絶える時間とし，pingコマンドを0.01 sec間隔で送信（-i 0.01オプションを利用）した際のロスパケット数から算出した．また，通信処理による影響を評価するため，VM上でHTTPによるファイルダウンロードを実行中にライブマイグレーションを実施し，同様に処理時間と途絶時間を計測した．ダウンロードするファイルとして，Debian DVDのISOイメージファイル[*8]を利用している．

ライブマイグレーション評価実験の結果を**表 3**および**表 4**に示す．表3は処理時間を，表4は途絶時間を表す．それぞれ単位はsecである．ファイルダウンロード中の評価は「(ファイル転送による負荷あり)」として表記している．計測値は3回施行した結果の平均値とした．

処理時間に関しては，ローカルNFSサーバ利用と比較してEXAGE/Storage利用時に3割程度の増加となった．ただし，ファイルダウンロードを実施中の拠点間ライブマイグレーションに関しては，ほぼ同等の処理時間となっている．ファイルのダウンロードに際してメモリが更新されるため，メモリコピーが通信遅延の影響を受けることで処理時間が大きく延びたと考えられる．このことから，VMのメモリ更新頻度の影響が拠点間ライブマイグレーションでは重要となり，移行に際しては考慮が必要となる．ストレージマイグレーションでは，増分コピーとはいえ，メモリコピーよりも多くのデータ転送が発生するため，処理時間の延びが顕著に現れている．そのため，現状のKVMにおける実装を用いる限りでは最適解とはならない．

途絶時間に関しては，ローカルNFSサーバ利用とEXAGE/Storage利用でほぼ同様の結果が得られた．どちらの場合も，拠点間ライブマイグレーションにおいて2倍以上の途絶時間となっていることが分かる．拠点内ライブマイグレーションではファイルダウンロードの影響もなく，メモリコピーの増加は途絶時間に影響しないといえる．ストレージマイグレーションでは，拠点内・拠点間での差異は見られないが，ファイルダウンロードによる処理の影響が致命的である．

以上のことから，広域分散環境におけるライブマイグレーションを行う場合，提案手法である広域分散ストレージ利用はローカルNFSサーバ利用と遜色ない性能を有するといえる．また，ライブマイグレーションを実現するには共有ストレージの利用が必須であり，ローカルストレージ利用によるストレージマイグレーションは現実的ではないことが分かる．

表 3 ライブマイグレーション処理時間の比較（単位：sec）
Table 3 Comparison of processing time for live migration (sec).

	Exage /Storage	Local NFS	Storage Migration
拠点内移動	9.61	7.29	505.4
拠点内移動 (ファイル転送による負荷あり)	11.5	8.34	604.9
拠点間移動	11.98	9.55	473.5
拠点間移動 (ファイル転送による負荷あり)	25.73	26.91	581.2

表 4 ライブマイグレーション途絶時間の比較（単位：sec）
Table 4 Comparison of down time for live migration (sec).

	Exage /Storage	Local NFS	Storage Migration
拠点内移動	0.24	0.21	0.39
拠点内移動 (ファイル転送による負荷あり)	0.17	0.23	43.27
拠点間移動	0.57	0.56	0.41
拠点間移動 (ファイル転送による負荷あり)	0.77	0.63	43.22

[*8] http://ftp.jaist.ac.jp/pub/Linux/debian-cd/7.0.0/ia64/iso-dvd/debian-7.0.0-ia64-DVD-1.iso

3.5 マイグレーション中のI/O性能評価

拠点間ライブマイグレーションに関して，移動の前後におけるディスクアクセスの性能を比較した．移動前はKU-KVM01上で，移動後はHU-KVM01上で各VMからのSequential ReadおよびSequential Write性能を計測した．Sequential Read計測にはhdparmコマンドを，Sequential Write計測にはddコマンドをそれぞれ利用し，10回ずつ施行した結果の平均値として算出している．計測結果を**表 5**に示す．単位はMB/secである．

移動前のEXAGE/StorageとローカルNFSサーバを比較すると，Write性能に関して大きな違いが見て取れる．EXAGE/StorageではWrite処理時に冗長性を確保する処理が追加されるため，この点でNFSサーバにおける処理よりも劣ると考えられる．一方，移動後に関しては，ローカルNFSサーバ利用でReadおよびWrite性能がEXAGE/Storageの場合よりも大きく低下していることが分かる．

表 5 ライブマイグレーション前後のSequential Read/Write性能比較（単位：MB/sec）
Table 5 Performance Comparison of Sequential Read and Write before and after Live Migration (MB/sec).

	Exage /Storage	Local NFS	Storage Migration
Read（移動前）	142.3	150.5	98.8
Read（移動後）	115.3	50.3	261.5
Write（移動前）	45.3	109.3	97.6
Write（移動後）	37.3	61.6	345.6

移動後の VMM ではローカル NFS サーバをリモートマウントすることになり，拠点間の通信遅延の影響を受けていると考えられる．EXAGE/Storage の Read および Write 性能が移動前後で劣化している点は，各拠点におけるコアサーバの差と考えられる．EXAGE/Storage はコアサーバに対して並列に処理するため，コアサーバの台数が多い移動前の金沢大学拠点で性能が高くなっている．なお，比較対象として計測したストレージマイグレーションの結果では，拠点で利用する VMM のストレージ性能に大きく影響を受ける結果となった．

4. おわりに

本稿では DC 内で完結する低遅延環境を対象としたスケールアウトストレージシステムを高遅延環境において適用可能にするためのアーキテクチャを再設計し，国内 3 拠点からなる広域分散ストレージのための検証環境を構築して I/O パフォーマンスとライブマイグレーションを評価した．拠点内に存在する NFS サーバと本提案手法を実装した広域分散ストレージのパフォーマンスを比較し，提案手法は他拠点へブロックを複製しながらも拠点内に存在する NFS サーバと同等の I/O 性能を示すことを明らかにした．この結果から各拠点は DR のためのストレージと自拠点の仮想化基盤のためのストレージとを区別することなく利用できることを示した．

また通信遅延（RTT）が 18 msec の環境において，拠点内に存在する NFS サーバを用いたライブマイグレーションと提案手法を用いたライブマイグレーションは同等の性能となることを確認した．また，マイグレーション後の I/O 性能について，拠点間の通信遅延による影響が 1 拠点の NFS サーバを複数拠点で共有利用する場合と比較して，小さいことを確認した．日米間の通信では 100〜200 msec 程度の遅延が発生するなど，世界規模のグローバルな通信では遅延が大きな問題になりうる．コアサーバ数を増加させることによりパフォーマンスを向上させ，高遅延環境における検証をすることが今後の課題である．

謝辞 本研究は平成 25 年度北海道大学情報基盤センター共同研究「インタークラウド環境での広域分散ストレージ実験と検証」，平成 25 年度国立情報学研究所共同研究「広域分散仮想化環境に関する研究」，平成 24 年度学際大規模情報基盤共同利用・共同研究拠点公募型共同研究「分散クラウドシステムにおける遠隔連携技術」による支援，および JSPS 科研費課題番号「24500083」の助成を受けました．本研究の実証実験にあたり，日本学術振興会産学協力研究委員会インターネット技術第 163 委員会（ITRC）および地域間インタークラウド分科会（RICC）からの支援をいただきました．コンピュータリソースのご提供をいただいた各大学，SINET4 の回線をご提供いただいた国立情報学研究所，および，クラスタストレージ技術である EXAGE/Storage をご提供いただいた株式会社インテック，および，アクセスサーバとして UCS をご提供いただいた Cisco Systems 合同会社に感謝します．

参考文献

[1] 柏崎礼生：スモールスタートで始める大学の仮想化基盤の構築と運用の実情，インターネットと運用技術シンポジウム 2012 論文集，pp.94–101 (2012).
[2] 坂田智之，長谷川孝博，水野信也，永田正樹，井上春樹：情報セキュリティの観点からみた静岡大学の全面クラウド化，情報処理学会研究報告，Vol.2011-IOT-14, No.7, p.1 (2011).
[3] 松原義継，大谷 誠，江藤博文，渡辺健次，只木進一：プライベートクラウドによる電子メール管理コストの低減とサービスレベルの改善—佐賀大学の事例，情報処理学会研究報告，Vol.2011-IOT-14, No.8, pp.1–6 (2011).
[4] Shikida, M., Miyashita, K., Ueno, M. and Uda, S.: An evaluation of private cloud system for desktop environments, *Proc. ACM SIGUCCS 40th Annual Conference on Special Interest Group on University and College Computing Services* (*SIGUCCS '12*), pp.131–134 (2012).
[5] 宮下夏苗，上埜元嗣，宇多 仁，敷田幹文：大学におけるプライベートクラウド環境の構築と利用，第 3 回インターネットと運用技術シンポジウム，pp.17–24 (2010).
[6] 棟朝雅晴，高井昌彰：北海道大学アカデミッククラウドにおけるコンテンツマネジメントシステムの展開，第 10 回情報科学技術フォーラム情報科学技術レターズ，pp.15–18 (2011).
[7] Plummer, D.C., Bittman, T.J., Austin, T., Cearley, D.W. and Smith, D.M.: *Cloud Computing: Defining and Describing an Emerging Phenomenon*, Gartner Research, G00156220 (2008).
[8] Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I. and Zaharia, M.: Above the Clouds: A Berkeley View of Cloud Computing, UCB/EECS-2009-28 (2009).
[9] Badger, L., Grance, T., Patt-Corner, R. and Voas, J.: *DRAFT Cloud Computing Synopsis and Recommendation*, NIST Special Publication 800-146 (2012).
[10] Shafer, J.: I/O virtualization bottlenecks in cloud computing today, *Proc. 2nd Conference on I/O Virtualization* (*WIOV '10*), p.5 (2010).
[11] 関谷勇司：広域分散クラウドへの挑戦と課題，信学技報，Vol.111, No.375, IA2011-63, pp.49–54, 電子情報通信学会 (2012).
[12] Mikami, S., Ohta, K. and Tatebe, O.: Using the Gfarm File System as a POSIX Compatible Storage Platform for Hadoop MapReduce Applications, *12th IEEE/ACM International Conference on Grid Computing* (*GRID*), pp.181–189 (2011).
[13] Ghemawat, S., Gobioff, H. and Leung, S.-T.: The Google file system, *Proc. 19th ACM Symposium on Operating Systems Principles* (*SOSP '03*), pp.29–43 (2003).
[14] Azagury, A., Dreizin, V., Factor, M., Henis, E., Naor, D., Rinetzky, N., Rodeh, O., Satran, J., Tavory, A. and Yerushalmi, L.: Towards an object store, *Proc. 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies* (*MSS '03*), p.165 (2003).
[15] Factor, M., Meth, K., Naor, D., Rodeh, O. and Satran, J.: Object storage: The future building block for storage systems, *Local to Global Data Interoperability – Chal-*

lenges and Technologies, pp.119–123 (2005).
[16] Nakagawa, I. and Nagami, K.: Jobcast – Parallel and distributed processing framework Data processing on a cloud style KVS database, *Journal of Information Processing*, Vol.21, No.3 (2013).
[17] 首藤一幸：key-value ストアの基礎知識, Software Design, 2010 年 2 月号 (2010).
[18] DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P. and Vogels, W.: Dynamo: Amazon's Highly Available Key-value Store, *Proc. 21th ACM SIGOPS Symposium on Operating Systems Principles* (*SOSP '07*), pp.205–220 (2007).

柏崎 礼生 （正会員）

1999 年北海道大学工学部システム工学科卒業．2003 年同大学大学院修士課程修了．2005 年同大学院博士課程中途退学．工学修士．同年北海道大学情報科学研究科助手（後に助教）．2010 年東京藝術大学芸術情報センター特任助教．適応的ネットワークルーティング，インタークラウドコンピューティングに関する研究に従事．情報ネットワークの可視化，人工生命，アニメーション，絶対領域に興味を持つ．電子情報通信学会，IEEE，ACM 各会員．

北口 善明 （正会員）

1995 年新潟大学理学部物理学科卒業．1997 年同大学大学院自然科学研究科修士課程修了．同年株式会社インテックに入社．2004 年電気通信大学大学院情報システム学研究科博士課程単位取得満期退学．2005 年同大学博士（工学）取得．2009 年金沢大学総合メディア基盤センター助教．ネットワークの運用管理および IPv6 の研究に従事．電子情報通信学会会員．

近堂 徹 （正会員）

2001 年広島大学工学部第二類（電気系）卒業．2006 年同大学大学院工学研究科博士課程修了．現在，広島大学情報メディア教育研究センター准教授．博士（工学）．コンピュータネットワーク，リアルタイムマルチメディア通信，QoS 保証技術に関する研究に従事．電子情報通信学会会員．

楠田 友彦

1978 年生．2000 年中央大学理工学部管理工学科卒業．2000 年株式会社インテック入社．ネットワークの経路制御に関する研究に従事．

大沼 善朗

株式会社インテック先端技術研究所事業開発部特別研究員．1993 年に通信系企業に入社し，高精細画像通信に関するソリューション開発に従事．2006 年からサーバ仮想化に関する企画・提案に従事．2008 年に株式会社インテック・ネットコアに入社．EXAGE の初期構想段階から参画し，設計・開発を経て，現在は製品責任者を担当．

中川 郁夫 （正会員）

1993 年東京工業大学大学院総合理工学研究科修士課程修了．同年株式会社インテック入社．2002 年株式会社インテック・ネットコア設立，同社取締役就任．2005 年東京大学大学院情報理工学研究科にて博士号取得．博士（情報理工学）．2010 年株式会社インテックシステム研究所取締役．2011 年株式会社インテック主席研究員．2012 年大阪大学サイバーメディアセンター招聘准教授（兼務）．

阿部 俊二 （正会員）

1980 年 3 月豊橋技術科学大学工学部情報工学課程卒業，1982 年 3 月同大学大学院修士課程修了．1996 年 5 月博士（工学）取得（東京大学）．1982 年 4 月（株）富士通研究所入社，1995 年 6 月学術情報センター，2000 年 4 月国立情報学研究所，文部科学省研究振興局・学術調査官兼務（2008 年 4 月～2012 年 3 月）．現在，国立情報学研究所・准教授，総合研究大学院大学・准教授（兼任），SINET 利用推進室長（兼任）．通信ネットワークの性能評価/性能改善方式，トラフィック解析・制御方式等の研究開発および SINET 構築/運用/利用促進活動等に従事．電子情報通信学会，IEEE 各会員．

横山 重俊

1979 年大阪大学理学部数学科卒業．1981 年大阪大学大学院理学研究科修士課程修了（数学専攻）．同年日本電信電話公社横須賀研究所入所．オペレーティングシステム，分散処理技術，インターネット技術，クラウドコンピューティング基盤技術等の研究開発に従事．1989〜1991 年マサチューセッツ工科大学客員研究員．現在国立情報学研究所勤務，電子情報通信学会会員．博士（情報学）．

下條 真司 （正会員）

1986 年大阪大学基礎工学部大学院後期課程修了．同年大阪大学・助手．1989 年同大型計算機センター・講師．1991 年同助教授，1998 年同教授．2000 年同大学サイバーメディアセンター副センター長，2005 年同センター長，2007 年同副センター長．2008 年から情報通信研究機構大手町ネットワーク研究統括センターセンター長，上席研究員．2011 年から情報通信研究機構テストベッド研究開発推進センターセンター長を兼任．現在に至る．マルチメディア情報システムのアーキテクチャの研究に従事．データベースとネットワークに関連したマルチメディア応用システムに興味を持つ．工学博士．電子情報通信学会，IEEE Computer Society 各会員．