Vol.2016-OS-136 No.2
2016/2/29

仮想デスクトップのストレージトラフィックにおけるブロック間相関関係の解析

熊野 達夫[1,a)]　小沢 年弘[1,b)]

概要：企業や大学などの組織で仮想デスクトップインフラストラクチャー（VDI）が一般的に使われるようになってきている．VDI では複数のユーザが 1 つのストレージ装置を共有するため，ストレージが性能のボトルネックになる可能性がある．このボトルネックを解消するためには，VDI によるストレージ I/O の特性を知る必要がある．そこで，我々は企業で約 300 ユーザによって大規模に使われている VDI サービスで，サーバとストレージ間のネットワークに流れるデータをキャプチャすることにより，ストレージのトレースを取得した．取得したトレースを，アクセスされるブロックの論理アドレスと時刻の観点で分析した．各ブロックを，アクセス頻度の時間変化の類似性でクラスタリングしたところ，アドレスが連続したブロックが同じクラスタに分類されることが多いことが分かった．また，複数の日にわたってこの傾向が大きくは変化しないことが分かった．この結果から，ブロックごとのアクセス傾向の類似度を利用して効果的に階層制御を行う可能性を見出した．

1. はじめに

企業や大学などの組織で VDI が一般的に使われるようになってきている．VDI によってセキュリティの向上や管理の容易さ，信頼性の向上，環境の均一化がもたらされる [1]．VDI は仮想化技術を利用し，一般ユーザのデスクトップ PC をデータセンタのサーバに集約する．VDI では複数のユーザが 1 つのストレージ装置を共有するため，ストレージが性能のボトルネックになる可能性がある．このボトルネックを解消するためには，VDI によるストレージ I/O の特性を知る必要がある．大規模な VDI システムのストレージは一般に iSCSI やファイバーチャネル，FCoE（Fibre Channel over Ethernet）などの SAN（Storage Area Network）を用いて構成される．これは，SAN を用いることにより，サーバを追加してシステムをスケールアウトしたり，サーバの故障時に別のサーバに処理を引き継ぐことができるためである．

本研究では，大規模に実運用されている VDI のストレージトラフィックを分析することで，ストレージシステムの階層制御やキャッシュ制御，先読みアルゴリズムなどの研究に貢献することを目的とする．我々は高スループットで長時間のネットワークキャプチャを行うツール [2], [3] を使用し，社内で大規模に利用されている VDI のストレージトラフィックのトレースを取得した．取得したトレースは連続した 5 日間分で，データのサイズは 3.7 TiB だった．取得したトレースから業務時間内のストレージのリクエストを抽出し，リクエストのストレージへの到達時刻と対象ブロックのアドレスに注目して分析を行った．分析の結果，以下の特徴が新たに明らかになった．

- LBA（論理ブロックアドレス）が近いブロックへのアクセスは日中の時間変動の傾向が似ている．容量が 4.6 TiB の論理ボリュームについて，LBA を 1 GiB ごとに分割して（これをエクステントと呼ぶ．），1 分ごとのアクセス数の日中の時間変動の傾向が最も似ているエクステントを求めると，LBA の差が 50 GiB 以内のエクステントが 70%を占めていた．
- 日中の時間変動の傾向が似ているエクステントは，複数の日にわたって関係が大きくは変化しない．エクステントのアクセス数の時間変動を多次元ベクトルとして，最初の日に日中の時間変動の傾向が最も似ているエクステントについて，複数の日にわたって初日とのユークリッド距離の比率を調べると，4 日間にわたって平均値の変動は ±7% 以内に収まっていた．
- 日中のアクセス傾向の時間変動が似ているエクステントをクラスタリングすると，複数の日にわたって傾向が大きくは変化しない．トレースを取得した 5 日間のうち，それぞれの日について業務時間内のアクセス傾向の日中の時間変動が似ているものが同じクラスタに

[1] 株式会社富士通研究所
[a)] kumano_tatsuo@jp.fujitsu.com
[b)] t.ozawa@jp.fujitsu.com

© 2016 Information Processing Society of Japan

属するように分類した．このクラスタを初日のものと比較すると，80%〜85%のエクステントが同じクラスタに分類されていた．

2. 関連研究

従来から，ストレージシステムの改善を目指してトレースを用いた研究は盛んに行われてきた．大規模な実環境のトレースを使った研究としては，各種サーバの負荷を調べたもの [4] や Web サービスの Key-Value ストアへのアクセス傾向を分析したもの [5]，科学技術計算を行うシステムでのワークロードを分析したもの [6], [7]，企業のファイルサーバへのアクセス特性を調べたもの [8] が知られているが，大規模なシステムでの実データを使った VDI に関するものはない．仮想環境でのストレージトラフィックについての研究もある [9], [10], [11] が，これらはベンチマークツールなどの人工的な負荷を用いて評価している．VDI のストレージに関しては，人工的に生成した負荷を用いた分析 [12], [13] や小規模なシステムでの分析 [14] に留まる．

本研究ではデータマイニングの手法を用いてブロック間の相関関係に注目して分析する．ストレージシステムのワークロードにおいて，ブロック間の相関関係を利用したものとしては Li らによる研究 [15], [16] が知られている．Li らの手法ではシステム上で単一のアプリケーションが動作することを想定して，直後のアクセスを予測してプリフェッチを行う．単純なベンチマークツールによるワークロードでは効果が出ているが，VDI では複数のユーザが使用する複数のアプリケーションによるアクセスが混在すると考えられるので，実際のデータを用いて評価する必要がある．これ以外に，Seo らによる研究 [17] では，決定木を用いてシーケンシャル，ランダムなどの負荷パターンを判別する．ウェブサーバやメールサーバ，ファイルサーバを単体で動かした場合には効果があるが，仮想環境では負荷の混在によってランダム性が増すことが知られている [18] ので，VDI に適用するのは難しい．ネットワークの分野で実際の大規模なデータをデータマイニングの手法で分析したものとしては，Wang らの研究 [19] がある．Wang らは携帯電話の基地局ごとの通信量をもとに基地局を分類してモデル化を行い，負荷の予測を行っている．ストレージの分野でも，サーバごとの負荷予測は同様の手法で可能と考えられるが，本研究では複数のアプリケーションによる負荷が混じった状態でのブロック単位の分析を行う．

3. トレースの取得

3.1 VDI

VDI によってセキュリティの向上や管理の容易さ，信頼性の向上，環境の均一化がもたらされる [1]．このため，企業や大学などの大規模な組織で幅広く使われるようになっている．VDI では，仮想化技術を用いてデスクトップ環境をデータセンタのサーバに集約する．大規模な VDI システムでは，サーバは SAN によってストレージと接続し，それぞれのユーザが使用する仮想マシン（VM）のデータはこのストレージに格納する．

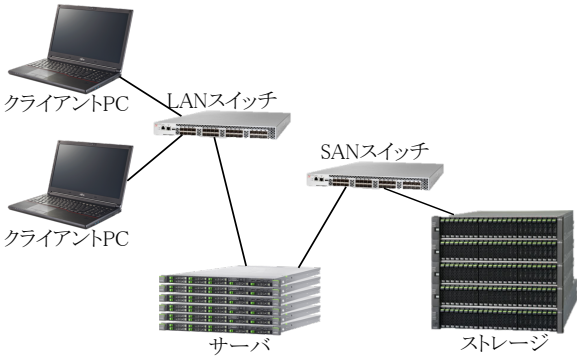


図 1 一般的な VDI でのハードウェア構成．クライアント PC は LAN を経由してサーバに接続し，サーバは SAN を経由してストレージに接続する．

一般的な VDI でのハードウェア構成を図 1 に示す．VDI では，サーバ，ストレージ，サーバとストレージを接続する SAN，クライアント PC，サーバとクライアント PC を接続する LAN（Local Area Network）から構成されることが多い．

VDI を構成するソフトウェアとしては，サーバ上で動作する Microsoft Windows Server[20] や Xen[21]，VMware vSphere[22] などのハイパーバイザや，これらを管理する Citrix XenDesktop[23] や VMware Horizon[24] などの VDI 管理ソフトウェア，ハイパーバイザで実行される VM で動作する Microsoft Windows[25] や Linux などの OS がある．VM では様々な OS が動作するが，共通の OS を複数ユーザで使用することが多いため，VDI では OS のマスターイメージを共有する機能が提供されている．この機能を使用することでストレージの使用量を減らすことができるというメリットがあるが，場合によってはユーザが自由にアプリケーションをインストールできないなどの制約が発生することもある．

3.2 被測定対象システム

本研究でトレースを取得する対象とした VDI システムのハードウェア構成を図 2 に示す．VDI を動かすサーバ（VDI サーバと呼ぶ．）は 6 台あり，1 台のストレージに 10GbE FCoE の SAN を経由して接続している．VDI サーバは全て富士通製の PRIMERGY RX200 S8 で，CPU は Intel Xeon E5-2695v2（2.4 GHz，12 コア），メモリは 256 GiB を搭載している．ストレージは 1 台で，富士通製の ETERNUS DX90 S2，VM が使用する HDD は 900 GB SAS 10 krpm を 60 台搭載しており，RAID 6 で 8 TB の論理ボリューム（LUN）を 6 つ構成している．LUN と VDI サーバは静的に 1 つずつ対応して割り当てている．ユーザが使用する

© 2016 Information Processing Society of Japan

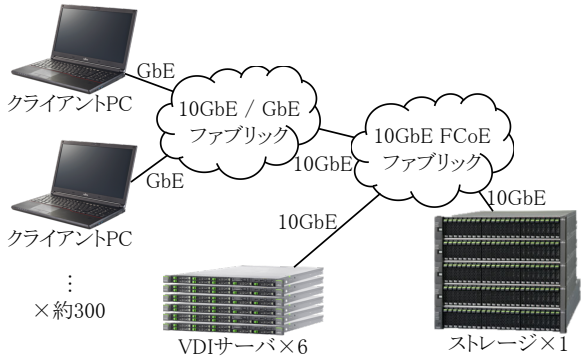


図 2 本研究でトレースを取得する対象とした VDI システムのハードウェア構成．6 台の VDI サーバが 10GbE の FCoE ファブリックを経由してストレージに接続している．VM 数は約 300，ストレージは 8TB × 6 と，大規模である．

VM は，32 ビットまたは 64 ビットの Windows 7 が動作している．割り当てたコアの数は 2 で，メモリは 2.32 GiB，HDD は 100 GiB である．1 台の VDI サーバ上で動作する VM の数は，約 50 である．このシステムでは，OS のマスターイメージを共有することはせず，VM を配備するときにマスターイメージのコピーを行っている．VM に割り当てたシステム領域への変更はログオフしても保存されるため，ユーザは通常の PC と同様にアプリケーションをインストールすることができる．Windows のシステム更新やセキュリティソフトのアップデートもユーザが行う．VM は 24 時間，常に起動したままで運用しており，ユーザはログオンしたまま接続／切断を繰り返すか，ログオン／ログオフを繰り返して使用している．このため，始業時間帯に OS イメージを一斉に読み出すブートストームと呼ばれる現象は発生しない．しかし，Windows のシステム更新やセキュリティソフトのアップデートは一斉に起こる可能性がある．夜間は別のストレージにデータをバックアップしており，ストレージには高い負荷がかかっている．VDI 管理ソフトウェアは Citrix XenDesktop 5.6，ハイパーバイザは Microsoft Windows Server 2008 R2 の Hyper-V を使用している．対象システムの諸元をまとめたものを表 1 に示す．

3.3 トレースの取得方法

VDI におけるストレージのトレースを取得するため，サーバとストレージの間を流れる FCoE のネットワークに流れるパケットをキャプチャした．FCoE ファブリックのポートミラー機能を使い，ストレージが接続しているポートをパケットをキャプチャするサーバ（キャプチャサーバと呼ぶ．）にコピーした．キャプチャサーバを加えたシステムのハードウェア構成を図 3 に示す．今回のデータ取得ではミラー対象のポートの送信／受信をまとめて 1 つのポートにミラーしたため，まれに，ファブリック内部でバッファ溢れが発生して約 0.1%の数のパケットがミラーできなかった．キャプチャサーバ上では我々が開発したソフトウェア [2], [3] を使い，パケットをペイロードも含めて pcap 形式で保存した．キャプチャサーバは富士通製の PRIMERGY RX300 S7 を 1 台使用した．CPU は Intel Xeon E5-2670（2.6 GHz，8 コア），メモリは 256 GiB，HDD は 15 TiB，ネットワークインターフェイスカードは Intel 82599 10 GbE を搭載している．キャプチャサーバの諸元を表 2 に示す．

表 1 対象システムの諸元．

VDI サーバ	機種	Fujitsu PRIMERGY RX200 S8
	CPU	Intel Xeon E5-2695v2 （2.4 GHz，12 コア）
	メモリ	256 GiB
	OS	Microsoft Windows Server 2008 R2
	VDI	Citrix XenDesktop Hyper-V
	台数	6
ストレージ	機種	Fujitsu ETERNUS DX90 S2
	HDD	900 GB SAS 10 krpm × 60
	LUN	8 TB（RAID 6）× 6
	台数	1
ネットワーク	LAN	10GbE / GbE ファブリック
	SAN	10GbE FCoE ファブリック
VM	CPU	2 コア
	メモリ	2.32 GiB
	HDD	100 GiB
	OS	Windows 7（32 bit または 64 bit）
	動作 VM 数	約 300

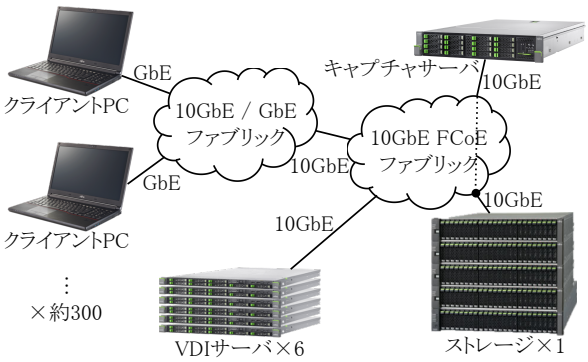


図 3 キャプチャサーバを加えたシステムのハードウェア構成．キャプチャサーバはポートミラーによってストレージの通信内容をコピーして受け取る．

表 2 キャプチャサーバの諸元．

機種	Fujitsu PRIMERGY RX300 S7
CPU	Intel Xeon E5-2670（2.6 GHz，8 コア）
メモリ	256 GiB
HDD	15 TiB
NIC	Intel 82599 10 GbE（デュアルポート） × 1
OS	CentOS 6.4 x86_64
台数	1

© 2016 Information Processing Society of Japan

表 3 取得したデータの概要.

期間	5 日と 3 時間 12 分間
データサイズ	3.7 TiB
コマンド数	1,483 M 個（読み 83%，書き 17%）
転送量	70.4 TiB（読み 90%，書き 10%）

4. 取得したトレースの分析

4.1 取得したデータ

取得したデータの内容としては，コマンドのパケットについては時刻，レスポンス時間，MAC アドレス，読み書き種別，LUN の識別子，LBA（論理ブロックアドレス），サイズ，データのパケットについては時刻，MAC アドレス，読み書き種別，LUN の識別子，LBA，サイズ，データ内容の 512 バイト毎の SHA-1 ハッシュ値を求めたものを格納している．2015 年 9 月 1 日（火）19 時 9 分から 9 月 6 日（日）22 時 21 分までの 5 日と 3 時間 12 分間のトラフィックからトレースを取得した．取得したトレースに含まれるコマンド数は 1,483 M 個，転送量は 70.4 TiB，コマンド数で見るとストレージからの読み出しが 83%で書き込みが 17%，転送量で見ると読み出しが 90%で書き込みが 10%だった．取得したデータの概要を表 3 に示す.

4.2 分析方法

4.2.1 分析対象データの処理

取得したトレースには夜間のデータも含まれている．本研究で解析対象とした VDI システムでは夜間にバックアップを行っているため，ストレージへの負荷が高い．ユーザが使用する VM が発行するストレージトラフィックの解析を目的としているため，8 時から 20 時までの 12 時間分のみを解析対象とした．解析対象としたトレースに含まれるコマンド数は 907 M 個，転送量は 22.7 TiB，コマンド数で見るとストレージからの読み出しが 81%で書き込みが 19%，転送量で見ると読み出しが 86%で書き込みが 14%だった．

本研究では時間的，空間的なアクセス傾向について解析する．ストレージへのアクセスは，時刻はマイクロ秒単位で記録されていて，LBA は 512 バイト単位でコマンドが発行される．ある LUN における，コマンドの発行時刻と LBA の例を図 4 に示す．各点がアクセスを表している．大局的なアクセス傾向をつかむため，ここでは，時刻と LBA を一定の単位で区切ってコマンドの情報を集計した．時刻を区切ったものをエポック，LBA を区切ったものをエクステントと呼ぶ．集計する情報は，読み書き別のコマンド数と転送量とする．図 4 に示した例について，一定の単位で区切ってコマンド数を集計した例を図 5 に示す．この例では，エポックは 1 分，エクステントは 1 GiB 単位で区切っている．

本研究では，エクステント間の相関関係に注目して解析

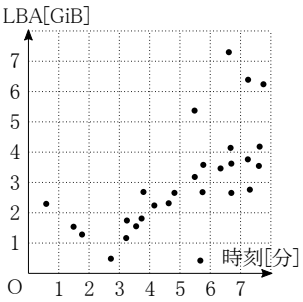


図 4 コマンドの発行時刻と LBA の例.

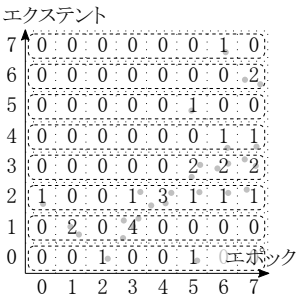


図 5 図 4 を一定の単位で区切って集計した例.

を行う．各エクステントについて，エポックの数に対応する多次元ベクトルと考えて，エクステント間の相関関係を求める．図 5 に示した例では，0〜7 の 8 つのエクステントが存在する．これらを，0〜7 の 8 つのエポックに区切って集計したものを，それぞれ 8 次元のベクトルとして扱う．この例では，エクステント i を表すベクトルを $\vec{x}_i$ とすると，以下の値を持つ.

$$\vec{x}_0 = (0,0,1,0,0,1,0,0)$$
$$\vec{x}_1 = (0,2,0,4,0,0,0,0)$$
$$\vec{x}_2 = (1,0,0,1,3,1,1,1)$$
$$\vec{x}_3 = (0,0,0,0,0,2,2,2)$$
$$\vec{x}_4 = (0,0,0,0,0,0,1,1)$$
$$\vec{x}_5 = (0,0,0,0,0,1,0,0)$$
$$\vec{x}_6 = (0,0,0,0,0,0,0,2)$$
$$\vec{x}_7 = (0,0,0,0,0,0,1,0)$$

この例では分かりやすくするために値をそのまま多次元ベクトルにしているが，実際の解析では，正の値を持つものは自然対数をとっている．エクステント i，エポック j におけるアクセス数を $a_{i,j}$ とすると，エクステント i に対応する多次元ベクトル $\vec{x}_i$ の $j+1$ 次元目の値 $x_{i,j}$ は以下の式で求める．

$$x_{i,j} = \begin{cases} 0 & \left(a_{i,j} = 0\right) \\ \log a_{i,j} & \left(a_{i,j} > 0\right) \end{cases}$$

4.2.2 相関関係の評価基準

エクステント間の相関関係を表す尺度として，多次元ベクトルのユークリッド距離を用いる．距離が小さいエクステント同士はアクセスの時間変化の傾向が似ていると言える．エクステント $\vec{x}_p$ と $\vec{x}_q$ のユークリッド距離 $d\left(\vec{x}_p,\vec{x}_q\right)$ は，エポックを j とすると，以下の式で表される.

$$d\left(\vec{x}_p,\vec{x}_q\right) = \sqrt{\sum_j \left(x_{q,j} - x_{p,j}\right)^2}$$

アクセスの傾向が似ているエクステントのグループが存在して，これらの関係がある程度の期間にわたって大きく変動しない場合，このグループを事前に調べておくことで，

© 2016 Information Processing Society of Japan

ストレージシステムの階層制御やキャッシュ制御，先読みなどを効果的に行えるようになる可能性がある．似た属性のものをグループに分ける手法として，k-means[26] や階層的クラスタリングが知られている．本研究ではクラスタリング結果の時間変動に注目して解析をする．k-means は乱数を使うので，クラスタリング結果同士の比較が行いにくい．このため，階層的クラスタリングによってエクステントのグループ分けを試みる．階層的クラスタリングのアルゴリズムとしては，Ward 法 [27] を使用する．VDI では VM ごとにアクセス傾向が異なることが予想されるので，クラスタ数は 1 つの LUN に対応した VM 数である 50 とする．

4.2.3 相関関係の日次変動

本研究で解析対象とするデータは 5 日分の業務時間帯のデータを含んでいる．アクセスの傾向が似たエクステントの相関関係が複数の日にわたってどのように変化するかを確認するため，以下の 2 つの基準で評価する．

(1) 最も距離が近いエクステントとの距離の変化
(2) それぞれの日でクラスタリングした結果の共通性

1 の距離の変化では，まず分析対象とする初日について，全エクステント間で距離を求める．次に，各エクステントについて，最も距離が近いエクステントを求める．初日を 0 としたときの m 日目のエクステント i に対応する多次元ベクトルを $\vec{x}_{m,i}$，初日のデータでエクステント p と最も近いエクステントを $\mathrm{nn}(p)$ とすると，初日にエクステント p と最も距離が近いエクステントとの距離 $\mathrm{dist}_{0,p}$ は以下の式で表される．

$$\mathrm{dist}_{0,p} = d\left(\vec{x}_{0,p}, \vec{x}_{0,\mathrm{nn}(p)}\right)$$

翌日以降については，全エクステント間で距離を求めることはせず，各エクステントについて，初日に最も距離が近かったエクステント間の距離のみを求める．初日に求められた距離をこの距離で割ったものを，距離の変化とする．m 日目のエクステント i について，距離の変化 $\mathrm{dist\_rate}_{m,i}$ は以下の式で表される．

$$\mathrm{dist\_rate}_{m,i} = \frac{d\left(\vec{x}_{0,i}, \vec{x}_{0,\mathrm{nn}(i)}\right)}{d\left(\vec{x}_{m,i}, \vec{x}_{m,\mathrm{nn}(i)}\right)}$$

全エクステントについてこの値を計算し，この平均値で距離の変化を評価する．各エクステントで距離が変化しない場合はこの値は 1 に近づき，日が経過して距離が大きくなると，この値は小さくなって 0 に近づく．

2 のクラスタリング結果の共通性では，それぞれの日についてエクステントをクラスタリングした結果を，初日のデータを用いたクラスタリング結果と比較する．評価する値としては，初日にクラスタリングした結果の各クラスタについて，それぞれの日にクラスタリングした結果と最も共通のエクステントを多く持つものを見つけ，共通のエクステント数を求める．この値を合計したもので評価する．初日のクラスタ k と最も共通のエクステントを多く持つ m 日目のクラスタを $\mathrm{cc}(m,k)$，2 つのクラスタ k,l の共通のエクステント数を $\mathrm{num\_intersect}(k,l)$ とすると，初日と m 日目の共通のエクステント数の割合 $\mathrm{intersect\_rate}_m$ は以下の式で表される．

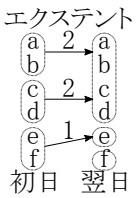


図 6 共通のエクステント数を求める例.

$$\begin{aligned}\mathrm{intersect\_rate}_m &= \frac{\sum_k \mathrm{num\_intersect}\,(k, \mathrm{cc}(m,k))}{\sum_k \mathrm{num\_intersect}\,(k, \mathrm{cc}(0,k))} \\ &= \frac{\sum_k \mathrm{num\_intersect}\,(k, \mathrm{cc}(m,k))}{\text{number of extents}}\end{aligned}$$

これは 0 から 1 の範囲の値をとり，クラスタリング結果が似ていると 1 に近づき，異なると小さくなる．例えば，a, b, c, d, e, f の 6 つのエクステントがあり，これらを m 日目に 3 つのクラスタ $c_{m,1}$, $c_{m,2}$, $c_{m,3}$ に分ける例を考える．ここで，初日には $c_{0,1} = \{a,b\}$，$c_{0,2} = \{c,d\}$，$c_{0,3} = \{e,f\}$，翌日には $c_{1,1} = \{a,b,c,d\}$, $c_{1,2} = \{e\}$, $c_{1,3} = \{f\}$ とクラスタリングされたとする．この場合，初日のクラスタ $c_{0,1}$, $c_{0,2}$, $c_{0,3}$ のそれぞれについて翌日のクラスタと最も共通のエクステントが多いものを求めるとそれぞれ $c_{1,1}$, $c_{1,1}$, $c_{1,2}$ となり，共通のエクステント数は $|c_{0,1} \cup c_{1,1}| + |c_{c_{0,2}} \cup c_{1,1}| + |c_{0,3} \cup c_{1,2}| = 2+2+1 = 5$ となる．初日と翌日の共通のエクステント数の割合は $\mathrm{intersect\_rate}_1 = \frac{5}{6}$ として求められる．この例を図示したものを図 6 に示す.

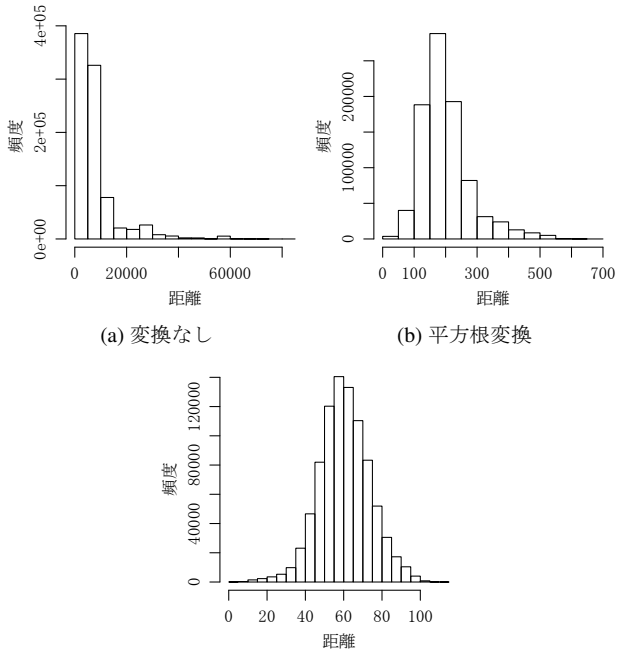
4.3 分析結果

4.3.1 距離の分布

分析対象とするシステムでは，6 台ある VDI サーバをそれぞれストレージ装置の 1 つの LUN に対応付けている．約 300 ユーザを負荷分散のために 6 台の VDI サーバに分けており，それぞれでは同じような動きをしていると考えられるため，ここでは 1 つの LUN に注目して解析を進める．

4.2.1 では集計した値を対数変換することにした．取得したトレースで，変換を行わない場合，平方根変換，対数変換の 3 種類について，全エクステント間の距離を求めて分布を調べたものを図 7 に示す．使用したのは 2015 年 9 月 2 日（水）の 8 時から 20 時までのトレースで，ある 1 つの LUN の読み込み回数を集計したものである．エクステントは 4 GiB 単位，エポックは 1 分間隔に区切って集計した．この結果から，変換なしでは距離が大きいもののばらつきが多く，対数変換をすることによって正規分布に近い分布になっていることが分かる．このため，今後の解析で

© 2016 Information Processing Society of Japan



(a) 変換なし　(b) 平方根変換

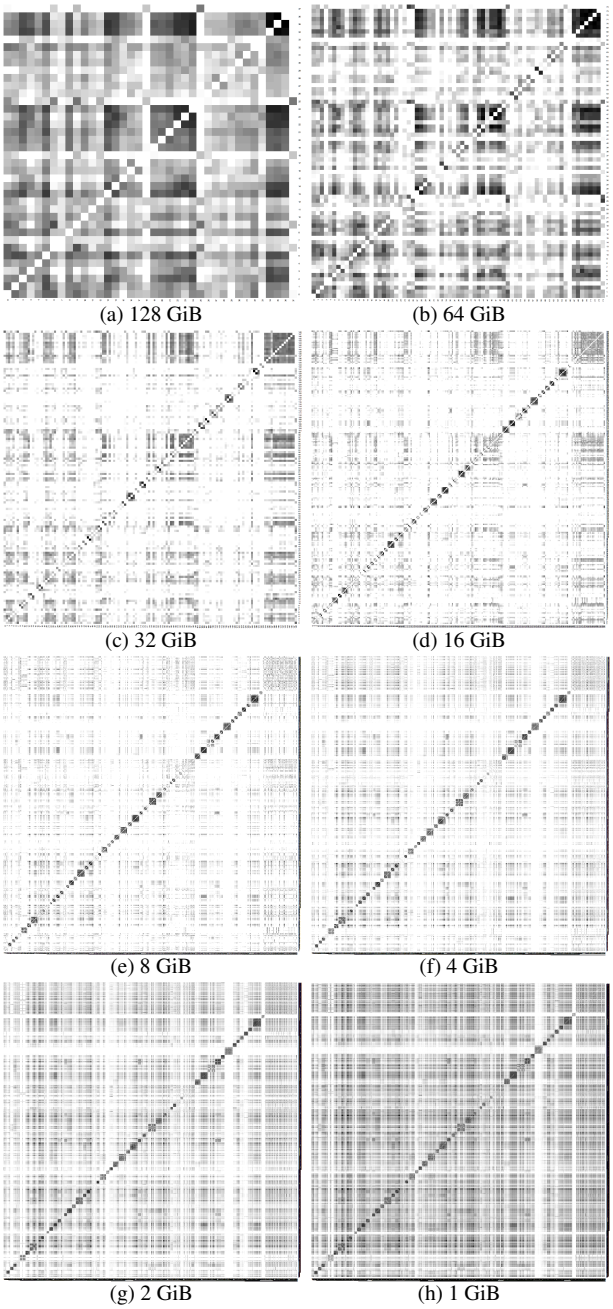
(c) 対数変換

図 7 集計した値に対して変換なし，平方根変換，対数変換をした場合の全エクステント間の距離の頻度分布．

は集計した値を対数変換したものを対象にする．

4.3.2 エクステント間の距離

アクセスの傾向が似たエクステントが存在するかを確認するため，全エクステント間で距離を求めて可視化する．2015 年 9 月 2 日（水）の 8 時から 20 時までのトレースを使い，ある 1 つの LUN の読み込み回数を集計したものについて，エクステントのサイズを 128 GiB，64 GiB，32 GiB，16 GiB，8 GiB，4 GiB，2 GiB，1 GiB と変化させながら，全エクステント間で距離を求めて行列形式で表したもの（距離行列と呼ぶ．）を図 8 に示す．エポックの間隔は 1 分とした．グラフの縦軸と横軸はどちらもエクステントを表しており，色は対応するエクステント間の距離を表している．エクステントは LBA の順に並んでおり，左下が最も小さい LBA を表している．黒い部分は距離が小さいことを表している．今回距離として使用したユークリッド距離は 2 つのエクステントを交換しても同じ値になるので，距離行列は対角要素に対して対称になる．同じエクステント同士の距離は 0 になるため，ここでは値を削除して白色で表している．エクステントのサイズが 32 GiB より小さいとき，距離行列の対角要素付近に黒い正方形の塊を見つけることができる．これは，LBA の近いエクステントで，ある程度の大きさのグループが存在し，これらの間ではアクセスの傾向が似ていることを意味する．エクステントのサイズを小さくするに従って，このグループがはっきり確認できるようになるので，これ以降はエクステントのサイズが 1 GiB の場合に注目して解析をする．エクステントのサイズが小さくなるにしたがって，アクセスが発生する時間が極端に少ないエクステントが増える．これらの間



(a) 128 GiB　(b) 64 GiB

(c) 32 GiB　(d) 16 GiB

(e) 8 GiB　(f) 4 GiB

(g) 2 GiB　(h) 1 GiB

図 8 エクステントのサイズを変えながら全エクステント間で距離を求めた結果．黒い部分は距離が小さいことを表している．

では距離が近くなるため，LBA が広範囲に離れた多数のエクステントと距離が近いものも増えている．

アクセスの傾向が似ているエクステントのグループに関する分析をする前に，まずは各エクステントから最も近いエクステントについて特徴を確認する．2015 年 9 月 2 日（水）の 8 時から 20 時までのトレースを使い，エクステントは 1 GiB，エポックは 1 分単位で区切って集計したものについて，各エクステントから最も近いエクステントの LBA を描画したものを図 9 に示す．グラフの縦軸は注目するエクステントの LBA（TiB），横軸は最も近いエクステントの LBA（TiB）とした．対角要素付近に多くの点が存在することから，多くのエクステントは LBA の差が小さ

© 2016 Information Processing Society of Japan

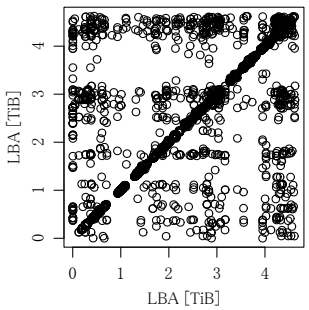


図 9 最も近いエクステントの LBA.

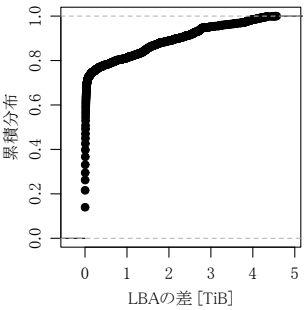


図 10 最も近いエクステントとの LBA 差の累積分布.

いエクステントと距離が近いことが分かる．最も近いエクステントとの LBA の差について，累積分布を求めたものを図 10 に示す．グラフの横軸は LBA（TiB）の差を表す．LBA の差が小さいエクステントとアクセスの傾向が最も似ているエクステントが多数を占めることが分かる．LBA の差が 1 GiB 以下のエクステントの割合は 14%，10 GiB 以下は 47%，50 GiB 以下は 70%であった．

距離が近いエクステントのグループが日によって変動するかを確認するため，2015 年 9 月 2 日（水），3 日（木），4 日（金），5 日（土），6 日（日）の 5 日間について，それぞれ 8 時から 20 時までの 12 時間分のトレースを使って，全エクステント間の距離を求めた．求められた距離行列を図 8 と同様の形式で図 11 に示す．ここで，エクステントに対応する多次元ベクトルは 12 時間分で，他の日のエクステントとは距離を計算せず，同じ日の中で距離を求めた．この 5 日分の距離行列を比較すると，対角要素付近の距離が小さいグループはどれも同じ場所に存在することが分かる．このことから，アクセスの傾向が似たグループは複数の日にわたって変動しない可能性があることが分かった．

より定量的に評価するため，4.2.3 で述べた，最も距離が近いエクステントとの距離の変化について分析する．先程と同様に，2015 年 9 月 2 日（水），3 日（木），4 日（金），5 日（土），6 日（日）の 5 日間について，それぞれ 8 時から 20 時までの 12 時間分のトレースを使った．まず，9 月 2 日の 12 時間分のトレースについて全エクステント間の距離を求め，各エクステントについて最も距離が近いエクステントを求めた．次に，5 日分それぞれについて，上記で求められたエクステント間の距離を計算し，9 月 2 日のものと比較をした．各エクステントについて，9 月 2 日のトレースを使ったときの距離を求められた距離で割った値をエラーバー付き平均折れ線グラフで示したものを図 12 に示す．ここで，エラーバーの範囲は 95%信頼区間を表している．また，平均と分散の変動を表 4 に示す．これらの結果から，4 日間にわたって平均値の変動は ±7% 以内に収まっており，9 月 2 日に最も近かったエクステント間の距離は大きくは変化しないことが分かった．

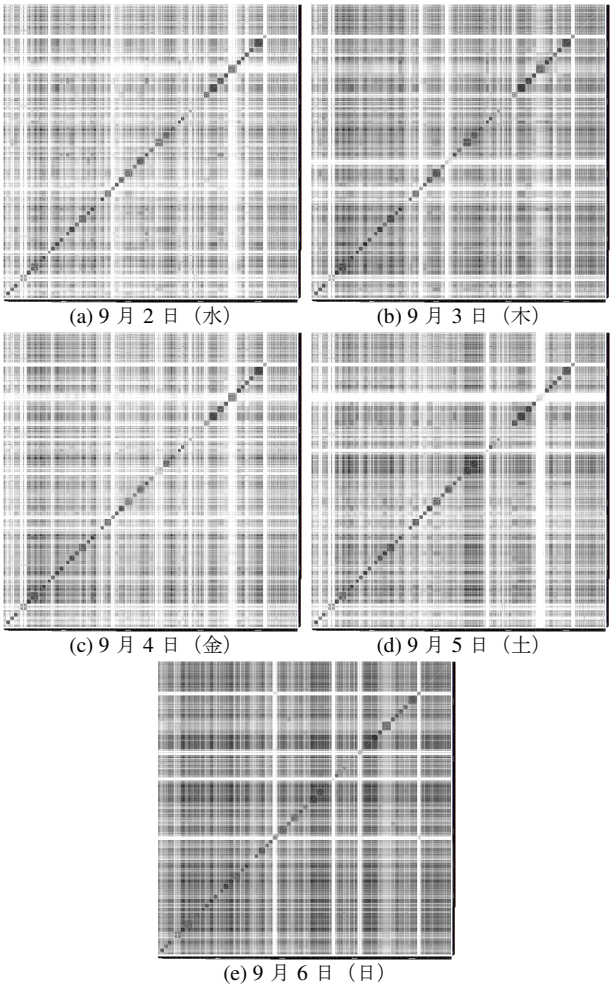


図 11 それぞれの日で，全エクステント間で距離を求めた結果．黒い部分は距離が小さいことを表している．

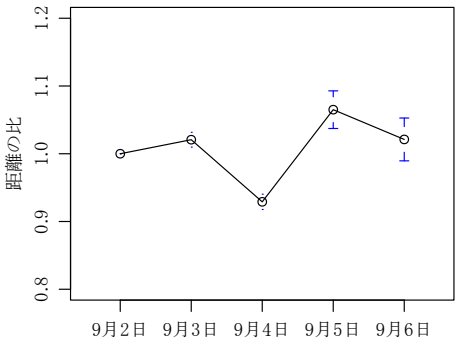


図 12 最も近いエクステントとの距離の変化.

表 4 最も近いエクステントとの距離の変化.

日付	平均	分散
9 月 2 日（水）	1	0
9 月 3 日（木）	1.02	0.10
9 月 4 日（金）	0.93	0.11
9 月 5 日（土）	1.07	0.65
9 月 6 日（日）	1.02	0.84

4.3.3 エクステントのクラスタリング

全エクステント間の距離を求めた際に，距離が近いエクステントのグループが存在することが分かったので，これ

© 2016 Information Processing Society of Japan

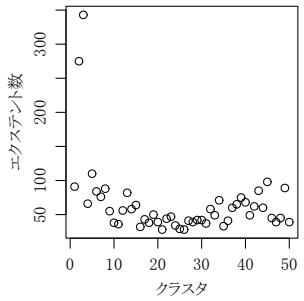


図 13 各クラスタに含まれるエクステントの数.

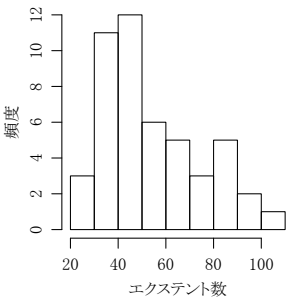


図 14 クラスタに含まれるエクステント数の頻度分布.

らをクラスタリングによって分類する．4.2.2 で述べた通り，ここではユークリッド距離を用いて Ward 法による階層的クラスタリングを行い，エクステントを 50 個のクラスタに分ける．まずは 1 日分のデータについてエクステントのクラスタリングを行い，結果を評価する．2015 年 9 月 2 日（水）の 8 時から 20 時までのトレースを使い，ある 1 つの LUN の読み込み回数を集計したものについて，エクステントは 1 GiB，エポックは 1 分単位で集計したものをエクステントごとにクラスタリングした．得られた各クラスタに含まれるエクステントの数を図 13 に示す．エクステント数が他のクラスタに比べて飛びぬけて多いクラスタが 2 つあり，残りはほぼ 100 以下になっている．エクステント数の多い 2 つのクラスタを除外したときの，各クラスタに含まれるエクステント数の頻度分布を図 14 に示す．40 個程度のものが最も多く，その周辺に分布していることが分かる．横軸にエポック，縦軸にエクステントの LBA をとり，アクセス回数を対数変換したものが一定の閾値を超えたものについて，クラスタによって色分けして描画したものを図 15 に示す．ここでは閾値を 4 とした．クラスタ数は 50 としたが，区別のために色は 8 つだけ使っているので，同じ色が複数のクラスタに対応している．この図から，同じ高さの矩形が横方向に連続して並ぶ形が見え，それらが別のクラスタに分類されていることが分かる．横軸をクラスタ，縦軸をエクステントの LBA としたグラフを図 16 に示す．赤色で示した部分は縦方向に点が続いているもので，各クラスタに縦方向に連続した点が見えることから，LBA が連続したエクステントが同じクラスタに分類されているものが多いことが分かる．

複数の日について，全エクステント間の距離を求めた結果から，距離が近いエクステントのグループが変動しないことが 4.3.2 で分かった．クラスタリングの結果についても日によって変動しないことを確認するため，複数の日についてクラスタリングを行い，4.2.3 で述べた方法によって，各エクステントが同じようにクラスタリングされるかどうかを調べる．クラスタリング結果が日によって変動するかを確認するため，2015 年 9 月 2 日（水），3 日（木），4 日（金），5 日（土），6 日（日）の 5 日間について，それ

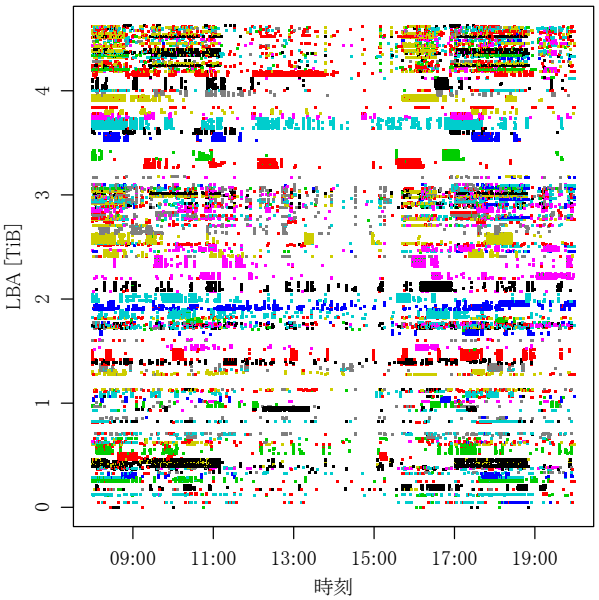


図 15 ブロックアクセスの時間変化をクラスタによって色分けしたもの.

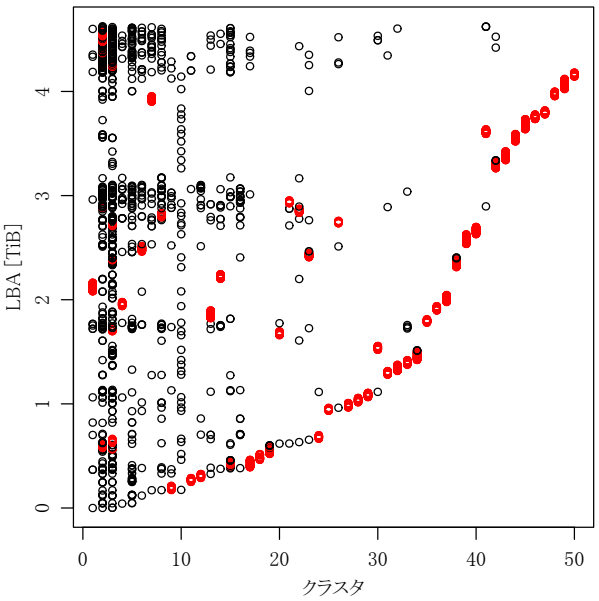


図 16 クラスタとエクステントの LBA の関係.

ぞれ 8 時から 20 時までの 12 時間分のトレースを使って，先程と同様にクラスタリングを行った．ここでもエクステントは 1 GiB，エポックは 1 分単位で集計した．9 月 2 日のトレースを使って求めた各クラスタについて，それぞれの日で最も共通のエクステントを含むクラスタを求めて共通のエクステント数の割合で評価した．この値が 1 に近いと共通のエクステントが多いことを表す．この結果を図 17 に示す．この結果から，数日間にわたって，クラスタリング結果のうち 80%〜85%が同じクラスタに分類されることが分かった．

© 2016 Information Processing Society of Japan

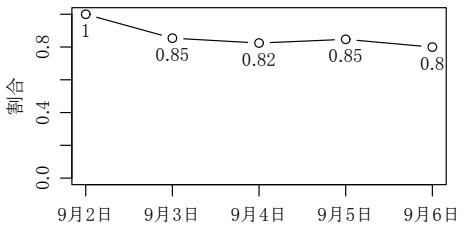


図 17 クラスタリング結果の共通エクステント数の割合.

5. まとめ

企業における VDI でのストレージへのアクセス傾向を知るため，約 300 ユーザによって大規模に使われている VDI システムを対象としてストレージトラフィックのトレースを取得した．取得したトレースから業務時間内のストレージのリクエストを抽出した結果，LBA が近いブロックへのアクセスはアクセスの傾向が似ていることが分かった．また，時間変動の傾向が似ているエクステントは，複数の日にわたって関係が大きくは変動しないことが分かった．さらに，アクセス傾向が似ているエクステントをクラスタリングしたところ，複数の日にわたって傾向が大きくは変化しないことが分かった．これらの結果から，アクセス傾向の相関を考慮してストレージシステムの階層制御やキャッシュ制御，先読みアルゴリズムの改良を行う可能性を見出した．今後はより長時間の傾向の分析と，特性を利用した実用的なアルゴリズムの提案を行っていく．

謝辞 トレースの取得に際してご協力いただいた富士通研究所の田村雅寿氏，若宮賢二氏に感謝の意を表したい．

参考文献

[1] VMware Infrastructure: VDI Server Sizing and Scaling.
[2] Tamura, M., Iizawa, K., Maeda, M., Kato, J., Kumano, T., Nomura, Y. and Ozawa, T.: Distributed object storage toward storage and usage of packet data in a high-speed network, *Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific*, IEEE, pp. 1–6 (2014).
[3] Fujitsu: Fujitsu Develops Technology Enabling High-Speed Search while Accumulating Data at 40-Gbps, `http://www.fujitsu.com/global/about/resources/news/press-releases/2014/0414-02.html`.
[4] Kavalanekar, S., Worthington, B., Zhang, Q. and Sharda, V.: Characterization of storage workload traces from production windows servers, *Workload Characterization, 2008. IISWC 2008. IEEE International Symposium on*, IEEE, pp. 119–128 (2008).
[5] Atikoglu, B., Xu, Y., Frachtenberg, E., Jiang, S. and Paleczny, M.: Workload analysis of a large-scale key-value store, *ACM SIGMETRICS Performance Evaluation Review*, Vol. 40, No. 1, pp. 53–64 (2012).
[6] Wang, F., Xin, Q., Hong, B., Brandt, S. A., Miller, E. L., Long, D. D. and McLarty, T. T.: File system workload analysis for large scale scientific computing applications, *Proceedings of the 21st IEEE/12th NASA Goddard Conference on Mass Storage Systems and Technologies*, pp. 139–152 (2004).
[7] Carns, P., Harms, K., Allcock, W., Bacon, C., Lang, S., Latham, R. and Ross, R.: Understanding and improving computational science storage access through continuous characterization, *ACM Transactions on Storage (TOS)*, Vol. 7, No. 3, p. 8 (2011).
[8] Leung, A. W., Pasupathy, S., Goodson, G. R. and Miller, E. L.: Measurement and Analysis of Large-Scale Network File System Workloads., *USENIX Annual Technical Conference*, Vol. 1, No. 2, pp. 5–2 (2008).
[9] Gulati, A., Kumar, C. and Ahmad, I.: Storage workload characterization and consolidation in virtualized environments, *Workshop on Virtualization Performance: Analysis, Characterization, and Tools (VPACT)* (2009).
[10] Ling, X., Ibrahim, S., Jin, H., Wu, S. and Tao, S.: Exploiting spatial locality to improve disk efficiency in virtualized environments, *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2013 IEEE 21st International Symposium on*, IEEE, pp. 192–201 (2013).
[11] Ling, X., Ibrahim, S., Wu, S. and Jin, H.-J.: Spatial Locality Aware Disk Scheduling in Virtualized Environment (2014).
[12] Park, J., Kim, Y. and Kim, Y.: Analysis of VDI Workload Characteristics, *The Third International Conference on Digital Enterprise and Information Systems (DEIS2015)*, p. 11 (2015).
[13] Wang, X., Zhang, B. and Luo, Y.: Optimizing interactive performance for desktop-virtualization environment, *Pervasive Computing and the Networked World*, Springer, pp. 541–555 (2013).
[14] Shamma, M., Meyer, D. T., Wires, J., Ivanova, M., Hutchinson, N. C. and Warfield, A.: Capo: Recapitulating Storage for Virtual Desktops., *FAST*, pp. 31–45 (2011).
[15] Li, Z., Chen, Z., Srinivasan, S. M. and Zhou, Y.: C-Miner: Mining Block Correlations in Storage Systems., *FAST*, Vol. 4, pp. 173–186 (2004).
[16] Li, Z., Chen, Z. and Zhou, Y.: Mining block correlations to improve storage performance, *ACM Transactions on Storage (TOS)*, Vol. 1, No. 2, pp. 213–245 (2005).
[17] Seo, B., Kang, S., Choi, J., Cha, J., Won, Y. and Yoon, S.: IO workload characterization revisited: A data-mining approach, *Computers, IEEE Transactions on*, Vol. 63, No. 12, pp. 3026–3038 (2014).
[18] Tarasov, V., Hildebrand, D., Kuenning, G. and Zadok, E.: Virtual machine workloads: the case for new benchmarks for NAS., *FAST*, pp. 307–320 (2013).
[19] Wang, H., Xu, F., Li, Y., Zhang, P. and Jin, D.: Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment, *arXiv preprint arXiv:1510.04026* (2015).
[20] Microsoft: Windows Server 2012 R2 overview, `https://www.microsoft.com/en-us/server-cloud/products/windows-server-2012-r2/`.
[21] Xen: The Xen Project, `http://www.xenproject.org/`.
[22] VMware: Server Virtualization with VMware vSphere, `http://www.vmware.com/products/vsphere/`.
[23] Citrix: XenDesktop VDI Virtual Desktop Infrastructure, `https://www.citrix.com/products/xendesktop/`.
[24] VMware: VDI Virtual Desktop Infrastructure with Horizon 6, `http://www.vmware.com/products/horizon-view/`.
[25] Microsoft: Windows 10 for business, `https://www.microsoft.com/en-us/windowsforbusiness/`.
[26] MacQueen, J. et al.: Some methods for classification and analysis of multivariate observations, *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1, No. 14, Oakland, CA, USA., pp. 281–297 (1967).
[27] Ward Jr, J. H.: Hierarchical grouping to optimize an objective function, *Journal of the American statistical association*, Vol. 58, No. 301, pp. 236–244 (1963).

© 2016 Information Processing Society of Japan