Vol.2014-IOT-27 No.2
Vol.2014-SPT-11 No.2
2014/10/9

データ間の関連性を利用したトラフィックスパイクの波及に対する予防手法の提案とその評価

尾上裕太郎[†1]　王家宏[†1]　児玉英一郎[†1]　高田豊雄[†1]

近年，AmazonDynamo や IIJ GIO などシステムにサーバの追加，削除を行うだけで必要に応じた性能の最適化が可能な動的な分散ストレージサービスが登場してきた．しかし，そのような分散ストレージシステムを用いた場合でも急激なアクセス増加によりシステムの応答性が低下してしまう可能性が指摘されている．そこで，本研究では，データ間の因果関連性や共起関連性という概念を用いることによりアクセスの集中が発生するデータをあらかじめ予測しておき，予測したデータに対するレプリカを別のサーバに配置を行うことで，アクセスの分散を行う手法を提案する．本提案手法の性能を検証するため，性能評価を行い，有効性を確認した．

An Approach to Preventing the Effect of Traffic Spikes and Its Performance Study

YUTARO ONOUE[†1]　JIAHONG WANG[†1]
EIICHIRO KODAMA[†1]　TOYOO TAKATA[†1]

In recent years, dynamic distributed storage services such as Amazon Dynamo and IIJ GIO have been provided, which is characterized by dynamically adding and removing servers according to users' needs to optimize system performance. For these storage systems, however, there is the possibility that system responsiveness decreases suddenly for sudden increase in access traffic. This paper gives an approach to solving the problem. In the proposed approach, the concepts of causal association and cooccurrence association among Web data have been introduced, data that will become system performance bottleneck according to the associations are predicted, replicas of the target data will be distributed to some other servers when the associated data have become hotly accessed, so that traffic loads could be balanced. We have confirmed the effectiveness of the proposed approach by experiments.

1. はじめに

近年，Amazon Dynamo [1]や IIJ GIO [2]など，システムにサーバの追加，削除を行うだけで必要に応じた性能の拡張が可能な分散ストレージサービスが登場してきた．しかし，サーバの増減による性能の拡張だけでは解決できない問題が指摘されている．それは，特定のデータにユーザからのアクセスが集中するようなワークロード下において，アクセスの集中が発生したデータを保持するサーバの応答性が低下してしまうという問題 [3]である．

この問題を解決するための研究として，Intelligent Workload Factoring 機構 [4]や，Adaptive Replication Degree 機構 [5]が提案されている．前者は，アクセス集中によるシステム負荷が一定以上になると，アクセスの集中したデータを自動的にパブリッククラウドに移動させることで，後者は，ユーザからのアクセスをリアルタイムで監視し，ユーザからの急激なアクセス集中が発生した際に，アクセス集中したデータの複製を別のサーバにも配置することでこの問題を解決している．しかし，これらの研究ではアクセス集中（トラフィックスパイク）が発生した後に，それを検知し対処を行うため，対処より先にシステムの応答性能の低下が発生してしまう可能性がある．また，スパイクの発生したデータと関連する，別のデータにまでアクセスの集中が波及する可能性があるが，既存の研究では，この種の発生要因を持つトラフィックスパイクを検知することができない．そこで，本稿では，データマイニングを用い予めアクセス集中が予測されるデータを探知し，探知したデータの複製（レプリカ）を別のサーバにも配置（レプリケーション）することで，トラフィックスパイク発生時におけるサーバの応答性能低下の問題を解決する手法について提案する．また，提案する手法に基づき，システムを実装し，性能評価を行った．本稿では性能評価の結果，及びその考察についても述べる．

2. 関連性によるアクセス集中データの予測

本提案手法は IIJ GIO や Amazon Dynamo 等，Ｗｅｂサーバとして運用するための大量のローカルデータを保有する分散ストレージサービスでの利用を想定している．対象サービスに対し，本手法を適用することで，アクセスによる負荷を分散し，サーバの応答性能低下の問題を解決する．

本提案手法ではトラフィックスパイクが発生したデータと関連性の高いデータにもトラフィックスパイクが発生する可能性が高いと仮定し，対象サービスが保有するストレージプール内のローカルデータ間の関連性を分析する事で，アクセスが集中するデータの予測を行う．本稿ではより詳

†1 岩手県立大学大学院ソフトウェア情報学研究科
Graduate School of Software and Information Science,
Iwate Prefectural University

細に分析をするため，関連性を因果関連性，共起関連性，時間的関連性， 構造的関連性の４つに分類して考える．以下，それぞれの関連性について事例を交えて説明する．

2.1 因果関連性

因果関連性とは「ストレージプール内の特定のデータのアクセスが上昇したら，それが原因となり，関連する他のデータにもアクセスが集中する」という考えに基づく関連性である．以下に，この関連性によりトラフィックスパイクが発生した事例を紹介する．

事例１：大規模な災害の発生がＷｅｂニュースで報道される．それが原因となり，避難情報を提示しているサイトに対してアクセスが集中した．
事例２：ある有名アーティストの死去がＷｅｂニュースで報道される．それが原因となり，アーティストの作品の情報を提供しているサイトに対し，アクセスが集中した．

2.2 共起関連性

共起関連性とは「ストレージプール内の特定のデータへアクセスが集中した際，それと共起し，関連する他のデータへもアクセスが集中する」という考えに基づく関連性である．以下に，この関連性によりトラフィックスパイクが発生した事例を紹介する．

事例１：有名音楽ユニットが解散し，ユニットを構成するメンバー１とメンバー２のブログに対しアクセスが集中した．
事例２：原発問題が社会的ニュースになり，原発について述べている複数のブログ，ニュースサイトに対しアクセスが集中した．

2.3 時間的関連性

時間的関連性とは「特定のデータにアクセス集中が発生した一定時間後に，関連する他のデータにもアクセスが集中する」という考えに基づく関連性である．以下に，この関連性によりトラフィックスパイクが発生した事例を紹介する．

事例１：スマートフォンの新製品が発表され，その一週間後に、各キャリアの料金プランを表示するサイトにアクセスが集中した．
事例２：有名政治家が死去し，その一定時間後に，暗殺説が流布し，その話題を紹介しているＷｅｂサイトにアクセスが集中した．

2.4 構造的関連性

構造的関連性は，上記３つの関連性と異なり，データの構造に基づく関連性である．以下に，この関連性によりトラフィックスパイクが発生した事例を紹介する．

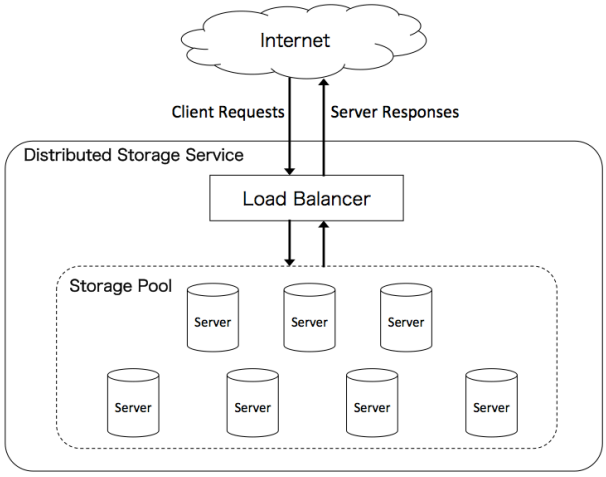

図 1　システム構成図

事例１：スパイクが発生したデータのハイパーリンク先に対して，アクセスが集中した．
事例２：ある企業が掲載した謝罪文にアクセスが集中，その企業のフロントページにもアクセスが集中した．

3. システムモデル

図１に本提案手法のシステム構成図を示す．本システムは，ユーザからのアクセスの解析や，各種関連性の計算などを行うロードバランサと，対象サービスが運用する大量のデータが格納されているストレージプールにより構成されている．ストレージプールはスケーラブルなサーバクラスタにより構成されており，負荷に応じたサーバの増減が可能となっている．

図２にロードバランサの具体的な処理の流れを表したシステムモデルを示す．ロードバランサは，トラフィックスパイク検出を行うトラフィックスパイク検出器①と，どのレプリカを何個配置するか，レプリカをどのサーバに配置するか等を決定するレプリカ配置装置②，クライアントからのアクセスログを保存するアクセス履歴ＤＢ③，データ間の関連性の計算を行う関連性計算機④，計算された関連性を格納する関連性ＤＢ⑤の５つの要素により構成される．

以下，本システムの動作について述べる．まず始めに，関連性によるレプリケーションを行うために必要となる関連性ＤＢの構築方法について述べる．関連性ＤＢの中にはストレージプールに保管されているデータ同士の関連性の情報が格納されている．関連性は，アクセス履歴ＤＢに保存されているアクセスログデータと，ストレージプール内に保管されている対象サービスのコンテンツデータを用い，関連性計算機で計算することにより算出される．算出されたデータ間の関連性データを関連性ＤＢに格納することにより，データベースを構築する．

Vol.2014-IOT-27 No.2
Vol.2014-SPT-11 No.2
2014/10/9

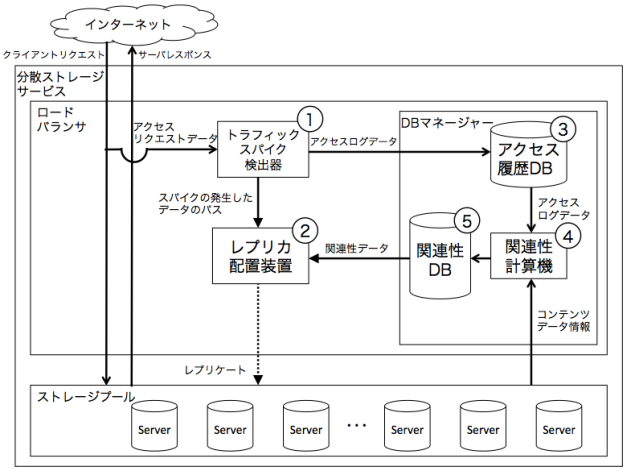

図 2　システムモデル

次に，トラフィックスパイク発生時のシステムの動作について述べる．まず，ユーザからのアクセスリクエストをトラフィックスパイク検出器で監視する．そこで，トラフィックスパイクが観測された際，トラフィックスパイクが発生したデータのパスをレプリカ配置装置に送る．レプリカ配置装置は関連性ＤＢを参照し，トラフィックスパイクが発生したデータと関連性のあるデータを検索する．その後，各データに割り当てるレプリカ数の計算を行う．計算が終了した後，各データのレプリカをストレージプール内のどのサーバに配置するか決定し，トラフィックスパイクが発生したデータと，それと関連性のあるデータに対しレプリケーションを行う．

以上の動作により，トラフィックスパイクの発生したデータと，それに関連するデータのレプリケーションを行い，トラフィックスパイクの発生する危険性の高いデータに対してトラフィックスパイクが発生する前にレプリカを配置することが可能となる．

4. 関連性の抽出手法

２節では，４つの関連性を紹介した．しかし，現在，データ間の関連性を抽出する方法は存在しない．そこで，本節では，データ間の関連性を抽出するために，因果関連性と共起関連性の抽出手法について述べる．

4.1 因果関連性の抽出手法

因果関連性は，アクセス履歴ＤＢに保存されている過去のユーザからのアクセスを解析し，「あるデータへのアクセスが上昇したら，他のデータへのアクセスも上昇する」という相関ルールを発見することで抽出を行う．なお，アクセス履歴ＤＢにはクライアントからシステムへのアクセスログがそのまま格納されている．また，相関ルールの抽出には Apriori アルゴリズム [6]を用いた．因果関連性の抽出アルゴリズムを次に示す．

Algorithm: 因果関連性の抽出アルゴリズム
Input: 閾値S，時間T，アクセス履歴ＤＢ
Output: 因果関連性を関連性ＤＢへ格納

1. アクセス履歴ＤＢを参照し，ストレージプールに保存されている全てのhtmlファイルへのアクセスが何回行われたかをカウントする．
2. 閾値S以上のアクセスが行われたhtmlファイルを抽出する．
3. 抽出したhtmlファイル毎に，該当htmlファイルに対して行われた全てのアクセスから，T時間以内に同一ホストにより行われた別のhtmlファイルへのアクセスを抽出する．
4. 抽出したデータを用い，htmlのファイル名と，T時間以内に行われた別のhtmlファイルへのアクセスを一つのトランザクションとしてまとめ、Aprioriアルゴリズムへの入力データを生成する．
5. 生成した入力データを用い，Aprioriアルゴリズムにより相関ルールの抽出を行う．
6. 抽出した相関ルールをデータ間の因果関連性と定義して関連性ＤＢへ格納する．

上のアルゴリズムでは閾値S以上のデータ対してのみAprioriアルゴリズムでの処理を行うようにしている．これは，平時のアクセスが少ないＷｅｂページはトラフィックスパイクが発生する可能性が低いためであり，また，データマイニングによる処理コストを少なくするためでもある．本提案手法で適用を想定しているサービスではストレージプール内に膨大なデータを保持しており，これら全てのデータに対してデータマイニングの処理を行うのは現実的ではない．

手順３，４では，あるユーザが特定のページにアクセスした後に，別のページへアクセスするというアクセス履歴を基に， Apriori アルゴリズムへの入力データを作成している．そのため，時間 T はユーザがページ間を推移する平均時間を基に設定する．これにより， ユーザのページ推移の情報に基づいた，因果関連性の抽出が可能となる． 手順６では抽出した関連性を関連性ＤＢへと格納している．関連性ＤＢへの格納方法は節５を参照．

4.2 共起関連性の抽出手法

共起関連性はストレージプール内のデータに対し，クラスタリング処理を行う事で抽出する．まず，ストレージプール内に保存されている html ファイルの文章部分に対して，クラスタリング処理を施す．処理の結果，同一クラスタに属するデータ同士を共起関連性のあるものと定義する．なお，クラスタリング処理には k-means 法 [7]を用いた．共起関連性の抽出アルゴリズムを以下に示す．

Algorithm:共起関連性の抽出アルゴリズム
Input:クラスタ数，TF の閾値，ストレージプール内のコンテンツデータ
Output:共起関連性を関連性 DB へ格納

1. ストレージプール内に保存されている全ての html ファイルの文章部分を形態素解析器にかけ，各形態素に分割する．
2. 分割した形態素の中で，固有名詞及び一般名詞からなる単語を抽出する．
3. 抽出された単語について，TF-IDF 法を用い文章全体における頻度を算出し，各文章に対して文章行列を作成する．
4. 文章行列に対し，k-means 法によるクラスタリング処理を施す．
5. クラスタリング処理の結果，同一クラスタに属するデータ同士を共起関連性のあるものと定義し，関連性 DB へ格納する．

上の共起関連性抽出アルゴリズムでは，文章の特徴となる語を抽出するために固有名詞と一般名詞を抽出している．形態素解析器により抽出できる品詞としては固有名詞，一般名詞の他に，形容詞や動詞，代名詞など，様々な種類が存在するが，Ｗｅｂページ固有の特徴を表している単語は一般名詞か固有名詞に多いため，この二つの品詞を抽出する．また，文章の特徴をより高い精度で抽出するため，TF-IDF 法により文章全体における重要な名詞の抽出を行う．TF-IDF 法による処理の後，クラスタリング処理を行うために文章行列を作成する．文章行列とは各文章に対してどの単語が出現しているのかを表した行列であり，この作成した文章行列に対してクラスタリング処理を行う．本提案では大量のデータに対し，クラスタリング処理を施すため，非階層クラスタリングの代表的な手法であり，かつ精度が高いと知られている k-means 法の Hartigan-Wong アルゴリズム [8]を用いた．

この k-means 法によるクラスタリング処理により，類似度の高い文章毎に分類し，同一クラスタに属するデータ同士を関連性ＤＢへ格納する．関連性ＤＢへの格納方法は節５を参照．

5. システム構築について

上では関連性によるレプリカ配置手法を提案したが，システムを構築するためには，これ以外にも，レプリカの配置や関連性データベースの構築などを行う必要がある．

レプリカの配置は，html ファイルと，その中で参照されているコンテンツデータ（画像，動画等）を一つのレプリカの単位とし，このレプリカをコネクション数が最も少ないサーバに割り振ることでアクセスの分散を行う．割り振るレプリカの数については，発生したトラフィックスパイクの規模に応じ，段階的に分けることにより決定する．レプリカのリリースに関しては，リリース自体にもコストがかかる事と，今後も時間をおいてスパイクが発生することを考慮し，能動的に行うことはしない．ただし，後に，別のトラフィックスパイクが発生した際，過去のレプリカデータと現在のレプリカデータを入れ替えることで受動的に削除を行う．

表 1　関連性 DB の例

RID	関連性の種類	データ１のパス	データ２のパス
001	因果関連性	./local/17th_century	./local/18th_century
002	共起関連性	./local/ad_hoc	./local/routers
003	共起関連性	./local/ad_hoc	./local/Wi-Fi
004	因果関連性	./local/2004_in_film	./local/Miracle(film)

関連性ＤＢの構成例を表１に示す．関連性ＤＢは，“レコードＩＤ”，“関連性の種類”，“データ１のパス”，“データ２のパス”を１つのレコードとして持ち，１つのレコードがデータ１とデータ２との関連性を表すものとする．なお，ストレージプール内でデータの更新がある度に，関連性データベースも更新する．

6. 性能評価

本提案手法の有効性を検証するため，前述したシステムモデルに基づき，関連性計算機，関連性ＤＢ，アクセス履歴ＤＢの実装を行い，性能評価を行った．この節では，評価に用いた各種環境，及び性能評価の結果と，その考察ついて述べる．

6.1 評価内容

評価に用いたデータセットは過去にトラフィックスパイクが発生した，マイケルジャクソン死亡時（２００９年６月２４日，２５日）の英語版 Wikipedia のページビューデータ [9]を基に作成した．また，この内，一時間で１０万アクセス以上，かつ，２０分間にアクセスが１０倍に上昇していたＷｅｂページをトラフィックスパイクの発生したページと定義した．２００９年の７月２４日の２１時から２３時の間にトラフィックスパイクの発生していたＷｅｂページのタイトル一覧を表２に示す．トラフィックスパイクが発生していたＷｅｂページ１５件の内，１４件についてはマイケルジャクソンの家族やリリースしたアルバムについてのページであり，スパイクの発生していたページの大部分がマイケルジャクソンと直接関連性のあるものであった．残りの１件の Elvis Presley であるが，これについても，マイケルジャクソン，エルヴィスプレスリー共に有名音楽アーティストという繋がりがあり，間接的な関連性があると言える．

表 2　トラフィックスパイクが発生したページ一覧

番号	Wikipedia ページタイトル
1	Bad (album)
2	Billie Jean
3	Dangerous (album)
4	Elvis Presley
5	HIStory: Past, Present and Future, Book I
6	Invincible (Michael Jackson album)
7	Jackson family
8	Janet Jackson (musician)
9	La Toya Jackson
10	List of best-selling albums worldwide
11	Michael Jackson album descography
12	Off the Wall (album)
13	Randy Jackson
14	The Jackson 5
15	Thriller (album)

次に，評価に用いた閾値について述べる．因果関連性抽出アルゴリズムの閾値Ｓは１００００アクセス/１時間，時間Ｔは１８０秒で，Apriori アルゴリズムの最小支持度，最小確信度は共に０.８に設定し評価を行った．共起関連性抽出アルゴリズムのクラスタ数は８０，初期中心値はランダム，クラスタの繰り返し数は１０，TF の閾値は３に設定し評価を行った．

また，今回の実装では，データマイニングの処理に統計解析ソフトの R [10]を，形態素解析には形態素解析器の TreeTagger [11]を，各種データベースの構築には，データベース管理システムの MySQL [12]を利用した．

6.2 評価結果

各種関連性によりトラフィックスパイクが発生するデータを予測できるかという観点に基づき評価を行った．具体的には，過去にトラフィックスパイクが発生した Michael Jackson のＷｅｂページと関連性の高いページを前述のアルゴリズムにより抽出し，抽出したＷｅｂページに対してトラフィックスパイクが発生していたかどうかの確認を行った．

因果関連性により Michael Jackson のＷｅｂページと関連性があると判断されたページの一覧を表３に示す．２１３６８件中，７件のＷｅｂページに因果関連性があると判断された．また，抽出された７件のＷｅｂページの内，５つのＷｅｂページに対してトラフィックスパイクが発生していた．この結果により，因果関連性の適合率は７１.２％であることが分かった．また，実際にトラフィックスパイクが発生していたページは１５件あり，この内，因果関連性により予測できたＷｅｂページは５件であった．つまり，トラフィックスパイクの発生していた全Ｗｅｂページの内，因果関連性により抽出できた割合は３３.３％であった．

表 3　因果関連性により抽出されたＷｅｂページ一覧

番号	Wikipedia ページタイトル	最小支持度	最小確信度
1	Elvis Presley	0.99601593	1
2	Farrah Fawcett	0.87649403	0.88
3	Janet Jackson	0.85258964	0.88065843
4	Randy Jackson	0.85737051	0.85737051
5	The Beatles	0.82071171	0.84773662
6	The Jackson 5	0.85737051	0.84
7	Thriller (album)	0.81274900	0.83950617

表 4　共起関連性により抽出されたページ（一部抜粋）

番号	Wikipedia ページタイトル
1	Australian Football League
2	Buckmasking
3	Elvis Presly
4	Indie rock
5	Indigo children
6	Jackson family
7	Janet Jackson
8	Janice Dickinson
9	Kyle Reese
10	List of best selling albums worldwide

次に，共起関連性により Michael Jackson のＷｅｂページと関連性があると判断されたページの一部を表４に示す．２１３６８件中，３７件のＷｅｂページに共起関連性があると判断された．この３７件の内，８つのＷｅｂページに対してトラフィックスパイクが発生しおり，共起関連性の適合率は２１.６％であると分かった．また，実際にトラフィックスパイクが発生していた１５件のＷｅｂページの内，８件が抽出されており，トラフィックスパイクの発生していた全データの内，共起関連性により抽出できた割合は５３.３％という結果となった．

6.3 考察

今回の評価では，因果関連性の適合率は７１.２％，共起関連性の適合率は２１.６％という結果となった．また，トラフィックスパイクの発生した全データ１５件の内，予測できた割合（以下，網羅率と呼称）は因果関連性で３３.３％，共起関連性で５３.３％であった．因果関連性の適合率は比較的高い値であり，実際のシステムにおいても適用可能な精度だと考えられる．反面，網羅率は３３.３％とあまり高い値ではなかった．逆に，共起関連性の適合率は２２.２％と非常に低い値となったが，網羅率は５３.３％と因果関連性に比べ高い数値となった．共起関連性の適合率が低くなった要因として，Michel Jackson のＷｅｂページと関連しているページが多数あり，その一部にしかトラフィックスパイクが発生していなかったという点が考えられる．今回，共起関連性により抽出されたＷｅｂページの大多数が Michael Jackson のページと関連するものであったが，

その内，トラフィックスパイクが発生していたページは一部のみであった．関連するＷｅｂページの内トラフィックスパイクの発生していないページは，平時のアクセス頻度もあまり高く無いものが多い．つまりは，Ｗｅｂページとして存在はしていても，ユーザからのアクセスが少なく，トラフィックスパイクの発生する可能性も低いページであると考えられる．今回の共起関連性による関連データの抽出では，因果関連性のようにアクセスの頻度については考慮せず，文章の類似度のみで関連度を測っていたため，関連度は高いがトラフィックスパイクの発生しづらいデータまで抽出してしまった可能性が高い．しかし，トラフィックスパイクの発生したＷｅｂページの内，共起関連性でのみ抽出できたものも多く，この二つの関連性を上手く組み合わせることで，より高い精度での関連データの抽出が期待できると考えられる．

また，今回の評価では，因果関連性と共起関連性で，適合率と網羅率に大きな差があったが，適合率と網羅率の値は，トレードオフの関係にあり，抽出アルゴリズムの閾値の設定により大きく変動する．今回の因果関連性による関連データの抽出では，Apriori アルゴリズムを最小支持度，最小確信度，共に０.８に設定し評価を行ったが，この最小支持度，最小確信度の値を高く設定すればするほど，適合率は高くなる．反面，抽出されるＷｅｂページの数も減り結果，網羅率も下がる．逆に，低く設定すると抽出されるＷｅｂページの数も多くなり再現率は上がるが網羅率は下がる．また，計算コストも高くなる．共起関連性による抽出の場合もクラスタ数の増減により結果が変化する．クラスタ数を多く設定すればするほど，適合率が高くなる傾向があるが，網羅率が下がる．逆に，少なく設定すればするほど，適合率は低くなり網羅率はあがる．これらの点を踏まえ，適切な閾値を設定する必要がある．

また，適合率，網羅率以外の要素においても，閾値による性能の変化は存在する．因果関連性抽出アルゴリズムの閾値 S はデータマイニングの処理を行う範囲を示している．この値を低く設定すると，より多くのデータに対してデータマイニングの処理を行う事になり，結果としてシステム全体の運用コストが高くなってしまう．逆にこの値を高く設定すると，データマイニングの処理を行うデータの量は減るが，反面，トラフィックスパイクの発生したデータを逃してしまう可能性も高くなる．今後は，これらのバランスを満たす事のできる最適な閾値を見つける必要がある．

7. おわりに

本稿では，データ間の関連性を利用し，トラフィックスパイクの波及に対する予防手法の提案と，その評価を行った．評価の結果，ある程度の精度でトラフィックスパイクの発生するデータを予測することができた．よって，この手法を分散ストレージサービスに適用することで，大規模なトラフィックスパイク発生時における、サーバの応答性能の低下の問題の解決が期待できると考えられる．

しかし，今現在運用されているストレージサービスに対し，実際に本提案手法を適用するには課題も存在する．本稿で述べた提案手法では，全てのデータを一元に管理し，その中のデータに対して関連性の計算を行うことでトラフィックスパイクの発生するデータの予測を行っている．このため，事例で挙げたトラフィックスパイクに対応するためには，対象サービスがニュースサイトやブログサイトなど多くの種類のサイトを一元に管理している必要がある．現在ではまだ，単一のクラウド事業者が多くの種類のサイトを管理していることは少ない．しかし，今後クラウドサービスが一般に普及するにつれて，一つの事業者が扱うデータの種類，量は多くなっていくと予想でき，それに伴い本提案も有効に機能するようになると考えられる．

この手法をより実用的なものにするために，今後は，因果関連性及び共起関連性抽出アルゴリズムの適切な閾値の設定方法について検討していく予定である．また，時間的関連性，構造的関連性についての実装，評価や，レプリカの配置速度の評価も行う予定である．

参考文献

1) G. D. Candia, D. Hastorun, M. Jampani, G. Kakulapati, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: Amazons Highly Available Key-Value Store”, In Proc. ACM SIGOPS Symposium on Operating Systems Principles, pp.205–220 (2007).
2) IIJ GIO http://www.iij.ad.jp/GIO/
3) P. Bodik, A. Fox, M. J. Flanklin, M. I. Jordan and D. A. Patterson, “Characterizing, Modeling, and Generating Workload Spikes for Statefull Services”, In Proc. ACM Symposium on Cloud Computing, pp.241–252 (2010).
4) H. Zhang, G. Jiang, K. Yoshihira, H. Chen, and A. Saxena, “Intelligent Workload Factoring for a Hybrid Cloud Computing Model”, In Proc. World Conference on Services, pp.701–708 (2009).
5) 加藤 純, 前田 宗則, 小沢 年弘, “動的なレプリカ数調節によるデータスパイク平準化”, 情報処理学会報告, Vol. 2012-OS-122, No.8, pp.1–10 (2012).
6) R. Agrawal, T. lmielinski, and A. Swami, “Fast algorithms for mining association rules”, In Proc. 20th International Conference on Very Large Data Bases, pp.487–499 (1994).
7) J. MacQueen, “Some Methods for Classifications and Analysis of Multivariate Observations”, In Proc. 5th Berkeley Symposium on Mathematical Statistics and Probabillity, Vol. 1, pp.281–297 (1967).
8) J. A. Hartigan and M. A. Wong, “Algorithm AS 136:A K-Means Clustering Algorithm”, In Proc. Royal Statistical Society Series C, Vol.28, No.1, pp.100 –108 (1979).
9) Page view statistics for Wikimedia projects
http://dumps.wikimedia.org/other/pagecounts-raw/
10) R http://www.r-project.org/
11) TreeTagger
http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
12) MySQL http://www-jp.mysql.com/