構造化P2Pネットワークにおける スケーラブルで正確なデータ集計手法

武田 敦志[1,a)]　生出 拓馬[2]　高橋 晶子[3]　菅沼 拓夫[2]

概要：通信機能を有するセンサ端末やスマート端末が普及したことにより，様々な場所の大量のデータをインターネットを介して取得できるようになった．これらのデータを集計することにより，様々な場所の観測が可能となると考えられている．そこで，本稿では，構造化 P2P ネットワークを介して得られる大量のデータを正確に集計するスケーラブルな仕組みを提案する．提案手法では，構造化 P2P ネットワークのルーティングテーブルを利用することによりスケーラブルなデータ収集を実現し，タイムスロットの概念を用いることにより正確なデータ集計を実現する．本稿では，提案手法のデータ集計手順を述べ，提案手法のスケーラビリティと正確性について説明する．また，提案手法のプロトタイプ実装を用いた実験の結果より，提案手法がスケーラブルなデータ集計手法であること，及び，提案手法が従来手法よりも正確なデータ集計を実現していることを示す．

1. はじめに

通信機能を有するセンサ端末やスマート端末が普及したことにより，様々な場所の大量のデータをインターネットを介して取得できるようになった．これらのデータを活用することにより，広域の消費電力や地域間の人の移動などを観測する情報提供サービスが実現できると考えられる．そこで，我々は，広域に分散したセンサ端末やスマート端末から得られるデータを構造化 P2P ネットワークを用いて管理するネットワークアプリケーションを開発している [1]．図 1 は我々が開発しているネットワークアプリケーションの概念図である．このネットワークアプリケーションは動的負荷分散機能を有する構造化 P2P ネットワークを用いてデータ管理を行うため，従来のクライアントサーバモデルのネットワークアプリケーションに比べてスケーラビリティや耐故障性に優れるという特徴を持つ．そのため，大規模災害等により情報インフラの一部が破壊されたとしても，従来のネットワークアプリケーションよりも安定して情報サービスを提供することが可能となる．

一方，センサ端末やスマート端末から得られる大量のデータを活かした情報サービスを提供するためには，これらの端末から得られるデータを効率よく収集し，それらのデータの合計値や平均値などの集計結果を得るための仕組みが必要不可欠である．現在までにも，構造化 P2P ネットワークによって管理されているデータを効率的に集計するためのデータ集計手法が提案されている [2][3][4][5]．しかし，これらのデータ集計手法では，1 回のデータ集計を行うために必要となる通信データ量が $O(N)$（N は構造化 P2P ネットワーク上のノード数）となるため，データ集計を行うノード数を増やすことが困難であるというスケーラビリティの問題がある．一方，構造化 P2P ネットワークのルーティング情報を用いることにより，スケーラブルなデータ集計を実現する手法も提案されている [6][7]．しかし，これらのデータ集計手法では，異なる時刻に計測されたデータに基づいてデータ集計を行うため，データ集計結果が不正確になるという正確性の問題がある．

本稿では，構造化 P2P ネットワークを介してデータを収集し，それらのデータの合計値，平均値，最大値や最小値を得るためのスケーラブルで正確なデータ集計手法を提案する．提案手法では，スケーラブルにデータを集計するために非同期型データ集計手法を利用し，正確なデータ集計結果を得るためにタイムスロットに基づくデータ集計手法を導入する．提案手法では，構造化 P2P ネットワーク上のそれぞれのノードが独立してデータの収集を行い，収集されたデータをもとに部分的なデータ集計結果を計算する．また，ネットワーク全体のデータ集計結果が必要となった場合，他のノードの持つ部分的なデータ集計結果を収集

[1] 東北学院大学
Tohoku Gakuin University
[2] 東北大学
Tohoku Uniersity
[3] 仙台高等専門学校
Sendai National College of Technology
[a)] takeda@cs.tohoku-gakuin.ac.jp

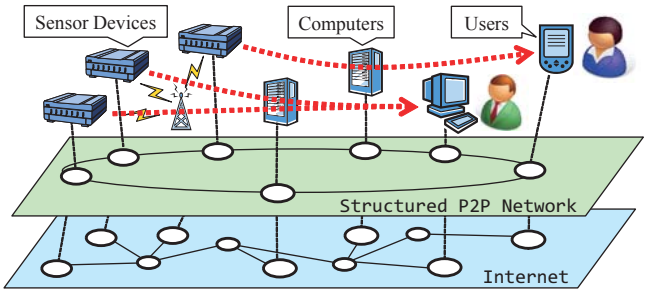

図 1　構造化 P2P ネットワークを用いたアプリケーション
Fig. 1　Network application of structured p2p network.

し，収集した部分的なデータ集計結果を統合することにより全体の集計結果を得る．このように，それぞれのノードが非同期にデータ収集と部分的なデータ集計を実行することにより，データ集計のために各ノード間で送受信される通信データ量は $O(\log N)$（N は構造化 P2P ネットワーク上のノード数）となる．そのため，提案手法はスケーラビリティに優れるという特徴があり，ネットワーク上のすべてのノードが任意のタイミングでネットワーク全体のデータを対象とした集計を行うことが可能となる．また，提案手法では，それぞれのノードは他のノードで計測されたデータの履歴を収集し，同じ時刻に計測されたデータのみを対象としたデータ集計を行うため，各ノードの持つデータが変化したとしても正確なデータ集計結果を得ることができる．

本稿では，提案するデータ集計手法を実現するための詳細な手順を述べ，提案手法がスケーラビリティと正確性に優れていることを説明する．また，提案手法を導入したネットワークアプリケーションの実装について紹介し，このアプリケーションを用いて行った評価実験について述べる．この評価実験では，実装したアプリケーションをコンピュータ上で動作させ，提案手法に必要となる通信データ量と提案手法で得られるデータ集計結果を計測した．これらの実験結果より，提案手法のデータ集計に必要となる通信データ量が $O(\log N)$ であること，及び，提案手法のデータ集計結果が従来手法よりも正確であることを確認した．

本稿は以下の内容で構成される．まず，2 章で関連研究を紹介し，多数のノードからのデータを集計するときの技術的課題を整理する．次に，3 章で提案手法のデータ集計手順について説明する．特に，非同期型データ集計手法とタイムスロットに基づいたデータ集計手法について述べ，提案手法がスケーラビリティと正確性に優れていることを説明する．さらに，4 章では実装したネットワークアプリケーションを用いて行った実験の結果を示し，従来手法に対する提案手法の優位性を検証する．最後に，5 章で本稿で提案するデータ集計手法について考察する．

2. 関連研究

近年，多くの構造化 P2P ネットワークの構成法が提案されている．Chord[8]，Pastry[9]，Tapestry[10] などの構造化 P2P ネットワークは分散ハッシュテーブル（Distributed Hash Table，DHT）と呼ばれる構造化 P2P ネットワークであり，高いスケーラビリティと耐故障性を有するという特徴を持つ．分散ハッシュテーブルでは，それぞれのノードは一方向ハッシュ関数によって得られた値に基づいて仮想的な ID 空間上に配置され，それぞれのノードは仮想的な ID 空間上の位置に基づいて通信相手となるノードや管理対象となるデータを決定する．この手法では，仮想的な ID 空間上のノードの位置を一方向ハッシュ関数を用いて決定するため，仮想的な ID 空間上のそれぞれのノードの位置が均等に分散されることを期待できる．これにより，それぞれのノードが送受信するメッセージも均等に分散されるため，分散ハッシュテーブルはそれ以前の通信ネットワークに比べてスケーラビリティの点で優れている．一方，分散ハッシュテーブルでは，仮想的な ID 空間上のノードやデータの位置を一方向ハッシュ関数を用いて決定するため，範囲検索などの指定された条件を満たすデータのみを対象とした検索を実行することは難しい．そのため，分散ハッシュテーブルでは，集計対象となるデータのみを収集し，それらのデータの集計結果を得るためのスケーラブルな仕組みを実現することは困難である．

ネットワークアプリケーションがネットワーク全体の変化に対応しなくてはならない場合，ネットワーク全体の状況を把握する必要があるため，ネットワーク全体のデータを集計する機能が必要不可欠となる．構造化 P2P ネットワーク全体のデータの集計結果を得るための手法として，各ノードが構造化 P2P ネットワークの一部のノードからデータを収集し，その収集した一部のデータからネットワーク全体のデータ集計結果を推測する手法が提案されている [11]．しかし，通常の構造化 P2P ネットワークでは，仮想的な ID 空間上のノード位置の分散が完全な均一とはならないため，この手法を用いてネットワーク全体のデータの合計値や平均値などの集計結果を正確に推測することは困難である．一方，データ集計を可能とする範囲検索機能を有する構造化 P2P ネットワークも提案されている [12][13][14][15]．これらの構造化 P2P ネットワークでは，検索条件を満たす複数のデータを分散ハッシュテーブルよりも効率的に収集することが可能である．しかし，集計対象となるデータを持つすべてのノードから個別にデータの収集を行う必要があるため，ネットワーク全体のデータ集計に必要となる通信データ量や計算時間が $O(N)$（N はネットワーク上のノード数）となり，構造化 P2P ネットワーク全体のデータ集計を行うための仕組みとしてはス

ケーラビリティの点に問題がある．

ネットワーク上のそれぞれのノードが個別に部分的な集計を行うことにより，$O(\log N)$ の計算時間でデータ集計を行うことができる構造化 P2P ネットワークも提案されている [2][3][4][5][6][7]．これらの構造化 P2P ネットワークでは，それぞれのノードはネットワーク上の一部のノードからデータを収集し，その収集したデータに基づいた部分的なデータ集計結果を計算する．また，あるノードがネットワーク全体のデータ集計結果を必要としている場合，そのノードは他のノードより部分的なデータ集計結果を収集し，その収集した部分的なデータ集計結果を統合することによりネットワーク全体のデータ集計結果を得る．これらのデータ集計手法では，仮想的な ID 空間上のノード位置に基づいたツリー状のネットワークを構成し，このネットワークに基づいてデータ収集や部分的なデータ集計結果の計算を行うため，データ集計に必要となる計算時間は $O(\log N)$ となる．これらのデータ集計手法は大きく 2 つのタイプに分類することができる．本稿では，これらのデータ集計手法のタイプを「同期型データ集計手法」と「非同期型データ集計手法」と呼ぶ．同期型データ集計手法では，部分的なデータ集計とネットワーク全体のデータ集計を同期して行う [2][3][4][5]．すべてのデータ集計処理を同時に行うため，正確なデータ集計結果を得ることが可能となる．一方，1 回のデータ集計に必要となる通信データ量は $O(N)$ であるため，全体のデータ集計を行うノード数が限定されるというスケーラビリティの問題がある．この手法に対し，非同期型データ集計手法では，部分的なデータ集計とネットワーク全体のデータ集計を非同期に行う [6][7]．各ノードは定期的に部分的なデータ集計結果を更新しており，1 回の部分的なデータ集計結果の更新のために必要となる通信データ量は $O(\log N)$ である．また，1 回の全体の集計結果の計算のために必要となる通信データ量も $O(\log N)$ となる．そのため，非同期型データ集計手法では，ネットワーク上のすべてのノードが任意の範囲のデータを集計することが可能であり，同期型データ集計手法に対してスケーラビリティに優れたデータ集計手法であるといえる．一方，従来の非同期型データ集計手法では，データの計測時刻を考慮せずにデータ集計を行っているため，得られるネットワーク全体のデータ集計結果の正確性に問題がある．

我々は現在までに非同期型データ集計手法の研究開発を進めており [7]，本稿で提案するデータ集計手法の基本的な考え方も非同期型データ集計手法と同様である．しかし，従来の非同期型データ集計手法は SkipGraph や Chord#などの特定の構造化 P2P ネットワークを対象として設計されていたのに対し，本稿で提案するデータ集計手法は一次元の仮想的 ID 空間を用いるすべての構造化 P2P ネットワークを対象としている．また，提案手法では，タイムスロットに基づくデータ集計の仕組みを導入してデータの計測時刻を考慮したデータ集計を行うことにより，従来手法よりも正確なデータ集計結果を得ることが可能となる．

3. スケーラブルで正確なデータ集計手法

3.1 非同期型データ集計手法

構造化 P2P ネットワーク上でネットワーク全体のデータ集計を行うためには，そのネットワーク上のすべてのノードが持つデータを収集する必要がある．しかし，データを持つすべてのノードから個別にデータの取得を行った場合，計算のために必要となる時間が $O(N)$（N はネットワーク上のノード数）となる問題がある．そこで，提案手法では，それぞれのノードがネットワークの一部のノードが持つデータを収集し，これらのデータに基づいた部分的なデータ集計を行う．また，あるノードがネットワーク全体のデータ集計結果を必要とする場合，そのノードは他のノードが持つ部分的なデータ集計結果を収集し，収集した部分的なデータ集計結果を統合することによりネットワーク全体のデータ集計結果を得る．この仕組を用いることにより，ネットワーク全体のデータ集計を行うために必要となる時間は $O(\log N)$ となる．また，提案手法では，部分的なデータ集計とネットワーク全体のデータ集計は非同期に行われる．そのため，本稿では，提案手法に用いるこのデータ集計手法を「非同期型データ集計手法」と呼ぶ．この非同期型データ集計手法は，全体のデータ集計に必要となる通信データ量が $O(\log N)$ であるため，スケーラビリティに優れているという特徴を持つ．

本稿で提案するデータ集計手法では，すべてのノードは以下の属性を有する．

$$
\begin{aligned}
\text{node} &:= < \text{id}, \text{value}, \text{routes} > \\
\text{id} &:= \text{INTEGER} \\
\text{value} &:= \text{INTEGER} \\
\text{routes} &:= \{\ \text{route}_0, \text{route}_1, \text{route}_2, \cdots \} \\
\text{route}_i &:= < \text{node}_i, \text{value}_i >
\end{aligned}
$$

ここで，*id* はそのノードの ID，*value* はそのノードが持つデータ，*routes* はそのノードのルーティングテーブルを示している．また，ルーティングテーブルは転送先となるノード $node_i$，及び，部分的なデータ集計結果 $value_i$ のリストとして構成されている．構造化 P2P ネットワークでは，すべてのノードはそのノードの ID に基づいて仮想的な ID 空間に配置され，その仮想的な ID 空間上の位置に基づいてルーティングテーブルを作成する．あるノードがメッセージの送信や転送を行う場合，そのノードはルーティングテーブルから送信先のノードを選び，そのノードに対してメッセージを送信する．この動作をすべてのノードが行うことにより，メッセージは目的ノードに到達する．

本稿で提案するデータ集計手法では，それぞれのノードは部分的なデータ集計結果 $value_i$ をルーティングテーブ

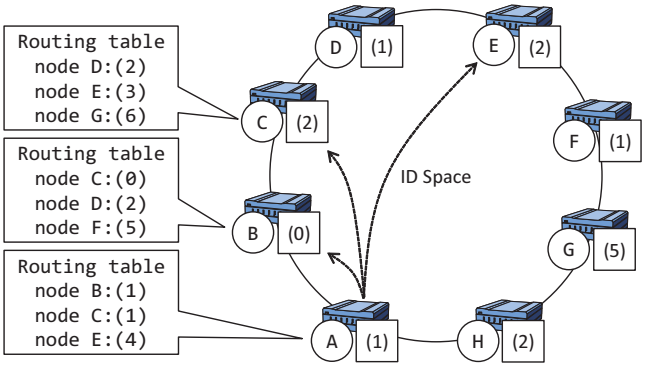

図 2　提案手法を導入した構造化 P2P ネットワーク
Fig. 2　Structured p2p network with proposed mechanism.

ルの転送先ノード $node_i$ と関連付けて管理する．ここで，この部分的なデータ集計結果 $value_i$ は，仮想的な ID 空間上で $[id, node_i.id)$ の範囲に存在するノードが持つデータの集計結果となる．図 2 は本稿で提案するデータ集計手法を導入した構造化 P2P ネットワークの例である．この図の構造化 P2P ネットワークには A から H までのノードが参加しており，それぞれのノードは括弧内に書かれたデータ（数値）を持っている．ここで，ノード A は，ノード B，C，E を転送先ノードとするルーティングテーブルを持っている．また，このルーティングテーブルには，部分的なデータ集計結果が転送先ノードと関連付けて登録されている．ノード A のルーティングテーブルに登録されている部分的なデータ集計結果は以下の通りとなる．

- 転送先ノード B と関連付けられた範囲 [ノード A, ノード B) の部分的なデータ集計結果
- 転送先ノード C と関連付けられた範囲 [ノード A, ノード C) の部分的なデータ集計結果
- 転送先ノード E と関連付けられた範囲 [ノード A, ノード E) の部分的なデータ集計結果

図 2 では，上記の部分的なデータ集計結果をルーティングテーブルの括弧内のデータ（数値）として示す．

提案手法では，ノードが $[id, dst.id)$ の範囲の部分的なデータ集計を行う場合，そのノードは $dst.id$ に対して検索メッセージを送信する．この検索メッセージを受信したノードがこのメッセージを転送するとき，そのノードはルーティングテーブル内の転送先ノードに関連付けられている部分的なデータ集計結果を転送する検索メッセージに追加する．検索メッセージを転送するすべてのノードが上記の動作を行うことにより，検索メッセージは $[id, dst.id)$ の範囲の部分的なデータ集計を行うために必要となるすべてのデータを得ることができる．ここで，構造化 P2P ネットワークで検索メッセージの転送に必要となる通信データ量は $O(\log N)$（N はネットワーク上のノード数）であるため，1 個の部分的なデータ集計結果を得るために必要となる通信データ量も $O(\log N)$ となり，部分的な集計結果を含むルーティングテーブルを更新するために必要となる通信データ量は $O((logN)^2)$ となる．ただし，構造化 P2P ネットワークのネットワーク構造が均一であった場合，1 個の部分的なデータ集計結果を得るために必要となる通信データ量は $O(1)$ に近づく．そのため，ネットワーク構造の偏りを減らす仕組みを持つ多くの構造化 P2P ネットワークの場合，部分的なデータ集計結果を含むルーティングテーブルを更新するための通信データ量は $O(\log N)$ と同等とみなすことができる．

```
01:  // update routing nodes and aggregated values
02:  n.updateRoutingTable ( ) {
03:    value = n.value;
04:    node = n.successor;
05:    c_distance = distance(node);
06:    p_distance = 0;
07:    for (i = 0; p_distance < c_distance; i = i + 1) {
08:      n.routes[i].value = value;
09:      n.routes[i].node = node;
10:      value = aggregate(value, node.routes[i].value);
11:      node = node.routes[i].node;
12:      p_distance = c_distance;
13:      c_distance = distance(node);
14:    }
15:  }
```

図 3　Chord#におけるルーティングテーブル更新処理の擬似コード
Fig. 3　Pseudo-code for routing table updating of Chord#.

図 3 に，構造化 P2P ネットワークの 1 つである Chord#[12] に提案手法を導入したときのルーティングテーブル更新手順の擬似コードを示す．ここで，*distance* は指定されたノードまでの仮想的な ID 空間における距離を計算する関数であり，*aggregate* は指定された 2 つのデータを集計する関数である．各ノードは，自身のルーティングテーブルを更新するとき，そのルーティングテーブルの転送先ノードに関連付けられている部分的なデータ集計結果も更新する．Chord#などのノード間のホップ数に基づいてルーティングテーブルを作成する構造化 P2P ネットワークの場合，ルーティングテーブルに登録されてる 1 個の部分的なデータ集計結果を更新するためには転送先のノードから得られる 1 個の部分的なデータ集計結果のみを必要とする．そのため，Chord#や Waon[15] などのホップ数に基づくルーティングテーブルを用いる構造化 P2P ネットワークの場合，部分的なデータ集計結果を含むルーティングテーブルを更新するための通信データ量は $O(\log N)$ となる．

ノードが構造化 P2P ネットワーク全体のデータ集計を行う場合，そのノードは他のノードが計算した部分的なデータ集計結果を収集し，収集した部分的なデータ集計結果を統合することにより全体のデータ集計結果を得る．このとき，上述した部分的なデータ集計結果に必要なデータ

```
01: // get an aggregation result between node.id and id
02: n.getAggregation ( id ) {
03:   if ( n.isResponsible(id) ) {
04:     return n.value;
05:   }
06:   else {
07:     route = n.getRoute( id );
08:     ret = route.getAggregation( id );
09:     return aggregate( route.value, ret );
10:   }
11: }
```

図 4　データ集計手順の擬似コード
Fig. 4　Pseudo-code for calculating data aggregation.

を集める手順と同様に，構造化 P2P ネットワークの検索メッセージを用いてネットワーク全体のデータ集計に必要となる部分的なデータ集計結果を収集する．まず，全体のデータ集計結果を必要とするノードは，そのノード自身の ID を宛先とした検索メッセージを送信する．このメッセージを転送するノードは，転送先ノードに関連付けられた部分的なデータ集計結果を検索メッセージに登録する．すべてのノードが上記動作を行うことにより，この検索メッセージにはネットワーク全体なデータ集計を行うために必要となるすべての部分的なデータ集計結果が登録される．図 4 に提案手法におけるデータ集計手順の擬似コードを示す．ここで，*n.getRoute* は指定された宛先 *id* へのメッセージを送信ための転送先ノードを返す関数であり，*route.getAggregation* は検索メッセージを送信して部分的なデータ集計結果を得る関数である．この集計手順を用いることにより，仮想的な ID 空間上でノード自身の ID から指定された ID までの範囲にあるデータを対象としたデータ集計を行うことが可能である．また，対象範囲の終点 *id* にノード自身の ID を指定することでネットワーク全体のデータ集計結果を得ることができる．提案手法では，1 個の検索メッセージを利用してデータ集計のためのデータを収集するため，ネットワーク全体なデータ集計を行うために必要となる通信データ量は $O(\log N)$（N はネットワーク上のノード数）となる．これより，本稿で提案するデータ集計手法はスケーラビリティに優れているといえる．

3.2　タイムスロットに基づくデータ集計手法

3.1 で述べたとおり，非同期型データ集計手法はスケーラビリティに優れたデータ集計手法である．しかし，非同期型データ集計手法では部分的なデータ集計と全体的なデータ集計を非同期で行っており，集計対象となるデータの計測時刻を考慮していないため，集計対象となるデータが刻々と変化する状況では正確な集計結果を得ることが難しいという問題がある．そこで，本稿では，時間とともにデータが変化する状況においても正確なデータ集計を可能とするタイムスロットに基づくデータ集計手法を提案する．

本稿では，各ノードが持つデータの値は刻々と変化するものと想定する．そこで，時刻 t のときにノード i が持つデータの履歴 $\boldsymbol{V}_i(t)$ を以下のように定義する．

$$\boldsymbol{V}_i(t) = \{v_i(t), v_i(t-1), \cdots\}.$$

ここで，$v_i(t)$ は時刻 t のときにノード i が持つデータの値である．従来の非同期型データ集計手法では，ノード i が持つデータ v_i とノード j が持つデータ v_j の集計結果 v_{ij} は以下のように計算される．

$$v_{ij}(t) = aggregate(v_i(t - \Delta t_i), v_j(t - \Delta t_j)).$$

ここで，*aggregate* は集計処理を行う関数である．また，Δt はデータの計測時刻とデータの集計時刻との時間差である．従来の非同期型データ集計手法では時刻 t，$t - \Delta t_i$，$t - \Delta t_j$ がそれぞれ異なっているため，ノードが持つデータが刻々と変化した場合に正確なデータ集計を行うことが困難であった．

データ集計の正確性の問題を解決するため，本稿では，タイムスロットに基づいてデータ集計を行う非同期型データ集計手法を提案する．提案手法では，それぞれのノードはデータの履歴を管理し，このデータの履歴をタイムスロットに合わせて集計することにより，従来手法よりも正確なデータ集計結果を得る．提案手法では，データ集計結果は $\boldsymbol{V}_{ij}(t)$ は以下のように計算される．

$$\begin{aligned}\boldsymbol{V}_{ij}(t) &= aggregate(\boldsymbol{V}_i(t), \boldsymbol{V}_j(t)\} \\ &= \{v_{ij}(t), v_{ij}(t-1), \cdots\} \\ v_{ij}(t) &= aggregate(v_i(t), v_j(t))\end{aligned}$$

ここで，$v_i(t)$ と $v_j(t)$ は同じ時刻 t に計測されたデータである．そのため，データの履歴 $\boldsymbol{V}_i(t)$ 及び $\boldsymbol{V}_j(t)$ が十分に長い場合，提案手法によるデータ集計結果 $\boldsymbol{V}_{ij}(t)$ は従来手法によるデータ集計結果よりも正確となる．通常，計測直後のデータを非同期に収集することは難しいため，最も新しいデータに基づく集計結果 $v_{ij}(t)$ は不正確である．しかし，より古いデータに基づく集計結果はより正確なものになる．具体的には，t_{update} をルーティングテーブルの更新間隔とし，N をネットワーク上のノード数とすると，データ集計結果 $v_{ij}(t - t_{update} \log_2 N)$ は十分に正確である．

タイムスロットに基づくデータ集計手法では，データ集計結果を得るために送受信するメッセージ数は $O(\log N)$（N はネットワーク上のノード数）となる．さらに，正確なデータ集計結果を得るために必要となるデータ履歴の長さも $O(\log N)$ となる．そのため，このデータ集計手法では，データ集計結果を得るために必要となるデータの大きさは $O((\log N)^2)$ である．しかし，通常のデータ集計で必要となるデータ履歴は検索メッセージの転送に必要なデータ（宛先や制御データ）に比べて小さいため，データ履歴

の大きさが通信データ量に与える影響は限定的である．そのため，提案手法に必要となる通信データ量は $O(\log N)$ と同等とみなすことができ，タイムスロットに基づくデータ集計手法は十分にスケーラブルであるといえる．また，タイムスロットに基づくデータ集計手法では，従来の非同期型データ集計手法よりも正確なデータ集計結果を得ることができる．

4. 実験・評価

4.1 実装

本稿で提案するデータ集計手法を評価するため，構造化 P2P ネットワーク Chord#[12] を介してデータ集計を行うネットワークアプリケーションを実装した．このアプリケーションには本稿で提案した非同期型データ集計手法とタイムスロットに基づくデータ集計手法を導入しており，それぞれのノードはこれらのデータ集計手法に従ってネットワーク全体のデータ集計を行う．実装したアプリケーションでは，データの集計や構造化 P2P ネットワークの維持を行うため，それぞれのノードがお互いにコンピュータネットワークを介した UDP 通信を行う．そのため，実装したアプリケーションが動作することにより，コンピュータネットワーク上に通信データが発生する．また，提案手法を従来手法と比較するため，従来のデータ集計手法でネットワーク全体のデータ集計を行うネットワークアプリケーションも実装した．

本実験では，複数のアプリケーションを 1 台のコンピュータ上で動作させることにより，本稿で提案したデータ集計手法の動作や性能を検証した．本実験で用いたコンピュータの仕様を表 1 に示す．本実験では，すべてのノードを 1 台のコンピュータ上で動作させたが，それぞれのノードは別のスレッドとして実行されており，それぞれのノードが持つデータ（計測データ，ルーティングテーブルなど）の記憶領域は完全に分離されている．すべてのノードは 30 秒ごとにルーティングテーブルの更新処理を実行し，すべてのノードが 30 秒ごとにネットワーク全体のデータ集計を実行する．このとき，データの集計や構造化 P2P ネットワークの維持に必要となるノード間の通信はループバックインタフェースを介して行われる．そのため，ノード間の通信に発生する遅延は無視できるほどに小さい．一方，ノード間の通信データ量がループバックインタフェースの許容量を越えて増加した場合にはパケットの損失が発生する．上記の実験環境において実装したアプリケーションを動作させ，ネットワーク上に発生する通信データ量とノードによって計算されるデータ集計結果を計測した．提案手法と従来手法の比較を行うため，本実験では以下の実装を用いて実験を行った．

- Synchronous Data Aggregation
  同期型データ集計手法
  （従来手法）
- Asynchronous Data Aggregation without time-slot
  タイムスロットを用いない非同期型データ集計手法
  （従来手法）
- Asynchronous Data Aggregation with time-slot
  タイムスロットを用いる非同期型データ集計手法
  （提案手法）

この章では，上記実装を用いた実験の結果を示し，提案手法と従来手法との比較評価について述べる．

表 1　実験で用いたコンピュータの仕様
Table 1　Computer specifications for experiments.

CPU	AMD Opteron 4176 × 2
Memory	48 GByte
OS	FreeBSD 9.2
Runtime Environment	OpenJDK Runtime Environment 1.7

4.2 通信データ量の評価

図 5 に 1 秒間に各ノードが送受信した通信データ量の平均値を示す．図 5 の通信データ量には，データ集計処理に必要となる通信データ量だけではなく，構造化 P2P ネットワークを維持するために必要となる通信データ量も含まれる．従来手法である同期型データ集計手法では，ネットワーク全体のデータ集計を行うために必要となる通信データ量が $O(N)$（N はネットワーク上のノード数）となるため，ノード数の増加に比例して必要となる通信データ量も増加する．しかし，非同期型データ集計手法では，ネットワーク全体のデータ集計を行うために必要となる通信データ量は $O(\log N)$ である．そのため，非同期型データ集計手法に必要となる通信データ量は同期型データ集計手法に必要となる通信データ量よりも少ない．

提案手法であるタイムスロットに基づく非同期型データ集計手法ではそれぞれのノードのデータの履歴を用いてデータ集計を行う．そのため，提案手法に必要となる通信データ量は従来の非同期型データ集計手法に必要となる通信データ量よりも多い．ただし，図 5 より，提案手法の通信データ量と従来の非同期型データ集計手法の通信データ量の差は小さく，提案手法が十分にスケーラブルであることがわかる．

4.3 データ集計結果の正確性の評価

本実験では，実装したアプリケーションを 500 個動作させることで 500 個のノードによって構成される構造化 P2P ネットワークを作成し，1 個のノードによって計測されるネットワーク全体のデータ集計結果を計測した．ここでのデータ集計では，各ノードが持つデータの合計値を集計している．実験開始の時点では，すべてのノードは値「0」の

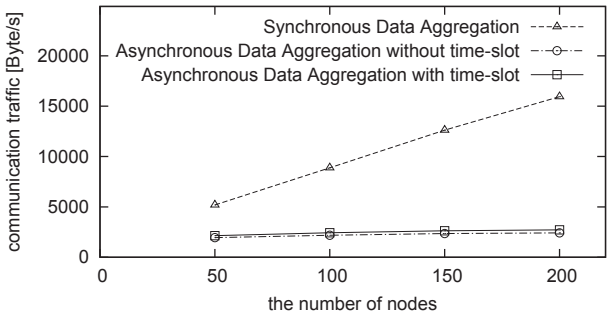

図 5　通信データ量の変化
Fig. 5　Amount of communication data.

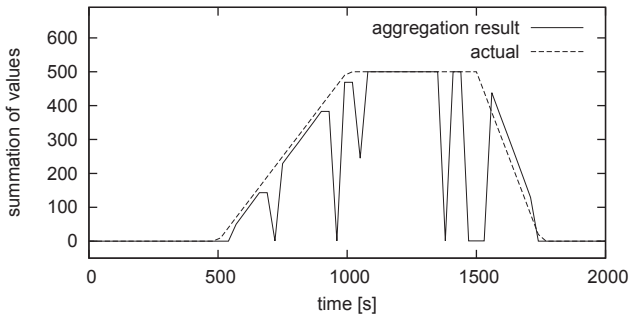

図 6　同期型データ集計手法によるデータ集計結果
Fig. 6　Result of Synchronous Data Aggregation.

データを持っている．また，実験開始から 500 秒を経過すると，1 秒につき 1 個のノードが持つデータが「0」から「1」へと変化する．さらに，実験開始から 1000 秒を経過すると，1 秒につき 2 個のノードが持つデータが「1」から「0」へと変化する．

図 6 に同期型データ集計手法によるネットワーク全体のデータ集計結果を示す．同期型データ集計手法では，ネットワーク全体のデータ集計処理に同期して部分的なデータ集計処理も実行されるため，ネットワーク全体のデータ集計結果は正確である．しかし，通信帯域などのネットワーク資源の不足によりパケット損失が発生する場合，データ集計処理に必要となるデータを収集できないため，ネットワーク全体のデータ集計結果は不正確となる．

図 7 に従来の非同期型データ集計手法によるネットワーク全体データ集計結果を示す．非同期型データ集計手法では，それぞれのノードが行う部分的なデータ集計とネットワーク全体のデータ集計は非同期で実行される．この仕組みにより，非同期型データ集計手法に必要となる通信データ量は同期型データ集計手法に必要となる通信データ量より少ないため，ネットワーク資源の不足によるパケット損失は発生しにくい．そのため，それぞれのノードが持つデータが変化しない場合，非同期型データ集計手法によって得られるネットワーク全体のデータ集計結果は同期型データ集計手法のデータ集計結果よりも正確である．しかし，従来の非同期型データ集計手法ではデータの観測時刻

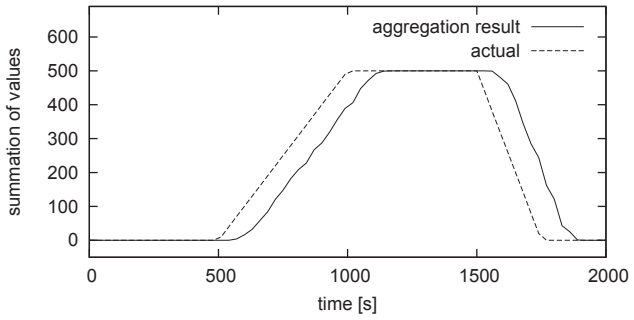

図 7　従来の非同期型データ集計手法によるデータ集計結果
Fig. 7　Result of Asynchronous Data Aggregation without a time-slot idea.

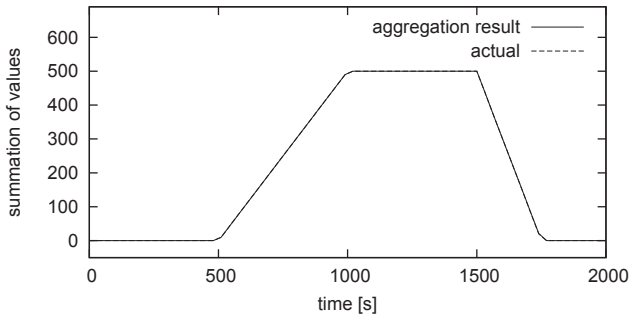

図 8　タイムスロットに基づく非同期型データ集計手法によるデータ集計結果
Fig. 8　Result of Asynchronous Data Aggregation with a time-slot idea.

を考慮せずにデータ集計を行うため，それぞれのノードが持つデータが時間とともに変化した場合，正確なデータ集計結果を得ることはできない．

図 8 に提案手法によるネットワーク全体のデータ集計結果を示す．上述のとおり，ノードが持つデータが時間とともに変化した場合，従来の非同期型データ集計手法では正確なデータ集計結果を得ることが困難であった．この問題を解決するため，提案手法ではタイムスロットに基づくデータ集計手法を導入し，それぞれのデータの計測時刻を考慮したデータ集計処理を行う．そのため，提案手法では，それぞれのノードが持つデータが時間とともに変化したとしても，正確なデータ集計結果を得ることができる．一方，提案手法では，最新のデータを元にしたデータ集計結果を得ることは難しい．そのため，提案手法によって得られるデータ集計結果のリアルタイム性は低い．

4.4　議論

表 2 に本稿で提案するデータ集計手法と従来手法との比較を示す．同期型データ集計手法では，それぞれのノードが行う部分的なデータ集計処理はネットワーク全体のデータ集計処理と同期して実行される．このデータ集計手法では最新のデータを用いて集計処理を行うため，データ集計

表 2　提案手法と従来手法との比較
Table 2　Comparison of data aggregation mechanisms.

	スケーラビリティ	集計結果の正確性	リアルタイム性
同期型データ集計手法（従来手法）	低い	高い (*)	高い
従来の非同期型データ集計手法（従来手法）	高い	低い	低い
タイムスロットに基づく非同期型データ集計手法（提案手法）	高い	高い	低い

(*) ネットワーク資源（メモリ，通信帯域など）が十分である場合

結果のリアルタイム性に優れている．さらに，ノード間の通信が十分に行われている場合，ネットワーク全体のデータ集計結果の正確性は高い．一方，データ集計処理に必要となる通信データ量は $O(N)$（N はネットワーク上のノード数）であるため，同期型データ集計手法はスケーラビリティの面に問題がある．

非同期型データ集計手法では，それぞれのノードが行う部分的なデータ集計処理はネットワーク全体のデータ集計処理から独立して実行される．それぞれのデータ集計処理に必要となる通信データ量は $O(\log N)$ であるため，非同期型データ集計手法は同期型データ集計手法に比べてスケーラビリティの点で優れている．しかし，従来の非同期型データ集計手法では，データの計測時刻を考慮せずにデータ集計処理を行うため，各ノードが持つデータの値が時間とともに変化する場合にはデータ集計結果が不正確になるという問題がある．一方，本稿で提案する非同期型データ集計手法では，タイムスロットの概念を用いることにより，データの計測時刻を考慮したデータ集計を行う．そのため，従来の非同期型データ集計手法と比較して，提案手法によって得られるネットワーク全体のデータ集計結果の正確性は高い．しかし，非同期型データ集計手法では最新のデータを収集することは難しく，計測されたデータが集計結果に反映されるまでに時間を要するため，データ集計結果のリアルタイム性は低い．

同期型データ集計手法はリアルタイム性に優れており，非同期型データ集計手法はスケーラビリティに優れている．この比較より，センサ端末を用いた状況記録システムなどのスケーラビリティが重要となるネットワークアプリケーションには提案手法である非同期型データ集計手法が適している．一方，センサ端末を用いた警告システムなどのリアルタイム性が重要となるネットワークアプリケーションには同期型データ集計手法が適している．

5. おわりに

本稿では，構造化 P2P ネットワークを介してネットワーク全体のデータ集計を行うスケーラブルで正確なデータ集計手法を提案した．提案手法では，タイムスロットに基づくデータ集計手法を非同期型データ集計手法に導入することにより，スケーラビリティの問題とデータ集計結果の正確性の問題を解決している．本稿では，提案手法に必要となる通信データ量は $O(\log N)$（N はネットワーク上のノード数）であり，提案手法が高いスケーラビリティを有していることを述べた．また，提案したデータ集計手法では，それぞれのノードが持つデータが時間とともに変化したとしても正確なデータ集計結果を得られることを説明した．さらに，提案手法を導入した構造化 P2P ネットワークのアプリケーションを実装し，この実装を用いた実験結果より，提案手法が従来手法に比べてスケーラビリティとデータ集計結果の正確性の点で優れていることを示した．

一方，多くのノードがネットワークから一度に離脱した場合，本稿で提案したデータ集計手法では正確な集計結果を得ることができない．そのため，ノードの参加や離脱が頻繁に発生する状況でも正確なデータ集計結果を得られるようなデータ集計手法の実現が今後の課題として挙げられる．また，本稿では合計や平均などの単純なデータ集計を対象として提案手法の評価を行ったが，分散や偏差などのより複雑などの複雑な集計の場合はノード間の通信回数が増加することが予想される．今後はより複雑なデータ集計への対応とその評価を行う予定である．

Acknowledgment

本研究は文部科学省科学研究費補助金若手研究（B）（25730064）の助成を受けて実施したものである．

参考文献

[1] Oide, T., Takahashi, A., Takeda, A. and Suganuma, T.: A Robust P2P Information Sharing System and its Application to Communication Support in Natural Disasters, *International Journal of Software Science and Computational Intelligence*, Vol. 5, No. 4, pp. 20–39 (2013).

[2] González-Beltrán, A., , Milligan, P. and Sage, P.: Range queries over skip tree graphs, *Computer Communications*, Vol. 31, No. 2, pp. 358–374 (2008).

[3] Cai, M. and Hwang, K.: Distributed Aggregation Algorithms with Load-Balancing for Scalable Grid Resource Monitoring, *Proceedings of the 21th International Parallel and Distributed Processing Symposium (IPDPS 2007)*, pp. 1–10 (2007).

[4] Graffi, K., Stingl, D., Rueckert, J., Kovacevic, A. and Steinmetz, R.: Monitoring and Management of Structured Peer-to-Peer Systems, *Proceedings of the 9th International Conference on Peer-to-Peer Computing (P2P '09)*, pp. 311–320 (2009).

[5] Schulz, S., Blochinger, W. and Hannak, H.: Capability-Aware Information Aggregation in Peer-to-Peer Grids,

Journal of Grid Computing, Vol. 7, No. 2, pp. 135–167 (2009).

[6] Abe, K., Abe, T., Ueda, T., Ishibashi, H. and Matsuura, T.: Aggregation Skip Graph: A Skip Graph Extension for Efficient Aggregation Query over P2P Networks, *International Journal On Advances in Internet Technology*, Vol. 4, No. 3, pp. 103–110 (2012).

[7] Takeda, A., Oide, T. and Takahashi, A.: A Structured Overlay Network for Aggregating Sensor Data, *Proceedings of the 7th International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA2012)*, pp. 684–689 (2012).

[8] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F. and Balakrishnan, H.: Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications, *IEEE/ACM Transactions on Networking*, Vol. 11, No. 1, pp. 17–32 (2003).

[9] Rowstron, A. and Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, pp. 329–350 (2001).

[10] Zhao, B. Y., Huang, L., Stribling, J., Rhea, S. C., Joseph, A. D. and Kubiatowicz, J. D.: Tapestry: A Resilient Global-scale Overlay for Service Deployment, *IEEE Journal on Selected Areas in Communications*, Vol. 22, No. 1, pp. 41–53 (2004).

[11] Shafaat, T. M., Ghodsi, A. and Haridi, S.: A Practical Approach to Network Size Estimation for Structured Overlays, *Lecture Notes in Computer Science*, Vol. 5343, pp. 71–83 (2008).

[12] Schütt, T., Schintke, F. and Reinefeld, A.: Range queries on structured overlay networks, *Computer Communications*, Vol. 31, No. 2, pp. 280–291 (2008).

[13] Aspnes, J. and Shah, G.: Skip Graphs, *ACM Transactions on Algorithms*, Vol. 3, No. 4 (2007).

[14] Bharambe, A. R., Agrawal, M. and Seshan, S.: Mercury: Supporting Scalable Multi-Attribute Range Queries, *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM 2004)*, Vol. 34, pp. 353–366 (2004).

[15] Takeda, A., Oide, T. and Takahashi, A.: Simple Dynamic Load Balancing Mmechanism for Structured P2P Network and its Evaluation, *International Journal of Grid and Utility Computing*, Vol. 3, No. 2–3, pp. 126–135 (2012).