原著論文

Peer-to-Peer型クラウドコンピューティング技術による拡散データを統合する分散データベースサービスの検討†

土屋 健*1・吉永 浩和*2・小柳 恵一*3

本論文では、Peer-to-Peer型クラウドコンピューティング技術により、ネットワークに拡散するデータベースを仮想的に統合することで、巨大な論理データベースサービスの構築手法を提案している。現在、多くの機能・方式をもつデータベースが存在し、それぞれが機能・特性に応じたデータベースを構築してきた。これら情報を連携、活用することで高度なwebサービスの創出を期待できる。本稿の提案モデルの構築するオーパレイネットワークは、インターネットに拡散しているデータベース方式、ロケーションに関係なく仮想的に単一データベースとして統合し、SQLによるRDBMS型と同様のデータ操作性と、分散ノードによるスケーラビリティ性という特性をもつことを評価によって明らかにしていた。また、提案モデルの特性を活用することで、ビッグデータへの対応手法について考察を行い、提案手法を実用化するために要求される機能、性能に関して検討を行っている。これら検討に基づき、今後の課題として、現在の実装の改善と、データのレプリケーション手法を明らかにすることで、提案モデルの実用性を高めることができるとまとめている。

キーワード：分散データベース、Peer-to-Peer、クラウドコンピューティング、仮想化技術

1. はじめに

近年、提供するインターネットサービスの一般化、多様化に伴い、web系サービスを中心とするトラフィックが増大している。近年では、サービスを提供するノードをサービス状況に応じて柔軟に割り当てることで、サーバ、ネットワーク、電力など限りある資源を効率的に活用する様々なクラウドコンピューティング基盤技術が提案されており、利用が加速している。現在、web系サービスを構成するサーバは、LAMP（Linux, Apache, My SQL, PHP）で構成されることが一般的となっている。いま、あるweb系のサービスをクラウドサービス化するためには、サービスを構成するサーバ間で、ストレージレベルでの共有サービスが行われ、上述のサービス構成、利用するデータを複

製し、同一のサービス環境を構築できるようにする。そして、複製するサーバ数をサービス状況に適應して制御することで、このサービスのクラウドサービス化を実現できる。筆者らもこれまでに文献[1]において、分散ストレージ技術に関する研究を行っている。

このとき、サーバにおいてデータ更新、データの追加が頻発するサービス（例えばSNS）を提供していた場合、ストレージレベルのデータ複製ではサーバ間の同期が不完全となる。そのため、サーバごと異なるデータがデータベースに存在することになる。そして、データ同期にはそれぞれのデータベース（本構成であればMy SQL）間で毎回のマージが必要となる。つまり、サービスのクラウド化にはデータベースがサービス拡大のボトルネックとなっているといえる。ユーザとインタラクションにより、データ更新、追加が発生するサービスをクラウド化するためには、ストレージレベルの複製では不十分であり、サービスを構成するサーバ間でデータベースのデータを共有する手法を検討することで、データ同期の必要性がなくなると考える。

前述したweb系サービスを構成するノードが採用するデータベース方式は大部分がRDBMS（Relational Data Base Management System）型であり、この方式では分散化によるサービスのスケーラビリティ性の向

† Research on Distributed Database Service to Integrate Diffusion Data by Peer-to-Peer Cloud Computing Technology
Takeshi TSUCHIYA, Hirokazu YOSHINAGA and Keiichi KOYANAGI

*1 諏訪東京理科大学 経営情報学部
Faculty of Business Administration and Information, Tokyo University of Science, Suwa

*2 ログリー株式会社
Logly Inc.

*3 早稲田大学 理工学術院 大学院情報生産システム研究科
Graduate School of Information, Production, and Systems, WASEDA University

上に対応している。特に、この方式ではACID性(Atomicity Consistency Isolation Durability)を有することが最大の特徴であり、このために、Master/Slave型による数台程度の限られたデータベースの分散のみを可能としている。結果として、Masterが一元的にデータベース上の情報を管理することから、近年のサービスで求められる秒間10,000リクエストを超えるトラフィック、テーブル当りのデータ数が1,000,000を超えるデータ、頻繁かつ、大量に更新されるデータの処理には向いていない。しかし、格納されているデータに対し、高いACID性を確保しているので、データ更新の一貫性、複雑なクエリによるデータの高度な操作、処理が可能であるという特徴を持つ。

サービスのスケラビリティ性の課題に対し、KVS(Key Value Store)型のデータベースを中心としたNoSQL型データベースが提案されている[2, 3]。KVSではハッシュ関数により一意の値に数値化された「Key」と「Value」を関連づけて、「Key」の値に基づいて管理する手法である。これら手法ではスケールアウトによる無制限に近いスケラビリティ性による負荷分散と、高可用性といった特徴を持つ。しかし、RDBMS型のようなトランザクションの機能を持たず、複雑な条件検索、集計処理といったこれまでデータベースの基本と考えられる処理がKVSをはじめとするNoSQL型では利用できない。そのため、RDBMS型に相当するトランザクションの機能を利用するにはアプリケーション、ユーザ側においてデータ処理の機能の実装が必要となる。つまり、SQLコマンドのようなプログラマブルな処理に対応していない。そのため、NoSQL型データベースと呼称されている。

インターネットに拡散する様々なデータベースは前述した2方式に大別することができる。このとき、E.Brewer[4]が提唱する分散システムの特性を表すCAP(Consistency Availability Partition tolerance)定理に基づく、RDBMS型はCAの特性を重視し、NoSQL型のデータベースではAPの特性を重視している。つまり、目的とする機能は同一であっても、思想の異なるデータベースであるため、全てのデータベースがどちらかの方式に統一することは困難といえる。今後もインターネット上には様々な方式のデータベースが増加する可能性がある。

現在のインターネットサービスにおいて、爆発的に増加しているデータはSNS(Social Network Service)で使われる個人情報や、スマートフォンなど携帯端末による位置情報、行動の履歴などいわゆる“ビッグデータ”と呼ばれる情報の集積が増大している。現在ではサービスごとにデータが収集され、巨大なデータ

領域を持つNoSQL型データベースにおいて管理されている。さらに、これらデータの分析に基づく情報はリコメンデーションサービス、マーケティング広告などに活用されている。しかし、それぞれのサービスが集める情報からは自サービス以上の情報の収集、分析はできない。また、同種のサービスであれば、収集される情報も類似する情報を集めている。このとき、NoSQL型ではデータに対し、データ条件による検索などの複雑な処理ができず、大量なデータであるビッグデータの活用は困難となっている。

本研究ではPeer-to-Peer(P2P)型クラウドコンピューティング技術を利用し、インターネットに拡散する情報を格納するデータベースを統合的に管理するための論理空間を提案する。ユーザはこの論理空間を経由して、トランザクションを生成することで、格納されているデータベース、方式に関わらずデータの操作、スキーマ情報の生成管理といったデータベース機能の利用を可能とする。結果として、インターネットに拡散しているデータベースを単一のデータベースとして仮想化し、データの統合管理、共有化を可能とする技術である。従って、情報の高密度化によるサービス性の向上、RDBMS型データベースのスケールアウト化、データマイニングによるリコメンデーションサービスの高度化の実現が期待できる。

本論文は以下のように構成される。2章では提案手法への要求を既存技術と比較しながら分析を行い、3章において本稿の提案手法を明らかにする。4章では提案手法の実装評価を行い、5章ではその結果に基づき考察する。そして、6章に本論文をまとめる。

2. 構成する論理空間のモデルに要求される機能

本章ではまず、想定される分散データベースの環境を明らかにし、そして提案手法への要求を明らかにする。

2.1 想定されるデータベース環境

web系サービスを中心としたネットワークサービスにおいて利用されるデータ、サービスが近年急激に増加している。このデータは、SNSを中心としたデータであり、特徴として、頻繁かつ、大量に情報の更新が行われる。このとき、アプリケーション、ユーザから動的かつ、効率的にサービスに利用される情報を生成、管理するためにデータベースは利用されている。

1章でも述べたが、現在利用されるデータベースは様々な方式が提案されており、各サービスはそれぞれデータベースを管理している。これらデータベース

はネットワーク、インターネットに接続されており、それぞれの方式、セキュリティレベルに応じた手法によってネットワーク経由で、データの取得・管理といったデータ操作を可能としている。

2.2 モデルへの要求

本稿で提案するデータベースを統合的に管理する論理空間では、現在サービスを単位として管理されるデータベース上のデータを統合的に管理することにより、“ビッグデータ”のような個人情報、頻繁に更新される履歴情報を含む巨大な情報をネットワークサービスで共有化する。それにより、提供されるサービス性を向上、コンピュータ資源、ネットワーク資源の効率化を実現する。このとき、アプリケーションサービスからデータ管理の機能を分離・レイヤ化することで、今後も増大するであろうネットワークサービスの構築を容易とすることが期待できる。これらを実現するためには、本論文では以下の点を満たすデータ管理を行う論理空間の構成技術が要求されることとなる。

1章で述べたが、現在利用されるデータベースは2つの方式に大別ができ、それぞれの機能・特性は異なる。本稿で提案する論理空間では、データが属するデータベース方式に関係なくユーザ及び、アプリケーションからそれぞれのデータの操作の機能が要求される。特に、前述したビッグデータの利用を想定すると、No SQL型データベースの持つサービス状況に適應してノード構成の変更ができる柔軟性、ノード分散によるスケーラビリティ性と、データに対してRDBMS型のもつデータ操作性の機能が要求される。特に、この論理空間のインターフェースとして現在のRDBMS型データベースのインターフェースに親和性の高い方式（例えばSQL）を採用することで、アプリケーション資産の再利用性、開発性の向上を期待することができる。

ネットワーク上で利用可能であるデータベースサービスは、それぞれあらかじめセキュリティポリシーを定めることにより、各々のセキュリティレベルに応じて運用されている。そのため、複数のデータベースを統合的に管理することは、このポリシーが異なる可能性が高い。従って、本稿の提案では、それぞれのデータベースのポリシーに適應した手法で提案する論理空間から柔軟にデータ、テーブルの操作を可能とする手法が要求される。

以上のことから、本稿の提案手法では少なくとも上記の要求される機能を満たす必要がある。

2.3 関連研究

近年、実用化が進むNo SQL型のデータベースは1章で述べたKVS型以外にも列指向型、ドキュメント指向型など複数の方式が存在する。特に、Google Big table [5]、Apache Cassandra [6]など実装サービスの多い列指向型ではデータは列を単位として複数のノードで分散管理されていることから、高速かつ効率的にデータを管理、複製を可能としている。ドキュメント指向型のMongo DB [7]ではスキーマを持たず、不定形のデータを保存可能としている。これらNo SQLと呼ばれるデータベースは、従来のRDBMS型と比べて大量のデータ処理、速度やスケーラビリティ性などに特化した方式が多く、一方で、データの整合性に関して保証が行われていない。つまり、その特徴は様々であり、想定するサービスに応じてデータベースを選択することになる [8]。今後も特定の方式を持つデータベースに統一されることはなく、データがそれぞれのデータベースに遍在することを意味しており、本稿の提案する仮想化された論理空間を利用することでこの遍在するデータの共有化を期待できる。

分散するデータベースの仮想化による統合手法として、データベースフェデレーション技術が検討されている [9]。文献 [9]では、ドメイン内に分散するデータベースを統合サーバに仮想化し、低負荷で分散するデータを管理するために、フェデレーションとレプリケーションを組み合わせることで実現している。しかし、1章で前述したRDBMS型の分散と同様に、特定の統合サーバに仮想化するため、Master/Slaveによる分散手法の課題を解決していないこと、同一ドメインなど狭いエリア内に分散するデータベースのみを対象とすることといった本稿の提案と比較して限定された範囲、スケーラビリティにおいて、実用性を主眼に置いた提案となっている。

本稿の提案と類似する目的を持つ提案としてNo SQL型を上回るスループットと、RDBMS型の特徴であるACID性をもつVolt DB [10]が提案されている。しかし、高スループット化のためにオンメモリで動作する。そのため、データベースとしてのデータスケーラビリティに限界が存在する。また、データベースがあらかじめ特定のSQL文に基づいてデータ分散が行われるため、このパターンにあてはまらないSQLコマンド以外は利用できない。従って、パターン化される検索を中心としたサービスなど利用可能なサービスのみに制限されることになる。

筆者らはこれまでに、サービスを提供するサーバ側を中心としたクラウドコンピューティング環境だけでなく、ユーザ側も対象となるユビキタスサービス環境

において、オブジェクトの検索、具体的には提供するサービス、管理するコンテンツを対象とした分散検索技術に関して検討を進めてきた[11, 12, 13]。これら文献中において触れているが、データ管理方式に関係なく検索サービスの提供が可能であることから、本稿で提案するデータを統合管理する論理空間の情報検索サービスとしての利用を期待することができる。

3. DBを統合管理する論理空間のモデル

本章では2.1節で明らかにした論理空間への機能要求、2.3節の関連研究を踏まえて提案モデルを明確化する。

3.1 データの統合管理を実現する論理空間の概要

本節では拡散するデータベースを統合する論理空間の概要を示す。

本稿の提案する論理空間は、プラットフォームを利用するノードによって、インターネット上にオーバーレイする空間として構築される（以降では提案する論理空間をオーバーレイネットワークと示す）。そして、インターネットに拡散しているデータベースのデータ、テーブルはこのオーバーレイネットワーク上の領域として対応付けられ、その領域に該当データがマッピングされる。そして、アプリケーション、ユーザはこのオーバーレイネットワークへアクセスすることで、マッピングしたデータ、テーブルへのアクセスが可能となる。

オーバーレイネットワーク上の各データ、テーブルは全てオーバーレイネットワーク上の領域を示すアドレスと、マッピングされたデータ、もしくはデータが管理されているノードのポインタ情報とマッピングしてオーバーレイネットワークに登録されており、一意に管理されている。

全てのデータベース上のデータは、前述した動作を繰り返すことにより、対象となる全データがオーバーレイネットワーク領域に登録され、データベースに関係なくデータの操作、テーブルの生成が可能となる。

オーバーレイネットワークのスケラビリティ性を確保するため、オーバーレイネットワークはクラウドコンピューティング基盤技術、つまりインターネット上のノードの協調によりネットワークを管理する。具体的にはデータと、スキーマ情報を分散するノードにおいて協調して管理することにより、管理するデータの規模、データベースサービスの状況に適応して、柔軟に分散管理するノードの追加、変更を可能としている。つまり、Peer-to-Peer 型クラウドコンピューティング技術に基づく柔軟なデータベースサービスの構築を

図1 提案モデル概要

目指すことになる。

アプリケーションおよび、ユーザから、オーバーレイネットワークに仮想化されたデータの操作を行うために、インターフェースとしてSQLを採用する。これにより、既存のアプリケーションとの親和性も高く、開発者も低コストで、大量のデータを活用したサービス、データの高度な操作を実現できる。

図1には本稿で提案するオーバーレイネットワークの各機能の概要を示している。雲型で示されているオーバーレイネットワーク上に対応付けられた各データベースと、上部にはこの空間上のデータから生成されたテーブルを示している。アプリケーションからはSQLコマンドによってこのテーブル、データの操作が可能となる。

3.2 分散環境における論理空間の構築

本節では論理空間であるオーバーレイネットワークの構築手法について明らかにする。

3.2.1 分散プラットフォーム技術

構築されるオーバーレイネットワークは、サービススケーラビリティ性の観点から、特定の固定的なノードが提供するのではなく、ノードの協調によって論理空間を構築する。筆者らはこれまでにPeer-to-Peer (P2P) 型分散協調プラットフォーム技術の検討を進めており[11, 12, 13]、本研究においてもノードがP2P型クラウドコンピューティング基盤（以下、プラットフォームとする）を採用することで、一般ノード、固定的なサーバ、サービスを提供するノードなど特定ノードだけでなく、ユーザ主導で柔軟にノードの追加、変更を可能とすることで、論理空間の分散スケ

ラビリティを決定できる。プラットフォームで構築されるオーバーレイネットワーク上では全てのノード、オブジェクトは128bitのIDによって識別される。また、一般ノードもオーバーレイネットワークを構成することから、ノード移動に伴う論理空間からの着離脱や、状況に応じたサービススケールの変更にはプラットフォームの機能で対応する。また、各プラットフォームでは後述する手法により分散したデータもしくは、実データへのポインタ情報を管理している。

3.2.2 論理空間を構成するノードの選定

ダイナミックかつ適応的にオーバーレイネットワークを構築するには、ノード選定にこれまでに筆者らが検討した先行研究[14]を利用する。このとき、条件の良いノードから順に、構築するオーバーレイネットワークの規模に応じて、データを分散管理するノードを決定する。この選定手法では、複数の観点から評価を総合的に判定し、安定度、処理能力、ネットワーク環境といったメトリック評価に基づく順位が決定される。そして、ベストエフォートとなるノードがオーバーレイネットワーク構成ノードとして選定される。

3.3 分散環境におけるデータの管理手法

本節では分散するデータベースのデータと、オーバーレイネットワーク領域へマッピング手法を明らかにし、分散ノードでの管理手法について検討する

3.3.1 既存データの論理空間へのマッピング

インターネットに分散する各種データベース上のテーブル、データは、オーバーレイネットワーク上の領域と対応付けることにより、アプリケーションからデータの操作・取得が可能となる。

まず、オーバーレイネットワーク上に対応付ける領域を確保する。このとき、構築するオーバーレイネットワークはRDBMS型データベースと同様の2次元の座標空間(64bit×64bit)を構築する。この2次元座標を持つ論理空間において、“(3.2.1で示したオーバーレイネットワーク上の) ID+テーブル名(データベース名)”をハッシュ関数により数値化し、この値に対応する領域をマッピングするデータの始点(図2: Starting Point)とする。各領域の始点は一意に決定し、容易にアクセスすることが可能となる。あらかじめ、マッピングするデータ量、テーブルサイズに相応した領域を始点より確保し、この範囲にテーブル、データのマッピングを行う。この領域へのマッピングの手法に関しては3.3.1項に後述する。また、テーブル情報であるデータ関係性もマッピングされた領域情報に基

図2 データのマッピング手法

づいて更新する。始点はこの領域および、実データベースへの情報を管理する。そして、全ての始点情報はキャッシュデータとして一元的に管理され、各始点を容易に検索することを可能としている。

確保された領域は、その大きさによって、他のテーブル、データベースの始点、または確保しようとする領域と重複するケースが想定される。そのため、現在は確保しようとした領域がすでに他のテーブル、データベースによって確保されていた場合では、領域の始点の導出には“オーバーレイネットワーク上のID+テーブル名+1”としてハッシュ値化を行う。この場合でも、他の領域と重なるケースも想定され、以降では“+2, +3, +…”と数値を増加させることで対応している。

始点からデータをマッピングするために確保する領域の大きさには制限をしていないが、基本的にはテーブル、データベースの管理するデータ数に加え、想定される追加されるデータ分の領域のみを確保する。新規のデータベースであれば想定されるデータ量に応じた領域を確保する。また、想定外のデータ増加に伴い、あらかじめ確保した領域を超えるケースでは、確保する領域を追加することになる。このとき、隣接する領域に確保できる場合では、始点の管理する情報を更新し、データのマッピングが可能となる。隣接する領域に確保できない場合では、現在は前述した領域が重複した場合と同様に、拡張する領域を別の領域に確保し、始点に分割された領域に関する情報を追記することで対応している。今後、分割して管理される領域が増加する場合には、領域の再配置の手法の検討が必要となる。2次元の座標空間を持つテーブル、データは、その関係性を維持して確保されている領域にマッピングする。No SQL型データベースのように座標空間を持たないデータは、データIDに基づいて、オーバーレイネットワークへマッピングされる。

3.3.2 マッピングにおけるセキュリティへの対応

2.2節において検討した各データベースに設定されたセキュリティレベルへ柔軟に対応するために、(a) オーバレイネットワークにおいて、マッピングされたデータを全て管理する手法、(b) オーバレイネットワークではデータへのポインタ情報のみを管理し、必要に応じてマッピング元のデータベースからデータを取得する手法、の2種類の手法に対応している。(a)の手法は、(b)のように毎回データベースへの取得処理が発生しないため、オーバレイネットワークからのテーブル情報、データ取得までの応答性が優れ、遅延少なく操作が可能である。しかし、(a)の場合ではデータがマッピング元のデータベースの外部で管理され、データの複製に制限されるデータ、具体的には著作権、個人情報などに関連するデータが想定され、データベースからのデータ複製は制限される。(b)の手法では、オーバレイネットワークに直接データをマッピングするのではなく、格納されているデータ実体へアクセスするためのポインタ情報のみを管理することで、各データベースが設定するセキュリティレベルに対応すると同時にデータ操作を可能としている。

3.3.3 座標空間の管理

オーバレイネットワークにマッピングされたデータは、2.2節で検討したRDBMS型と類似した64bit × 64bitの領域を持つ2次元のテーブル型である座標空間を持つ論理空間として管理される。

従って、2次元の空間として、複数ノードの協調による分散管理のためのアルゴリズムが必要となる。このとき、管理コストの低減化を図るため、2次元の空間をそのまま分割して管理するのではなく、低次元化するためにZ-Ordering[15]を採用する。これによりテーブル型の座標空間全体を1次元の128bitにより表すことができる。詳細は文献[15]において示されているが、座標空間のそれぞれの軸方向ごとにビットを割り当て、重ね合わせねることで1次元化を行う。このとき、元の2次元空間は全て1次元の128bit値として表される。この値は、各座標がそれぞれ1次元の値と対応することから、この値をオーバレイネットワーク領域のユニークなアドレスとして利用する。また、この1次元化した値に基づきテーブル、データを分散配置する場合、他ノードに問い合わせることなく各ノードにおいて一意にオーバレイネットワーク上の元の座標を算出できることから、ノードにおける管理項目の削減、問い合わせのためのプロトコルも不要となる。従って、分散環境においても効率的なテーブル、データのデータ管理及び、操作が可能となる。

3.3.4 1次元情報の分散ノードにおける管理

分散するノードにおいて管理するために、1次元化された座標情報は、図3に示すスキップリスト[16]をベースとした階層化リスト構造で管理される。このとき、リスト構造であるため、各ノードはリストの接続元及び接続先となるノードIDのみの情報を管理している。

図に示す“ $x=0$ ”、“ $y=0$ ”、“ $x=y$ ”のそれぞれの階層では、オーバレイネットワークの座標空間にあわせて、“ x 軸方向”、“ y 軸方向”、そして“対角軸方向($x=y$)”の各リンクをあらかじめ生成・管理することで、テーブル操作、領域内のデータ取得におけるデータ操作、およびノード移動の効率の向上を期待できる。この他にも必要に応じて、リスト階層を追加することを可能としている。

最下層のリストはオーバレイネットワーク上に確保されている領域全てがリストとして管理される。このとき、図中で示しているが全ての領域に対してリンクを生成するのではなく、オーバレイネットワークに領域を確保すると同時に、このリストが生成され、データのマッピング、ポインタ情報が管理されることになる。

分散するノードが協調してリスト情報を管理するために、3.2.1項で示した各ノードIDに基づいて図中最下層のリストを分割して管理する。このとき、オーバレイネットワークを構成するノードは、Chord[17]のリング管理手法と同様に、各ノードIDを超えない範囲に関して分割して管理する。つまり、利用者は、各ノードIDを見ることで管理しているリスト番号の大まかな範囲を推測でき、効率的に目的の領域の情報を取得することを可能としている。今、各領域はハッシュ関数によって始点が決めていることは3.3.1項で述べた。そのため、特定のノードが分担する範囲にデータが集中するケースは頻発することはないと考えるが、映像などサイズの大きいデータが格納された場合では分担する領域の範囲は少なくとも、ノードへの負荷が大きくなる。そのため、P2P型プラットフォームの機能として、隣接するリストを管理するノード間

図3 1次元化した数値の分散管理手法

で、定期的にオーバーレイネットワーク構成ノードの情報を更新し、予期しないノードの離脱や、ノードにおける負荷などを確認する。そして、必要があれば、プラットフォームの調停により、リストの分担する範囲を自律的に調整し、構成ノード間の負荷の分散化を行う。極端にノードへの負荷が大きくなる場合では、オーバーレイネットワークを構成するノードを追加し、逆に分担する領域が消失したノードはオーバーレイネットワーク構成ノードから離脱するといった柔軟なシステムとしての機能を提供することになる。

3.3.5 データ更新への対応

オーバーレイネットワークへマッピングしたテーブル、データ、そしてマッピング元のテーブル、データは状況に応じて情報の更新が発生する。しかし、これら更新に対し、データ整合性 (Consistency) を確保するには、データ元のデータベースと、オーバーレイネットワークにマッピングされたデータがそれぞれ独立して更新されると、その整合性を維持することが困難となる。そのため、オーバーレイネットワークにマッピングしたテーブル、データベースを単位とするデータ更新手法を以下に示す (i), (ii) のいずれかの手法に限定する。

- (i) マッピング元のデータベースではテーブル、データの更新を禁止し、オーバーレイネットワークの操作によってのみ、テーブル、データの追加・更新を可能とする。
- (ii) ユーザ/アプリケーションからの操作が、テーブル、データの操作及び、読み出しに限定できるのであれば、マッピング元のデータベースのみ追加・更新を可能とし、オーバーレイネットワークからはテーブル、データ操作のみを可能とする。

いずれかの手法に追加・更新の操作を限定することにより、P2P型分散システムで議論・採用されることの多いEventual Consistencyモデルによるデータ整合性を確保することができる。更新手法の制限により、更新から十分に時間が経過して分散ノードの同期が終了することで、テーブル、データの整合性は保証できる。

(i) のケースでは、オーバーレイネットワーク上のテーブル、データとしてユーザ/アプリケーションからの操作が行われ、追加・更新情報は3.3.4項で述べた領域を管理するノードに通知された時点で反映される。このとき、他にはオーバーレイネットワーク上のデータ、テーブルの追加・更新は発生せず、Eventual Consistencyモデルを満足する。また、オーバーレイ

ネットワークにマッピングされたテーブル、データの追加・更新情報をマッピング元に反映させることも可能であるが、マッピング元が異なるテーブル、データ情報を反映させることができず不完全となる。そのため、現在はオーバーレイネットワーク上での更新情報はマッピング元に反映はさせていない。(i) のケースが想定されるサービスとして、複数のアプリケーション/ユーザでテーブル、データを共有し、従来のRDBMS型と同様にそれぞれがテーブル、データの追加・更新、といった操作を行うことを想定している。

(ii) のケースでは、マッピング元のデータベースのみが更新され、テーブル、データが順次オーバーレイネットワーク上にマッピングされる。このとき、テーブル、データの追加・更新を監視するインスタンスをマッピング元のデータベースごとに配置することで、更新情報を常に監視してデータ整合性を満足することになる。いま、アプリケーション/ユーザはテーブル、データの追加、更新などの書き込み操作を行うデータベースと、テーブル、データの操作と読み出しを行うオーバーレイネットワークと操作によって異なるデータベースを利用することになる。このとき、(ii) の手法はオーバーレイネットワークを利用するよりも、遅延が少なくなることが予想され、ビッグデータに対してオーバーレイネットワークを適用することが検討できる。詳細は5.3節において検討するが、(ii) を利用することで、データ種別ごとに複数のNo SQL型データベースを採用した場合でも、オーバーレイネットワーク上でデータ間の関係性テーブルを生成、管理することができるため、アプリケーションに要求される機能の負荷が軽減すると考える。

3.4 論理空間のインターフェースとデータ操作

論理空間であるオーバーレイネットワークのIF(インターフェース)として、SQLを採用する。SQLの利用により、エンジニアの開発効率の向上、既存アプリケーション資産、ライブラリの再利用性の向上を期待できる。

アプリケーション、ユーザによって生成されたSQLコマンドは、オーバーレイネットワークのインターフェース部で受信し、SQLの解析が行われる。3.3.1項で示したテーブル、データへのアクセスには前述した始点の情報を利用し、マッピングされた領域へ到達できる。そして、プラットフォームでは、座標情報を3.3.4項で述べた1次元化を行い、図3のスキップリストにおいて該当範囲を管理するノードから情報を取得することになる。

オーバーレイネットワークにマッピングされたすべて

のデータに対して、データの追加・更新、条件による検索、関係性の生成、新たなテーブルの生成が可能となっている。具体的なSQLコマンドレベルで述べれば、select によって各領域から条件による抽出を行い、これらデータをjoinによって結合して新しいテーブルを領域にマッピングする。このとき、テーブル情報の更新・追加・削除、新たにオーバレイネットワークに生成されたテーブル情報から関係性の生成といったRDBMS型サービスの機能が利用可能となる。つまり、実データの管理されるデータベースに関係なく、単一のデータベース上のデータとして取り扱うことができる。

オーバレイネットワークに生成されるテーブルは、識別のためのオブジェクトIDにより一意に管理され、現在の実装では“テーブル名”をIDとして利用指定する。このIDは3.3.1項で示した始点と同様に、テーブルの生成時にハッシュ関数により数値化され、このテーブルのオーバレイネットワークにおける配置が決定される。このとき、すでに配置先が利用されている場合では、オブジェクトIDとして“テーブル名+1”として別の場所にテーブルを生成する。複数のプロセス・アプリケーションにおいてテーブル上のデータを利用するためにはオブジェクトIDによりテーブルへアクセスすることができる。

3.5 現在の実装

本章で示した各機能に基づいて、図4に示すように構築した。本論文の提案モデルはミドルウェアとして動作し、その上位にこれらの機能を利用するアプリケーションが配置される。提案モデルを実装したミドルウェアは表1に示す環境で構築されており、以降の評価で利用されることになる。図中では、評価に用いられる計測アプリケーションが配置されている。現在

表1 実装環境

OS	Windows 7 Professional
CPU	Xeon 2.4GHz
RAM	4 GB
DB	My SQL 5.1.44
Lang	Java SDK 1.6

の実装では、オーバレイネットワークの機能は3つのレイヤから構成され、各レイヤの提供する機能の概略を図中に示す。

4. 評価

本章では実装した提案モデルを利用して、E-Commerceにおけるデータベースのサービス性に関する評価を行う。

4.1 評価の環境

本評価は、実ノードによる評価では分散ノードのスケラビリティを得ることが困難であること、シミュレーションでは分散データベースとしての実用性評価が困難であることから、図4で示した現在の実装を用いたエミュレーションによって評価を行う。このとき、2台の実ノード上に100ノードずつ、計200ノードのオーバレイネットワークを構成するノードを配置してエミュレーションを行う。このとき、従来のRDBMS型では数台のノード、No SQL型では分散ノード数に制限されないスケールフリーであるサービスも多く存在することから、本評価でもP2P型分散プラットフォームとして、O(100)程度のノードスケラビリティは要求されると考え、200ノードによる評価を行う。

全てのノードは、それぞれ独立したインスタンスであり、各ノードはそれぞれの管理するデータに基づいて独立して動作する。また、このノードに発生するインタラクションも全てノードがネットワークを利用した場合と同様のプロトコルが利用される。このとき、各ノードのもつIDに従って、IDの奇数、偶数によって配置するノードの決定をしている。従って、図3で示したノードIDに基づくスキップリストでは、この実ノード間でインタラクションが多く発生し、プロトコル評価としては実ノードに近い評価ができることを想定している。

オーバレイネットワークへマッピングされるデータは、図4の実装に、RDBMS型データベース6台が接続される。このとき、各データベースに含まれるテーブル、データをオーバレイネットワークへマッピングして評価に利用することになる。

図4 評価用の実装

4.2 エミュレーションによる評価

本評価では、データベースベンチマークTPC (Transaction Processing Performance Council)のwebサービス系の評価を行うTPC-W[18]を参考にE-Commerce系サービスを想定した評価を行う。

いま、本評価では、独立したE-Commerce系webサービスの統合を想定し、各サービスがそれぞれ運用していたデータベース計6台をそれぞれオーバーレイネットワークへマッピングする。これらデータベースはRDBMS型のデータベースであり、サービスが管理する情報(顧客情報、画像を含む商品情報)が含まれている。このとき、各データベースは十分に大きく10,000のオーダのデータを管理し、[18]で利用されるデータ、テーブルを用いている。

一般ユーザを想定する100プロセスが同時にこのwebサービスへアクセスする。各プロセスはそれぞれトップページからwebサイトを訪問し、ランダム生成されるクエリによってマッピングされたオーバーレイネットワーク上の商品検索を行い、100件/ページとなる検索結果ページを生成して検索上位のページより表示する。つまり、該当する検索結果が1000件であれば、検索結果として10ページ生成されることになる。各ユーザはwebサイトを訪問し、検索結果を取得・表示する動作を行う。具体的には、オーバーレイネットワークがユーザからクエリを受信し、検索結果となるテーブルを生成するまでに要した時間を評価の対象としている。このとき、それぞれのユーザにおける結果を平均化し、評価としてまとめる。このとき、検索結果の生成には、マッピングされたオーバーレイネットワーク上のテーブル、データを横断的に検索し、検索結果ごとのテーブルおよび、ページの生成が行われることになる。

4.3 各項目における評価

本節では各観点に基づき、評価の結果をまとめる。

4.3.1 ノードの分散によるサービスの応答性

ノード分散と、データベースとしてサービス性を評価するために、オーバーレイネットワークの検索によって得られる2つの検索結果(1000件及び、1400件)から、ページの生成(10ページ、14ページ)に要した時間と、オーバーレイネットワークを構成する分散ノード数の関係をまとめ、図5に示す。

x軸はオーバーレイネットワークを構成する分散ノード数、y軸はそれぞれのデータがページ生成するまでに要した時間を示している。この図中ではデータ数の異なるどちらのケースにおいても、x軸が10ノードの

図5 データ操作と分散ノード数の関係

範囲までほぼ一定となることを示している。ここでは示していないが、評価は分散ノード数が200まで行っている。分散ノード数が増加しても評価に要する時間はほぼ一定であり、分散ノード数が増加しても同様の傾向が続く。つまり、この評価ではノードの分散に起因して、テーブル、データ操作に遅延が発生しているのではないといえる。しかし、図中では、1400件と1000件のケースを比較すると明らかにデータ件数に比例して、操作に要する時間が大きくなることがわかる。図中の分散ノード数1の場合は、固定の1ノードで同様の評価であるから、どちらの場合でもノードの分散に伴い約85%の評価に要する時間の増加が観測できる。つまり、同一のデータ量であれば分散ノード数の変化、つまり分散スケーラビリティはオーバーレイネットワークのサービス性、プロトコルについて大きく影響しないことがわかる。

しかし、評価に用いた現在のオーバーレイネットワークの実装では、データの分散よりも、プラットフォームにおいて処理する情報量の増加に起因して、評価に要した時間の増加が発生している。このとき、情報の分散に起因するプラットフォームのパフォーマンスの低下は、一定程度の範囲であれば許容できる。しかし、現在の評価では、ノードの分散数に比例するのではなく、分散するノードのプラットフォームにおける処理量の拡大(本評価では1000件から1400に増加)に伴い大きな時間を要していると考ええる。そのため、提案モデルの根源的な問題ではなく、図4で示した現在の実装上の問題といえる。

4.3.2 データベースとしてのサービス性評価

本項の評価では検索結果に含まれるデータ数と、評価に要した時間の関係を明らかにする。これにより、現在の実装における分散データベースとしてのサービス性・実用性に関して評価を行う。いま、3.3節で述

図6 提案モデルのサービス評価

べたように、オーバレイネットワークへマッピングされるデータの管理手法にはデータ複製の課題から、オーバレイネットワーク上で2つの管理手法へ対応していた。本項の評価では、これら手法に関して評価を行い、図6にその結果を示す。図中x軸は評価において検索結果として生成されたデータ数、y軸は評価に要した時間を示している。このとき、(a)データをオーバレイネットワーク上の領域にマッピングしている場合、(b)オーバレイネットワークには各データのポインタ情報のみをマッピングした場合の結果を示している。

図6では(a)、(b)どちらのケースでも、対象となるデータ数の増加に伴い、評価に要した時間は増大する。特に、(b)のケースにおける増加する割合が、(a)のケースと比較して大きいことを観測できる。このとき、評価では、(b)のケースと比較して、マッピング元となる各データベースに対し、検索クエリに対する検索結果を受け取り、それぞれの結果から、オーバレイネットワークとしてこれらを統合した検索結果のためのテーブルが生成される。このとき、オーバレイネットワーク外における処理が必要となるため、(a)と比較して評価に要する時間が増加することになる。

(a)においても、マッピング元のデータベースにはアクセスしないが、5000件で約10秒の時間を要しており、前項においても触れたが、データベースとしての性能は既存のRDBMS型と比べて現状の実装では十分な性能を持つとは言えない。(a)におけるサービス性能の改善は、(b)における評価の改善と直結するといえる。

5. 考察

本章では、3章で提案したモデル、及び4章におけるその評価から、提案するオーバレイネットワークモデルの特性と、それに基づいた想定される利用につ

いて定量的、定性的にまとめ、今後の検討すべき課題について述べる。

5.1 データベースとしての考察

提案するオーバレイネットワークモデルのデータベースとしての機能性、サービス性について考察する。

5.1.1 トランザクションへの対応

トランザクションが満たすべきACID性の各項目(原子性・整合性・独立性・永続性)を提案モデルに対して考察する。

オーバレイネットワークにおけるデータ整合性に関しては、3.3節で触れたデータ追加・更新を2つのうちのどちらかに制限することでEventual Consistencyに対応することは議論した。

原子性に関しては、提案モデルでは従来のRDBMS型と同様に、SQLコマンドにより、オーバレイネットワーク上のデータを操作可能としているため満たしているといえる。

独立性は、上記の整合性に基づくデータ追加・更新手法の制限と、実装上の排他制御により確保することができると考える。

分散するノードによって構成されるオーバレイネットワークでは、提供するノードが替わった場合でも、操作の永続性は保たれるといえる。

以上の各項目を満たすことから、オーバレイネットワークにマッピングされた情報に対して、トランザクションの要件を満たすといえる。同時に、オーバレイネットワーク自体が、大きなデータベースとしてみなすことができる。

5.1.2 データ関係性情報の生成

4.3節に示したように、オーバレイネットワークにマッピングされたデータは、マッピング元のデータベースを意識することなく、横断的なデータの検索・操作によるテーブルの生成を可能としていた。つまり、ネットワークに拡散するデータなど、データベース作成時には分割されて格納されていたデータに対して、新たにデータ移行することなくオーバレイネットワークへデータをマッピングするだけで関係性を生成、そしてアプリケーションにおいて利用することが可能となる。具体的には、文献[19]において、ビッグデータを扱うサービスではデータベースの書き込み性能確保の目的から、RDBMS型データベースをNoSQL型として利用し、性能を確保するためにマスター分割(Sharding)も頻繁に行われている。このとき、仕様変更などがあっても、データベースを移行しない限り、

データの関係性を生成・管理すること、テーブルを生成することはできない。しかし、本稿で提案したオーバーレイネットワークにこのNo SQL型で管理されるデータをマッピングすることで、データベース統合を行わなくとも後天的にデータ関係性を生成することが可能となる。つまり、提案手法はビッグデータを対象としたサービスに対して、データベースの再構築なく適用でき有効であるといえる。

5.1.3 データベースサービスの可用性

提案するオーバーレイネットワークは、3.2節及び、4.2節で示したように、固定ノードではなく、P2P型プラットフォームを適用したノード間の協調により構成される。従って、構成するノードは柔軟にノードの追加・削除、そして変更が可能となっている。また、分散ノード数はサービス品質に影響を与えないことを示していた。従って、トラフィックの増大に起因するノード負荷、ノード障害・ネットワーク障害の発生時には当該ノード及び、その管理領域の変更することにより、サービスを継続することができる。このとき、固定ノードで提供される従来のRDBMS型のサービスよりデータベースサービスとしての可用性は高いと考察することができる。また、ビッグデータなどに利用されるNo SQL型サービスは提案モデルと同様にP2P型基盤を採用することから、これらと同等の可用性を持つといえる。

耐故障性の観点に立つと、分散ノードで構成される提案手法では予期しないノードの離脱への対応が必要となる。このとき、データベースとしては前述したようにサービスは継続できるが、離脱ノードが管理するテーブル情報、データの損失により、データベースのACID性のうち、整合性、永続性が損なわれる。これに対し、あらかじめオーバーレイネットワーク上のテーブル、データの複製(レプリケーション)を作成することで、テーブル情報、データ損失の回避が可能となる。同時に、これら複製情報は、キャッシュとして利用することも想定でき、今後検討すべき課題の一つと考察できる。

5.1.4 アプリケーションの開発性

オーバーレイネットワークではマッピングされているテーブル、データ操作のインターフェースとしてSQLを採用していた。これにより、従来のRDBMS型サービスの操作性により、分散するデータベース上のデータ、テーブルを統合的に取り扱いが可能であることを評価において明らかとしていた。このとき、従来のRDBMS型が提供する機能と変わらないことから、

ユーザ及び、アプリケーションの開発者は、モジュール内のデータベース指定をオーバーレイネットワーク上のテーブル、データへ変更することでサービスを容易にオーバーレイネットワークへ適用することができる。つまり、既存のソフトウェアを軽微な変更でそのまま利用可能であること、モジュール、ライブラリをそのまま移行できるため、資産の再利用性、SQLの採用による新たな技術習得が不要であるため、提案モデルはエンジニアによる開発の効率性が高い特徴を有するといえる。

また、No SQL型を利用するアプリケーションでは、格納される大量のデータに対して、データベース性能の課題から、位のアプリケーションにおいてデータ関係性を生成・管理している。このとき、新たに追加となるデータエントリ、機能の度に、開発者はアプリケーションの新機能本体に加え、テーブル、データ操作の開発が必要となる。このとき、提案モデルの活用により、従来のRDBMS型と同様に、アプリケーションの外部で大量のデータに対して、テーブル、データ関係性の生成・管理が異なるサービス間で管理データを共有することが可能となり、開発工数の削減となる。つまり、開発の効率性が向上したといえる。

5.2 分散情報システムとして

本節では提案するオーバーレイネットワークモデルに対し、分散情報システムの観点から考察する。

5.2.1 ノード分散のスケーラビリティ

提案するオーバーレイネットワークはP2P型オーバーレイネットワーク手法の採用、4.3節の評価結果より、2台以上の分散であれば分散ノード数のスケールに起因して、データベースサービス品質の低下は発生していなかった。つまり、分散情報システムとしては状況に適応して分散スケールを動的に変更でき、サービスに影響しない柔軟なシステム、提案モデルはスケールアウト可能なシステムであるといえる。また、従来のRDBMS型サービスの機能をNo SQL型スケールで提供できる特徴を持つといえる。

5.2.2 構成ノードの負荷

提案モデルでは、3.3節で示したマッピング領域を決定するハッシュ関数の利用により、マッピングされるデータの論理空間内のデータ平衡化が行われていた。また、オーバーレイネットワーク上の領域は、図3に示したスキプリストとして、構成するノードが状況に応じて分割範囲を決定していた。このとき、大きいデータや、頻繁にアクセスされる情報は、論理空間

の特定の範囲に集中することはなく、複数のノードに分散することで、各ノード負荷の最小限化を図っている。

以上のことから、オーバレイネットワークを構成するノードは、インターネットに接続される一般ノードによって、オーバレイネットワークを構成することができると考察できる。同時に、負荷の分散化を可能とすることで、ビッグデータなどの大量データを管理する場合においても、スキップリストのノード分割を変更することで自律的に負荷の分散化が行われる。従って、システムとしての可用性・耐障害性は高いと考察できる。

ビッグデータ、webコンテンツといった高頻度で特定ノードが管理する情報へアクセスが集中する場合では、5.1.2項でも述べたが、マッピングされたデータの複製をあらかじめオーバレイネットワーク内で管理しておくこと、利用されたデータのキャッシュを準備することにより、特定ノードへのアクセスの集中に起因する負荷の軽減を期待することができる。一方で、データ読み出しのみが行われる場合であれば問題は存在しないが、データ自体が更新される情報であれば、データ本体、複製、キャッシュデータそれぞれの間で情報の整合性を損なう可能性がある。そのため、本稿では検討しておらず今後の検討課題の一つであると考察できる。

5.2.3 セキュリティレベルへの対応

提案モデルでは、それぞれのデータベースのセキュリティポリシーに柔軟に対応していた。しかし、図6の評価で示したように、セキュリティポリシーに対応することで、情報システムとしての性能は低下していた。

提案モデルの実用化設計、ソースコードのリファクタリングなどにより、性能の改善を期待することはできる。しかし、いずれのケースにおいても各データベースのポリシーに対応することで、サービス性の低下を免れることはできない。これは、全てのP2P型の分散システムが抱える課題であり、データ情報の保護ためには分散性が制限されるケースが存在すると考える。いま、図6の評価は、全てのデータがこの情報の保護下にある特殊な状況であり、実際のサービスでは一部の特定データベース、テーブルに対してデータ情報の保護が行われることを想定している。データの分散を許容できないケース、一般ノードへデータ分散が許容できないケースなど様々な運用上のケースが存在するが、これらは運用上の課題として、提案モデルを構成するノードを柔軟に検討することで対応できると考えている。

5.3 ビッグデータへの対応

SNSなどのweb系サービスから生成される“ビッグデータ”に対し、提案するオーバレイネットワークの適用を検討する。

ビッグデータを生成するweb系サービスについては5.1.2項で触れたが、これらサービスではデータベース性能を重視するため、SQL型データベースをNoSQL型による管理、Shardingによるデータベースの分割管理が頻繁に行われている。このとき、アプリケーションの機能追加、新サービスによるデータ利用などで、分割されて管理するデータ間に関係性が必要となる場合は再データベース分割（Re-Sharding）を行うことでデータベースの再構築を行っている。そのため、近年ではビッグデータに対し、このRe-Shardingが頻繁に発生するといった非効率な運用が行われている[19]。提案するオーバレイネットワークモデルでは、NoSQL型、RDBMS型問わず、オーバレイネットワーク内にてデータ関係性を生成・管理することができるため、Shardingによるデータベースの分割、分散化が進んだ場合においても、各データベースに対して、Re-Shardingすることなくデータ関係性、テーブル情報の生成・管理を高い分散スケーラビリティにおいて可能とすることができる。従って、提案モデルをビッグデータに適用することは、サービスの効率性、開発性の向上に非常に有益となることが期待できる。

4.3節で示したように、現在の実装ではそのままビッグデータにそのまま適用できる性能は現在のところ有していないと考察できる。今後、実装の改善により、遅延の減少を期待することができるが、提案モデルはインターネットに分散するノードの協調により構成されることに起因して、固定ノードによる集中型のサービスと比較して、ある程度の範囲で遅延を避けることはできない。従って、実装の改善と同時に、この条件下で、提案するオーバレイネットワークによるビッグデータへの対応に関して考察を行う。時系列に沿って生成されるビッグデータを管理するためには、データベースは少なくとも以下に示す2つの処理に対応する機能が要求される。

1つ目の処理はデータの作成・更新処理が挙げられる。具体的には、あらかじめオーバレイネットワークにマッピングされたデータ、新たにオーバレイネットワークに格納されたデータ、これらデータ、テーブル情報の更新、さらに新たなデータ、テーブル情報をオーバレイネットワークへマッピングといったこれら処理操作が要求される。いま、3.3節で述べたように、データ整合性の問題から、データの追加・更新は2つの手法に制限されていた。このとき、遅延少なく

更新処理を行うことができるのは、マッピング元となるデータベースを更新する手法であり、更新された情報は順次オーバーレイネットワークへマッピングされる。ビッグデータへの対応を考えた場合、マッピング元となるデータベースがNo SQL型による軽量な方式を採用することで、既存の多くのビッグデータサービスと同様に、データ更新処理に対応することができる。オーバーレイネットワークへのマッピングも前述したようにEventual Consistencyによるデータ整合性を採用しており、特定のデータばかりが更新、参照処理を繰り返されるケースは稀であると考えられる。このとき、更新されたデータが対応領域へマッピングされるまでに時間を要しても、オーバーレイネットワークの利便性から許容できると考察する。

2つ目の処理はオーバーレイネットワーク上のデータ、テーブル情報を検索し、取得するといった読み込みに関する処理である参照処理が発生する。参照処理は、あらかじめ生成済みである情報、具体的にはオーバーレイネットワークに格納されているテーブル及び、データ情報の参照を行うことであり、新たに情報を生成しない。つまり、ビッグデータを対象とするのであれば、オーバーレイネットワークに格納されている大量の情報をできるだけ低遅延で検索、参照を行うことが必要となる。このとき、参照処理を低遅延で行うには、ユーザの近傍にオーバーレイネットワーク領域のレプリケーションを配置することで、ビッグデータのような高頻度の参照処理に対する処理負荷、遅延を回避することができる。異なる複製間で更新処理が同時に行われる場合には、データ整合性の観点から領域の複製範囲をテーブル単位などに限定することで対応できる。複製の配置は、オーバーレイネットワークを構成するノードに対して、処理負荷を分散し、ノードあたりの負荷を軽減することで、データベース性能に対して効果を期待できる。

以上のことから、ビッグデータに対して、現在の実装をそのまま適用するだけでは不十分な性能であるが、本評価で示すように提案モデルの各機能は十分にビッグデータに対しても対応できる。

5.4 想定される利用環境

ここまでの検討から、提案モデルを実システムとして期待される利用環境について議論する。

分散システムとしてノード分散のスケラビリティ性に優れていたこと、P2P型オーバーレイネットワークによる高い柔軟性を有することから、データベースをサービス状況に適応して、構成するノードの分散スケラビリティの規模を増大・削減できるデータベ

スのクラウドコンピューティング化を具現化する手法として利用が期待できる。また既存のSNSサービスでは、管理するデータを内部向けサービスにおけるターゲット広告、マーケティングなどに利用している。提案モデルの利用により、データベースがアプリケーションから独立していることから、オーバーレイネットワーク上全てのデータ、テーブル情報はマーケティングなどデータ解析に利用でき、さらにこれら情報とデータ元のSNSサービスとは関連のないから独立したアプリケーションと組み合わせサービスの高度化を期待することができる。また、提案モデルは、広大な論理空間のデータ、テーブルをRDBMS型の操作性を持つことから、多くの独立したサービス、アプリケーション／ユーザに対して、共有・利用するデータ基盤、データベースレイヤとしての可能性を見込むことができる。

評価から、現在の実装ではデータベースとして十分な性能を得られていないが、Blob型データ、ハイビジョン映像などのファイルサイズの大きいメディアデータ管理基盤として利用すれば、従来でも移動・複製といった処理に時間を要しており、現在の実装データベースサービスとしての応答性よりも、データベース管理による操作性・機能性といった利便性の向上が期待される。つまり、現在の実装においても、一定条件下では実システムとしての利用を期待できるといえる。

5.5 今後の課題

考察より、提案モデルは従来のRDBMS型サービスと比較して、同等の機能を有することに起因したサービス性、開発性、構成ノードの分散に起因するスケラビリティ性、可用性が優れていることを明らかにしていた。しかし、現在の実装による評価では、サービスの応答性の向上、トランザクション遅延の削減が急務であった。しかし、分散システムであることに起因して、遅延の改善には限界があることも述べた。そのため、今後の検討すべき課題として、実装の改善と並行して、5.2節、5.3節で述べたオーバーレイネットワークのレプリケーション、キャッシュを論理空間で管理する手法について検討を行う。これら手法により、提案モデルの実システムとして利用できる可能性が増大するものと期待できる。

6. おわりに

本稿ではP2P型クラウドコンピューティング技術により、ネットワークに拡散するデータベースを統合して仮想的な単一のデータベースを構築するオーバーレイネットワークモデルを提案していた。

現在, 多くの機能・方式のデータベースがネットワーク上に拡散しており, それぞれがデータ管理機能をユーザ, アプリケーションへ提供してきた。しかし, web系サービスの進化により, 大量かつ高頻度に更新されるデータの管理といったデータベースもサービス性の向上が要求されてきている。同時に, ネットワークに拡散しているこれら情報を連携, 活用することで高度なサービスの創出を期待することができる。本稿の提案モデルでは, 拡散するデータベースを対応付けることにより仮想的に統合して, SQLによるRDBMS型と同等のデータ操作の機能を提供していた。同時に, 分散ノードの協調によるサービススケーラビリティを有することで大量のデータにも対応可能となっていた。

実装した提案モデルの評価から, 現在の実装ではそのままビッグデータに対応できるという性能は得られていないが, オーバレイネットワークが提供する更新処理, データのレプリケーション, データのキャッシュなど複製手法を活用することでビッグデータに対応できると考察していた。今後の課題として, データ複製手法の検討と, 実装の改善によるデータベースサービスの性能改善が急務であるとまとめている。

謝辞

本研究の一部は, 総務省「戦略的情報通信研究開発推進制度(SCOPE)」(採択課題番号: 122304003)の研究助成によるものである。

参 考 文 献

- [1] 吉永浩和, 土屋健, 小柳恵一, “分散協調ストレージシステムにおけるネットワーク規模および一貫性を考慮した複製配置手法の提案”, 日本知能情報ファジィ学会学会誌 “知能と情報” 22 巻, 2 号, pp.246-256, 2010 年 2 月
- [2] F.Chang, J.Dean, S.Ghemawat, W.C. Hsieh, D.A. Wallach, M.Burrows, T.Chandra, A.Fikes, R.E. Gruber, “Bigtable: A Distributed Storage System for Structured Data”, ACM Transactions on Computer Systems, Vol.26 Issue 2, Jun. 2008
- [3] W.Vogels, “Eventually consistent”, Communications of the ACM - Rural engineering development, Vol.52 Issue 1, Jan. 2009
- [4] E.Brewer, “Towards Robust Distributed Systems”, Symposium on Principles of Distributed Computing (PODC), Portland, 2000
- [5] F.Chang, J.Dean, S.Ghemawat, W.Hsieh, D.Wallach, M.Burrows, T.Chandra, A.Fikes, and R. Gruber, “Bigtable: A Distributed Storage System for Structured Data”, Seventh Symposium on Operating System Design and Implementation, Seattle, WA, Nov., 2006
- [6] A.Lakshman, P.Malik, “Cassandra: a decentralized structured storage system”, ACM SIGOPS Operating Systems Review, Vol.44 Issue 2, Apr. 2010
- [7] MongoDB, “http://www.mongodb.org/”
- [8] E.Meijer, G.Bierman, “DATABASES A co-Relational Model of Data for Large Shared Data Banks”, ACM Queue, Vol.9, Issue 3, pp.1-19
- [9] 大川昌弘, 黒沢亮二, 福田剛志, “分散データベース環境における複製データの仮想化による SQL 処理の最適化手法”, 信学会技報, 第18回データ工学ワークショップ, Mar. 2007
- [10] U.F.Minhas, R.Liu, A.Abounaga, K.Salem, J.Ng, S.Robertson, “Elastic Scale-out for Partition-Based Database Systems”, Proc. of Int. Self-managing Database Systems (SMDB '12), Washington, DC, Apr. 2012
- [11] 土屋健, 吉永浩和, 小柳恵一, “コンテンツ意味情報を利用した P2P 型分散検索手法の実現”, 電子情報通信学会論文誌, Vol. J90B, No.7, pp.648-659, Jul. 2007
- [12] 土屋健, 吉永浩和, 小柳恵一, “意味情報インデックス独立分散管理によるユビキタス環境向けの P2P 型分散検索手法の検討”, 日本知能情報ファジィ学会学会誌 2009 年 2 月
- [13] T.Tsuchiya, H.Sawano, M.Lihan, H.Yoshinaga, and K.Koyanagi, “A Distributed Information Retrieval Manner Based on the Statistic Information for Ubiquitous Services”, Progress in Informatics No.6, pp.63-78, Apr., 2009
- [14] 吉永浩和, 土屋健, 小柳恵一, “Peer-to-Peer 環境における協調作業アプリケーションのための適応型調停プラットフォーム”, 情処会論, Vol.46, NO.8, pp.2162-2174, 2005 年 8 月
- [15] G.M.Morton, “A computer Oriented Geodetic Data Base and a New Technique in File Sequencing”, Technical Report, 1966, Ottawa, Canada: IBM Ltd.
- [16] W.Pugh, “Skip lists: a probabilistic alternative to balanced trees”, Communications of the ACM 33, pp.668-676, Jun. 1990
- [17] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for Internet applications,” in: ACM SIGCOMM 2001, San Diego, CA, 2001, 149-160
- [18] TPC-W: TPC Transaction Processing Performance Council, “http://www.tpc.org/tpcw/”
- [19] DeNA, “Mobage を支える技術”, 技術評論社, 2012 (2012 年 5 月 31 日 受付) (2012 年 10 月 3 日 採録)

[問い合わせ先]

〒391-0292 長野県茅野市豊平5000-1

諏訪東京理科大学 経営情報学部

土屋 健

TEL: 0266-73-9647

FAX: 0266-73-1231

E-mail: tsuchiya@suou.waseda.jp; tsuchiya.takeshi@rs.tus.ac.jp

著者紹介

つちや たけし
土屋 健 [正会員]

2009年早稲田大学大学院情報生産システム研究科博士課程修了，博士(工学)。現在，諏訪東京理科大学経営情報学部助教。情報通信ネットワーク，分散協調情報基盤に関する研究に従事。IEEE，情報処理学会，電子情報通信学正会員。

よしなが ひろかず
吉永 浩和 [非会員]

2011年早稲田大学大学院情報生産システム研究科博士課程修了，博士(工学)。現在，ログリー株式会社代表。レコメンドシステムや広告システムの事業開発に従事。

こやなぎ けいいち
小柳 恵一 [非会員]

1975年慶応義塾大学工学部電気工学科卒業。1977年同大学大学院修士課程修了。1998年大阪大学大学院通信工学専攻博士課程修了，博士(工学)。1977年日本電信電話公社(現NTT)入社。2003年早稲田大学大学院情報生産システム研究科教授。ネットワークミドルウェア，知能化ネットワークの研究開発に従事。博士(工学)。1996年度電気通信普及財団賞受賞。IEEE Senior Member，ACM 会員，電子情報通信学会員

Research on Distributed Database Service to Integrate Diffusion Data by Peer-to-Peer Cloud Computing Technology

by

Takeshi TSUCHIYA, Hirokazu YOSHINAGA and Keiichi KOYANAGI

Abstract :

In this paper, we propose the strategy for the distributed database technology to integrate data of diffused databases on the Internet by Peer-to-Peer cloud computing technology, which enables to construct logical data spaces for controlling and managing them by SQL.

In recent years, there are several kinds and functions of databases, and they have been constructed original databases including their characteristics and faculties. Utilizing them would be expected to create new wonderful services. The proposal model in this paper has following characteristics: one is realizing controls and managements of data relationships by SQL over the functions of database like RDBMS, KVS and locales of each database. The other is the service scalability by collaborations among distributed nodes.

From the result of evaluation, proposal model in this paper clarifies that it combines the advanced data management by SQL and service scalability by collaborations of distributed nodes. As for the next step, we discuss about the improvement of service performance by the replication technology such as caching manner.

Keywords : Distributed Databases, Peer-to-Peer, Cloud Computing, Virtualization

Contact Address : Takeshi TSUCHIYA

Faculty of Business Administration and Information, Tokyo University of Science, Suwa
5000-1, Toyohira, Chino city, Nagano Pref, 391-0292, JAPAN
TEL : 0266-73-9647
FAX : 0266-73-1231
E-mail : tsuchiya@suou.waseda.jp;tsuchiya.takeshi@rs.tus.ac.jp