データベースシステム　125-28
（２００１．７．１８）

位置情報データベースの統合利用に関する一考察

秋山 豊和 †　原 隆浩 ‡　西尾 章治郎 ‡
† 大阪大学サイバーメディアセンター
‡ 大阪大学大学院工学研究科情報システム工学専攻

近年，GXML や OpenGIS など地理情報を統一的に扱う規格の標準化が進められており，これに伴って位置情報をもつデータ（位置情報データ）を扱うアプリケーションが急速に普及するものと考えられる．分散管理されている位置情報データを有効利用するためには，それらのデータを位置情報をキーにして効率的に検索可能なシステムの開発が必要となる．一般に公開された位置情報データの検索については，従来のロボット型やディレクトリ型の検索エンジンの利用が考えられるが，特定の組織間でプライベートにデータを共有する場合には，各サーバが提供するデータ間の関係を考慮することで，より効果的な共有が可能である．そこで本稿では，複数の組織で分散管理されている位置情報データの共有方法を提案する．さらに，提案した共有方法を用いた場合のシステム性能について，シミュレーション実験により評価を行う．

A Consideration on Integration of Location Relevant Information Databases

Toyokazu AKIYAMA†　Takahiro HARA‡　Shojiro NISHIO‡
†Cybermedia Center, Osaka University
‡Dept. of Information Systems Engineering, Graduate School of Engineering, Osaka University

Due to a recent standardization of the geographic information data, e.g. GXML, OpenGIS, the applications which deal with location relevant data become more important and widespread. To utilize those location relevant data managed by different organizations, a spatial search system using location information as a search key is required to be developed. The conventional robot and registration base search engines are applicable for searching public location relevant data. However, when location relevant data are shared privately among particular organizations, different approaches which consider the relationship among the data are more effective. In this paper, we propose a new method which shares location relevant data managed by different organizations. We also show the simulation results regarding the performance evaluation of our proposed method and the analysis of the simulated results.

1　はじめに

近年，GXML や OpenGIS など地理情報を統一的に扱う規格の標準化が進められており，これに伴って位置情報をもつデータ（位置情報データ）を扱うアプリケーションが急速に普及するものと考えられる．そこで，各サーバにおいて分散管理されている位置情報データを有効利用するためには，これらのデータを位置情報をキーにして効率的に検索可能なシステムの開発が必要となる．道路や建物などの地理情報のように一般に公開された位置情報データについては，従来のロボット型の検索エンジンにより情報を自動的に収集したり，公開されているディレクトリ型の検索エンジンへ情報を登録することで，検索することが可能となる．一方，位置情報を記録する対象によっては，外部に公開せずにプライベートに位置情報を共有したいという要求も考えられる．例えば，比較的大規模な組織において，ネットワークの運用管理を行う上でネットワーク機器や計算機の位置情報を共有したい場合などが考えられる．このような情報はセキュリティ上，組織内部や関連する組織間でプライベートに共有するのが望ましい．

組織間でプライベートに位置情報を共有する場合，主に次の２つの方法が考えられる．

(1) 中央に位置情報データベース統合用のサーバを設置し，統合したい情報を持つデータベースから情報を収集して蓄積する．データアクセス時は，統合用サーバを経由して各データベースが管理するデータにアクセスする．この方法は，統合用サーバの負荷が大きく，中央に比較的高性能なサーバを用意できる場合に適している．
(2) 各組織で分散管理している位置情報データベースに，データを共有したいサーバの情報を記録する．データアクセス時には，自身がもつ情報

に基づいて要求データを持つサーバにアクセスする．この方法では，サーバ間で共有する情報を工夫すれば，各サーバにかかる処理負荷が低減できるものと考えられる．さらに，ボトムアップ的にシステムを構築する場合に適している．

本稿では特に (2) の方式に注目し，比較的柔軟に位置情報データが共有可能なデータ共有方法について考察する．また，サーバ間でどのような情報を共有すればサーバにかかる負荷をそれほど増加させることなく，データを効果的に共有できるかについても考察する．さらに，考察結果に基づいてデータの共有手法を提案し，提案手法を用いてシステムを構築した場合のシステムの処理負荷を調べるため，シミュレーション実験を行う．

以下，2 章では関連研究について述べる．3 章では本稿で提案するデータの共有方法について説明する．4 章では提案手法の評価のために行ったシミュレーション実験の結果を示し，最後に 5 章で本稿のまとめとする．

2 関連研究

位置（地理）情報を扱うためのデータベースシステムについては，これまでに様々な研究が行われている [1, 3, 7, 8, 9, 10]．文献 [7] では，GIS をサポートするためのデータベースシステムに必要な機能について議論している．文献 [3, 8] では，オブジェクト管理システム SHORE[2] を用いた地理情報管理システム PARADISE について述べている．PARADISE では特に，大規模な地理情報データを高速に処理するためにデータを計算機に再分配し，空間演算を並列に処理する方法に注目している．本稿では独立に管理されている位置情報データを統合利用することを考えているため，これらの研究とは目的が異なる．

文献 [1] では，幾何情報データと従来の文字情報などのデータを統合的に扱うシステムについて議論している．また，そのような統合システムの構築に CORBA を用いた場合のシステム設計について述べている．このように，地理情報を扱うラッパーやメディエータの研究では，主に幾何情報データとその他のデータの統合利用に注目したものが多い．文献 [9, 10] では，モバイル端末の位置情報を管理するシステムについて述べている．この論文では端末の位置を点で扱っているため，データ構造として高速に点検索を実行できる SR-Tree を用いている．位置情報を提供するサーバでは，非常に大規模な分散システムに対応するため，各サーバの管理するデータの範囲を緯度，経度によって分割しており，緯度，経度の情報をキーとして検索を行う．本稿で議論する位置情報データの共有方式では，各サーバが管理するデータ領域は限定しておらず，別々の組織において独立に管理されている位置情報データを比較的柔軟に統合できる．

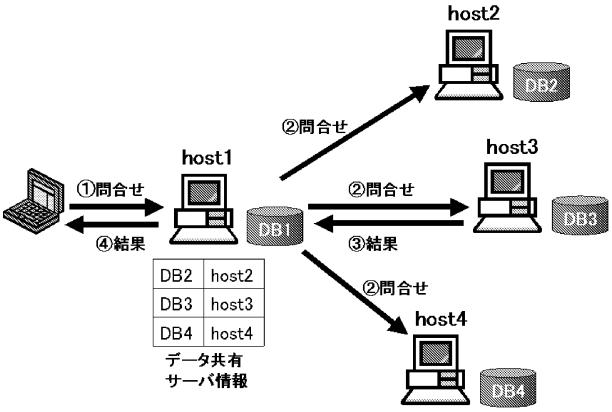

図 1. 単純な共有方式

空間データに対するアクセス手法に関する研究も数多く行われている [4, 6, 5]．文献 [4] では，多次元データのインデクスに関する研究についてまとめている．文献 [6] では，複数のディスクを用いた場合に，多次元データのインデクス手法である R-Tree を用いて処理を高速化する方法について議論している．文献 [5] でも同様に多次元データのインデクス手法である Grid ファイルについて，複数の計算機にデータを分散配置する際のデータ処理の高速化について議論している．このように，空間データのアクセス手法に関する研究では，処理の高速化のみに注目しており，分散管理されているデータの統合利用については考慮していない．

3 位置情報データの共有方法

1 章で述べた (2) の方式によって，特定のサーバ間で位置情報を共有するためには，各サーバに情報を共有したいサーバの情報を記録する必要がある．サーバに接続するために必要最低限な情報であるサーバの識別子のみを記録した場合，問合せが発生すると全てのサーバに対して問合せを行う必要がある（図 1）．

そのため，データ検索時のサーバの負荷を低減するためには，サーバの識別子だけではなく，サーバが持つデータ領域に関する情報も共有し，検索時に余分なサーバへの検索を防ぐ必要がある（図 2）．しかし，逆にデータ領域に関する情報を共有すると，データの更新時にデータ領域が変更された場合，データを共有しているサーバにもデータ領域の更新を伝播させる必要があるため，より多くの負荷がかかることになる．

以上のようなことを考慮して，本稿ではまずデータ共有のための基本的なサーバ間接続方式を提案す

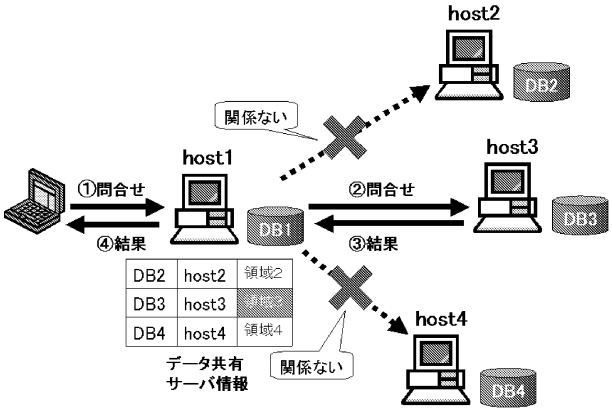

図 2. データ領域を利用した共有方式

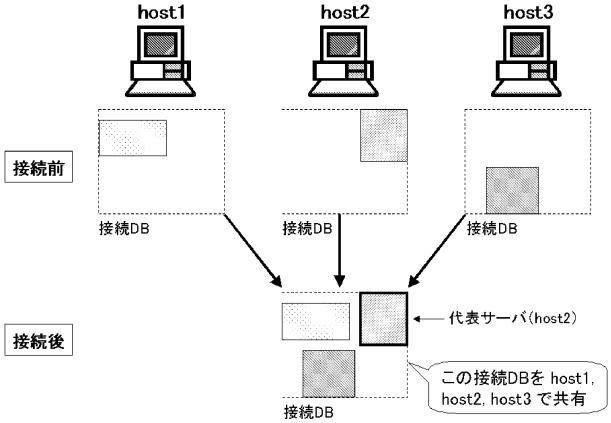

図 3. 密結合の例

る．さらに，提案した接続方式における範囲検索の手順およびデータの追加，更新が発生した場合の動作について説明する．

位置情報データに対するアクセス手法としては，さまざまなものが提案されている [4]．本稿では，実装が容易で比較的良い性能を示す R-Tree に注目する．R-Tree では，格納するデータの範囲に接する最小の矩形領域 (MBB: Minimum Bounding Box) を用いてデータアクセスの効率化を実現している．本稿においても，データ検索時の処理効率化には位置情報データの MBB を利用し，サーバが保持するデータのデータ領域は，全てのデータの MBB を含む MBB を用いることにする．

3.1 基本的な接続方式

まず，サーバ間の基本的な接続方式として，密結合と疎結合の２つの方式を提案する．各接続方式では，データを共有している接続先サーバの情報を自身のデータベースに保存する．以下では，このデータベースを接続 DB と呼ぶ．また，サーバの MBB とはサーバが保持する全てのデータの MBB を含む最小の矩形領域を示す．同様にサーバ群の MBB とは全てのサーバの MBB を含む最小の矩形領域を示す．

密結合： データを共有している全てのサーバの識別子および各サーバの MBB を，接続 DB に保持する．密結合によって接続されたサーバのうちの一つが代表サーバとなり，各サーバの MBB の更新時には代表サーバが同時更新の制御を行う．代表サーバの識別子の情報は全てのサーバで保持する．代表サーバの識別子は，データ検索時に密結合で接続されているサーバ群を識別するために用いられる．密結合で接続された場合，データ検索時には全てのサーバの MBB を接続 DB に保持しているため効率的に検索を実行できるが，各サーバの MBB の更新時には全てのサーバに更新を通知する必要があるため非常にコストがかかる．以下では，密結合で接続されたサーバ群を密結合サーバ群と呼ぶ．また，密結合サーバ群に含まれるサーバの MBB を密結合 MBB と呼ぶ．

疎結合： データを共有しているサーバ間に上下関係があり，上位のサーバと下位のサーバにおいて，異なる形式で接続相手のデータ領域の情報を保持する．上位のサーバおよび下位のサーバはいずれも密結合サーバ群の代表サーバとする．上位サーバでは，下位サーバの識別子および下位サーバの密結合サーバ群が持つデータ領域に関する情報を自身の密結合サーバ群と共有する．下位サーバでは，上位サーバの識別子を自身の密結合サーバ群と共有する．上位サーバの接続 DB の要素が，自身の MBB または自身の密結合サーバ群の MBB と重複する場合，重複領域に関する情報も自身の密結合サーバ群と共有する．下位サーバの識別子は複数もつことができるが，上位サーバの識別子は，データ検索，更新などの手続きの簡単化のため一つだけ持てるものとする．つまり，上位サーバと下位サーバは，1 対多の関係である．また，疎結合の接続により構成されるサーバ間の階層関係は多階層にもなり得る．

上位サーバの接続 DB に保持される下位サーバの接続 DB の全ての要素を含む MBB を下位疎結合 MBB と呼ぶ．また，下位サーバの接続 DB に保持される，上位サーバの接続 DB の要素と下位疎結合 MBB との交差領域（複数存在する可能性がある）を上位疎結合 MBB と呼ぶ．各サーバの接続 DB には，これら上位および下位疎結合 MBB とともに，MBB に含まれるデータを保持するサーバの識別子も保持する．

疎結合で接続された場合，上位サーバの接続 DB の要素の更新によって，下位疎結合 MBB

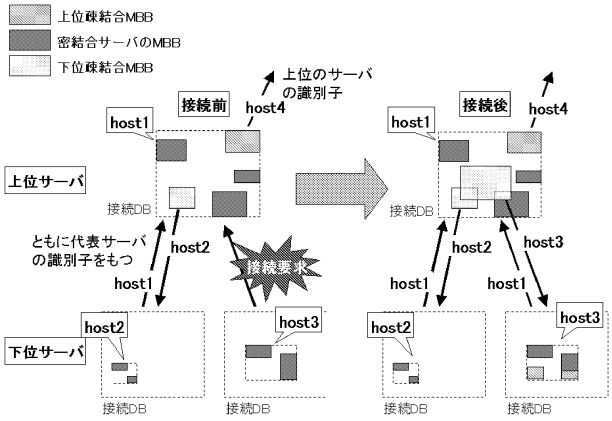

図 4. 疎結合の例

との交差領域が変化しない限り，下位サーバに更新は伝播しない．また逆に，下位サーバの MBB 更新によって下位サーバの密結合サーバ群の MBB が更新されない限り，上位サーバには更新は伝播しない．そのため，(i) 下位疎結合 MBB の重複領域をできるだけ小さくし，(ii) 新規データの追加やデータ更新ができるだけ密結合サーバ群の MBB に影響しないように接続することによって，更新の伝播を抑えることができる．しかしその反面，データ検索時には，接続 DB に記録されていないサーバにアクセスする際に，他のサーバを経由する必要がある．

密結合および疎結合の例を図 3，4 に示す．

3.2 範囲検索の手順

位置情報データ検索の中で最も重要なものの一つである範囲検索に注目して，検索の効率について議論する．以下では，前節で述べた基本的な接続方式を用いた場合の範囲検索の手順について述べる．

1. クライアントから問合せを受け取ったサーバ（問合せ発生サーバ）は，まず，自分の保持する接続 DB に対して問合せを行い，問合せ領域と交差する MBB を求め，その MBB に含まれるデータをもつサーバの識別子を求める．求めたサーバに対して，さらに問合せを行う．この際，問合せ発生サーバを検索対象から省くため，問合せ発生サーバの識別子を問合せ先のサーバに渡す．また，問合せ先のサーバにおいて，問合せ発生サーバの所属する密結合サーバ群を識別できるように，問合せ発生サーバの識別子に加えて，代表サーバの識別子も渡す．
2. 接続 DB の要素には，3.1 節で述べたのように上位疎結合 MBB，密結合 MBB，下位疎結合 MBB があり，その他に，自分が保持するデータを全て含むローカル MBB も含まれている．基本的に問合せ領域と交差領域を持つ MBB を担当するサーバに問合せを送信するが，疎結合で接続された上位サーバについては，上位疎結合 MBB と問合せ領域が交差領域を持たない場合でも，問合せ領域が自身の接続 DB の全ての要素を含む MBB に包含されない場合には，問合せを行う．
3. 他のサーバからの問合せを受けたサーバでは，問合せ発生サーバと同様に，接続 DB に対して問合せを行い，返ってきたサーバの識別子リストを用いて再帰的に問合せを行う．問合せ発生サーバの場合と異なり，問合せ元サーバの密結合サーバ群が，自分が所属する密結合サーバ群と異なる（異なる密結合サーバ群から問合せを受け取った）場合のみ，新たに問合せを発生する．また，問合せ元サーバは，新たな問合せの対象から除外する．
4. 問合せ先のサーバから受け取った結果を問合せ元に返し，最終的に，問合せ発生サーバが結果をクライアントに返す．

3.3 データの追加・更新・削除の影響

提案方式では，サーバ間で保持するデータに関する情報を共有するため，各サーバに対してデータの追加，更新，削除が発生した場合，サーバ間で共有する情報も更新する必要がある．データの追加，更新，削除が頻繁に発生する環境においては，共有情報の更新コストがサーバの負荷や検索性能に大きく影響するものと考えられる．以下では，データ追加，更新，削除が発生した際の，サーバ間の情報の更新手順について述べる．

1. データの変更には，以下の 3 つの場合が考えられる．
   (a) データの追加が発生した場合
   (b) データの削除が発生した場合
   (c) データの更新が発生した場合

   (a) の場合，新規データの MBB がローカル MBB に包含されていれば，ローカルに処理する．包含されていない場合は，手順 2 に進む．(b)，(c) の場合，ローカル MBB が変化するかどうかを調べるために，削除または更新が発生したデータを除く全てのデータの MBB を調べる必要がある．ローカル MBB が変化していれば手順 2 へ進む．
2. ローカル MBB の変更により，接続 DB の全ての要素を含む MBB が変化した場合，まず代表サーバに通知し，以後の処理は代表サーバで実行する．代表サーバでは，データ更新中を示すフラグをたてる．さらに，疎結合で接続された

上位サーバに対して，下位疎結合 MBB の更新を通知するため，手順 3 に進む．接続 DB の全ての要素を含む MBB が変化しない場合，密結合サーバ群に対しローカル MBB の変更を通知するため，手順 4 に進む．

3. 疎結合で接続された上位サーバに対して，下位疎結合 MBB の更新を通知する．この通知により，上位のサーバ群において，手順 2 からの更新手続きが開始される．この過程で，さらに上位のサーバ群の更新が発生する可能性がある．上位サーバでも手順 2 と同様に，データ更新中を示すフラグをたてる．このフラグにより複数の更新が同時に発生した場合に情報の整合性を保つことができる．ただし，上位のサーバで発生した更新と，下位のサーバで発生した更新がデッドロックを起こす可能性があるため，タイムアウト処理およびランダムなタイミングでの再更新処理を行う必要がある．
上位サーバにおいて，下位疎結合 MBB の更新により下位疎結合 MBB と，接続 DB の他の要素との交差領域に変化が生じた場合は，新しい交差領域群を上位疎結合 MBB として下位サーバに返す．上位疎結合 MBB には，交差している MBB をもつサーバの識別子を記述するため，下位のサーバでは，新しい上位疎結合 MBB の情報を基にして上位疎結合 MBB の追加，削除，更新を行うことができる．
4. 密結合サーバ群に対し，ローカル MBB の更新を通知する．上位疎結合 MBB の更新があった場合にはこれも通知する．
5. ローカル MBB の更新および上位疎結合 MBB の更新により，下位疎結合 MBB とこれらの MBB の交差領域に変化が生じた場合は，新しい交差領域群を上位疎結合 MBB として下位サーバに通知する．下位サーバ群において手順 4 からの更新手続きが開始される．下位のサーバでも上位のサーバと同様に，データ更新中を示すフラグをたてる．
6. 以上の手順が終了したら，関連したサーバにコミット通知を送信し更新処理を完了する．

3.4 新規接続の手順

提案方式において新たにサーバ同士が接続した場合，サーバ間の情報を更新する必要がある．ここでは，新規接続の手順について述べる．切断した場合にも同様な更新が発生するが，ここでは省略する．新たに接続した場合の下位疎結合 MBB，密結合 MBB，上位疎結合 MBB の更新の通知については，データ追加の際の手順とほぼ同じであるが，接続元サーバにおける手順が異なる．以下に，密結合および疎結合で接続する場合についてそれぞれ示す．

密結合の場合： 疎結合として 1 つの上位サーバにしか接続しないため，新規の密結合の接続元および接続先のサーバがどちらも疎結合で上位サーバに接続している場合については考えない（あるいはどちらかのサーバが所属するグループの最上位の代表サーバを接続元とする）．以下に手順を示す．

1. 接続元サーバが代表サーバでない場合は，代表サーバに接続処理を依頼する．以下では接続元を代表サーバとする．
2. 接続先サーバが代表サーバでない場合は，接続先を代表サーバに変更する．
3. 接続元および接続先のサーバ間で新規代表サーバを決定する．
4. どちらかのサーバが上位疎結合を持っている場合，下位疎結合 MBB の変更を通知する．また，上位疎結合を持っていたサーバが代表サーバでなくなった場合，代表サーバの変更も通知する．下位疎結合 MBB の更新については 3.3 節と同様である．
5. 接続元および接続先のサーバ間で接続 DB の内容を交換し，それぞれの密結合サーバ群に，接続相手側の接続 DB の内容，手順 4 で返された上位疎結合 MBB および新規代表サーバの識別子を通知する．上位疎結合を持っていなかったサーバ群には，上位サーバの識別子も通知する．
6. 下位疎結合についても，交差領域の変更通知および代表サーバの変更通知を行う．

疎結合の場合： 密結合の場合と同じく，疎結合として 1 つの上位サーバにのみ接続するため，疎結合により上位サーバに接続しているサーバからの疎結合要求は考えない（あるいはそのサーバが所属するグループの最上位の代表サーバを接続元とする）．以下に手順を示す．

1. 接続元サーバが代表サーバでない場合は，代表サーバに接続処理を依頼する．
2. 接続先が代表サーバでない場合は，接続先を代表サーバに変更する．
3. 接続先サーバに，新規下位疎結合 MBB および下位サーバの識別子を追加する．この新規追加によって生じる更新処理は 3.3 節と同様である．新規追加した疎結合 MBB と他の MBB の交差領域を，上位疎結合 MBB として接続元サーバに返す．
4. 接続先サーバから受け取った上位疎結合 MBB および上位サーバの識別子を密結合サーバ群に通知する．

表 1. シミュレーションのパラメータ

パラメータ	値
全領域の広さ	6400×6400
総サーバ数	16, 32, 64
サーバの MBB の幅および高さ（サーバ数）	400~1600(16), 300~1200(32), 200~800(64)
問合せ領域とサーバの MBB の中心の距離（サーバ数）	0~1600(16), 0~1200,3200,6400(32), 0~800(64)
問合せ領域の幅および高さ	10~1600(16), 10~1200(32), 10~800(64)
オブジェクトの大きさ	1~300

5. 上位疎結合 MBB の更新によって生じた交差領域の変更を下位疎結合サーバ群に通知する．

4 データ共有方法に関する考察

提案方式では，サーバ間の接続方式のみについて議論しており，どのサーバ同士を接続するかは指定していない．しかし，疎結合は非対称の接続であるため，接続形態によって，問合せ性能や更新時の負荷などが大きく変化するものと考えられる．そこで，提案方式を用いてサーバ間を接続した場合に，更新や問合せにかかる処理負荷をシミュレーションによって算出し，それに基づいて実際に接続するサーバを選択する方法について考察する．

シミュレーションで使用したパラメータを表 1 に示す．シミュレーションでは，指定した広さの領域に指定した個数のサーバの MBB をランダムに配置し，それらのサーバを結合した場合の問合せコスト，データ追加，削除コストなどについて調べた．問合せは，ランダムに選択したサーバに対して，表 1 に指定した範囲で範囲検索を発生させた場合にアクセスされたサーバ数および経由ホスト数を調べた．また，データ追加，削除については，ランダムに選択したサーバに対して，サーバの MBB とほぼ同程度の領域でデータを追加および削除した場合に，サーバ間の情報の更新手続きによってアクセスされたサーバ数および経由ホスト数を調べた．

サーバ間接続方式としては，密結合のみ (CLOSE)，疎結合のみ (LOOSE)，密結合と疎結合の混合 (MIX) について比較を行う．疎結合を含む接続方式としてはランダムに接続対象を選んだ場合 (LOOSE, MIX) と，接続されるサーバおよびサーバの MBB が既知であるとして，最も距離が近い要素を選んだ場合 (LOOSE-NEAR, MIX-NEAR) を用いる．

4.1 提案方式の評価

総サーバ数を変化させた場合に，各問合せでアクセスされたサーバ数および最大経由サーバ数の平均を図 5，6 に示す．図 5 から明らかなように，全て密結合で接続した場合がアクセスされるサーバ数が最も少ない．また，疎結合のみの場合よりも，密結合を混合した場合の方がアクセスされるサーバ数が少なくなっていることが分かる．最も距離が近いサーバを選択して接続した場合には，アクセスするサーバ数が大幅に減少していることも分かる．実際にシステムを構築する場合には，あらかじめ接続されるサーバの情報を知ることは難しいが，文献 [4] で述べられている空間インデクスと同様に，何らかの方法で距離が近いもの同士をクラスタリングする必要がある．図 6 の最大経由サーバ数についても同様のことがいえる．

図 7，8 に，問合せ領域の中心とサーバ MBB の中心の距離を変化させた場合に，各問合せでアクセスされたサーバ数および最大経由サーバ数の平均を示す．総サーバ数は 32 とした．問合せ領域を変更しても，密結合では全てのサーバの情報を知っているため結果は変化しない．疎結合のみの場合および密結合を混合した場合においても，問合せ領域を変更しても結果が変化しない．これはランダムに接続しているため疎結合における上位サーバと下位サーバの領域の重なりが多く，問合せ領域がサーバの MBB に近い場合にも，多くのサーバにアクセスしてしまうためと考えられる．最も距離が近いサーバを選択して接続した場合は，問合せ領域の範囲を広くするにしたがってアクセスされるサーバ数，経由するサーバ数ともに増加している．このとき，疎結合の上位サーバは，下位サーバよりもアクセス頻度が高くなっているものと考えられる．各サーバに対するアクセス数の偏りを調べるため，最も距離が近いサーバを選択したものについて，各サーバへのアクセス数の分散を求めた．結果を図 9 に示す．分散は最も小さい値に対する比で表示している．結果から分かるように，各サーバから遠い距離にあるオブジェクトに対する問合せが増えるにつれて，サーバ間の負荷の偏りが大きくなることが分かる．

最後に，データの追加および削除が発生した場合の更新処理において，アクセスされるサーバ数を調べた．結果から，明らかに密結合のみの方式においてアクセスされるサーバ数が多いことが分かる．密結合と疎結合を混合した方式が，疎結合のみの方式に比べて，アクセスされるサーバ数が多くなっているが，密結合のみの場合に比べると十分小さい値に抑えられている．これは更新されるデータの範囲が，サーバの MBB の大きさにほぼ等しいため，接続 DB の全ての要素を含む MBB の更新がそれほど頻繁に発生しないためであると考えられる。

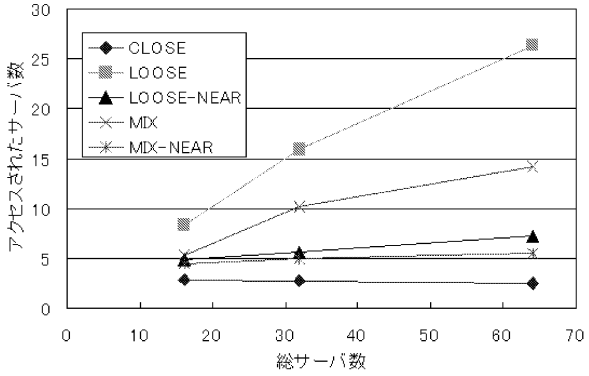

図 5. 問合せでアクセスされたサーバ数

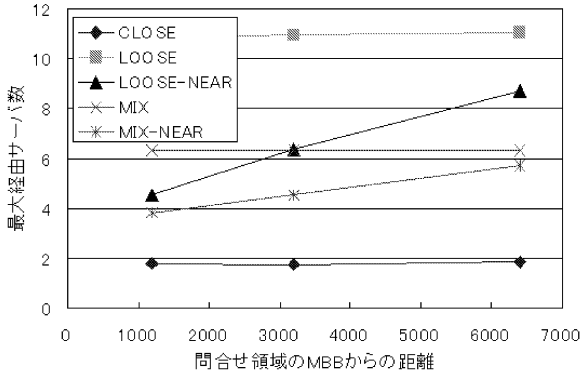

図 8. 問合せ領域を変更した場合の最大経由サーバ数

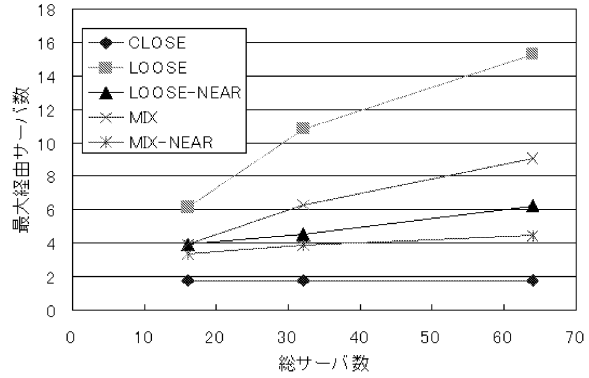

図 6. 問合せにおける最大経由サーバ数

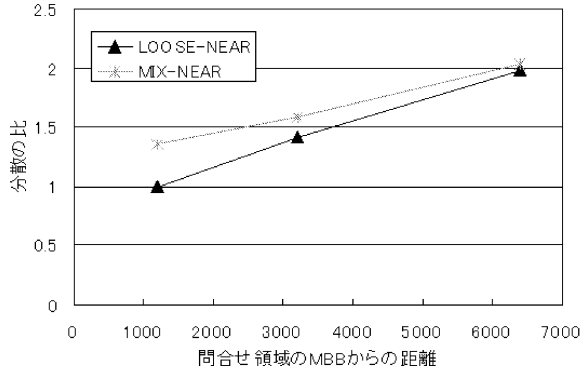

図 9. 問合せ領域を変更した場合のサーバ間の負荷の偏り

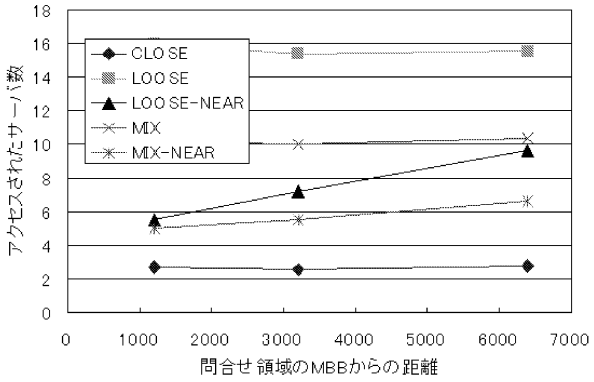

図 7. 問合せ領域を変更した場合にアクセスされたサーバ数

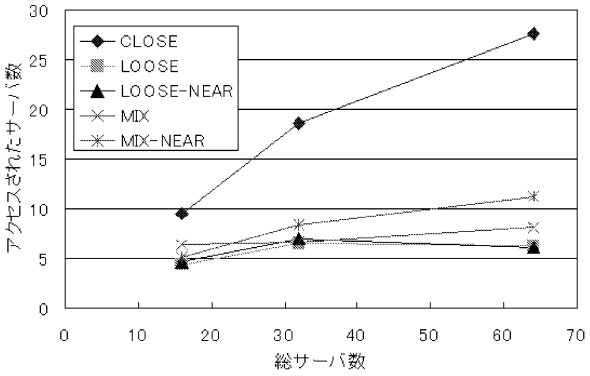

図 10. データ更新でアクセスされたサーバ数

4.2 接続方式の拡張

実験結果から，提案方式は以下のような項目について考慮する必要がある．

接続するサーバの MBB 間の距離：空間データに対するアクセス手法の研究において述べられているように，各サーバの MBB をできるだけクラスタ化することにより，サーバにかかる負荷を低減できる．しかし，接続されるサーバの情報を前もって知ることは困難であるため，大幅な接続変更を行わない限り，最適な接続方式の実現は難しいものと考えられる．運用面の容易さやサーバにかかる負荷を考慮して，できるだけ接続関係の再構成を行わずに，サーバの MBB 間の距離をより小さくする接続方法について検討する必要がある．

アクセス傾向：問合せにおいてアクセスされるサーバ数および経由するサーバ数は，密結合のみの場合が良い結果を示した．また，問合せの領域の中心と各サーバの MBB の中心の距離を大きくすると，疎結合を利用した方式ではアクセスするサーバ数および経由するサーバ数がともに増加した．したがって，アクセス傾向に偏りがある場合や，問合せ発生頻度がデータ更新頻度より高いサーバが存在する場合には，比較的距離が離れている場合でも密結合によって接続する方が良い結果を示すものと考えられる．そのため，各サーバのアクセス傾向も考慮する必要がある．

サーバの処理能力：問合せの領域の中心と各サーバの MBB の中心の距離を大きくすると，疎結合を利用した方式ではサーバ間の負荷の偏りが大きくなった．疎結合の上位に接続されるサーバの負荷は下位のものよりも高くなる可能性があるため，各サーバに対するアクセス頻度を考慮した上で，サーバの処理能力に合わせて接続形態を決定する必要がある．

キャッシュの利用：密結合のみの方式と比較して，疎結合を利用すると経由するサーバ数が多くなるため，問合せの応答時間が長くなる．問合せの応答時間を短縮するためにキャッシュの利用が考えられる．データそのものをキャッシュした場合，更新頻度が高いとデータの新鮮度が低下してしまう．疎結合を利用した方式では，サーバがクラスタ化されていれば，疎結合の上位のサーバが保持する情報は更新頻度が比較的低くなると考えられるため，データだけではなく，経由するサーバの情報のキャッシュを考慮したキャッシュ戦略について検討する必要がある．

5 おわりに

本稿では，異なる組織で運用されている位置情報データの共有方式について述べた．基本的な接続方式を提案し，シミュレーション実験によって提案方式を用いた場合にアクセスされるサーバ数や経由するサーバ数について調べた．シミュレーションの結果から，提案した接続方式においてデータ共有を考える上で考慮すべき点について考察した．今後は，4 章で述べた内容に基づいて提案方式を拡張し，拡張方式の評価を行う必要がある．

謝辞

本研究は，日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」の研究助成によるものである．ここに記して謝意を表す．

参考文献

[1] Amann, B.: Integrating GIS components with mediators and CORBA, Technical Report 97-09, Cedric-CNAM, CNAM, Paris (1997).

[2] Carey, M. J., DeWitt, D. J., Franklin, M. J., Hall, N. E., McAuliffe, M. L., Naughton, J. F., Schuh, D. T., Solomon, M. H., Tan, C. K., Tsatalos, O. G., White, S. J. and Zwilling, M. J.: Shoring up persistent applications, *Proceedings of the ACM SIGMOD Conference*, pp. 383–394 (1994).

[3] DeWitt, D. J., Kabra, N., Luo, J., Patel, J. M. and Yu, J.-B.: Client-server Paradise, *Proceedings of the 20th VLDB Conference*, pp. 558–569 (1994).

[4] Gaede, V. and Günther, O.: Multidimensional access methods, *ACM Computing Surveys* Vol. 30, No. 2, pp. 170–231 (1998).

[5] J. Li, D. R. and Srivastava, J.: Algorithms for loading parallel grid files, *Proceedings of the ACM SIGMOD Conference*, pp. 347–356 (1993).

[6] Kamel, I. and Faloutsos, C.: Parallel R-trees, *Proceedings of the ACM SIGMOD Conference*, pp. 195–204 (1992).

[7] Medeiros, C. B. and Pires, F.: Databases for GIS, *ACM SIGMOD Record*, Vol. 23, No. 1, pp. 107–115 (1994).

[8] Patel, J., Yu, J., Kabra, N., Tufte, K., Nag, B., Burger, J., Hall, N., Ramasamy, K., Lueder, R., Ellmann, C., Kupsch, J., Guo, S., Larson, J., Witt, D. D. and Naughton, J.: Building a scaleable geo-spatial DBMS: technology, implementation, and evaluation, *Proceedings of the ACM SIGMOD Conference*, pp. 336–347 (1997).

[9] 竹内奏吾, 中村嘉志, 多田好克: インターネットにおける地理位置情報システムの設計と実装, 情報処理学会マルチメディア，分散，協調とモーバイル シンポジウム (DICOMO'99) 論文集, pp. 405–410 (1999).

[10] 渡辺恭人，大西孝義，佐藤雅明，植原啓介，村井純: GLI システムの改良と実証実験，第 3 回プログラミングおよび応用のシステムに関するワークショップ (SPA2000)，URL: http://www.jaist.ac.jp/SPA2000/proceedings/ (2000).