

ラウンドロビン法で分割された分散地理データベースの

6 X - 2

並列 Spatial Join 演算の高速化

田村 慶一, 金子 邦彦*, 牧之内 顕文*

九州大学大学院システム情報科学府, 九州大学大学院システム情報科学研究院*

1. はじめに

2つの空間データの集合を空間的な関係で結び付ける演算を Spatial Join という。Partition parallelism を用いた並列 Spatial Join がこれまでに幾つか^[1]提案されている。Spatial Join では、複数の処理方式が考えられる時、最もベストな方式を選べるように、コストの見積もりを行う^[2]。並列 Spatial Join の処理時間の高速化のために、何らかの形でコストを見積もり、コストによりベストな方式を選択したい。本稿では、並列 Spatial Join のためのコスト見積もりについて扱う。

我々は、並列 Spatial Join のコストをアクセスするページ数で見積もる。データベースシステムでは、データはディスク中に格納されている。ディスクからのデータの読み出しの単位をページという。Spatial Join の2つの入力データの数や、問い合わせの条件によってアクセスするページ数を近似的に見積もる。実際の実行時にアクセスするページ数はデータベースシステムにおいて取り出せ、見積もりとの比較をし、場合によっては見積もりの修正を行うことができる。

2. 並列 Spatial Join とコスト見積もり

我々は、従来の研究と同様に partition parallelism を使い、並列処理を行っている。空間データの分割方法はラウンドロビン法を用い、予め空間インデックスを作成しておく。Spatial Join では、「2つの空間データが交差するものを取り出す」や「2つの空間データの間の距離がある距離のものを取り出す」といった演算を扱うため、一方の分割したデータ1つと他方の分割したデータすべてを比較して結果を求める。よって、分割したデータ毎に空間インデックスを作成すると共に、分割していないデータの空間インデックスを作成する。

空間インデックスとして R*-tree を用いた時の Spatial Join の処理は、Filter Step と Refinement Step の2つのステップから構成される。Filter Step では R*-tree を用いて MBR がある空間的な関係を満たすすべてのペアを結果の候補として取り出す。次に Refinement Step では実際の空間データの形状を調べ結果の候補から、空間的

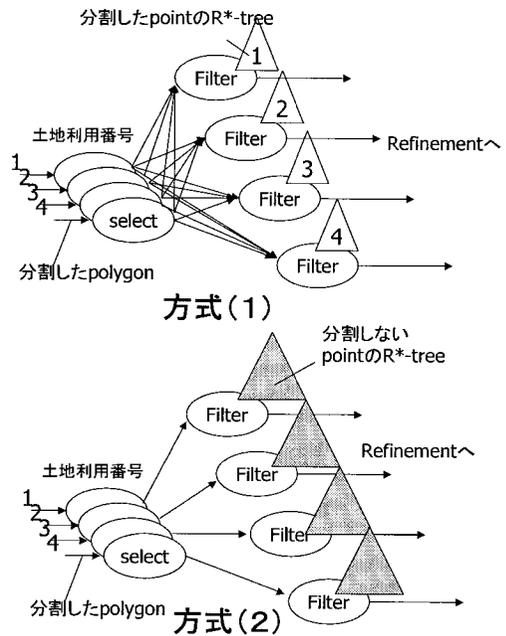


図 1: 方式 (1) と方式 (2)

な関係を満たさないものを取り除く。

Spatial Join の2つの入力のうちどちらか一方に空間インデックスが存在するような operation を本研究では研究の対象としている。分割したデータ毎に作成した R*-tree を使用する方法 (方式(1)とする) と、分割していないデータで作成した R*-tree を使用する方法 (方式(2)とする) の2つが考えられる。polygon データを土地利用番号で selection した結果を、point データと Spatial Join するような場合 (拡張 Sequoia 2000 ベンチマーク Query10), 処理の概観は図1のようになる。

Spatial Join の2つの入力空間データを A と B とする。A の方に R*-tree が存在し、A の個数を N 、B の個数を M とする。コストを $cost = PR + PB + C$ と見積もる。PR はアクセスされる R*-tree のページ数、PB はアクセスされる B のページ数である。C は Refinement

Performance Evaluation of Parallel Processing of Spatial Join on Distributed Geographical Databases using Round Robin Partition
 Keiichi Tamura, Kunihiko Kaneko and Akifumi Maki-nouchi
 Graduate School of Information Science and Electrical Engineering, Kyushu University

表 1: Filter Step におけるページアクセス数

方式 (1)	土地利用番号 11				土地利用番号 0			
	サイト 1	サイト 2	サイト 3	サイト 4	サイト 1	サイト 2	サイト 3	サイト 4
R*-tree 部分	255	256	256	253	69	69	66	71
データ 部分	1108	1108	1108	1108	24	24	24	24
方式 (2)	サイト 1	サイト 2	サイト 3	サイト 4	サイト 1	サイト 2	サイト 3	サイト 4
R*-tree 部分	680	670	675	682	101	101	100	97
データ 部分	277	277	277	277	6	6	6	6

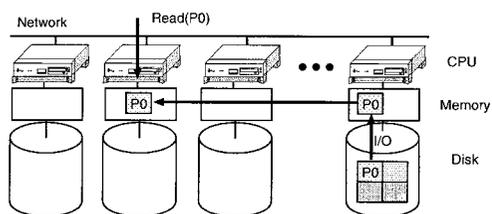


図 2: I/O, 通信と CPU コスト

Step でアクセスされるページ数である。使用する計算機の数とデータの分割の数を n とすると、各計算機でのコストは方式 (1) では $cost(1) = PR/n + PB + C/n$, 方式 (2) では $cost(2) = PR + PB/n + C/n$ となる。

並列データベースでは I/O コスト, 通信コストと CPU コストを足し合わせたものが問い合わせのコストである。あるページがアクセスされるとデータが存在する計算機でページが読み出され転送される (図 2)。I/O と通信コストはアクセスするページ数が増加するほど大きくなる。Filter Step では MBR の比較を行い比較回数が増えるほど CPU コストが大きくなる。MBR の比較回数は実装方法によって変わるが、MBR の比較を行ったデータの総数に比例すると考えた。MBR の比較を行うデータの総数はアクセスするページ数が増加するほど大きくなる。

R*-tree のページ数は葉ノードのノード数で近似できると仮定すると、 $1 < PR < N/m$ (m は R*-tree のエン트리数) で、 PB は $PB = M/e$ (e はページに格納できる B のオブジェクトの数) となる。この時、 $PR < PB$ が $eN < mM$ の時に成り立ち、 $PR > PB$ が $eN > mM$ の時に成り立つ。よって、 $eN > mM$ の時、方式 (1) を選択した方がアクセスするページ数が少なくなり、逆に $eN < mM$ の時、方式 (2) を選択した方がアクセスするページ数が少なくなると予想される。

3. 実験

拡張 Sequoia 2000 ベンチマークの Query10 を用いて Filter Step でアクセスするページ数の測定を行い、コストの見積もりとの比較を行う。Sun Microsystems UL-

TRA10 (CPU UltraSPARC 440MHz, Memory 1024 Mbyte, Disk 34GB Seek Time : 9ms 7200rpm buffer:2048KB × 6 RAID5) を 1 台, Sun Microsystems ULTRA5 (CPU UltraSPARC 270MHz, Memory 128 Mbyte, Disk 30GB Seek Time: 9.5ms 7200rpm buffer:512KB) を 4 台使用する。ネットワークは 100M イーサネットで、データベースシステムは、我々の研究室で開発を行った「出世魚」を使用した。使用したデータの総数は、point データが 30636 個、polygon データが 374428 個である。検索の条件は、LANDUSE (土地利用番号) = 11 (データ総数は 31892 個) と 0 (データ総数 540) である。計算機を 4 台使用して並列 Spatial Join を行う。

表 1 に測定結果を示す。2 つの方式でのページアクセス数の違いは Filter Step でのページアクセス数であるので、Filter Step のみの結果の示している。結果から polygon のオブジェクトは 1 ページに約 22 個格納されていることが分かり、 $e = 22$ で使用した R*-tree の葉ノードのエントリ数は $m = 150$ である。この値をコスト見積りに当てはめると土地利用番号が 11 の場合は、 $eN < mM$ が成り立ち、土地利用番号 0 では $eN > mM$ が成り立つ。土地利用番号 11 では方式 (2) の方がページアクセス数が少なくなっており、見積もりと一致する。同様に土地利用番号 0 では方式 (1) の方がページアクセス数が少なく、見積もりと一致する。

4. おわりに

研究対象としている Spatial Join の並列処理でアクセスされるページ数の見積もりと、実際の実験により得られたページアクセス数とが一致していることを確認した。

参考文献

- [1] Jignesh P, Jie-Bing Y, Navin K, Kristin T, Biswadeep N, Josef B, Nancy H, Karthikeyan R, Roger L, Curt E, Jim K, Shelly G, David D, and Jeffrey N. Building a scaleable geo-spatial dbms: Technology, implementation, and evaluation. SIGMOD 1997, pp. 336-347.
- [2] Shashi S, Sanjay C, Siva R, Andrew F, Xuan L, Chang-tien L: Spatial Databases - Accomplishments and Research Needs. TKDE 11(1), pp. 45-55, 1999.