IP-SAN 統合型 PC クラスタにおける並列相関関係抽出の実行

原 明日香　(指導教員：小口 正人)

1　はじめに

近年、発達する情報化社会では、データの蓄積と運用が非常に重要になってきている。また、情報システムにおいて処理されるデータ量が膨大になってきている。ユーザにとって重要なデータが蓄積されているにも関わらず、使いこなせていない場合が少なくない。

本研究では膨大なデータから有益な規則や関係を抽出する相関関係抽出において代表的な Apriori アルゴリズムと FP-growth アルゴリズムを利用し、両者の比較を行う。パラメータ条件によっては相関関係抽出における計算量、データ処理量は非常に多くなるため、並列化が不可欠となる。そこで分散メモリ型並列計算機である PC クラスタを用いるが、その際にバックエンドのストレージネットワークを統合した IP-SAN 統合型 PC クラスタを用いて実行し、性能評価を行う。

2　IP-SAN 統合型 PC クラスタ

各ノードが独立して動作する CPU、メモリ、二次記憶を保有し、ノードが必要に応じてネットワークを介し互いに通信することで全体として並列分散処理を実現する分散メモリ型並列計算機の各ノードに汎用のパーソナルコンピュータとネットワークを用いたものを PC クラスタ [図 1] という。Front-end(ノード間通信) は LAN、Back-end(ストレージアクセス) は SAN でネットワーク接続されている。

そこで図 2 のように Front-end と Back-end のネットワークを同じ IP ネットワークに統合した IP-SAN 統合型 PC クラスタの実現を考えた。ネットワークを統合することでネットワーク構築コストと管理コストの削減が可能となる。

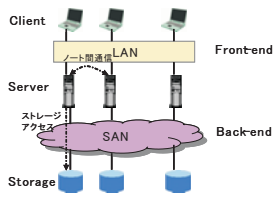

図 1: 通常の PC クラスタ

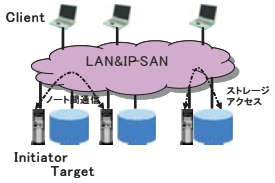

図 2: IP-SAN 統合型 PC クラスタ

3　相関関係抽出とその並列化

相関関係抽出では、巨大なデータから有益な規則性や関係を抽出するためにあるパターンが現れる頻度 (サポート値) を調べる。その頻度が多ければ、そのパターンから得られる関係は有意義なデータとなり、販売戦略などに活用出来る。

相関関係抽出で扱うデータはしばしば巨大であるため、データベースを分散し計算処理を並列化して、多数台のコンピュータをネットワークで接続した PC クラスタなどの環境でマイニング処理を実行する並列相関関係抽出の研究が行われている [1]。以下に相関関係抽出の代表的な 2 つのアルゴリズムの概要を説明し、本研究で用いる並列化アルゴリズムを紹介する。

3.1　Apriori アルゴリズム

1994 年に Agrawal らによって提案されたもので、発見された頻出アイテムセットから候補アイテムセットを生成し、繰り返し数え上げを行っていくアルゴリズムである。1 回目のデータベーススキャンで長さ 1 の頻出 1 アイテムセットを抽出し、それらを元に長さ 2 の候補 2 アイテムセットを生成する。2 回目のデータベーススキャンで候補 2 アイテムセットから頻出 2 アイテムセットを抽出する。これを候補アイテムセットが生成されなくなるまで繰り返していくことで、頻出パターンをすべて発見していくアルゴリズムである。

Apriori アルゴリズムには、候補アイテムセットを格納するために大容量のメモリが必要となる、何度も繰り返しデータベースをスキャンする可能性があるといった問題点がある。

Apriori をベースにした並列相関関係抽出のアルゴリズムはいくつか提案されているが、本研究ではハッシュ関数を使用して Apriori を並列化する HPA (Hash Partitioned Apriori)[2] を用いる。

3.2　FP-growth アルゴリズム

2000 年に Han らによって提案された [3] もので、巨大なトランザクションデータベースから相関関係抽出に必要な情報をコンパクトに圧縮したデータ構造である FP-tree を利用している。候補パターンを生成せずに頻出パターンを抽出することで、Apriori アルゴリズムの問題点を改善したアルゴリズムである。

FP-tree は次のように構築される。1 回目のデータベーススキャンで、各アイテムのサポート値を求め、頻出アイテムを抽出する。抽出された頻出アイテムをサポート値により、頻度が降順になるように並び替え (これを F-list とする)、空 (null) のラベルを持つ木のルートを作成 (これを T とする) する。2 回目のデータベーススキャンで、F-list に従ってトランザクションから頻出アイテムを抽出し、ソートする。T が F-list の要素である子を持っていれば、その子のカウントを 1 増やし、持っていなければ、新しくカウント 1 を持つ子を作る。F-list の最後までこの操作を繰り返す。FP-tree の構築例を図 3 に示す。

TID	Items	Frequent Items
100	f,a,c,d,g,i,m,p	c,f,a,m,p
200	a,b,c,f,l,m,o	c,f,a,b,m
300	b,f,h,j,o	f,b
400	b,c,k,s,p	c,b,p
500	a,f,c,e,l,p,m,n	c,f,a,m,p

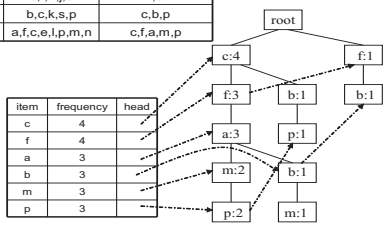

図 3: FP-tree の構造例

FP-growth は構築された FP-tree の性質を利用することにより、頻出パターンを発見していくアルゴリズムである。FP-growth の並列相関関係抽出のアルゴリズムは、本研究では HPA を元に行われた既存研究 [4] で提案された PFP (Parallelized FP-growth) を用いる。

4 実験内容と実験環境

HPA アルゴリズムと PFP アルゴリズムの並列化プログラムを、ローカルデバイス (SCSI ディスク) を用いた PC クラスタ、IP-SAN 統合型 PC クラスタ、バックエンド IP-SAN を用いた非統合型 PC クラスタで実行し、そのときの実行時間をそれぞれ測定する。実験には 4 台の PC を Gigabit Ethernet で接続した PC クラスタを用いる。IP-SAN を用いる場合には、iSCSI ターゲット用の PC がもう 4 台接続されている。各 PC は CPU が Pentium4 1.5GHz、メインメモリが 384MB、OS が Linux 2.6.9-1667(Fedora core 3) である。

5 実行結果と考察

アイテム数を 1000 とし、トランザクション数が 10K、20K、40K、80K、160K のトランザクションデータを用い、最小サポート値を 0.7 %としたときの実行時間を、ローカルデバイスを用いた PC クラスタ、IP-SAN 統合型 PC クラスタ、バックエンド IP-SAN を用いた非統合型 PC クラスタ上で測定した。今回用いた PFP プログラムの性質により、比較的小規模なトランザクションデータを用いて実験を行った。

図 4 に同じ大きさのトランザクションデータにおける 2 つのアルゴリズムの実行時間を示す。明らかに PFP アルゴリズムの方が速いことがわかる。これは、HPA アルゴリズムが頻出アイテムセットから候補アイテムセットを作るという動作を、データの大きさに関係なく何度も繰り返し行っているためである。反対に PFP アルゴリズムはデータの大きさによって、FP-tree の大きさが変わるため、データ量が小さい今回の実験では、良い結果が出たと考えられる。

図 5 に HPA アルゴリズム、図 6 に PFP アルゴリズムのそれぞれのクラスタにおける実行時間を示す。どのクラスタにおいても実行時間はほとんど変わらなかった。IP-SAN 統合型 PC クラスタは、バックエンドとフロントエンドで同じネットワークを使用するため性能が落ちる可能性が予想されたが、本実験で性能は変わらないということが分かった。

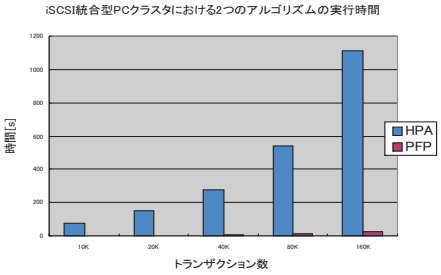

図 4: トランザクションデータごとの 2 つのアルゴリズムの実行時間

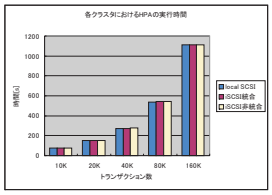

図 5: HPA アルゴリズムの実行結果

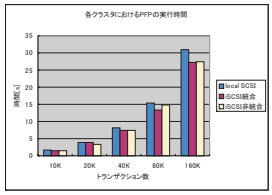

図 6: PFP アルゴリズムの実行結果

6 まとめと今後の課題

Apriori の並列化アルゴリズム HPA と FP-growth の並列化アルゴリズム PFP を、ローカルデバイスを用いた PC クラスタ、IP-SAN 統合型 PC クラスタ、バックエンド IP-SAN を用いた非統合型 PC クラスタにおいて実行し、実行時間を比較した。HPA アルゴリズムと PFP アルゴリズムの比較では、今回用いたトランザクションデータが小規模であったということもあり、今回の実験では PFP アルゴリズムの方が格段に速いという結果になった。

ただし、トランザクションデータ数を多くすると、FP-tree が大きくなってしまい PFP アルゴリズムは今回の実験環境ではうまく動作しなかったが、その場合も HPA アルゴリズムはきちんと動作していた。このことからデータ量と実行環境に応じて使用するアルゴリズムを使い分ける必要があると考えられる。

ローカルデバイスを用いた PC クラスタ、IP-SAN 統合型 PC クラスタ、バックエンド IP-SAN を用いた非統合型 PC クラスタの比較では、どちらのアルゴリズムにおいても、3 つの PC クラスタにおける実行結果はほとんど変わらなかった。同じ性能を示すなら、コストが安く、ネットワークの管理のしやすい IP-SAN 統合型 PC クラスタが有効であると考えられる。

参考文献

[1] 福田剛志、森本康彦、徳山剛志:”データマイニング”, 共立出版

[2] 小口正人、喜連川優:”ATM 結合 PC クラスタにおける動的リモートメモリ利用方式を用いた並列データマイニングの実行”, 電子情報通信学会論文誌,Vol.J84-D-I, No.9, pp.1336-1349, 2001 年 9 月

[3] Jiawei Han, Jian Pei, and Yiwen Yin:”Mining Frequent Patterns without Candidate Generation”,ACM SIGMOD2000,pp.1-12,May2000

[4] Iko Pramudiono and Masaru Ksuregawa: ”Tree structure based Parallel Frequent Pattern Mining on PC cluster”, DEXA2003,pp.537-539,September 2003

[5] 原明日香、神坂紀久子、小口正人:”IP-SAN 統合型 PC クラスタにおける相関関係抽出の実行”, 情報処理学会第 69 回全国大会,5S-5,2007 年 3 月発表予定