並列データマイニング実行時の IP-SAN 統合型 PC クラスタのネットワーク特性解析

原　明日香[†]　神坂　紀久子[††]
山口　実靖[†††]　小口　正人[†]

近年発達する情報化社会において、処理される情報量が爆発的に増大しており、その中からユーザが必要とする情報を高速に取り出すことが求められている。そこで膨大なデータを処理するために、フロントエンドとバックエンドのネットワークを同一の IP ネットワークに統合した IP-SAN 統合型 PC クラスタを構築した。ただしアプリケーション実行時にクラスタのノード間通信や I/O がどのように振舞いシステム性能に影響を与えているのかなど詳しい解析は行われていない。そこで本研究では、複数 Initiator から単数 Target にアクセスすることでネットワークおよび Target に高負荷をかけた環境において、相関関係抽出などのデータマイニングを実行し、システムのモニタを行い、解析することによって、IP-SAN 統合クラスタの詳しい振舞を明らかにする。

Analysis characteristics of IP-SAN consolidated PC cluster when parallel datamining is executed

Asuka Hara,[†] Kikuko Kamisaka,[††] Saneyasu Yamaguchi[†††] and Masato Oguchi[†]

In the information society that has developed in recent years, the processed volume of data increases explosively. Information required from users is requested to be taken out immediately from them. In order to process huge data, we have constructed IP-SAN consolidated PC cluster that integrated the front-end and the back-end network into the same IP network. However, the analysis how the communication between nodes of the cluster and the execution of I/O influences the behavior of system performance, when the application is executed, is not performed in detail. Thus, in this research, the network and Target is burdened with a heavy load by accessing a single target with multiple initiators. The parallel association rule mining is executed, while the system is monitored and analized, so that detailed behavior of IP-SAN consolidated PC cluster is clarified.

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。そこでデータマイニングの中で、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注目した。相関関係抽出のためのアルゴリズムとして代表的なものに、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 Apriori アルゴリズムと、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 を利用することで候補パターンを生成せずに頻出パターンを抽出する FP-growth アルゴリズムとがある。

パラメータの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。大量の情報を高速に処理するためには、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 PC クラスタと、増大する情報量を蓄積するための技術として登場した SAN(Storage Area Network) を併せて用いることが考えられる。

PC クラスタ上で上記の２つのアルゴリズムを実行

† お茶の水女子大学
Ochanomizu University
†† 情報通信研究機構 (NICT)
National Institute of Information and Communications Technology
††† 工学院大学
Kogakuin University

するためにはそれらのアルゴリズムを並列化しなければならない。PC クラスタ上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は多数行われている。その中で Apriori をベースと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) を用い、FP-growth の並列相関関係抽出アルゴリズムは HPA を元に行われた既存研究で提案された PFP(Parallelized FP-growth) を用いる。

通常 SAN を用いた PC クラスタではノード間通信を行うフロントエンドは LAN，ストレージアクセスを行うバックエンドは SAN でネットワーク接続されている。これに対し我々は、ネットワーク構築コストと管理コストの削減を目指し、フロントエンドとバックエンド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 クラスタの実現を検討している。

これまで使用してきたクラスタの実験においては、ネットワークに比べてローカルストレージの帯域幅が狭いため、IP-SAN 統合型 PC クラスタの性能が低下しないことが分かった。そこで、我々は以前より高性能なストレージ (SATA ディスクと RAID0 構成の SAS ディスク) を導入した IP-SAN 統合型 PC クラスタを新たに構築し、HPA アルゴリズムと PFP アルゴリズムをそれぞれローカルデバイスを用いた PC クラスタおよび Initiator と Target を 1 対 1 接続した IP-SAN 統合型 PC クラスタ上で実行した。そしてその実行時間を測定し、そのときのネットワークトラフィック、CPU 使用率、メモリ使用率などのモニタを行った。その結果、HPA においてはどのクラスタも実行時間はほとんど変わらず、PFP においては基本性能測定で格段に性能が良かった SAS ディスクを用いた PC クラスタの実行時間が必ずしも短くないことがわかった。また、ネットワークトラフィックのモニタリングから、どちらのアルゴリズムにおいてもどのクラスタもネットワークの帯域にはまだ余裕があり、iSCSI を用いた場合も、iSCSI のトラフィックはネットワークにはあまり大きな影響を与えていないことが分かった。

iSCSI の一般的は用いられ方としては、数台の Initiator が 1 台の Target に対しアクセスする形が多いと考えられる。そこで、本研究では、Initiator と Target の接続台数を変化させ、これまでよりよりネットワークに高負荷をかけたときの、IP-SAN 統合型 PC クラスタの振舞を観察し、解析を行っていく。

2. IP-SAN 統合型 PC クラスタ

2.1 PC クラスタにおける SAN の利用

近年、情報システムにおいて処理されるデータの量が膨大になってきたことから、ストレージ分野においてネットワークストレージ技術が発展し、サーバとストレージを結ぶネットワークである SAN が登場して、普及するようになった。HPC 分野では、PC クラスタの記憶装置において、計算ノードーストレージ間のバックエンドのネットワークに SAN を用いることが多くなっている。SAN は、分散したストレージをネットワークで統合し、集中管理とディスク資源の効率的な活用を可能にしている。

図 1 は、SAN を用いて構築した PC クラスタの例である。現在、SAN としては、高速な専用回線である Fibre Channel を用いる FC-SAN が普及している。一般に、ディスクへの I/O 処理を行うストレージアクセスはノード間通信と比べてバースト性が高く、転送データ量が多いため、計算ノード (サーバ) ーストレージ間のバックエンドには高速な FC-SAN を用いることが多い。しかし FC-SAN では、FC 用のスイッチが高価であることなど、PC クラスタに導入して管理するにはコスト面で障害がある。

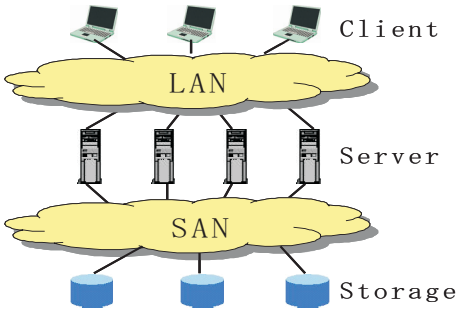

図 1 SAN を用いた PC クラスタ

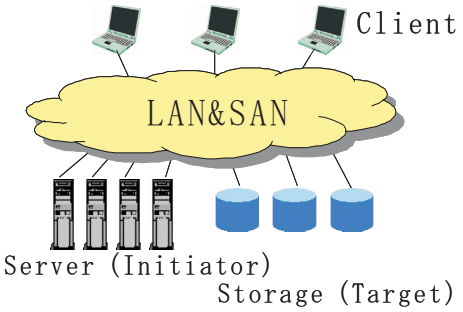

図 2 IP-SAN 統合型 PC クラスタ

IP-SAN は、TCP/IP ネットワークで構築する次世

代の SAN である。図 1 に示すように、FC を用いて構築する従来の SAN に代わり、バックエンドのネットワークを IP-SAN で構築することにより、安価なコストで PC クラスタのストレージを導入、運用が出来る。今後、Gigabit Ethernet/10Gigabit Ethernet が広く普及していくであろうことを考慮すると、IP-SAN をバックエンドに持つ PC クラスタが使用されるようになると考えられる。

2.2 IP-SAN 統合型 PC クラスタと性能への懸念

我々は、図 2 に示すように、計算ノード (サーバ) ー ストレージ間のバックエンドネットワークを、ノード間を接続するフロントエンドに統合した iSCSI 接続の IP-SAN 統合型 PC クラスタを提案し、評価を行っている。iSCSI(Internet SCSI) は、2003 年 2 月に IETF により正式認証された IP-SAN のプロトコルであり、SCSI コマンドを TCP/IP パケットの中にカプセル化することでブロックレベルのデータ転送を行う。iSCSI の階層構造は、図 3 のようになっている。

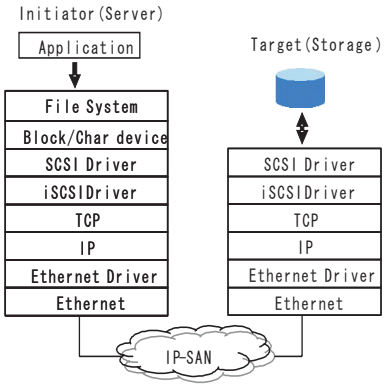

図 3 iSCSI の階層構造

SAN を使用した PC クラスタでは、一般にフロントエンド LAN とバックエンド IP-SAN が別々になっているため、2 つの異なるネットワークの構築が必要になる。これに対し、IP-SAN 統合型 PC クラスタでは、iSCSI を使用することで、双方のネットワークを TCP/IP と Ethernet を用いたコモディティなネットワークに統一することができる。それにより、ネットワーク構築コストの削減と運用管理の効率化が可能となる。

しかし、IP-SAN 統合型 PC クラスタは、フロントエンドにおけるノード間通信とバックエンドのストレージアクセスにおけるバルクデータが、TCP/IP over Ethernet である同一のネットワーク経由で混在して転送される。そのため、フロントエンドとバックエンドのネットワークを個々に構築する非統合型と比較して、並列分散処理実行時のネットワークへの負荷が懸念される。例えば、ノード間通信とストレージアクセスで同じネットワークリソースを使用するため、互いに衝突する可能性がある。その結果、ストレージアクセスのバルクデータにより並列計算のためのノード間通信が多大な影響を受け、全体の性能が劣化する可能性が推測される。従って、バックエンドネットワークをフロントエンドネットに統合した IP-SAN 統合型 PC クラスタは、非統合型 PC クラスタと比較して、統合がどの程度性能に影響を及ぼすかを評価する必要がある。

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 [2]。まず 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。次に 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)[3] 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 [4]。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 の構造例を図 4 に示す。

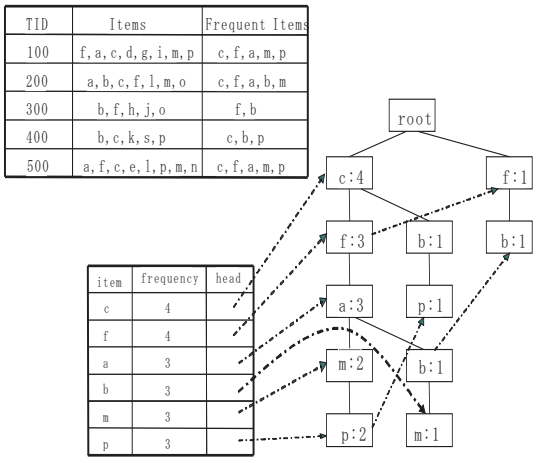

図 4 FP-tree の構造例

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 [5] で提案された PFP(Parallelized FP-growth) を用いる。

FP-growth アルゴリズムは Apriori アルゴリズムと比較して極めて高速であると言われている。ただしデータの性質によっては、FP-tree が巨大になってしまう可能性のある点が問題である。

4. 研究概要

4.1 既存研究

我々は、フロントエンドとバックエンドのネットワークを同じネットワークに統合することで、ノード間通信とストレージアクセスが互いに衝突するため、全体の性能が劣化する可能性があるのではないかと考えたが、文献 [6] で述べた実験においては、性能が劣化しないということがわかった。この実験に用いた PC クラスタの各ノードはデータ領域として SCSI ディスクを用いている。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。そこでネットワーク越しにストレージへアクセスする IP-SAN を使いながらも性能が落ちなかった原因を解明するため、ノード間通信とネットワークストレージアクセスを並行して複数のプロセスにより動作させ、ネットワークに高負荷をかけ、性能への影響を評価した [7]。その結果、ネットワークと比較してローカルストレージの帯域幅が低いことなどによりネットワーク帯域に余裕があるため、IP-SAN 統合型 PC クラスタの性能が落ちないということが分かった。

これらの実験を踏まえ、アクセス性能が良いとされる SAS ディスクを用い、データの読み書きを高速化する RAID0 構成とすることで、以前の実験環境よりもストレージ性能を強化した新しい IP-SAN 統合型 PC クラスタを構築した。これにはクラスタの構築・管理ツールである Rocks[8] を用いた。

新しく構築した環境において、並列相関関係抽出アルゴリズムをローカルデバイスを用いた PC クラスタおよび IP-SAN 統合型 PC クラスタ上でそれぞれ実行し、実行時間を測定し、その時の通信状況をモニタリングツールである Ganglia[9] を用いてネットワーク使用率、メモリ使用率をモニタリングした [10]。その結果、HPA においてはどのクラスタにおいても性能はほとんど変わらず、PFP においてはベンチマークで格段に性能が良かった SAS ディスクを用いたクラスタの性能がやや劣るということ、IP-SAN のトラフィックを統合してもネットワークの帯域にまだ十分余裕があることが分かった。

4.2 複数台 Initiator による Target アクセス

これまでの実験においては、IP-SAN 統合型 PC クラスタの Initiator と Target を 1 対 1 接続していたが、iSCSI の用いられ方を考えると、数台の Initiator が 1 台の Target にアクセスする形が一般的であると考えられる。その際には Target およびその接続ネッ

トワークにトラフィックが集中し、高い負荷がかかる。そこで本研究では、既存研究で構築したストレージ性能を強化した PC クラスタ上で、よりネットワークに高負荷がかかる環境を作り、並列相関関係抽出のアルゴリズムをローカルデバイスを用いた PC クラスタおよび IP-SAN 統合型 PC クラスタ上でそれぞれ実行する。その時の通信状況を Ganglia を用いて観察することで、IP-SAN 統合型 PC クラスタの動作解析を行う。具体的には以下の図のように Initiator と Target を n 対 1 で接続させることでネットワークに高負荷をかける。

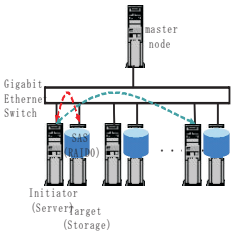

図 5 Initiator 対 Target:1 対 1

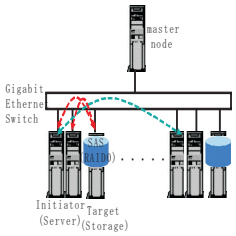

図 6 Initiator 対 Target:2 対 1

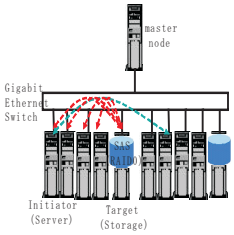

図 7 Initiator 対 Target:4 対 1

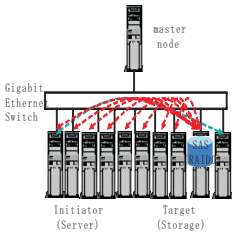

図 8 Initiator 対 Target:8 対 1

5. 実験結果と考察

5.1 Bonnie++

まず、Initiator と Target を 1 対 1、2 対 1、4 対 1、8 対 1 でそれぞれ接続することで、ネットワークに高負荷をかけた環境を構築した。そこで、SAS ディスク (RAID0 構成) であるローカルデバイス、SAS ディスク (RAID0 構成) を Target ストレージとして接続させた iSCSI のストレージについて、ハードディスクベンチマークツールの Bonnie++[11] を用いて、Sequential read アクセスと Sequential write アクセスを測定した。プラットフォームとしては、ローカルデバイスを用いた PC クラスタおよび IP-SAN 統合型 PC クラスタで実行し、実行時間をそれぞれ測定して、そのときのネットワークトラフィック、CPU 使用率、パケット I/O をモニタリングした。実験には 8 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 Target 用の PC がもう 8 台接続されている。iSCSI Initiator PC として、CPU が Intel Xeon 3.6GHz、メインメモリが 4GB、HDD が 250GB SATA、OS が Linux 2.6.18(CentOS 4.5)、iSCSI Target PC として、CPU がクアッドコア Intel Xeon 1.6GHz、メインメモリが 2GB、HDD が 73GB SAS × 2、OS が Linux 2.6.18(CentOS 4.5) であるマシンを使用した。また、iSCSI の Initiator として open-iscsi-2.0-865[12]、Target として iSCSI Enterprise Target(IET)-0.4.15[13] を使用した。

Target における Initiator からのアクセスの合計スループットを図 9 に示す。

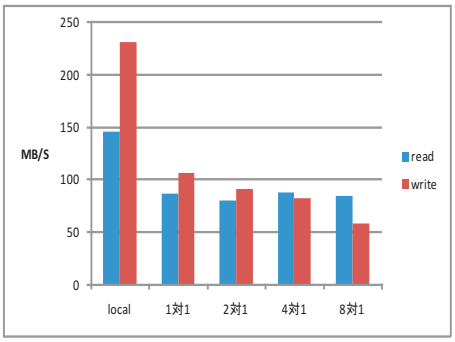

図 9 bonnie++による Sequential read アクセスと Sequential write アクセスの実行結果

この結果から、まずローカルアクセスと iSCSI アクセスを比較した結果、ローカル write が極めて速いことから、iSCIS write はその半分以下に留まっているが、ローカル read と比較して iSCSI read はその 2/3 近い性能が出ている。また、Sequential read アクセスの場合においては、1 対 1、2 対 1、4 対 1、8 対 1 のどの場合においても合計スループットは殆んど変わらず、Sequential write アクセスの場合においては、1 対 1、2 対 1、4 対 1、8 対 1 と負荷を高くしていくごとにスループットがやや低下するということがわかる。

図 10 に各接続方式における Initiator のネットワークトラフィック、図 11 に各接続方式における Target のネットワークトラフィックを示す。

これれの結果から、どの場合においても最大 60Mbytes/sec(約 500Mbps) のトラフィックしか流れ

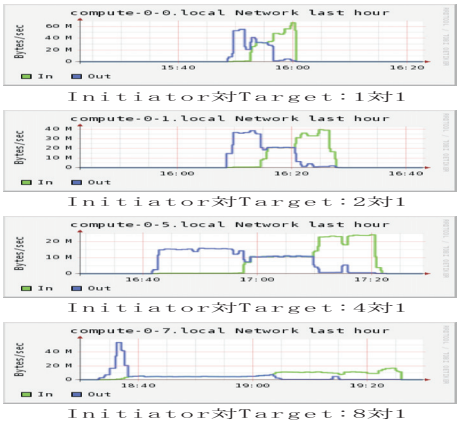

図 10 各接続方式における Initiator のネットワークトラフィック

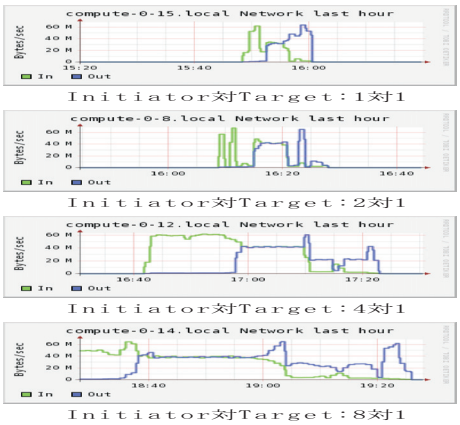

図 11 各接続方式における Target のネットワークトラフィック

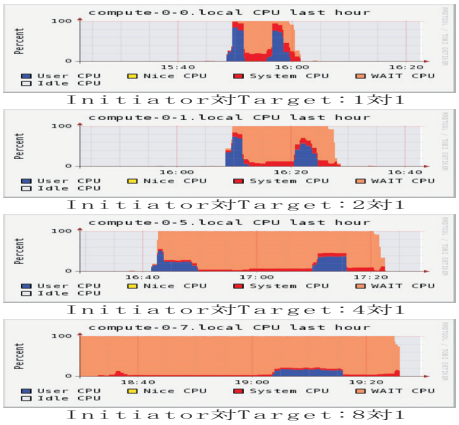

図 12 各接続方式における Initiator の CPU 使用率

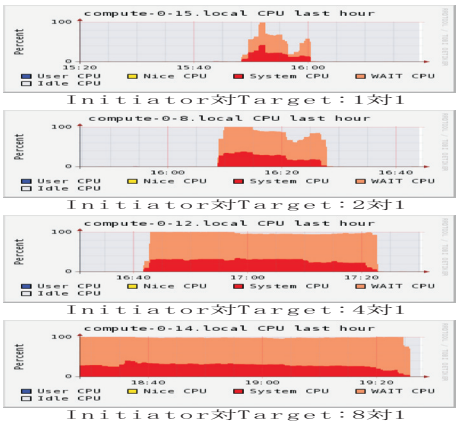

図 13 各接続方式における Target の CPU 使用率

ておらず、本実験では Gigabit Ethernet を使用しているため、ネットワーク帯域にはまだ余裕があることがわかる。1 台の Target が処理するデータ量が 1 対 1、2 対 1、4 対 1、8 対 1 と負荷を高くしていくごとに増加していくため、処理時間が長くなっているが、図 11 に示されるように Target のネットワークトラフィックの最大値はどの場合もほぼ一定であり、ネットワークにはそれ以上の負荷はかかっていないことがわかる。

図 12 に各接続方式における Initiator の CPU 使用率、図 13 に各接続方式における Target の CPU 使用率を示す。また、図 14 に各接続方式における Initiator の Packet I/O、図 15 に各接続方式における Target の Packet I/O を示す。

CPU 使用率の結果から、Initiator と Target のどちらとも、負荷が高くなるにつれて実行時間が長くなり、Target にアクセスが集中した場合においても CPU は実行時間に比べて待ち時間が長く、ストレージアクセスのための CPU 処理にはまだ余裕があると言える。これに対し、図 13 に示された Target の CPU の待ち時間もそれに伴い長くなっている。パケット I/O の結果から、パケット I/O は最大限に行われており、これがボトルネックとなっていると考えれる。

以上の結果から、Sequential read アクセスの場合においては、1 対 1、2 対 1、4 対 1、8 対 1 のどの場合においてもスループットは変わらず、Sequential write アクセスの場合においては、1 対 1、2 対 1、4 対 1、8 対 1 と負荷を高くしていくごとにスループットが低下するということがわかった。ネットワークにはまだ余裕があり、CPU も待ち時間が長い状態であるが、パケット I/O がボトルネックとなり、Target にアクセスを集中させた場合の Sequential write アクセスの性能が低下していると考えられる。

図 14　各接続方式における Initiator の packet I/O

図 15　各接続方式における Target の packet I/O

5.2　並列データマイニング

Bonnie++と同様に、Initiator と Target を 1 対 1、2 対 1、4 対 1、8 対 1 でそれぞれ接続することで、ネットワークに高負荷をかけた環境を構築した。そこで、HPA アルゴリズムと PFP アルゴリズムの並列化プログラムについて、アイテム数を 1000 とし、トランザクション数が 1M、2M、4M、8M 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 %として実行した。プラットフォームとしては、先の Bonnie++実験時と同じ環境を用いた。

図 16 に HPA アルゴリズム、図 17 に PFP アルゴリズム実行時の各クラスタにおける実行時間を示す。この結果から、HPA、PFP どちらのアルゴリズムにおいても、どのクラスタにおいても実行時間はほとんど変わらないことがわかる。

HPA アルゴリズム実行時の各接続方式における Initiator と Target のネットワークトラフィックをそれぞれ図 18 と図 19 に、PFP アルゴリズム実行時の各接続方式における Initiator と Target のネットワークトラフィックをそれぞれ図 20 と図 21 に示す。

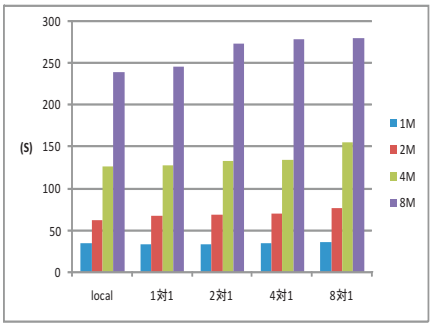

図 16　各接続方式における HPA の実行時間

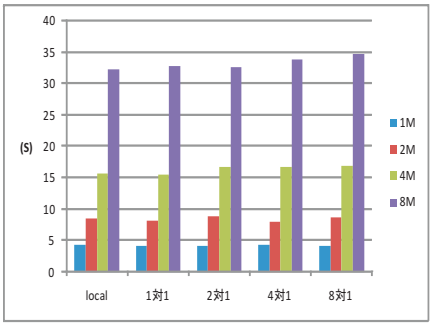

図 17　各接続方式における PFP の実行時間

これらの結果から、どちらのアルゴリズムにおいてもネットワークの帯域にはまだ余裕があることがわかる。iSCSI を用いた場合も、iSCSI のトラフィックはネットワークにあまり大きな影響を与えていない。バックエンドネットワークをフロントエンドネットに統合してもネットワークの帯域を使い切ることはなく、IP-SAN 統合型 PC クラスタが有効であることがわかる。

HPA アルゴリズム実行時の各接続方式における Initiator の CPU 使用率とメモリ使用率をそれぞれ図 22 と図 23 に、PFP アルゴリズム実行時の各接続方式における Initiator の CPU 使用率とメモリ使用率をそれぞれ図 24 と図 25 に示す。また、HPA アルゴリズム実行時の各接続方式における Initiato と Target のパケット I/O をそれぞれ図 26 と図 27 に、PFP アルゴリズム実行時の各接続方式における Initiato と Target のパケット I/O をそれぞれ図 28 と図 29 に示す。

これらの結果から、HPA においては何回もデータベーススキャンを繰り返し行うため、Initiator の CPU 使用率が高くなっており、同時にパケット I/O の量も多くなっていることがわかる。PFP においては、計

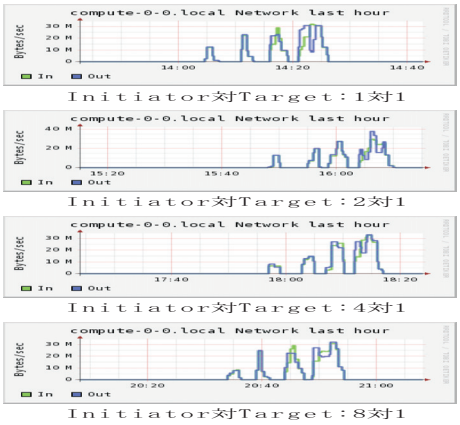

図 18 各接続方式におけるの HPA 実行時の Initiator のネットワークトラフィック

図 19 各接続方式における HPA 実行時の Target のネットワークトラフィック

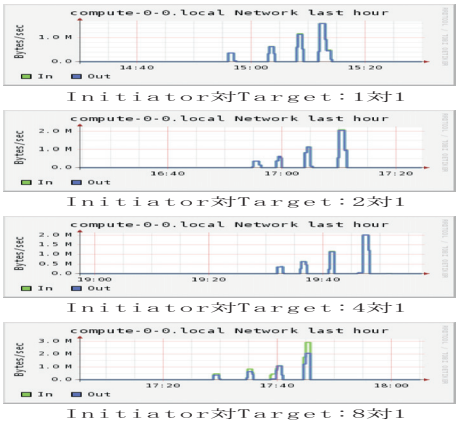

図 20 各接続方式における PFP 実行時の Initiator のネットワークトラフィック

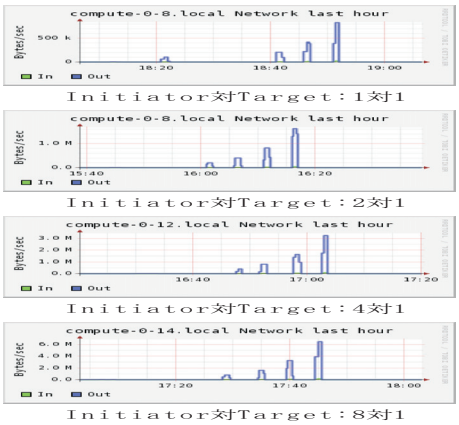

図 21 各接続方式における PFP 実行時の Target のネットワークトラフィック

算量が少ないため、あまり Initiator の CPU が使われておらず、パケット I/O も少ないことがわかる。また、どちらの場合においても、Target の CPU はほとんど使用されていない。

以上の結果から、HPA アルゴリズムと PFP アルゴリズムのどちらの場合においても、IP-SAN 統合型 PC クラスタにおいて Target にアクセスを集中させた場合の性能はほとんど変わらないということがわかった。また、ネットワークトラフィックのモニタリングにより、今回の実験では IP-SAN のトラフィックを統合してもネットワークの帯域にまだ十分余裕があることがわかった。これらは、今回使用した HPA のプログラム、PFP プログラムどちらの場合においても、ノード間通信やストレージアクセスといった処理だけでなく、これは計算処理が主に行われており、ノード間通信とストレージアクセスを行うパケットの衝突がネットワーク上であまり起こることがないためだと考えられる。したがって、IP-SAN 統合型 PC クラスタはどちらのプログラムを実行させたときでも有効であると言え、iSCSI に関しては複数台の Initiator を 1 台の Target に接続しても性能が落ちていないため、IP-SAN 統合型 PC クラスタの柔軟なシステム構成を実現することが可能である。

6. まとめと今後の課題

本研究では、相関関係抽出アルゴリズムである 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 クラスタおよびネットワークに高負荷をかけた IP-SAN 統合型 PC クラスタにおいてそれぞれ実行し、実行時間を測定して、その時のネットワークトラフィック、CPU 使用率、パケット I/O をモニタリングした。Initiator と Target を n 対 1 に接続することで、ネットワークに高負荷を与えた。ハードディスクベンチ

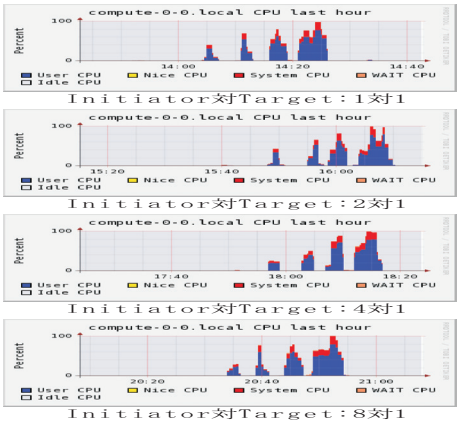

図 22 各接続方式における
HPA 実行時の Initiator の CPU 使用率

図 23 各接続方式における
HPA 実行時の Initiator のメモリ使用率

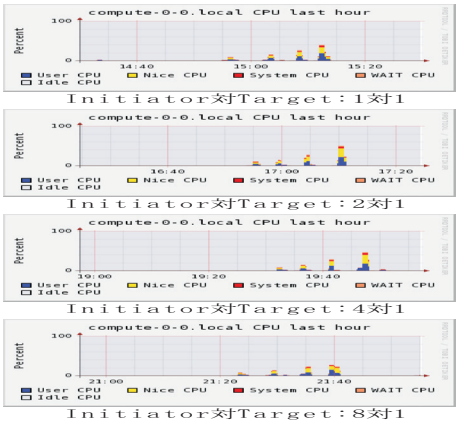

図 24 各接続方式における
PFP 実行時の Initiator の CPU 使用率

図 25 各接続形式における
PFP 実行時の Initiator のメモリ使用率

マークの結果から、Sequential read アクセスの場合においては、1 対 1、2 対 1、4 対 1、8 対 1 のどの場合においてもスループットは変わらず、Sequential write アクセスにおいては、1 対 1、2 対 1、4 対 1、8 対 1 と負荷を高くしていくごとにスループットが低下することがわかった。また、ネットワークトラフィック、CPU 使用率、パケット I/O のモニタリングから、ネットワーク帯域にはまだ余裕があり、CPU は待ち時間が長くなっていることから、パケット I/O の部分がボトルネックとなり、Sequential write アクセスにおいては、性能が少し劣化することがわかった。

それに伴い、並列データマイニングを実行させた際も高負荷を与えた場合において、IP-SAN 統合型 PC クラスタの性能が低下することが予想されたが、本研究の実験結果からネットワークに高負荷を与えた場合においても、IP-SAN 統合型 PC クラスタは並列データマイニングの実行で良好な性能を発揮していることがわかった。並列データマイニングにおいては、ノード間通信やストレージアクセスといった処理だけでなく、計算処理が多く行われており、ノード間通信とストレージアクセスを行うパケットの衝突がネットワーク上であまり起こることがないためだと考えられる。そのため並列データマイニング実行時には、我々が提案している IP-SAN 統合型 PC クラスタは有効であると言える。

今後は、他の様々なプログラムなどを実行させ、OS に対してカーネルモニタの導入などを行うことで、更なる IP-SAN 統合型 PC クラスタの詳細な振る舞いを解析し、最適化などを行っていきたい。

謝　　辞

本研究は一部，文部科学省科学研究費特定領域研究課題番号 18049013 によるものである．

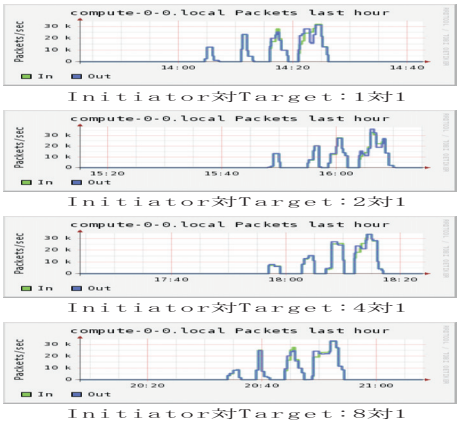

図 26 各接続方式における
HPA の実行時の Initiator のパケット I/O

図 27 各接続方式における
HPA の実行時の Target におけるパケット I/O

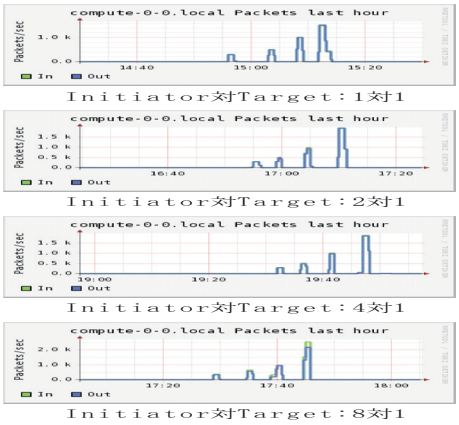

図 28 各接続方式における
PFP の実行時の Initiator のパケット I/O

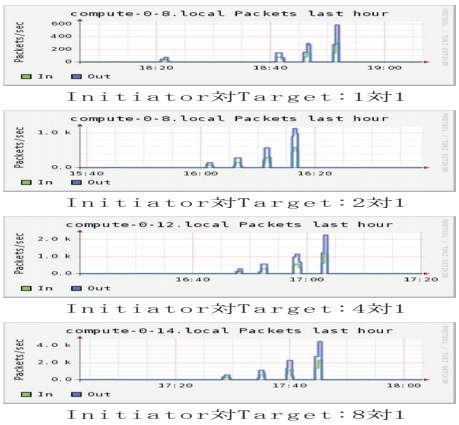

図 29 各接続方式における
PFP の実行時の Target におけるパケット I/O

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) R. Agrawal, T. Imielinski, A. Swami:”Mining Association Rules Mining between Sets of Items in Large Databades,” In Proceedings of the International Conference on Very Large Data Bases,pp.487-499,1994.
3) 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
4) J. Han, J. Pei, and Y. Yin:”Mining Frequent Patterns without Candidate Generation,”ACM SIGMOD2000,pp.1-12,May2000
5) Iko Pramudiono and Masaru Ksuregawa: ”Tree structure based Parallel Frequent Pattern Mining on PC cluster,” DEXA2003,pp.537-539,September 2003
6) 原明日香、神坂紀久子、小口正人:”並列相関関係抽出実行時の IP-SAN 統合型 PC クラスタの特性評価”, 分散、協調とモバイル (DICOMO2007) シンポジウム,2007 年 7 月
7) 神坂紀久子、山口実靖、小口正人:”IP-SAN 統合型 PC クラスタにおける複数プロセスによる同時アクセス時の性能評価”, 分散、協調とモバイル (DICOMO2007) シンポジウム,2007 年 7 月
8) Rocks Cluster:http://www.rocksclusters.org/
9) Ganglis Monitoring System:http://www.ganglia.info/
10) 原明日香、神坂紀久子、山口実靖、小口正人:”IP-SAN 統合型 PC クラスタにおける並列相関関係抽出実行時のシステム特性解析”, データ工学ワークショップ (DEWS2008),2008 年 3 月
11) Bonnie++:http://www.coker.com.au/bonnie++/
12) Open-iSCSI:http://www.open-iscsi.org/
13) iSCSI-Enterprise Target:http://sourceforge.net/projects/iscsitarget/