DEIM Forum 2009 E7-4

並列データ処理アプリケーション実行時の IP-SAN 統合型 PC クラスタの動作解析

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

† お茶の水女子大学　〒 112–8610 東京都文京区大塚 2–1–1
†† 情報通信研究機構　〒 184-8795 東京都小金井市貫井 4–2–1
††† 工学院大学　〒 163–8677 東京都新宿区西新宿 1–24–2
E-mail: †asuka@ogl.is.ocha.ac.jp, oguchi@computer.org, ††kikuko.kamisaka@nict.go.jp,
†††sane@cc.kogakuin.ac.jp

あらまし　近年、情報システム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、本研究ではバックエンドとフロントエンドのネットワークを統合した IP-SAN 統合型 PC クラスタを構築して利用した。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 の実行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、現実の環境を想定し、さまざまなデータ処理アプリケーションを実行中にデータバックアップ処理が動作するなどのケースにおいて、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。

キーワード　Storage Area Network，IP-SAN, iSCSI, PC クラスタ, 並列分散処理

Analyzing characteristics of PC cluster consolidated with IP-SAN in the execution of data-intensive applications

Asuka HARA[†], Kikuko KAMISAKA[††], Saneyasu YAMAGUCHI[†††], and Masato OGUCHI[†]

† Ochanomizu University　2–1–1 Otsuka, Bunkyo-Ku, Tokyo 112–8610 Japan
†† National Institute of Information and Communications Technology　4–2–1 Nukui, Koganei-city, Tokyo, 163–8677 Japan
††† Kogakuin University　1–24–2 Nishi-shinjuku, Shinjuku-ku, Tokyo, 163–8677 Japan
E-mail: †asuka@ogl.is.ocha.ac.jp, oguchi@computer.org, ††kikuko.kamisaka@nict.go.jp,
†††sane@cc.kogakuin.ac.jp

Abstract　In the information society that has been developing in recent years, the volume of processed 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 PC cluster consolidated with IP-SAN 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, various data processing applications are executed, while the system is monitored and analized, so that detailed behavior of PC cluster consolidated with IP-SAN is clarified.

Key words　Storage Area Network，IP-SAN, iSCSI, PC cluster, Parallel Distributed Processing

1. はじめに

近年、パーソナルコンピュータにおけるコモディティなハードウェアの価格低下と性能改善が進み、大規模な科学技術計算、データベース処理やデータマイニングのようなハイパフォーマンスなデータ処理アプリケーションが PC クラスタ上で実行されるようになっている。大規模な PC クラスタにおいては、Fibre Channel や InfiniBand のような高速な専用回線がノードとストレージの間のネットワークとしてよく使用されている。しかしながら、IP ネットワークをベースとした Storage Area

Network (IP-SAN) の出現により、コモディティな技術をベースとしたネットワークだけで PC クラスタの構築が可能となってきている。

そこで本研究では、IP-SAN の代表的なプロトコルである iSCSI を用いることでフロントエンド LAN とバックエンド SAN のネットワークを統合した IP-SAN 統合型 PC クラスタの提案し、システムの最適化を目指す。本稿においては、Target に接続する Initiator の数を変化させることにより、ネットワークと Target に高負荷をかけながら、ストレージのベンチマークと並列データマイニングおよび並列バイオインフォマティクスのアプリケーションを用いて IP-SAN 統合型 PC クラスタの評価を行う。特に、並列データ処理アプリケーション実行中にデータバックアップ処理が同時に動作するような環境を想定して、IP-SAN 統合型 PC クラスタの性能限界を確認する。

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 には、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。

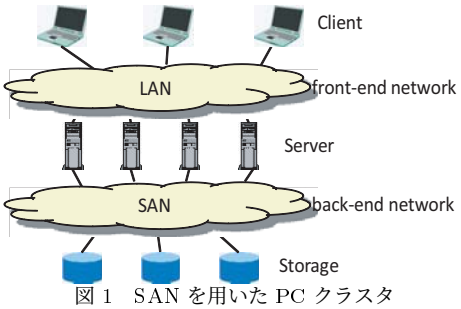

図 1　SAN を用いた PC クラスタ

IP-SAN は、TCP/IP ネットワークで SAN を構築する次世代の SAN である。図 1 に示す PC クラスタにおいて、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。

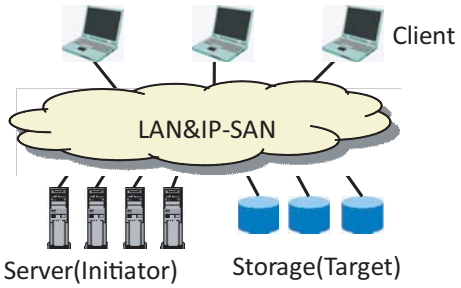

図 2　IP-SAN 統合型 PC クラスタ

2.2 IP-SAN 統合型 PC クラスタと性能への懸念

我々は、クラスタに対する IP-SAN 導入の考え方を一歩進め、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(Internet SCSI) 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 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 [1]。iSCSI の性能についての議論が数多くされている [2] [3] [4]。iSCSI の階層構造は、図 3 のようになっている。

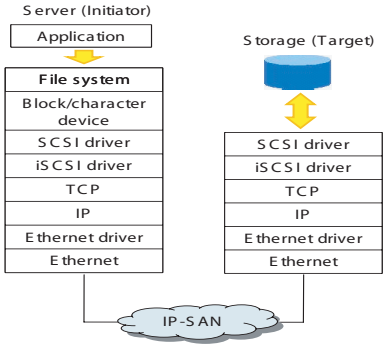

図 3　iSCSI の階層構造

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評価する必要がある。

3. データ処理アプリケーションとその並列化

PC クラスタで行われる計算として、データ処理の重要度が近年高まってきた。PC クラスタ上でデータ処理を実行させるには並列化が必要となってくる。代表的な 2 つのデータ処理アプリケーションを紹介し、それらの並列化について述べる。

3.1 データマイニングと相関関係抽出

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情報となり、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 [7]。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。

3.1.1 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 [5]。

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) を用いる。

3.1.2 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 [6]。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-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。図 4 に FP-tree の構築例を示す。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 [7] 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。

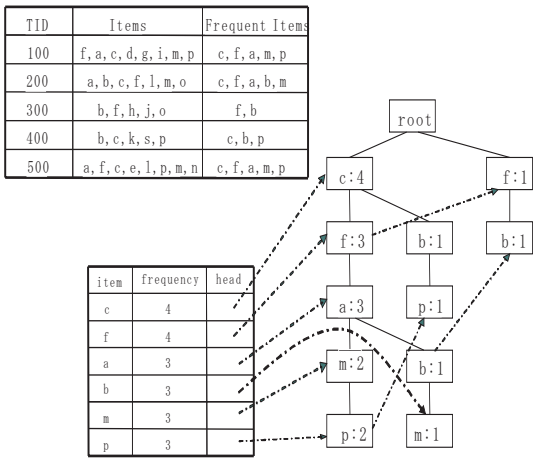

図 4 FP-tree の構造例

3.2 バイオインフォマティクスと BLAST

バイオインフォマティクスは、遺伝子情報の分析や活用により、実際に生命のシステムの、少なくともその一部を、コンピュータの中に再現することを可能にしようと試みている。

バイオインフォマティクスの手法として、問い合わせ配列 (核酸, またはアミノ酸配列) と相同性の高い配列が DB に存在するかどうかを配列 DB 全体にわたって検索する相同性検索がよく用いられる。相同性検索としてよく利用されるものに、BLAST(Basic Local Alignment Search Tool) があり、これは DNA の塩基配列あるいはタンパク質のアミノ酸配列のシーケンスで、ペアワイズのシーケンスアライメントを実行するものである。

バイオインフォマティクスで扱うデータも巨大であるため、並列化してデータを処理することが求められる。BLAST を並列化したものに、MPI ライブラリを用いて BLAST を並列化し実行する mpiBLAST があり、これをを本研究で用いる [8]。

4. 研究概要

本研究では、複数 Initiator と単数 Target を Gigabit Ethernet で接続し、ネットワークと Target に高負荷がかかる環境にした IP-SAN 統合型 PC クラスタの性能評価を行う。また、Target ストレージとして、近年における高性能なストレージである SAS ディスクを RAID0 に構成したものを用いる。表 1 にクラスタノードのスペックを示す。

表 1 Experimental setup：PCs

OS	Initiator : Linux 2.6.18 Target : Linux 2.6.18
CPU	Initiator : Intel Xeon 3.6GHz Target : quad-core Intel Xeon 1.6GHz
Main Memory	Initiator : 4GB Target : 2GB
HDD	Initiator : 250GB SATA Target : 73GB SAS × 2 (RAID 0)

クラスタの構成と管理を行うために Rocks [9] を導入し、クラ

スタのモニタリンスツールとして Ganglia [10] をインストールした。また、iSCSI の Initiator として open-iscsi-2.0-865 [11]、Target として iSCSI Enterprise Target(IET)-0.4.15 [12] を使用する。

4.1 既存研究

我々は、複数の Initiator から単数の Target にアクセスすることで、接続ネットワークと Target に高負荷をかけた IP-SAN 統合型 PC クラスタ上で、ハードディスクベンチマーク bonnie++、並列データマイニング HPA および PFP を実行し、性能を評価した [13]。その結果、bonnie++においては、sequential read アクセスのスループットは低下せず、sequential write アクセスのスループットは、ストレージ I/O が原因で少し低下した。

また、HPA および PFP においては、主に計算処理が行われているため、ネットワーク衝突があまり起こらず、性能は変わらなかった。いずれの場合においても、ネットワーク統合によるネットワークバウンドにはならず、IP-SAN 統合型 PC クラスタは良好な振舞いであった。

4.2 実験内容

そこで本実験では、より現実的な状況を考え、バックグラウンドプロセスとしてデータバックアップ処理などを想定した bonnie++による I/O 処理を行いながら、並列データ処理アプリケーションを同時に実行させ、複数のプロセスを用いることで、IP-SAN 統合型 PC クラスタの性能評価を行う。さらに、そのときの振舞いを解析することで性能の限界を調べ、システムの最適化を行う。

5. 実験結果と考察

5.1 複数プロセス (bonnie++と HPA)

まず、バックグランドプロセスとして bonnie++によるディスク I/O を実行しながら、並列データマイニングのアプリケーション HPA を動作させることで、IP-SAN 統合型 PC クラスタの評価を行った。データマイニングのトランザクションデータは、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。

図 5 に bonnie++を動作させながら HPA プログラムを実行させたときの実行時間を示す。

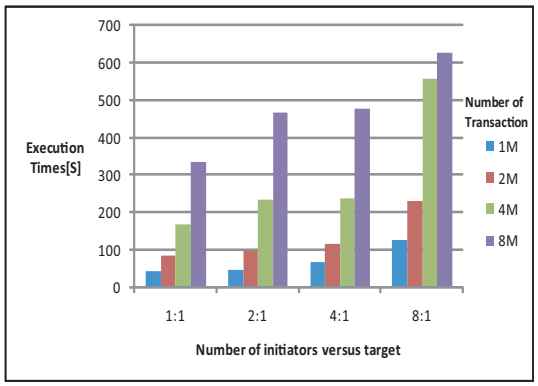

図 5 bonnie++と HPA を実行させたときの実行時間

HPA のみを実行したときと比べ、高負荷をかけたときに実行時間が長くなっており、さらに、1 台の Target に接続される Initiator の数が増えるにつれて実行時間が長くなっている。

図 6 に bonnie++を実行させながら HPA を実行させたときの Target のネットワークトラフィックを示す。

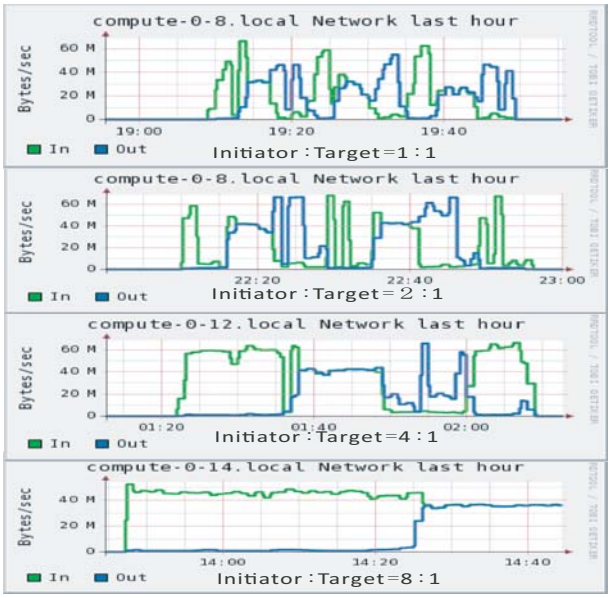

図 6 bonnie++と HPA を実行させたときの Target のネットワークトラフィック

この結果から、Target のネットワークスループットは最大 60Mbytes/sec (約 500Mbps) であり、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。したがって、ストレージアクセスとデータマイニングを同時に実行することでネットワークに高負荷をかけても、ネットワークの帯域幅には大きな影響を与えることはない。

図 7 に Initiator の CPU 使用率、図 8 に Target の CPU 使用率、図 9 に Iniatiator と Target を 8 対 1 に接続させたときのストレージ I/O をそれぞれ示す。

図 7、図 8 から、Initiator 側の"USER CPU"が高くなっており、Target 側の"WAIT CPU"が高くなっていることがわかる。これらから、ストレージアクセスによる I/O 処理ととデータマイニングによる計算処理が忙しく実行されていることがわかる。また図 9 の結果から、1 台の Target にアクセスする Initiator の台数を増やすと、ストレージ負荷が高くなっていることがわかる。

したがって、Target におけるストレージアクセスが軽い場合は、Initiator の CPU がボトルネックとなり、Target にストレージが集中した場合は、Target のストレージ I/O がボトルネックとなっていると考えられる。結果として、iSCSI のトラフィックを統合してもネットワークの帯域幅には余裕があることが分かった。

また、bonnie++、HPA のみを実行させたときと、bonnie++と HPA を同時に実行させたときの、Initiator と Target の接続

図 7　bonnie++と HPA を実行させたときの Initiator の CPU 使用率

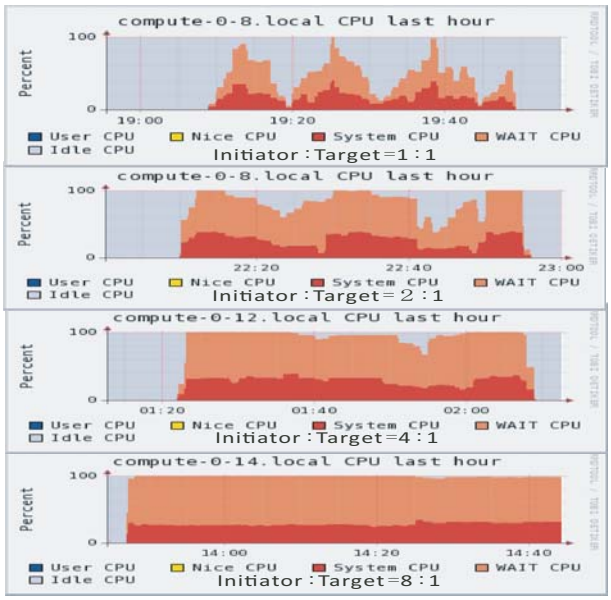

図 8　bonnie++と HPA を実行させたときの Target の CPU 使用率

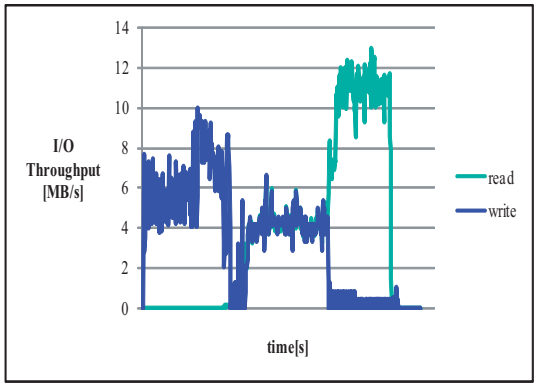

図 9　bonnie++と HPA を実行させたときの Target のストレージ I/O (Initiator:Target ＝ 8 ： 1)

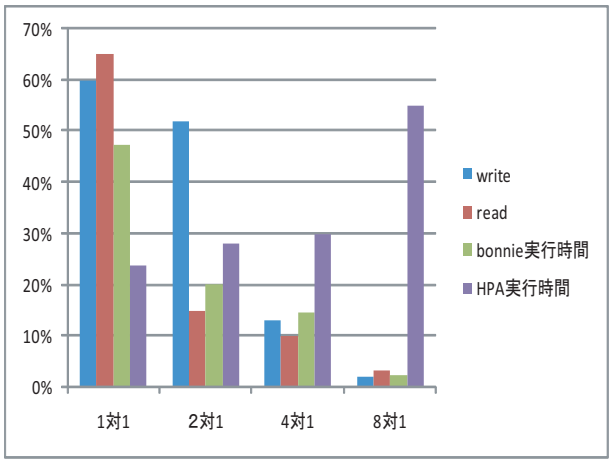

図 10　単数プロセスと複数プロセスのときの性能差 (HPA)

台数による性能の低下率を比較した。図 10 にその結果を示す。

この結果から、Initiator と Target を 1 対 1 に接続させたとき、bonnie++に関する処理の低下が大きく、HPA に関する処理はあまり影響を受けていない。このとき bonnie++側から考えると、bonnie++は一連のデータの流れでディスクの読み書きを行うものであり、その処理が HPA の I/O に邪魔されることで流れが途切れてしまい低下していると考えられる。HPA 側から考えると、読み書きをあまり多く伴わない処理であるので、1 台の Target にかかる I/O 処理が比較的軽いため、影響が少なかったと考えられる。

それとは対照的に Initiator と Target を 8 対 1 に接続させたとき、bonnie++に関する処理にはほとんど影響がなく、HPA に関する処理が大きな影響を受けていることがわかる。このとき bonnie++側から考えると、bonnie++単独のときも I/O 処理は負荷が高く限界であったため、その状況で多少の HPA の I/O 処理が加わってもあまり影響がないためであると考えられる。HPA 側から考えると、bonnie++による I/O 処理がディスクの限界となっており、なかなか I/O 処理を行うことが出来ず性能低下したのではないかと考えられる。

これらの結果を踏まえ、本実験環境における Initiator と Target の最適な接続数を考えると、Initiator 対 Target を 4 対 1 に接続したときに、ストレージアクセスとデータマイニングの処理が最もお互いに影響を及ぼさないと言える。

5.2　複数プロセス　(bonnie++と mpiBLAST)

次に、バックグランドプロセスとして bonnie++によるディスク I/O を実行しながら、並列バイオインフォマティクスのアプリケーション mpiBLAST を 1 プロセスおよび 2 プロセスを同時に実行させることで、IP-SAN 統合型 PC クラスタの評価を行った。そのときバイオインフォマティクスのデータとしては、NCBI BLAST の nt データ [15](約 25GB) を用いた。mpiBLAST を実行する前処理として、nt データを 8 つに分割した。この分割処理は、インデックスを張ることで、大規模なデータを処理しやすくするものである。

1 プロセスの mpiBLAST のみを実行させたとき、2 プロセスの mpiBLAST のみを実行させたときと、bonnie++と 1 プ

ロセスの mpiBLAST、bonnie++と 2 プロセスの mpiBLAST を同時に実行させたときの、Initiator と Target の接続台数による性能の低下率を比較した。図 11 に、その結果を示す。

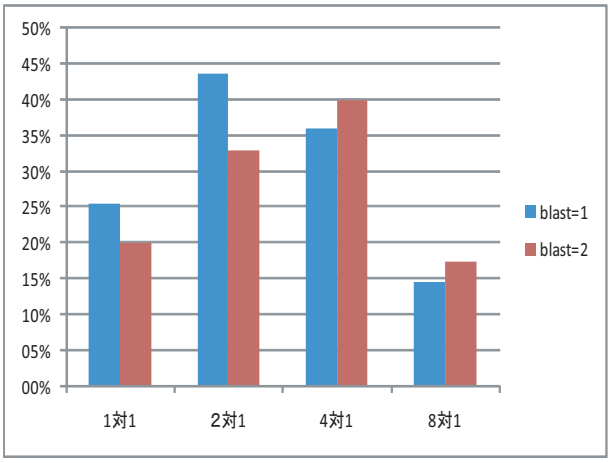

図 11　単数プロセスと複数プロセスのときの性能差 (mpiBLAST)

この結果から、bonnie++と 1 プロセスの mpiBLAST を実行させたときは、Initiator と Target を 2 対 1 に接続した場合に最も性能が低下し、bonnie++と 2 プロセスの mpiBLAST を実行させたときは、Initiator と Target を 4 対 1 に接続した場合に最も性能が低下するということがわかった。この時の Ganglia によるモニタリングにより、ネットワークトラフィックと CPU 使用率にはどの場合においてもまだ十分な余裕があることもわかった。したがって、ディスク I/O により性能が決まっていると考えられる。

1 台の Target からみると、mpiBLAST が 1 プロセスの場合の方が、mpiBLAST が 2 プロセスの時よりもアクセス頻度が高くなり、Initiator と Target を 2 対 1 に接続させた時点でディスク I/O の限界となり、4 対 1、8 対 1 と負荷を高くしていっても既に性能に差が表れないと推測される。また、mpiBLAST が 2 プロセスの場合は、Initiator と Target を 4 対 1 に接続させたときにディスク I/O が限界となっていると考えられる。

以上の結果から、bonnie++によるストレージアクセスと並列データ処理アプリケーションによる処理を同時に実行させた場合においても、ネットワーク統合によるネットワークバウンドにはならず、IP-SAN 統合型 PC クラスタにおいても、システムの性能に影響を与える要因としては、しばしば PC クラスタにおいてボトルネックとされるディスク I/O であるということがわかり、本実験範囲内における、IP-SAN 統合型 PC クラスタの有効性が実証された。また、データ処理アプリケーションにおけるディスク I/O の量でシステム全体の性能低下率が決まると考えられる。

6.　まとめと今後の課題

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と Target に高負荷をかけた環境において、ハードディスクベンチマークと並列データマイニングおよび並列バイオインフォマティクスを同時に実行することで、より現実的なデータバックアップ処理等を想定した状況における IP-SAN 統合型 PC クラスタの評価を行い、システムの性能限界の確認を行った。

この結果、今回の実験範囲内においては、いずれの場合においてもネットワーク帯域にはまだ十分な余裕があり、iSCSI のトラフィックを統合してもシステム性能に影響を及ぼさないことから、我々が提案している IP-SAN 統合型 PC クラスタは有効であり、その柔軟なシステム構成を実現することが可能であると言える。

また、単独プロセスと複数プロセスの比較結果から、Initiator と Target の最適な接続数は、バックグラウンドでデータのバックアップなどストレージアクセスが連続的に行われている状況においては、アプリケーションのディスク I/O の量に左右され、ディスクのコストとのトレードオフとなる。

今後の課題としては、他のデータ処理アプリケーションを用いて幅広いアプリケーションにおけるシステムの評価を行い、さらに OS に対してカーネルモニタなどの導入を行うことで、IP-SAN 統合型 PC クラスタのシステム内部の振舞を詳細に解析していきたいと考えている。

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．

文　　献

[1] iSCSI RFC: http://www.ietf.org/rfc/rfc3722.txt
[2] D. Xinidis, A, Bilas, and M. D. Flouris, “Performance Evaluation of Commodity iSCSI-based Storage Systems,” MSST2005, April 2005.
[3] A. Joglekar, M. E. Kounavis, and F. L. Berry, “A Scalable and High Performance Software iSCSI Implementation,” USENIX FAST 2005, pp.267-280, December 2005.
[4] B. K. Kancherla, G. M. Narayan, and K. Gopinath, “Performance Evaluation of Multiple TCP connections in iSCSI,” MSST2007, September 2007.
[5] R. Agrawal, T. Imielinski, A. Swami: “Mining Association Rules Mining between Sets of Items in Large Databades,” VLDB1994, pp.487-499, September 1994.
[6] J. Han, J. Pei, and Y. Yin: “Mining Frequent Patterns without Candidate Generation,” ACM SIGMOD2000, pp.1-12, May 2000.
[7] Iko Pramudiono and Masaru Kitsuregawa: “Tree structure based Parallel Frequent Pattern Mining on PC cluster,” DEXA2003, pp.537-539, September 2003.
[8] mpiBLAST: http://www.mpiblast.org/
[9] Rocks Cluster: http://www.rocksclusters.org/
[10] Ganglia Monitoring System: http://www.ganglia.info/
[11] Open-iSCSI: http://www.open-iscsi.org/
[12] iSCSI-Enterprise Target: http://sourceforge.net/projects/iscsitarget/
[13] Asuka Hara, Kikuko Kamisaka, Saneyasu Yamaguchi, and Masato Oguchi :“ Analyzing Characteristics of PC Cluster Consolidated with IP-SAN using Data-Intensive Applications, ”In Proc. 20th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS2008), No.631-042, November 2008
[14] Bonnie++: http://www.coker.com.au/bonnie++/
[15] nt: ftp://ftp.ncbi.nlm.nih.gov/blast/db/FASTA/nt.gz