

IP-SAN 統合 PC クラスタを用いた トラフィック特性と I/O 性能に関する考察

神坂 紀久子[†] 山口 実靖[‡] 小口 正人[†] 喜連川 優[§]

近年、コモディティなハードウェアの性能向上と低価格化により、大規模科学技術計算やデータマイニング処理等を PC クラスタにおいて実行することが一般的になった。一方、IP ネットワークを用いたストレージ統合技術である IP-SAN により構築されたクラスタは、効率的なディスク資源の使用、ストレージの導入・管理コストの削減を可能にする。IP-SAN を使用した PC クラスタでは、通常、並列計算を行うフロントエンドの LAN とストレージアクセスを行うバックエンドの IP-SAN が分離したネットワーク構成になっている。そこで本稿では、ネットワークが分離していることによる運用管理負荷を削減することを目的として、双方のネットワークを統合した IP-SAN 統合型 PC クラスタを実現した。また、ネットワーク統合により、並列分散処理を実行した際に性能が劣化する可能性があるため、I/O 処理を伴う並列ベンチマークを使用して性能評価を行い、実行結果を考察した。

A study of traffic characteristics and I/O performance on an IP-SAN consolidated PC cluster

Kikuko Kamisaka[†] Saneyasu Yamaguchi[‡] Masato Oguchi[†]
Masaru Kitsuregawa[§]

Recently, performance improvement and price plummet of hardware allow for executing an order of magnitude scientific computation, databases and data mining applications on PC clusters generally. IP-SAN is consolidation technique of storage using IP networks, and clusters constructed with IP-SAN can use disk resource efficiently and reduce storage management costs. PC clusters using IP-SAN usually have network composition in which front-end and back-end networks are separated.

In this paper, for the reduction of operational management costs, we have constructed an IP-SAN consolidated PC cluster that integrates both networks. We also have measured the I/O performance and analyzed the traffic characteristics.

1 はじめに

近年、大規模科学技術計算やデータベース、データマイニング処理等は、汎用的なハードウェア性能の飛躍的向上と低価格化により、PC を用いて構築した分散メモリ型の並列計算機である PC クラスタ

において実行することができるようになった。

一方、計算機で処理されるデータ量の急激な増大に伴い、サーバ機とストレージデバイスを高速な専用のネットワークで接続する SAN (Storage Area Network) が注目を集めている [1]。SAN はネットワークによってストレージを統合する技術であり、分散されたストレージに存在する大容量データを容易に管理できる。また大規模なデータを扱う PC クラスタにおいて、サーバ機とストレージデバイスを SAN で接続することにより、ディスク資源の効率的な活用が可能となる。

SAN を使用した一般的な PC クラスタでは、サー

[†] お茶の水女子大学
Ochanomizu University

[‡] 工学院大学
Kogakuin University

[§] 東京大学生産技術研究所
Institute of Industrial Science,
The University of Tokyo

バ間を接続するフロントエンドの LAN とディスクアクセスを行うバックエンドの IP-SAN のネットワークが分離されている。しかし、そのようなネットワーク構成では、異なるネットワークの構築と管理が必要になる。

そこで、分離して存在するクライアント、サーバ、ストレージをネットワーク統合することによって、運用管理負荷およびネットワーク構築コストを低減することを考える。しかし、双方のネットワークを統合した場合、ストレージアクセスのトラヒックの影響により、並列分散処理実行時の性能が大幅に劣化する可能性がある。

本稿では、フロントエンドトラヒックとバックエンドトラヒックを統合した IP-SAN 統合型 PC クラスタを実現した。また、IP-SAN 統合型 PC クラスタにおいて、ボトルネックとなる I/O 性能に着目し、それが全体の並列分散処理の性能に与える影響とトラヒックの特性について考察した。

2 研究背景

2.1 SAN を用いた PC クラスタ

従来、大規模なデータを扱う PC クラスタによる並列分散処理システムは、各サーバ毎に固有のローカルデバイスを所有し管理をする非共有ディスク (Shared nothing) 方式が一般的に使用されてきた (図 1)。この場合、各ノードは所有するディスク資源に直接アクセスできるが、ディスク毎にデータが分散されており、そのためディスク資源を効率良く活用し、管理することは容易でない。またこの方式では、多ノード構成が容易なため高いスケーラビリティを得ることはできるが、重要なデータを保持するサーバに障害が発生した場合にクラスタシステム全体に影響するため、可用性の確保も困難である。

それに対し、サーバがディスクをネットワークにより共有する共有ディスク方式 (Shared Disk) は、システム障害への対応やディスク資源の効率的な活用を可能にしている (図 2)。ストレージ分野では、近年のネットワークストレージの発展により、ディスクを共有するストレージ統合技術である SAN (Storage Area Network) が登場し、普及している。それに伴い、ストレージシステムを SAN で構築したクラスタが用いられるようになった。

現在、SAN として、高速な Fibre Channel を用いて構築する FC-SAN が一般的に使用されている。しかし、FC を使用する SAN (FC-SAN) では、FC 用のスイッチやインタフェースなどのハードウェアが高価であり、FC の管理技術者が少ないことから、ス

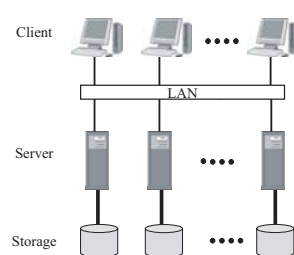


図 1: 非共有ディスク

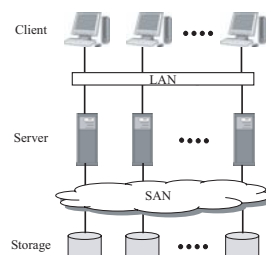


図 2: 共有ディスク

トレージシステムの導入および管理コストがかかるという問題が存在する。そこで、Gigabit/10Gigabit Ethernet の登場などにより、IP ネットワークを使用した IP-SAN が提案され、注目を集めている。IP-SAN は TCP/IP プロトコルと Ethernet で構築される SAN であり、ハードウェアが安価で管理技術者も多いため、それらのコストを大幅に削減できる。

クラスタの構築において、クラスタを新規に導入する場合や増設する場合には、構築・管理コストはノード数に比例するため、可能な限りそれらのコストを抑えることが重要になる。SAN を用いて大規模あるいは小規模な PC クラスタを構成する場合においても、ストレージシステムに従来の FC-SAN の代わりに IP-SAN を用いて構築することにより、汎用ネットワークの PC クラスタを安価なコストで利用することが可能になる。

2.2 iSCSI

IP-SAN で使用される技術には、FCIP, iFCP, iSCSI などのプロトコルが存在するが、中でも 2003 年 2 月に IETF により正式承認された iSCSI (Internet SCSI) プロトコル [2] が最も期待されている。iSCSI では、SCSI コマンドを発行してストレージサービスを要求する側のホストをイニシエータ、ストレージサービスを提供する側のストレージ装置をターゲットと呼ぶ。iSCSI プロトコルは、このサーバ機 (イニシエータ) とストレージ装置 (ターゲット) 間で、ブロックレベルのデータ転送を行う通信プロトコルである。通常の SCSI では、計算機とストレージデバイスを SCSI ケーブルで接続し、SCSI プロトコルを使用して通信するのに対し、iSCSI は標準的に使用されている SCSI コマンドを TCP/IP パケットの中にカプセル化する。これにより、iSCSI は標準の SCSI 命令体系を使用したまま、IP ネットワーク上でストレージへのシームレスなアクセスを可能にする。

2.3 関連研究

iSCSI を用いた IP-SAN の性能評価の関連研究として、まず文献 [3] において、Ng らは早期に独自の SCSI over IP 実装を用いて IP ストレージの性能に関する詳細な測定と解析を行った。同文献では 8KB のブロックサイズにおけるシーケンシャルアクセスの性能を測定し、人工的な遅延が存在する環境下で異なる OS とファイルシステムにおける性能比較を行っている。また同文献では、独自の実装においてホスト側におけるキャッシュの適用やアプリケーションによるプリフェッチが有効であることを指摘している。

文献 [4] においては、Sarkar らによる iSCSI のソフトウェア実装と TOE (TCP Offload Engine) や HBA (Host Bus Adapter) を用いた iSCSI ハードウェア実装の比較に関する研究が行われており、ハードウェア実装は、CPU の負荷を軽減させることはできるが、総合的にはソフトウェア実装の方が性能が高くなることが実証されている。本稿では CPU 負荷を軽減するだけでなく、並列分散処理における I/O 処理の総合的な評価を対象としているため、iSCSI のソフトウェア実装を用いている。

文献 [5] において、Aiken らは iSCSI ソフトウェア実装における遅延を考慮した性能の比較と詳細な分析を行っている。同文献では、iSCSI ソフトウェア実装と FC-SAN の性能比較を行い、また SAN 環境と WAN 環境における性能比較を行っている。同文献の実験の結果、iSCSI ソフトウェア実装は、大きいブロックサイズにおいては FC と同様の性能を得られたことが確認されている。

また、SAN を用いたクラスタに関する研究として、合田らの文献 [6] では、共有読み込み及び動的デクラスタリング機能を持つストレージ仮想化機構を提案した。それを用いた動的負荷分散と動的資源調整の方式を設計し、FC-SAN 接続のストレージを持つ PC クラスタを用いて並列データマイニングアプリケーションを用いた性能評価を行っている。

3 IP-SAN 統合クラスタ

図 3 は、IP-SAN を用いて構築した PC クラスタの例である。一般に、IP-SAN を用いた PC クラスタでは、並列計算などのノード間通信はフロントエンドの LAN で行われ、ディスクへの I/O 処理を伴うストレージアクセスはバックエンドの IP-SAN で行われる。この場合、各ネットワークはスイッチなどで分離されているため、ノードからストレージへアクセスする際のバルクデータと、並列計算のため

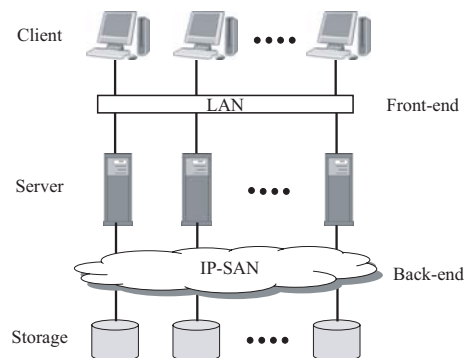


図 3: IP-SAN を用いた PC クラスタ

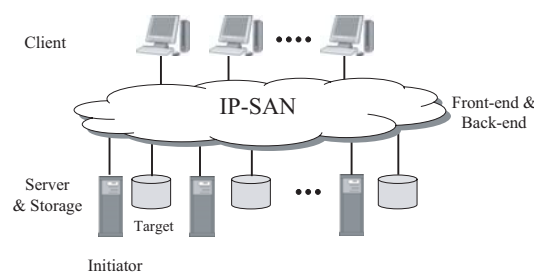


図 4: IP-SAN 統合 PC クラスタ

にノード間で通信されるデータは混在することはない。そのため、データの I/O 処理によるネットワーク負荷を並列計算処理から分離することができる。しかし、そのような IP-SAN のクラスタでは、ネットワーク構成がより複雑になり、異なるネットワークの構築が必要になるため、運用管理の面においても容易ではない。

そこで図 4 に示すように、運用管理の効率化とネットワーク構築コスト削減を目的として、分散して存在するクライアント、サーバ、ストレージを同一の IP ネットワークに統合した IP-SAN 統合 PC クラスタを実現する。IP-SAN 統合 PC クラスタによって、クラスタを増設あるいは新規に導入する場合においても、ネットワークの構築コストや管理コストが軽減できる。また、クライアントからサーバとストレージの一元管理も可能になる。

しかし、双方のネットワークを統合すると、フロントエンドで行われていたノード間通信とバックエンドで行われていたストレージアクセスは、同一の IP ネットワーク経由でデータが転送される。そのため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、並列分散処理を実行する際の性能が劣化する可能性がある。また、I/O を伴うディスクアクセスは、並

表 1: 性能評価実験環境：使用計算機

OS	initiator : Linux 2.6.10 target : Linux 2.6.10
CPU	Intel Pentium 4 CPU 1500MHz
Main Memory	384MB
HDD	36GB SCSI HDD
NIC	Intel(R) PRO/1000 MT

列計算のノード間通信に比べて大幅に速度が低い
ため、大規模なデータを扱うクラスタを用いる
場合には、I/O 性能がボトルネックとなる
可能性があり、I/O 性能について十分に考
慮しなければならない。

4 iSCSI の基本性能

本稿では、IP-SAN 統合型 PC クラスタを iSCSI
を用いて構築して評価する前の基礎実験とし
て、iSCSI によるディスクアクセスのスル
ープットを測定し、ローカル SCSI ディ
スクアクセスの場合と比較した。

4.1 実験環境

実験に用いたシステム環境を表 1 に示す。
iSCSI を使用する場合には、サーバ機 (イ
ニシエータ) 1 台とストレージデバイス (タ
ーゲット) 1 台を使用し、それらは Giga
bit Ethernet スイッチで接続されてい
る。また、iSCSI の実装には、ニューハ
ンプシャー大学 InterOperability Lab[7]
が提供しているオープンソースのソフトウ
ェア実装 (UNH-iSCSI Ver. 1.5.2) を用
いている。

4.2 スループット測定結果

本実験では、1GB のファイルに対して、
ベンチマークアプリケーションで指定した
ブロックサイズ毎に、シーケンシャルリー
ド/ライトアクセスを実行し、測定を行
った。図 5 は、スループット測定結果で
ある。

ブロックサイズは 512Byte から 8KB
までを測定したが、ブロックサイズによ
るスループットの違いはどちらの場合も
見られなかった。また、ローカル SCSI
と iSCSI のどちらの場合も、シーケン
シャルライトアクセスはリードアクセス
よりもスループットが低く、ローカル
SCSI の場合には 74%、iSCSI の場合
には 34% に減少している。さらに、
ローカル SCSI と iSCSI を比較すると、
ローカル SCSI ではリード/ライトア
クセスが 27MB/sec、20MB/sec であ
るのに対し、iSCSI では 14MB/sec、
5MB/sec

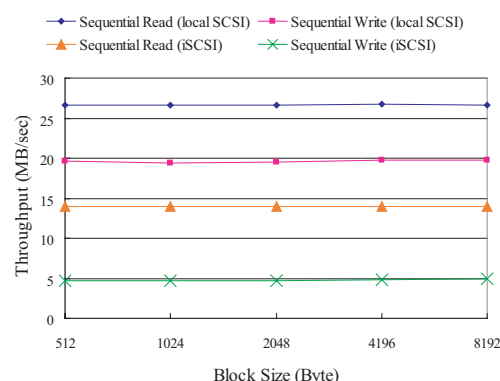


図 5: iSCSI のスループット

となり、大幅にスループットが低い。リー
ドアクセスでは、ローカル SCSI は iSC
SI の約 2 倍、ライトアクセスでは約 4
倍である。

ネットワークを介してディスクにアクセ
スすることによる性能劣化は当然のこと
であるが、iSCSI は SCSI over iSC
SI over TCP/IP という複雑な階層構
造を持っていることも原因の一つである
と考えられる。

5 NAS Parallel Benchmark による性能測定

次に、IP-SAN 統合型 PC クラスタを
iSCSI を用いて構築し、IP-SAN 統
合 PC クラスタが I/O 処理を伴う並
列分散処理を実行した際に、性能をどの
程度劣化させるか調査するため、一般
的な並列ベンチマークを使用した並列
演算性能を測定した。

本実験では、サーバの並列計算を行う
ノード数を 4 として以下の場合の性能
を評価する。

- 全ノードが各々のローカル SCSI
ディスクにアクセスした場合
- 各ノードがそれぞれ異なるディス
クに iSCSI を介してアクセスした
場合

5.1 実験環境

実験に用いたシステムは、前節の実
験と同様である (表 1)。ローカル
SCSI ディスクを使用する場合の実
験環境を図 6 に示し、iSCSI ディ
スクを使用する場合の実験環境を図 7
に示す。図 7 に示すように、イニ
シエータとターゲットは同じ Giga
bit Ethernet スイッチで接続し、
ノード間通信もストレージアクセス
もこのスイッチを介してデータが
転送されることによって IP-SAN
統合 PC クラ

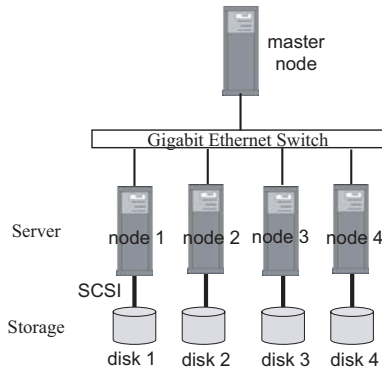


図 6: ローカル SCSI ディスクを用いた場合の実験環境

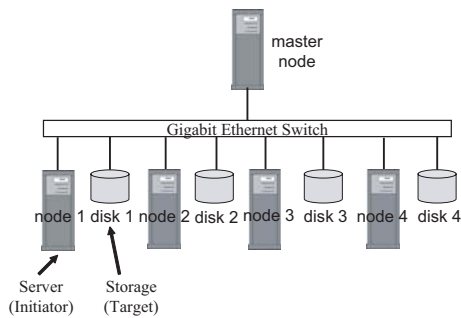


図 7: iSCSI ディスクを用いた場合の実験環境

スタを実現している。ただし、本稿の実験環境の場合、イニシエータとターゲットは1対1接続になっており、各ノード(イニシエータ)は特定のストレージデバイス(ターゲット)に接続する構成となっている。

並列分散処理に使用する MPI ライブラリには、MPICH2-1.0.3[8]を、Fortran コンパイラには、Intel Fortran Compiler 9.0 for Linux を使用した。

5.2 NAS Parallel Benchmark

性能測定を行う際の並列ベンチマークには、NAS Parallel Benchmark (NPB) を用いた。

NPB は、NASA Ames Research Center で開発された、航空関連の流体シミュレーションの実行性能を並列コンピュータ上で評価するベンチマークである。5つの Parallel Kernel Benchmarks と3つの Parallel CFD (Computational Fluid Dynamics) Application Benchmarks から構成されている。

本実験で使用した NPB は、MPI ベースのソースコード実装を用いている Ver.2.4[9][10] である。このバージョンの NPB は、対象問題 BT (Block Tri-diagonal) に対して、大量の I/O 処理を行うアプリ

表 2: NPB の Class と問題サイズ

Class	Size	Mbytes written
W	24 × 24 × 24	22.12
A	64 × 64 × 64	419.43
B	102 × 102 × 102	1697.93

ケーション実行時の性能を測定するベンチマークである NPB I/O (BTIO) を含んでおり、実験にはこれを使用した。BT は、非優位対角な 5 × 5 ブロックサイズの三重対角方程式を解くものである。またこの I/O 処理を実行する NPB I/O には、I/O 処理と性能測定の方法によって異なる 4 つの実行オプションがあり、それらは以下のようにになっている。

- full : MPI I/O with collective buffering
 - 各ノードのメモリに分散したデータをプロセッサの subset 上に集める
 - 集められたデータの再構成を行った後、単一のファイルとして write 処理を実行する
- simple : simple MPI I/O without collective buffering
 - 各ノードのデータは再構成されず、各ノードに分散したデータを集めない
 - 各ノードのデータを単一のディスクに write 処理を実行するため、ディスクへの seek 操作が要求される
- fortran : Fortran direct I/O
 - simple と同じだが、MPI I/O library calls の代わりに、Fortran 77 のファイル操作が使われる
- epio : parallel I/O
 - 各ノードに分散したデータを集めることなく、各々のノードが所有するディスクに並列に I/O 処理を行う

通常、並列計算後の結果は1つのファイルに集められるのが普通であるが、本実験では各サーバにストレージが接続している場合の並列 I/O 性能を測定する目的のため、epio を用いている。また、fortran, simple についても測定を行った。ただし、full については現在の環境ではコンパイルが実行できなかったため、本稿の実験から除外している。実行した

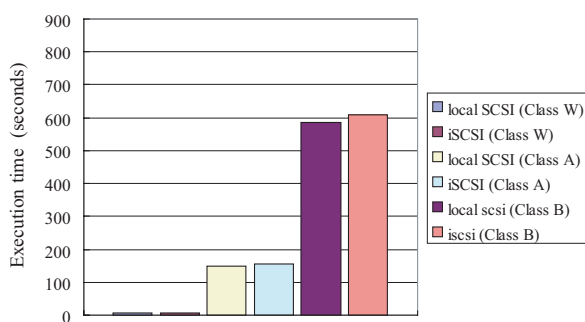


図 8: NPB I/O の実行時間 (epio, node=4)

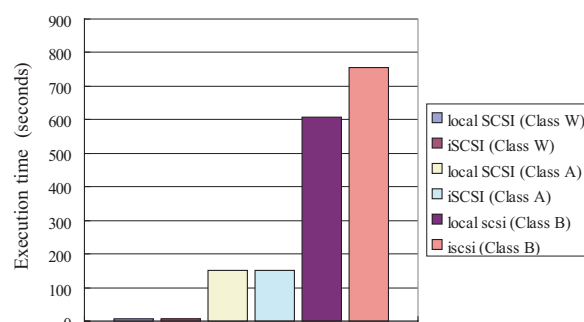


図 10: NPB I/O の実行時間 (simple, node=4)

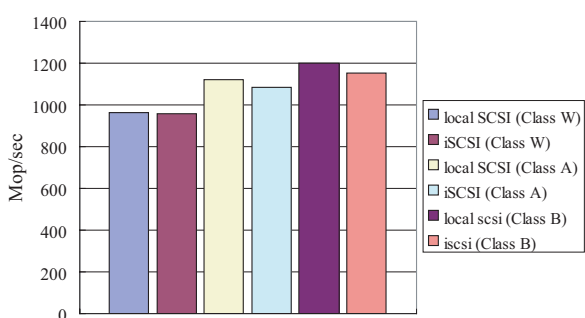


図 9: NPB I/O の Mops 値 (epio, node=4)

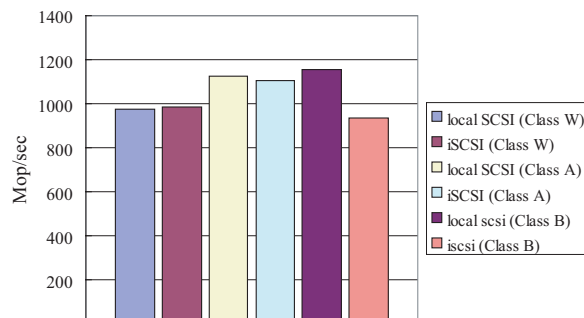


図 11: NPB I/O の Mops 値 (simple, node=4)

NPB の Class, 配列サイズ, write 処理されるデータサイズを表 2 に示す。表 2 に示した問題サイズの反復回数は、いずれも 200 である。

NPB I/O では、並列計算によって得られたデータが目的のディスクに正確に書き出されているか検証する verification という処理が行われている。verification の処理では、まず並列計算によって得られた解がファイルとしてディスク上に書き出される。次に、ディスク上に書かれたファイルを再度メモリ中に読み戻し、読み出されたデータとファイルに書き出したデータが同じであると判断することにより、出力と入力とともに正しく実行されたことを検証する。この際、verification test のために読み戻されるデータがまだディスクに書き出されずにシステムバッファに存在する場合もあるため、ファイルの書き出しを行ってから、verification test の read 操作を行うまでの間隔を極力あけることによってそれを回避している。

5.3 ローカル SCSI ディスクを用いた PC クラスタと IP-SAN 統合 PC クラスタの比較

ローカル SCSI ディスクを用いた PC クラスタと iSCSI ディスクを用いた IP-SAN 統合 PC クラスタにおいて、オプションとして epio で実行した場合の NPB の実行時間と Mops (Million Operations Per Second) 値を図 8, 9 に示す。Mops 値は 1 秒あたりの 100 万演算数である。比較的問題サイズが小さい Class W と Class A の場合には、実行時間と Mops 値ともに、ローカル SCSI ディスクの場合と iSCSI ディスクの場合では、性能に大きな差はなかった。これらの Class においては、表 2 に示すように入出力が行われるデータ量は少ないため、I/O バウンドではなく、基本的に複数ノードを用いた並列処理演算が支配的であると考えられる。

Class B の問題サイズでは、実行時間がローカル SCSI ディスクでは 585 秒であるのに対し、iSCSI ディスクの場合は 610 秒であった。また、Mops 値は、ローカル SCSI ディスクの場合が 1200 であるのに対し、iSCSI ディスクの場合は 1152 であり、やや

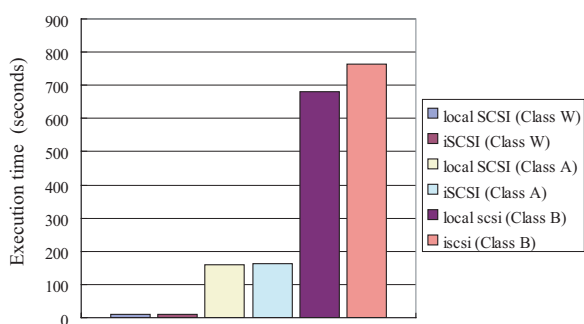


図 12: NPB I/O の実行時間 (fortran, node=4)

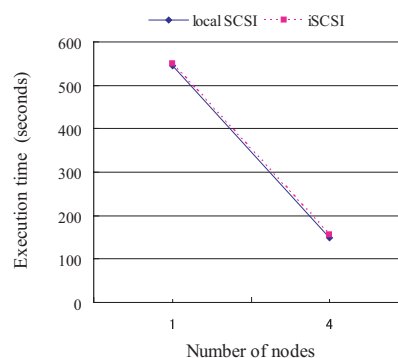


図 14: 1 ノードと 4 ノードにおける NPB I/O の実行時間 (epio, Class A)

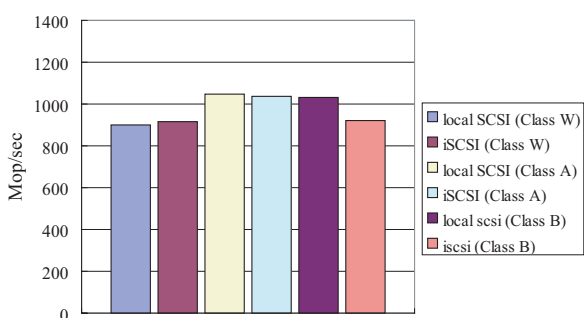


図 13: NPB I/O の Mops 値 (fortran, node=4)

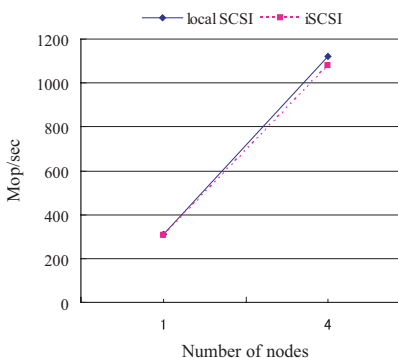


図 15: 1 ノードと 4 ノードにおける NPB I/O の Mops 値 (epio, Class A)

性能が低下している。前節のスループットを測定した基礎実験におけるシーケンシャルライト処理の性能差が4倍であったことから、PC クラスタにおいては、iSCSI ディスクを用いた場合はローカル SCSI ディスクを用いた場合と比較して並列分散処理性能の大きな差はなく、やや劣る程度である。これは、この Class ではある程度のデータ量の入出力が行われるが、全体の実行処理量に比べて、I/O バウンドになる程の負荷ではなかったと考えられる。また、並列処理のノード間通信とストレージアクセスのバルクデータ転送が同じネットワーク上で行われることによるパフォーマンス低下の懸念についても、図7のように各ノードの I/O が直接衝突することのないよう iSCSI ターゲットを配置すれば問題ないことが示された。

5.4 NPB I/O の実行時オプションによる比較

5.3 節の実験と同様に、図 10, 11 は、オプションを simple として実行した際の NPB の実行時間と Mops 値である。図 12, 13 は、オプションを fortran とした場合の NPB の実行時間と Mops 値である。

simple, fortran では、各プロセッサ上で並列計算を実行した結果のデータは、1つのファイルとして単一のディスクに書き出される。その際、各ノードのメモリから1つのノードのメモリへデータを転送する処理が行われる。epio では、各ノードのプロセッサ上で計算されたデータがそのまま各ノードが所有するディスクそれぞれに書き出される。すなわち、ノード数が4の場合、それぞれのディスクに書き出されたファイルのサイズは simple, fortran の4分の1である。この場合、単一のディスクに書き出すための転送処理が必要ないため、epioの方が simple, fortran より並列分散処理性能が高くなる。図 8, 10, 12 に示すように、Class B においては、epio の実行時間が 600 秒前後であるのに対し、simple と fortran の実行時間が 610~770 秒と長くなっている。また、図 9, 11, 13 に示すように、Class B における Mop 値も、epio が 1150~1200 であるのに対し、simple は 930~1150, fortran は 920

～1030 となり, epio より処理性能が低くなっている。

5.5 ノード数による比較

図 14, 15 は, ローカル SCSI ディスクを用いた PC クラスタと iSCSI を介したディスクを用いた IP-SAN 統合 PC クラスタにおいて, ノード数を 1 または 4 にした場合の NPB I/O の epio による実行時間と Mops 値である。すなわち, ノード数 1 の場合はディスクの数も 1 であり, ノード数 4 の場合はディスクの数も 4 である。

これらの図より, ローカル SCSI ディスクを用いた場合と iSCSI を介したディスクを用いた場合は, 実行時間と処理数ともに大きな差はない。IP-SAN 統合 PC クラスタにおいて, イニシエータとターゲットを 1 対 1 に接続した理想的なケースを設計しているためである。また, 4 ノード PC クラスタの性能は 1 ノードに比べ, 実行時間と処理数はローカル SCSI ディスクが 3.6 倍, iSCSI の場合が 3.5 倍となり, 4 倍にはやや低い。

6 まとめと今後の課題

本稿では, SAN を使用した PC クラスタにおいて, 運用管理の効率化とネットワーク構築コスト削減を目的として, 分離して存在するフロントエンドの LAN とバックエンドの SAN を統合する IP-SAN 統合 PC クラスタ環境を iSCSI を用いて実現した。また, フロントエンドトラヒックとバックエンドトラヒックの統合によって, I/O 処理が並列計算処理に影響を与える可能性があるため, IP-SAN 統合 PC クラスタにおいて, I/O 処理を伴う並列アプリケーション実行時の性能を評価した。その結果, 各ノードの I/O が直接衝突することがないように IP-SAN 統合 PC クラスタを設計した場合には, 各ノードのローカルストレージを用いた場合の並列分散処理性能に近ける可能性のあることがわかった。すなわち, クラスタの構成法とアプリケーションの種類によっては, IP-SAN 統合 PC クラスタは望ましい性能を発揮できると考えられる。

今後の課題として, IP-SAN 統合 PC クラスタを設計する際に, 各ノードとストレージを 1 対 1 接続するのではなく, 特定のストレージに複数のサーバがアクセスするような環境を構築する。それにより, ノード間通信とストレージアクセスが完全に混在する環境において, どの程度並列分散処理性能が劣化するかを本稿の実験環境と比較して調査する。また, ストレージアクセスのバルクデータ処理がノード間通信のデータ処理に影響しないように, 効率的にスケジューリングする機構を考える。

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。

参考文献

- [1] Storage Networking Industry Association, <http://www.snia.org/>.
- [2] iSCSI Draft, <http://www.ietf.org/rfc/rfc3720.txt>.
- [3] Ng, W. T., Hillyer, B., Shriver, E., Gabber, E. and Ozden, B.: Obtaining High Performance for Storage Outsourcing, *Proc. FAST 2002, USENIX Conference on File and Storage Technologies*, pp. 145–158 (2002).
- [4] Sarkar, P., Uttamchandani, S. and Voruganti, K.: Storage over IP: When Does Hardware Support help?, *Proc. FAST 2003, USENIX Conference on File and Storage Technologies*, pp. 231–244 (2003).
- [5] Aiken, S., Grunwald, D., Pleszkun, A. R. and Willeke, J.: A Performance Analysis of the iSCSI Protocol, *Proc. 20th IEEE Symposium on Mass Storage Systems and Technologies (MSS '03)*, pp. 123–135 (2003).
- [6] 合田和生, 田村孝之, 小口正人, 喜連川優: SAN 結合 PC クラスタにおけるストレージ仮想化機構を用いた動的負荷分散並びに動的資源調整の提案とその評価, 電子情報通信学会和文論文誌 D, Vol. J87-D-I, No. 6, pp. 661–674 (2004).
- [7] InterOperability Lab in the University of New Hampshire, <http://www.iol.unh.edu/>.
- [8] MPICH2, <http://www-unix.mcs.anl.gov/mpi/mpich>.
- [9] NAS Parallel Benchmark (NPB), <http://www.nas.nasa.gov/Software/NPB>.
- [10] NPB-MPI 2.4 I/O, <http://www.nas.nasa.gov/News/Techreports/2003/PDF/nas-03-002.pdf>.