

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

神坂 紀久子[†] 山口 実 靖^{††}
小 口 正 人[†] 喜 連 川 優^{†††}

TCP/IP ベースのストレージ統合技術である IP-SAN が登場したことにより、PC クラスタにおける計算ノード-ストレージ間のネットワークに IP-SAN を使用することが可能となっている。現在のところ、SAN を使用した PC クラスタでは一般にフロントエンドの LAN とバックエンドの SAN のネットワークを個々に構築しているが、IP-SAN の使用により、これら双方のネットワークを統合し、運用管理負荷を削減できる。本稿では、バックエンドのネットワークをフロントエンドに統合した IP-SAN 統合型 PC クラスタ環境において、マクロベンチマークを使用して並列分散処理性能を評価し、非統合型の性能と比較した。また、マイクロベンチマークとして、複数のプロセスにより I/O やノード間通信が繰り返し実行されたネットワーク高負荷時における評価も行なった。

Performance Evaluation of PC cluster system integrated with IP-SAN in simultaneous access by multi-processes

KIKUKO KAMISAKA,[†] SANEYASU YAMAGUCHI,^{††} MASATO OGUCHI[†]
and MASARU KITSUREGAWA^{†††}

With the advent of TCP/IP-based consolidation technique of storage, IP-SAN becomes available as networks between cluster nodes and storage for PC clusters. Usually, front-end LAN and back-end SAN are established separately on PC clusters. However, both networks can be integrated by using IP-SAN on PC clusters, which leads to the reduction of construction and operational management costs. We have proposed a PC cluster system integrates back-end IP-SAN into front-end LAN. In this paper, we have evaluated its parallel/distributed processing performance by using a macro-benchmark. We also evaluated it in simultaneous access by multi-processes in consideration of heavily-loaded networks.

1. はじめに

近年、コモディティなハードウェア性能の飛躍的向上と低価格化により、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を PC クラスタにおいて実行することが一般的になった。並列計算機で取り扱うデータ量も年々大規模化し、PC クラスタにおいてデータ処理を扱うアプリケーションの重要性が増してきている。

現在、HPC 分野などに使用される大規模な PC クラスタでは、計算ノード-ストレージ間のネットワークに

Fibre Channel(FC) や InfiniBand などの高速な専用回線が多く使用されている。しかし、TCP/IP ベースの SAN(Storage Area Network)¹⁾ として、IP-SAN や主要プロトコルである iSCSI が登場したことにより、コモディティなネットワークだけを使用した PC クラスタの構築が可能になった。

そこで我々は、通常は個々に構築される計算ノード-ストレージ間のバックエンド IP-SAN をフロントエンド LAN に統合した IP-SAN 統合型 PC クラスタを提案し、その可用性について評価を行っている。IP-SAN 統合型 PC クラスタはストレージトラヒックと MPI トラヒックを統合したことによる並列分散処理性能への影響の懸念があることから、本稿では、IP-SAN 統合型 PC クラスタの性能を評価し、非統合型の PC クラスタと比較した。本稿の実験では、NAS Parallel Benchmark(NPB)²⁾ の並列計算ならびに I/O を行う

[†] 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

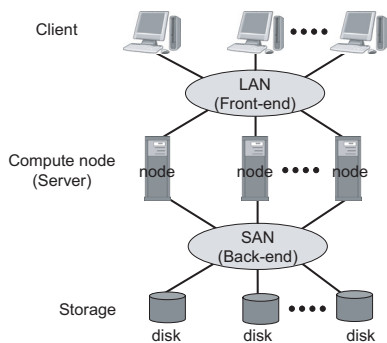


図 1 SAN を用いた PC クラスタ

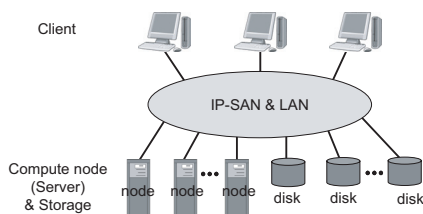


図 2 IP-SAN 統合型 PC クラスタ

アプリケーションを使用したマクロベンチマークと、複数プロセスによりノード間通信および I/O の並行処理を実行するマイクロベンチマークを用いて性能評価を行う。

本稿の構成は以下の通りである。まず、2 章において提案システムである IP-SAN 統合型 PC クラスタについて述べる。3 章でマクロベンチマークによる性能評価、4 章でマイクロベンチマークによる性能評価、5 章で関連研究について述べ、最後に 6 章でまとめる。

2. IP-SAN 統合型 PC クラスタ

2.1 SAN を用いた PC クラスタ

近年、計算機で取り扱うデータ容量が飛躍的に増大したことから、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-ストレージ間のバックエンドのネットワークに SAN (Storage Area Network) 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

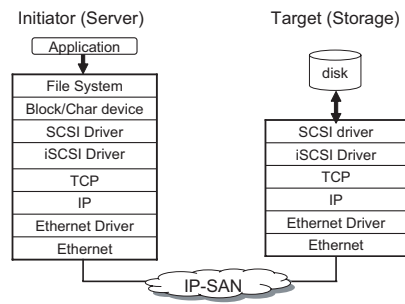


図 3 iSCSI の階層構造

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パスト性が高く、転送データ量が多いため、計算ノード (サーバ) -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 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。

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、運用ができる。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思われる。

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) -ストレージ間のバックエンドネットワークを、ノード間の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)³⁾ は、2003 年 2 月に IETF により正式承認された 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。

SAN を使用した PC クラスタでは、一般的にフロントエンド 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。しかし、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

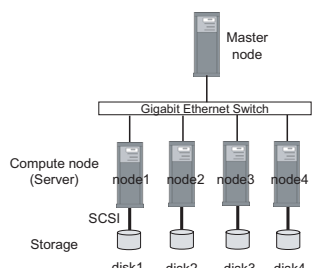


図 4 ローカルストレージを使用した PC クラスタの実験環境

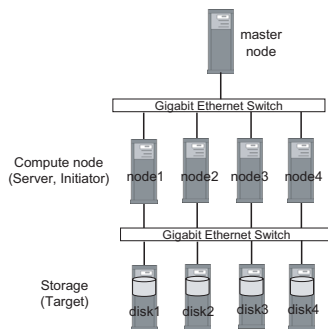


図 5 非統合型 PC クラスタの実験環境

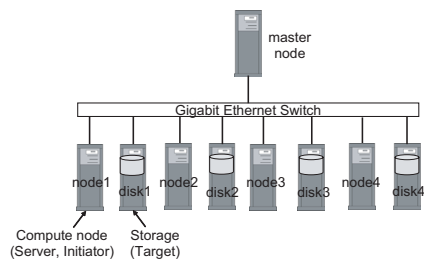


図 6 IP-SAN 統合型 PC クラスタの実験環境

表 1 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

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

なった。

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路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。よって、バックエンドネットワークをフロントエンドに統合した IP-SAN 統合型 PC クラスタは、非統合型の 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詳細に評価する必要がある。

3. マクロベンチマークによる性能評価

本稿では、まずマクロベンチマークとして、NAS Parallel Benchmark(NPB)²⁾の並列計算ならびにI/Oを実行するベンチマークアプリケーションを使用して、IP-SAN 統合型 PC クラスタの並列演算性能を評価した。

3.1 実験環境

本実験では、計算ノード数を 4 として、以下の PC

クラスタ環境における性能を評価する。

- ローカルストレージを使用した PC クラスタ
- 非統合型 PC クラスタ
- IP-SAN 統合型 PC クラスタ

ローカルストレージを使用した PC クラスタでは、図 4 に示すように、ベンチマークを実行する全ノードが各々のローカル SCSI ディスクに直接接続される。非統合型 PC クラスタでは、図 5 に示すように、フロントエンドの LAN とバックエンドの IP-SAN が、異なる Gigabit Ethernet スイッチを使用して別々に構築されるため、ノード間通信とストレージアクセスは異なるスイッチを介して実行される。我々が実現した IP-SAN 統合型 PC クラスタでは、図 6 に示すように、各ノードは、フロントエンドの LAN とバックエンドの IP-SAN を統合したネットワークを介して各々のストレージに接続される。この場合、ノード間通信もストレージアクセスも、同じ Gigabit Ethernet スイッチを介してデータが転送される。

実験に用いた PC のシステム環境を表 1 に示す。iSCSI の実装には、ニューハンプシャー大学 Inter-Operability Lab⁴⁾ 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. 1.6.0) を用いており、iSCSI ターゲットは FILE モードで動作させた。ただし、本稿の実験環境において、iSCSI を用いたストレージアクセスでは、ノード (イニシエータ) とストレージ (ターゲット) を 1 対 1 接続としている。そのため、各ノードは特定のストレージに接続する構成となっている。また並列分散処理に使用する MPI ライブラリには、MPICH2 Ver. 1.0.3⁵⁾ を使用している。

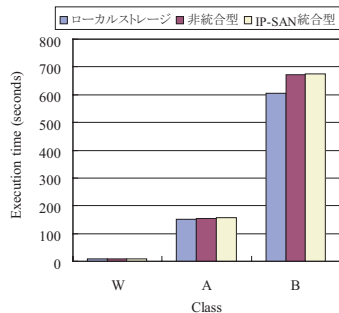


図 7 NPB I/O の実行時間 (“simple” option, 4-nodes)

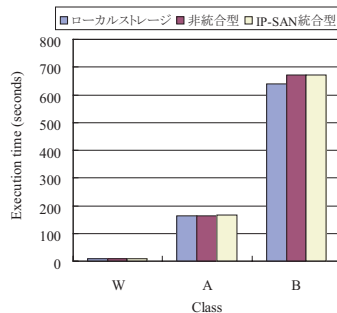


図 9 NPB I/O の実行時間 (“fortran” option, 4-nodes)

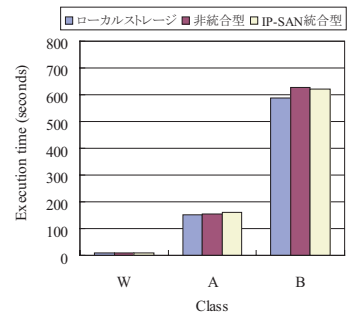


図 11 NPB I/O の実行時間 (“epio” option, 4-nodes)

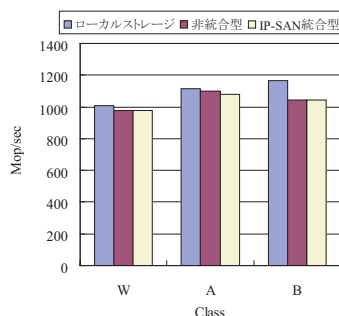


図 8 NPB I/O の Mops 値 (“simple” option, 4-nodes)

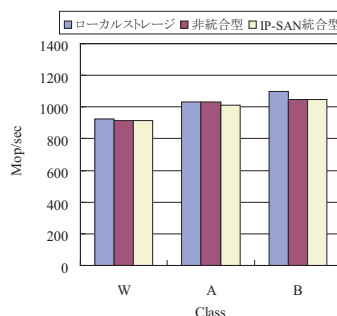


図 10 NPB I/O の Mops 値 (“fortran” option, 4-nodes)

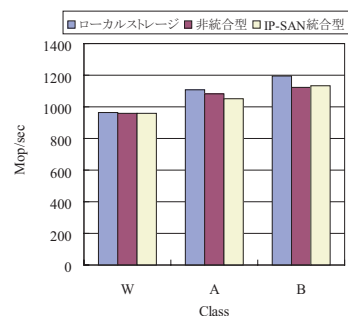


図 12 NPB I/O の Mops 値 (“epio” option, 4-nodes)

3.2 NAS Parallel Benchmark I/O Version 2.4

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 NPB I/O Ver.2.4 である。NPB は並列計算のベンチマークであるが、このバージョンでは大量の I/O を行う並列計算ベンチマークアプリケーションを実行する NPB I/O (BTIO)⁶⁾ が使用できる。ただし、NPB I/O は対象問題 BT (Block Tri-diagonal) に対してのみ実行可能であるため、本実験では BT を対象として NPB I/O を使用した。実行した NPB I/O の Class, 配列サイズ, ストレージに書き出される合計データサイズを表 2 に示す。NPB I/O には、I/O と性能測定の方法によって異なる実行オプションがあり、本実験で使用した実行オプションは次の 3 つである。

- “simple” : simple MPI I/O without collective buffering
- “fortran” : Fortran direct I/O
- “epio” : parallel I/O

オプション “simple” では、各ノードのメモリに分散したデータが単一のプロセッサ上に集められ、単一

のファイルとして書き出される。その際、各ノードのデータは再構成されず、ディスクへの seek 操作が要求される。“fortran” は、simple と同じ動作を行うが、MPI I/O ライブラリの代わりに、Fortran 77 のファイル操作が使用される。“epio” は、各ノードに分散したデータは集められることなく、各々のノードが所有するディスクに並列に I/O を行う。

3.3 NPB による実験結果と考察

本実験では、NPB の実行時間と Mops (Million Operations Per Second) 値を測定した。Mops 値は 1 秒間あたりの 100 万演算数である。オプションを “simple” で実行した場合の結果を図 7, 8 に示す。また、“fortran” の結果を図 9, 10 に、“epio” の結果を図 11, 12 に示す。

問題サイズが小さい Class W の場合には、実行時間と Mops 値ともに、ローカルストレージを使用した PC クラスタ、非統合型 PC クラスタ、IP-SAN 統合型 PC クラスタでは、ほぼ同じ値が得られた。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が少なく、並列演算が支配的なためであると考えられる。

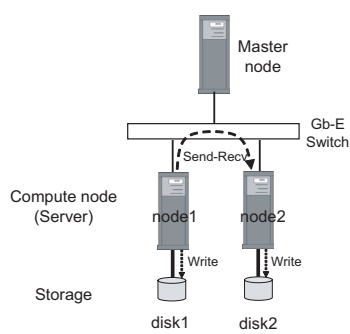


図 13 ローカルストレージを使用した PC クラスタの実験環境

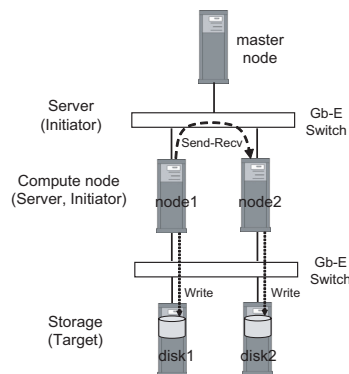


図 14 非統合型 PC クラスタの実験環境

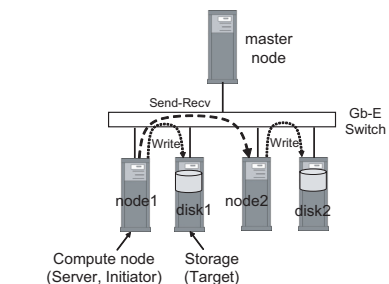


図 15 IP-SAN 統合型 PC クラスタの実験環境

一方、Class B の問題サイズでは、IP-SAN を用いた場合はローカルストレージの場合と比較して他の問題サイズよりも性能差が大きくなる。その傾向は、図 8 に示すように、“simple” の場合が最も顕著にみられる。これは、単一のプロセッサにデータが集められる際に、IP-SAN を介してデータ転送していることが原因である。また、図 12 に示すように、“epio” は他のオプションと比較していずれも性能が良いのは、並列 I/O が実行され、各ノードのデータを一つのノードへ転送する処理が省略されるためである。

全体として、IP-SAN 統合型 PC クラスタは、非統合型と比較して並列演算処理性能に大きな差がないという結果が得られた。このことから、本ベンチマークのような大量の I/O 処理を伴う並列分散処理を実行した場合、ネットワークにあまり大きな負荷がかかっていないため、性能への影響は少なく、IP-SAN 統合型 PC クラスタは非統合型と同程度の性能が発揮できると考えられる。

4. マイクロベンチマークによる性能評価

4.1 実験概要と実験環境

通常、I/O を伴う並列分散処理では、ノード間におけるメッセージの送受信 (Send/Recv) とノード-ストレージ間におけるストレージアクセス (Read/Write) が行われる。IP-SAN 統合型 PC クラスタは、ノード間通信のデータとストレージアクセスのデータが同じ NIC を介して送受信され、同一ネットワーク上に混在する。その場合、IP-SAN 統合型 PC クラスタにおいては、それらの処理が並行して実行されたときに性能に影響が及ぶ可能性がある。

そこで本実験では、IP-SAN 統合型 PC クラスタにおいてマイクロベンチマークを使用し、MPI による Send-Recv のノード間通信と Write によるストレ-

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た場合の性能への影響を評価する。

ローカルストレージを使用した PC クラスタ、非統合型 PC クラスタ、IP-SAN 統合型 PC クラスタの実験環境は、図 13 ~ 図 15 であり、Send-Recv と Write を実行するノード数はいずれも 2 である。また使用計算機は表 1 と同じである。Send-Recv を実行するプログラムは、送信ノード (ノード 1) において MPLSend() を、受信ノード (ノード 2) において MPLRecv() 関数を用いている。Write を実行するプログラムには、write システムコールを用い、各ノードに接続されているストレージに対して並列にデータを書き出す。ただし、本実験では、イニシエータ側のキャッシュによる性能への影響を排除することを目的として、ファイルシステムを介さない raw デバイスを使用するシーケンシャルライトアクセスを実行した。そのため、ベンチマークソフトウェアにより発行されたシステムコールは、イニシエータ側のどのキャッシュにも書き込まれず、ターゲットの SCSI 層へ転送され、ディスクに書き込まれた時点で終了となる。

4.2 単独の Write と Send-Recv のスループット

まず、Send-Recv、Write をそれぞれ単独で実行した際のスループットを測定した。図 16 はその測定結果である。Send-Recv 単独で実行する場合は、メッセージサイズを 1MB とし、2 ノード間で片道 Send-Recv を 1000 回発行する。Write 単独の場合には、1 ノードにおいて、I/O ブロックサイズを 1MB とし、ノードに接続されたローカルストレージあるいは IP-SAN 接続のストレージに対して各々のノードが raw デバイスに対する Write を 1000 回発行する。

図 16 に示すように、2 ノード間における Send-Recv のスループットは、493.6Mbps であった。また、ロー

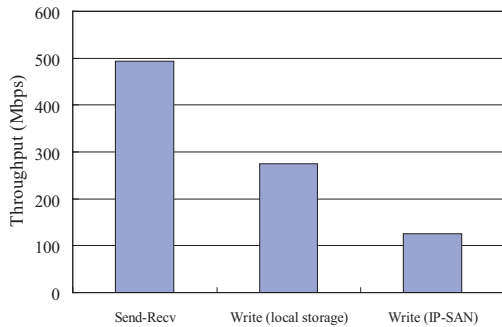


図 16 Write, Send-recv の各処理の実行時間

カルストレージに対するシーケンシャルライトアクセスのスループットは 275.3Mbps であり, Send-Recv のスループットの 0.56 倍となった。一方, IP-SAN を介したストレージへのシーケンシャルライトアクセスは 126.3Mbps であり, Send-Recv のスループットの 0.26 倍という結果が得られた。

これらの結果から, IP-SAN を介したストレージへのライトアクセスとローカルストレージへのライトアクセスには倍程度のアクセス速度差があり, さらに, IP-SAN のライトアクセスは Send-Recv と比較すると約 1/4 とかなり遅いことがわかった。

4.3 複数プロセスによる同時アクセス時のスループット

並列分散処理における最小単位の動作として, Send-Recv によるノード間通信と各ノードに接続されたストレージへのアクセスを同一ノード上でそれぞれ独立したプロセスとして並行に実行させた際のスループットを測定する。

IP-SAN 統合型 PC クラスタにおいては, ノード間通信の Send-Recv とストレージアクセスの Write がほぼ同じ処理時間で並行して実行された場合に, 非統合型と比較して性能への影響が大きくなるであろうと推測される。しかし, ノード間通信とストレージアクセスには性能差があり, IP-SAN を介した Write のスループットは Send-Recv と比較して約 1/4 の性能である。そこで本実験では, Send-Recv が Write とほぼ同じ実行時間になるように, 1 プロセスあたりの繰り返し回数を Write の場合の 4 倍の 4000 回に設定した。Write と Send-Recv の各プロセスの設定パラメータは以下の通りである。

- Send-Recv の 1 プロセス: 繰り返し回数=4000 回, メッセージサイズ=1MB
- Write の 1 プロセス: 繰り返し回数=1000 回, I/O

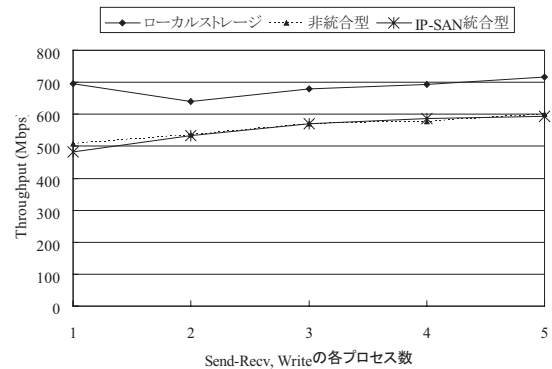


図 17 複数プロセスを発行させた場合における全プロセスの合計スループット (I/O size:1MB, message size:1MB, 2-nodes)

ブロックサイズ=1MB

本実験では, IP-SAN 統合型 PC クラスタの同一ノードにおいて, 上記の Send-Recv と Write の各プロセスを複数並行して動作させることにより, 意図的にネットワークに高負荷をかけた場合の性能への影響を評価する。Write と Send-Recv の各プロセス数を 1 から 5 まで変動させた場合の全プロセスの合計スループットを図 17 に示す。この場合, Write と Send-Recv を合わせた全プロセス数は 2 から 10 となっている。また, 図 18 は, 同ベンチマーク上において, Write を実行するプロセスの合計スループットだけを表したものであり, 図 19 は同様に Send-Recv を実行するプロセスの合計スループットだけを表したものである。

図 17 に示すように, Send-Recv と Write を 1 プロセスずつ実行させた場合における全プロセスの合計スループットは, ローカルストレージの場合は 695.13Mbps, 非統合型の場合は 508.83Mbps, IP-SAN 統合型 PC クラスタでは 482.05Mbps であった。よって, IP-SAN 統合型はローカルストレージの 0.69 倍であるが, 非統合型と比較してスループットは 0.95 倍とやや性能が低下するという結果が得られた。また, Send-Recv, Write の各プロセス数が 2, 3 と大きくなるにつれ, その性能差が小さくなる。理論的にはプロセス数を増加させるにつれ, 合計スループットが増加するはずであるが, プロセス数の増加によるスループットの向上はどの環境においてもあまり見られなかった。

Write のスループットでは, 図 18 に示すように, Send-Recv と Write を 1 プロセスずつ実行させた場合における IP-SAN ストレージはローカルストレージと比べ, スループットは半分以下の値に劣化してい

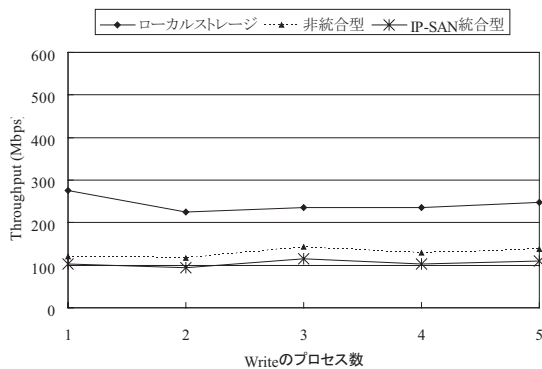


図 18 複数プロセスを発行させた場合における Write を実行するプロセスの合計スループット (I/O size:1MB)

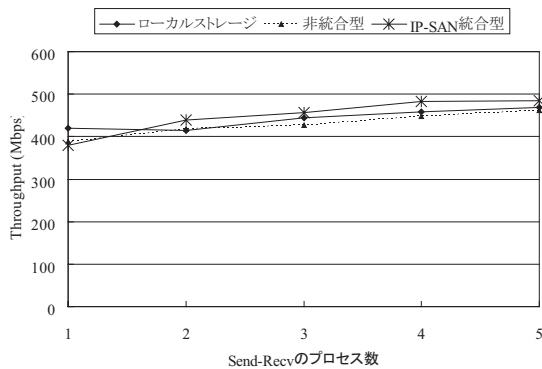


図 19 複数プロセスを発行させた場合における Send を実行するプロセスの合計スループット (message size:1MB, 2-nodes)

る。また、IP-SAN 統合型と非統合型では、統合型の方がややスループットが低下した。図 19 に示すように、ノード間通信による Send-Recv は、統合型と非統合型で同じネットワークを介して通信されるが、どのプロセスにおいてもストレージアクセスを並行して動作させることによる影響はあまり大きくないことがわかった。

これらの結果から以下の事が考えられる。まず、Send-Recv や Write を行うプロセス数を増やしていった場合、本実験環境においてはネットワークより先に CPU の負荷が一杯となる。その場合、ネットワークにはまだ余裕があるため、IP-SAN 統合型も非統合型と同程度の実行性能を達成することができる。ただし、この関係は、マシンやネットワークの性能が向上すれば変化するものと考えられる。例えば、クラスタの計算ノードを新しいマシンに置き換え、CPU の性能を上げた場合、CPU はさらに高い負荷に耐えるように

なり、ネットワークの帯域をぎりぎりまで使い切るようになるかもしれない。そのなると、統合型と非統合型で多少の実行性能差が出るようになるであろう。しかしその場合でも、今後はネットワークを例えば Gigabit Ethernet から 10Gigabit Ethernet へスケールアップしたならば、まだネットワークには余裕がある環境となり、IP-SAN 統合型と非統合型の実行性能差は無くなる可能性が高いと考えられる。

5. 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⁷⁾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。

文献⁸⁾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。

文献⁹⁾において、Aiken らは iSCSI ソフトウェア実装と FC-SAN の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¹⁰⁾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。

6. まとめと今後の課題

iSCSI を用いた PC クラスタにおいてバックエンドネットワークをフロントエンドに統合した IP-SAN 統合型 PC クラスタは、フロントエンドとバックエンドを個々に構築する必要がないため、クラスタの構築および運用管理コストが削減できる。本稿では、それらのネットワークを個々に構築した非統合型 PC クラスタと比較して、性能にどの程度影響を与えるかということの評価した。その際、マクロベンチマークとして NAS Parallel Benchmark による並列演算処理性能と、ノード間通信および I/O を実行するマイクロ

ベンチマークによる性能を測定した。マイクロベンチマークにおいては、ノード間通信や I/O を行うプロセス数を増やし、ネットワークに負荷をかけた場合の性能を評価した。さらに、並列処理演算の NPB とストレージアクセスをクラスタ上の別々のノードで同時に実行した場合の性能評価を行った。これらの結果、iSCSI を使用した場合には、バックエンドに IP-SAN を持つ PC クラスタと比較して、双方のネットワークを統合しても、ほぼ同等の並列分散処理性能を達成できることがわかった。

今後の課題として、CPU がボトルネックとなることがないように、高性能な計算ノードを使用し、iSCSI の I/O 性能についても検証しながら IP-SAN 統合型 PC クラスタの性能評価を行う。また、本稿の実験環境は単純なシステム構成であるため、IP-SAN 統合型モデルをさらに大規模なクラスタ環境において適用することを考える。本稿の実験では NPB I/O による性能と複数プロセスによる同時アクセス時の性能の評価を行ったが、今後はさらに実用的なベンチマークアプリケーションを使用し、様々なパターンにおける性能を解析する。

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) NAS Parallel Benchmark (NPB): <http://www.nas.nasa.gov/Software/NPB>.
- 3) iSCSI RFC: <http://www.ietf.org/rfc/rfc3722.txt>.
- 4) InterOperability Lab in the University of New Hampshire: <http://www.iol.unh.edu/>.
- 5) MPICH2: <http://www-unix.mcs.anl.gov/mpi/mpich>.
- 6) NPB-MPI 2.4 I/O: <http://www.nas.nasa.gov/News/Techreports/2003/PDF/nas-03-002.pdf>.
- 7) 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*.
- 8) 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*.
- 9) Aiken, S., Grunwald, D., Pleszkun, A. and Willeke, J.: A Performance Analysis of the iSCSI Protocol, *Proc. 20th IEEE Symposium on Mass Storage Systems and Technologies (MSS '03)*.
- 10) 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.