社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

iSCSIを用いたPCクラスタにおける バックエンドネットワーク統合による性能への影響評価

神坂紀久子† 山口 実靖†† 小口 正人† 喜連川 優†††

† お茶の水女子大学 〒112–8610 東京都文京区大塚 2–1–1
†† 工学院大学 〒163–8677 新宿区西新宿 1–24–2
††† 東京大学生産技術研究所 〒153–8505 東京都目黒区駒場 4–6–1
E-mail: †kikuko@ogl.is.ocha.ac.jp, oguchi@computer.org, ††sane@cc.kogakuin.ac.jp

あらまし 近年，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 クラスタにおけるクラスタノード-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 の使用は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 iSCSI 接続 PC クラスタ環境において，NAS Parallel Benchmark と基礎的な並列処理および I/O を実行するプログラムによる並列分散処理性能を測定した．

キーワード Storage Area Network，iSCSI, PC クラスタ, 並列分散処理，NAS Parallel Benchmark

Performance Evaluation of Back-end Network Consolidation on iSCSI-connected PC Cluster

Kikuko KAMISAKA†, Saneyasu YAMAGUCHI††, Masato OGUCHI†, and Masaru KITSUREGAWA†††

† Ochanomizu University 2–1–1 Otsuka, Bunkyo-Ku, Tokyo 112–8610 Japan
†† Kogakuin University 1–24–2 Nishi-shinjuku, Shinjuku-ku, Tokyo, 163–8677 Japan
††† Institute of Industrial Science, The University of Tokyo 4–6–1 Komaba, Meguro-ku, Tokyo, 153–8505 Japan
E-mail: †kikuko@ogl.is.ocha.ac.jp, oguchi@computer.org, ††sane@cc.kogakuin.ac.jp

Abstract Recently, with the advent of TCP/IP-based consolidation technique of storage, IP-SAN has been employed on networks between cluster nodes and storage on PC clusters. Usually, front-end LANs and back-end SANs are established separately on PC clusters. Because both networks can be integrated using IP-SAN on PC clusters, it leads to the reduction of operational management costs. In this paper, we have measured the parallel/distributed processing performance on iSCSI-connected PC cluster that integrates both networks using NAS Parallel Benchmark and basic parallel processing and I/O program.

Key words Storage Area Network，iSCSI, PC Cluster, Parallel Distributed Processing, NAS Parallel Benchmark

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．

従来より，HPC 分野などに使用される大規模な PC クラスタでは，クラスタノード–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使用されてきた．しかし，TCP/IP と Ethernet を使用する IP-SAN(IP-Storage Area Network) [1] である iSCSI プロトコル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

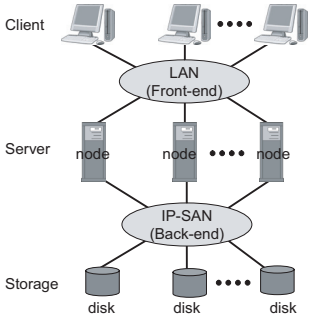

図 1　SAN を用いた PC クラスタ

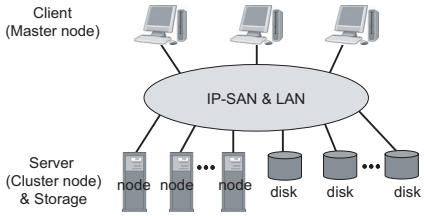

図 2　IP-SAN 統合 PC クラスタ

クラスタの構築が可能になった．また，個々に構築していたノード間のフロントエンドネットワークとノード–ストレージ間のバックエンドネットワークを一つのネットワークに統合することにより，構築や運用管理コストを削減できる．

そこで本稿では，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環境が，それらのネットワークを個々に構築した場合と比較して性能にどの程度影響を与えるかということを，NAS Parallel Benchmark と基礎的な並列処理プログラムを用いて測定し，考察した．

2.　IP-SAN 統合 PC クラスタ

2.1　ローカルデバイスを使用した PC クラスタ

従来，大規模なデータを扱う並列分散処理システムの記憶装置には，ローカル接続のストレージデバイスが使用されてきた．しかし，各サーバ毎に固有のローカルデバイスを所有し管理する形となるため，ディスク毎にデータが分散されており，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．

またこの場合，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，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．

2.2　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．SAN は，分散したストレージをネットワークで統合することによって，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 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，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．そのため，ストレージアクセスを行うクラスタノード (サーバ) –ストレージ間のバックエンドには高速な FC-SAN を用いることが多くなった．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．

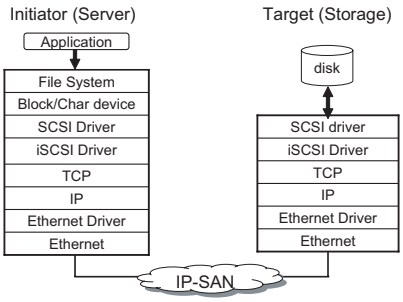

図 3　iSCSI の階層構造

そこで，TCP/IP ネットワークを使用した IP-SAN が次世代の SAN として注目を集めている．従来の FC-SAN の代わりに IP-SAN を用いることにより，PC クラスタを汎用ネットワークで構築し，安価なコストで運用することが可能になる．FC-SAN の導入および管理コストが高いことや 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，今後は 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．

しかし，フロントエンドとバックエンドを個々に構築していたのでは，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．

2.3　iSCSI を使用した IP-SAN 統合 PC クラスタ

IP-SAN のプロトコルとしては，2003 年 2 月に IETF により正式承認された iSCSI(Internet SCSI) [2] が現在最も期待されている．iSCSI では，SCSI コマンドを TCP/IP パケットの中にカプセル化することにより，イニシエータと呼ばれるサーバ機とターゲットと呼ばれるストレージ装置の間で，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．TCP/IP と Ethernet を使用する iSCSI を使用することによって，個々に構築していたノード間のフロントエンドとノード (サーバ) –ストレージ間のバックエンドを一つのコモディティなネットワークに統合した PC クラスタの構築が可能になる．

そこで我々は，図 2 に示すように，ネットワーク構築コストの削減と運用管理の効率化を目的として，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 PC クラスタを実現した [3]．IP-SAN 統合 PC クラスタによって，各トラヒックのデータを一つのネットワークで転送できるようになるだけでなく，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削減できる．

しかし，これら双方のネットワークを統合した場合，フロントエンドのノード間通信とバックエンドのストレージアクセスのバルクデータが，同一の IP ネットワーク経由で混在して転送されることによるネットワークへの負荷が懸念される．例えば，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．

そこで本稿では，バックエンドのネットワークをフロントエ

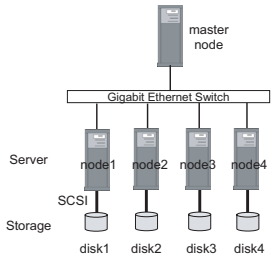

図 4　ローカルデバイスを使用した PC クラスタの実験環境

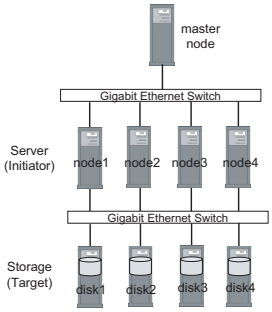

図 5　バックエンド IP-SAN を用いた PC クラスタの実験環境

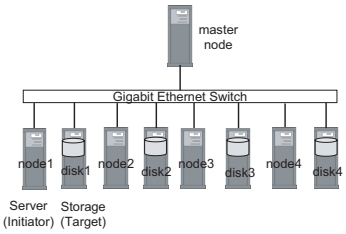

図 6　IP-SAN 統合 PC クラスタの実験環境

ンドのネットワークに統合した IP-SAN 統合 PC クラスタの性能を実験により測定し，考察した．

3. NAS Parallel Benchmark による実験

IP-SAN 統合 PC クラスタが一般的な I/O 処理を伴う並列分散処理を実行した際に，性能にどの程度影響を及ぼすかを調査するため，マクロベンチマークを使用した並列演算性能を測定した．

3.1 実験環境

iSCSI を用いた IP-SAN 統合 PC クラスタを構築し [3]，新たに比較対象として，フロントエンドネットワークとバックエンドネットワークを統合せずに別々に持つ iSCSI 接続 PC クラスタを構築した．

本実験では，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．

- ローカルデバイスを使用した PC クラスタ
- バックエンド IP-SAN を用いたクラスタ
- IP-SAN 統合 PC クラスタ

ローカルデバイスを使用した PC クラスタでは，図 4 に示すように，全ノードが各々のローカル SCSI ディスクにアクセスする．バックエンド IP-SAN を用いたクラスタでは，図 5 に示すように，各ノードはバックエンドのネットワークを介して各々の iSCSI ディスクと接続する．我々が実現した IP-SAN 統合 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とバックエンドを統合したネットワークを介して各々の iSCSI ディスクと接続する．この場合，イニシエータとターゲットは同じ Gigabit Ethernet スイッチで接続され，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．

実験に用いた PC のシステム環境を表 1 に示す．iSCSI ターゲットは PC を用いて構築され，iSCSI の実装には，ニューハンプシャー大学 InterOperability Lab [4] 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．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．

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

また並列分散処理に使用する MPI ライブラリには，MPICH2-1.0.3 [5] を，Fortran コンパイラには，Intel Fortran Compiler 9.0 for Linux を使用した．

3.2 NAS Parallel Benchmark

まずマクロベンチマークとして，HPC 分野などで一般的に使用されている並列ベンチマークである NAS Parallel Benchmark (NPB) を使用して性能を測定した．

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 である EP，MG，CG，FT，IS と，3 つの Parallel CFD (Computational Fluid Dynamics) Application Benchmarks である LU，SP，BT 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 [6] [7] である．このバージョンの NPB は，大量の I/O 処理を行うアプリ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) が使用できる．NPB I/O (BTIO) 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．BT とは，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．実行した 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．

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，それらは次のようになっている．

- full：MPI I/O with collective buffering
- simple：simple MPI I/O without collective buffering
- fortran：Fortran direct I/O
- epio：parallel I/O

full は，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，集められたデータの再構成を行った上で，単一のファイルとして write 処理を実行する．simple は，各ノードのデータは再構成せずに各ノードのデータを単一のディスクに write 実行するため，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 library

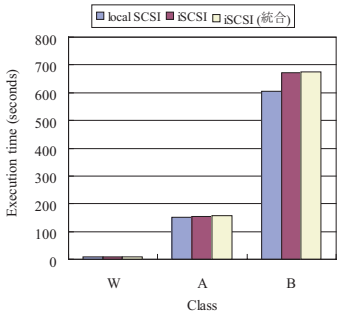

図 7　NPB I/O の実行時間
(simple，node=4)

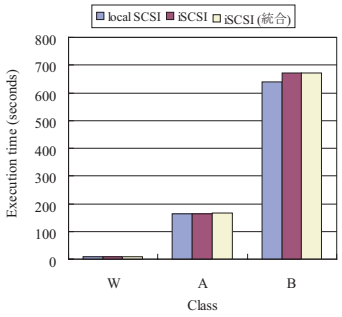

図 9　NPB I/O の実行時間
(fortran，node=4)

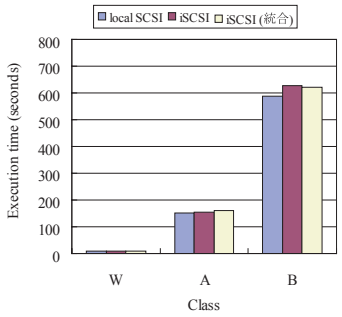

図 11　NPB I/O の実行時間
(epio，node=4)

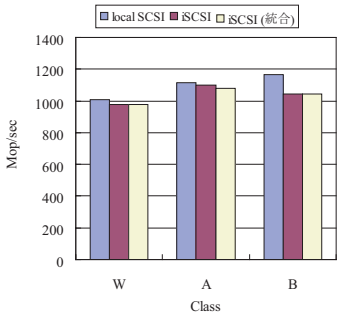

図 8　NPB I/O の Mops 値
(simple，node=4)

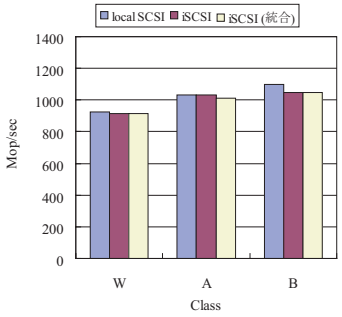

図 10　NPB I/O の Mops 値
(fortran，node=4)

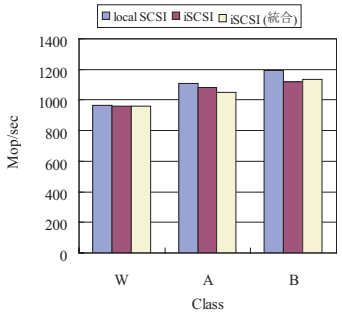

図 12　NPB I/O の Mops 値
(epio，node=4)

calls の代わりに，Fortran 77 のファイル操作を使用する．epio は，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．そのため，他のオプションを実行した場合と異なり，各ノードに作成されるファイルサイズは表 2 の値をノード数で割ったものになる．

本実験では，simple，fortran，epio について 5 回の測定を行った．ただし，full については現在の環境では実行できなかったため，本稿の実験から除外している．

3.3　NPB による実験結果と考察

ローカルデバイスを使用した PC クラスタ，バックエンド IP-SAN を用いたクラスタ，IP-SAN 統合 PC クラスタにおいて，オプションを simple で実行した場合の NPB の実行時間と Mops (Million Operations Per Second) 値を図 7，8 に示す．また，オプションを fortran で実行した場合の NPB の実行時間と Mops (Million Operations Per Second) 値を図 9，10 に示す．オプションを epio で実行した場合の NPB の実行時間と Mops (Million Operations Per Second) 値を図 11，12 に示す．Mops 値は 1 秒間あたりの 100 万演算数である．

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，ローカルデバイスを使用した PC クラスタ，バックエンド IP-SAN を用いたクラスタ，IP-SAN 統合 PC クラスタでは Mops 値，経過時間ともに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，複数ノードを用いた並列処理演算が支配的なためであると考えられる．

一方，Class A の問題サイズでは，オプションによる差はあるものの，IP-SAN 統合 PC クラスタの場合はバックエンド IP-SAN と比較して少し性能が悪くなる．しかしこの場合には，Class A と同様，ローカルデバイスを使用した場合と比較しても大きな差はない．

Class B の問題サイズでは，IP-SAN を用いた場合はローカルデバイス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，単一のディスクにデータを書き出す際に，iSCSI ネットワークを介しているためであると考えられる．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くなっている．epio の場合，単一のディスクにデータを書き出すのではなく，各々のノードのディスクに write を実行してる．そのため，他のオプションと異なり，各ノードの結果データを一つのノードへ転送する処理は行われないため，性能が良くなっていると考えられる．

全体として，Class B における IP-SAN を用いた場合とローカルデバイスを用いた場合との差はあるものの，バックエンド IP-SAN を用いたクラスタと IP-SAN 統合 PC クラスタでは，並列演算処理性能にほぼ差がないといえる．

4.　単純な並列プログラムによる実験

4.1　実験概要

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 PC クラスタでは，ノード間通信とストレージアクセスのデータが，同一のネットワーク経由で混在して転送される．よって，I/O 処理を伴う並列計算などを実行する場合には，ネットワークに負荷がかかり，全体の並列分散処理性能が劣化することが懸念される．

そこで本実験では，IP-SAN を使用した PC クラスタにおいて，バックエンドのネットワークをフロントエンドに統合することによって性能にどの程度影響を与えるのかということを調査するため，マイクロベンチマークを使用した並列分散処理性

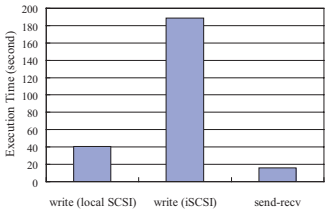

図 13　write，send-recv の各処理の実行時間

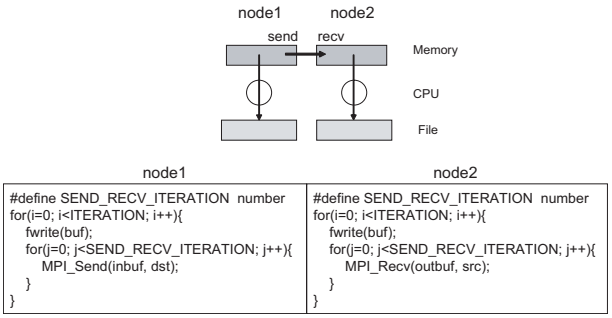

図 14　並列プログラムの動作と擬似コード

能を測定した．実験環境は，図 4〜図 6 と同じであるが，ノード数はいずれも 2 としている．また使用計算機も 1 と同じものを使用している．

4.2　write，send-recv の各処理の実行時間

I/O を伴う並列分散処理では，ノード間でメッセージの送受信 (send，recv)，ノード–ストレージ間でストレージアクセス (read，write) が行われる．図 13 は，1 ノードにおいてストレージデバイスへの write のみを行った場合の実行時間と，2 ノード間でメッセージの送受信 (send-recv) のみを行った場合の実行時間を示している．また，write 単独で実行する際には，ローカル SCSI デバイスと iSCSI を介したストレージデバイスにおいて，I/O ブロックサイズを 1MB とし，最終的に 1GB のファイルを書き出すプログラムを実行している．send-recv 単独で実行する際には，メッセージサイズを 1MB とし，1000 回の send-recv を実行するプログラムによって測定した．

図 13 より，ローカル SCSI デバイスに対する write，iSCSI を介したストレージデバイスに対する write，send-recv の実行時間は，41 秒，189 秒，16 秒である．よって，ローカル SCSI デバイスに対する write，iSCSI デバイスに対する write，send-recv のバンド幅はそれぞれ，24.5MB/sec，5.3MB/sec，62.4MB/sec となる．これらの結果から，iSCSI デバイスへの write とローカル SCSI デバイスへの write には大きな差がある．さらに，iSCSI デバイスへの write を行うストレージアクセスと比較して，send-recv を実行するノード間通信は著しく速いことがわかる．

4.3　メッセージ送信回数変動による実行時間

次に，並列分散処理において，余分な処理を極力省き，write と send-recv を繰り返すだけの最も基礎的な並列処理および I/O を行うプログラムを作成した．図 14 は，作成した並列プログラムの動作と単純化した擬似コードである．このプログラムはまず，各ノードからそのノードに接続しているストレージデバイスに対して write を実行し，その後，送信ノード (ノード 1) では MPI_Send() を，受信ノード (ノード 2) では MPI_Recv() のブロッキング通信を実行する処理を繰り返す．

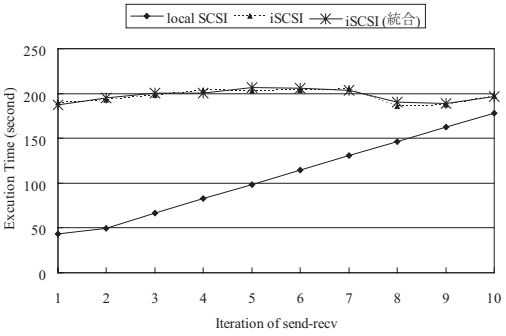

図 15　メッセージ送信回数変動による実行時間
(I/O size:1MB，message size:1MB)

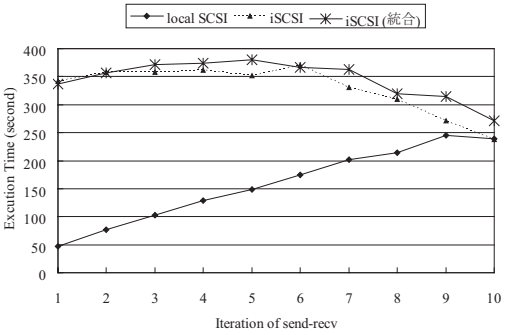

図 16　ノード間通信に負荷をかけた場合における
メッセージ送信回数による実行時間
(I/O size:1MB，message size:1MB)

本実験では，ローカルデバイスを使用した PC クラスタ，バックエンド IP-SAN を用いたクラスタ，IP-SAN 統合 PC クラスタにおいて，図 14 を実行し，ノード 1 における実行時間を測定した．その際，ストレージへの write とノード間通信の send-recv が，同一ネットワーク上でデータ転送されたときの並列分散処理性能への影響をみるため，send-recv の繰り返し回数 (メッセージ送信回数) を増加させた．

図 15 は，I/O サイズを 1MB とし，1 回に送信されるメッセージサイズを 1MB にしたときの実行時間である．最終的に各ノードに接続するストレージに作成されるファイルサイズは，ノード 1 とノード 2 共に 1GB である．同図より，ローカル SCSI デバイスを使用した場合は，send-recv の繰り返し回数が 2 以上になると回数に比例して実行時間が長くなる．これは，図 13 に示すように，ローカル SCSI デバイスへの write が send-recv の約 2.5 倍の実行時間がかかり，send-recv の繰り返し回数が 2 より小さい場合には I/O が支配的になるためである．逆に，send-recv の繰り返し回数が 2 以上の場合には，送信するメッセージ量に依存する．

一方，IP-SAN に統合した場合は，バックエンド IP-SAN を使用した場合と比較して，ほぼ同じ実行時間となり，send-recv の繰り返し回数が 10 までは両者の結果がほぼ一定の値となった．図 13 において，iSCSI ディスクへの write は send-recv の 11.8 倍の実行時間であることから，send-recv の繰り返し回数が 10 までは write の実行が支配的になる．また，本実験で使

用したプログラムでは，write の命令に fwrite() 関数を使用し，ファイルシステムを介してストレージにアクセスしている．その際，write 関数の命令を先に発行しても，システムバッファにデータが書き出された時点ですぐに次の send 関数に移行すると考えられる．これらのことから，send-recv の繰り返し回数が 10 付近では，write と send の各処理の実行の遅延はほとんどなく，write と send の実行がオーバーラップする部分が大きいと考えてよい．

IP-SAN 統合 PC クラスタは，ストレージアクセスのデータとノード間通信により送受信されるメッセージが同じ NIC を介して同一ネットワーク上に混在する．その場合，それらの処理がオーバーラップしたときに性能に影響する可能性があるが，本実験の結果ではバックエンドの IP-SAN をフロントエンドに統合した影響は殆ど見られなかった．これは，iSCSI を介したストレージアクセスの通信性能がかなり低いことが原因の一つと考えられる．つまり，iSCSI の I/O により送信されるパケットの頻度が低く，send-recv を行っている通信に殆ど影響を与えなかったためである．

4.4 ノード間通信に負荷をかけた場合の実行時間

次に，サーバ (ノード) がある並列分散処理を実行している間に他の通信も行っている場合を想定し，意図的にノード間通信を行っているネットワークに負荷をかけ，4.3 節と同様の測定を行った．ここでは，ノード 1 からノード 2 の IDE ディスクに対して，rcp コマンドを使用して 2GB のファイルを繰り返しコピーする作業を実行しながら，4.3 節と同じ並列プログラムを実行させ，ノード 1 における実行時間を測定した．

図 16 は，その結果である．4.3 節の結果と異なり，IP-SAN 統合 PC クラスタは，バックエンド IP-SAN を持つ場合と比較して，並列分散処理性能がやや劣化している．特に I/O の割合よりもメッセージ送受信の割合の方が大きい場合に差が表れている．このことから，ノード間通信を行っているネットワークに大きな負荷がかかっていない場合，またはノードが他の通信を大量に行ったりすることなく，一般的な並列分散処理のみを実行する場合には大きな影響がなく，IP-SAN 統合 PC クラスタは統合していないものと同程度の性能が発揮できると考えられる．

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 [8] 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 [9] 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 [10] 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 [11] では，共有読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 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場合と比較して性能にどの程度影響を与えるかということを評価した．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，基礎的な並列処理および 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．その結果，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，本稿では IP-SAN 統合 PC クラスタの性能評価がまだ限定的である．今後は，カーネル内部の処理を把握，解析しながら，ストレージへのリードアクセスを実行した場合，または並列分散処理を実行している間に，サーバが他の処理を実行する場合など，フロントエンドのネットワークに負荷をかけた状態での性能を評価していく．

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] “Storage Networking Industry Association”. http://www.snia.org/.
[2] “iSCSI Draft”. http://www.ietf.org/rfc/rfc3720.txt.
[3] 神坂紀久子, 山口実靖, 小口正人, 喜連川優：“IP-SAN 統合 PC クラスタを用いたトラヒック特性と I/O 性能に関する考察”, 夏のデータベースワークショップ 2006 (DBWS2006), 電子情報通信学会技術研究報告, DE2006-50〜91.
[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] “NAS Parallel Benchmark (NPB)”. http://www.nas.nasa.gov/Software/NPB.
[7] “NPB-MPI 2.4 I/O”. http://www.nas.nasa.gov/News/Techreports/2003/PDF/nas-03-002.pdf.
[8] W. T. Ng, B. Hillyer, E. Shriver, E. Gabber and B. Ozden: “Obtaining High Performance for Storage Outsourcing”, Proc. FAST 2002, USENIX Conference on File and Storage Technologies.
[9] P. Sarkar, S. Uttamchandani and K. Voruganti: “Storage over IP: When Does Hardware Support help?”, Proc. FAST 2003, USENIX Conference on File and Storage Technologies.
[10] S. Aiken, D. Grunwald, A. Pleszkun and J. Willeke: “A Performance Analysis of the iSCSI Protocol”, Proc. 20th IEEE Symposium on Mass Storage Systems and Technologies (MSS ‘03).
[11] 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, 第 J87-D-I 巻.