4A-3

iSCSIを用いたIP-SAN統合PCクラスタの性能に関する一考察

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 ‡工学院大学 §東京大学生産技術研究所

1 はじめに

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ベース処理等をPCクラスタにおいて実行することが一般的になった．従来より，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]のプロトコルである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．

そこで本稿では，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，コモディティな一つのネットワークに統合したIP-SAN統合PCクラスタをiSCSIを用いて実現する．また，双方のネットワークを統合した環境が，それら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を評価した．

2 IP-SAN統合PCクラスタ

2.1 SANを用いたPCクラスタ

PCクラスタにおいて，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)を用いて構築することが多くなった．SANはサーバ機とストレージデバイスを接続する高速な専用のネットワークであり，分散されたストレージの統合と集中管理を可能にする．IP-SANは，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クラスタの導入，運用ができる．IP-SANの代表的なプロトコルとしてはiSCSIがあり，SCSIコマンドをTCP/IPパケットの中にカプセル化することにより，ブロックレベルのデータ転送を行う．

通常，SANを使用したPCクラスタでは，ノード間のフロントエンドとノード–ストレージ間バックエンドのネットワークを個々に構築する．しかし，その場合は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．

2.2 iSCSIを使用したIP-SAN統合PCクラスタ

PCクラスタにiSCSIを使用することで，個々に構築していたフロントエンドと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することが可能になる．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

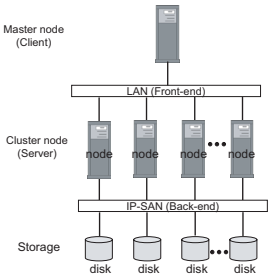

図1: IP-SANを用いたPCクラスタ

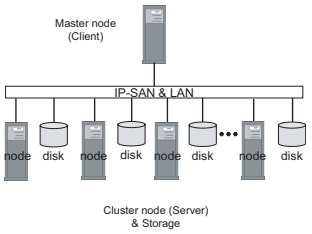

図2: IP-SAN統合PCクラスタ

表1: 実験環境：使用計算機

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

IP-SAN統合PCクラスタを実現した[2]．しかし，その場合，フロントエンドで行われるノード間通信とバックエンドで行われるストレージアクセスのデータが，同一のIPネットワーク経由で混在して転送され，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全体の性能が劣化する可能性がある．

3 IP-SAN統合PCクラスタと非統合PCクラスタの比較実験

本稿では，IP-SAN統合PCクラスタが，ネットワークを個々に構築した場合と比較して並列分散処理性能にどの程度影響を及ぼすかを評価した．

3.1 実験環境

本実験では，ローカルストレージを用いたPCクラスタ，バックエンドにIP-SANを用いた非統合PCクラスタ(図1)，IP-SAN統合PCクラスタ(図2)の3つの環境において性能を比較する．非統合PCクラスタにおけるフロントエンドとバックエンドのネットワーク，そしてIP-SAN統合PCクラスタにおけるネットワークはGigabit Ethernetを用いて構築した．実験に用いたPCのシステム環境を表1に示す．またIP-SANにはiSCSIを使用しているが，本稿の実験環境ではノードとストレージは1対1接続になっており，各ノードは特定のストレージデバイスに接続する構成となっている．

3.2 マクロベンチマーク

まずマクロベンチマークとして，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は並列演算性能を測定するベンチマークであるが，本実験で使用し

Study of Performance in PC cluster system integrated with IP-SAN using iSCSI

Kikuko Kamisaka†, Saneyasu Yamaguchi‡, Masato Oguchi† and Masaru Kitsuregawa§

†Ochanomizu University, ‡Kogakuin University, §Institute of Industrial Science, The University of Tokyo

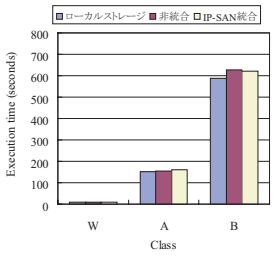

図 3: NPB I/O の実行時間

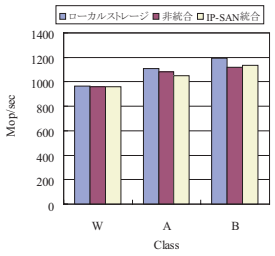

図 4: NPB I/O の Mops 値

た NPB 2.4 I/O は，対象問題 BT (Block Tri-diagonal) に対し，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できる．また本実験において実行した NPB の Class は，W，A，B である．NPB 2.4 I/O の実行オプションとして並列 I/O を行う epio を用いており，Class W においては全ノードで 22MB，A においては 420MB，B においては 1698MB のデータをストレージに書き出す．並列計算を行うノード数は 4 とした．

NPB の実行時間と Mops (Million Operations Per Second) 値を図 3，4 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，3 つの PC クラスタ環境でほぼ同じ値が得られた．これらの Class においては入出力が行われるデータ量が少なく，並列処理演算が支配的なためである．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やや性能差が大きくなる．しかし，IP-SAN 統合クラスタと非統合クラスタでは，並列演算処理性能にほぼ差がないといえる．

3.3 マイクロベンチマーク

次に，マイクロベンチマークとして，ノード間通信とストレージアクセスがほぼ同時に行われるような並列プログラムを作成し，統合による影響をさらに詳細に評価した．

図 5 は，作成した並列ベンチマークの擬似コードである．本実験ではノード数を 2 とし，この並列プログラムを実行した際の処理時間を測定した．このプログラムでは，各ノードからそのノードに接続され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() を実行する．fwrite 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-recv 関数に移行するため，ほぼ同時に write と send-recv が実行されているといえる．図 5 における “SEND_RECV_ITERATION” は，send-recv の繰り返し回数，つまりメッセージ送信回数である．ノード間通信で実行される send-recv とストレージアクセスで実行される write は性能差があると考えられるため，統合の性能への影響が大きくなるような状況を想定し，このメッセージ送信回数を増加させて性能を測定した．

図 6 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されたストレージに作成されるファイルサイズは，各ノードそれぞれ 1GB である．同図より，ローカルストレージを使用した場合には，メッセージ送信回数が 2 以上になると比例して実行時間が長くなる．これは，メッセージ送信回数が 2

node1

```
#define SEND_RECV_ITERATION  number
for(i=0; i<ITERATION; i++){
    fwrite(buf);
    for(j=0; j<SEND_RECV_ITERATION; j++){
        MPI_Send(inbuf, dst);
    }
}
```

node2

```
#define SEND_RECV_ITERATION  number
for(i=0; i<ITERATION; i++){
    fwrite(buf);
    for(j=0; j<SEND_RECV_ITERATION; j++){
        MPI_Recv(outbuf, src);
    }
}
```

図 5: 並列プログラムの擬似コード

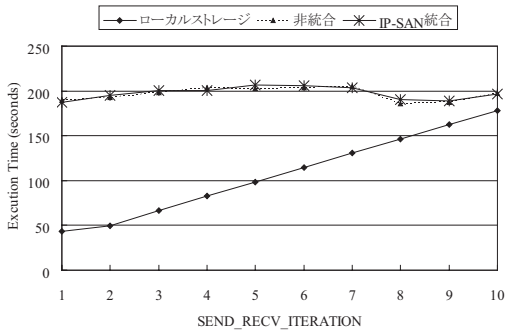

図 6: メッセージ送信回数変動による実行時間

より小さい場合には I/O が支配的になり，メッセージ送信回数が 2 以上の場合には送信するメッセージ量に依存するためである．一方，IP-SAN 統合クラスタの場合は，非統合クラスタの場合と比較してほぼ同じ実行時間となった．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．

IP-SAN 統合 PC クラスタは，ストレージアクセスのバルクデータとノード間通信のデータが同一ネットワーク上に混在することによる性能劣化が懸念された．しかし本稿の実験の結果から，バックエンドの IP-SAN をフロントエンドのネットワークに統合した性能への影響はほとんどなく，有効であるといえる．

4 まとめと今後の課題

PC クラスタの構築および運用管理コストを削減するため，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．本稿では，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．その結果，バックエンドに IP-SAN を持つ非統合 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，有効であることがわかった．

今後の課題として，IP-SAN 統合 PC クラスタの性能を様々な状況においてさらに詳細に評価する．また，IP-SAN 統合 PC クラスタに高速なネットワークを適用することを考える．

参考文献

[1] “Storage Networking Industry Association”. http://www.snia.org/.

[2] 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.