Wire Speed Storage（WSS）アーキテクチャ

大江和一[†] 渡辺高志[†] 西川克彦[†]

Wire Speed Storage（WSS）は高バンド幅のネットワークで結合されたコンピュータシステムに低レイテンシでの応答，ネットワーク帯域を満たすスループット性能の提供を目指したストレージシステムである．Giganet cLAN を用いた試作システムにおいて，1）ネットワーク帯域の 93%の性能が得られること，2）CL3 台からのアクセスで最大 320 MB/sec のスループット性能が得られることを確認した．また，実アプリケーションとして，wwwBLAST を WSS 上で実行し，ローカルディスクと比較して約 2 倍の性能を達成できることを確認した．

Wire Speed Storage Architecture

Kazuichi Oe,[†] Takashi Watanabe[†] and Katsuhiko Nishikawa[†]

In this paper, we present the Wire Speed Storage (WSS) architecture. The WSS is a storage system that aims to provide low latency response and high throughput I/O performance for a computer system connected with a high-bandwidth network. We developed the WSS experimental system using Giganet cLAN and confirmed 1) to turn the 93% of network throughput to the I/O throughput, 2) to turn the 320 MB/sec throughput performance in case the three client nodes access the WSS simultaneously. We also tried to run the wwwBLAST on the WSS and confirmed that the WSS was twice as fast as local disks.

1. はじめに

現在の計算機システムの中でストレージシステムがかかえる課題としては，1）高速化，2）コストパフォーマンスの向上があげられる．高速化に関しては，計算機システムの CPU，メモリ，内部バス，ネットワークが GB/sec オーダのデータ処理が行えるところまで到達しており，ストレージにも同等な処理性能が必要になってきている．コストパフォーマンスに関しては，通信ネットワークとは別に I/O ネットワーク（SAN）を構築することがストレージのコストを押し上げる要因の 1 つになっている．

これらの問題を解決するために，著者らは Wire Speed Storage（WSS）アーキテクチャを提案している．WSS は，InfiniBand や Remote DMA（RDMA）と TCP/IP Offload の実装を前提にした 10 G Ethernet のような低レイテンシ・高スループット通信ネットワークで結合されたコンピュータシステムに，低レイテンシでの応答とネットワーク帯域を満たすスループット性能を提供するストレージシステムである．

本稿では，高速化およびコストパフォーマンスの向上を実現する WSS アーキテクチャについて詳細に説明していく．また，ネットワークとして cLAN☆を使用した試作システムにおける性能評価結果についても報告する．

2. WSS の概要

2.1 WSS の構成

WSS は，StorageEngine（SE），CacheEngine（CE），RealStorage（RS）の 3 コンポーネントを低レイテンシ・高スループットネットワークで相互結合した構成をとる（**図 1** 参照）．

SE はクライアント（CL）側からのアクセス要求時に指定されるストレージ領域と，実際にデータを管理している CE または RS のネットワークアドレス/CE または RS 内アドレスとのマップを管理している．CL からのアクセス要求を SE が受け付けると，実際にデータを管理している CE または RS を検索し，該当コンポーネントにデータ転送命令を送出する．

CE には大容量メモリが実装されており WSS のス

† 株式会社富士通研究所
Fujitsu Laboratories Ltd.

☆ Emulex（旧 Giganet）社の製品で，VI（Virtual Interface）通信をサポートしている．

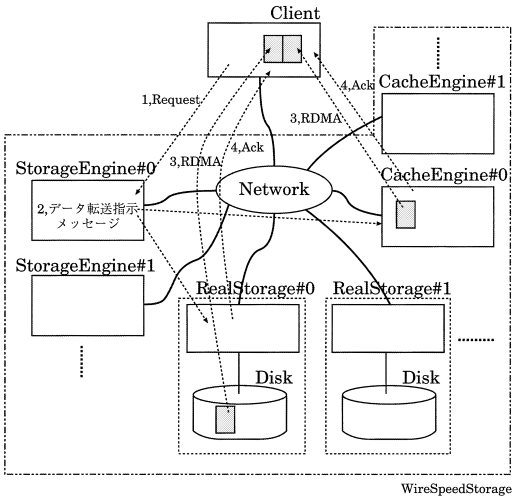

図 1 WSS アーキテクチャ
Fig. 1 WSS architecture.

トレージキャッシュとして機能する．SE からの指示に基づいて CL，SE，CE，RS のどれかと CE 内指定メモリ領域とのデータ転送を直接実行する．データ転送実行後，Ack も直接 CL 側に送信する．

RS はディスク装置を内蔵しており，WSS のデータ格納庫として機能する．SE からの指示に基づいて CL，SE，CE，RS のどれかと RS 内指定ディスク領域とのデータ転送を実行する．データ転送実行後，Ack も直接 CL 側に送信する．

このような構成のアーキテクチャにより，次のような特徴を実現した．

- ワイヤスピードを実現する高性能：
  データ転送をデータを持つ複数のノード（CE，RS）から CL に直接行うことにより，低レイテンシ高スループットを実現する．
- 高コストパフォーマンス：
  モジュール構成を採用することにより，各モジュールをブレードサーバ（**図 2** 参照）のようなコストパフォーマンスの良いプラットフォームを用いて実現できる．たとえば，CE はメモリ容量を増やしたメモリブレード，RS はディスクを増やしたディスクブレードを採用することで高コストパフォーマンス化が図れる．また，ブレードサーバのバックボーンネットワークを I/O ネットワークとして使用することで SAN の構築が不要となる．
- 高スケーラビリティ：
  ネットワーク接続する CE，RS の台数を増減することにより，キャッシュ容量やディスク容量を増減できる．キャッシュ容量は必要なだけとることが可能なため，WSS は CE へのキャッシュヒットを前提とした低レイテンシストレージシステムである．

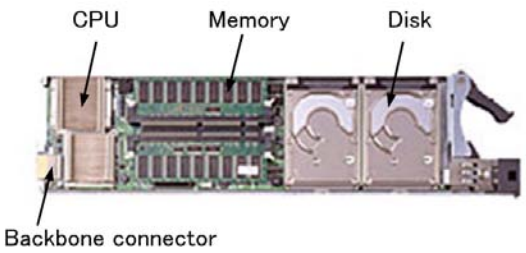

図 2 ブレードサーバ
Fig. 2 Blade server.

2.2 WSS の実現方法

2.1 節で説明した高性能，高コストパフォーマンス，高スケーラビリティを実現するために WSS では以下を行った．

- キャッシュメモリを実装し，CL との RDMA による直接転送が可能な CE をネットワーク上に配置した．CE の制御は SE からのデータ転送指示メッセージで行う．
- 実ディスク装置を実装し，CL との RDMA による直接転送が可能な RS をネットワーク上に配置した．RS の制御は SE からのデータ転送指示メッセージで行う．
- ブレードシステムに適用する場合，CE はサーバブレードよりメモリの実装量を増やしたメモリブレードを使用する．RS はサーバブレードよりディスクの実装量を増やしたディスクブレードを使用する．メモリブレードに関しては，コストパフォーマンスを向上させるためマザーボード上の PCI 等 I/O バス上に安価なメモリを実装することでチップセット等の制約を受けることなく安価にメモリ容量を確保する．
- システム内の SE，CE，RS を増減させることで性能，容量のスケーラビリティを確保する．
- CL を含めた全コンポーネントを同一のネットワークに結合する．

2.3 データ転送方法

2.3.1 概　　要

SE は CL からデータアクセス要求を受け付けると実際にデータを管理している CE または RS を検索し，各 CE または RS ごとにデータ転送指示メッセージを生成&送出する．CE または RS は受信したデータ転送指示メッセージに従い，実際のデータ転送が行われる．

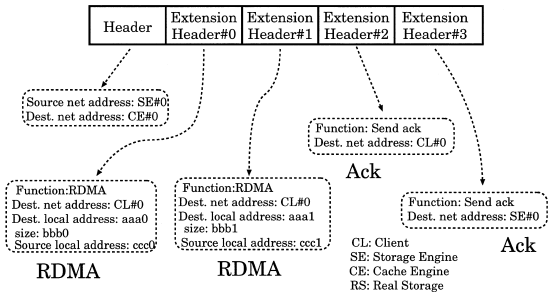

図 3 データ転送指示メッセージ
Fig. 3 Data-transfer-ordered message.

2.3.2 データ転送指示メッセージの生成

データ転送指示メッセージは Header と複数の Extension Header から構成される．各 Extension Header は，CE または RS が実行する 1 処理に対応する．たとえば，CL に RDMA を 2 回実行し CL と SE に Ack を返す場合は，RDMA が指示された Extension Header 2 つと Ack 転送が指示された Extension Header 2 つが添付される（**図 3** 参照）．SE は CL 側からアクセス要求を受け取ると，以下の手順でデータ転送指示メッセージを生成する．

(1) CL 側から受け取ったアクセス要求より，実際にデータを持っている CE または RS のネットワークアドレス，内部アドレス，サイズ情報を SE 内部で管理しているマップ情報を使用して検索する．

(2) 検索結果を使用してデータ転送指示メッセージを生成する．該当するデータが複数の CE または RS にまたがる場合は，CE または RS ごとにデータ転送指示メッセージを生成していく．具体的な転送方法は Extension Header に書き込まれる．転送する領域が複数に離散している場合は，領域ごとに RDMA 等を指示する Extension Header を追加していく．

(3) CL 側に返送する Ack メッセージを生成し，Extension Header としてデータ転送指示メッセージに追加する．

(4) SE 側に返送する Ack メッセージを生成し，Extension Header としてデータ転送指示メッセージに追加する．

2.3.3 Read 処理

Read 処理は CE ヒットした場合とヒットしない場合で動作が異なる．CE ヒットした場合は，データを管理している CE に対してデータ転送指示メッセージを送付することでデータ転送を行う．

CE ヒットしなかった場合は RS に対してデータ転送指示メッセージを送付する．SE でのデータ転送指示メッセージ生成時に CE の空き領域を獲得し，CL への RDMA と Ack メッセージ送信を指示する Extension Header の後に獲得した CE 領域に対する RDMA を指示する Extension Header を追加する（**図 4** 参照）．こうすることで，次回のアクセス要求からキャッシュヒットが望める．

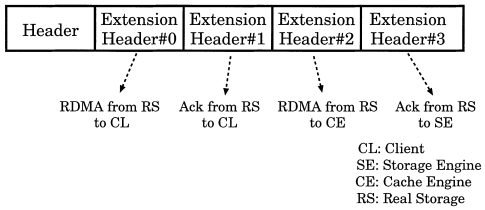

図 4 データ転送指示メッセージ（CE アンヒット時）
Fig. 4 Data-transfer-ordered message (CE unhit).

2.3.4 Write 処理

Write 処理はつねに CE へのデータ転送を行う．SE でのデータ転送指示メッセージ生成時に，該当する領域が存在しなかった場合は CE の空き領域を新たに割り当て，その領域に書き込みを行う Extension Header を添付する．CL からの要求に対応したデータ転送指示メッセージのすべてのデータ転送が完了した後，別途 SE から CE に対して RS 領域への書き込みを指示するデータ転送指示メッセージを送付することで CE 領域のデータと RS 領域のデータを一致させる．

2.4 CE 領域制御

SE 上で行う CE 領域制御について説明する．CE 領域は RS 領域のキャッシュとして使用される．割当て済み領域の状態としては，coherent と dirty の 2 つが存在する．CL から CE 領域へのデータの書き込み等で RS 領域とのデータの一貫性がとれなくなると状態を dirty に遷移させる．dirty 状態の領域のデータは，CL との転送が終了した後 RS に転送され coherent 状態に戻る．状態を遷移させるタイミングは，SE 側がデータ転送指示メッセージに対する Ack メッセージを受信した時点になる．なお，CE から RS にデータ転送を開始するタイミングは設定により CL との転送直後または dirty 状態のデータがある程度溜まった後にまとめて実行のどちらかになる．

CE の空き領域が枯渇したときの領域制御方法としては LRU（least recently used）を使用している．

2.5 高信頼機能

ストレージシステムとして要求される信頼性を確保するため，WSS では以下の方法を採用した．

SE は failover で信頼性を確保する．待機系 SE を

準備しておき，主系 SE がダウンすると，待機系 SE が処理を引き継いで CL からの処理要求に対応する．

CE は dirty 状態の領域に関して複数の CE 領域に冗長にデータを置くことで信頼性を確保する．CL から CE へのデータ転送時，異なった CE の領域にデータを冗長に書き込む．RS 領域へデータの転送が完了した時点で，冗長な領域を解放する．

RS はミラーリングすることで信頼性を確保する．CE の dirty 状態の領域を RS に書き込む場合，ミラーリングされた複数の領域にデータを書き込むことになる．

通信機能の高信頼化として VI（Virtual Interface）connection がエラー等で切断された場合の再 connection 機能を採用する．SE，CE，RS 各コンポーネント間の通信は VI を使用している．VI 通信を行うためには通信を行いたいコンポーネント間であらかじめ VI connection を生成しておく必要がある．

なお今回の試作にともなう性能評価では，低頻度ではあるが VI connection 切断による WSS システムの停止が発生した．このため VI connection 切断時の再 VI connection 機能のみを実装した．CL〜WSS 間のプロトコルとして使用した SRP（SCSI RDMA Protocol）では WSS のように複数のコンポーネント（SE，CE，RS）からの応答を想定してないため，CL 側のフロー制御の問題で大量に通信を行った場合に VI connection が切れてしまう場合があるためである．今回の試作にともなう性能評価では，VI connection 切断以外の問題は起きなかった．

SE，CE，RS の高信頼機能の実装および評価は今後の課題としたい．

2.6 机上性能見積り

WSS アーキテクチャの有効性を示すために，机上性能見積りを行った．見積り方法を単純化するため，CL で転送をシリアライズしたモデルを使用した．CE から RDMA と Ack 送信が 1 回ずつ発生する場合（**図 5** 参照）の処理レイテンシとスループット性能をネットワーク帯域が 100 MB/sec の場合と 1 GB/sec の場合で見積もる．ネットワーク片道レイテンシは cLAN の実測値より 20 μsec（4.2 節参照）に設定した．CL，SE，CE の処理レイテンシは cLAN を用いた実験システムで WSS の基礎性能評価を行った文献 1) での測定結果を元に 20 μsec，40 μsec，20 μsec に設定（**表 1** 参照）した．

クライアントから図 5 のモデルで Read を実行した場合の処理レイテンシの計算方法は，Request，データ転送指示メッセージ，Ack メッセージのレイテンシ（ネットワーク片道レイテンシ），CL，SE，CE 各処理レイテンシ，さらに RDMA 実行に必要なレイテンシを加算することで求めた．100 MB/sec ネットワークの場合は，140 μsec☆ + アクセスサイズ（Bytes）/ 100（MB/sec）で求められ，1 GB/sec ネットワークでは，140 μsec + アクセスサイズ（Bytes）/ 1000（MB/sec）で求まる．

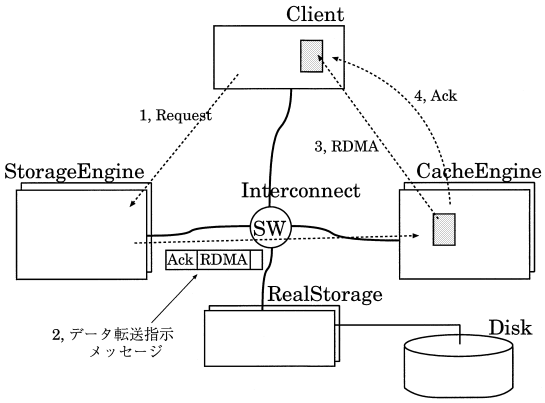

図 5 机上性能見積りモデル
Fig. 5 Performance estimated model.

表 1 机上性能見積りのためのパラメータ
Table 1 Parameters of performance estimate.

項目	見積り値（μsec）
ネットワーク片道レイテンシ (a)	20
CL 処理レイテンシ	20
SE 処理レイテンシ	40
CE 処理レイテンシ	20

(a)：アプリケーションからの実測値 (cLAN)

設定したパラメータを使用して性能を見積もると，100 MB/sec ネットワークの場合で処理レイテンシが 150 μsec（1 KB），スループットが 98.7 MB/sec（1024 KB），1 GB/sec ネットワークの場合で処理レイテンシが 141 μsec（1 KB），スループットが 477.6 MB/sec（128 KB），879.7 MB/sec（1024 KB）となった（**表 2** 参照）．

この机上見積りより，100 MB/sec ネットワークでは WSS の処理レイテンシは 150 μsec 前後になり，従来のディスク装置のレイテンシが数 msec であることと比較して 2 桁近く処理レイテンシが速くなることを示している．スループットは，128 KB のデータ転送で約 90 MB/sec 得られることが分かる．これはネットワーク帯域の 90%に相当する．

一方，1 GB/sec ネットワークでは，WSS の処理レ

☆ ネットワーク片道レイテンシ × 3 + CL 処理レイテンシ + SE 処理レイテンシ + CE 処理レイテンシ

表 2 机上性能見積り結果
Table 2 Results of performance estimate.

アクセスサイズ	100 MB/sec ネットワーク		1 GB/sec ネットワーク	
	A	B	A	B
1 KB	150	6.7	141	7.1
4 KB	180	22.2	144	27.8
16 KB	300	53.3	156	102.6
64 KB	780	82.1	204	313.7
128 KB	1420	90.1	268	477.6
1024 KB	10380	98.7	1164	879.7

A：処理レイテンシ（μsec）
B：スループット（MB/sec）

イテンシは 141 μsec だった．ネットワークスループット以外は 100 MB/sec ネットワークと同じレイテンシ値を使用して見積もったが CPU 性能の向上等でこれら遅延時間が縮まることも予想され，処理レイテンシがさらに縮まる可能性もある．スループットは，128 KB のデータ転送で 477.6 MB/sec 得られ，ネットワーク帯域の 48%くらいにしか相当しないことが分かる．しかし，これは CL で転送をシリアライズする転送モデルを採用したためで，実際には複数のデータを同時に転送することによるパイプライン効果で今回の見積り値よりも大きなスループット性能が得られるはずである．また，128 KB のデータ転送では 477.6 MB/sec だが，1,024 KB のデータ転送ではネットワーク帯域の 88%くらいに相当する 880 MB/sec のスループット性能が得られることが分かる．処理レイテンシが短くなれば，さらに小さなアクセスサイズでワイヤスピードが得られる．

3. 試作システムの概要

試作システムの構成について説明する．ネットワークは，Emulex（旧 Giganet）社の製品で VI（Virtual Interface）通信をサポートした cLAN を使用した．1U PC サーバを cLAN で結合したプラットフォーム上に SE，CE，RS 機能をアプリケーションとして実装した．CL として，WSS との通信機能をアプリケーションとして実装した場合（CL-APP）と SCSI 下のドライバとして実装した場合（CL-DRV）の 2 種類を準備した．実際にユーザアプリケーションが WSS を使用するには SCSI 等のデバイスドライバまたはファイルシステムを通したアクセスとなり，ユーザアプリケーションから見える性能には WSS 以外に SCSI 等のデバイスドライバやファイルシステムのオーバヘッドが加算されたものになる．そこで WSS のみの性能を抽出する目的で CL-APP を，SCSI 下のドライバとしてアクセスした場合の性能を抽出する目的で CL-DRV を設定した．CL〜WSS（SE）間プロトコルは SRP（SCSI RDMA Protocol）（文献 4) 参照）を使用した．

表 3 試作システムの基本仕様
Table 3 Specification of prototype system.

PC	CPU: PentiumIII 1 GHz × 2 Chipset: Server Works HE SL Memory: 1-4 GB Disk: Ultra SCSI 160 (10,000 rpm)
Network	Nic: cLAN1000 Switch: cLAN5000 Driver: cLAN-1.3.0
OS	Linux 2.2.20
WSS	Linux のアプリとして実装 クライアント〜サーバ（SE）間プロトコルは SRP を使用 1CE あたりのキャッシュ容量は 700 MB がデフォルト値 RS 主記憶容量は 64 MB
CL	Linux 上のアプリとして実装（CL-APP） SCSI 下の SRP driver として実装（CL-DRV） 主記憶容量は 1 GB がデフォルト値 PC，Network，OS の仕様は WSS と同じ

試作システムは，CL，SE，CE，RS を各々1 台ずつ実装した基本構成，CE，RS を最大 5 台までつなげた複数台構成の 2 つを準備した．複数台構成の CL 数は 1 台を基本に評価項目によって最大 3 台まで測定可能な環境を準備した．

試作システムの基本仕様を**表 3** に示す．

4. 性能評価およびその考察

4.1 性能評価方法の概要

WSS は従来のストレージ装置とは異なりキャッシュ容量（CE 容量）を大きくとれることがアーキテクチャ上の特徴である．実運用でもほとんどのデータが CE ヒットするような使い方を念頭においている．性能評価では，一部の評価項目を除いて CE ヒットを前提に以下の観点で評価を行った．

(1) SCSI 層等の OS オーバヘッドを除いた WSS の基本性能を評価するため，CL-APP を用いてスループット，レイテンシ，および性能のスケーラビリティを検証する．
(2) WSS の運用時を想定し，SCSI デバイスドライバを通してアクセスを行った場合に関して，スループット，レイテンシ，および性能のスケーラビリティを検証する（CL-DRV 使用）．
(3) 実際のアプリケーションに適用した場合の効果を検証する．

表 3 に試作システムの基本仕様を示したが，CL，CE 数，および CL，CE のメモリ容量等性能に影響

表 4 各性能評価項目ごとの仕様差分
Table 4 Specifications depending on performance evaluation.

	WSS 基本	SCSI 経由	実アプリ
CL 数	1，2，3	1	1
1CL あたりの主記憶容量	1 GB（※）	1 GB	1 GB
CE 数	1，2，4	1，4	5
1CE あたりのキャッシュ容量	700 MB	700 MB	700 MB × 4 300 MB × 1

※：CL-APP では CL メモリ容量は性能に影響しない．

するパラメータは各性能評価項目ごとに必要に応じて変更を行った（デフォルト値は CL 主記憶容量 1 GB，CE キャッシュ容量 700 MB である）．各性能評価項目ごとの設定については**表 4** にまとめた．各性能評価項目の節でも説明する．

4.1.1 WSS 基本性能の評価

WSS 基本性能の評価は CL-APP を使用して行う．CL-APP は WSS CL 用 Linux SCSI driver（CL-DRV）にデバック，性能測定機能を付加し，ユーザプロセスとして起動したものである．そのため CL-APP での性能評価は組み込みの性能測定機能を使用する．性能測定機能は CL-APP のコマンドラインからシーケンシャル性能測定用の rwsync コマンド，ランダム性能測定用の random コマンドを入力することで起動される．rwsync コマンドの引数としては，1）先頭ブロック番号，2）ブロック数，3）繰返し数を与える．rwsync コマンドが起動されると以下のような内部処理が実行される．

（1） Read/Write に必要なメモリ領域の割当て
（2） Read/Write ごとに先頭ブロック番号とブロック数より SCSI コマンドを繰返し数分生成する．
（3） write 用に生成した SCSI コマンドを使用して CL-APP 内部関数として実装してある queuecommand を繰返し数分続けて実行する．なお，queuecommand は CL-DRV と共通の関数で CL-DRV では SCSI の上位ドライバから呼び出される．
（4） Write SCSI コマンド実行完了を待つ．
（5） Read 用に生成した SCSI コマンドを使用して CL-APP 内部関数として実装してある queuecommand を繰返し数分続けて実行する．
（6） Read SCSI コマンド実行完了を待つ．
（7） Read したデータを元データと比較する．
（8） 実行結果を出力する．

random コマンドの引数としては，1）開始ブロック番号，2）終了ブロック番号，3）ブロック数，4）繰返し数を与える．内部処理は，基本的には rwsync コマンドと同一だが SCSI コマンドに与える先頭ブロック番号を乱数で開始ブロック番号と終了ブロック番号の間に入るように設定していく．

rwsync コマンド，random コマンドの双方ともブロック数と繰返し数を調整することで iozone（4.1.2 項参照）を実行した場合と同等なワークロードを WSS にかけることができる．

この CL-APP 性能測定機能を使用して，スループット，レイテンシ関して，当初の目的どおりの性能が得られているかどうかを検証するために CL，SE，CE，RS を各々1 台ずつ実装したシステム構成（基本構成）を使用して検証を行った．

性能のスケーラビリティに関しては，CE，RS を 4 台までつなげたシステム構成（複数台構成）を使用して検証を行った．CL3 台までではあるが，複数 CL を用いた場合のスループット性能の評価を行った．

なお，SE，CE，RS，CL-APP の実装に関しては文献 1) も参照していただきたい．

4.1.2 SCSI ドライバ経由での評価

CL-DRV の性能評価は，ターゲットとなるパーティションに iozone（文献 3) 参照）を実行することで行う．CL-DRV は CL-APP とは異なり，read/write システムコールを介して WSS とアクセスを行うため，I/O ベンチマークとして iozone を使用した．iozone では-i0(write/rewrite)，-i1(read/re-read)，-i2(random-read/write) オプションに加えて Record Size と File Size を与えて実行した．この場合，write，rewrite，read，re-read，random-read，random-write の順番で各 I/O は 1 回あたり Record Size のアクセスを File Size に達するまでの処理が実行される．なお，-i0 オプションはシーケンシャル write 性能を，-i1 オプションはシーケンシャル read 性能を測定するオプションである．

この iozone を使用してスループット，レイテンシ関して，当初の目的どおりの性能が得られているかどうかを検証するために CL，SE，CE，RS を各々1 台ずつ実装したシステム構成（基本構成）を使用して検証を行った．

性能のスケーラビリティに関しては，CE，RS を 4 台つなげたシステム構成（複数台構成）を使用して検証を行った．

なお，CL-DRV の実装に関しては文献 2) も参照していただきたい．

4.1.3 実アプリケーションの検証

実際のアプリケーションによる評価としてはNCBI☆wwwBLAST☆☆を用いて，WSS 上に DB を置いた場合，ローカルディスク上に DB を置いた場合，およびローカルバッファキャッシュ上に DB を置いた場合の比較実験を行った．ファイルシステムは Ext2 ファイルシステムを使用した．

WSS のシステム構成としては，CE を 5 台つなげたシステム構成（複数台構成）を使用して検証を行った．

4.2 cLAN の基礎性能値

WSS の性能評価に際し，cLAN の基本性能を測定した．cLAN の仕様上のスループット値は 1.25 Gbps で，実測値はスループットが 113 MB/sec，レイテンシが 20.5 μsec（Wait☆☆☆の場合）である．

4.3 WSS 基本性能

4.3.1 スループット，レイテンシの評価結果

CL-APP に組み込んだ性能測定機能の rwsync コマンド，random コマンドを利用して測定を行った．CL-APP では，性能測定機能内部で確保したメモリ領域と WSS との間で直接転送を行うため，CL 側主記憶容量の影響は受けない．WSS の CE 容量は 700 MB である．性能測定機能から WSS に対して合計で 256 MB のアクセス量になるように繰返し数を設定して測定を行った．

処理レイテンシは，Read が 168.5 μsec（512 bytes），Write が 206.9 μsec（512 bytes）だった．Write が Read より処理レイテンシが大きいのは，cLAN に RDMA Read 機能がなく，Write 時に CL-APP 側に RDMA Write を実行させているためである．RDMA Read 機能がサポートされたネットワークを使用すれば，Write も Read と同等な処理レイテンシになると予想される．

シーケンシャルアクセス時のスループットは，Read が 112.3 MB/sec（128 KB），Write が 109.0 MB/sec（128 KB）だった．

ランダムアクセス時のスループットは，Read が 112.5 MB/sec（128 KB），Write が 105.6 MB/sec（128 KB）だった（**図 6** 参照）．

なお，本稿では特に注釈がない限り Read はシーケンシャル Read を Write はシーケンシャル Write を指す．

文献 1) により詳細なデータが掲載されているのでそちらも合わせて参照していただきたい．

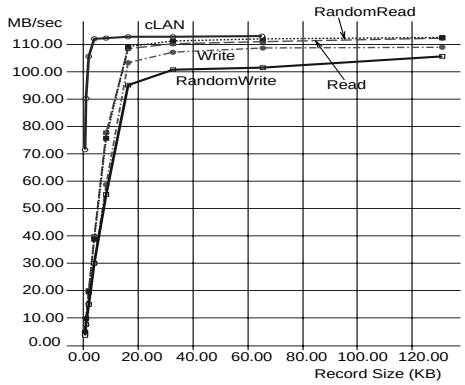

図 6 基本構成のスループット性能（CL-APP）
Fig. 6 Throughput performance of prototype system (CL-APP).

4.3.2 性能のスケーラビリティ

4.3.2.1 CE 数を変化させた場合

CE の領域割当ては RS とは異なり，CE1 台の領域を使い切るまで新たな CE 領域を割り当てない方法で実装している．これは CE1 台でネットワーク帯域を満たすスループット提供が可能なためストライプする必要がないためである．ここでは転送に使用する CE 数が増加することによる性能への影響（主に性能劣化）を検証する目的で測定を行った．

CL-APP によるアクセスなので CL 側の主記憶容量は性能には影響しない．CL からのアクセス量は全体で 256 MB になるように設定した．WSS 側はこの 256 MB のデータを CE1 台にキャッシュ，CE2 台にキャッシュ，CE4 台にキャッシュの 3 通りの設定を行い測定を実施した．256 MB のデータを CE2 台または CE4 台にキャッシュするために，CE1 台あたりの容量を各々128 MB，64 MB に設定して実験を行った．その結果，CE 数に関係なく Read，Write，Random Read，Random Write で 100〜110 MB/sec 前後の性能が得られることが分かった（**図 7** 参照）．

4.3.2.2 RS 数を変化させた場合

WSS では CL からの大部分のアクセスを CE ヒットで賄うことを前提にしているので，CL から RS ストライプでのアクセスは稀なケースとなる．しかし，基礎性能評価の観点で性能評価を行った．

CE を無効にし RS 数を 1，2，4 と変化させることで複数 RS 転送におけるスケーラビリティを検証した．CE を無効にして WSS を立ちあげたので，CL からのリクエストはすべて RS に送られる．CL からのデータ転送量は 256 MB とした．

性能測定結果より，Read，Write に関しては RS 数

☆ National Center for Biotechnology Information
☆☆ Basic Logical Alignment Search Tool
☆☆☆ VIPL による受信待ち方法の 1 つ．メッセージが受信キューに入るまで VIPL 内で Wait する．

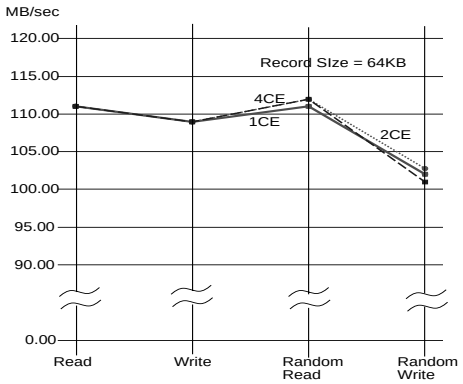

図 7 CE 数を変化させた場合の性能（CL-APP）
Fig. 7 Throughput performance if CE number is moved (CL-APP).

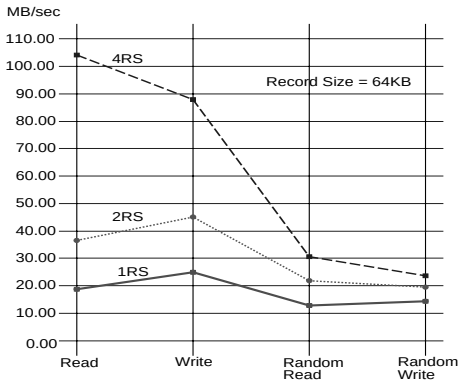

図 8 RS 数を変化させた場合の性能（CL-APP）
Fig. 8 Throughput performance if RS number is moved (CL-APP).

に応じた性能の増加が確認できた．たとえば Read では 19，37，104 MB/sec と性能が増加していく．一方，Random Read，Random Write は RS 数の応じた性能の増加は確認できなかった．たとえば Random Read では 12，21，30 MB/sec であった（**図 8** 参照）．

4.3.2.3 CL 数を変化させた場合

WSS 内に CE3 台準備し CL 数を 1，2，3 台と変化させて測定を実施した．各 CL には異なった CE との間でデータ転送が発生するようにした．各クライアントごとに 256 MB のデータ転送が発生するようにした．

その結果，CL1 台 110 MB/sec，CL2 台 220 MB/sec，CL3 台 320 MB/sec となり，CL 数に比例して性能が増加していくことが確認できた（**図 9** 参照）．

4.3.2.4 CPU 使用率

CL-APP 測定時の各コンポーネントの CPU 使用率も合わせて測定した．その結果，クライアントの CPU 使用率が 30〜50%．SE は 5〜27%．CE/RS は

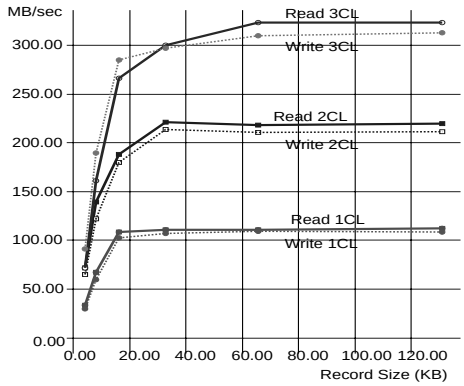

図 9 CL 数を変化させた場合の性能（CL-APP）
Fig. 9 Throughput performance if CL number is moved (CL-APP).

表 5 CL-APP の CPU 使用率
Table 5 CPU busy rate of CL-APP.

コンポーネント	CPU 使用率（%）
クライアント	30-50
SE	5-27（※ 1）
CE/RS	0-50（※ 2）

※ 1：27%は CL3 台で Record Size が 16 KB の値
※ 2：SE から依頼された転送要求に応じて変動．CL 数の影響はなかった．

0〜50%だった．SE の CPU 使用率は CL3 台で Record Size が小さいとき（16 KB）に最大 27%まで増加したが，それ以外の場合は 5〜10%の範囲に収まっていた．CE/RS 数が増減しても SE の CPU 使用率には変化はなかった（**表 5** 参照）．

4.3.3 考　　察

4.3.3.1 スループット，レイテイシ

机上見積り結果より，ネットワーク帯域が 100 MB/sec でも 1 GB/sec でも処理レイテンシが 150 μsec 程度になり，ネットワークが提供する帯域の 90%程度のスループット性能を提供できることが分かった．

実効 113 MB/sec の cLAN で結合された試作システムで性能評価を行ったところ，処理レイテンシが 168.5 μsec，スループットが 105.5 MB/sec（ネットワーク帯域の 93%）となり机上見積りがほぼ正しいことが確認できた．

Read と Random Read 性能はほぼ一致するのに対して Random Write は Write より低い性能値しか出ないことが読み取れる．SE の統計情報より，Random Write ではアクセス領域が重なった Random Write 要求に関して前の命令が終わるまで SE で待ちが発生していることが分かった．このため，Write より低い性能値になったと考えられる．

4.3.3.2 CE 数を変化させた場合

複数 CE からの転送では，SE は CE ごとにデータ転送指示メッセージの生成&転送が必要になる．SE 内部の管理テーブルも増加するため，複数 CE からの転送にともなう SE オーバヘッドの増加が懸念された．しかし，今回の 4CE までの実験では複数 CE から CL 側に転送することによる性能低下は観測されなかった．

CE からのアクセスでは，1CE でワイヤスピードに到達するので，複数 CE からのアクセス時に性能低下が起きないことが重要と考えている．今回の性能評価では CE 数に関係なく 110 MB/sec 前後の値がつねに得られることが分かった．

Random Write 性能が他と比較して 10 MB/sec 程度性能値が低いのは 4.3.3.1 項と同様でアクセス領域が重なった要求の前命令の完了待ちのためと考えられる．

4.3.3.3 RS 数を変化させた場合

複数 RS を用いた場合の性能については，4 ストライプまで性能が向上していくことが確認できた．たとえば Read では，RS1 台 19 MB/sec，RS2 台 37 MB/sec，RS4 台 104 MB/sec となり，ディスク数に応じた性能が出ていることが分かる（図 8 参照）．RS2 台から RS4 台で性能が 3 倍近く増加しているが，これは RS2 台と RS4 台時での各 RS の disk read 性能に違いが出たためである．なお，RS のバッファキャッシュが性能評価に影響しないようにするため，各 RS は 64 MB の主記憶容量で初期化し，各 RS ごとに 160 MB のアクセスが発生するようにした．

RS2 台で性能測定を行うと各 RS あたり 18 MB/sec の性能☆となったが，RS4 台で性能測定を行うと各 RS あたり 28 MB/sec の性能となった．これは RS 数，Record Size，Stripe Size の関係より，各 RS でのディスクアクセスにともなうシーク量が異なるためと考えられる．

Random Read/Random Write では台数効果は認められるが，Read/Write と比較して特に RS 数が 2 台，4 台での性能増加が乏しいことが分かる．RS2 台では各 RS あたり 11 MB/sec（Random Read），4 台では 8 MB/sec（Random Read）となり，RS 数が増えることで RS 単体性能が下がることが原因だった．RS 数が増えデータが分散することでシーク時間が長くなったためと考えられる．

4.3.3.4 CL 数を変化させた場合

複数 CL を用いた場合の性能については，各 CL に各々異なった CE から転送が発生するように設定した場合で CL 数に応じた性能向上が確認できた．CL3 台までであるが，最大 320 MB/sec のスループットが得られた．

実験データとして取得してないが CE1 台から複数 CL にデータ転送を行った場合，CE1 台のスループット（100 MB/sec 前後）で抑えられることが容易に予想される．今回の実験結果のように，複数 CL との転送を行う場合は CL ごとにアクセスが発生する CE を分散するようにすると CL 数に応じた性能向上が得られる．

4.3.3.5 CPU 使用率

SE の CPU 使用率は 10%程度であることが分かった．CE/RS は 0〜50%だった．CL3 台で Record Size が小さい（16 KB）と SE の CPU 使用率が 27%位まで上昇することが分かった．ここまでの実験結果より，CE4 台までではあるが CL1 台へのデータ転送に関わる CE 数が増加しても CL に提供する性能に影響しないことが分かっている（4.3.3.2）．また，異なった CE から転送が行われる前提で CL3 台までではあるが CL 数に比例してリニアに性能向上することが分かっている（4.3.3.4）．これらの場合における CE の CPU 使用率が CL 数によって変動しないことも分かっている（表 5）．

今回測定したデータのみでは，CL 数を 4 以上に増やした場合の SE，CE，RS の CPU 使用率を予測することはできないが，SE の CPU 使用率がネックでサポート CL 数に制限が出る可能性があることは分かる．SE の処理能力がネックになった場合は複数 SE に処理を分散する等が必要になってくる．

4.4 SCSI ドライバを経由した場合の性能検証

4.4.1 スループット，レイテンシの評価結果

CL-DRV の評価では，CL 側主記憶容量を 128 MB，CE 容量を 700 MB に設定して iozone を実行した．iozone は合計 256 MB のデータ転送を行うように設定した．iozone から CL-DRV へのアクセス方法は，通常のブロックデバイス経由に加えて WSS に対応するブロックデバイスに Linux raw キャラクタデバイスをバインドした Linux raw キャラクタデバイス経由での評価も行った．通常のブロックデバイス経由で測定を行うと Record Size☆☆が 64 KB 未満ではカーネル内部で read/write 要求がが蓄えられ，WSS に対して

☆ 各 RS の性能は各 RS の統計情報採取機能を使用して測定できる．

☆☆ 1 回の read/write 時に指定するサイズ

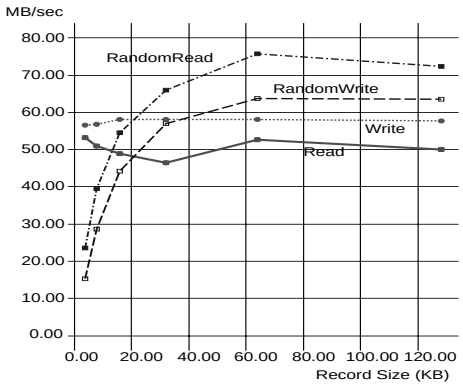

図 10 基本構成のスループット性能（CL-DRV，block device）
Fig. 10 Throughput performance of prototype system (CL-DRV, block device).

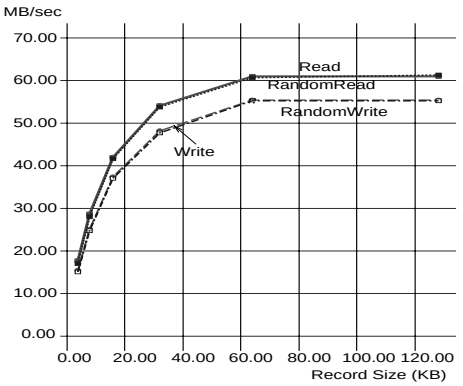

図 11 基本構成のスループット性能（CL-DRV，raw device）
Fig. 11 Throughput performance of prototype system (CL-DRV, raw device).

は 1 回のリクエストの大部分が 64 KB になっていることが分かった．iozone から read/write 要求が呼び出されたときの Record Size そのままの値で WSS に送出した場合の性能を測定するために Linux raw キャラクタデバイス経由での測定も追加した．

処理レイテンシは，Read が 225 μsec（4 KB），Write が 271 μsec（4 KB）だった．CL-APP より処理レイテンシが増加しているのは，Linux カーネルの処理オーバヘッドが加わったためである．Read より Write の方が処理レイテンシが大きいのは CL-APP と同じ理由である．

シーケンシャルアクセス時のスループットは，ブロックデバイス経由時に Read が 52.6 MB/sec（64 KB），Write が 58.2 MB/sec（64 KB）だった．Linux raw キャラクタデバイス経由時に Read が 61.1 MB/sec（128 KB），Write が 55.4 MB/sec（128 KB）だった．なお，ブロックデバイス経由と Linux raw キャラクタデバイス経由でピーク性能が出る Record Size が異なるのは，ブロックデバイス経由では 128 KB のアクセス要求が WSS に送出されないためである．

ランダムアクセス時のスループットは，ブロックデバイス経由時に Read が 75.7 MB/sec（64 KB），Write が 63.8 MB/sec（64 KB）だった．Linux raw キャラクタデバイス経由時に Read が 61.3 MB/sec（128 KB），Write が 55.2 MB/sec（128 KB）だった（**図 10**，**図 11** 参照）．

文献 2) により詳細なデータが掲載されているのでそちらも合わせて参照していただきたい．

4.4.2 性能のスケーラビリティ

4.4.2.1 CE4 台時のスループット性能

CE4 台（容量 2.8 GB）環境に iozone 2 GB を実行した．CL 側の主記憶容量は 1 GB に設定した．iozone 2 GB の 1 回あたりのアクセスサイズは 128 KB〜16 MB まで変更して実行した．その結果，つねに 70 MB/sec 前後のスループット性能を得られることが確認できた（**図 12** 参照）．

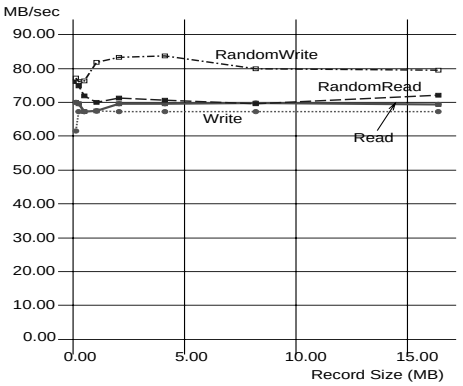

図 12 4CE（容量 2.4 GB）時のスループット性能（CL-DRV）
Fig. 12 Throughput performance of 4CE (2.4 GB, CL-DRV).

4.4.2.2 RS 数を変化させた場合

CE を無効にし RS 数を 1〜4 まで変化させることにより，CL〜RS 間で直接転送を行った場合の性能のスケーラビリティを検証した．CL 側の主記憶容量は 1 GB に設定し，iozone 2 GB を実行した．RS 数が 2〜4 の場合は RS 間でストライプ構成とし，ストリップサイズは 16 KB とした．

その結果，ストライプ数 2 までは性能が向上するがストライプ数 3 以降はかえって性能が低下することが分かった（**図 13** 参照）．ストライプ数 4 のケースで Ack を SE でまとめて 1 つ返すようにしたら性能が低下しないことも分かった（**図 14** 参照）．

4.4.2.3 CPU 使用率

iozone 2 GB 実行時の各コンポーネントの CPU 使

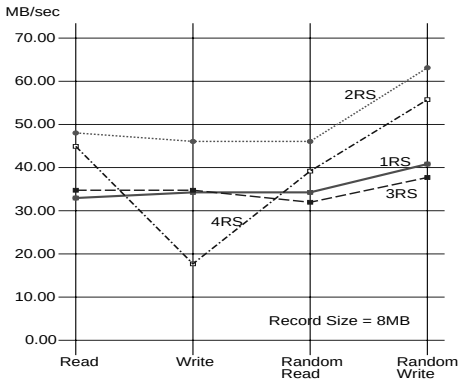

図 13 RS 数を変化させた場合の性能（Record Size=8 MB）
Fig. 13 Throughput performance if RS number is moved (Record Size=8 MB).

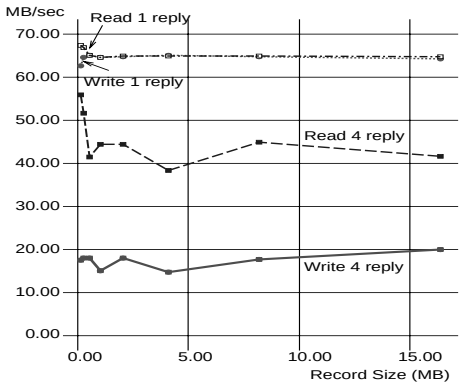

図 14 SE から Ack をまとめて返した場合の性能
Fig. 14 Throughput performance if SE replies Ack message.

表 6 iozone 2 GB 実行時の CPU 使用率
Table 6 CPU busy rate on executing iozone 2 GB.

コンポーネント	CPU 使用率（%）
クライアント	30-80（※）
SE	5-10
CE/RS	0-50

※：RS 数を増やすと CPU 使用率が上昇した．

用率もあわせて測定した．その結果，クライアントの CPU 使用率が 30〜80%．SE は 5〜10%．CE/RS は 0〜50%だった．CE/RS 数が増減しても SE の CPU 使用率には変化はなかった（**表 6** 参照）．

4.4.3 考　　察

4.4.3.1 スループット，レイテンシ

SCSI 下からアクセスすると処理レイテンシが 225 μsec，スループットは 61.6 MB/sec になった．机上見積り値，WSS 単体の性能評価結果と比較して処理レイテンシが増加したのは，Kernel 内 SCSI 共通層のオーバヘッドが加わったためである．スループットが低下したのは，CL 側 kernel 内でコピーが発生するためである．今回使用した PC のメモリ間コピー性能が約 200 MB/sec のためネットワークスループット（約 110 MB/sec）と合わせると 70 MB/sec 程度が上限となる．SCSI 経由からのアクセスでワイヤスピードにするためには，1）SCSI 共通層のレイテンシ削減，2）CL 側 kernel 内のコピーの削減，または CL 側ノードにメモリ間コピー性能が高いものを選択する必要がある．

表 7 WSS へのリクエスト数
Table 7 Request number to WSS.

アクセスサイズ (KB)	16	32	64	合計容量
random-read	6	112	2832	185 MB
random-write	1	33	3649	234 MB

※ iozone random-read/write256 MB を実行した．

4.4.3.2 ブロックデバイスと Linux raw キャラクタデバイスの性能差

Record Size が 4 KB のときブロックデバイス経由では 50 MB/sec 以上の性能が得られているのに，Linux raw キャラクタデバイス経由では 20 MB/sec 未満である．これは，ブロックデバイスはバッファキャッシュ経由のアクセスとなり，64 KB 単位で WSS 側にアクセス要求が出るためである．ブロックデバイス経由の 4 KB アクセスでも Random Read/Random Write は 20 MB/sec 前後の性能しか得られないことが分かる．これはシーケンシャルアクセスのように 64 KB 単位で WSS 側にアクセス要求を出せないためである．

ブロックデバイス経由の Random Read/Random Write 性能に関して，Record Size が 32 KB 以上で Read/Write 性能を上回ることが測定結果より確認できる．strace☆を利用して iozone の動作を調査したところ Random Read/Random Write では同じファイル領域に複数回アクセスが発生していることが確認できた．また，SE の統計情報より CL との総転送量が 256 MB に届いてないことも確認できた．**表 7** は Record Size 64 KB で iozone Random Read/Random Write 256 MB 実行時に SE が受け付けたアクセスサイズごとの Read/Write リクエスト数およびその合計容量である．この表より Read の総転送量が 185 MB，Write の総転送量が 234 MB しか行われてないことが分かる．これらデータより，バッファキャッシュの同じ領域に複数回リクエストが発生

☆ Linux でアプリケーションが実行したシステムコールをダンプするコマンド

し，結果として WSS 側へのアクセス量が減少したため Read/Write 性能を上回った．

ブロックデバイス経由の性能に関して，Record Size=128 KB のときに性能が下がってしまう（ピークは 64 KB）．原因としては 128 KB のときもデータを分割して WSS には 64 KB 単位でアクセスを行ってしまうため，Kernel 内部で何らかの性能劣化が発生したためと考えてる．SE の統計情報より 128 KB 以上のアクセス要求が発生していないことは確認した．

ピーク性能がブロックデバイス経由の方が Linux raw キャラクタデバイス経由より低い理由はバッファキャッシュを経由することによるコピー等のオーバヘッド増加のためと考えている．

4.3.3.1 項では，Write より Random Write の方が性能が低くなる理由について考察した．しかし図 11 では，Write と Random Write の性能値がほぼ一致していることが読み取れる．SE の統計情報より SE が受け付けた SRP コマンドレベルでは待ちの発生はなく，Random Write 性能が劣化しなかった．iozone の Random Write 生成方法が CL-APP 性能測定機能と異なるため，ほぼ同じタイミングで領域が重なる Write 命令が出なかったことがその原因と考えている．

4.4.3.3 CE4 台時のスループット性能

CE4 台からのアクセスでも 70 MB/sec 前後の値がつねに得られ，CL-DRV でも複数 CE からの転送による性能低下がないことが確認できた（図 12 参照）．

また，Random Write が他のアクセス方法より 10 MB/sec 程度性能が良いことが読み取れる．これは，Random Write した領域に関して同じ領域に 2 重，3 重に書き込んでしまう場合があり実ストレージ（WSS）へのアクセス量が減少したためである（4.4.3.2 参照）．

4.4.3.4 RS 数を変化させた場合

性能測定結果より 2 ストライプまでは性能が向上するが 3 ストライプ以降はかえって性能が低下することが分かった．原因は CL 側の CPU 負荷が高いためであることが分かった．今回の実装では，ストライピングでは，関係するすべての RS から Ack を CL に返しており，この Ack 処理が性能低下の原因であることが分かった．CL 側の CPU 負荷を下げる目的で SE でまとめて 1 つ Ack を CL に返すようにする実験を行ったところ，Read で 30 MB/sec，Write で 40 MB/sec 程度だったスループット性能が各々60 MB/sec 以上の性能値になることが確認された．複数 RS をストライピングする場合は，CL の CPU 使用率をどうやって下げるのかがスループット性能を得るために重要であることが分かった．Linux の SCSI フレームワークのオーバーヘッドの問題もあるが，WSS 側としては Ack 数を減らすことが有効であることが分かった．

4.4.3.5 CPU 使用率

SE の CPU 使用率は 10%程度，CE/RS の CPU 使用率は 0〜50%で現在のシステム規模では CPU ネックとなっている箇所がないことが確認できた．

4.5 実際のアプリケーションによる評価

4.5.1 ローカルディスクの基礎性能評価

WSS との性能比較に利用するローカルディスクの基礎性能評価を iozone を使用して行った．

ローカルディスクは，Ultra SCSI160 10,000 rpm のディスク 2 台で software RAID を構成したボリュームを使用した．software RAID は Linux に添付されていた mkraid コマンドを使用して生成した．RAID LEVEL 0，ディスク数 2，ストライプサイズ 8 KBytes とした（**表 8** 参照）．ノードの仕様は WSS 側と同等である．このボリュームに対する iozone 実行結果を**図 15** に示す．測定結果より，Read がつねに 70 MB/sec 位，Write がつねに 55 MB/sec 位の性能になっていることが読み取れる．ランダム性能に関しては，Record Size が 4 MB 以上でシーケンシャル性能を上回ること

表 8 software RAID 構成パラメータ
Table 8 Parameters of software RAID.

raiddev	/dev/md0	
	raid-level	0
	nr-raid-disks	2
	persistent-superblock	0
	chunk-size	8
	device	/dev/sdb
	raid-disk	0
	device	/dev/sdc
	raid-disk	1

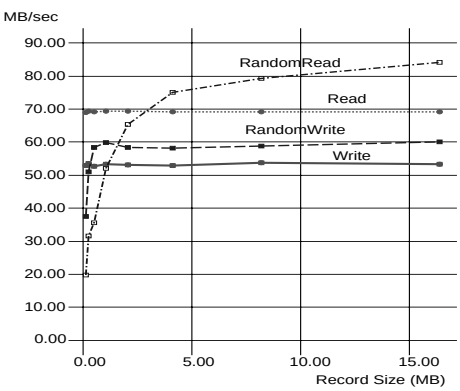

図 15 wwwBLAST に使用したローカルディスク性能
Fig. 15 Performance of disk that is used www BLAST estimate.

が分かった．これは 4.4.3.2 項と同じく一部のデータがバッファキャッシュヒットしたためである．測定結果にバッファキャッシュの影響が出ないようにするため，主記憶容量を超えるサイズを指定して iozone を実行した（主記憶容量 1 GB に対して iozone のデータサイズ 2 GB）．ランダムリードに関しては，一部のデータが一度読み込んだデータと重なってしまい実ストレージまでアクセスがいかなかったためシーケンシャルリード性能を上回った．ランダムライトに関しては，同じ領域に 2 重，3 重に書き込んでしまう場合があり，実ストレージへのアクセス量が減少しシーケンシャルライト性能を上回った．

4.5.2 wwwBLAST の評価

実際のアプリケーションに WSS を適用した場合の効果を検証するために NCBI wwwBLAST を用い，WSS に DB を置いた場合，ローカルディスクに DB を置いた場合，CL バッファキャッシュに DB を置いた場合で性能を比較した．DB の容量は約 3 GB☆である．

WSS 側は CE の容量を 3.1 GB に設定して wwwBLAST を動かすノードにマウントした．wwwBLAST を動かすノードの主記憶容量は 1 GB である．なお，wwwBLAST を動かすノードには WSS の各コンポーネントを動かしているものと同じ仕様の PC を使用した（表 3 項参照）．

ローカルディスク側は，Ultra SCSI160 10,000 rpm のディスク 2 台で software RAID を構成したボリュームを使用し，このノード上で wwwBLAST を動作させた（4.5.1 項参照）．

バッファキャッシュ側は，主記憶容量を 2 GB に設定したノード上で wwwBLAST を動作させた．ノードの仕様は WSS 側と同等である．DB の容量は約 3 GB だが，実際にアクセスが発生するのは 2 GB 未満の領域になるため，主記憶容量 2 GB ですべてのアクセスにおけるバッファキャッシュヒットが望める．

WSS 側もローカルディスク側もシーケンシャルアクセスで 70 MB/sec 程度の性能が得られる構成にして実験を行ったが，wwwBLAST で検索を実行すると WSS 側はローカルディスク側の 1/2 以下の実行時間で検索が終了した☆☆．そのときの WSS 側スループットが 55〜60 MB/sec 程度だったのに対し，ローカルディスク側は 20〜25 MB/sec 程度しか出なかった．

☆ 検索時に実際にアクセスされる領域は 2 GB 未満．

☆☆ 最短ケースで WSS 28 秒，ローカルディスク 72 秒だった．Query によってはこれより検索時間は増加したが，実行時間比はほぼ同じだった．

表 9 wwwBLAST 実行結果
Table 9 Results of www BLAST.

	A	B	WSS
実行時間（秒）	72	14	28
I/O スループット（MB/sec）	20-25	0	55-60
検索サーバ CPU 使用率（%）	25-30	100	90-100

A：ローカルディスク（主記憶 1 GB）
B：バッファキャッシュ（主記憶 2 GB）

バッファキャッシュ側は WSS 側よりさらに高速でほぼ WSS の 1/2 の時間で検索が完了した．このとき，ストレージへのアクセスはまったく発生しておらず，すべてバッファキャッシュヒットしたことが分かる（**表 9** 参照）．

なお，WSS に DB を置いた場合はあらかじめ CE 上に DB データをキャッシュした後に実験を行った．CL バッファキャッシュに DB を置いた場合もあらかじめ CL バッファキャッシュ上に DB データをキャッシュした後に実験を行った．

4.5.3 考　　察

wwwBLAST での実験は，表 9 からローカルディスクでは I/O がボトルネックとなっていたが，WSS ではディスクの I/O 性能が向上し CPU がボトルネックとなっていることが分かる．ローカルディスク側は 256 KB 以下のランダム Read が大量に発生したため，I/O 待ちとなった．

一方，バッファキャッシュと比較すると WSS は約 2 倍の検索時間がかかっていることが分かる．バッファキャッシュヒットした場合は，I/O 処理は主にノード内メモリコピー時間のみとなり，WSS と比較して半分の時間での検索が可能になった．しかし，WSS は複数の CL 間でのデータ共有が可能である．今回実験したシステムは SRP を使用して各 CL に SCSI device としてマウントしたためデータ共有はできないが，WSS で DAFS（Direct Access File System）（文献 5) 参照）サポートも可能であり，この場合複数 CL に対してデータ共有機能が提供され，複数 CL で同時に検索の実行が可能になる．複数 CL からのアクセスでは，WSS が提供可能な帯域は，「データを保存している CE 数 ×1 ポートの帯域」となる．なお，今回の実装における CE 領域割当ては，CE1 台の領域を使い切るまで新たな CE 領域を割り当てない，という方法で行っている．複数 CL に十分な帯域を提供するには適当な単位で分割してデータを複数の CE に分散して置くようにする必要があるだろう．

最後に WSS に対する iozone 実行結果（図 12 参照）と wwwBLAST 実行時に計測されたスループット性能

値（表 9 参照）とを比較する．iozone では 70 MB/sec 位の性能が得られているのに wwwBLAST 実行時は 60 MB/sec 前後の性能しか得られてないことが分かる．性能が低くなった原因としては，1）wwwBLAST 実行で CPU 時間が消費されてしまった，2）EXT2 ファイルシステムのオーバヘッドが加算されたため，を想定している．

5. 関連研究

WSS に関連する研究，製品について述べる．

東京大学のネットワーク RAID 方式[6),7)] は，Ethernet 等で相互に結合した PC/WS 上のローカルディスクを使用してネットワークレベルで RAID を実現している．また，ネットワーク RAID 上にネットワーク RAID ファイルシステム（NRFS）を実現し，NFS でのアクセスを可能にしている．ネットワーク RAID 方式は，専用のキャッシュを持たないため，ランダムアクセス時の性能が十分に得られない可能性がある．

東京大学のデータレゼボワール[8)] は高レイテンシ高スループットネットワークを介して遠隔のデータ共有を実現している．ネットワークの帯域を使い切ることに関しては WSS と共通の課題解決を目指しているといえるが，WSS は I/O ネックアプリケーションの実行時間短縮を目指しているのに対し，データレゼボワールは遠隔のデータ共有を，ユーザプログラムからトランスペアレントに実現することを目指している．

日立製作所のハイブリッドスターネット方式[11)] は RAID 装置内部のネットワークに関して，大きなサイズのデータ転送にはスループット重視のネットワークを使用し，小さなサイズの制御情報転送にはレイテンシ重視のネットワークを使用することを提案している．提案のネットワークを採用した RAID 装置で性能評価を行い 920 MB/sec のスループットと 160 kIO/sec のトランザクション性能を達成した．これは従来タイプのバス構造 RAID と比較して 2.5〜5 倍の性能向上に相当する．WSS と比較した場合，このような従来型 RAID 装置は，1）キャッシュ搭載容量のスケーラビリティが小さく，ほぼ 100%のキャッシュヒットを狙える WSS と比較してキャッシュミスの割合が高くなること，2）通信ネットワークとは別に I/O ネットワーク（SAN）の構築が必要になること，3）RAID 装置内部ネットワークに専用のものを準備する必要がありコスト面で割高であること，がいえ，実効性能とコストパフォーマンスの両面で WSS が上回る可能性がある．

すでに製品として販売されているシリコンディスクについて述べる．性能面では専用 I/O ネットワーク経由ではあるが 350 MB/sec のスループット性能を達成している製品☆も存在する．この性能値は cLAN ベースで実装した WSS より高い値になるが，表 2 の机上見積り結果より InfiniBand や 10 G Ethernet 等の 1 GB/sec ネットワークを使用してアクセスサイズが 128 KB を超えると WSS の性能の方が上回ることが分かる．また，シリコンディスク製品は InfiniBand や 10 G Ethernet 等の通信ネットワークとは別に専用の I/O ネットワークを準備する必要がある．システム全体で 128 GB 位までの容量を提供できる製品も存在するが，1 台あたり 16 GB 前後の容量が上限でそれ以上は別ボリュームとして認識されるため運用時の柔軟性に問題が残る．

6. まとめ

100 MB/sec および 1 GB/sec 帯域のネットワーク上で帯域の 90%を利用できる WSS アーキテクチャを提案し，100 MB/sec 帯域のネットワークを用いた試作システムにより以下が実証できた．

- CL 側からデバイスドライバのオーバヘッドを取り除いたアクセス方法でネットワークが提供する帯域の 93%（105.5 MB/sec）に相当する I/O 性能を実現した．
- 複数 CL 側からデバイスドライバのオーバヘッドを取り除いたアクセス方法で CL3 台までではあるが最大 320 MB/sec のスループット性能が得られた．
- CL 側から SCSI ドライバを経由したアクセス方法で 70 MB/sec のスループット性能が得られた．
- wwwBLAST DB を WSS 上に置くと，ローカルディスク上に DB を置いたときに問題であった I/O ボトルネックを解消し CPU ネック問題になることを確認した．

また，CL 側にメモリを大量に実装したシステムとの比較も行った．その場合の性能比較では WSS の方が劣ることが wwwBLAST の実験結果より分かったが，WSS は，1）複数 CL 間でデータ共有が可能なこと，2）ストレージとしての信頼性が提供できること，3）コストパフォーマンスで有利となること，という利点がある．

7. 今後の課題

性能面に関しては，1）より帯域の大きなネットワー

☆ QikDATA，PLATYPUS Technology

クでの WSS の有効性の評価，2）大きなシステム構成でのスケーラビリティの評価，3）wwwBLAST 以外で I/O ネックになっているアプリケーションに適用した場合の評価を考えている．

また，高信頼機能を加えたうえでの評価もあわせて行っていきたいと考えている．

今回の評価では，SRP を使用して WSS とのアクセスを行ったが，RDMA 転送による高速ネットワークファイルシステムである DAFS（Direct Access File System）（文献 5) 参照）でアクセスした場合の評価も今後行う計画である．

参 考 文 献

1) 大江和一，渡辺高志，西川克彦：Wire Speed Storage (WSS) アーキテクチャ，SWoPP2002, 信学技法，Vol.102, No.275, pp.1–6 (2002).
2) 渡辺高志，大江和一，西川克彦：ワイヤスピードストレージアーキテクチャ（WSS）における SCSI RDMA Protocol（SRP）の実装と評価，SWoPP2002，信学技法，Vol.102, No.275, pp.7–12 (2002).
3) http://www.iozone.org/
4) http://www.t10.org/
5) http://www.dafscollaborative.org/
6) 松本　尚：NIC を活用したネットワーク RAID 方式の提案，情報処理学会研究会報告，Vol.2000, No.74, pp.79–84 (2000).
7) 松本　尚：ネットワーク RAID ファイルシステム．http://www.ssspc.org/nrfs/
8) 平木　敬，稲葉真理，玉造潤史，来栖竜太郎，生田裕吉，古賀久志，陣崎　明：超高速ネットワーク用データ共有システム：データレゼボアールの性能評価，SWoPP2002，信学技法，Vol.102, No.276, pp.29–34 (2002).
9) http://www.platypus.net/
10) http://www.lustre.org
11) 高橋直也，黒須康雄：キャッシュメモリと共有メモリをもつディスクアレーの高速化手法，電子情報通信学会論文誌 D-I，Vol.J86-D-I, No.6, pp.375–388 (2003).

(平成 15 年 5 月 9 日受付)
(平成 15 年 8 月 29 日採録)

大江　和一（正会員）

1988 年九州大学工学部情報工学科卒業．同年富士通株式会社に入社．現在，株式会社富士通研究所にてストレージシステムの研究開発に従事．

渡辺　高志

2000 年早稲田大学大学院理工学研究科情報専攻（修士）修了．同年株式会社富士通研究所に入社．現在，株式会社富士通研究所にてストレージシステムの研究開発に従事．

西川　克彦

1982 年東京大学工学部電子工学科卒業．同年株式会社富士通研究所に入社．以来図面認識の研究，ビデオサーバの研究開発等を経て，現在サーバ・ストレージシステムの研究開発に従事．映像情報メディア学会会員．