高遅延広帯域ネットワーク環境下における iSCSI プロトコルを用いたシーケンシャルストレージアクセスの性能評価ならびに その性能向上手法に関する考察

山口　実靖† 　小口　正人†† 　喜連川　優†

† 東京大学生産技術研究所　〒 153-8505 目黒区駒場 4-6-1
†† 中央大学研究開発機構　〒 162-8473 東京都新宿区市谷本村町 42-8
E-mail: †{sane,kitsure}@tkl.iis.u-tokyo.ac.jp, ††oguchi@computer.org

あらまし　現在, 安価でかつ容易な SAN 構築手法として iSCSI が注目されている. 本稿では, 高遅延広帯域ネットワーク環境における iSCSI プロトコルを用いたシーケンシャルアクセスの性能 (スループット) について述べる. まず, 高遅延広帯域環境における iSCSI プロトコルによるシーケンシャルアクセスの性能評価を行う. 結果, 既存の OS システムなどでは数 ms 秒程度の遅延環境であっても性能が著しく劣化することが分かった. 次に, iSCSI プロトコルの動作を説明し遅延の増加により性能が著しく劣化する原因がシーケンシャルアクセスのブロック単位の小ささにあることを述べる. 最後に, 大きなブロック単位によるシーケンシャルアクセスの性能評価を行う. これにより, 遅延の増加によるスループットの劣化を 大幅におさえられる (最大 10 倍以上の向上) ことが確認できた.

キーワード　ストレージシステム, ネットワークストレージ, iSCSI

Performance Evaluation of Sequential Storage Access using iSCSI Protocol in Long-delayed High throughput Network

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

† Institute of Industrial Science, University of Tokyo　4-6-1 KOMABA MEGURO-KU, TOKYO 153-8505, JAPAN
†† Research and Development Initiative, Chuo University　42-8 Ichigaya Honmura-cho, Shinjuku-ku Tokyo, 162-8473, Japan
E-mail: †{sane,kitsure}@tkl.iis.u-tokyo.ac.jp, ††oguchi@computer.org

Abstract　The iSCSI protocol is becoming increasingly important as a convenient and cost-effective means of constructing Storage Area Networks (SANs). This paper details a comprehensive overview of performance issues pertaining to the iSCSI protocol with specific focus on latency. In particular, we investigate the performance of iSCSI for sequential accesses in high-latency high-throughput networks. Our performance evaluation demonstrates that the performance of iSCSI degrades with increase in network latency. Moreover, we observe behaviors pertinent to sequential accesses and find evidence of the degradation.

Key words　Storage System, Network Storage, iSCSI

1. はじめに

超大容量のデータを高速に処理するためのシステムとして, SAN(Storage Area Network. 大量のストレージを扱うためのストレージ専用の高速ネットワーク) が注目を集めており, その実績は高い評価を得ている. SAN を導入することにより, DAS(Direct Attached Storage. サーバにローカル配置するストレージ) などのみを使用する場合と比べて, その管理コストが大幅に削減できると言われている. しかし, 現在一般に使われている ファイバチャネルを用いて構築する SAN (以下, “FC-SAN” と呼ぶ) は, 接続距離の限界, ハードウェアコストの高さなどの問題も明らかになっており, Ethernet と TCP/IP を用いて構築する安価な SAN (以下, “IP-SAN” と呼ぶ) が次世代 SAN として注目を集めている [1]. IP-SAN のためのプ

ロトコルも, SCSI over IP である **iSCSI** プロトコルなどが IETF [2], [3] により標準化がなされている. しかし, iSCSI を用いる IP-SAN の課題として転送速度の遅さも指摘されている [4]. 我々が iSCSI プロトコルを用いた IP Storage の性能を測定したところ (第 3. 章にて後述), 高遅延環境においてその性能が著しく劣化することが分かった. iSCSI に限らず一般に高遅延ネットワーク環境下ではレスポンスタイム (ここでは “1 個のコマンドが発行されてからその結果を得られるまでの時間” のことを指す) が長くなることは避けられないが, 高遅延かつ広帯域のネットワークであれば連続的なデータ転送時におけるスループットの低下は避けられるはずである.

本稿では, 各種遅延のネットワーク環境下において iSCSI プロトコルを用いてストレージにシーケンシャルリードアクセスを行ったときの転送速度の評価およびその向上手法について述べる. シーケンシャルアクセスは, データのバックアップやデータマイニングなどの応用で用いられ, その性能評価と性能向上はこれらを遅延のあるネットワーク環境下で行う上で重要となる. 特にデータバックアップは遠隔地に対して行うことが少なくないためこれが重要といえる.

本稿は以下のように構成される. 第 2. 章で研究背景として, 安価で使用が容易な IP-SAN および iSCSI への期待が高まっていることを述べる. 第 3. 章では, 各種ネットワーク遅延状況における iSCSI シーケンシャルリードの性能について評価を行い, それが遅延の増加にともない激しく劣化することを示す. 次に, 第 4. 章において iSCSI プロトコルによるシーケンシャルリードの振る舞いを説明し, 遅延の増加に伴いスループットが著しく低下する原因は SCSI のリードブロックサイズの小ささにあることを述べる. そして第 5. 章で, 大きな SCSI ブロックでのリードアクセスの評価実験を行い, ブロックサイズを大きくすることにより高遅延時のスループットを大幅に改善することが可能であることを示す. 最後に第 6. 章においてまとめと今後の課題を述べる.

2. 研究背景

2.1 iSCSI

データウェアハウスや大規模な WWW サイトなどで大容量のデータを扱う場合, ストレージ装置をそれぞれのサーバの周辺装置としてサーバ単位で管理する手法ではその管理に膨大なコストが必要となる. そこで, 大規模なストレージ装置を中心に配置し高速なストレージ専用ネットワークでサーバ群に接続する SAN が普及してきた. SAN を用いてストレージを集約することにより, ストレージの一元管理が可能となり管理コストを削減することが可能となる. しかし現在普及している FC-SAN はファイバチャネル (以下 “FC”) を用いているため, 接続距離の限界, 相互接続性, ハードウェアコストの高さ, FC 管理技術者の少なさ, FC 接続距離の限界などの問題点も明らかになってきている. そこでこれらの問題を解決する手法として Ethernet と TCP/IP を用いて SAN を構築する IP-SAN が注目されるようになった. IP-SAN の利点としてまず, 相互接続性の高さ, 価格の低さ, 運用技術を持った管理者の多さ, 接続距離の長さ, が挙げられる. さらに, Ethernet の性能向上速度の速さ (FC よりも先に 10Gbit/s 対応), システム内ネットワーク (クライアント-サーバ間とサーバ-ストレージ間の両方) の IP によるシームレスな統合なども期待されている.

IP-SAN におけるデータ転送プロトコルとしては, SCSI ベースの iSCSI, FC ベースの FCIP や iFCP などが IETF [2], SNIA [6] などにより標準化が行われており, iSCSI は 2003 年 2 月に IETF により承認された. 新規に IP-SAN を構築する場合は iSCSI が, 既に FC-SAN を構築してある場合は FCIP や iFCP が有用であるといえる. 本稿では, SCSI ベースのプロトコルである iSCSI について述べる. iSCSI は, サーバとストレージを TCP/IP ネットワークで接続し, ネットワーク越しに SCSI プロトコルで通信するためのプロトコルである. iSCSI では SCSI プロトコル (SCSI コマンド, SCSI レスポンス, SCSI データなど) を iSCSI PDU(Protocol Data Unit) の中に格納し, iSCSI PDU を TCP/IP 上で転送する.

2.2 本研究の位置づけ

本稿では, iSCSI の最大の特徴である iSCSI プロトコルと TCP/IP の振る舞いやこれらの関係に注目して考察をする. 以下の考察は, これらの与える影響とその他の要素の影響を明確に分離するために, ストレージデバイスは十分に高速と見なせる環境において行われていおり, 以下の議論はストレージデバイスが十分に高速であっても高遅延環境ではネットワークにより性能が激しく劣化してしまうこととその理由, そしてその解決策を示すものである. 実環境での性能を考察するにはさらにストレージデバイスの振る舞いも考慮する必要がある. 具体的には, 以下の議論において下位レイヤーから得られると仮定している性能が述べられるがこれをストレージデバイスの性能に併せて低く評価する必要がある. ただし, 高遅延ネットワーク環境下における性能はネットワーク I/O 待ちによるアイドル状態の有無が支配的となり実ストレージでの応用の性能はこれに近いと予測する.

2.3 関連研究

文献 [4] において, Ng らは独自の SCSI over IP 実装を用いて 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, そのスループットが遅延時間にほぼ反比例することを指摘している. また, SCSI over IP を用いるにあたって一貫性の問題の無いアプリケーションにおけるネットワークの手前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効果的であると指摘している. 文献 [5] において, Sarkar らは低遅延環境におけるブロックサイズと iSCSI スループットの関係を紹介している. 低遅延環境においては CPU による処理がスループットを制限するため, さらなる高性能を得るためにはハードウェアによる TCP/IP 処理と iSCSI 処理が重要であると主張している. しかし, ネットワークの影響が大きい高遅延環境についての考察はなされていない.

3. iSCSI シーケンシャルアクセスの性能評価

本章では, iSCSI プロトコルを用いてネットワーク越しにストレージにシーケンシャルアクセスするときの性能とネットワー

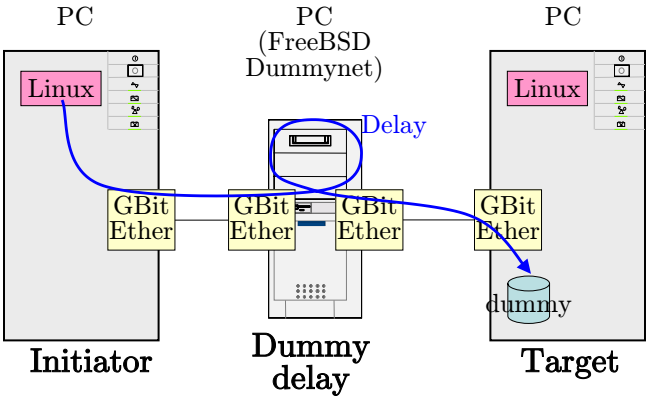

図 1　実 験 環 境

表 1　性能評価実験環境 1

iSCSI Initiator, Target	UNH IOL Draft 18 reference implementation ver. 3
iSCSI MaxRecvDataSegmentLength	16777215 Byte
iSCSI MaxBurstLength	16777215 Byte
iSCSI FirstBurstLength	16777215 Byte
ベンチマーク	Single Thread

クの遅延時間の関係について述べる．純粋な iSCSI プロトコルのみの影響を調べるために, iSCSI Target はメモリモードで動作させた．以下の測定は物理的なディスクアクセスを伴わせず iSCSI プロトコルに基づいてシーケンシャルリードアクセスの手続きを行った場合の性能である．よって, 本結果は iSCSI プロトコルによるネットワークストレージを使用する際の性能の上限を示す物であり, 実際の性能がこれ以下になることを示す．

3.1　実 験 環 境

性能評価実験は以下の環境で行った．図 1 のように, iSCSI Initiator(サーバ) と iSCSI Target(ストレージ) を Gigabit Ethernet で接続して TCP/IP 接続を確立する．Ethernet の接続は, 途中に人工的な遅延装置として FreeBSD Dummynet [7] を挟んでクロスケーブルで接続をした．iSCSI Initiator および Target の実装は, ニューハンプシャー大学 InterOperability Lab [8] が提供する reference implementation を用いた (以下, この iSCSI の実装を UNH と呼ぶ). iSCSI の実装と設定については表 1 に記す．Initiator, Target, Dummynet はすべて PC 上に構築し, Initiator と Target には Linux を, 遅延装置 には FreeBSD をインストールした．これらの PC の詳細は表 2 の通りである．

実験は, Initiator 計算機から Target 計算機に iSCSI で接続を行い, Initiator 計算機の OS(Linux) 上から raw デバイスに対してシーケンシャルリードを行った．iSCSI Target は メモリモードで動作をさせた．これは, 無限に高速なストレージデバイスと見なせる．よって, シーケンシャルリード実験を行ったがディスクへのアクセスは伴っていない．

3.2　実 験 結 果

上記の測定実験を行い図 2 の結果を得た．横軸 (1 Way Delay) は, Initiator 計算機と Target 計算機の間の片道遅延である．“0ms”, “1ms”, “2ms”, “4ms”, “8ms”, “16ms” は遅延装置 (Dummynet) 経由で Initiator 計算機と Target 計算機を接続しそれぞれの値の片道遅延を人工的に作成した．“0ms” は Dummynet を経由するが人工的には遅延を作成しなかった場合である．“0ms” の遅延は片道 140μs 程度である．図中の “iSCSI(UNH)” が, 上記の実験条件における iSCSI を用いてシーケンシャルリードを行ったときのスループットである．図中の “Socket” は,“iSCSI(UNH)” と同条件 (片道遅延, TCP Window Size が等しい) において単純なソケット通信を行ったときのスループットであり, これが実験環境 (iSCSI にとっては下位レイヤー) が提供できる限界の通信速度である (以下, この単純なソケット通信を “素のソケット通信” と呼ぶ)．シーケンシャルリードのブロックサイズは 500KB(注1)であり, TCP Window Size は 1MB である．ブロックサイズとは OS に対して raw デバイスの read システムコールを行う際にアプリケーションが指定したサイズである (このブロックサイズの意味に関しては, 第 4.4 節で述べる)．“Socket” のスループットの上限が約 40[MB/sec] となっているのは, Dummynet の限界である．

測定結果より, iSCSI プロトコルを用いてシーケンシャルリードを行うときのスループットは Target と Initiator 間の遅延の増加に伴い, 大きく低下することが分かった．また, 図中の “iSCSI(UNH)” を “Socket” を比較することにより iSCSI のスループットは, 下位レイヤーの提供できるスループットと比較して著しく低下していることが分かった．この結果に関する考察は次章において行う．

表 2　性能評価実験環境 2：使用計算機

CPU	Pentium4 1.5GHz
Main Memory	128MB
OS	Initiator, Target ： Linux 2.4.18 − 3 Dummnet ： FreeBSD 4.5 − RELEASE
Network Interface	Gigabit Ethernet Card Intel PRO/1000 XT Server Adapter Initiator, Target ： Gbit NIC × 1 Dummynet ： Gbit NIC × 2

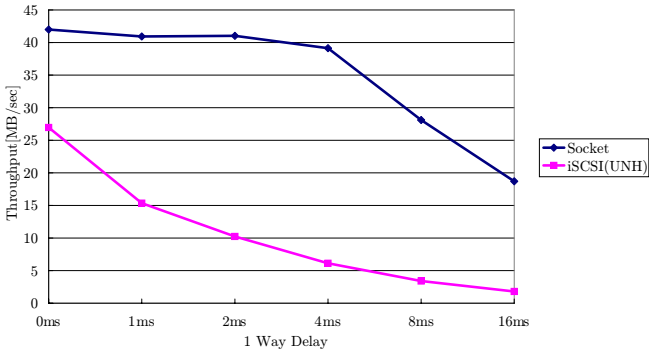

図 2　実験結果：iSCSI シーケンシャルリードスループット
シーケンシャルリードのブロックサイズは, 500KB. TCP Window Size は 1MB である．

(注1)：500KB は 500×1024B

4. iSCSI プロトコルの振る舞いと性能への影響

本章において, iSCSI プロトコルを用いてストレージにシーケンシャルアクセスする際の振る舞いを説明し, それを元に前章のように性能が遅延のある環境で著しく低下する理由を考察する. 遅延の少ない環境において性能の差がある理由は, iSCSI プロトコル処理などのオーバーヘッドによるものであるが, 本稿ではこれについては言及しない.

4.1 前章の測定結果の考察

(iSCSI でなく) 一般のネットワーク通信では, 高遅延かつ広帯域環境であれば大量のデータを片方向に送信するときのスループットを高くすることは可能であるはずである. しかし, 第 3. 章のように Linux OS 上からシーケンシャルリードを行った測定のスループット (図 2 の “iSCSI(UNH)”) は遅延の増加にともない, 大きく低下した. これに対して素のソケット通信である, “Socket” は片道遅延時間 4ms までは遅延時間に依存せず実験システムの提供するスループット (Dummynet の性能もシステムの一部とする) が計測されており, 8ms 以降では実験環境から期待される十分な性能が得られていない. 一般に TCP の受信 Window Size は 2 × 片道遅延時間 × スループット 以上必要であり, スループットは (TCPWindowSize)/(2× 片道遅延時間) 以下に制限されてしまう. TCP Window Size(受信 Window Size) は受信者が送信者に対して通知する値であり, 送信者が (受信者からの)Ack を受け取ることなしに送信することを許される量の上限である. TCP Window Size は TCP ヘッダの中に記載されており, 16bit 割り当てられている. よって, 通常は受信 Window Size は 64KB が上限となるが, TCP Window Scaling (RFC 1323) [9] により 1GB までこれを拡大することができる. さらに送信側の輻輳 Window が激しく変動するため高遅延環境下でのスループットは (TCPWindowSize)/(2× 片道遅延時間) を大きく下回ることがある. 輻輳 Window とはネットワークの輻輳を回避するために設けられている Window であり, 受信側から大きな受信 Window を許可されても送信側が送信を自制する. パケットロスなどの直後はこれが大きく低下しスループットも大きく低下することになる.

4.2 iSCSI のプロトコルスタックと性能の限界

iSCSI プロトコルは SCSI プロトコルを TCP/IP プロトコルで転送するものであるので, SCSI over TCP/IP (over Ethernet) というスタックになる(注2). よって, iSCSI を用いる通信は下位レイヤーである TCP/IP の提供するスループットを超えることができない. 逆に, iSCSI レイヤーでこれを劣化させてしまうことはあるため, この劣化を最小限におさえることが重要となる (第 3.2 節の結果は大きく劣化させてしまうことを示す).

同様に, TCP/IP を用いる通信は, 下位レイヤーである Ethernet の提供する通信速度を超えることができないが, TCP/IP レイヤーでこれを劣化させてしまうことがあるため, この劣化も最小限におさえることが重要である. 前章の実験の例では, 高遅延環境にも関わらず十分に大きくない TCP Window Size で通信を行うと TCP が通信中にアイドル状態時間を作成しスループットを劣化させることになる.

(注2): iSCSI PDU の中に SCSI CDB を包含しているため, 厳密には SCSI over iSCSI とも表現できるが本稿では iSCSI PDU 内に含まれている SCSI CDB も含めて iSCSI レイヤーと呼ぶ

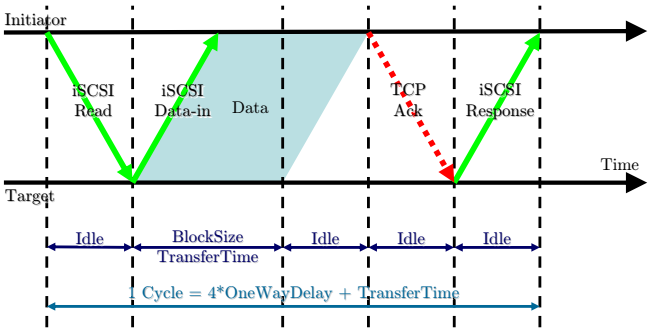

図 3 iSCSI プロトコルによるシーケンシャルリードの振る舞い

4.3 iSCSI プロトコルによるシーケンシャルリードの振る舞い

ブロックサイズが小さいときの iSCSI プロトコルによるシーケンシャルリードの振る舞いを以下に示し, 性能の低下の理由を考察する. 一般にシーケンシャルリードを行うアプリケーションはファイルに対して繰り返し大きなブロックサイズの read システムコールを発行してストレージにアクセスを行う. 前章の計測実験では, raw デバイスに対してベンチマークアプリケーションから 500KB の read システムコールを繰り返し発行した. アプリケーションから iSCSI デバイスに対する read システムコールが発行されると, 図 3 の様な振る舞いが繰り返される. すなわち, 最初に iSCSI Initiator から iSCSI Target に対して iSCSI Read コマンド PDU が送信され, それが iSCSI Target に到着する (図 3 における “iSCSI Read”). 次に iSCSI Target がその iSCSI Read コマンドに対する返信となる iSCSI Data-in PDU を iSCSI Initiator に対して送信し, それが iSCSI Initiator に到着する (図 3 における “iSCSI Data-in” および “Data”). そして, iSCSI Initiator 計算機から iSCSI Target 計算機に対して TCP Ack が送信され, それが iSCSI Target に到着する (図 3 中の “TCP Ack”). ただし, この TCP Ack は Initiator 計算機が受信した TCP パケット (ここでは iSCSI Data-in を含む TCP パケット) に対して TCP/IP 実装が送信するものであり, iSCSI の実装が明示的に送信するものではない. 最後に, iSCSI Target が iSCSI Response PDU を iSCSI Initiator に送信し, それが iSCSI Initiator に受信され (図 3 における “iSCSI Response”), 1 回の Read コマンドが終了する. よって, 以上を繰り返しの単位 (以下, これを iSCSI Seq. Read 単位と記す) としてシーケンシャルリードが行われていく. 図 3 の様に, iSCSI Read 1 回の中に通信相手の返答を待つ時間が 4 個含まれており (図中の “Idle”), ネットワークを使用していない時間が含まれている. また, 実例として, 前章の測定実験において片道遅延時間 8ms の場合のパケットの遷移も図 4, 図 5 に併せて記す. 図 4 は実際の時間におけるパケットの遷移である. 図の “T→I : iSCSI Data-in” は 91 個のパケットで構成されており, “I→T : iSCSI Read”, “I→T :

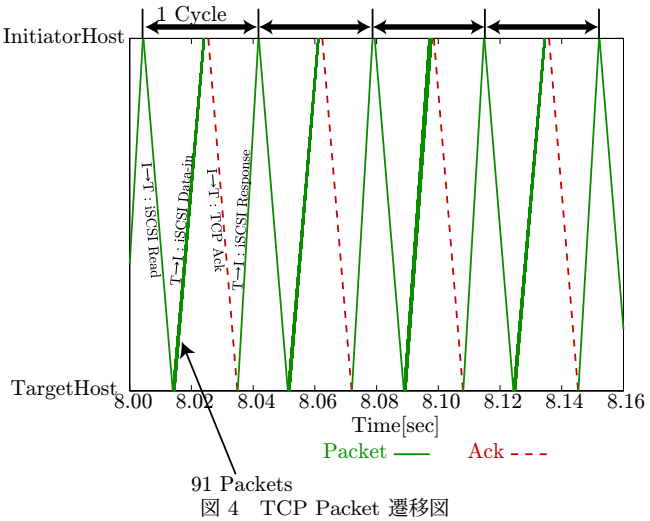

図 4　TCP Packet 遷移図

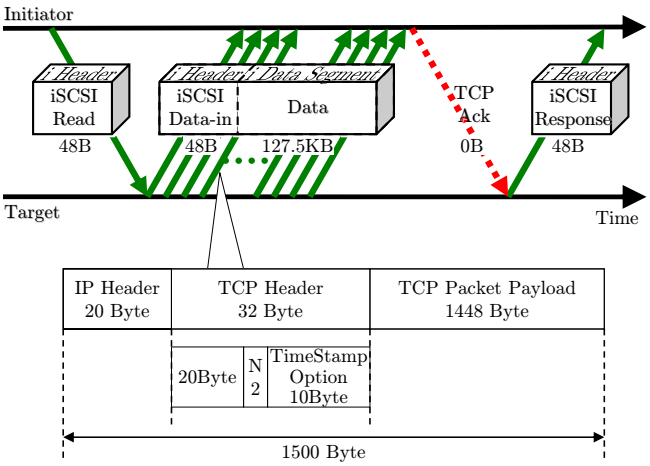

図 5　TCP Packet 遷移模式図

片道遅延時間 8ms, TCP Window Size 1MB 環境において, iSCSI シーケンシャルリードを行ったときのパケットの遷移図. アプリケーションはブロックサイズ 500KB の read を発行した. “I→T” は “Initiator から Target へ”, “T→I” は “Target から Initiator へ” の意.

TCP Ack”, “T→I : iSCSI Response” は 1 個のパケットである. “T→I : iSCSI Data-in” の 91 個の TCP パケットは, ペイロード (TCP/IP ヘッダを除く) の合計が 130608 Byte であり, これは 48Byte の iSCSI ヘッダ と 127.5KB の iSCSI データで構成されている. データサイズが 500KB でなく 127.5KB である理由は, 第 4.4 節で後述する. 図 5 は TCP/IP レイヤーと iSCSI レイヤーを区別して表現した模式図である. iSCSI は SCSI over TCP/IP over Ethernet であるので, SCSI プロトコルが iSCSI PDU に格納されそれが TCP/IP で送信される. 図中の矢印が TCP レイヤーであり, 矢印 1 本が TCP パケット 1 個に相当する. 矢印上に重ねて記した直方体が iSCSI レイヤーであり, 直方体 1 個が iSCSI PDU 1 個に相当する (“i Header” は iSCSI ヘッダ, “i Data Segment” は iSCSI Data Segment を意味する). 直方体の下に添えて記したのが iSCSI レイヤーにおけるデータのバイト数である. TCP/IP ヘッダは含まれていない. 図 5 における “TCP Ack” は, ペイロードが 0Byte であり TCP/IP ヘッダのみで構成される TCP パケットである. Ack 情報は TCP ヘッダに記載されている. 図の様に, iSCSI レイヤーで作成されたデータ (iSCSI PDU) が TCP/IP 実装に渡され, TCP/IP の実装がこれを TCP パケットに分割し 各パケットに TCP/IP ヘッダを付加し 下位レイヤーの Ethernet に送信する. 本実験の “iSCSI Data-in” の例では以下の手順でパケットが作成される. ①Target ストレージが 127.5KB の転送データを作成する. ②iSCSI ドライバが 転送データに iSCSI ヘッダ (48Byte) を付加し iSCSI PDU(127.5KB+48B) を作成する. ③TCP/IP 実装が iSCSI PDU を MSS(Maximum Segment Size : ヘッダを除いたペイロードの大きさの最大値) ごとに分割し, 各パケット に TCP/IP ヘッダ を付加し TCP/IP パケットを作成する. この例では下位レイヤーが Ethernet であるので MTU(Maximum Transmission Unit. 1 個のパケットの最大サイズ) が 1500 バイトであり, TCP/IP ヘッダが 52 バイトなので, MSS は 1448 バイトとなる. 52 バイトの TCP/IP ヘッダは 20 バイトの IP ヘッダ, 20 バイトの TCP ヘッダ, 12 バイトの TCP ヘッダオプション (Nop, Nop, Time Stamp Option) から構成される. ④Ethernet が TCP/IP パケット (各 1500 バイト) に 14 バイトの Ethernet ヘッダおよび 4 バイトのトレーラを付加し, 転送する.

iSCSI Seq. Read 単位で転送されるデータ量は, 単位の最初に Initiator が発行する iSCSI Read コマンド PDU で要求されているブロックサイズである. 1 単位に要する時間は同図より

$$\begin{matrix}1\text{ サイクルに}\\ \text{要する時間}\end{matrix} = 4 \times \text{片道遅延時間} + \text{データ転送時間} \quad (1)$$

$$\text{データ転送時間} = \frac{\text{要求データ}}{\text{下位レイヤーのスループット}} \quad (2)$$

となる. “下位レイヤーのスループット” とは iSCSI レイヤーにとっての下位レイヤーの提供するスループットのことであり, 素のソケット通信がこれにあたる. 以上より, iSCSI プロトコルによるシーケンシャルリードのスループットは

$$\text{スループット} = \frac{\text{ブロックサイズ}}{4 \times \text{片道遅延} + \frac{\text{ブロックサイズ}}{\text{下位レイヤースループット}}} \quad (3)$$

とモデル化することができる. また下位レイヤーである, 素のソケット通信のスループットは (TCP Window Size)/(2× 片道遅延時間以下に制限される. 以上をまとめると, iSCSI でのスループットは以下の制限を受けることになる.

TCP/IP による制限

$$\text{素のソケット通信スループット} \leqq \frac{\text{TCPWindowSize}}{2 \times \text{片道遅延時間}} \quad (4)$$

iSCSI プロトコルによる制限

$$\text{iSCSI スループット} = \frac{\text{ブロックサイズ}}{4 \times \text{片道遅延} + \frac{\text{ブロックサイズ}}{\text{素のソケット通信スループット}}} \quad (5)$$

第 3.2 節の例では iSCSI(UNH) 実測値とモデルの差は, 遅延 1ms において 9% , 2ms において 8% , 4ms において 4% , 8ms において 1% , 16ms において 4% である. ただし, 第 4.4 節において後述する理由により平均ブロックサイズは 125KB としとした.

このことから, iSCSI スループットは遅延時間に依存し遅延

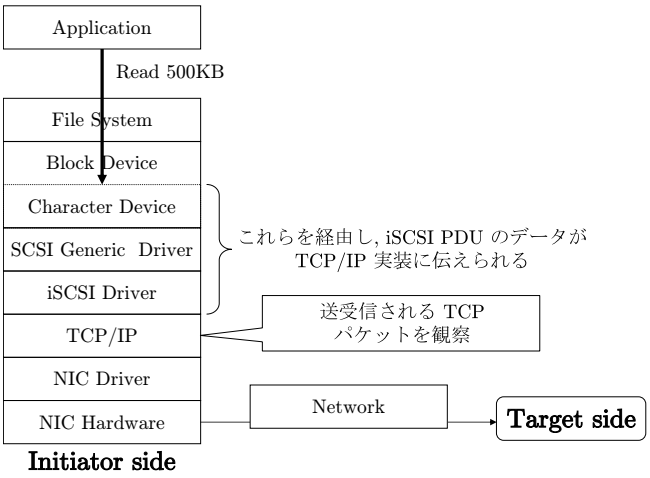

図 6　実験環境の階層構造

の増加に伴いスループットが低下することがわかり, 遅延時間が既知である高遅延環境では, **広い TCP Window Size**, **大きな iSCSI ブロックサイズ**を用いることが重要であると考えられる.

4.4　性能の低下の理由

次に, 実環境(注3)で iSCSI 用いる場合の問題について述べ, 第 3. 章の測定において遅延の増加に伴いスループットが著しく低下した理由を説明する. 第 3. 章の測定は, Linux OS から iSCSI プロトコルを用いてリモートストレージに接続し, その raw デバイスに対してベンチマークアプリケーションから read システムコールを発行することにより行っている. OS やドライバ等の実装に強く依存することであるが, 一般にアプリケーションから read システムコールを発行すると, それが キャラクタデバイス, SCSI ドライバ, iSCSI Initiator ドライバなどを経由して, TCP/IP 実装に対して送信すべきデータ (すなわち, Read コマンドを含む iSCSI PDU) が伝えられることになる. これらを経由した結果, アプリケーションの意図した通りに伝えられるとは限らない. 本実験環境 (第 3.1 節参照) の例においては, 図 6 の様な階層構造になっており, 実際に Target に送信されている iSCSI PDU を観察すると アプリケーションから raw デバイスに対して 500KB の read コマンドを発行した結果, 3 個の “127.5KB(注4) Read の iSCSI PDU” と 1 個の “117.5KB(注5) Read の iSCSI PDU” が作成され 1PDU ずつ Target に対して送信される (すなわちアプリケーションが OS に対して発行したシステムコールのブロックサイズと実際に Target に送られる iSCSI PDU に記載されている Read コマンドのブロックサイズが異なる) . 第 4.3 節で述べたように iSCSI Seq. Read 単位にはアイドル状態が含まれることから, このようにアプリケーションが OS に対して発行した read システムコールのブロックをより小さいサイズのブロックに分割することはスループットを大きく低下させることになる.

本測定の例では iSCSI Seq. Read 単位内におけるアイドル時間 (ネットワークを使用していない) 時間の比率は表 3 の通りである. 表の様に, 高遅延環境下ではアイドル時間が多くの割合を占めておりこれがスループット低下の主たる理由であることが分かる.

表 3　アイドル時間比率

片道遅延時間	1ms	2ms	4ms	8ms	16ms
アイドル時間比率	0.57	0.73	0.84	0.88	0.91

以上のように, 高遅延環境で iSCSI プロトコルを用いる際に 小さいブロックサイズの iSCSI Read コマンドを多数発行してシーケンシャルリードを行うことはスループットを著しく低下させることになり, 第 3. 章の測定において低いスループットが計測された理由はブロックサイズの小ささ (約 125KB) にあることが確認された. 一般にブロックサイズを小さくすることにより性能が下がることがあるが, ネットワークを経由する iSCSI ではその影響が非常に大きくなってしまうと言える.

5.　性能向上手法

本章では, 前章で述べた遅延による性能低下の理由を考慮し遅延隠蔽の方法を述べ, その評価を行う.

5.1　遅延隠蔽手法

前章の考察から高遅延環境においても iSCSI シーケンシャルリードのスループットを低下させないためには, 以下の 2 点が重要であると言える.

（1） TCP Window Size を大きくし, TCP/IP 層におけるスループットの低下を減らす.

（2） iSCSI Read コマンドの Block サイズを大きくし, iSCSI プロトコルによるスループットの低下を減らす.

まず “(1)TCP Window Size を大きくする” であるが, iSCSI プロトコルは TCP/IP 上のプロトコルであるためこの性能を高く保つことが重要である. 次に, “(2)iSCSI Block サイズを大きくする” であるが, 前述の図 3 および表 3 の様に iSCSI Seq. Read 単位には実際にデータを転送している時間とアイドル時間が含まれており高遅延環境下ではアイドル時間がこの多くを占めていることがスループット低下の最大の原因であるため, “(2)” によりアイドル時間の比率を相対的に低下させて iSCSI のスループットを下位レイヤーのスループットに近づけることが最重要である (これは, “最大” ブロックサイズを大きくするということであり “最小” アクセス単位には影響しない. 通常最小アクセス単位は 512 バイトである). また, これまでに言及されていないが iSCSI プロトコルにも MaxBurstLength(注6)等のパラメータが定められておりこれらも同様に十分に大きな値とする必要がある.

5.2　評 価 実 験

前節で述べた手法により高遅延環境における iSCSI スループットをどの程度改善可能であるかを評価するために評価実験を行った. 第 4.4 節で述べた理由により実験環境ではアプリケーションから大きなブロックサイズの iSCSI Read コマン

(注3)：ただし, ストレージデバイスへの実アクセスは考慮していない
(注4)：これは SCSI チャンクサイズの 255 個分に相当
(注5)：これは SCSI チャンクサイズの 235 個分に相当
(注6)：iSCSI Data-in PDU 等のペイロードのサイズの最大値

ド PDU を発行できないため 簡易 iSCSI Initiator を試作し測定を行った. 試作 Initiator は, ドライバではなく通常のユーザ空間のアプリケーションとして動作し, iSCSI Target 計算機と TCP/IP コネクションを確立し iSCSI プロトコルで通信を行う. UNH の実装同様 iSCSI draft 18 [3] に従っている. 試作 Initiator は, iSCSI draft 18 で定められる仕様のうち, iSCSI login 処理, SCSI Read 処理のみ実装されている. 具体的には, Login Request PDU の作成とこれに伴う Login Parameter(MaxRecvDataSegmentLength, MaxBurstLength, FristBurstLength など) の指定, SCSI Read Command PDU の作成と送信処理, SCSI Data-in PDU, SCSI Response PDU の受信処理, Read 処理の繰り返しに伴う シーケンスナンバー (Initiator Task Tag, Command Sequence Number, Status Sequence Number) の処理のみを実装している. ErrorHandling 等のその他の処理は実装されていない.

実験環境は, 第 3.1 節と同様である. 実験は, (1) 素のソケット通信, (2) 通常の iSCSI アクセス, (3) 試作 Initiator によるアクセス の 3 測定を行い, 図中ではそれぞれ (1)“Socket”, (2)“iSCSI(UNH)”, (3)“iSCSI(KI)” と記す. (1) は Initiator 計算機と Target 計算機の間のソケット通信のスループットである. (2) は Initiator 計算機から Target 計算機 に iSCSI 接続を行いその raw デバイスに対して シーケンシャルリードを行う. Initiator 側, Target 側とも UNH の iSCSI 実装を用いている (Target はメモリモード). OS の提供する SCSI ドライバや iSCSI Initiator ドライバ等を経由する. (3) は Initiator 計算機から試作 Initiator を用いて, iSCSI Target に対して iSCSI PDU を送り シーケンシャルリードを行う. Initiator 側が試作 Initiator であり, Target 側が UNH の iSCSI Target 実装である (同様に Target はメモリモード). 試作 Initiator は, ソケットに対して直接 iSCSI PDU を送信するため OS の提供する SCSI ドライバや, iSCSI ドライバは経由しない. 本実験では (2) と (3) の違いとしてブロックサイズの他にドライバを経由するか否かがあるが, ネットワーク遅延と比べてこの差は十分に小さい. 参考のため, 同ブロックサイズにおける (2) と (3) の比較を付録 1. において述べる.

実験結果を示す. まず, 第 3. 章と同様の実験を試作 Initiator を用いてブロックサイズ 125KB, 1MB, 2MB, 4MB で行ったものを図 7 に示す. また, これを iSCSI(UNH) と比較して比で表したものを図 8 に示す.

“Socket” と “iSCSI(UNH)” は, 第 3.2 節のものと同じデータであり, “iSCSI(UNH)” のブロックサイズはアプリケーションからの指定が 500KB, 実際に送信される iSCSI PDU の Read コマンドにおいて平均 125KB である. 図 8 の “Ratio” は iSCSI(UNH) と iSCSI(KI) のスループットの比 iSCSI(UNH)/iSCSI(KI) である. 図中の iSCSI(UNH) と iSCSI(KI) の比較によりブロックサイズを大きくすることにより高遅延環境におけるスループットが大幅に改善されていることが分かる. 遅延の増加に伴うスループットの劣化および素のソケット通信に対するスループットの劣化は大幅に改善されているが, まだ劣化が確認されるのは, ブロックサイズを大きくすることにより iSCSI Seq. Read 単位内のアイドル率を相対的に減少させることが可能であるが遅延時間の影響は存在するからである.

次に, ブロックサイズの変化とスループットの変化の関係を図 9 に示す. 図から, iSCSI(UNH) ではアプリケーションから大きなブロックで read システムコールを発行しても 127.5KB 毎に細分化されるためスループットの向上が起きないのに対し, 試作

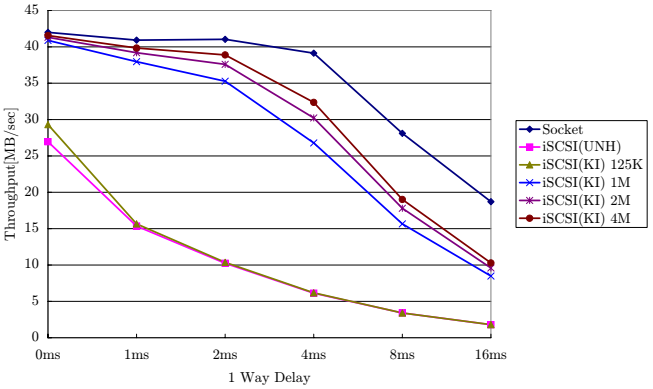

図 7　実験結果：試作 Initiator による iSCSI シーケンシャルリードスループット

UNH Initiator のブロックサイズはアプリケーションからの指定は 500KB であるが, 実際は平均 125KB のブロックの iSCSI PDU が送信されている. 試作 Initiator による シーケンシャルリードのブロックサイズは 125KB, 1MB, 2MB, 4MB. TCP Window Size は 1MB である.

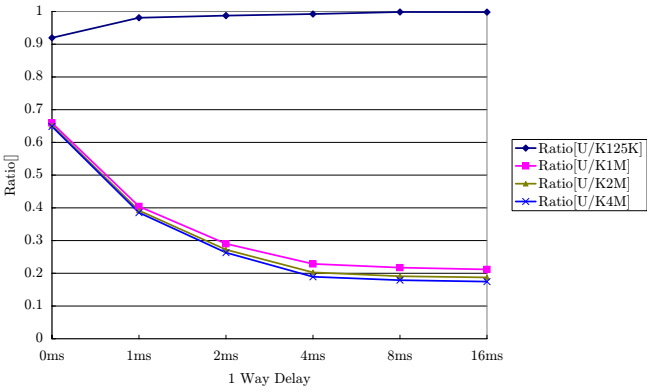

図 8　実験結果：スループット比 [UNH/試作 Initiator]

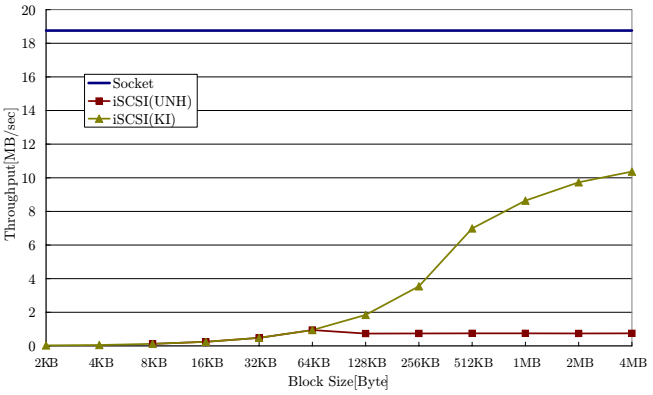

図 9　ブロックサイズとスループットの関係

片道遅延時間 16ms, TCP Window Size 1MB 環境において, 各種ブロックサイズでシーケンシャルリードを行ったときのスループット.

Initiator ではブロックのサイズを増加させることによりシーケンシャルリードのスループットが単調に増加していることが分かる. この例では iSCSI(UNH) は最高で 0.941[MB/sec] のスループットであるのに対し, iSCSI(KI) では最高で 10.4[MB/sec] となり, ブロックの細分化がスループットを著しく劣化させていることと, これを避けることによりスループットは大きく向上 (本実験の例では 10 倍以上) させることが可能であることが確認された. ただし, 片道遅延時間 16ms において 10.4[MB/sec] のスループットは下位レイヤーにより与えられるスループットの半分強に減少していることを意味し, さらなる工夫も必要と言える.

参考として, 付録 2. に遅延 0ms, Window Size 1MB における測定結果を記す.

6. おわりに

本稿では高遅延広帯域ネットワーク環境下で iSCSI プロトコルを用いてシーケンシャルアクセスを行うときの性能について述べた. まず, 実験により一般的な OS システム上で iSCSI プロトコルを用いるとネットワーク遅延の増加に伴いシーケンシャルアクセスのスループットが著しく低下することを示した. つぎに, iSCSI プロトコルの振る舞いを説明しスループット低下の原因が Read ブロックサイズの小ささにあることを述べた. そして, 試作 iSCSI Initiator を用いて大きなブロック単位でのシーケンシャルアクセスの性能を評価し, スループットの劣化を大幅に抑えられることを示した.

遅延の増加によるスループットの著しい低下は数 ms 程度から観察されており, これの回避は iSCSI を実用する上で重要になると考える. 特に, 広域 SAN などではこれが重要である. 本稿で示した実験例の範囲でも片道遅延 4ms 程度までは下位レイヤーの提供する速度とほぼ等しい速度が得られており, 片道遅延 16ms の状況において 10[MB/sec] 程度のスループットは確認されている. 各ネットワークインフラの状況に依存するが提供されているネットワークの性能が十分速いとき, 国内程度の距離であれば iSCSI プロトコルを用いて十分なスループットが確保できると言える.

今後は, 実ストレージを用いての評価, リードのみならずシーケンシャルライトアクセスの評価やその性能向上, などを行っていく.

文　献

[1] 喜連川優, “ストレージネットワーキング”, オーム社出版局, 2002

[2] IETF : http://www.ietf.org/

[3] IETF IPS, http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-19.txt http://www.ietf.org/html.charters/ips-charter.html

[4] Wee Teck Ng, Bruce Hilly Elizabeth Shriver, Eran Gabber, Banu Ozden, “Obtaining High Performance for Storage Outsourcing”, *Proc. FAST 2002, USENIX Conference on File and Storage Technologies*, January 28-29, 2002, pp. 145-158

[5] Prasenjit Sarkar and Kaladhar Voruganti, “IP Storage: The Challenge Ahead”, *Proc. of Tenth NASA Goddard Conference on Mass Storage Systems and Technologies*, April 2002

[6] SNIA : http://www.snia.org/

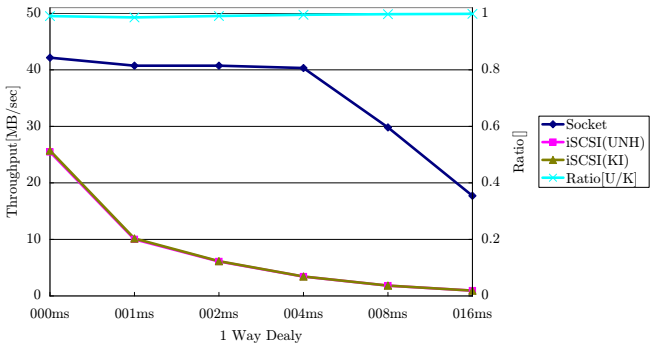

図 A·1　同ブロックサイズにおける iSCSI(UNH) と iSCSI(KI)

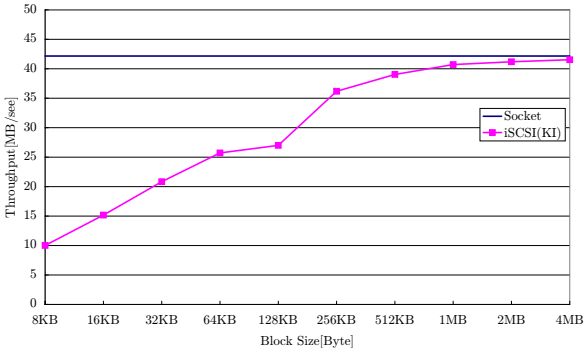

図 A·2　ブロックサイズとスループットの関係 (遅延 0ms, Window Size 1MB)

[7] L. Rizzo, ”dummynet”, http://info.iet.unipi.it/ luigi/ip_dummynet/

[8] The University of New Hampshire’s InterOperability Lab http://www.iol.unh.edu/consortiums/iscsi/iscsi_linux.html

[9] RFC 1323 TCP Extensions for High Performance http://www.ietf.org/rfc/rfc1323.txt

付　録

1. 同ブロックサイズにおける比較

図 A·1 に同ブロックサイズ (ともに 64KB) における iSCSI(UNH) と iSCSI(KI) の比較例を示す (Window Size はともに 1MB). ブロックサイズが同じであれば両スループットはほぼ同じ値を示し, SCSI ドライバ等を経由するか否かの違いはネットワーク遅延の影響と比べて十分に小さいことが確認される (第 5.2 節参照).

2. 低遅延環境におけるスループット

遅延 0ms, Window Size 1MB における素のソケット通信と試作 Initiator のスループットを図 A·2 に記す. 第 3.2 節で前述の通り, “0ms” の遅延は片道 140μs 程度である. 低遅延環境であれば, 試作 Initiator は下位レイヤーの提供するスループットとほぼ等しいスループットを得ることが可能である.