2005－DBS－137（II）（76）
2005／7／14

iSCSIネットワークストレージにおける ファイルアクセス性能に関する考察

山口 実靖† 小口 正人‡ 喜連川 優†

†東京大学 生産技術研究所
‡お茶の水女子大学理学部情報科学科

要 旨

FC (Fibre Channel) を用いる現世代の SAN の欠点を克服する SAN として, TCP/IP, Ethernet, iSCSI を用いる IP-SAN が期待を集めている. IP-SAN は FC を用いる SAN より性能が劣るとの欠点も指摘されているが, そのプロトコルスタックが複雑な多段構成となっており振る舞いの把握が困難であり, 性能向上の実現が容易でない. そこで我々は IP-SAN の全層を統合的に観察可能である IP-SAN トレースシステムを提案し, ファイルシステム等を扱えないながら簡易実装を作成し, その有効性を示した. 本稿では, さらにファイルシステムの振る舞いの観察もできる解析システムを提案し, その実装の紹介を行う. そして, iSCSI ストレージ上のファイルシステムにおけるファイルアクセス性能を計測し紹介する. 計測の結果, iSCSI ネットワークストレージにおいてはジャーナリングの有無が性能に対して非常に大きな影響を与えることや, ジャーナリングを行わない場合はネットワーク遅延時間やファイルへのデータ書き込みの負荷はファイルシステム実装により効果的に隠蔽され性能への影響が軽減されていることや, ジャーナル使用時はファイルへのデータ書き込み負荷が性能に大きな影響与えることが分かった.

File Access Performance on iSCSI Network Storage

Saneyaus Yamaguchi† Masato Oguchi‡ Masaru Kitsuregawa†

†Institute of Industrial Science, University of Tokyo
‡Department of Information Sciences, Faculty of Science, Ochanomizu University

Abstract

IP-SAN and iSCSI are expected to remedy the problems of Fibre Channel-based SAN. iSCSI has a structure of multilayer protocols. A typical configuration of the protocols to realize this system is as follows: SCSI over iSCSI over TCP/IP over Ethernet. Thus, in order to improve the performance of the system, it is necessary to precisely analyze the complicated behavior of each layer. We proposed an IP-SAN analysis tool that monitors each of these layers, and constructed the first implementation which could not support file systems but support only raw devices. In this paper, we propose an integrated trace system which can also monitor behaviors of file systems, and introduce the implementation of the proposed system. In addition, we present the measured file access performances on the iSCSI network storage. It is obtained that network latency and file sizes do not have very large impact on file access performances in the case of not using the journaling system. On the other hand, the sizes of files have large impact on the performances in the case of using the journaling system.

1 はじめに

ストレージ技術の進歩により計算機システムが扱うストレージ容量が増加し, その管理コストの増大が計算機システムの大きな問題の一つとなっている. この問題に対する解決策として SAN (Storage Area Network) を用いてストレージを集約し管理する手法が提案された. SAN を用いたストレージ集約の効果は高く, すでの多くの企業で採用されている. しかし, FC (Fibre Channel) を用いている現世代の SAN は, FC は接続距離に限界がある, FC の導入コストが高い, FC 技術者の数は多くはない, FC の相互接続性は必ずしも高くない, などの欠点も指摘され, これらの問題を解決する SAN として TCP/IP と Ethernet を用いた IP-SAN が提案された. IP-SAN 用のデータ転送プロトコルとしては, 2003 年 2 月に IETF により承認された iSCSI [1] が標準的なデータ転送プロトコルとして期待を集めている. IP-SAN は, 接続距離に限界がない, 導入コストが低い, IP 技術者の数が多い, 相互接続性が高いなどの特徴があり, FC-SAN の欠点を解決すると期待されている反面, (1) 性能が FC-SAN より劣る, (2) CPU 使用率が高い, などの欠点が指摘されている. よって, これらの IP-SAN の欠点の解決が現ストレージシステムの重要な課題となっている.

本研究では, IP-SAN の性能向上について考察を行う. 我々は IP-SAN の性能向上の実現を困難にしている原因が多数のプロトコルで構成されている IP-SAN システムの複雑さにあると考え, 文献 [2] において IP-SAN システムの振る舞いの把握を容易にする IP-SAN トレースシステムを提案し, ファイルシステムを扱えないながらも試作システムを実装しその有効性を示した. 本稿では, より現実的な応用の考察として iSCSI ストレージ上に構築されたファイルシステムの振る舞いの観察とファイルアクセス性能に関する考察を行う. 特に, ネットワーク遅延とファイルアクセス性能の関係や, ジャーナリング手法の性能への影響に関して調査を行う.

本稿は, 以下の様に構成されている. 第 2 章において提案する IP-SAN トレースシステムとファイルシステムのトレースの紹介を行う. 第 3 章において iSCSI 接続のネットワークディスクにおけるファイルアクセス性能を測定しそれを紹介する. 第 4 章においてファイルアクセス処理の解析を行い, ファイルサイズとネットワーク転送データサイズの関係について考察を行う. 最後に, 第 6 章において本稿のまとめを述べる.

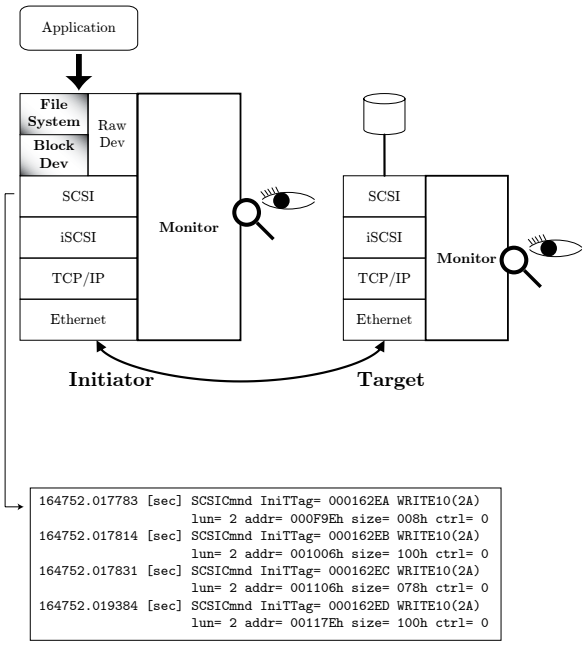

図 1: IP-SAN と トレースシステムの概要

2 ファイルシステムのトレース

本章では, IP-SAN のトレースシステムおよびファイルシステムのトレースの紹介を行う.

iSCSI を用いた IP-SAN は図 1 の様な構成をしている. すなわち, “SCSI over iSCSI over TCP/IP over Ethernet” という複雑なプロトコルスタックで構成されている. そして, 多くの場合はさらにこれら階層の上にブロックデバイスとファイルシステムまたは raw デバイスが配置される.

我々は IP-SAN システム全体を網羅的にトレースできるシステムを構築し俯瞰的なシステムの振る舞いの観察を実現することが性能劣化原因の発見や性能向上の実現に重要であると考え, まず raw デバイスやファイルモード iSCSI ターゲットのみに対応した IP-SAN トレースシステムを実装した [3].

実装は, 図 1 の構成をしている. すなわち, オープンソースである OS 実装と iSCSI ドライバ実装を用いて IP-SAN システムを構築し, これら各層にトレース用のコードを挿入し IP-SAN システムの振る舞いの把握を可能とした.

提案実装は SCSI 層, iSCSI 層, TCP/IP 層の各層の振る舞いの履歴を残すことが可能であり, かつこれらプロトコルの翻訳が可能である. よって, 通常ユーザ空間から把握が困難な IP-SAN システム内部の振る舞いをユーザ空間から観察することが可能である. 図 1 の様に SCSI 層で実際に発行された SCSI 命令の内容が翻訳され, ユーザがその内容を確認することが可能である.

当該実装を実際に raw デバイスに対する並列ショートブロックに適用しその振る舞いを観察したところ性能制限要因が的確に発見され, 試作ながら提案システムが IP-SAN の性能向上の考察に有効な手法であることが確認された [3, 4].

次に, 文献 [5] においてより現実的な応用の性能向上の実現のために ext2 ファイルシステム, ext3 ファイルシステムの振る舞いの把握ができるトレースシステムの実装を行った. 同実装による, ext2 ファイルシステムにおけるファイル作成処理のトレース例を 図 2, 図 3 に示す. ext2 ファイルシステムにおいてファイル作成処理は, 図 2 に示される手順で行われる. すなわち, システムコール “open()” が発行されカーネル内の ext2 実装の sys_open() が呼び出され, その後 filp_open(), open_namei() 等が順次呼び出され, 同図内の一連処理を経て, 最終的にファイルが作成される. これらの一連の処理で通過する各通過点に図 2 の様に A から U までの名を付け, 各通過点を通過した時刻をグラフに示し図 3 が得られる. 図 3 の横軸は図 2 に示される通過点名であり, 縦軸は各通過点の通過時刻である. だたし, 時刻はファイル作成処理開始時刻 (sys_open() の開始時刻である通過点 A の通過時刻) を時刻ゼロとした相対時刻である. 図 3 には, 実験 A におけるファイル作成処理のトレースが 3 例 (Exp. A-1, A-2, A-3) と実験 B におけるファイル作成処理のトレースが 3 例 (Exp. B-1, B-2, B-3) の合計 6 トレースが描かれている. 実験 A は, iSCSI 接続

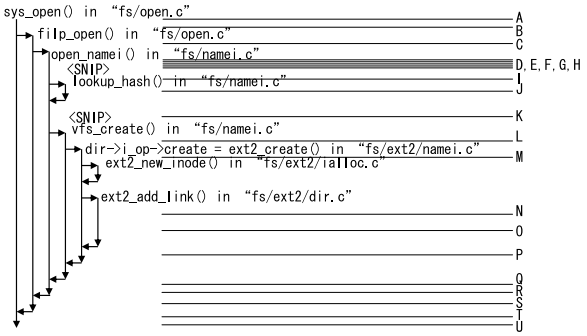

図 2: ext2 ファイルシステムにおけるファイル作成処理

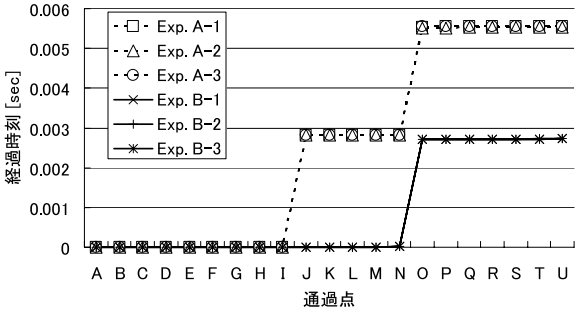

図 3: ファイル作成処理の追跡

ディスク上に構築された ext2 ファイルシステムの単一ディレクトリに 1 バイトのファイルを連続して 100,000 個作成した実験であり, 図 3 には 100,000 作成の中から 3 例が描かれている. 実験 B は, 実験 A の直後に同様に 100,000 ファイルの作成を行った実験である. 図 3 より, 実験 A の全例において通過点 I と J の通過時刻に大きな差があることと, 通過点 N と O の通過時刻に大きな差があることが確認できる. よって, ファイル作成処理に要した時間のほぼ全ての時間は, 通過点 I-J 間の処理と通過点 N-O 間の処理に消費されていることが確認できる. 図 2 より, これらの処理は lookup_hash() の処理と ext2_add_link() 前半部の処理であることが分かる. 両処理はともに ext2 ファイルシステムにおける dentry 検索処理であり, 線形検索として実装されているため非常に多くの時間を要することとなっている. このように, 提案トレースシステムを用いてファイル作成処理を解析することによりファイルシステム, ネットワーク, HDD などから構成さ

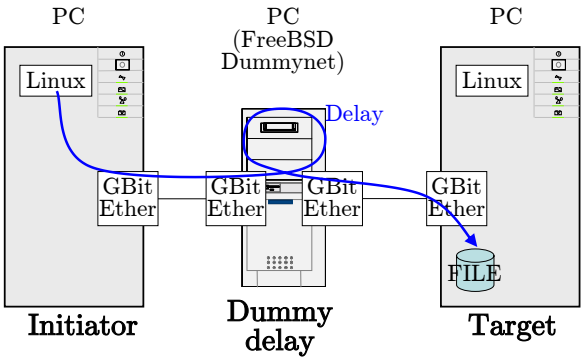

図 4: 実験環境

れている複雑な IP-SAN システムの処理群の中から多くの時間を消費している処理を的確に発見し, 定量的に考察することが可能となる. 同様に, 実験 B においては通過点 N と O の間の処理が消費時間のほとんどを占有していることや, 実験 A と異なり通過点 I と J の間の `lookup_hash()` が消費している時間が非常に少ないことなどが確認できる. これは同じ実験を連続して行ったために, 2 回目の実験 B では ext2 ファイルシステムの dentry キャッシュがヒットし `lookup_hash()` が短時間で終了したためである. この様に提案システムを用いてファイル作成処理の追跡を行うことにより, 1 回目の実験 A と 2 回目の実験 B の性能に再現性がない理由が dentry キャッシュのヒットであることや, その影響がどの程度の時間であるかを定量的に考察することが可能となった.

3 ファイルアクセス性能

本章において, iSCSI 接続ネットワークディスク上にファイルシステムを構築しファイルアクセスを行うときの性能の測定結果を紹介する.

3.1 実験環境

図 4 の様な IP-SAN システムを構築し, 同環境でファイルアクセス性能を測定した. すなわち, iSCSI イニシエータとターゲットを PC を用いて構築し, イニシエータとターゲットの間に人工的なネットワーク遅延装置を配置した. 遅延装置は, FreeBSD Dummynet [6] と PC を用いて構築し, イニシエータ-遅延装置間および遅延装置間-ターゲット間は Gigabit Ethernet クロスケーブルで接続した. イニシエータとターゲットの PC の仕様は表 1 の通りであり, 遅延装置に使用した PC の仕様は表 2 の通りである. 実験のためにイニシエータ-ターゲット間で iSCSI 接続を確立し, そのデバイスを ext3 ファイルシステムでフォーマットした. ファイルシステムの設定としては, (1) “ext3nj” : ext3 ファイルシステムでジャーナル無し, (2) “ext3mj” : ext3 ファイルシステムでメタデータのみジャーナリングする, (3) “ext3dj” : ext3 ファイルシステムでメタデータとデータをジャーナリングする, の 3 種類を用意した. また, ファイルシステムのブロックサイズは 4 [KB] とし, ジャーナル使用時はジャーナル領域を同一パーティション内に 8192 ブロック確保した.

表 1: 性能評価実験環境 2 : 使用計算機

CPU	Pentium 4 2.80GHz
Main Memory	1GB
OS	Linux 2.4.18-3
Network Interface	Gigabit Ethernet Card Intel PRO/1000 XT Server Adapter

表 2: 性能評価実験環境 3 : 使用計算機

CPU	Pentium4 1.5GHz
Main Memory	128MB
OS	FreeBSD 4.5-RELEASE
Network Interface	Gigabit Ethernet Card Intel PRO/1000 XT Server Adapter × 2

iSCSI ドライバの実装としてはニューハンプシャー大学の InterOperability Lab [7] が開発しているオープンソース実装 [8] version 1.5.02 を用いた. iSCSI ターゲットは, ターゲット OS のファイルシステム上

に存在するファイルをディスクイメージとしてイニシエータに提供できる “ファイルモード” を用いた. よって, iSCSI ターゲットへのアクセスは HDD デバイスアクセスではなく, ターゲット OS 上のファイルへのアクセスとなっている.

また, バッファキャッシュのフラッシュの頻度は, `nfract` 値 (bdflush を起床するための汚れたバッファの閾値) が 40%, `ntract_sync` 値 (ブロックされている bdflush を起床するための汚れたバッファの閾値) が 60%, `age_buffer` 値 (汚れたバッファをディスクへ書き込むタイムアウト値) が 30 秒, `interval` 値 (kupdate が動作する間隔) が 5 秒とした.

TCP 受信 Window サイズは 8MB を用いた. これによるスループットの制限は片道遅延時間 4 [ms] においても約 1 [GBytes/sec] であり十分に大きな値と言える.

3.2 ファイル作成実験

前述の IP-SAN 環境において下記の実験を行った. iSCSI 接続のディスク上に上記の 3 通りのファイルシステムを構築し, 同ファイルシステム上に空のディレクトリを作成し, 同ディレクトリ内に固定ファイルサイズのファイルを多数個作成した. ファイルサイズは, 1KB, 10KB, 20KB の 3 種類で行い, 作成ファイル個数は, 100 個から 2000 個まで変化させた. イニシエータ-ターゲット間の片道遅延時間は 0 [ms], 1 [ms], 2 [ms], 4 [ms] の 4 種類に変化させた. 0 [ms] とは遅延装置で人工的には遅延を付加しなかった場合であり, 実際は片道で約 0.13 [ms] 程度の遅延が伴う.

3.3 実験結果

上記の実験を各 130 回ずつ行い, 図 5, 6, 7 の結果を得た. 図の横軸は作成ファイル数を表しており, 縦軸は平均 1 ファイル作成時間である. まずこれら 3 個の図より, ファイル作成時間はジャーナリングの有無に大きく依存していることが確認された. 特に, データのジャーナリングを行うか否かにより性能が大きく変わることが確認された.

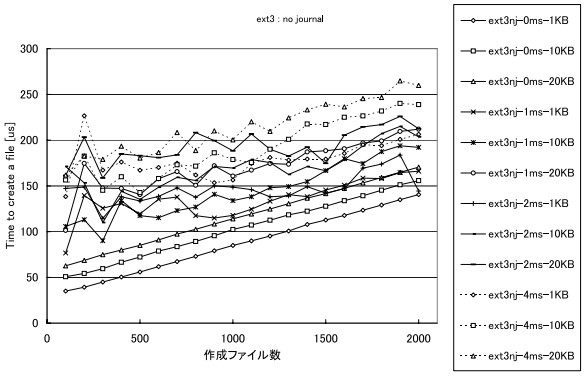

図 5: 実験結果 A: ext3 ジャーナルなし

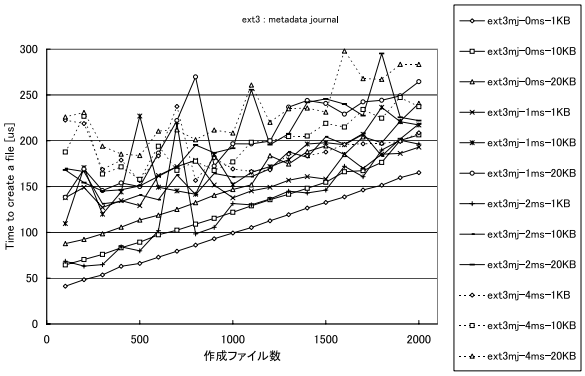

図 6: 実験結果 B: ext3 メタデータジャーナル

次に図 5 より, ジャーナルを用いない ext3 ファイルシステムにおいて平均 1 ファイル作成時間は, 総作成ファイル数, イニシエータ-ターゲット間遅延時間, ファイルサイズの増加に伴い増加していることが確認された. また, 同実験の範囲では総作成ファイル数およびイニシエータ-ターゲット間遅延時間の増加の影響の程度に比べ, ファイルサイズの影響が小さいことが確認された.

そして図 6 より, メタデータのみジャーナリングを行う場合も平均 1 ファイル作成時間は総作成ファイル数, イニシエータ-ターゲット間遅延時間, ファイルサイズの増加に伴い増加していることが確認された. そして, 図 5 同様に同実験の範囲ではファイルサイズの作成時間への影響が他の要素と比較し小さいことが確認された.

これに対して図 7 より, ファイルのデータのジャーナリングを行う場合, ファイルサイズがファイル作成時間に対して最も大きな影響を与え, それと比較

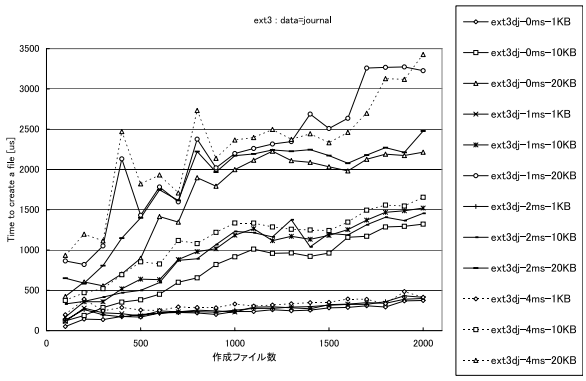

図 7: 実験結果 C: ext3 データ=ジャーナル

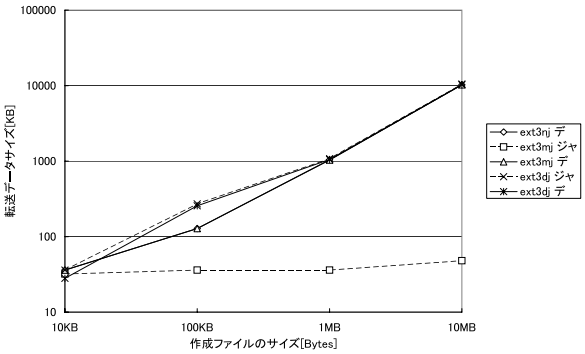

図 8: ファイル作成処理における iSCSI 層での転送データ量

してイニシエータ-ターゲット間の遅延時間の与える影響は小さいことが確認された.

4 ファイルサイズと転送データサイズ

次に, 本章ではユーザがファイルシステムインターフェイスを通してアクセスするファイルのサイズと, ファイルシステムの下位層 (SCSI 層, iSCSI 層など) で実際に送受されるデータサイズの比較を行う. 提案解析システムにより iSCSI 層において送出された iSCSI PDU (Protocol Data Unit) を記録し, 実際にネットワークに送出された SCSI WRITE 命令 PDU および Data-Out PDU を翻訳しファイルシステムの下位層で送受信されているデータサイズを計測し, 図 8 を得た. 横軸がユーザ空間から OS のシステムコールを用いて作成したファイルのサイズであり, 縦軸が iSCSI 層がペイロードとして実際に転送したデータのサイズである[1]. iSCSI 層で転送されたデータには, ファイルの中身の実データの他に, ファイルのメタデータやブロックサイズに起因するパディングデータが含まれるためファイルサイズより大きくなる. また, ファイルサイズと転送サイズの比は図 9 の様になった.

図の ext3nj, ext3mj, ext3dj は順に ジャーナリングなし, メタデータのみジャーナル, データもジャーナルを表しており, “ext3nj デ”, “ext3mj デ”, “ext3dj デ” は各手法においてデータ領域に書き込まれたデータのサイズである. 同様に, “ext3mj ジャ” と “ext3dj ジャ” は各手法においてジャーナル領域に書き込まれたデータのサイズである.

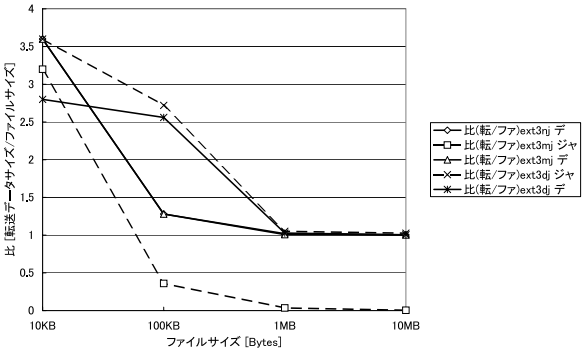

図 9: 作成ファイルサイズと iSCSI 転送データサイズの比

これらの図より “メタデータのみジャーナル” では, ジャーナル領域への書き込みはデータサイズに依存せずに非常に小さく, “データもジャーナル” ではファイルサイズが十分に大きい例ではジャーナル領域への書き込みがデータサイズとほぼ等しく, 結果としてデータもジャーナリングする手法はファイル作成時間がデータのサイズに強く依存することが分かる.

[1]縦軸の値に Ethernet のヘッダとトレーラ, TCP/IP ヘッダ, iSCSI ヘッダの各サイズを加算した値がネットワークケーブルで転送されたデータの総量となる.

5 関連研究

iSCSI を用いた IP-SAN の性能の評価に関する研究としては, 文献 [9, 10, 11, 12, 13, 14, 15] があげられる. 文献 [9] は早期に SCSI over IP の性能評価を行った開拓的な研究である. Sarkar らは文献 [10, 11] において CPU 使用率に着目し iSCSI アクセスの性能についての評価を行い, iSCSI アクセスにおける CPU 使用率の増加の程度や, TOE(TCP Offload Engine) の貢献の限界などを示している. 文献 [13] は, iSCSI, SMB, NFS の性能評価と比較を行っている. 文献 [12] は iSCSI 性能を評価しソフトウェア処理手法は FC-SAN に匹敵する性能を提供できることを示している. 文献 [14] は, IP 接続ストレージのアクセス手法として iSCSI と NFS の比較を行い, NFS に対する iSCSI の優位性等を示している. 文献 [15] は, IPsec や SSL を用いる iSCSI の性能を評価している. 以上の研究は各種状況における iSCSI 性能の評価を行ったものであり, iSCSI の性能を知る上で有用な研究であると言える. しかし, システムの外部から負荷を与え性能を評価したものでありシステム内部の振る舞いについて考察を行ったものではない.

藤田らの文献 [16] は, iSCSI ターゲットの内部の実装手法も考慮して iSCSI 性能の評価を行い, OS のカーネルに変更を加える手法や低レベルインターフェイスを使用する実装手法が性能において優れていることを指摘してる. ターゲット実装に着目し詳細な考察を行っている点において, システム全体の考察を目指す我々の研究と目的が同じでは無いが, ターゲットシステム実装の詳細な考察を行った既存の研究として価値が高いと思われる.

また, ネットワーク監視システムとして多くの実装が既に存在しているが, それらはネットワーク上に配置され転送されるパケットを監視するか計算機上に監視プロセスを起動し資源使用率などの統計を観察するにとどまり, OS のカーネル空間内部の振る舞いの観察や個別の I/O 要求の詳細な振る舞いの解析を行える物ではない. よって, 本研究で提案するシステム内部の振る舞いの把握や個別の要求のトレースの実現はこれらの実装に対しても十分に有用な手法であると考えられる.

6 おわりに

本稿では, iSCSI を用いた IP-SAN の性能に関する考察として, ファイルアクセス性能についての考察を述べた. まず, ext2, ext3 ファイルシステムの実装に観察用コードを挿入しその振る舞いを観察する手法を提案し, 提案システムの実装, 機能, 応用例とその効果を紹介した. 次に, 各種遅延, ファイルサイズ, ファイルシステムのジャーナル手法においてファイル作成時間を計測し, その性能の紹介を行った. そして, 最後に提案システムを用いて, ファイルシステム上のファイルサイズと実際にネットワークで転送されるデータ量の関係を明らかにした.

今後は, ジャーナル領域やデータ領域のバッファがフラッシュされるタイミングと性能の関係, フラッシュ方針と性能の関係を調査し, これらをもとに性能向上手法の考察を行っていく.

参考文献

[1] J. Satran et al. Internet Small Computer Systems Interface (iSCSI). `http://www.ietf.org/rfc/rfc3720.txt` , April 2004.

[2] 山口実靖, 小口正人, and 喜連川優. “iSCSI ストレージアクセスのトレースシステム”. *DBSJ Letters Vol.3 No.3*, 2004.

[3] 喜連川優 山口実靖 小口正人. “iSCSI ストレージアクセスのトレースシステム”. In **夏のワークショップ** *DBWS 2004* **電子情報通信学会技術研究報告データ工学 信学技報** *Vol.104 No.177*, July 2004.

[4] 喜連川優 山口実靖 小口正人. “iSCSI を用いた IP ネットワークストレージシステムのトレース解析”. In **マルチメディア通信と分散処理ワークショップ論文集**, pages 173–178, Dec. 2004.

[5] 山口実靖, 小口正人, and 喜連川優. IP-SAN トレースシステムを用いたストレージアクセス解析. In **電子情報通信学会第** *16* **回データ工学ワークショップ**, March 2005.

[6] L. Rizzo. dummynet. `http://info.iet.unipi.it/~luigi/ip_dummynet/` , 2004.

[7] University of new hampshire interoperability lab. `http://www.iol.unh.edu/` , 2004.

[8] iSCSI reference implementation. `http://www.iol.unh.edu/consortiums/iscsi/downloads.html`, 2004.

[9] Wee Teck Ng, Bruce Hilly Elizabeth Shriver, Eran Gabber, and Banu Ozden. Obtaining High Performance for Storage Outsourcing. In *Proc. FAST 2002, USENIX Conference on File and Storage Technologies*, pages 145–158, January 2002.

[10] Prasenjit Sarkar, Sandeep Uttamchandani, and Kaladhar Voruganti. Storage over IP: When Does Hardware Support help? In *Proc. FAST 2003, USENIX Conference on File and Storage Technologies*, March 2003.

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

[12] Stephen Aiken, Dirk Grunwald, and Andy Pleszkun. A Performance Analysis of the iSCSI Protocol. In *IEEE/NASA MSST2003 Twentieth IEEE/Eleventh NASA Goddard Conference on Mass Storage Systems and Technologies*, April 2003.

[13] Yingping Lu and David H. C. Du. Performance Study of iSCSI-Based Storage Subsystems. *IEEE Communications Magazine*, August 2003.

[14] Peter Radkov, Li Yin, Pawan Goyal, Prasenjit Sarkar, and Prashant Shenoy. A performance Comparison of NFS and iSCSI for IP-Networked Storage. In *Proc. FAST 2004, USENIX Conference on File and Storage Technologies*, March 2004.

[15] Shuang-Yu Tang, Ying-Ping Lu, and David H. C. Du. Performance Study of Software-Based iSCSI Security. In *1st International IEEE Security in Storage Workshop*, December 2002.

[16] 藤田智成 小河原成哲. “iSCSI ターゲットソフトウェアの解析”. In **先進的計算基盤システムシンポジウム** *SACSIS 2004*, May 2004.