D-018

IP ネットワークストレージシステムのトレース解析
Trace Analysis of IP Network Storage System

山口 実靖† **小口 正人**‡ **喜連川 優**†
Saneyasu Yamaguchi Masato Oguchi Masaru Kitsuregawa

1. はじめに

Ethernet と TCP/IP で構築する IP-SAN は, 導入コストの高さなどの FC-SAN の欠点を解決する SAN として大きな期待を集めている [1]. 本稿では iSCSI を用いた IP-SAN の “アクセストレースシステム” を提案する.

iSCSI プロトコルスタックは図 1 の様な多段構成となる. さらに IP-SAN ではサーバ計算機とストレージ機器が協調しながら動作するためこれらの統合的な解析が重要であると考えられる. 本稿ではこれら全層が観察可能であり, 統合的な解析が可能である “IP-SAN トレースシステム” を提案する. そして, それを高遅延環境下における並列ショートブロックアクセスに実際に適用しその有効性を示す.

2. IP-SAN トレースシステム

本稿で提案する “IP-SAN トレースシステム” は, 図 1 の様な構造をしている. オープンソース OS 実装 (Linux 2.4.18) とオープンソース iSCSI 実装 (ニューハンプシャー大学の InterOperabilityLab が配布する iSCSI 実装 ver.1.5.02) を用いて IP-SAN 環境を構築し, 各層にその振る舞いをトレースできるモニタコードを適用する. そして, モニタされた各層の振る舞いを統合的に解析し, アプリケーションによる I/O 要求の発行から HDD デバイスまでの振る舞いの把握を可能とする. 解析結果を可視化することにより図 2 の様な図が得られる. 同図の縦軸は iSCSI ストレージアクセスの各層を表しており, 上から順に①システムコールの発行, ②raw デバイス層, ③SCSI 層, ④iSCSI 層, ⑤TCP/IP 層, ⑥Ethernet によるパケットの転送, ⑦TCP/IP 層, ⑧iSCSI 層, ⑨SCSI 層, ⑩HDD デバイスへのアクセス, を意味している. ①〜⑤がサーバ計算機, ⑦〜⑩がストレージ機器における処理の記録である. 同例では, SCSI 層の上位層として

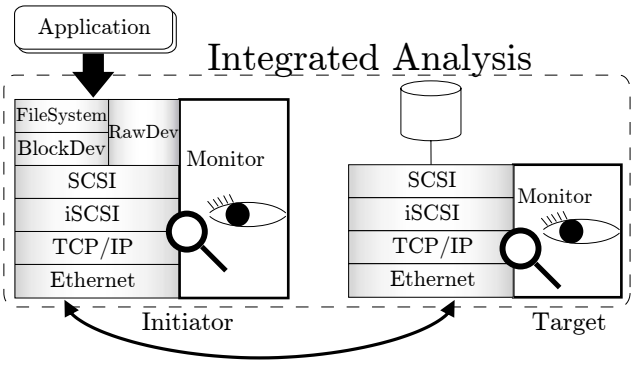

図 1: iSCSI プロトコルスタックと解析システム

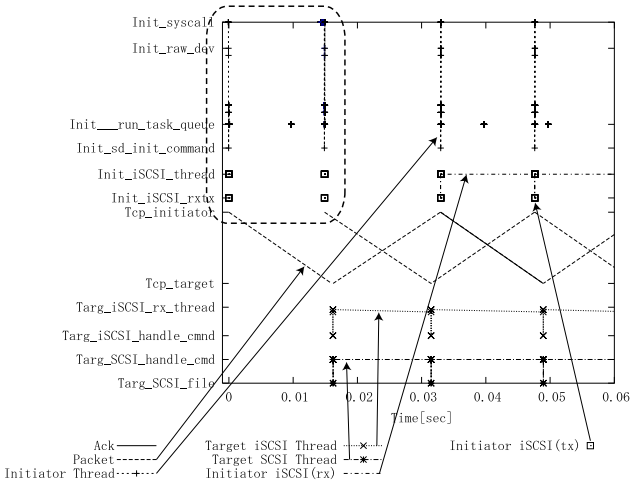

図 2: 並列 iSCSI アクセスのトレース: A

raw デバイスを使用している. また, iSCSI ターゲットとして “File Mode” を使用したため最下層はストレージ機器上のファイルアクセスのトレースとなっている. 横軸は, 各層において各処理が行われた時刻を表している. このように各層の処理を時間軸上に表示することにより I/O 処理の流れを視覚的に把握することが可能となり, 実際に時間を多く消費している処理や, 待ち状態にある処理などを確認するとが可能となる.

3. 提案システムの評価

本章では提案解析システムを実際に高遅延環境下における並列ショートブロック iSCSI リードアクセスに適用し, その有効性を示す. サーバ計算機-ストレージ機器間の片道遅延時間が 16ms の環境下においてベンチマークを複数プロセス同時に動作させ, 全プロセスの合計性能を計測した. 各ベンチマークは iSCSI 接続の raw デバイスに対し 512 バイトのシステムコール `read()` をシーケンシャルに 2048 回ずつ発行する. iSCSI ターゲットは “File Mode” で動作させ, ファイル内容が実メモリ上にキャッシュされている状態で計測を行った. よって I/O 要求は必ずストレージ機器の SCSI 層まで到達し, メモリ上のキャッシュをヒットすることになる. 上記の実験を行い, 図 4 の “can_queue=2(default)” の結果を得た. 同図におけるトランザクション数とは, システムコール数のことである. 同結果より, プロセス数の増加に伴う合計性能の向上は, 並列度 2 において飽和となり, iSCSI プロトコルスタックのいずれかの層において並列度が 2 に制限されていると予想される.

次に, 並列度制限に関する巨視的な解析結果を示す. 図 2 が, 上記実験のプロセス数 3 における iSCSI スト

†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

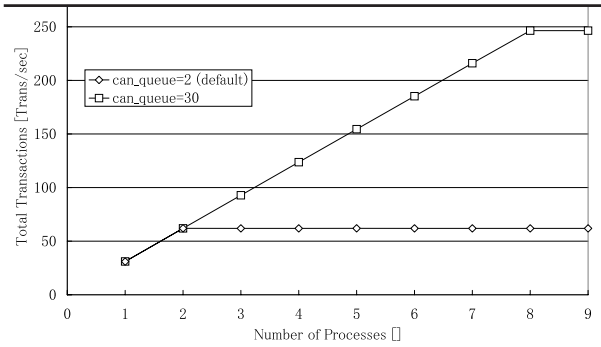

図 3: 並列 I/O の合計性能

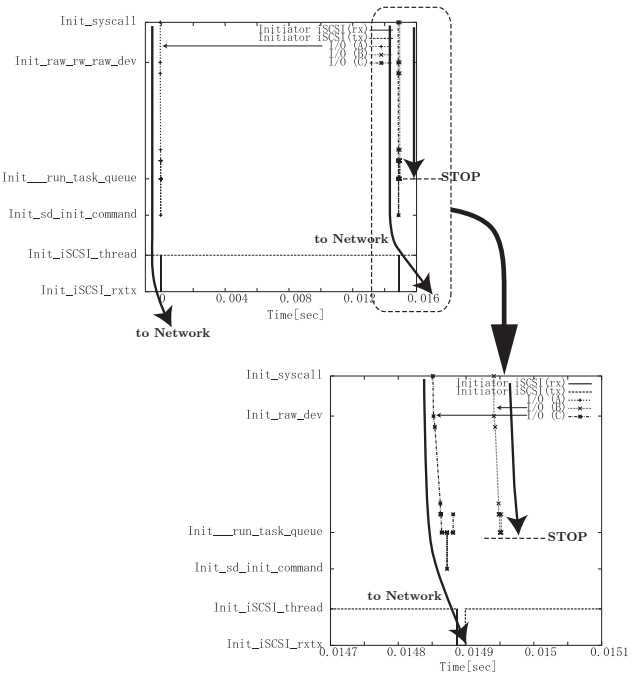

図 4: 並列 iSCSI アクセスのトレース: B

レージアクセの可視化結果である. 同図より各往復時間内にサーバ計算機からストレージ機器に対して 2 個の I/O 要求しか送出されていないことが確認でき, 並列数 2 の制限はサーバ計算機側に存在することが分かる. また, 図 2 の破線部を拡大し表示すると, 図 4 の左上の様になる. 図 4 左上よりシステムコールはストレージ機器からの応答を待つことなしに 1 往復時間 (32ms) 内に 3 個発行されていることや, ベンチマーク “I/O(A)” の要求は時刻 0.000 秒に発行され raw デバイス層, SCSI 層, iSCSI 層を経由し, TCP/IP 層に至りストレージ機器に送出されていることが確認できる. 図 4 左上の破線部を拡大することにより, 同図右下が得られる. 図 4 右下よりベンチマーク “I/O(C)” のシステムコールは時刻 0.01485 秒に発行され, 同様にネットワークに送出されていることが確認できる. これに対し, ベンチマーク “I/O(B)” では, システムコールが時刻 0.01494 秒に発行され直後に raw デバイス層を通過しているが, SCSI 層

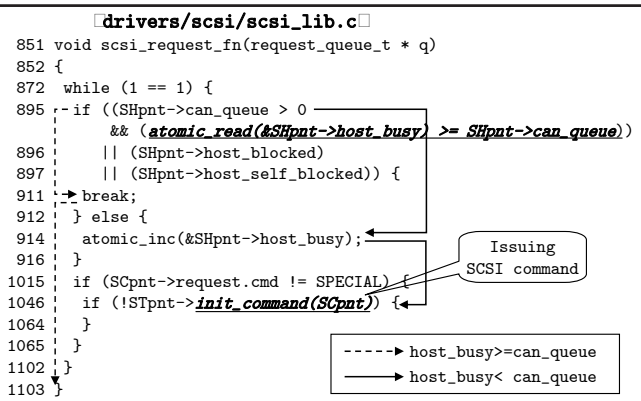

図 5: Linux SCSI 層のトレース

の SCSI 命令の発行に至っておらず, SCSI 命令の同時発行上限が 2 となっていることが確認できる.

次に微視的な解析結果を示す. SCSI 命令が発行される最初の 2 要求と, 発行されない 3 個目の要求のトレースの分岐点は Linux SCSI 層実装における図 5 の部分である. 同実装は現在のアクティブな命令数 “host_busy” と下位層 (iSCSI 層) が同時に受け付け可能である命令数 “can_queue” の比較部である. 最初の 2 要求 (I/O(A),(C)) では host_busy がそれぞれ 0, 1 であり, can_queue が 2 である. よって, “host_busy<can_queue” に示される処理 (914 行目において host_busy をインクリメントし 1046 行目において SCSI 命令を発行する) が記録された. 3 個目の要求 (I/O(B)) では host_busy が 2 であり, “host_busy=>can_queue” に示される処理 (SCSI 命令を発行しない) が記録された. よって, iSCSI 実装の can_queue の値が 2 であることが合計性能制限の理由であると予想される. そこで, can_queue 値を 30 に設定し (初期値は 2 である) 性能を測定し, 図 4 の “can_queue=30” を得た. 同測定では合計性能は並列数 8 までほぼ線形に上昇しており, 同例においてはトレースシステムを適用し合計性能の限界を約 4 倍に向上させることが可能であった.

また同実験において解析システムの適用が性能に与えるオーバーヘッドは 1 並列, 4 並列時において 0.2%未満, それ以外において 0.1%未満となり, 十分に少ないオーバーヘッドで観察が可能であったと言える.

4. おわりに

本稿では, IP-SAN プロトコルスタックの全レイヤーを統合的に解析できるシステムを提案し, その有効性を示した. 今後は, ファイルシステムや実 HDD デバイスを用いたシステムの解析を紹介していく予定である.

参考文献

[1] 喜連川優 山口実靖, 小口正人. “iSCSI 解析システムの構築と高遅延環境におけるシーケンシャルアクセスの性能向上に関する考察”. 電子情報通信学会論文誌 *D-1*, 87, February 2004.