D-036

トレースシステムを用いた IP-SAN におけるファイル操作性能に関する解析

An Analysis for Performance of File Operations on IP-SAN with Trace System

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

1．はじめに

ストレージは定期的なバックアップ作業が必要であり，膨大な管理費用が必要となる．この問題の解決策として SAN(Storage Area Network)を用いてストレージを集約する手法が提案され，近年は iSCSI などを用いる IP-SAN(TCP/IP と Ethernet を用いた SAN)が注目を集めている．IP-SAN は低い導入コスト，制限のない接続距離，高い相互接続性などの利点があると期待されている反面，性能が低い，CPU 使用率が高いという欠点も指摘されており，IP-SAN の性能向上は計算機システムの非常に重要な課題の一つである．しかし，IP-SAN は複雑なプロトコルスタックにより構成されており，これら全てが OS のカーネルの内部で動作するためその振る舞いの把握が非常に困難となっており，結果としてその性能向上の考察も困難となっている．そこで我々はカーネル内部の振る舞いを観察可能である IP-SAN のトレースシステムを提案し，試作システムを用いその評価を行い，試作段階においても高い有効性があることを示した[1]．本稿ではファイルシステム等も含めた実装を行い，その評価を行う．

2．統合トレースシステム

iSCSI を用いた IP-SAN におけるストレージアクセスはイニシエータ側においてファイルシステム，ブロックデバイス，SCSI 層，iSCSI 層，TCP/IP 層，Ethernet 層を経由し，ターゲット側において Ethernet 層，TCP/IP 層，iSCSI 層，SCSI 層を経由して HDD デバイスにアクセスすることとなる．そこでオープンソース実装を用いこれら全層にストレージアクセスのログを記録する機能を追加した．これによりアプリケーションのシステムコール発行から，ネットワークを超え HDD にアクセスするまでの I/O 処理を追跡することが可能となる[1]．

3．評価ファイル作成処理への適用

本章において提案トレースシステムの適用例を示し，その評価を行う．具体的には連続微少ファイル作成実験に対して提案システムを適用し，その性能決定要因の解析を行う．ファイル作成実験は以下の環境において行った．iSCSI イニシエータとターゲットは PC で構築され，同機の CPU は Pentium4 2.8GHz，メインメモリは 1GB，OS は Linux 2.4.18-3，NIC は Intel ServerAdapter PRO/1000 XT となっており，イニシエータとターゲットは Gigabit Ethernet クロスケーブルで接続した．iSCSI ドライバ実装はニューハンプシャー大学が配布する実装を使用した．実験は iSCSI 接続されたデバイスを ext2 ファイルシステムでフォーマットし空のディレクトリに対して 1 バイトのファイルを連続して 100,000 個作成し，その総作成時間を計測した．ファイル名は 10 桁の通し番号＋".txt"の 14 文字(例えば 12345 個目のファイルは"0000012345.txt")とした．ファイル作成はシステムコール"open()"により行った．また，実験はディレクトリ内にファイルが存在しない状態にしてから行われた．この 100,000 個ファイル作成実験を 5 回連続で行い(それぞれを A，B，C，D，E と呼ぶ)，図 1 の結果を得た．縦軸は 100,000 個のファイルの総作成時間を表している．同結果より，①100,000 個のファイルの作成には 450 秒～600 秒程度要すること，②初回(実験 A)のみ処理時間が長いことが確認された．また，実験 A(初回)の 100,000

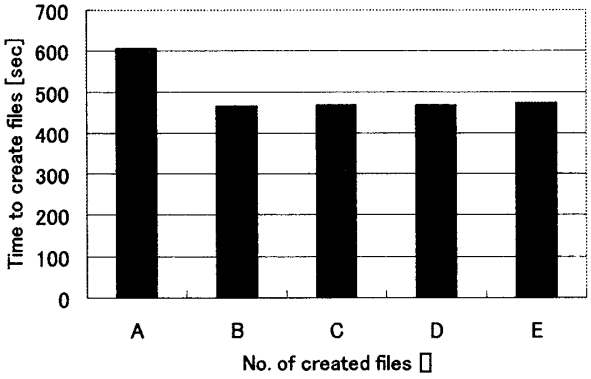

図1: ファイル作成時間

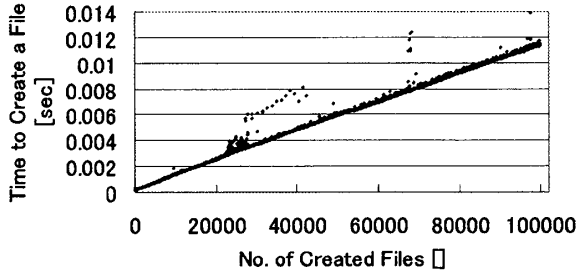

図2: ファイル作成時間の推移(実験 A)

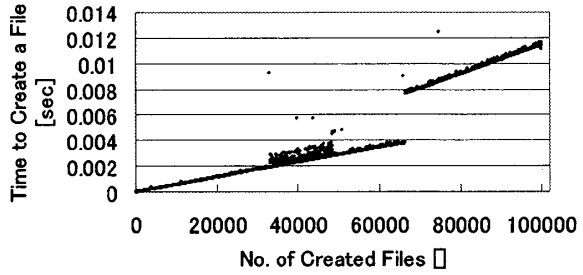

図3: ファイル作成時間の推移(実験 B)

†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

回のファイル作成操作における各作成の処理時間は図 2 の様に，実験 B(2 回目)の各処理時間は図 3 の様になった．横軸は作成したファイルの番号で 1 個目から 100,000 個目までを表している．縦軸は 1 ファイルごとの作成処理の所要時間である．図 2，図 3 より③ファイル作成時間は既存ファイル数の増加に伴い増加する傾向にあり 100,000 個目のファイル作成には 10m 秒以上の時間を要している(これはネットワーク往復時間の 0.16m 秒と比べ十分に大きい)こと，④実験 A と実験 B では前半約 66,000 個のファイル作成時間が異なっており前者が後者の 2 倍程度であること，などが確認された．以上がカーネル空間の外部から計測を行い得られた考察である．

つぎに，提案システムを用いて同実験の解析を行いその性能について考察を行う．ext2 ファイルシステムにおけるファイル作成処理は図 4 の手続きで行われる．図内の左の矢印はカーネル内の関数呼び出しスタックであり，“sys_open()”が呼び出され，各関数の呼び出しを経てファイルが作成される．これら一連の操作の各通過点に A～U と名付けそれらを同図内の右に記した．実験 A と B のファイル作成処理の各通過点の通過時刻は図 5 の様になった．横軸は各通過点名を表し，縦軸は各通過点を通過した時刻である．ただし，“通過時刻”とは各ファイル作成開始時を時刻ゼロとして同時刻からの相対時刻である．図 5 には実験 A における 47624 個目から連続する 3 個のファイル作成(順に Exp. A-1, A-2, A-3)，実験 B における同じく 47624 個目から 3 個のファイル作成(順に Exp. B-1,B-2, B3)のトレースである．

同図よりまず，実験 A においては通過点 I と J の通過時刻に大きな差(2.8m 秒)があること，通過点 N と O の通過時刻に大きな差(2.7m 秒)があることが確認される．すなわち通過点 I と J の間にある処理と N と O の間の処理が大きく時間を消費していることが確認され，この 2 処理が処理時間の殆どを占めており(全処理時間の 99.5%)性能に支配的であると結論付けられる．同様に実験 B においても通過点 N と O の間の処理が大きな時間(2.7m 秒)を消費していることが確認された．図 4 より，通過点 I と J の間の処理は“lookup_hash()”関数による dentry(ext2 ファイルシステムにおける i ノードのディレクトリへの登録)の検索処理である．同様に通過点 N と O の間の処理は“ext2_add_link()”関数内において dentry を検索する処理である．両者とも同ファイルシステム内において線形探索として実装されているため既存のファイル数に比例し，多大な処理時間を要する結果となった．

次に，実験 A と B の性能の違いとして前半約 66,000 個の作成時間に約 2 倍の違いがあることが確認されているが，この原因は通過点 I と J の間の処理(“lookup_hash()”)に要する時間の違いであることが分かる．これは，ext2 ファイルシステム実装における dentry のキャッシュがヒットし検索が省略された場合にこの検索時間が極めて短い時間で終了するからあり，本例では 5 回の実験で使用したファイル名群が同一であるために 2 回目の実験以降は同キャッシュがヒットし処理時間が短くなっている．

以上のように提案トレースシステムを用いることによりファイル作成操作の進行を追跡可能となり，その性能の決定過程を定量的に考察できるようになった．また，実験の初回実行時と次回以降で性能に再現性が無いことが確認さ

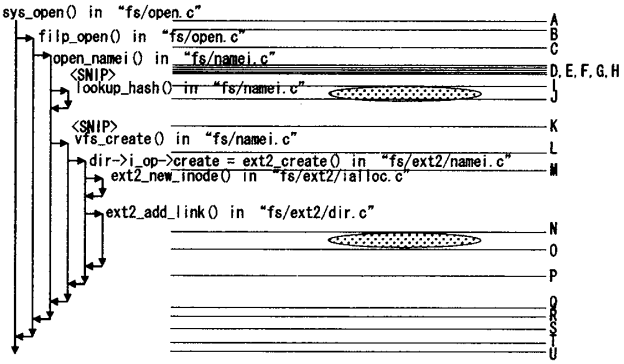

図 4: ext2 ファイルシステムにおけるファイル作成処理

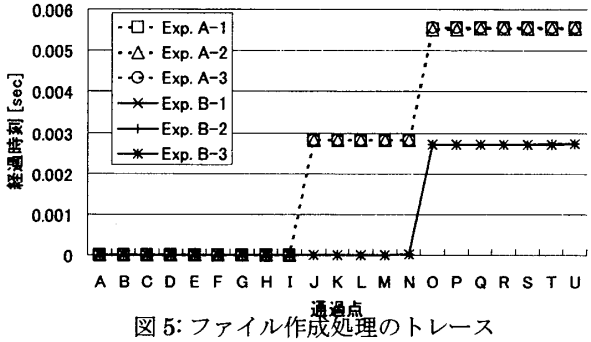

図 5: ファイル作成処理のトレース

れたが，その理由を定量的に解説することが可能となった．具体的には，多段のプロトコルで構成されている IP-SAN システムの各種処理の中で多くの時間を消費しており性能に支配的な処理はファイルシステムにおける dentry 検索処理であることが分かった．ネットワークや HDD デバイスアクセスの処理時間が性能に支配的である例においては容易に性能決定要因を予想することが可能であるが，多段プロトコルスタックで構成され多くの要因が性能決定要因となりうる IP-SAN においては提案システムの様なカーネル内部の振る舞いをイニシエータ側からターゲット側まで追跡可能なツールによる定量的な考察が重要であると言える．

4．おわりに

本稿では iSCSI のトレースシステムを提案し，ファイルシステム等の振る舞いも含め実用的なアプリケーションの統合的な解析を行えるシステムの実装の紹介を行った．そして，適用例として微少ファイルの多数作成に対する適用を紹介し，同実験において性能に支配的な処理を多段プロトコルスタック内から的確に発見することが可能であることや，性能に再現性がなく 2 回目以降の処理時間が短くなる理由を明確に確認できることなどを示した．

今後は，本解析により判明した性能決定要因をふまえ，性能向上手法を考察し，その実現を目指していく．

参考文献

[1] 山口実靖，小口正人，喜連川優，“IP ネットワークストレージシステムのトレース解析,” 第 3 回情報科学技術フォーラム一般講演論文集第 2 分冊，pp.41-42, September 2004.