B-004

iSCSI ストレージアクセス時における TCP 輻輳ウィンドウとシステム性能の関連性評価

Relationship between TCP Congestion Window and System Performance on iSCSI Storage Access

豊田 真智子 † 山口 実靖 ‡ 小口 正人 †
Machiko Toyoda Saneyasu Yamaguchi Masato Oguchi

1. はじめに

マルチメディアコンテンツなどのデータを大量に蓄積するアプリケーションの登場により，コンピュータシステムにおいて処理するデータ量が増大している．そのため，ストレージ管理コストが大きな問題の 1 つとなり，この問題を解決するために SAN(Storage Area Network) が注目されている．

SAN では，ストレージ機能提供側を Target，ストレージ要求側を Initiator と呼ぶ．現在，SAN の中でもファイバチャネルを用いて構築する FC-SAN が普及し始め，企業などで導入されている．しかし，異なるメーカ間での相互接続性が低く，サーバの多くがファイバチャネルに非対応であること，導入や管理にコストが多くかかるため，容易に導入することができないというのが現状である．そこで，Ethernet と TCP/IP という汎用的な技術を用いて構築する IP-SAN の中でも，2003 年 2 月に IETF により承認された iSCSI が，次世代 SAN として大きな期待を集めている．

iSCSI を用いた遠隔ストレージアクセスの問題として，TCP/IP のみで構成されたネットワークと比較した際のスループット低下が挙げられる [1]．iSCSI は SCSI over iSCSI over TCP/IP over Ethernet というプロトコルスタックを構成し，ストレージアクセス時にはこれらすべてのプロトコルが複雑に関連して動作している．そのため，性能向上のためには，通信において重要な階層となる TCP 層の状態を把握し，評価することが重要である．

そこで本研究では，iSCSI を用いてストレージアクセスを行い，TCP 層の重要なパラメータである輻輳ウィンドウの値とスループットを測定し，これらの関連性を評価する．

2. Linux TCP 実装

Linux OS の TCP 実装は，状態機械として実装されている（図 1）．パケット受信時には，状態機械の状態によりその処理が異なる．

正常状態においては ACK 受信ごとに輻輳ウィンドウは増加する．しかし，エラーが発生すると異常状態に遷移し，輻輳ウィンドウが減少する．異常状態には，主に 3 つの状態が存在する．

状態 CWR は，TCP 実装が上位層から送信依頼されたデータを下位層に送信要求した際，デバイスドライバ内のバッファが溢れることにより生じる送信時のエラーである．Linux TCP はこれを Local Device Congestion の通知と判断する．このエラーが発生すると，状態が Open から CWR に遷移し，輻輳ウィンドウが減少する．

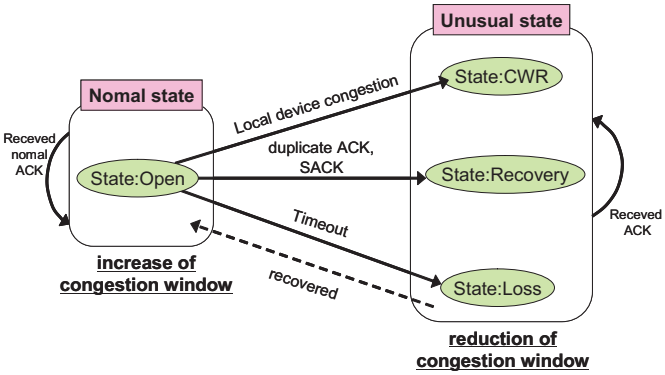

図 1: Linux TCP 状態機械遷移図

状態 Recovery は，SACK や重複 ACK を受信することで生じるエラーである．Linux TCP により，パケットロスであるが，輻輳ではないとみなされ，タイムアウトを待たずに高速再転送が行われる．このエラーが発生すると，状態が Open から Recovery に遷移し，輻輳ウィンドウが減少する．

状態 Loss は，一定時間異常たっても ACK が返ってこないことで検出されるエラーである．Linux TCP は輻輳とみなし，再転送が行われる．このエラーが発生すると，状態が Open から Loss に遷移し，輻輳ウィンドウが減少する．

異常状態に遷移した後，回復と判断されると再度正常状態に遷移し，輻輳ウィンドウが増加する．

3. スループット測定実験

iSCSI プロトコルを用いたストレージアクセスにおける性能を評価するため，基礎実験として，Initiator（サーバ）から Target（ストレージ）の raw デバイスへシーケンシャルリードアクセスを行い，ブロックサイズを変えてスループットの測定を行った．

3.1 実験環境

Initiator と Target 間は，Gigabit Ethernet で接続し，TCP/IP 接続を確立した．サーバとストレージ間が離れている場合のストレージアクセスを想定した実験を行うため，Ethernet の接続途中に人工的な遅延装置として FreeBSD Dummynet[2] を挟み，片道遅延時間を“ 2ms ”，往復“ 4ms ”に設定し，クロスケーブルで接続した．Initiator，Target，Dummynet はすべて PC 上に構築し，Initiator と Target には Linux を，Dummynet には FreeBSD をインストールした．Target はメモリモードで動作させ，ディスクへのアクセスは伴っていない．

†お茶の水女子大学
Ochanomizu University
‡東京大学生産技術研究所
Institute of Industrial Science, The University of Tokyo

表 1: 使用計算機

OS	Initiator, Target ： Linux2.4.18-3 Dummynet ： FreeBSD 4.5 - RELEASE
CPU	Intel Xeon 2.4GHz
Main Memory	512MB DDR SDRAM
Hard Disk	36GB SCSI HD
NIC	Intel PRO/1000XT Server Adapter on PCI-X (64bit, 100MHz)

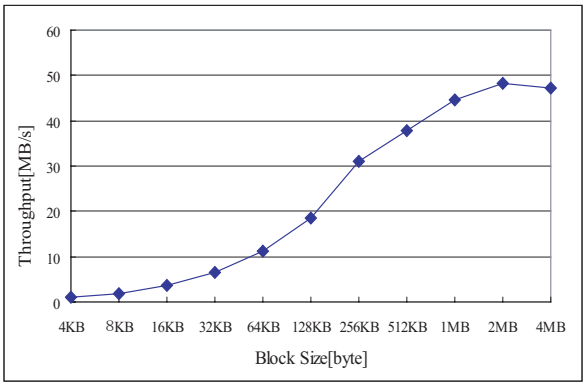

図 2: スループット測定結果

これは，無限に高速なストレージデバイスとみなすことができる．また，微小パケットによる性能低下を防ぐために，Nagle アルゴリズムを停止する TCP_NO_DELY オプションを用いた．実験で使用した計算機の環境を表 1 に示す．また，Initiato 及び Target の iSCSI 実装には，ニューハンプシャー大学 InterOperability Lab [3] が提供する UNH IOL reference implementation ver.3 on iSCSI Draft 18 を用いた．

3.2 実験結果

ブロックサイズを変化させた時のスループットは図 2 の結果となった．測定結果より，シーケンシャルリードアクセスのスループットは，ブロックサイズに比例して増加しており，ブロックサイズが 2MB 程度で飽和している．

4. 輻輳ウィンドウ測定実験

Initiator から Target へのシーケンシャルリードアクセスの性能について，さらに詳しく調べるために，TCP フロー制御に用いられている輻輳ウィンドウの値を観察した．実験は，Initiator から Target の raw デバイスへシーケンシャルリードアクセスを行い，輻輳ウィンドウは，自作ツール [4] を用いることで可視化した．この時，カーネルメモリ空間へのアクセス間隔は，3 秒に設定して測定を行った．実験環境は，3 章のスループット測定実験と同様である．

4.1 輻輳ウィンドウの時間変化

本実験における測定の結果から，ブロックサイズの変化によって輻輳ウィンドウの振る舞いが，大きく 2 つのパターンを示すことが観測された．ブロックサイズが

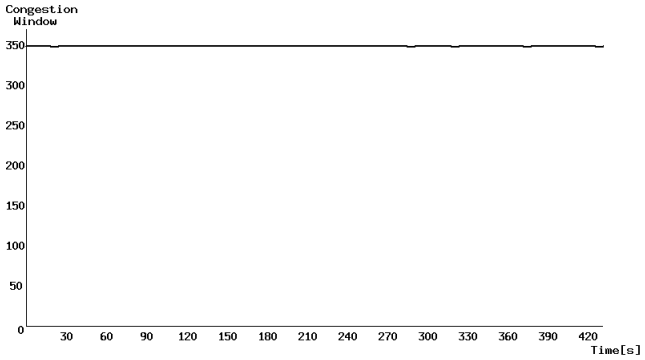

図 3: 輻輳ウィンドウの時間変化：490KB

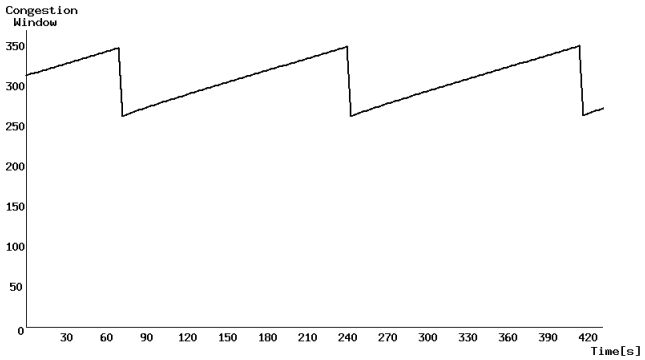

図 4: 輻輳ウィンドウの時間変化：500KB

490KB までは，ブロックサイズの増加につれて，輻輳ウィンドウの最大値も増加し，各ブロックサイズにおいては，輻輳ウィンドウはほぼ一定値に保たれて通信が行われた．しかし，ブロックサイズが 500KB 以上の時は値が徐々に増加し，ある値で低下し，再度増加するという典型的な鋸形の変化を示した．この 2 つのパターンの例として，ブロックサイズが 490KB における変化（図 3），500KB における変化（図 4）のグラフを示す．

4.2 イベント検出

輻輳ウィンドウが減少した原因を詳しく調べるため，推移中の輻輳ウィンドウの TCP 実装内におけるイベント発生（異常状態への遷移）の検出を行った．検出方法は，カーネルソースコードの各イベント遷移時の関数に，独自の関数を挿入することで可能にした．

ブロックサイズが 500KB の時の輻輳ウィンドウの時間変化グラフに，検出したイベントを加えたグラフが図 5 である．

イベント検出の結果，輻輳ウィンドウが最大値を示した時，状態 CWR に遷移することが確認された．このことから，輻輳ウィンドウが低下する原因は，デバイスドライバ内のバッファが溢れることにより，Local Congestion が生じるためであると考えられる．

一方，ブロックサイズが 490KB 以下においては，ほとんどイベントが検出されなかった．輻輳ウィンドウが一定値となるのは Linux TCP 実装独自のものであり，通信中に一度輻輳ウィンドウが設定されると，そのウィンドウの値を使い切らない限りは輻輳ウィンドウが変化しな

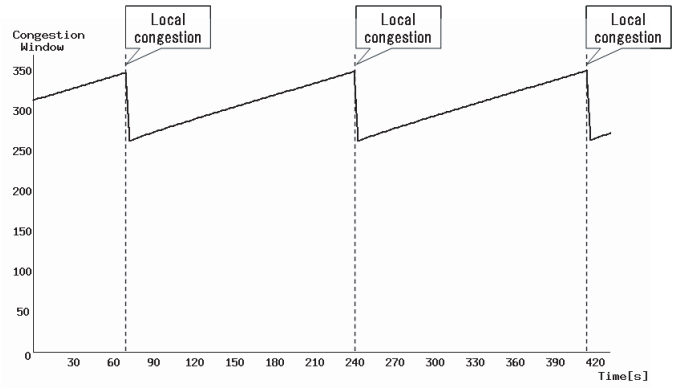

図 5: 輻輳ウィンドウとイベント発生の時間変化

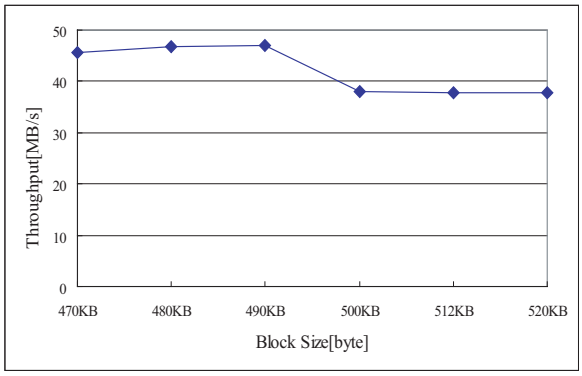

図 6: 輻輳ウィンドウが減少する境界付近におけるスループット

いという特徴を持つ．そのため，ブロックサイズ 490KB 以下では，一度設定された値を超えることなく，通信が行われていることがわかる．

5. スループットと輻輳ウィンドウの関連性

輻輳ウィンドウが異なる変化を示す境界付近のブロックサイズにおいて，再度詳細にスループットの測定を行った．結果は図 6 である．

3 章におけるスループットの測定では確認できなかったが，ブロックサイズが 500KB を境に，スループットが一時低下する様子が見られた．また，ブロックサイズが 500KB 以上において，スループットを時間軸上で観測したところ，輻輳ウィンドウが低下を示した時に，スループットも低下を示した．その様子を図 7 に示す．一方，ブロックサイズが 490KB 以下においては，あまり変化が見られず，安定した性能を示した．

スループットという観点においては，ブロックサイズが 500KB 以上の時の方が性能がよいが，この場合，輻輳ウィンドウが増加しては低い値にリセットされるという変化を繰り返すため，この変化を抑えるようにコントロールすることによって，さらなるスループットの向上が見込めると考えられる．ブロックサイズが 490KB 以下の時は，パケット数を一定に保ったままストレージアクセスを行うことができるので，この場合の方が安定した，効率の良い通信を行うことができる．

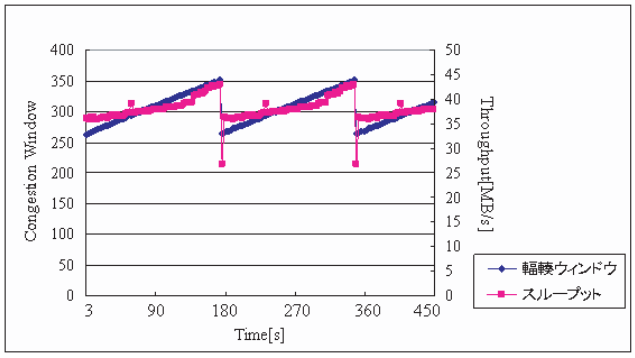

図 7: スループットと輻輳ウィンドウの時間変化

6. まとめと今後の課題

iSCSI プロトコルを用いてストレージアクセスを行い，スループットの測定，輻輳ウィンドウの測定を行った．ブロックサイズが 490KB 以下と 500KB 以上の場合で，輻輳ウィンドウが異なる時間変化を示すことが確認された．また，その時のスループットを測定したところ，500KB を境に一時低下を示した．輻輳ウィンドウが一定値をとるブロックサイズにおいては性能が安定している一方，鋸型の変化を示すブロックサイズにおいては，スループットも変化を示し，効率の良い通信が行われていないことがわかった．

通常 iSCSI においては，Initiator と Target 間でのストレージアクセスが大きなブロックサイズで実行されるため，本実験におけるブロックサイズが 500KB 以上のケースに当てはまる通信が多いと予想される．そのため，今後は，輻輳ウィンドウが落ちる直前の値を保持し，性能を劣化させずにストレージアクセスを行えるよう，輻輳ウィンドウをコントロールする手法を検討する予定である．

謝辞

本研究は，一部，文部科学省科学研究費特定領域研究課題番号 13224014 によるものである．

参考文献

[1] 山口実靖，小口正人，喜連川優：”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”，電子情報通信学会論文誌 Vol.J87-D-I，No.2，pp.216-231，February 2004.

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

[3] InterOperability Lab：”Univ, of New Hampshire”，http://www.iol.uhn.edu/consortiums/iscsi/

[4] 豊田真智子，山口実靖，小口正人：”iSCSI ストレージアクセス時の TCP フロー制御のリアルタイム可視化”，電子情報通信学会，B-16-9，p618，March 2004.