5J-5

iSCSIアクセス時の複数台Initiatorを用いたTCP輻輳ウィンドウ制御方式の一検討

豊田 真智子† 山口 実靖‡ 小口 正人†
†お茶の水女子大学 ‡東京大学生産技術研究所

1. はじめに

ストレージを効率よく安価に管理するために IP-SAN(IP-Storage Area Network) が登場し，その代表的なプロトコルとして iSCSI が利用されている．既存インフラを利用して安価に構築でき，遠隔地へのデータバックアップを容易に実現できるといったメリットがある反面，性能に関する課題を多く残している [1].

本稿では，iSCSI を用いて複数台のサーバ (Initiator) からストレージ (Target) にアクセスする環境を想定し，TCP パラメータである輻輳ウィンドウを動的にコントロールすることにより各 Initiator のスループットのばらつきを抑制し，効率的なストレージアクセスを実現する複数台 Initiator 輻輳ウィンドウコントロール手法を提案する．提案手法を高遅延環境下の iSCSI シーケンシャルリードアクセスに適用することにより，各 Initiator の性能が均一化され，公平なストレージ利用が可能となり，本手法の有効性が確認された.

2. 複数台 Initiator 輻輳ウィンドウコントロール手法

スループットのばらつきを抑制するために提案する複数台 Initiator 輻輳ウィンドウコントロール手法は，TCP ソースコードに独自の関数を挿入することで TCP パラメータをモニタすることができる仕組みを Target に実装し，Target からの輻輳ウィンドウ通知を受けて，アプリケーションがストレージアクセスのブロックサイズを調節する仕組みをミドルウェア機能として提供する機能を持つ．Linux TCP 実装における輻輳ウィンドウの変化は，一定値となるか増加後急激に低下するという変化を繰り返すかのどちらかである．輻輳ウィンドウが低下する原因としては，実験環境に依存するエラーを検出した場合と，ネットワーク状態に依存するエラーを検出した場合に分けることができる．本手法は，実験環境に依存するエラーである，送信側のデバイスドライバのバッファが溢れることによる CWR エラーを検出して輻輳ウィンドウが低下した場合に適用するものとする．また，モニタしているその他の TCP パラメータを用いて各 Initiator ごとの輻輳ウィンドウを識別して取り出すことにより，複数台 Initiator における制御手法の適用が可能となった．本手法による輻輳ウィンドウの制御手順を以下に示す.

1. Target で各 Initiator の輻輳ウィンドウをモニタし，変化を観察する.
2. 輻輳ウィンドウの振舞によって以下の異なる処理を行う.
   (a) 観察している Initiator のいずれかで CWR が検出され，輻輳ウィンドウが低下した場合
   - それまでにすべての Initiator の輻輳ウィンドウが同じ値で一定値となったと判断していれば，最適であるとして Initiator への通知は行わない
   - すべての Initiator が一定値であることを判断していない場合には，エラーが検出された Initiator の輻輳ウィンドウの最大値 (低下する直前の最大の輻輳ウィンドウ) を Initiator ごとに記録し，各 Initiator の輻輳ウィンドウ最大値の中で最も小さな値をすべての Initiator に通知し，通知した輻輳ウィンドウ値を Target においても記録する

   (b) 一度も CWR エラーを検出せずに，すべての Initiator の輻輳ウィンドウが同じ値で一定値であると判断した場合
   - その時の輻輳ウィンドウの限界値 (CWR エラーが起こらなかった場合の最大値) をすべての Initiator に通知し，通知した輻輳ウィンドウ値を Target においても記録する.
3. 通知を受けた Initiator では，ミドルウェアが輻輳ウィンドウからブロックサイズを決定し，アプリケーションがブロックサイズを再指定する.
4. Initiator から Target にシーケンシャルリードコマンドを送信し，ストレージアクセスを行う.
5. Target が Initiator に向けて要求されたブロックサイズのデータ転送を実行する.
6. この処理を，すべての Initiator の輻輳ウィンドウが同じ値で一定値と判断され，最大値と限界値の差が十分小さくなるまで繰り返す.

本手法適用後，CWR エラーは検出されるがすべての Initiator の輻輳ウィンドウはほぼ一定に保たれ，その時のブロックサイズが本手法から計算される最適値となる．なお，本手法においてミドルウェアが指定するブロックサイズは以下の式を用いて計算した.

転送ブロックサイズ $[byte]$ = 輻輳ウィンドウ値×最大転送単位（MTU）

本実験時の MTU（Maximum Transmission Unit）は Ethernet の最大セグメント長（1500Byte）から TCP/IP ヘッダ（オプションを含む）を除いた 1448Byte である.

3. 提案手法を用いた性能測定実験

iSCSI ストレージアクセスに提案手法を適用し，提案手法を適用しなった場合との比較実験を行なう．4 台の Initiator から Target へシーケンシャルリードアクセスを行い，iSCSI の使用が想定される高遅延環境下における性能を測定する．Initiator が指定するブロックサイズは，提案手法を用いない場合は 1024KB に，提案手法を用いた場合はその初期値を 1024KB に設定した.

A Study of Controlling TCP Congestion Window using Multiple Initiator on iSCSI Access
† Machiko Toyoda, Masato Oguchi
‡ Saneyasu Yamaguchi
Ochanomizu University (†)
Institute of Industrial Science, The University of Tokyo (‡)

表 1: 使用計算機

CPU	Initiator: Intel Pentium 800MHz Target, Dummynet: Intel Xeon 2.4GHz
Main Memory	Initiator: 640MB Target, Dummynet: 512MB
OS	Initiator, Target: Linux2.4.18-3 Dummynet: FreeBSD 4.9 - RELEASE
NIC	Initiator, Dummynet: Intel PRO/1000MT Server Adapter Target: Intel PRO/1000XT Server Adapter

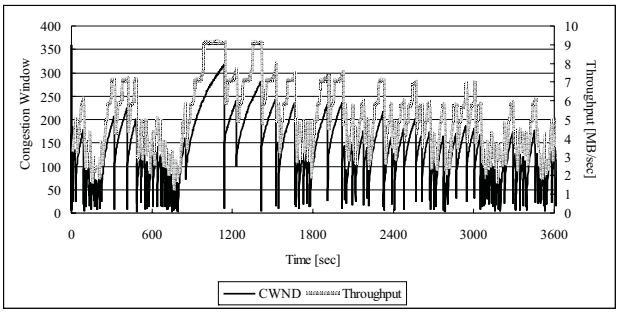

図 1: 提案手法を用いない場合の輻輳ウィンドウ，スループットの時間変化 (片道遅延時間:16ms)

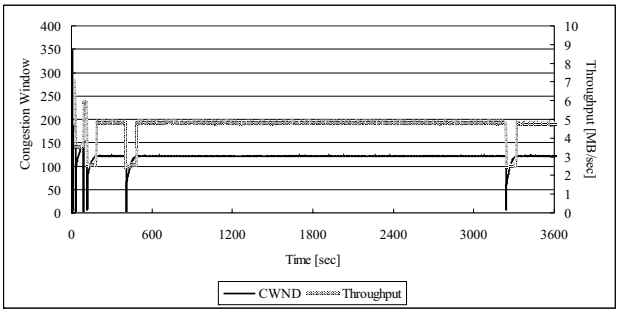

図 2: 提案手法を用いた場合の輻輳ウィンドウ，スループットの時間変化 (片道遅延時間:16ms)

3.1 実験環境

本実験は以下の環境で行った．Initiator と Target 間は Gigabit Ethernet で接続し，TCP/IP 接続を確立した．遠隔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，高遅延環境を構築した．iSCSI を利用したストレージアクセスにおけるネットワークの性能を調べるため，Target はメモリモードで動作させ，ディスクアクセスを伴わないように設定した．実験で使用した計算機の環境を表 1 に示す．

また，本実験で用いた iSCSI 実装において，Target にはニューハンプシャー大学 InterOperability Lab が提供する UNH IOL reference implementation ver.3 on iSCSI Draft 18[3] を用い，Initiator には UNH 実装の Initiator と同等の機能を持ち，かつ大きなブロックサイズのデータ転送も行える自作 Initiator を用いて実験を行った．

3.2 実験結果

前節の実験を行った結果として，提案手法を用いなかった場合と用いた場合それぞれの環境において，iSCSI シーケンシャルリードアクセスを行った場合の 4 台中 1 台の Initiator におけるスループット，輻輳ウィンドウの時間変化を図 1，2 に示す．

提案手法を用いない場合 (図 1)，Initiator のスループットは不安定となり，輻輳ウィンドウも不規則な増加減少を繰り返していることが確認される．また，輻輳ウィンドウが大きく成長した場合にはスループットは高くなるが，あまり成長せず小さな値である場合にはスループットは低い．一方提案手法を用いた場合 (図 2)，ブロックサイズを調節することにより，CWR エラーは検出されるが輻輳ウィンドウはほぼ一定値となる．CWR エラー検出時に輻輳ウィンドウが減少することによりスループットも一時低下するが，その回復につれてスループットも増加し，ほぼ安定した通信が行われている様子が確認される．

表 2: 片道遅延時間 8ms における各 Initiator 平均スループットの比較

Applied Method	Throught [MB/sec]			
	Initiator1	Initiator2	Initiator3	Initiator4
Not Using Proposed Method	6.86	7.12	8.03	12.09
Proposed Method	7.56	7.64	7.5	7.57

表 3: 片道遅延時間 16ms における各 Initiator 平均スループットの比較

Applied Method	Throught [MB/sec]			
	Initiator1	Initiator2	Initiator3	Initiator4
Not Using Proposed Method	3.73	6.71	4.29	4.57
Proposed Method	4.69	4.72	4.67	4.66

表 2，3 は，提案手法を用いた場合と用いなかった場合に 10GB のデータをリードした時の各 Initiator の平均スループットの一例を比較したものである．提案手法を用いないシーケンシャルリードアクセスの場合，アクセスブロックサイズが大きな値でであるため各 Initiator において CWR エラーが頻発し，その度に輻輳ウィンドウが急激に低下する．この振舞を不定期に繰り返すためにスループットも不安定となり，各 Initiator の平均スループットが大きく異なる場合がある．しかし，提案手法を適用することにより各 Initiator の性能が均一化され，どのユーザからも公平にストレージを利用することが可能となる．また，提案手法を用いなかった場合とほぼ同等の性能を保つことができるため，複数台の Initiator を用いて iSCSI ストレージアクセスを行う場合には提案手法を適用することが望ましいと言える．

4. まとめ

高遅延環境下の iSCSI ストレージアクセスにおいて，複数台 Initiator 輻輳ウィンドウコントロール手法を適用し，各 Initiator の性能を均一化して，不平等なくストレージアクセスを行うことができる手法を提案した．今後は Initiator と Target が多対多で接続された複雑な環境における性能向上手法を検討していきたい．

参考文献

[1] 豊田真智子，山口実靖，小口正人：“ 高遅延ネットワーク環境における iSCSI リードアクセス時の TCP 輻輳ウィンドウ制御手法の性能評価 ”，先進的計算基盤システムシンポジウム (SACSIS2005)，pp.443-450，2005 年 5 月．

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

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