社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＤＳＭ－46／2007－ＱＡＩ－24（９）
2007／7／19

構造型 P2P システム上の放送におけるより良いデータ分割方法

山之上 卓†

†鹿児島大学学術情報基盤センター 〒890-0065 鹿児島市郡元 1-21-35
E-mail: †yamanoue@cc.kagoshima-u.ac.jp

あらまし 構造型 P2P 通信システムにおけるリアルタイムデータ転送時間の改善について検討を行った。この通信システムにおけるデータ転送時の最適なデータ分割数の理論式を導出し、システムのデータ転送時間を計測し、理論値と照らし合わせ、システムの改良を行い、性能を改善することができた。この手法のストリーミングへの応用についても述べる。

キーワード P2P, TCP/IP, データ分割, 最適解, ストリーミング

A Better Partitioning of Data at Broadcasting on a Structured P2P system

Takashi YAMANOUE†

† Computing and Communication Center, Kagoshima University
1-21-35 Korimoto, Kagoshima 890-0065, Japan
E-mail: †yamanoue@cc.kagoshima-u.ac.jp

Abstract Communication quality improvement of a structured P2P communication system is described. A theoretical equation for optimum number of data partitioning is shown. Performance of the previous system was measured and it was compared with the theoretical equation. The system was improved using the result. The application of the theoretical equation to streaming is also considered.

Keyword P2P, TCP/IP, Data partitioning, Optimum value, streaming

1. はじめに

インターネットや LAN が普及し、これらのネットワークに多数のコンピュータ（ホスト、ノード、端末）が接続された分散システムが一般的に利用されるようになっている。分散システム上において、同一のデータを短時間で信頼性を持って多くのホストに配信したい場合がしばしば生じる。たとえば学校のパソコン教室では教師端末の画面を多数の生徒ホストに表示させるシステムが多く利用されている。このとき、パソコン端末の画面に少しでも欠けや誤表示があると授業に支障が生じる場合があるため、短時間で多数の生徒端末に信頼性を持って教師端末の画面を転送しなければならない。また、多くのホストに同一のソフトウェアをインストールしたい場合がある。このとき、一台ずつこのソフトウェアのインストールや更新を行うと多大な労力や時間を浪費してしまう。セキュリティ問題が深刻化する中，頻繁にソフトウェアの更新を行うことも重要になっているが、この場合も同一の更新情報を多数のホストに信頼性を持って短時間に送る必要がある。

同一のデータを短時間で信頼性を持って多くのホストに送信する手段の一つとして構造型 P2P 通信システムの利用がある。構造型 P2P 通信システムは一定の規則に従って TCP でホスト間を接続し、ホスト間で通信を行わせるものである。ホスト間通信が他のホスト間通信に影響を与えないスイッチを使ってホスト間を完全 n 分木状に接続した場合、任意のホストからデータを配信してすべてのホストがこのデータをすべて受け取るまでの時間は、およそホストの数の対数に比例する。TCP を使っているので信頼性を確保できる。

我々はホスト間を TCP で完全２分木状に接続した構造型 P2P 通信システムの研究を行っており、論文[1][2]において構造型 P2P 通信システムを使って１台のホストのディスプレイの画像を、ネットワークを通じて多数のホストに転送するシステムの通信特性について述べている。本報告では、論文[1][2]では述べられていなかったデータ分割数の最適値を表す理論式を示し、現在我々が開発している教育システム SOLAR-CATS[3]において画像転送速度の計測を行い、理論式との比較を行う。そしてこの結果を踏まえてシステムの改良を行い、その結果について述べる。また、音声や動画などのストリーミングデータの実時間配布への応用についても述べる。

2. 完全 n 分木上における同一データ配布

すべてのノード間の通信速度およびノードの能力

が一定で、一対の通信が他の通信に影響を与えないことを仮定した場合に同一データを1つのノードから多数のノードに転送する時間について考える。完全 b 分木状(木の高さは i)に接続されたノードの根から t 個に分割されたデータをすべてのノードに送信し終わるまでの時間 T の理論値は式(1)より求められる[1]。2分木(b=2)のとき、ノード数 N は b^i-1 となる。

$$T = C_B b(t+i-2)+d(t) \qquad (1)$$

ここで、C_B は直接接続された一対のホスト間における1データブロックの転送時間を示し、式(2)より求められる。

$$C_B = \frac{8s}{wt}+d_H \qquad (2)$$

ここで S は全体のデータのサイズ(バイト)、w はネットワークの通信速度(bps)、d_H はホスト間通信の遅延時間であり、ネットワーク機器(スイッチングハブ)の遅延時間とホスト間でのデータ転送時間を加えた値となる。

式(1)の $d(t)$ は転送経路に依存しない、アプリケーション側での処理時間であり、式(3)で近似できると思われる。

$$d(t) = a+ct \qquad (3)$$

ここで a と c はホストの性能やアプリケーションの処理に依存する定数である。

式(1)を変形すると

$$T = \frac{8sb}{w}+a+d_H b(i-2)+\frac{8s}{wt}b(i-2)+t(d_H b+c) \qquad (4)$$

となる。この式を、分割数 t を変数とする多項式であると解釈すると、

$$T = A+\frac{B}{t}+Ct \qquad (5)$$

となる。ここで

$$A = \frac{8sb}{w}+a+d_H b(i-2)$$

$$B = \frac{8sb(i-2)}{w}$$

$$C = d_H + c$$

である。

式(5)により、データサイズ、木の分岐数、木の高さ $i \equiv \log_2(N)$ の積が、バンド幅 w とデータの分割数 t の積と比べて非常に小さい場合、データ配信が終了するまでの時間は主にデータの分割数に比例することがわかる。この場合、データ転送時間はノード数 N にはほとんど依存しない。また、w はネットワーク機器で定まった値であり、アプリケーションプログラム側で変化させることはできない。従って、この場合(t が大きい場合)にデータ転送時間を短くするためには、1つのデータブロックを送り出すときの時間 c と、ホスト間でのデータ転送時間が含まれる d_H を小さくすることが有効であることがわかる。

t を変化させたとき転送時間が最小となる t の値 t_0 は、

$$T' = -\frac{B}{t^2}+c = 0 \qquad (6)$$

を t について解いた解の正の値であり、

$$t_o = \sqrt{\frac{B}{C}} = \sqrt{\frac{8sb(i-2)}{w(d_H b+c)}} \qquad (7)$$

となる。この式によると、分割数 t には最適な値が存在し、大きければよいとか小さければよい、というわけではないことがわかる。また、最適な分割数はおよそデータ全体の大きさと木の高さの平方根に比例し、バンド幅とホスト間通信の遅延時間の平方根に反比例することがわかる。

3. SOLAR-CATS の画面転送時間

我々は教育支援システム SOLAR-CATS を開発している[3]。SOLAR-CATS はグループ内のノード（端末）間を完全2分木状に接続した構造型P2P通信システムの一つである。SOLAR-CATS は、お絵かきプログラム、テキストエディタ、Web ブラウザ、簡単なプログラミング言語のプログラミング環境、英作文支援システムなどのアプリケーションを備えた一種のWISIWYS(What I See Is What You See)システムである。SOLAR-CATS は

- グループに参加している教師のアプリケーション操作を学生端末上で、実時間で表示する
- 教師や学生が独立してアプリケーションを操作する
- グループに参加している教師や学生がクラス内で

アプリケーション操作を共有することにより、共同作業を行う
- それぞれの端末において、アプリケーション操作の記録と再生を行う
- ファイヤーウォールで隔てられた端末教室の端末を接続して、1つのグループを作り、その中で操作の共有を行う

などの機能を持つ。JAVAで開発しているため、プラットフォーム独立であり、様々なOSが混在した環境でも利用できる。グループ内で1つの操作を共有し、なおかつグループ内の誰でもその操作を可能とするためには、グループ内のメンバーが勝手に独自の操作を行うことがないよう、排他制御が必要になる。SOLAR-CATSは排他制御の機構も持っている[4]。

SOLAR-CATSのお絵かきプログラムはグループ内の1つのホストの画面のビットマップ画像をグループ内の他のすべてのノードに転送する機能を持つ。今回、この画面転送機能の性能を計測し、2章で述べた理論式と比較を行った。この比較の結果、システムの改良すべき点が明らかになり、それに従ってシステムの改良を行い、性能を最適値に近づけることができた。

改良前のシステムにおいて1画面（幅 701pixel, 高さ 470pixel, 1pixel あたり 4byte, 1.3Mbyte）を根ノードからすべてのノードに転送するとき、必要な時間は表1で示される値であった。ここで使用した環境は、ネットワークは 100Mbps のスイッチ(遅延時間は 2.3μsec)、各ホストは CPU Intel Pentium 1.3GHz 以上、メモリ 500MB 以上、OS は Windows XP（葉ノードの1台は Vista)、を利用した。Java は JDK 1.4, 1.5, 1.6 を利用した。

表1．改良前のシステムにおける画面転送時間

ノード数 (根も含む)	時間(sec)（10回計測）		
	最短	最長	平均
3	8.84	9.51	9.01
6(i=3)	8.68	10.05	9.33

改良前のシステムでは、1画面を縦横 24pixel の正方形(2304byte)ずつ分割し、この正方形のデータを一度文字列に変換して 3456byte のデータとし、これを順番に送信していた。分割数は 572 である。

式(1)を見ると、式(2)の

$$C_B = \frac{8s}{wt} + d_H$$

および、式(3)の

$$d(t) = a + ct$$

を小さくすると時間を短縮できることが分る。このため、以下を行った。

○ 今までビットマップデータを一度文字列に変換して送信していた。この変換時間と変換したことによるデータ量の増加を抑えるため、ビットマップデータを binary データのままで送信するようにした。これにより式(2)の s と式(3)の c が小さくなることが期待できる。また、データ送信時およびデータ転送時のコピーの回数を少なくした。これにより、式(2)の d_H が小さくなることが期待できる。

○ 分割数を減らし、一度に多くのデータブロックを送信できるようにした。これにより、式(3)の t が小さくなることが期待できる。しかしながら、P2P のようにバケツリレーでデータ転送を行う場合、式(2)の値が大きくなることも意味するので適切な値を選ぶことが必要である。式(7)において、s=1.3Mbyte, b=2, w=100Mbps であり i=3(N=7), d_H =0.7msec(他の実験結果より推定), c=10msec(他の実験結果より推定)とした場合、この式の値は 7.0 となり、7分割付近で転送時間が最小になることがわかる。ここでは、縦横 192pixel の正方形(約 140KByte)を一つのデータブロックとし、画面を5分割することとした。

以上の改良を行い、表1の実験で使用したものと同じ環境の7台のグループで画像転送を行ったところ、1画面の転送時間は平均 2.0 秒となった。表2に改良を行ったシステムにおいて、1度に送るデータブロックの大きさを変えた場合の転送速度の計測結果(それぞれ 10 回計測)を示す。実験結果では分割数が最も少ない5分割のときが最も速かったが、ノード数7の計測の最中、5回目以降はプログラムが固まってしまい、計測できなかった。このように、分割数が少なくなると、実行が不安定になる場合もある。また表1と同じ 572 分割の場合でも改良後の方が速くなっている。転送される総データ量だけで考えると 2/3 程度の改善が得られるはずであるが、7 ノードの場合それ以上に改善されているので、文字列への変換を省いた効果も入っているものと思われる。

4. ストリーミングへの応用

多数のノードへ音声や動画のストリーミング配信を行う場合、信頼性は100%必要ではないため、IPマルチキャストでUDPパケットを送信する場合が多い。

表２．改良後のシステムにおける画面転送時間

ノード数（根も含む）	分割数（縦横幅）	時間(sec)(10回計測) 最短	最長	平均
3	572(24)	4.47	8.58	6.13
	143(48)	1.72	2.54	2.09
	36(96)	2.10	3.35	2.56
	21(128)	1.58	2.05	1.82
	9(196)	1.65	2.35	1.99
	5(256)	1.48	2.14	1.92
7	572(24)	3.21	9.75	4.67
	143(48)	2.03	4.95	3.11
	36(96)	2.21	3.09	2.64
	21(128)	1.97	2.45	2.25
	9(196)	1.90	2.48	2.18
	5(256)	1.99	2.08	2.01 (4回)

しかしながら、IPマルチキャストを行うためにはルータの設定が必要となる場合がある。関係ないホストがいても１つのLAN全体にデータを流してしまう欠点もある。また、UDPはネットワークに流れる他の通信に遠慮せずに送られるため、問題が生じる場合がある。２分木状にノードをTCPで結合したシステムでストリーミングが行えると、このような問題を解決できる可能性がある。

我々は教育支援システム SOLAR-CATS を開発している[3]。現在、SOLAR-CATS に動画と音声の送受信機能を実装しようとしており、一部動いている。２章で述べた最適なデータ分割を動画と音声の送信のようなストリーミング配信への応用を考えてみる。

4.1 ストリーミングで考慮する項目

２分木状にノードをTCPで結合したP2Pシステム上でストリーミングを行う場合に関係する事項は以下のものがある。なお、途中のノードで障害が発生する場合はないものとする。

○ データバンド幅 w_d (bps)

○ ネットワークバンド幅 w (bps)
データバンド幅はネットワークバンド幅を超えることはできない。ネットワークのバンド幅を超えた量のデータをTCPで連続して送ることは、データ圧縮を行なわない限り不可能である。また、データを圧縮し元に戻すには時間がかかる。

○ 許容遅延 T_a (ms)

電話などの実時間性が求められる場合は、許容できる遅延時間が決められている場合がある。

○ ネットワークの遅延
ネットワークの遅延は必然的に生じるものであるが、許容遅延を超える遅延があるネットワークは利用できない。これは、式(1)の d_H に含まれる。

○ ジッタ T_j (ms)

ネットワークの遅延時間のゆらぎである。分散で表す場合とpeak to peak で表す場合があるが、ここでは peak to peak で表すものとする。

○ バッファ時間 T_b

ジッタを吸収するため、受信側にバッファを置くことが多い。バッファ時間はジッタより大きい必要がある。T_b は式(3)の $d(t)$ に含まれるものとする。

○ エンコーディング、デコーディング時間
エンコーディング時間は、音声・画像データを符号化および圧縮するのに必要な時間である。デコーディングは、符号化・圧縮されたデータを元に戻す時間である。これらの時間は、式(3)の $d(t)$ に含まれる。

4.2 ストリーミングが行える条件

ストリーミングは、単位時間当たりに一定量あるいは変動する量のデータを連続して送信するものであるが、ここでは、u(byte)の固定量のデータ(符号化・圧縮のも含む)をt個に分割し、これを T_u/t (sec)時間ごとに連続して送信するものと考える。完全2分木(b=2)状に結合された高さiのノードのグループの根からストリーミングを行うとき、以下の条件を満たす必要がある。

$$C_u bt < T_u$$

$$d(t) > T_j$$

$$T_{stream} < T_a$$

ここで C_u は式(2)の C_b において、データサイズを表すsの代わりにuを用いたものである。T_{stream} は式(1)の C_b の代わりに C_u を用いたものである。

4.3 音声通信の場合の最適分割数の推定

20KHz のサンプリングレートで16ビット(2byte)モノラルの精度で音声をサンプリングした場合、圧縮をおこなわなければ、一秒あたりのデータ量は、40000 byte/sec となる。

T_a = 200ms , b=2, i=3(N=7)

とし、式(1)において d_H を無視すると、

$$\frac{T_u}{t}b(t+i-2)+d(t)<200\,\mathrm{ms}$$

これを変形して値を代入すると

$$T_u 2(1+1/t)+d(t)<200\,\mathrm{ms}$$

を満たさなければならないことになる。本来、ジッタの時間 T_j を吸収するバッファを用意すべきだが、これを使うとそこでまた遅延が発生するため、ここではジッタの吸収は行わないことにする。仮にd(t)はtにあまり依存せず、その値を20msとすると、左辺が最も大きくなるときのtは1なので、

$$T_u+20/4<50\,\mathrm{ms}$$

$T_u<45\,\mathrm{ms}$となる。$T_u=40\,\mathrm{ms}$ とすると、u=1600byte となる。これを式(7)のsとし d_H=10ms, c=0.04 と仮定して、あらためて式(7)に代入すると、t_0=0.2 となり、分割しない方が良いことがわかる。

なお、連続した音声をTCPで流す場合、分割したデータ間にすきまがあると、これが蓄積していき、受信側の遅延が大きくなっていく。この現象を避けるため、現在の実装では、そのデータが発生した時間と一緒にデータを送信し、一定以上の遅延が生じたらデータの読み捨てを行うようにしている。

4.4 画像通信の場合の最適分割数の推定

動画は遅延をあまり考えず、10秒間に 1 枚の圧縮なしの画像を送るものと仮定する。全2重の100Mbpsのネットワークで、2分木でこの動画を送ろうとすると、d_H を無視した場合、

[一枚の画面の大きさ(Mbyte)]
×8(bit/byte)×2(分木)×10(枚/sec) < 100Mbps

を満たす必要がある。これを満たすには、一枚あたりの画像の大きさが625KBより小さい値となる。したがって、1 ピクセル4byte で横500pixel, 縦300 pixelの解像度程度以内と推定できる。d_H=10ms, c=0.04としたとき、この場合の最適な分割数の推定値は3.99 であり、約4分割が良いことになる。

5. 関連研究

Tanenbaum のネットワークの教科書[5]には、ネットワークプログラム開発の指針として以下が示されている。

1. CPU スピードはネットワーク・スピードよりも重要である。
2. パケットの個数を減らし、ソフトウェアのオーバーヘッドを削減せよ
3. コンテキスト・スイッチを最小にせよ
4. コピーを最小にせよ
5. 沢山の大域幅は入手できるが、低遅延を得るのは難しい。
6. 輻輳からの回復を考えるよりも輻輳を回避せよ
7. タイムアウトを回避せよ

1 について、今回データ転送時にバイナリーと文字列の相互変換をなくすことなどにより、同じ分割数でも性能を改善することができた。今回の実験の場合は理論式および実験結果より 1 より 2 の効果の方が大きいことがわかるが、バケツリレーで多数のノードにデータを配信する場合は無条件に分割数を減らせば良いわけでもない。3 について、現在 SOLAR-CATS は多数のスレッドを同時に走らせている。これは改善できる可能性があるので今後の課題としたい。4 に対しては、ある程度減らすことができた。5 については複数のノード間でデータをバケツリレーしていく P2P 通信方式場合、遅延を短くすることは難しい。しかしながら、2 分木を使うことにより、バケツリレーの最大の段数は全ノード数の対数に比例しており、ある程度バケツリレーの段数を減らすことに成功しているともいえる。6 については TCP を使っていることにより、他に迷惑をかけることが比較的少なくなっているとも解釈できる。7 についてはデータを分割することが有効であると思われる。

データ転送と分割数に関する研究には、山本らの研究[6]もあるが、この研究では最適な分割数の理論式は示されていない。

遅延が大きなネットワークでファイル転送を行うには、複数の TCP 接続を平行して利用することも有効であることが示されている[8]。複数の TCP 接続を利用できるようにシステムを改良すると効果があるかもし

れない。

義家らは VOD でクライアント側からサーバに映像データ配信を要求したとき、ストリーミングデータを細かく分割することにより、クライアント側の待ち時間を短くする手法について述べている[7]。この研究で、分割数と待ち時間の関係を表すグラフが示されているが、も最適な分割数の理論式は示されていない。

6. 終わりに

データ分割数の最適値を表す理論式を示し、我々が開発している教育システム SOLAR-CATS において画像転送速度の計測を行い、理論式との比較を行い、改良のための検討を行ったことについて述べた。また、ストリーミングへの応用についても検討を行った。ストリーミングについてはジッタの吸収に関する検討を行っていないなど多くの課題を残している。今後システムの改良と実験を通じて、理論式の検証、改良を行っていく予定である。

7. 謝辞

本研究の一部は平成17 年度科学研究補助金基盤研究(C)17500041 の補助を受けた。

文　献

[1] Takayuki Hirahara, Takashi Yamanoue, Hiroyuki Anzai and Itsujirou Arita, "SENDING AN IMAGE TO A LARGE NUMBER OF NODES IN SHORT TIME USING TCP", Proceedings of the ICME2000, IEEE International Conference on Multimedia and Expo, pp.987-990, New York City, USA, July 30-Aug.2 ,2000.

[2] 平原貴行, 山之上卓, 安在弘幸, 有田五次郎: TCP を利用した分散ネットワーク環境のための電子黒板システム, 情報処理学会論文誌, vol.43, No.1, pp.176-184, 2002.

[3] 山之上　卓, "多数の端末上のアプリケーション操作の記録再生を行う教育支援システム", 情報教育シンポジウム論文集、情報処理学会シンポジウムシリーズ vol.2005, No.8, (IPSJ SIGCE SSS2005), pp.61-68, Akaigawa, Hokkaido, 21Aug.-23 Aug. 2005.

[4] 山之上　卓, "P2P 技術を利用した分散システム上の実時間操作共有システム",情報処理学会論文誌, vol.46, No.2, p.392-402,2005.

[5] A. S. Tanenbaum, "Computer Networks - 4th ed.", Prentice Hall, 2003.

[6] 山本 裕, 辻 洋, "遠隔データベースアクセス向け大容量データの分割送受信方式", 電気学会 C 部門論文誌, Vol.124, No. 5, pp.1076-1082, 2004.

[7] 義久智樹, 塚本昌彦, 西尾章治郎, "連続メディアデータ放送におけるデータの細分割による効率的なスケジューリング手法", 電子情報通信学会論文誌 D Vol.J87-D1 No.12 pp.1079-1088, 12 月, 2004.

[8] 飯田好美, 佐々木節, 鈴木聡, 八代茂夫, "SRB を利用した分散ファイルシステムの構築", 情報処理学会分散システム/インターネット運用技術研究報告 No.44(2007-DSM-44), pp.1-6, 3 月 2007.