

TCP を用いた電子黒板システムとその Java による試作

平原貴行† 山之上卓‡ 安在弘幸†

†九州共立大学工学部 ‡九州工業大学情報科学センター

概要

分散環境で利用できる、教師画面の一部を受講者端末に転送して表示する電子黒板システムの開発を行っている。信頼性があるが低速な TCP を用いて多数の端末へ画像データの転送を行うため、転送経路をツリー状に構成して並列かつ段階的に転送を行っている。このシステムは TCP を用いており、一般的なコンピュータネットワークで利用可能なほか、遠隔授業への応用も可能である。OS など問わず、あらゆる環境で利用可能になるよう、マルチプラットフォームでネットワーク言語とされる Java を用いて電子黒板システムの試作を行った。Java 版と C++ 版の比較を交え、システムについて考察する。

An Electronic Chalkboard System using TCP and a trial manufacture by Java

Takayuki Hirahara† Takashi Yamanoue‡ Hiroyuki Anzai†

†Faculty of Engineering, Kyushu Kyouritsu University

‡Information Science Center, Kyushu Institute of Technology

Abstract

We are developing an electronic chalkboard system which transmits a still image of teacher's screen to student's terminal and display. In order to transmit image data to a large number of student's terminals by TCP, which is reliable and low speed, the route of data transmission in this system is tree-shaped, and the data is transmitted in a parallel, step-by-step mode. This system employs TCP/IP, therefore, this system is able to use on a typical computer network and apply to teleteaching. In order to use this system on many environments, we developed our electronic chalkboard system by Java, which is multi-platform language.

1 はじめに

近年のパソコンの急速な価格対性能比の向上、およびネットワークの高速化を背景に、多数のパソコンやワークステーションなどを LAN で接続した分散環境が普及してきており、これを利用した教育支援システムの開発が盛んに行なわれている [1][2]. 我々は教師画面の静止画の一部を受講者端末に転送してその画面に表示する「電子黒板」と呼ぶシステムの開発を行なっている。このシステムは TCP/IP を用いており、一般的なパソコンネットワークを備えていれば特別な機器や配線の必要なく導入でき、またインターネットを介した遠隔授業への応用も可能である。

これまで我々は、UNIX オペレーティングシステムを採用するワークステーションシステム、Windows NT を採用するパソコンネットワークシステム用にそれぞれ電子黒板システムの開発を行なっている [3][4][5]. しかしながら、こうした特定環境に依存するシステムは移植性の面で問題があり、遠隔授業への応用を考えた場合にも異なるシステム間での利用が大きな問題となる。そこで我々は、特定の OS 等に依存しないマルチプラットフォーム言語である Java を用い電子黒板システムの開発を行なった。

2 電子黒板システムの概要

我々の電子黒板システムの概要について述べる。この電子黒板システムは、教師画面の一部分のイメージを複数の受講者端末へ一斉転送して表示させるものである。イメージなどの転送には一般的な通信プロトコルである TCP/IP を用いている。したがって、新たな器材の導入や配線を行わずとも、一般的なパソコンネットワーク上で利用出来る。また、インターネット経由での遠隔授業への応用も可能である。

電子黒板システムにおける問題点は、画像のようなサイズの大きいデータを多数の端末へいかに遅延を少なくして送るかという点である。このような多数の端末への一斉転送には一般にブロードキャストやマルチキャストが用いられているが、これらで用いられる UDP は信頼性の保証がないためデータの欠落を生じる恐れがある。静止画では動画などに比べデータ欠落による画面の虫食いはより目立つことから、我々は UDP は静止画の転送には適さないと判断し、信頼性を保証する TCP を用いている。しかしながら、TCP は基本的に 1 対 1 通信であり、またエラー訂正が行なわれるため、UDP と比べ低速である。このため、電子黒板のような一斉通信システムでは実用に耐える転送速度を得ることが困難と考えられる。

我々はこの問題の解決策として、並列プログラミングの技法を用いている。すなわち、データの転送経路をツリー状に構成し、データを段階的かつ並列に転送することで、転送完了時間の短縮を図っている。近年高速 LAN 技術の一つとして広帯域の専用型ハブであるスイッチングハブ (ATX, 10 Base TX, 100 Base TX など) [9][10] が普及してきており、これを導入することで TCP を用いた並列通信などが可能となった。スイッチングハブはコストも低いいため幅広く導入されている。

3 電子黒板システムの Java による試作

我々がこれまでに開発を行なっている電子黒板システムは特定のプラットフォームに依存したものであるが、これは移植性や継承の面で不利である。また、インターネット経由での遠隔授業への応用を考

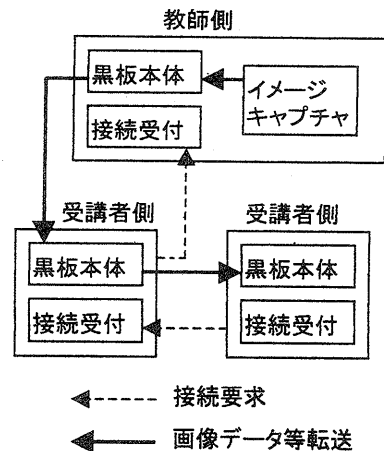


図 1: Java 版電子黒板システムの構成

えた場合、OS などが全く異なる種々のシステム同士を結ばなければならない場合が考えられる。このため、OS などの環境に依存しないシステムとする必要がある。

そこで我々は、OS などに依存しないマルチプラットフォーム言語である Java を用いて電子黒板システムの試作を行なった。これにより異なるシステムへの移植や、遠隔授業における異なるシステム間の接続が可能になると考えられる。また、従来のシステムでは、画面の色が受信側の環境に左右され、教師の意図通りに伝わらないことがあるという問題があったが、Java 版では環境に依存しない画像フォーマットを用いているため、この問題を解決することが出来る。

Java 版電子黒板システムの構成を図 1 に示す。基本的な枠組みは従来の C++ 版電子黒板システムとほぼ同じであるが、Java にはデスクトップイメージに直接アクセスする API がないため、教師画面の部分イメージを切り取るイメージキャプチャを C++ で別に作成し、Java で作成した教師側電子黒板本体と通信することで教師画面を取り込んでいる。

実際の授業では、電子黒板の起動後に遅刻してきた学生がこれに加わろうとしたり、また途中で学生が電子黒板から抜ける場合が考えられる。このような黒板起動中の受講者端末の起動状況の変化に対応するため、受講者端末の起動状況を監視して転送経路の組み替えを行なう監視サーバーを設けた (以降これをオブザーバー (Observer) と呼ぶことにする)。

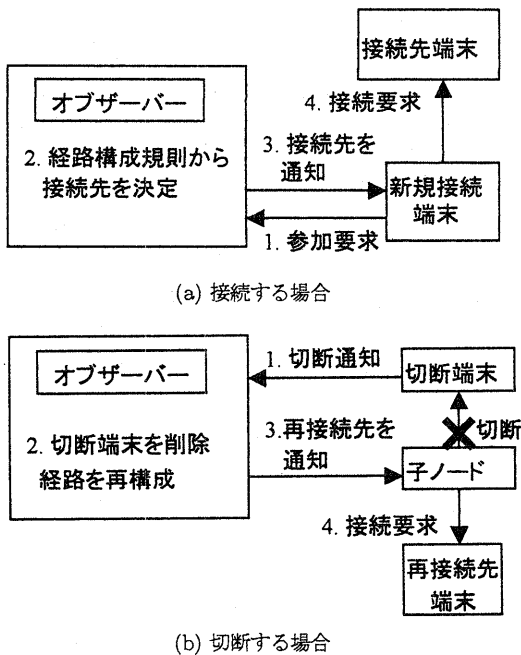


図 2: 監視サーバー (オブザーバー)

オブザーバーを介した転送経路再構成のしくみを図 2 に示す。電子黒板に参加しようとする受講者端末はまずオブザーバーにアクセスする。オブザーバーはこれを受けて転送経路に新しい受講者端末を追加し、接続先の端末をこの受講者端末に通知する。受講者端末は通知を受けて指定された接続先に接続することで電子黒板に加わることが出来る。電子黒板から抜ける場合も同様にオブザーバーにその旨を通知し、オブザーバーはこれを受けて転送経路から抜ける受講者端末を削除して経路を再構成し、接続先を変更する必要がある端末に新しい接続先を通知する。教師端末および各受講者端末は新たに参加する端末からの接続要求を受け付けるプロセスを本体と並行して実行しているため、オブザーバーは接続要求側の端末にのみ接続先の通知を行えばよい。

教師 (あるいは管理者) は、次章で説明する完全木や直列木、MBT などの転送経路の種類をオブザーバー起動時に選択できる。オブザーバーはこの転送経路の構成ルールに基づき、新たに加わるノードが接続すべきノードを決定する。一方削除については、現在は削除したノードに接続していたすべてのサブツリーを、その削除したノードの親ノードに接続するようにしている。

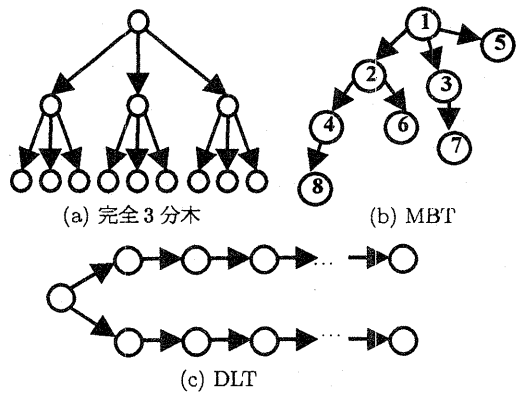


図 3: データ転送経路の構成

4 転送経路と転送完了時間

転送経路と転送完了時間について検証する。特に TCP のような 1 対 1 接続の場合は、データ転送経路の構成が転送完了時間に大きく影響する。また、インタプリタを採用する Java は、C++ に比べ低速であることから、実用に耐え得る転送完了時間が得られるかという問題がある。

4.1 種々の転送経路

ここでは大きく 3 種類の経路について考える [6]。一つは完全木 (Complete Tree) と呼んでいるもので、葉以外のすべてのノードが同じ数の子ノードを持つ木構造である (図 3(a))。各ノードの子の数が b であるとき、完全 b 分木と呼ぶことにする。深さ i の完全 b 分木では $\sum_{k=0}^i b^k$ であるが、実際のノード数が必ずしもこれを満たすとは限らない。この場合、不足する葉を削ることを認めることにする。

次の木構造は MBT (Minimum Broadcast Tree) [7][8] と呼ばれるものである。これは n 台のノードに対し $O(\log_2 n)$ の時間で転送を完了できる木構造であり、具体的には、 $1 \approx n$ のラベルを付けられた n 子のノードについて、ラベル i ($2 \leq i \leq n$) のノードをラベル $i - 2^{\lceil \log_2 i \rceil - 1}$ のラベルに接続することで構成できる (図 3(b))。MBT は単一データの転送では最も効率的で理想的な木構造であるとされている。

最後に直列木 (Linear Tree) について述べる。これは葉以外のどのノードも 1 つだけ子ノードを持つ木構造で、単純なパイプライン構成であると言える。

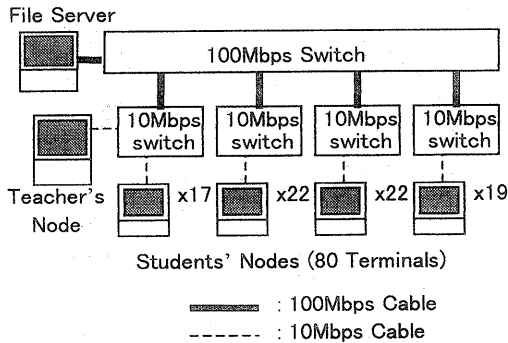


図 4: 実験で用いた端末教室の構成

しかし、通信路が半二重の場合、すなわち送受信を同時に行なえない場合は送信データと受信データが衝突する恐れがある。そこで、根のみ子ノードを2個とする木構造を提案した(図3(c))。これを2列直列木(Dual Linear Tree:DLT)と呼んでいる。これによりデータが2つの列を交互に流れるためデータの衝突を回避できる。なお、それぞれの列のノード数の差は高々1であるとする。

4.2 転送完了時間の測定

以上の経路構成を電子黒板システムに実装して転送完了時間の測定を行なった。測定を行なった木構造は完全2分木(BIN), 完全3分木(TRI), MBT, 2列直列木(DLT)であり, 640×480ピクセルのサイズの画像を転送し, すべての受講者端末で画像が表示されるまでの時間を測定している。また, 比較のため, C++版の電子黒板システムを用いて同一環境で測定を行なった。なお, Java版はJDK 1.0.2, C++版はMicrosoft Visual C++ 4.0を用いて開発している。また, Java版の実行はJIT(Just In Time)コンパイラを含むJDK 1.1.8のインタプリタを使用している。

実験環境を図4に示す。なお, OSはWindows NT Workstation 4.0である。CPUは教師端末がPentium 166MHz, 受講者端末はすべて486DX4 100MHzである。また, ネットワークアダプタは教師端末が3Com 3C905B-TX 100Mbps, 受講者端末はIntel EtherExpress PRO Adapter 10Mbps, スイッチングハブはMACNICA Grand Junction Fastswitch 10/100を用いている。

本実験では, 転送する画像データを複数のデータブロックに分割して連続転送を行なっている。端

末台数を20, 40, 78(C++版は75)台として測定を行なった結果を図5に示す。各グラフの横軸はデータブロック(分割)数を示す。転送完了時間の実用性について考えると, Java版は78台の場合で, 転送経路を完全2分木(BIN)とした場合に15分割で最短の転送完了時間(約4.8秒)を示している。C++版の75台における最短転送時間(BIN, 40分割で約1.5秒)に比べ約3倍の時間を要しているが, Java版では取り扱う画像フォーマットの違いのため, 同じ画像サイズでも実際の転送データサイズがC++版に比べ3倍となる。こうした点を考慮すると, JITコンパイラを用いることで十分実用に耐え得る転送完了時間が得られたと考えられる。

C++版の測定結果には各個色構成による転送完了時間の理論値を併記している。これらは以下の式で表される。 t 個のデータブロックを連続して転送する場合, 木の深さ i の完全 b 分木における転送完了時間は

$$C_B(tb + b(i-1)) + d(t) \quad (1)$$

で表される, また, ノード数が n の場合, MBTでは転送完了時間は

$$C_B t \lceil \log_2 n \rceil + d(t) \quad (2)$$

DLTでは

$$C_B [2t + \lceil (n-1)/2 \rceil - 1] + d(t) \quad (3)$$

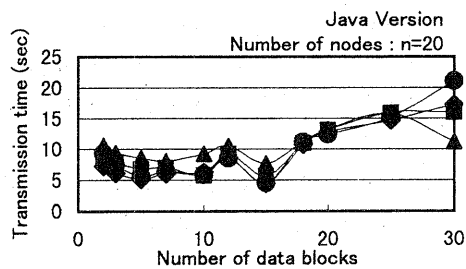
となる。ここに C_B は単一コネクシオンにおける1データブロックの転送時間を示し, この実験の場合, 元の画像のデータサイズ s , ハブおよびケーブルのバンド幅 w , ハブの遅延時間 d_H より

$$C_B = 8s/wt + d_H \quad (4)$$

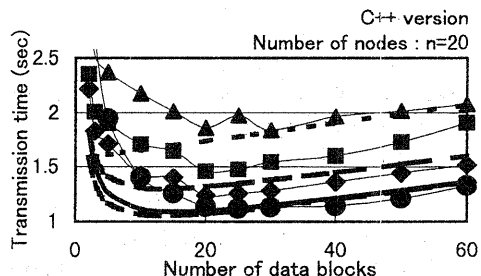
と推定される。本実験においては $w = 10^7[\text{sec}]$, $d_H = 3 \times 10^{-5}[\text{sec}]$ となる。また, $d(t)$ は転送経路に依存しない, アプリケーション側での処理時間である。これは主に教師側ノードにおける, 画像データを分割するための前処理に必要な時間であると考えられ, 分割されたデータブロック数 t に対して

$$d(t) = a + ct \quad (5)$$

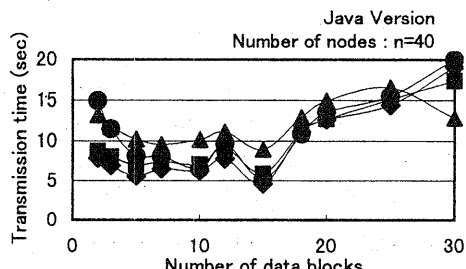
の形となることが予想される。ここに, a, c はパソコンの性能やアプリケーションの処理に依存する任意定数であり, $d(t)$ を無視した理論値と実測値との



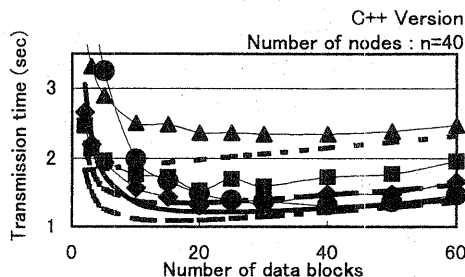
(a) Java 版, 20 台



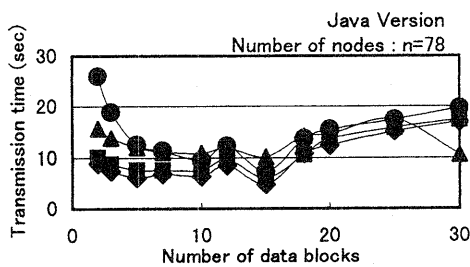
(b) C++ 版, 20 台



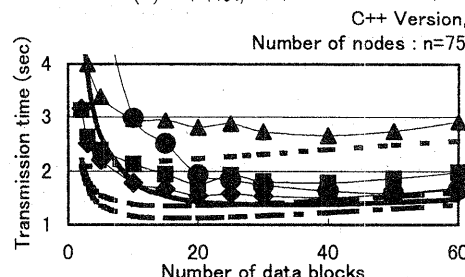
(c) Java 版, 40 台



(d) C++ 版, 40 台



(e) Java 版, 78 台



(f) C++ 版, 75 台

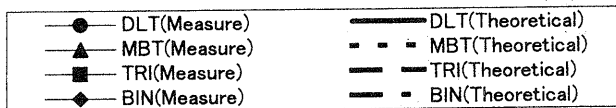


図 5: データ分割数と転送完了時間の関係

差分から $a \approx 0.3400, c \approx 8.269 \times 10^{-3}$ と推定している。

Java 版と C++ 版を比較すると、C++ 版の転送完了時間の実装値はほぼ理論に近い傾向を示しているのに対し、Java 版の実測値の傾向は C++ 版および理論値と異なり、分割数を増やした場合に転送時間は一様な増加ではなく増減を繰り返す傾向や、Java 版では 30 分割で MBT が他構造に対して最短の転送時間を示す一方、C++ 版では多分割数で最短の転送完了時間を示した DLT が最も遅くなるなど、C++ 版と異なる傾向が見られた。この原因の解明については今後の検討課題とするが、転送データサイズと Java 版、C++ 版の違い以外はほぼ同一

条件で実験を行なっているため、この傾向の違いは Java VM の内部構成に起因するものと推測される。

5 おわりに

分散環境を利用した TCP による電子黒板システムとその Java による試作について述べた。Java を用いて開発することにより、OS などを問わずあらゆる環境で利用でき、移植性が高められる他異なるシステム間を結ぶ遠隔授業にも対応できる。Java はインタプリタを用いるため速度が問題となるが、今回の測定で見た限りでは、JIT コンパイラを利用することでほぼ実用に耐え得る速度が得られたと考

えられる。

受講者の端末利用状況を監視して必要により転送経路の組み替えを行なうオブザーバーの導入により、黒板起動中の受講者のログイン、ログアウトなどにも柔軟に対応できるようになった。今後は、受講者の途中ログアウトによる経路切断に対するより最適な経路組み替えの手法の確立が課題となる。また、現在は指定の経路構成にしたがって新規参加端末の経路への追加を行なっているが、今後は転送完了時間面での最適化を考慮した手法、例えば、接続先の回線の混み具合などを判断してより最適な接続先を判断する方式などを検討している。

参考文献

- [1] John Bazik, "XMN - An X Protocol Multiplexor,"
<http://www.cs.brown.edu/software/xmn/home.html>.
- [2] T. Kawai, M. Ikeda, M. Okada, "Point-to-Multipoint Communication Protocol on Window-Based Network Presentation System", ICICE TRANS. INF. & SYST., Vol E80-D, No 2, 1997
- [3] T. Yamanoue, M. Shimizu, and T. Fujiki, "Development of an Electronic Chalkboard for a Large Classroom by Parallel Programming and Its Application to English Classes," Proceedings APITITE94, vol.2, pp.651-656, 1994.
- [4] 平原貴行, 山根真人, 山之上卓, 安在弘幸, 澤田崇, 堤 宏智, "TCP/IP プロトコルを利用したパソコンネットワークのための電子黒板システム," 分散システム運用技術シンポジウム'98 論文集, pp.79-85, 1998.
- [5] T. Hirahara, H. Anzai, T. Yamanoue, M. Yamane, T. Sawada, "Development of An Electronic Chalkboard System Through Parallel Programming," The Second Annual Teaching in the Community Colleges Online Conference, http://leahi.kcc.hawaii.edu/org/tcc_conf97/pres/yamanoue.html, 1997.
- [6] 平原貴行, 山之上卓, 山根真人, 安在弘幸, "電子黒板システムにおけるデータ転送経路構成及び配信方法の比較", 信学論 (DI), Vol.J82-D-I, No.8, 1999.
- [7] A. Farley, "Minimum broadcast network," Network, vol.9, pp.313-332, 1979.
- [8] A. Shastri, "Broadcasting in General Network I:Trees," Lecture Notes in Computer Science, Vol.959, pp.482-489, 1995.
- [9] Stelzer, G., "Versatile 10/100 Megabit/sec Ethernet connection combines 10-base T and 100-base TX performance," Elektronik (Germany), vol.45, no.13, pp.54, 56, 58-60, 1996.
- [10] van Gelder, T., "The aims of 100 Base-T," Elektronica (Netherlands), vol.42, no.16, 35-7, 1994.