

デスクトップ会議システム Open GroupMedia における アプリケーション共有機能の高速化*

6M-1

藤野 剛, 阿久津 正明, 村永 哲郎, 今井 徹†
(株) 東芝 研究開発センター 情報・通信システム研究所 第二研究所‡

1 はじめに

Open GroupMedia[1] は、音声、画像通信、共有ウィンドウからなるデスクトップ会議システムである。ユーザが普段利用しているエディタ、描画ツール、画像表示ツールなどのアプリケーションを全く変更することなく、画面上で共有して会議を行うことができる。このために疑似サーバ方式 [2] を採用しているが、描画ツールなど画面の変更要求の多いアプリケーションを使用するためには、描画速度が不十分であることがプロトタイプの実用により明らかになった。

本稿では、描画速度を向上させるために行った疑似サーバの高速化手法と、その性能評価の結果を述べる。

2 プロトタイプ

本研究ではプロトタイプを早期に作成し、社内にてその評価を行ってきた。環境は UNIX ワークステーション上の X ウィンドウである。

アプリケーションを共有する方法としては疑似サーバ方式を採用した。疑似サーバ方式では以下のようにして共有を実現する。

- アプリケーションと X サーバとの間に入りパケットを受け取る
- パケットに含まれるリソース ID を変換する
- パケットを複数の X サーバに分配する

疑似サーバはアプリケーションに対しては X サーバのように振舞い、X サーバからは一つのアプリケーションに見える。また、X ウィンドウシステムでは各 X サーバ毎にリソースに固有の ID を持たせて情報の管理を行っているのでパケットをそのまま分配するのではなく、リソースの ID の変

換をしてから送る必要がある。これで、X ウィンドウ上の個人用アプリケーションを変更せずに複数のディスプレイで利用する事が可能になる。

プロトタイプでは疑似サーバ方式の中でも図 1 のような構成をとっていた。

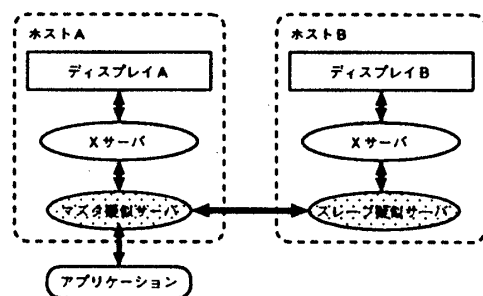


図1 分散疑似サーバ方式

これは各 X サーバ毎に疑似サーバを対応させる事で、各 X サーバ固有の情報をそれぞれの疑似サーバだけで保持させるためである。マスター疑似サーバは自分の情報を全てのスレーブ疑似サーバに送り、その情報をそれぞれのスレーブ疑似サーバが自分の担当している X サーバ用に変換して送信する。X サーバからの情報はマスタ用に変換してからマスタに送信する。

作成したプロトタイプは社内数箇所でも利用してもらいアンケート調査を行った。その結果おおむね良好な反応が得られたが、最も問題とされたのはその描画速度であった。とくに画像情報を多用するような描画ツール、画像表示ツールなどで描画の遅さが問題となった。

3 疑似サーバの高速化

描画速度の向上を目指してアプリケーション共有の実装方式を次のように変更した。

- スレーブ疑似サーバの廃止
- マスター疑似サーバから直接全ての X サーバにパケットを送信する構成とした (図2)。これによりクライアント側でのスレーブ疑似サーバと X サーバ間のパケット転送やコンテキストスイッチが無くなり、高速化が見込まれる。

* Acceleration of application sharing server in the desktop conference system "Open GroupMedia"

† G. Fujino, M. Akutsu, T. Muranaga, T. Imai

‡ Communication & Information Systems Research Lab. II, R & D Center, TOSHIBA Corp.

しかし、リソース ID の変換をマスタ疑似サーバで集中的に行うため、マスタ疑似サーバがボトルネックになる恐れがある。性能評価により効果の大きさを測定した。

● パケット送受信の効率化

プロトタイプでは送受信の際一つの X リクエスト、イベントなどを複数のパケットとして送信および読み込みをするなど一度に取り扱うのパケット長が小さく、送信回数が多かった。このためバッファリングを行い、パケット長を大きくして送受信回数を減らした。

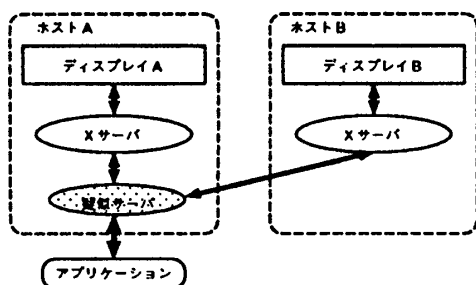


図2 集中疑似サーバ方式

4 性能評価

描画ツール tgif、画像表示ツール xv を使い、それぞれのアプリケーションが全てのディスプレイで起動 (表示) されるまでの時間を測定した。測定環境は SPARCClassic、Solaris2.3、X Window、Ethernet である。

図3において横軸が接続したディスプレイの数、縦軸が起動するまでの時間である。また、横軸0の値はアプリケーションを共有せずに起動する時の所要時間である。それぞれのアプリケーションについて、起動までの所要時間は約半分に短縮された。

● 一つのディスプレイだけで起動した場合は、分散疑似サーバ方式と集中疑似サーバ方式は同一の構成となる。この時 tgif では約二分の一、xv では約五分の一の所要時間である。これは前述した、パケット送受信の効率化の効果であると考えられる。

入出力に関するシステムコール (read, write, getmsg, putmsg) の回数を測定してみると、図4のようになった。集中疑似サーバ方式と分散疑似サーバ方式で、システムコールの回数に4倍以上の差があることがわかる。

● ディスプレイの数が増加しても (プロトタイプではスレーブ疑似サーバが増える)、分散疑似サーバ方式の方が集中疑似サーバ方式より高速である。

これは、スレーブ疑似サーバの廃止とパケットの送受信の効率化による寄与が大きく、リソース

ID 変換の集中による速度低下を上回ったためである。結果として2倍から4倍の速度向上が確認できた。

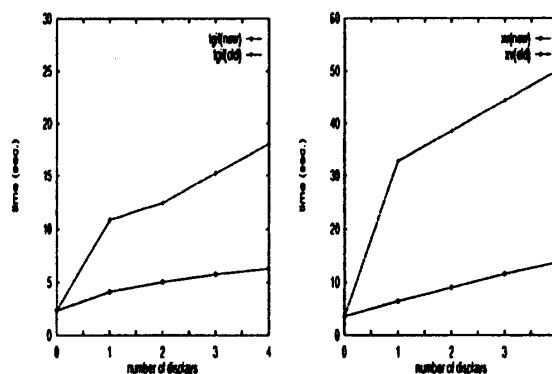


図3 tgif, xv の起動に要した時間

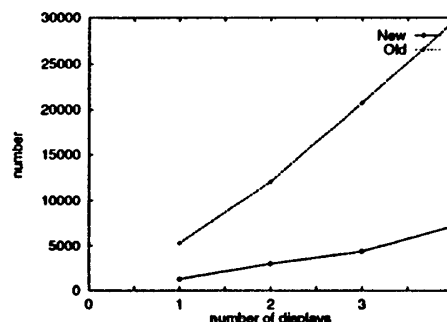


図4 tgif 起動時に呼ばれた入出力システムコール

5 まとめ

我々はデスクトップ会議システム Open Group-Media のアプリケーション共有サーバを再設計し、描画速度を従来の2倍から5倍に向上させた。運用評価を続け、実用的なシステムとするための改善を進めていきたい。

参考文献

- [1] 村永 哲郎, 他. Open GroupMedia — 個人用アプリケーションを共有するグループウェア・フレームワーク, 情報処理学会グループウェア研究会, Vol.GW-4, No.18, pp.121-128, October 1993.
- [2] Abdel-Wahab, H.M. and Fcitt, M.A., XTV: a Framework for Sharing X Window Clients in Remote Synchronous Collaboration, Proceedings of the IEEE TriComm'91: Communications for Distributed Applications and Systems, pp.159-167, 1991.