

複数ディスプレイを利用した高解像度ディスプレイ環境の構築

千葉 豪† 柴田 義孝†

† 岩手県立大学ソフトウェア情報学研究科

1 はじめに

近年、ブロードバンドネットワークサービスの普及により IP ネットワークを利用した高精細映像の配信や、三次元仮想空間における大容量データ通信、高解像度の地図画像の利用が実現されてきている。また、高性能ビデオカードを搭載した複数の計算機と複数表示装置を組み合わせたレンダリングクラスタの構築がソフトウェアレベルで可能となり、これまでの可視化、VR 分野において利用されてきた専用ワークステーションやマルチスクリーンを利用せずとも、汎用の計算機、ディスプレイ環境を応用することで容易に高精細で大画面かつ高臨場感を持つディスプレイ環境が実現できるようになった。

そこで本研究では、複数 PC クラスタおよび液晶ディスプレイを組み合わせ、これらを超高速ネットワーク上で相互接続することにより、多様なアプリケーションの出力結果を容易にストリームとして配信し、実質的にクラスタサイズに比例した解像度を達成できるプレゼンテーションシステムを提案しそのプロトタイプシステムを構築する。

2 タイルドディスプレイシステム

2.1 タイルドディスプレイ

本研究で用いられるタイルドディスプレイとは複数のディスプレイをタイル状に配置することで一つの大型かつ高精細なディスプレイ環境を実現するものである。これまで、VR や可視化分野においては特殊なハードウェアや複数プロジェクターを利用することで没入感のある映像や、高解像度な映像の表示を実現してきた。しかし、このようなハードウェアを利用するにはコストがかかる、システム構築が容易ではないといった問題があり、近年ではソフトウェアレベルでの分散レンダリングシステムが多数提案されている。

2.2 関連研究

DMX(Distributed Multihead X) ではネットワークを介して複数の X Window サーバを相互接続することで仮想的に単一の高解像度デスクトップ環境を実現するもので Linux をベースとしたシステムである [1]。

また、Chromium では OpenGL のコマンドをダンプし SPU(Stream Processing Unit) を用いたストリームベースの分散レンダリングライブラリを提供している [2]。

近年では、イリノイ大学が開発した SAGE(Scalable Adaptive Graphics Environment) ではピクセルデータをキャプチャ仮想フレームバッファを利用し、擬似的に高解像度なディスプレイ環境を実現している [3]。

しかし、これら既存のシステムでは複数アプリケーションの同時表示に対応していない、もしくはライブラリとしてアプリケーションに組み込む必要がありアプリケーション独立なシステムを実現できていないといった問題がある。

3 提案システム

3.1 目的

そこで本研究では、複数 PC クラスタと液晶ディスプレイを組み合わせ、これらを高速ネットワーク上で相互接続することにより、容易にアプリケーションの出力結果をストリームとして配信し、実質的にクラスタサイズに比例した高解像度ディスプレイ環境を達成できるプレゼンテーションシステムを開発する。

3.2 システム構成

本システムのネットワーク構成は図 1 のようになっており各拠点には出力装置としてタイルドディスプレイ、タイルドディスプレイへの描画処理を行う Rendering Cluster、各タイルの状態を管理する Tile Manager が存在しており、ユーザは Application Host 上でビデオ通信、CVE、イメージビューアーといったアプリケーションを動作させる。Application Host, Tile Manager, Rendering Cluster は超高速ネットワークを介し相互接続され Application Host 上でキャプチャされたピクセルデータを Rendering Cluster へストリームとして送信することで各アプリケーションの出力結果を任意の地点のタイルドディスプレイ上に描画することを可能とする。

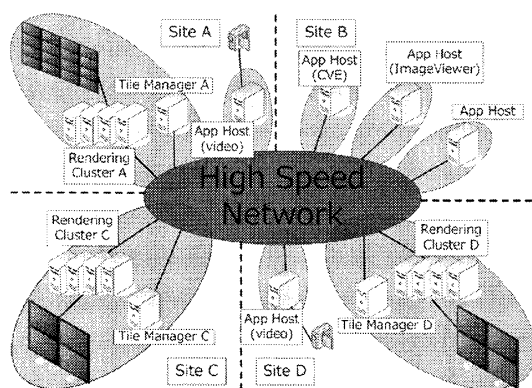


図 1 システム構成

3.3 アーキテクチャ

本システムのアーキテクチャを図 2 に示す。出力装置として前述したタイルドディスプレイを用いることを想定しており、Application Host, Tile Manager, Rendering Cluster といった 3 つの構成要素から成る。

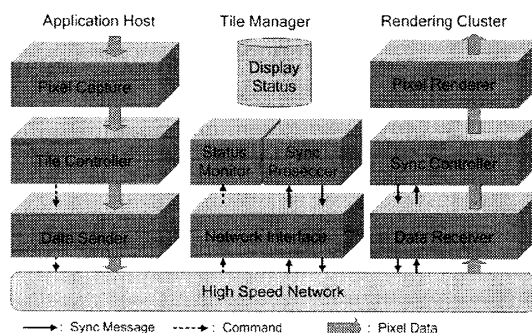


図 2 システムアーキテクチャ

Application Host では仮想ディスプレイドライバによるピクセルデータのキャプチャ、ピクセルストリームの配信や Tile Manager とのメッセージ通信が行われる。Tile Manager では内部データとしてタイルサイズ、セッション情報等を保持し、Application Host, Rendering Cluster とのメッセージ通信が行われる。Rendering Cluster ではピクセルストリームを受信しリサイズ等の処理を加えた後担当タイルへの描画処理を行う。

High-Resolution Display Environment with Multiple Displays

† Go Chiba(go@sb.soft.iwate-pu.ac.jp)

† Yoshitaka Shibata(shibata@iwate-pu.ac.jp)

Graduate School of Software and Information Science, Iwate Prefectural University (†)

3.4 システム機能

3.4.1 仮想ディスプレイドライバ

本システムでは仮想ディスプレイドライバの概念を導入し、仮想ディスプレイ機能によりアプリケーションと独立したピクセルデータのキャプチャを実現する。ドライバを適用することにより図3に示すように仮想ディスプレイがシステムに認識され、ドライバは仮想ディスプレイ上のピクセルデータを毎フレームキャプチャする。ユーザはそのディスプレイ上にアプリケーションの出力ウィンドウを移動させることでタイルドディスプレイ上へのアプリケーションの描画を可能とし、仮想ディスプレイ上でドライバによってキャプチャされたピクセルデータは共有メモリを経て Pixel Capture モジュールにて利用される。

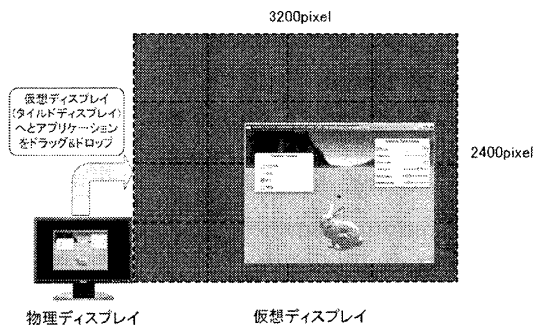


図3 仮想ディスプレイ

本手法ではドライバとして組み込み物理的に存在するディスプレイとは別の描画命令を取得することでアプリケーションに組み込む必要のないアプリケーション独立なキャプチャが可能となる。また、これによりライブラリとして組み込むことが困難な Web ブラウザ、Microsoft PowerPoint のようなネイティブなアプリケーションへの適用が容易になると考えられる。

3.4.2 イメージ分割処理

本システムでは Application Host でキャプチャされたピクセルデータを転送元でのサイズを保持したまま、各 Rendering Cluster へと転送される。その後、ピクセルデータを受け取った各クラスタでは担当するタイルのサイズに合わせるため分割、リサイズといった処理を行う。これにより、クライアント側での CPU 負荷の増大を防ぐ。

3.4.3 タイル間表示同期

ディスプレイ間の同期はメッセージのやりとりによって行われる。まず、Application Host はキャプチャしたピクセルデータをストリームとして各クラスタへ送信を開始する。各クラスタはそれぞれピクセルデータの受信が完了後描画の準備ができたことを Tile Manager へと通知する。Tile Manager では全てのクラスタからの通知を受け取ったら、各クラスタに更新メッセージを送信し、描画処理を開始するよう命じる。その後、各クラスタは先ほど受け取ったピクセルデータを用い担当するタイルサイズへとリサイズ等を行ったうえで描画処理を行う。

3.4.4 メッセージ通信

本システムではタイルドディスプレイの起動、停止をはじめとする制御をメッセージ通信によって行い、表1に示すメッセージ群が Application Host, Tile Manager, Rendering Cluster 間でやりとりされる。

MSG.LOGIN W H	Tile Manager へのログイン、サイズの通知
MSG.LOGOUT	Tile Manager からのログアウト
MSG.BOOT	Rendering Cluster の起動
MSG.DOWN	Rendering Cluster の停止
MSG.SYNC	タイル同期用メッセージ
MSG.UPDATE	タイル更新用メッセージ
MSG.MOVE X Y	タイルドディスプレイの描画領域を X Y に move
MSG.RESIZE W H	タイルドディスプレイの描画領域を W H に resize
MSG.**_OK	メッセージの成功を通知
MSG.**_NG	メッセージの失敗を通知

表1 制御メッセージ群

3.4.5 マルチセッション表示

本システムではピクセルデータのストリーミングにおいて IP マルチキャストの利用を可能とし、各 Application Host を一つのセッションとして管理する。また、先のアーキテクチャで述べた Application Host からのメッセージ機能を用い動的にセッションへの参加、離脱をすることで各セッション単位での柔軟な表示ディスプレイの割り当てを実現する。これにより、システムにおけるセッション管理が容易になり、タイルの枚数の増加させた際のピクセルストリームにおけるネットワークの利用帯域を軽減できる。

4 評価

本システムの評価として Application Host, Rendering Cluster 間でピクセルストリームを送信した際の各処理における処理時間の計測を行った。今回、評価には Application Host, Rendering Cluster とともに Dell Precision 380 を用い、OS には Windows XP SP2、CPU に Intel D CPU 3.20GHz、3.19GHz、メモリは 2.00GB RAM を利用した。

また、実際のピクセルデータは表2に示すものをそれぞれ RGBA の 1 ピクセルあたり 32bit のデータを転送した。

解像度	規格	縦横比	データ量 (bits)	データ比 (%)
800x600	SVGA	4:3	15,360,000	100.00
1024x768	XGA	4:3	25,165,824	163.84
1280x1024	SXGA	5:4	41,943,040	273.07
1600x1200	UXGA	4:3	61,440,000	400.00
2048x1536	QXGA	4:3	100,663,296	655.36
2560x2048	QSXGA	5:4	167,772,160	1092.27
3200x2400	QUXGA	4:3	245,760,000	1600.00

表2 利用データ一覧

実験では、Application Host, Rendering Cluster を 1Gbps のイーサネットにて相互に接続、本実験では Tile Manager をネットワークに参加させず TCP を用いたピクセル転送のみを行った場合の処理時間を計測した。その結果を図4に示す。

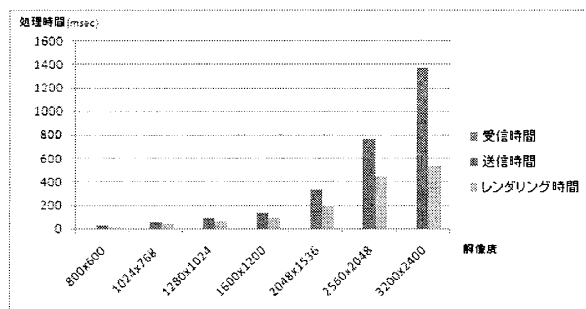


図4 計測結果

5 まとめ

本研究では複数 PC クラスタおよび液晶ディスプレイを組み合わせ、超高速ネットワーク上で相互接続することによりアプリケーションの出力結果を容易にストリームとして配信可能なプレゼンテーションシステムを提案、開発しその評価としてピクセルストリーム配信における処理時間の計測を行った。今後は加えてピクセルデータ圧縮を適用した場合の処理時間、CPU 利用率等を計測し圧縮処理によるトレードオフについて調査、検討していく。

参考文献

- [1] Distributed Multihead X Project. <http://dmx.sourceforge.net>.
- [2] Greg Humphreys, et al. Chromium: A Stream-Processing Framework for Interactive Rendering on Clusters, Proc. ACM SIGGRAPH 2002, pp. 693-702, July 2002.
- [3] Byungil Jeong, et al. High-Performance Dynamic Graphics Streaming for Scalable Adaptive Graphics Environment, Proc. ACM/IEEE SC 2006 Conference, November 2006.