

大規模並列計算機のための遠隔分散型画像表示システム

小畑 正 貴[†] 江草 俊 文[†] 中條 拓 伯^{††}

並列計算機での画像表示においては、その高速計算機能に見合うだけの画像表示能力が必要とされる。このため我々は、大規模並列計算機に対応できる遠隔分散型の画像表示システムを開発した。本システムの最大の特徴は、複数のプロセッサから複数のシリアルリンクで表示データを転送する点にある。これにより、(1) 高品位動画に対応できる高速データ転送能力、(2) 大規模システムでの設置の柔軟性、(3) プロセッサ数、解像度、ディスプレイ台数などに対するスケーラビリティ、といった利点を持っている。またプログラマブル論理デバイスの利用により、種々の表示装置や画面分割に対応することができる。本論文では、表示システムのハードウェア構成について述べ、ワークステーションクラスと本システムによるハイビジョン画像表示実験結果を示す。

A Remote Distributed Image Display System for Large Scale Parallel Computers

MASAKI KOHATA,[†] TOSHIFUMI EGUSA[†] and HIRONORI NAKAJO^{††}

For image output subsystems of high performance parallel computers, a high level image output capacity is required to correspond to the computing capacity. Thus a remote distributed image output system has been developed for large scale parallel computers. Transferring image data from processors is conducted in the system via high speed serial links in parallel, giving the following advantages: (1) high speed data transfer for high definition moving pictures, (2) flexibility of layout and connection for large size systems, (3) scalability corresponding to the number of processors, display size, and display units. By using programmable logic devices in controlling image display, the system can support various types of resolution in displaying and image data distribution modes. The hardware configuration, connection to a workstation cluster system, and experimental results for a HDTV graphic display are also described.

1. はじめに

並列計算機の重要な応用分野である科学技術計算においては、計算結果の可視化のための画像表示装置が必要である。特に超並列計算機の実現により、計算しながら実時間で可視化することが可能になってくると、高速計算に見合う表示の高速性が要求される。

並列計算機における画像表示システムの形態を図1に示す。多くの並列計算機における画像表示システムは(a)の形態であり分割処理した部分画像を1ノードに集めてそこで表示する。この方式では、高画質画像や動画のようにデータ量が多くなると、表示データの転送が並列計算のためのノード間通信を圧迫したり、

表示ノードに通信が集中したりする問題点が生じる。

(b)は画面を分割してフレームメモリを分散させる方式であり、特にグラフィックス応用に重点を置いたマシンで使われている¹⁾。1画面を複数プロセッサが同時にアクセスできるため、競合が軽減される。しかし分散配置されたフレームメモリからビデオ出力回路(D/A変換器など)へのデータ転送に高速性(数百MB/s)が要求されることから、複数ラックに及ぶような大規模システムへの拡張は実装上困難である。

(c)は分割したフレームメモリを遠隔に置く方式であり、動画などに対応できる高速表示能力と、大規模マシンにおける接続の容易性に対応できる画像表示システムとして有効である。この形の画像表示システムとして、AP1000の画像表示システムがあげられる²⁾。AP1000ではフレームメモリへの表示データの転送をリング型のネットワークによって行っている。

(c)では特に、プロセッサとフレームメモリ間の通信リンクに対して高速性や実装容易性が要求されるが、我々が本論文で提案するシステムではこれを複数のシ

[†] 岡山理科大学工学部情報工学科

Department of Information and Computer Engineering,
Faculty of Engineering, Okayama University of Science

^{††} 神戸大学工学部情報知能工学科

Department of Computer and Systems Engineering,
Faculty of Engineering, Kobe University

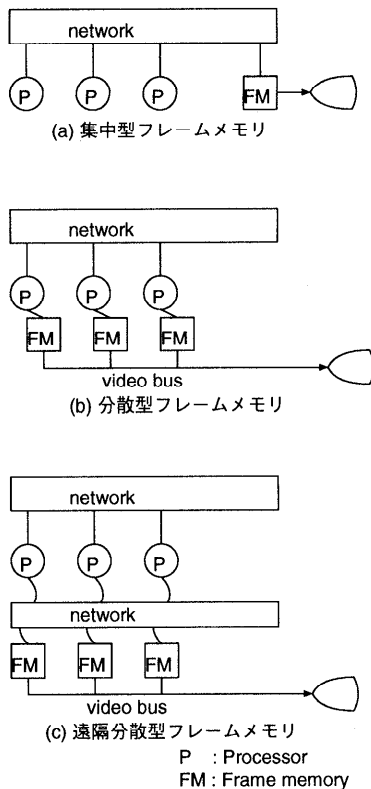


図1 並列計算機の画像表示システム

Fig. 1 Display systems of parallel computers.

リアルリンクによって行う。本システムは、分散共有メモリ型超並列計算機 JUMP-1 の画像表示サブシステムとして開発を始めたものであるが^{3)~6)}、他の並列システムにも対応できる柔軟性を持っている。

以下、シリアルリンクによる画像入出力システムの構成と特徴、試作した画像表示装置のハードウェア構成、実現方法、ワークステーションクラスタとの接続と表示実験について述べる。

2. 複数シリアルリンク接続による画像入出力システム

高い処理能力を持つ並列計算機に対応する画像入出力システムには以下のような要求があげられる。

(1) 高速な画像データ転送能力

並列計算機と画像入出力システムの間には、高解像度、動画の入出力に対応できるデータ転送能力が必要とされる。たとえば、ハイビジョン規格では、 $1,920 \times 1,035$ ドット、 30 フレーム/秒の動画表示での転送レートは約 180 MB/s となる。最近ではパーソナルコンピュータでも $2,048 \times 1,536$ ドット、ノンインタレース表示が可能であり、今後、さらに高い転送レートが

必要となる。

(2) 画像データ転送ネットワークの分離

I/O ノードにデータを集めて表示する従来方式では、大量の画像データが並列計算機内ネットワークを流れることになり、本来の計算のための通信が阻害されることになる。このため、画像入出力用ネットワークを分離する必要がある。

(3) 並列計算機のスケラビリティへの対応

プロセッサ数が多数になってくると、すべてのプロセッサを画像入出力ネットワークに接続することはできない。この際、画像入出力ポートを持つプロセッサまでは並列計算機内ネットワークを使った通信が必要となるが、この通信における局所性を確保するためには画像データの入出力ポートを持つノードを数多く分散して配置する必要がある。たとえば、 $2,048$ ノードのシステムで 16 ノードごとにフレームバッファに接続すると、 128 のノードに画像データの入出力ポートが必要となる。この際、接続数が接続する並列計算機に応じて柔軟に対応できること、画面の分割が負荷分散方法に柔軟に対応できることが要求される。

(4) カメラやディスプレイの解像度や台数に対するスケラビリティ

マルチスクリーンによる大画面表示や立体画像表示、マルチカメラによる立体画像入力、などの応用においては種々の解像度を持つ多数の画像入出力装置を接続することになる。この場合、種々のシステム構成や各機器で要求される通信容量に対して柔軟に対応できなければならない。

(5) 空間的に大きなシステムへの設置・接続の容易性

大規模な並列システムにおいては空間的規模も大きなものとなり、実装面での設置や配線の容易さも実用上重要である。

提案するシステムは、画面分割による画像データの並列入出力を多数のシリアルリンク接続で行うことにより、上記の要求に対応しようとするものである。ディスプレイやカメラを直接接続するフレームバッファは複数のフレームメモリユニットからなる。各フレームメモリユニットは分割画面の1部分を受け持ち、複数のフレームメモリユニットで1枚の画面を構成する。各フレームメモリユニットは1~数本のシリアルリンクポートを持ち、並列計算機の複数ノードと接続する。

要求(1)に対しては、1リンクあたり 100 Mbps 以上のシリアルリンクを複数使えば必要な転送レートが確保できる。(2)に対しては、並列計算機側では、シリアルリンクポートを持つノードまでは主ネットワークを使うことになるが、データ分割が適当であれば近

接通信が主体となり、主ネットワークへの負担は軽減される。(3)に対しては、フレームメモリの分割ユニット数の増減で対応する。画像データ転送における最大並列度は1台のフレームバッファから引き出せるシリアルリンク数に依存することになる。画面分割による入出力では分割画面間での競合はないので、この点についてはハードウェア実装上の問題が大きい。(4)に対しては、各機器に対してフレームバッファを用意し、画面の位置と対応するノードにそれぞれ接続することで対応できる。たとえば、立体表示の場合に、同じ部分領域の左右両眼に対応するリンクを近接するノードに接続しながら両画面を並列計算機全体に分散させるような接続ができる。また、フレームバッファ装置側で種々の解像度や画面分割方法に対しては、ディスプレイやカメラの同期制御回路やフレームメモリの読み書き制御回路にプログラマブル論理デバイスを用いることで対応する。(5)に対しては、シリアルリンクは、室内程度の範囲でツイストペアケーブルにより低エラーレートで接続できるため設置や接続が容易であり、汎用LANと比べて通信プロトコルによるオーバーヘッドも小さい。

以下では提案するシステムの実験システムを設計・試作し、実装までを含めて実現可能性や有効性を評価する。

3. 試作機のハードウェア構成

3.1 全体構成

表1に画像表示システムの仕様を、図2に全体構成を示す。ハイビジョン規格で動画像の表示が可能となるように仕様を決めている。ハイビジョン1画面分のフレームメモリは16分割され、表示データが16ノードから並列に転送される。

クラスタとフレームメモリ間の接続には STAFF-Link (Serial Transparent Asynchronous First-in First-out Link)^{4),5)}と呼ばれる高速なシリアルリンクを用いる。STAFF-LinkはJUMP-1のI/Oシステムの研究において開発されたシリアルリンク接続方式で、シリアル転送にはAMD社のTAXIチップ⁷⁾を用いている。プロセッサ側インタフェースは一般のFIFOメモリの信号とほぼ同じなので、種々のプロセッサや汎用バスへの接続が容易となっている。

ハイビジョン規格の画面はドット周波数74.25MHzで、1920×1035ドットの構成である。今回は回路簡略化のため画面構成を1920×1024(2Mドット)としている。各ドット24ビットなので、1画面の情報量は6MBとなり、毎秒30フレームで180MB/sの情報

表1 画像表示システムの仕様
Table 1 Hardware specification.

方式	ハイビジョン
画面構成	1920 × 1024 (規格は1920 × 1035) 24 bit/pixel (1画面あたり6MB)
ドット周波数	74.25 MHz
水平周波数	33.75 KHz
垂直周波数	60 Hz (30 フレーム/秒)
メモリ容量	6 MB × 2 バンク (ダブルバッファ)
分割数	16
転送速度	17.5 MB/s シリアルリンク × 16

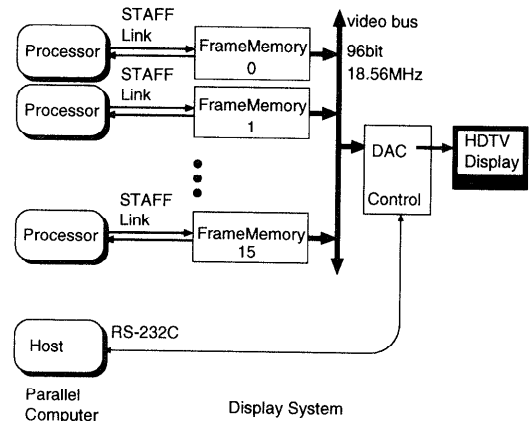


図2 画像表示システムの構成
Fig. 2 System configuration.

量となる。175 MHzのTAXIチップで最大17.5 MB/sの転送能力があり、これを16本使うと280 MB/sとなるので、表示速度に見合うだけのデータ転送幅が確保できる。

16枚のフレームメモリボードからディスプレイへのデータ転送はビデオバスによる。ビデオバスはVMEバスを拡張して利用する。ビデオバスのデータ幅は96ビットで、4ドット分を同時に転送する。

3.2 フレームメモリ

表示とデータ転送を並列に行うため、共有フレームメモリはダブルバッファ構成とし、一方が表示中に、他方に次の表示データをクラスタから転送する。フレームメモリボードの構成を図3に示す。

制御回路は2個のFPGA (Field Programmable Gate Array)により実現される。1個は並列計算機からのデータ転送を制御し、他方はビデオバスへのデータ転送を制御する。

画面分割は図4の6種類をサポートする。並列処理においては、負荷分散や通信量の最適化の点から、データの分割方法をブロック分割やサイクリック分割など問題によって変えることがある。並列計算機上で

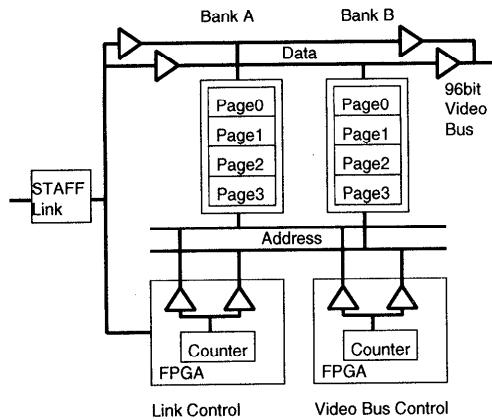


図3 フレームメモリボード

Fig. 3 Configuration of frame memory board.

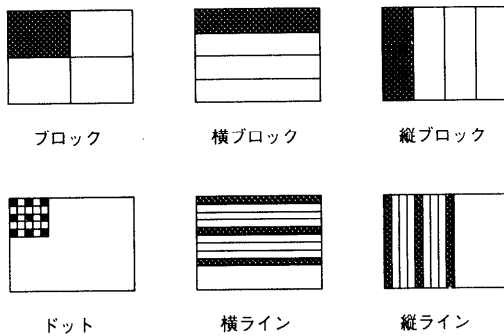


図4 分割モード

Fig. 4 Distribution modes.

の画面分割とフレームメモリ上での分割が一致しないと、並列計算機内のネットワークを使った表示データ転送が必要となる。これを少なくするためには、両者はできるだけ一致している方がよい。

分割方法およびフレームメモリへの割当ては静的で、プログラム実行前に分割モードを切り替えるものとする。画面分割を変更すると、各フレームメモリからのデータの読み出し順序の変更が必要になるが、これはビデオデータ読み出し側のFPGAの再コンフィギュレーションによって行う。

プロセッサからのデータ転送は、1画面分のデータを適当なドット数単位で分割したパケットによって行う。転送プロトコルもFPGAによって可変となっているので、種々の並列計算機に柔軟に対応できる。特に分散共有メモリマシンに対しては、本フレームメモリを共有空間にマッピングし、転送プロトコルをメモリ間での転送プロトコルと合わせることによって、フレームメモリを共有メモリの延長として取り扱うことも可能となる。この点に関しては文献4), 5)で提案

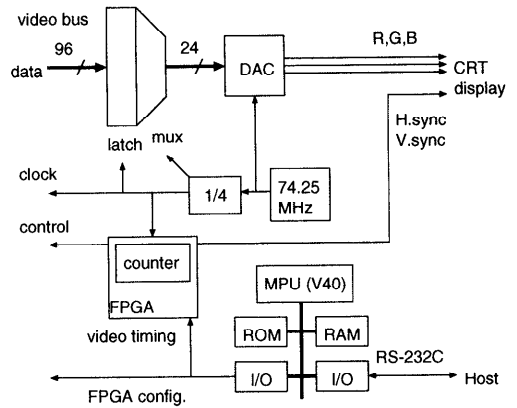


図5 画像表示コントロールボード

Fig. 5 Configuration of control board.

している。

3.3 コントロールボード

コントロールボードの構成を図5に示す。ビデオバスからのデータをマルチプレクサを通してDA変換器(DAC)に入力し、モニタに接続する。同期信号等のタイミングはFPGAで作成する。FPGAのコンフィギュレーションによって、ハイビジョン以外にもNTSCやVGAなどのディスプレイにも表示が可能である。

FPGAのコンフィギュレーションや全体の制御を行うためマイクロプロセッサを使用する。このマイクロプロセッサをRS-232Cでホスト(コンソール)と接続し、コンフィギュレーションデータのダウンロードや表示モードの切替えなどを行うようにする。

実装は、VMEダブルハイトのボードを使い、16枚のフレームメモリボードとコントロールボードとを標準VMEラックに収める。STAFF-Linkの部分はドータボードとして実装する。

3.4 データ転送と表示のタイミング

フレームメモリはダブルバッファになっているが、この切替え方式として以下の2つが考えられる。

(1) 画面切替え

片方を表示バッファ、他方を転送バッファとし、1画面更新ごとに切り替える。この方式では前の画面の一部だけを書き換えるということができないので、全画面データを転送しなければならない。次の画面の作成中にも表示画面は変化せず、切替え信号によって全画面は同時に新しい画面と切り替わる。データ転送後終了コマンドを送ることにより、フレームメモリのバッファ切替えを行う。データ転送時間は各ボードによって違ってくるため、切替えは全ボードが

終了コマンドを受けてからとなる。

(2) 画面分割

1画面を2分割し、一方の表示期間中に他方に転送する。分割方法としては、上下分割、奇数ライン/偶数ライン分割、インタレースにおけるフィールド分割などが考えられる。バッファ切替えはビデオ信号に同期して定期的に行われ、画素値の変更はすぐ表示に現れる。転送は変更画素に対してのみ行えばよい。ただし、実際は転送はブロック単位であるので、その画素を含むブロック転送となる。

また、データ転送のタイミングとしてユーザ指定とタイマ割込みが考えられる。

- (1) ユーザ指定：プログラム内にフレームメモリへのブロック転送関数を記述する。
- (2) タイマ割込み：OSが内部タイマ割込みまたはフレームメモリ側からの割込みにより定期的にブロック転送を実行する。

4. ワークステーションクラスタとの接続

4.1 ワークステーションクラスタ

本画像表示システムを、ワークステーションクラスタに接続して実験を行った。ワークステーションクラスタは、Sun SPARC station 5を5台用いて構成した。また、ソフトウェアとしては、分散フレームバッファ用に開発したライブラリと、仮想並列マシンライブラリPVMを組み合わせて使用している。

5台のワークステーションのうち4台で計算と画像データの転送を行い、1台はフレームバッファシステム全体の管理を行っている。プロセッサ間のネットワークは、Ethernet (10BASE-T)で構成されている(図6)。

フレームバッファの接続には、STAFF-Linkを簡略化した、S-BUS用シリアルリンクボード(1リンクあたり12.5 Mbytes/sec)を用いた。各リンクには約2mのツイストペアケーブルを用いている。このボードは、TAXIチップがメモリマップドI/Oとして見える単純なもので、DMAなどの機構は持っていない。したがって、シリアルリンクでの転送は、すべてソフトで行う必要がある。

システムの全景を図7に示す。

4.2 ソフトウェア

本画像表示システムをワークステーションクラスタに接続して使用するにあたって、そのハードウェア資源をより簡単に使用できるように、以下のようなソフトウェアを開発した。

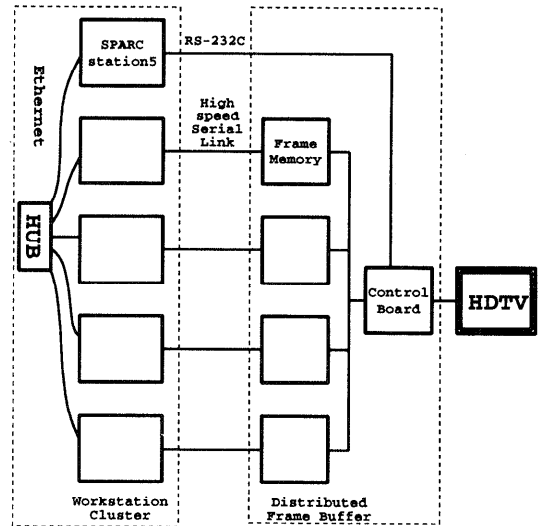


図6 ワークステーションクラスタとフレームバッファの接続

Fig. 6 Connection to a workstation cluster.

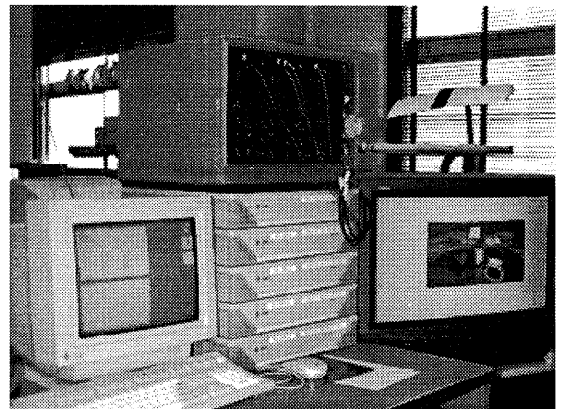


図7 ワークステーションクラスタと表示システム

Fig. 7 Workstation cluster and display system.

マスターデーモン：画像表示システム全体を制御し、スレーブデーモンから来たメッセージの振り分けや、リフレッシュのタイミング制御、スレーブデーモンの管理、タイマからのメッセージの処理を行う。
スレーブデーモン：スレーブデーモンは各ワークステーションで動作する。アプリケーションへのインタフェースの提供を主な仕事とし、マスターデーモンとの協調動作でリフレッシュの同期、自動リフレッシュなどを実現する。

描画ライブラリ：分散フレームバッファへの基本的な描画機能の提供。

フレームメモリボードの接続された計算機のそれぞれのメインメモリ上に各分割領域に対応したローカルフレームメモリ領域を用意し、通常の描画はすべてそ

のローカルフレームメモリ領域を経由して行う。これによって、画像データ転送が削減でき、それぞれの計算機で複数のプロセスを動作させた場合も矛盾なくフレームメモリを更新することができる。

ローカルフレームメモリ領域は、それぞれのスレーブデーモンが共有メモリとしてアプリケーションに提供し、アプリケーション側からは、専用ライブラリを介してアクセスできる。また、画面のリフレッシュ要求や自動リフレッシュなどのコマンドは、プロセス間通信を用いて実装されている。

4.3 リフレッシュ

このソフトウェアシステムでは、ローカルフレームメモリ領域の内容をフレームメモリへ反映させる方法として、

- (1) アプリケーションのプログラミング時に明示的に反映させるための関数呼び出しを書く方法
- (2) タイマ割込みなどを利用してデーモンに自動的にに行わせる方法

の2つの方法を実装しているが、その具体的方法について述べる。実装する際には、できるだけネットワークを介した通信が起こらないようにした。

まず、1の手順を図8に示す。

- (1) スレーブデーモンは、接続されたすべてのアプリケーションからのリフレッシュ要求を集め、マスターデーモンに要求を出す。
- (2) マスターデーモンはすべてのスレーブからの要求を集める。
- (3) マスターはすべてのスレーブから要求が集まるとスレーブへリフレッシュ命令を出す。同時に、自動リフレッシュのためのタイマーをリセットする。
- (4) スレーブデーモンは、ローカルフレームメモリ領域から分散フレームバッファへのデータ転送を行う。

次に、2の手順を図9に示す。

- (1) タイマを用いて周期的に割り込みをかけ、リフレッシュの要求を出す。
- (2) リフレッシュ要求が来ると、マスターデーモンはスレーブデーモンにリフレッシュ命令を出す。同時に、スレーブデーモンから来たリフレッシュ要求のフラグをクリアする。
- (3) スレーブデーモンは、ローカルフレームメモリ領域から分散フレームバッファへのデータ転送を行う。

したがって、アプリケーションからリフレッシュ要求をまったく出さなくても自動リフレッシュ機能が動

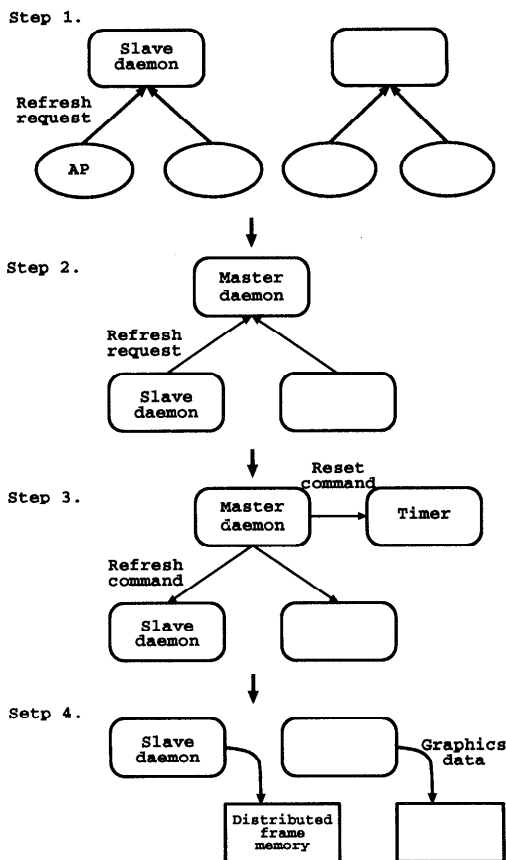


図8 明示的にリフレッシュする場合のメッセージの流れ
Fig. 8 Message flow in explicit refresh.

作するので、一定周期でローカルフレームメモリ領域と、フレームバッファの内容の同期がとれる。また、アプリケーション側から頻繁にリフレッシュ要求が来る場合には、自動リフレッシュ機能によるリフレッシュは起こらない。

5. 実験と評価

本システムは合計 16 枚のメモリボードを用いて 2048×1024 pixel の表示ができるようになっているが、ここでは計算用ワークステーションが 4 台なので、メモリボードを 4 枚使って中央の 4 分の 1 画面 (1024×512 pixel) のみを用いている。また、残りの 1 台では、フレームバッファシステム全体の管理を行わせている。

5.1 レイトレーシングの実時間表示

本フレームバッファの表示実験として、上記ワークステーションクラスターで簡単なレイトレーシングを実行した。使用したオブジェクトは、光源 1、平面 1、球体 3 で、平面にはチェック柄のマッピングを施した。

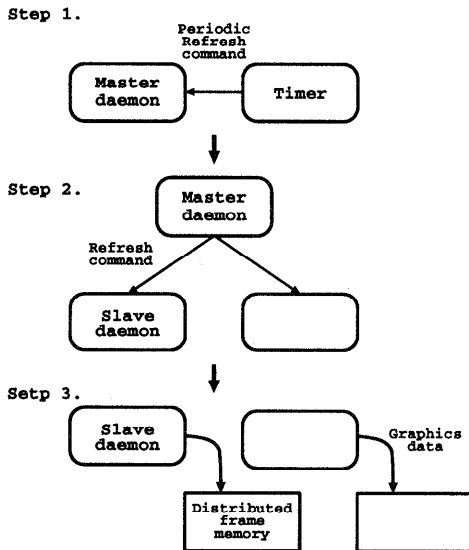


図9 自動リフレッシュする場合のメッセージの流れ
Fig. 9 Message flow in automatic refresh.

表2 並列レイトレーシングの実行時間
Table 2 Execution time of parallel ray-tracing.

総時間 (sec)	1フレーム実行時間 (sec)	1フレーム表示時間 (sec)
89.99	3.00	0.11

レイトレーシングの計算は4台で行い、1台は、分散フレームバッファの管理のみを行う。また、画像は1フレーム計算するごとに視点を移動し、3つの球体もそれぞれ移動させて計算した。画面の分割方法はライン分割であり、負荷の不均衡はほとんどなかった。

この測定では、1024×512 pixelの画像をレイトレーシングで計算しながら実時間で表示させた。1回の測定で30フレームの計算を行っている。測定の結果は、表2のようになった。

5.2 データ転送速度の比較

イーサネット画像データを1台のワークステーションに集める方式（ここでは集中型と呼ぶ）によるデータ収集時間を測定し、比較した。集中型の計測では、4台で計算を行い、残りの1台がデータを回収するようにしている。データ転送はPVMのデータ転送関数を用いて行っている。

ここでは、分散型、集中型ともに、メモリ中の画像データを繰り返し転送するという単純なテストプログラムを作成し、その転送にかかる時間のみを計測した。その結果を表3に示す。ただし、分散型は、4本のリンクの合計転送速度になっている。

表3から、集中型と比較して、分散型は、15倍程度の転送レートが出ている。フレームバッファ全体を

表3 データ転送速度
Table 3 Data transfer rate.

	転送 レート (MB/s)	1フレームの 表示時間 (s)	1秒あたりの 表示コマ数 (fps)
分散型	13.5	0.11	9.01
集中型	0.89	1.71	0.59

使った場合、データの転送量は4倍になる。しかし、分散型では、16本のシリアルリンクを使用することになるので、転送レートが4倍になり、同じ時間で表示できるのに対して、集中型では、転送レートの変化はないので、データ転送には4倍の時間がかかる。

分散型の転送レートは、シリアルリンクチップの最大値の27%程度の性能しか出ておらず、まだまだ改善の余地が残っている。これは、CPUを用いての転送をしている限り極端な高速化は望まれないが、DMAを使用するなどの方法をとれば、さらなる高速化を達成できる。

5.3 評価

上記実験結果より、最初にあげた目的・要求に対して提案システムを評価する。

(1) 転送能力

実験システムではリンクあたりの実行転送レートは3.4MB/sとなり、最大で毎秒9コマの表示性能となった。これは、データ転送を1バイトずつソフトウェアハンドシェイクで行っているためであり、計算機側の問題である。リンクおよびフレームバッファ側の性能には余裕があり、毎秒30コマまで対応できる。また、並列計算機のノード数が大きければ、各フレームメモリユニットから複数本ずつリンクを出すといった構成も考えられる。

(2) スケーラビリティ

実験システムでは、2,048 × 1,024ドット画面を16分割して、1フレームメモリあたり128Kドットとした。リンク数の増加に対しては、たとえば64分割にすれば1フレームメモリあたり32Kドット（ブロック分割で256 × 128ドット）で、64本のリンクで接続できる。さらに各フレームメモリに4本のリンクを設置すれば256リンクも可能である。この場合、1ボードは4フレームメモリユニットと16のリンクポートを持ち、ラックに16枚を実装する構成となる。ケーブルはツイストペアケーブルであり、LANにおけるコンセントレータに類似した形で接続できる。実験システムでは、VMEダブルハイトサイズの4層基板で、TTLを多用して実装したため1フレームメモリユニットが1ボードとなったが、大型基板、ASICの利用な

どにより上記の実装は可能である。

ただし、たとえば1台のフレームバッファに256リンク接続した場合はトータルで数GB/sの転送レートとなるが、通常の単体のディスプレイやカメラでそれだけの転送レートは必要ない。並列計算機から100本を超えるようなリンクを出す必要があるのは、立体画像の入出力のように複数台の画像入出力を同時に行うような応用に対してであると考えられる。

(3) 柔軟性

種々の解像度、画面分割方法に対してFPGAの再コンフィギュレーションで対応できる点に対しては、実験機上で表示して確認している。解像度についてはHDTVやマルチスキャンディスプレイ上での各種表示モード、画面分割については、ハードウェア構成で述べた各種分割モードについて対応を確認した。

実験システムは表示のみであるが、画像入力については基本的に同じ構成でデータの方向を逆にすれば対応できる。

(4) 接続の容易性

接続はモジュラジャックとツイストペアケーブルで行っており、ある程度の距離に対応できる、接続変更が容易である、といった利点がある。大規模システムにおける多くの基板との接続や、ワークステーションクラスタのように空間的に分散したシステムと接続、複数のディスプレイやカメラの接続などに対して柔軟に対応できる。

(5) 応用問題との対応

提案方式は画面分割による転送を行うものであり、並列計算での負荷分散が画面分割的である場合に最も効率良く対応できる。計算上での分割が不規則であったり、画面と直接対応しない場合は、並列計算機の主ネットワークの使用が増加することになる。この点については種々の応用に対する検討、評価が必要であり、課題として残されている。

6. おわりに

大規模並列計算機のための画像表示システムを開発し、ワークステーションクラスタによるハイビジョングラフィックスの表示実験を行った。複数のシリアルリンクによる接続とプログラマブル論理デバイスの使用により以下のような利点を持つシステムが実現できた。

- 空間的に大きなシステムに対して設置や接続が容易である。
- プロセッサ数、ディスプレイ数、解像度などに対して、シリアルリンクの追加によって表示能力を

柔軟に拡張できる。

- 種々の並列計算機に柔軟に対応できる。

はじめに述べたように、本システムは当初、分散共有メモリ型超並列計算機JUMP-1の表示装置として開発が開始されたが、現時点では統合化に至っておらず、この点が課題として残されている。またワークステーションクラスタとの接続においては、現状ではリンクの持つ性能が十分引き出せていない。今後、汎用バスとの接続を改善していく必要がある。

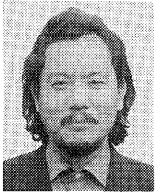
謝辞 本研究の一部は文部省科学研究費補助金、重点領域研究(1)課題番号04235130「超並列ハードウェア・アーキテクチャの研究」および、試験研究(A)(1)課題番号06508001「超並列計算機プロトタイプの開発と試作」による。

参考文献

- 1) Potmesil, M. and Hoffert, E.: The Pixel Machine: A Parallel Image Computer, *ACM Computer Graphics*, Vol.23, No.3, pp.69-78 (1989).
- 2) 大江和一, 稲野 聡, 佐藤弘幸: 高並列コンピュータ AP1000の分散ディスクビデオハードウェアとソフトウェア, 情処研報グラフィクスとCAD, No.94-CG-68, pp.41-48 (1994).
- 3) 平木 敬ほか: 超並列プロトタイプ計算機 JUMP-1の構想, 情処研報計算機アーキテクチャ, No.94-ARC-102, pp.73-84 (1993).
- 4) 中條拓伯ほか: 分散共有メモリ型超並列計算機 JUMP-1の入出力サブシステム, 情処研報計算機アーキテクチャ, No.94-ARC-104, pp.113-120 (1994).
- 5) 中條拓伯ほか: 分散共有メモリ型超並列計算機 JUMP-1におけるスケーラブル I/O サブシステムの構成, 情報処理学会論文誌, Vol.37, No.7, pp.1429-1439 (1996).
- 6) 小畑正貴, 中條拓伯: 超並列計算機 JUMP-1におけるハイビジョン画像表示システム, 情処研報計算機アーキテクチャ, No.94-ARC-108, pp.17-23 (1994).
- 7) Advanced Micro Devices Inc.: Am7968/Am7969 TAXIchip Handbook (1992).

(平成8年9月9日受付)

(平成9年6月3日採録)



小畑 正貴 (正会員)

昭和32年生。昭和55年神戸大学工学部電子工学科卒業。昭和60年同大学院自然科学研究科システム科学専攻博士課程修了。学術博士。昭和59年岡山理科大学理学部助手。平成8年岡山理科大学工学部教授。現在に至る。計算機アーキテクチャ、並列処理の研究に従事。電子情報通信学会, ACM, IEEE-CS 各会員。



江草 俊文 (学生会員)

昭和46年生。平成6年岡山理科大学工学部電子工学科卒業。平成8年同大学院修士課程修了。現在、同大学院博士課程在学中。並列計算機のプロセッサ間ネットワークのハードウェア、通信プロトコルなどに興味を持つ。



中條 拓伯 (正会員)

昭和36年生。昭和60年神戸大学工学部電気工学科卒業。昭和62年同大学院工学研究科電子工学修士課程修了。平成元年同大学工学部情報知能工学科助手。計算機アーキテクチャ、並列処理、並列入出力システムの研究に従事。電子情報通信学会, IEEE CS 各会員。