

3H-8

マルチコンピュータのための競合の起きない多数ポートフレームバッファ

西村 憲 向井 良 國井 利泰
 東京大学理学部情報科学科

1 はじめに

本稿では、次世代グラフィックスシステムのアーキテクチャとしてマルチコンピュータを考え、その際に問題となるフレームバッファのアクセス競合を解決する手段として、競合の起きない多数ポートフレームバッファを提案する。

2 マルチコンピュータに基づいたグラフィックスシステム

グラフィックスシステムの応用分野は、3次元 CAD、地形情報システム、ビジュアルシミュレーションなど多岐にわたっている。また画像生成アルゴリズムにはZバッファ法、レイトレーシング法など種々のものが目的に合わせて選択されている。従ってこれからのグラフィックスシステムには、ポリゴンを高速に描くという能力のみならず、種々の画像生成アルゴリズムに柔軟に対応でき、また画像生成以外の計算も高速に実行できることが望まれている。

本稿では、このような要求を満たすグラフィックスシステムの構成として、汎用プロセッサ複数個を通信リンクでネットワーク状に結合したマルチコンピュータ（疎結合マルチプロセッサ）を考える。

マルチコンピュータに基づいたグラフィックスシステムの利点は以下の通りである。まず、共有バスを持たないためプロセッサの数を大きくしてもアクセス競合による性能の劣化が無い点である。次に、ポリゴン描画専用のハードウェアと異なり、あらゆるグラフィックス処理に適用できる柔軟性を持つことである。

マルチコンピュータ上でグラフィックス処理を行なう場合、プロセッサへの処理の割り当ての方法には次の2つのアプローチが存在する。

- 画素並列型 プロセッサがN個あるとき、画面全体の画素の集合を互いに素なN個の部分集合に分割してそれぞれを別のプロセッサに割り付ける方式である。画像生成する際に、各プロセッサがどの図形に関する形状情報をアクセスするかが定まっていなため、全てのプロセッサのローカルメモリに全ての図

形情報のコピーを持つか、或いはブロードキャスト通信によって描画を行なう毎にプロセッサに供給しなければならない。後者の場合、プロセッサ数が多くなると通信にかかる時間が描画自身にかかる時間を上回る可能性がある。

- 図形並列型 場面を構成する図形群を部分集合に分割してそれぞれをプロセッサに割り当てる方式である。各プロセッサのローカルメモリに全図形情報の一部だけを格納しておけば十分なため、大容量メモリ、図形情報通信のどちらも必要がない。

マルチコンピュータでグラフィックス処理をする上で問題となるものの1つに、フレームバッファの構成がある。ポリゴン描画などの応用では全プロセッサが頻繁にフレームバッファにアクセスするため、共有バスを用いたのではアクセス競合による性能劣化が著しい。処理を画素並列型に限れば、各プロセッサは予め決まった領域のみをアクセスするので、フレームバッファをそれに合わせて分散化すればよい[1, 2]。しかし図形並列型の場合、各プロセッサのアクセス領域が未知である。そこで各プロセッサがどの位置の画素でも自由に書き込むことができる、競合の起きない多数ポートフレームバッファ(CMFB)を次章において提案する。

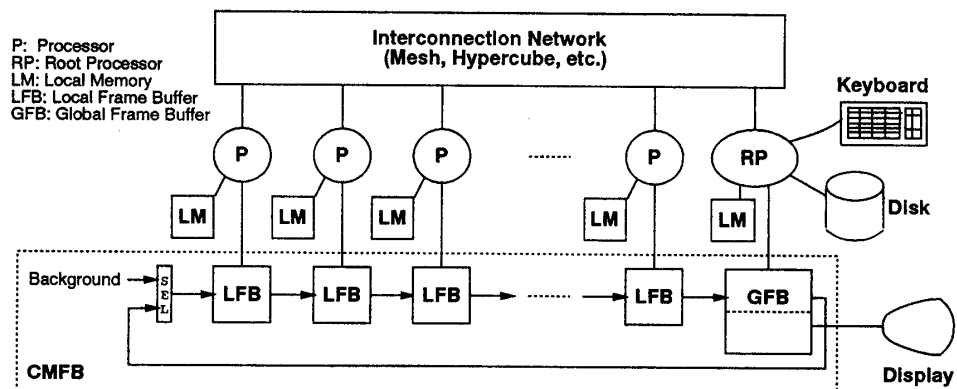
3 競合の起きない多数ポートフレームバッファ(CMFB)

3.1 CMFBの構成

図1にCMFBを持ったグラフィックス用マルチコンピュータの全体構成を示す。点線で囲まれた部分がCMFBに相当する。

CMFBは各プロセッサ毎に存在する局所フレームバッファ(LFB)とシステムに1つだけ存在する大域フレームバッファ(GFB)からなる。LFBは各プロセッサの出力する画素の色情報及び深さ情報(Z値)を保持する。LFBはページングによる仮想記憶に類似したメカニズムを内部に持ち、見かけ上全画面分のフレームバッファを持っているように扱える。GFBは画面全体の色および深さ情報を保持し、表示用と描画用にそれぞれ独立したバッファを持つダブルバッファ構成になっている。

図1. CMFBを持つマルチコンピュータ



LFB と GFB は 1 画素当たりのビット数に等しい幅を持つ通信路によって直線状に結合され、周期的に各 LFB に格納されている画像が合成されて GFB に転送されるようになっている。この転送は画面のスクリーンと同様な順序、すなわち画面の左上に始まり右下で終る順序で行なわれ、1 画面分の転送が終了すると即座に画面左上に戻り次の転送を続ける。スキャンの周期は CRT のそれと一致している必要はないが、同程度あるいはそれ以上であることが望ましい。

各画素の転送は次のように行なわれる。まず、図 1 において最左端の LFB にバックグラウンドの色と無限大の Z 値が入力される。各 LFB は左から受けとった Z 値と自分のバッファに格納されている Z 値とを比較し、もし前者の方が小さければ（つまり視点に近ければ）左から受けとった色及び Z 値をそのまま右の LFB に渡す。さもなければ、自分のバッファに格納されている色及び Z 値を右の LFB に渡す。こうして最終的に GFB には合成された色と Z 値が入る。

上の転送はパイプライン的に動作する。すなわち最左端の LFB では、2 番目の LFB に値を渡すのと同時にすぐに次の画面に関する処理が開始される。これにより、LFB の内容は最大 $wp + (n - 1)p$ の遅れを伴って GFB に転送される。ただし、 w は画素数、 p はパイプラインのピッチ、 n はプロセッサ数である。

最左端の LFB から入力するデータは切替えにより GFB の内容にすることも可能であり、これにより上で述べた転送とは逆に GFB の内容を LFB に取り込むことが実現できる。これは画像処理などの応用で使われる。

3.2 LFB の実現法

すべての LFB が全画面に相当する画像メモリを持つことはコストの面で難しい。そこで図 2 のように、画面を小さな領域（パッチ）に分割し、プロセッサがアクセスしたパッチのみに、画像メモリを割り当てる方式を考案した。これを仮想フレームバッファと呼ぶ。フレームバッファへのアクセスは局所性があるため、このような構成にすることによりかなりの画像メモリ節約が期待できる。

図 3 に仮想フレームバッファに基づいた LFB の構成を示す。画像メモリ (IM) は画素の色および Z 値を保持する。パッチテーブル (PT) には各パッチ毎に、IM が割り当てられているかどうかを示すフラグ及びパッチ左上点にあたる IM のアドレスが格納されている。PT はラッチなどと共にアドレス変換回路 (ATC) を形成し、これによって仮想デバイス座標から物理デバイス座標への変換がなされる。

画像走査時には ATC に走査カウンタの値が入力され、それをもとに IM が読み出され、更に後段 LFB へのデータが計算される。も

しこのとき IM 未割り当てのパッチだったならば、前段 LFB から受けとったデータが素通りする。CPU アクセス時には ATC に CPU のアドレスバスが接続され、物理アドレスが計算された後、IM がアクセスされる。もし IM 未割り当てのパッチをアクセスした場合には CPU に割り込みがかかり、新たなパッチ割り付けが行われる。

IM の大きさは有限であるので、フレームバッファへのアクセス局所性が弱い場合には IM が溢れることがある。この対処方法には次の 2 通りが考えられる。一つは、相互結合網によって他のプロセッサに転送する方法である。もう一つは LFB の内容が GFB に転送されるまで待つ手法である。前者の場合ラスタデータを通信するためそのオーバーヘッドが問題となる。後者の場合 CPU は最大で wp の間待たされるのでそれによる性能劣化が心配されるが、転送待ちの間に別の処理をするようにソフトウェアを工夫すれば、致命的な欠陥にはならないと考えられる。

4 おわりに

本稿ではマルチコンピュータのための競合の起きない多数ポートフレームバッファを提案し、その実現法を概説した。

ここで取り上げたシステムについては、現在 64 個のプロセッサを持つプロトタイプを検討中であり、完成後はパッチの大きさの最適値やそのアルゴリズム依存度、画像メモリの容量と性能の関係、そして他のフレームバッファ構成との性能比などを調べて行く予定である。

参考文献

- [1] H. Sato, M. Ishii, K. Sato, M. Ikesaka, H. Ishihata, M. Kakimoto, K. Hirota, and K. Inoue, "Fast Image Generation of Constructive Solid Geometry Using a Cellular Array Processor," *ACM Computer Graphics*, vol. 19, pp. 95-102, July 1985.
- [2] M. Potmesil, L. McMillan, E. M. Hoffert, J. F. Inman, R. L. Farah, and M. Howard, "A Parallel Image Computer with a Distributed Frame Buffer: System Architecture and Programming," *Proc. of Eurographics '89*, pp. 197-208, 1989.

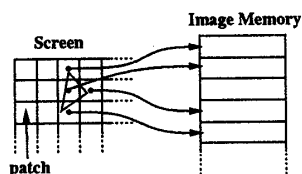


図 2. 仮想フレームバッファ

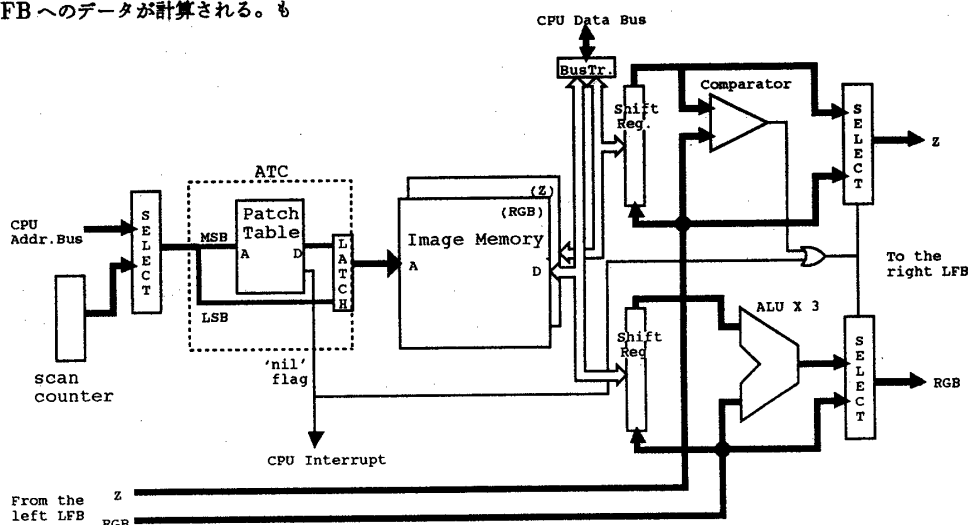


図 3. LFB の構成