

## 視覚表現によるハイパーキューブアルゴリズムのシミュレータ

1B-8

塩田 佳明, 渋谷 進

茨城大学工学部情報工学科

### 1 はじめに

これまでに、各種グラフアルゴリズムや、ソートングアルゴリズム等の動きを視覚的に表現して、アルゴリズムの理解、アルゴリズムの評価、新たなアルゴリズムの発達を促すようなシミュレータが作られている [1],[2].

また、一般に並列計算機システムの設計におけるデバックや、アルゴリズムの実行過程の確認は容易でない。本研究ではこれらの手助けを目的としてハイパーキューブアルゴリズムのシミュレータを作成した。シミュレータは、アルゴリズムの実行を指示するスクリプトファイルを読み込み、Xウィンドウ上に実行過程をアニメーション表示する。

作成したシミュレータにより、ハイパーキューブ結合のプロセッサ間通信の様子やデータの流が視覚的に確認できる。また、読み込むファイルを交換することで、ハイパーキューブ上で動作する別のアルゴリズムの実行表示も可能であり、上記の目的だけでなく、アルゴリズムやソフトウェアの開発設計支援等にも有用であるものと思われる。さらに、教育用ツールとしての役割も期待できる。

### 2 反射2進順レイアウトのハイパーキューブ

表現するハイパーキューブは図1に示すように反射2進順にレイアウトしている。反射2進順にレイアウトすることで、プロセッサ数が大きいときは昇順にレイアウトした場合の約4/9の面積で実現できる [3]. また、ハイパーキューブアルゴリズムだけでなく格子結合のアルゴリズムも実行可能である。

### 3 シミュレータ

#### 3.1 シミュレータの構成

シミュレータプログラムは、C言語で記述し、XプログラミングのライブラリはXlibのみを用いた。実行マシンはSun社のSparc-Stationシリーズである。

A Simulator of Hypercube Algorithms Using Visual Expression

Yoshiaki SHIOTA, Susumu SHIBUSAWA  
Faculty of Engineering, Ibaraki University

シミュレータの構成は図2のようになっている。プログラムは大きく3つの部分に分けられる。プログラム中、下側の2つの部分がグラフィック部であり、Xウィンドウ上に反射2進ハイパーキューブを描画して、アルゴリズムを表現するためのものである。この上に基本的なアルゴリズム操作群を付加することにより、シミュレータが成り立っている。アルゴリズム表現の際は、これら基本的な操作を組み合わせることにより複雑なアルゴリズムを実現する。そして最終的にスクリプトファイルを読み込んでアルゴリズムを実行する。

なお、モデルとした反射2進ハイパーキューブは、 $2^3 \times 2^3$  個のプロセッサを2次元平面上に配置した図1と同じものを用いている。

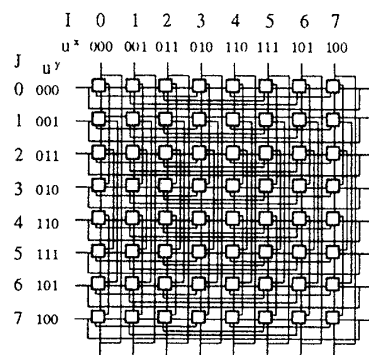


図1: 反射2進順レイアウトのハイパーキューブ

#### 3.2 プロセッサ間通信、データの表現

プロセッサ間データ通信の表現は、通信を行なっている通信リンクをウィンドウ上で太く描くことで表現した (図3)。

プロセッサが保有するデータは、ウィンドウ上のプロセッサの中にデータを書き入れることで表現している。このことにより、データの移動が理解し易くなっている。

#### 3.3 基本的なアルゴリズムの操作

ハイパーキューブへの操作としては、次のようなものが基本的なアルゴリズム操作として挙げられる。

- ブロードキャスト
- データのコピー
- データの四則演算

### ● 隣接プロセッサとのデータの比較交換

これらを組み合わせれば、ある程度複雑な操作が可能である。これらはシミュレータプログラムに関数として含まれている。

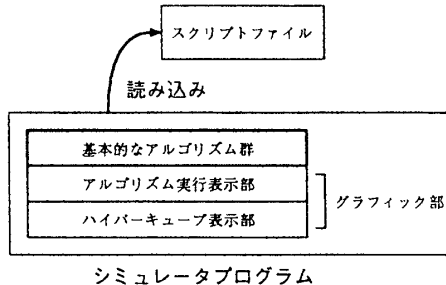


図2: シミュレータの構成

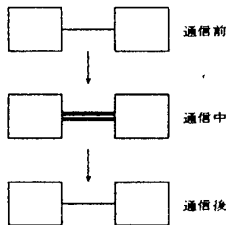


図3: データ通信の表現

### 3.4 スクリプトファイル

シミュレータにアルゴリズムを実行させるのにスクリプトファイルを使用する。この手法は [1] で用いられており、本研究ではこれを単純化したものを用いる。このファイルを使用することで、アルゴリズムの開発時には、実際に実行したい並列アルゴリズムをスクリプトとして記述してシミュレータに読み込ませることで、実行の様子を容易に確認できる。スクリプトファイルの内容は以下のようなものである。

```
DataInit(2)
```

```
Bitonic_Sort
```

DataInit(2) は、各プロセッサに 2 桁以下の整数をランダムに持たせ、Bitonic.Sort は、実際にバイトニックソートを実行する。現在は、このようにごく単純なスクリプトであるが、例えばバイトニックソートでもデータの比較交換というプリミティブな操作を組み合わせるに過ぎない。3.3 節で述べたようにこれらプリミティブな操作はシミュレータに関数として含まれており、内部ではこれらの関数を組み合わせて Bitonic.Sort を処理している。

## 4 実行結果

現在のところ表現しているアルゴリズムは、バイトニックソート、プレフィクス計算、行列積である。こ

のうち、行列積アルゴリズムの実行中の画面を図4に示す。これは  $4 \times 4$  の 2 行列の積を求めている。反射 2 進順に配置されているため最終結果も反射 2 進順になる。視覚的表現により、現在通信中のプロセッサや、データの流れる様子が理解し易い。

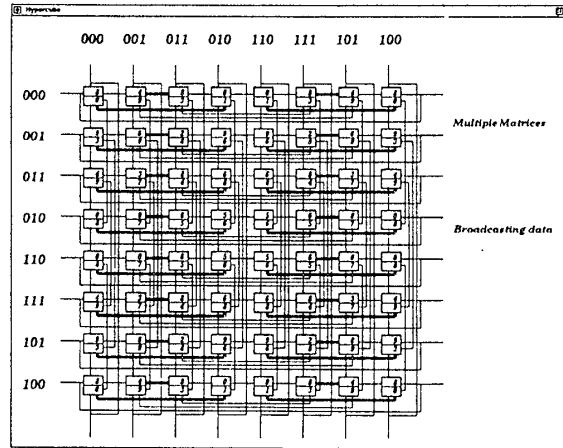


図4: 行列積アルゴリズムの実行画面

## 5 おわりに

本研究で作成したシミュレータにより、ハイパーキューブ上でのアルゴリズムの動き、データ転送の流れを視覚的に確認することができ、アルゴリズム開発の際の実行過程の確認や並列処理教育への応用が可能である。実際、本シミュレータを学生実験において並列処理を紹介するのに用いている。

今後は、比較交換やブロードキャストなどの操作を、C言語ライクな並列疑似コードによってスクリプトファイルに記述でき、それを解釈実行するようなシステムへの改良を考えている。これによって、計算機システムやシステムソフトウェアのデバック等の手助けに応用していく予定である。

## 参考文献

- [1] J.T.Stasko: Tango: A Framework and System for Algorithm Animation, IEEE Computer, Vol.23, No.9, pp.27-39 (1990).
- [2] 塩田 佳明, 渋谷 進: X ウィンドウを用いたハイパーキューブの並列アルゴリズム表現, 情報処理学会研究報告, Vol.93, No.49, pp.1-8 (1993).
- [3] 渋谷 進: 超立方体結合と大域バス結合より成る一相互結合モデルについて, 情報処理学会論文誌, Vol.32, No.4, pp.532-543 (1991).