2002年日本オペレーションズ・リサーチ学会
秋季研究発表会

# MMXテクノロジによる高速化手法を用いたセルオートマトン・シミュレータ用コンパイラの開発

琉球大学　＊赤嶺　有平　AKAMINE　Yuhei
琉球大学　遠藤　聡志　ENDO　SATOSHI
琉球大学　山田　孝治　YAMADA　Koji

## 1. はじめに

セルラ・オートマトン（CA）法は，局所的な相互作用を定義することで複雑現象を不規則性も含めて再現することができるため，流体の乱流，交通渋滞，自然災害など従来の微分方程式を基礎とするモデルでは解析の難しかった現象も解明できるとされている[1]．

CAでは，セルと呼ばれる要素の処理を理論上すべて並列に行い，全てのセルにおいて同一の処理手続きが適用される．そのため，SIMDを用いることで高速なCAシミュレーションが可能である．SIMDは，一つの命令の流れが複数のデータの流れを処理する並列アーキテクチャの一種である．

MMXテクノロジは，パーソナルコンピュータ用のプロセッサに搭載されたSIMD型演算命令セットで，普及率が非常に高くローコストである．筆者らは，MMXテクノロジを用いてCAシミュレータを高速化する手法を提案した[2]．

本研究では，提案手法を用いて高速なCAシミュレータを生成するコンパイラの開発を行った．本論文では，高速化手法と開発したコンパイラの評価実験結果について述べる．

## 2. 並列化コンパイラ

MMXテクノロジは，機械語命令セットであるため利用するにはアセンブラ言語を用いる必要がある．アセンブラ言語によるプログラミングは非常に難解な作業であるため，提案した高速化手法を用いたCAシミュレータを生成するコンパイラを開発した．開発したコンパイラ（DORAコンパイラ）は，ソースとしてDORA言語を用いる．DORA言語は，筆者らがCA用計算環境として開発を進めているDORAシステムのために設計したCA近傍則記述用言語である[3]．DORAコンパイラは，DORAシステムの一部として開発した．

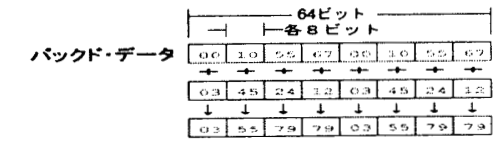


**図1　パックド・データ演算の例**

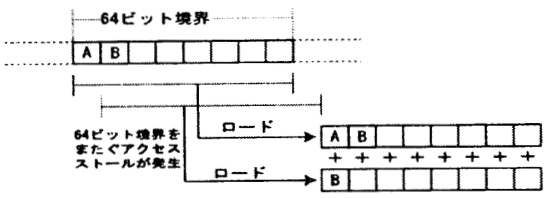


**図2　プロセッサ・ストールが発生する演算の例：要素A・B間でパックド・データ演算を行う場合，パックド・データ内でのA,Bの位置を合わせるために64ビット境界をまたぐアクセスが発生し，プロセッサがストールする．**

本節では，DORAコンパイラのコード生成に用いたCAシミュレータの高速化手法について述べる．

### 2.1.　パックド・データ演算

MMXテクノロジは，64ビット幅のデータをいくつかのビット幅の小さい複数のデータの集合として扱う．パックド・データ演算は，これらのデータの集合（パックド・データ）を同時に処理する（図1）．MMXテクノロジのロード命令では，メモリアドレスの６４ビット境界をまたぐメモリアクセスを行うとプロセッサがストールする．そのため，メモリを６４ビットのブロックに分けた時，各ブロック内の異なる位置にある複数の値に対してパックド・データ演算を行うと非常に演算効率が悪くなる（図2）．パックド・データ演算を用いた処理を行う場合，このような状況が発生しないように，データの配置を工夫する必要がある．

### 2.2.　データ構造の最適化

CAでは，状態値を持つセルが一様な格子空間上に配置されていると考える．各セルは，その近傍のセルの状態値を参照して自身の状態値を更新する．通常，計算機において2次元CAシミュレーションを行う場合，状態値は，ラスタ・グラフィックスのビデオメモリのように左上から順にメモリに配置される．CAでは，近傍セルの状態値を頻繁に参照するため，状態値の更新をパックド・データ演算で行う際に，プロセッサのストールが頻発する．

この問題に対して提案手法では，以下の処理を行うことで効率的なパックド・データ演算を実現する（図3）．

1．セルの格子空間を分割する
2．各小空間上で同一なローカル座標に位置するセルの値を取り出してパック化する
3．パックド・データ演算を用いて並列に処理する

セルの状態値は，各部分格子空間上で同一のローカル座標に位置するセルの状態値をパック化した状態でメモリに配置しておく（図4）．そうすることで，一回のロード命令で，パック化が終了する．さらに，MMXテクノロジを実装しているプロセッサは，1命令でロードと演算を実行できるため，場合によっては上記の２，３を同時に処理できる．

### 2.3. 条件付演算の並列化

MMX テクノロジによる並列演算では，その性質上条件分岐をおこなうことができないため，if 文のように条件に応じて処理を選択することができない．そのかわり，比較演算命令と呼ばれる，条件によってビットマスクを生成する命令が用意されている．提案手法では，条件文をすべて比較演算による条件式に置き換えて処理する．例えば，

$$\text{if( x = y ) x := 2 else x := 3}$$

などの条件文は，

$$x = \mathrm{comp}(x = y) \,\&\, 2 + \mathrm{comp}(x \neq y) \,\&\, 3$$

と置き換えることができる．ここで，comp(条件)は，「条件」が成り立つ時，全ビットが 1 である値（たとえば 8 ビット値なら 255），そうでない時 “0” を返す．& は，ビットごとの論理積である．

## 3. 評価実験

開発したコンパイラの性能を評価するために，C コンパイラと DORA コンパイラの双方を用いて CA シミュレータを作成し，実行速度の比較を行った．実験に用いた CA モデルは，ライフゲームと HPP モデルである．実験は，800MHz で駆動する Pentium® III プロセッサ上の Windows® 2000 professional で行った．状態値は 8 ビットで表現し，格子サイズ 256x256（セル数 65536 個）で全セルの状態値を 5000 回更新するのに必要な処理時間を計測した．C コンパイラ版は，Pentium Pro プロセッサ向けの実行速度に対する最適化のオプションをつけてコンパイルした．一般にこのオプションをつけてコンパイルした場合がもっとも実行速度が速くなる．

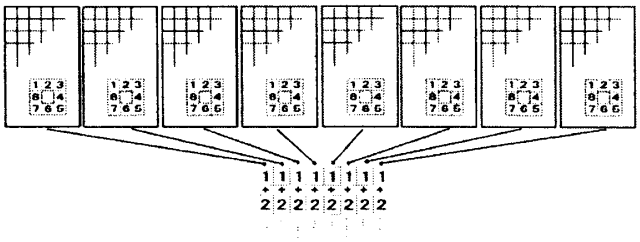

**図 3 状態値の分割配置**

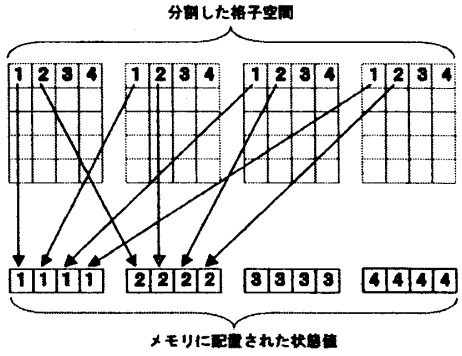


**図 4 メモリ上の状態値の配置：各部分空間上で同一のローカル座標に位置するセルの状態値をパック化する**

### 3.1. ライフゲーム

ライフゲームは，セルに接する 8 セルの状態値の合計と対象のセル自身の状態値に以下の規則を適用して，次の時間ステップのセル状態を決定する[4]．

- 状態和が 2 か 3 の場合，次の状態は 1
- 状態和が 4 以上または 1 以下の場合，次の状態は 0
- 状態和が 3 の時，次の状態は 1

これらの規則は if 文を用いて記述されるが，コンパイラによって比較演算に変換される．

### 3.2. HPP モデル

HPP モデルは，流体解析に用いられる格子ガスオートマトン法で最初に提案されたモデルである．このモデルでは正方格子を用い，衝突・散乱の規則として正面衝突のみを考える．衝突後の粒子はそれぞれ，それまで粒子のなかった方向へ跳ね返る．

### 3.3. 結果

実験結果を表 1に示す．DORA コンパイラによるシミュレータは，C コンパイラによるものと比較してライフゲームで 6.5 倍，HPP モデルで 4.5 倍高速化されることが示された．

ライフゲームにおいては，提案手法を用いてハンドコーディングで作成したところ，7.5 倍高速化されることがわかっている[2]．今回開発したコンパイラは，提案手法による並列化以外の最適化はまったく行っていないため，コンパイラのオプチマイザを改良すれば，さらに高速化される可能性がある．

| | | | |
|---|---|---|---|
| ライフゲーム | 18687 | 2854 | 6.5 |
| HPP モデル | 17405 | 3765 | 4.5 |
| | C コンパイラ | DORA コンパイラ | 速度比 |

**表 1 実験結果（単位はミリ秒）**

## 4. おわりに

本論文では，筆者らが提案した CA シミュレータの高速化手法について説明し，それを用いて CA シミュレータを自動生成する DORA コンパイラについて述べた．

DORA コンパイラによるシミュレータと C 言語によるシミュレータをライフゲームと HPP モデルについてそれぞれ実行速度の比較を行ったところ，良好な結果を得られた．

今後は，さらに多くのモデルについて実験を行い，オプチマイザの改良を行う必要がある．

**参考文献**


[1] S. Wolfram: Computation theory of cellular automata, Communications in Mathematical Physics, Vol. 96, pp. 15-57 (1984).

[2] 赤嶺, 遠藤, 山田:MMX テクノロジを用いたセルラ・オートマトン・シミュレータの高速化, 人工知能学会論文誌, Vol.16, No.1, pp.74-84 (2001).

[3] 赤嶺, 遠藤, 山田:セルラ・オートマトン・シミュレータ用インタプリタの開発, 人工知能学会論文誌, Vol. 17, No. 1, 採録決定 (2002).

[4] M. Gardner: The fantastic combinations of John Conway's new solitaire game ‘Life’, Scientific American, pp. 120-123 (1970).