

## マルチエージェントシミュレーションにおける並列処理方式の比較

森下 仙一†

蟻川 浩‡

村田 忠彦†‡

分散処理型マルチエージェントシミュレーションのMPIによる実装において、データの分割方法として正方形型と短冊型が考えられる。本稿では正方形型と短冊型のデータ分割方法がマルチエージェントシミュレーションの実行時間にどのような影響を与えるかについてシミュレーションプログラムを実装し、実行時間を測定する。その結果、大規模マルチエージェントシミュレーションプログラムを実装する際は正方形型分割方式を採用するとよいことを示す。

### Comparisons of parallel processing method in Multi Agent Simulation

Sen-ichi Morishita†

Hiroshi Arikawa‡

Tadahiko Murata†‡

This paper describes a distributed multi-agent simulation using Message Passing Interface. The data distribution method has the square type and the rectangle type. To study the influence of data distribution method on the simulation time, we have implemented the two types of data distribution algorithm with MAS program. And we evaluated the execution time of each program. As a result, we can conclude that it is necessary to adopt a square type of data distribution algorithm when we implement the large scale MAS program.

#### 1. はじめに

マルチエージェントシミュレーション(MAS)は、環境と自律的な行動をするエージェントから構成されているシミュレーションで、多数のエージェントが相互に作用しながら、環境も変化するという特徴を持っている。近年、MAS の枠組みにより、コンピュータ上で人工的に構成した「社会」(環境)の中で、人流や交通流、経済現象などの社会現象の再現、分析が試みられている[1,2]。

現実の社会を MAS に適用させる場合、通常の MAS よりも大規模なシミュレーションを行う必要がある。大規模 MAS を実現するための研究は、中島ら[3]によって行われている。しかし中島らは、大規模 MAS の実行基盤の構築については述べているものの、大規模 MAS の高速化については課題として挙げるに留まっている。そこで本稿では、大規模 MAS をグリッド・コンピューティングに適応させ、高速化を図っていく。

グリッド・コンピューティングとは、地理的、組織的に広範囲に分散した複数のコンピュータを、ネットワークで接続、共有することにより、大規模演

算向けコンピュータを仮想的に構成する手法である[4]。

MAS にはエージェント数が増加するにつれ、エージェントの処理に時間がかかり、計算時間が増加するという問題がある。そのため、グリッド・コンピューティングに MAS を適用させる場合、処理をどのように分散させるかを考慮しなければならない。処理の分散方法には、エージェントをエージェント数に応じて分割し、分割したエージェントが所属する環境を割り当てる方法(エージェント分割方式)、環境を環境の大きさに応じて分割し、分割した環境に所属するエージェントを割り当てる方法(環境分割方式)などが考えられる。本稿では、環境分割方式を採用する。

MAS を実現するモデルとして知られている Sugarscape 環境を分割の対象とする。Sugarscape 環境は 2 次元空間で表現されており、環境分割方式を適用して分割する方法としては、1 方向に分割する短冊型、2 方向に分割する正方形型などが考えられる。そこで本稿では、Sugarscape モデルの Sugarscape 環境を対象とし、環境分割方式を適用する際に、分割方法の MAS に与える影響について、実際にグリッド環境上に MAS を構築し、実験結果を元に得られた知見を述べる。

#### 2. 環境改善資金調達シミュレーション

本稿では、MAS の例として、西崎らによって考案された環境改善資金調達モデル[2]を採用する。

† 関西大学 総合情報学部

Faculty of Informatics, Kansai University

‡ 関西大学 政策グリッドコンピューティング実験センター

Policy Grid Computing Laboratory, Kansai University

このシミュレーションモデルは、人工社会モデルの1つである Sugarscape モデル[1]を拡張したものである。環境改善資金調達モデルについての詳細は、西崎らの論文[2]を参照してもらおうとし、環境改善資金調達モデルのベースである Sugarscape モデルについて、Sugarscape モデルの並列処理方法について以下で詳しく述べる。

### 3. Sugarscape モデルの並列化

単一計算機用の Sugarscape モデルを、並列化に対応させずにそのまま分割すると正しい結果が得られない。例えば、並列化に対応させずにただ分割した場合、エージェントの移動範囲が限定されてしまい、シミュレーションの結果に誤差が生じる。そこで Sugarscape モデルの並列化には、我々が提案した手法[5]を拡張する。拡張した点は、隣接区域との環境情報の通信の部分である。通信範囲をエージェントの視野の最大値にする方法を採用し、通信量を大幅に削減できるように拡張を行った。

### 4. MPIによるMASの実装

並列処理可能なMASの実装について示す。本稿では、MPI ライブラリを用いた場合のMASの動作フローについて述べる。

#### 4.1 MPIライブラリ

MPI ライブラリは分散メモリ型並列処理環境で並列プログラムを実装するために必要不可欠となる、プロセス間同士でメッセージを交換するための通信ライブラリの総称である。Message Passing Interface Forum [6]にて通信ライブラリの規格が策定され、規格に基づいて様々なライブラリが実装されている。MPI ライブラリとして有名なものとしては、MPICH [7]、LAM [8]、GridMPI [9]がある。

MPIは、主に1対のプロセス間の通信を行う「1対1通信」とプロセスのグループ間で通信を行う「集団通信」があり、これらの命令を適切に用いて並列プログラムを実装する。

#### 4.2 MPIを用いたMASの流れ

MPIを用いたMASでは、複数数のプロセッサを用いてシミュレーションを行う。各プロセッサとプロセスの実行の関係は図1ようになる。図1において、縦方向はプロセスの実行位置、横方向はプロセッサの数を意味する。図中P1からPnは各プロ

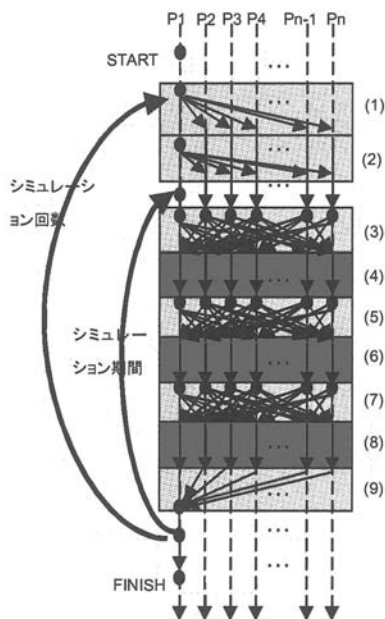


図1 MPIを用いた並列計算の動作フロー

セッサの番号を意味する。以下では、各プロセスの処理内容について説明する。

まず処理(1)において、Sugarscape 環境を分割した情報(各区域の情報)を、自分を含む他のプロセッサに配分する。その後、処理(2)において、各区域が担当するエージェントの初期情報を、自分を含む他のプロセッサに配分する。

処理(3)では、各プロセッサが処理するのに必要となる隣接区域の情報を交換し、自分の環境に付加する。その後、プロセッサ毎に処理(4)のエージェント意思決定処理が行われる。

処理(5)では、各プロセッサが持つ Sugarscape 環境の領域外に移動するエージェント情報を、該当する領域を担当するプロセッサに送るために通信を行う。その後、処理(6)ではエージェントの衝突判定を行い、処理(7)では、環境改善資金調達モデルを用いているのなら、必要なデータを集めつつ汚染物質の削減を行い、処理(8)で各領域における資源回復処理を行う。処理(9)で各領域のデータを収集し、ステップ毎のデータ集計を行う。

処理(3)~(9)を所定のステップまで繰り返し実行

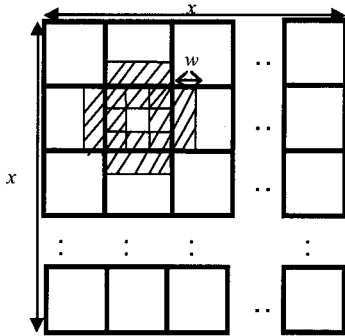


図2 正方形型の通信

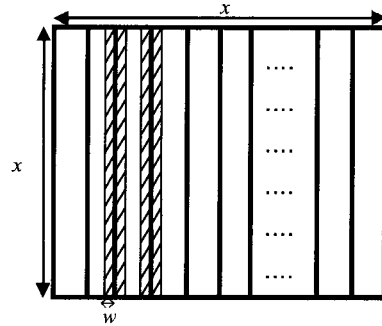


図3 短冊型の通信

し、1試行の計算が完了する。そして統計的に十分な数だけ試行し、シミュレーション結果を得る。

ここで、プロセス間で通信が必要となる処理は処理(1)、処理(2)、処理(3)、処理(5)、処理(7)、処理(9)の部分である。これらの処理にMPIライブラリを用いる。

処理(1)では、どのプロセスにも均一な大きさのデータを送信する必要があるため、「集団通信」であるMPI\_Scatter()を用いて通信を行う。

処理(2)では、プロセス毎に不均一な大きさのデータを送信する必要があるため、MPI\_Scatter()を用いてエージェント数の通信を行い、MPI\_Scatterv()を用いてエージェントの初期情報の通信を行う。

処理(3)と処理(5)では、「1対1通信」であるMPI\_Bsend()、MPI\_Recv()、MPI\_Probe()を用いて通信を行う。MPI\_Bsend()で隣接領域の全てにデータを送り、MPI\_Probe()でデータが送られてくるのを待つ。その後データが隣接領域から送られてくると、MPI\_Recv()を用いてデータの受信を行う。このとき、処理(3)は均一な大きさのデータをやり取りするので問題は無いが、処理(5)では不均一な大きさのデータをやり取りするため、MPI\_Recv()の前に「1対1通信」であるMPI\_Get\_count()を用い、データの大きさを取得する。

処理(7)では、「集団通信」であるMPI\_Allreduce()とMPI\_Allgather()を用いて通信を行う。まず汚染物質量の削減の計算に必要なデータを、MPI\_Allreduce()を用いて、全プロセスが取

得する。その後、MPI\_Allgather()を用いて汚染物質量の削減箇所を決定し、汚染物質量の削減を行う。

処理(9)では、「集団通信」であるMPI\_Reduce()を用いて、全プロセスでの処理結果の集計をプロセス1に集める。

## 5. 環境情報の分割方法の違い

環境情報分割方式を行うとき、正方形型か短冊型かによって通信時間に影響が出ると考えられる。通信時間を $T$ とすると、通信時間は $T \cong$  通信確立時間 + 環境情報通信時間 + エージェント情報通信時間のように定義できる。

図2に正方形型の場合の通信例を示す。図2は、マシン台数を $n$ 台としたとき、環境情報を $\sqrt{n} \times \sqrt{n}$ 分割した状態を表しており、斜線の部分が通信部分である。この分割方法のときの通信時間 $T_s$ は、環境の大きさを $x$ 、領域間を移動するエージェント数を $a$ 、エージェントの最大視野を $w$ 、1台との通信確立時間を $c_1$ 、1単位の環境情報の通信時間を $c_2$ 、1単位のエージェントの情報通信時間を $c_3$ とすると、式(1)のようにモデル化することができる。

$$T_s \cong 4c_1 + \frac{8wx}{\sqrt{n}}c_2 + 4ac_3 \quad (1)$$

同様に図3に短冊型の場合の通信例を示す。図3は、マシン台数を $n$ 台としたとき、環境情報を $n \times 1$ 分割した状態を表しており、斜線の部分が通信部分である。この分割方法のときの通信時間 $T_r$ は、式(2)のようにモデル化することができる。

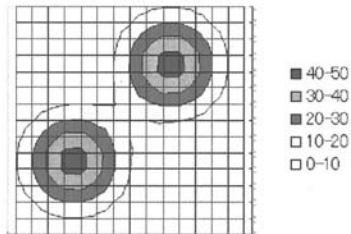


図4 資源量の分布 1

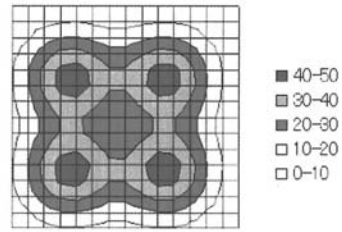


図5 資源量の分布 2

$$T_r \cong 2c_1 + 4nxc_2 + 2ac_3 \quad (2)$$

本稿では、エージェント情報通信時間を考慮しないとする。エージェント情報の通信量は、環境情報量が大きくなると、通信時間の一部しか占めなくなるからである。

式(1)と式(2)から、正方形型はマシン台数が多くなることで、環境情報通信時間が短くなる。一方、短冊型は隣接する領域が少ないため、通信確立時間が短くなる。このことから、環境情報の面積が狭いときは、データ転送量が少いため、通信確立時間が短い短冊型の方が通信時間が短くなり、環境情報の面積が広いときは、データ転送量が増加するので、データ転送量が少ない正方形型の方が通信時間が短くなると推測できる。

## 6. 実験

本稿では、環境情報の分割方法の違いについて検証するために、環境改善資金調達シミュレーションプログラムを使って評価実験を行った。本稿では、OA (Open access), VC (Voluntary contributions), LO (Lotteries) の3つのシミュレーションを対象とした。

### 6.1 実験環境

実験で用いた計算機単体の性能は CPU : Pentium4 3.0EGHz, メモリ : 2GB, OS : Linux 2.4.7., MPI ライブラリ : LAM 7.1.である。実験では一般的に広く使われているプロセッサを搭載した計算機9台を、PC クラスタとして構成した。すべての計算機は Gigabit Ethernet のポートを持っており、計算機間の通信は Gigabit Ethernet を介して行われる。

### 6.2 分割方法の比較

実験を行ったときのパラメータを以下に示す。

Sugarscape 環境については、単体での計算プログラムでは、 $180 \times 180$  の Sugarscape モデルを採用する。また、並列計算プログラムでは、マシン台数と分割方法により、 $180 \times 180$  の区画を正方形型に4分割した Sugarscape モデル ( $90 \times 90 \times 4$  台)、 $180 \times 180$  の区画を正方形型に9分割した Sugarscape モデル ( $60 \times 60 \times 9$  台)、 $180 \times 180$  の区画を短冊型に4分割した Sugarscape モデル ( $45 \times 180 \times 4$  台)、 $180 \times 180$  の区画を短冊型に9分割した Sugarscape モデル ( $20 \times 180 \times 9$  台)の計4つのモデルを採用する。単体での計算プログラム、並列計算プログラム共に、Sugarscape 環境の資源量の分布は、図4と図5の2つを用いる。

初期エージェント数は、単体での計算プログラム、並列計算プログラム共に、4320 とする。

その他文献[5]で示されたパラメータを用いる。

#### 6.2.1 エージェントの推移

資源量の配置を図4とした場合の、エージェントの推移を図6に示す。図6には、単体での結果と4種類の分割モデルの結果を示したが、いずれも同様の推移を示した。

5つの Sugarscape モデルのそれぞれの実行時間を図7に示す。図7より、単体での計算プログラムに比べ、並列計算プログラムの方が、実行時間が早くなっており、並列処理の効果が得られている。しかし、分割方法に着目してみると、ほとんど差がない。これは、正方形型の通信時間と短冊型の通信時間がほぼ等しいからである。

#### 6.2.2 エージェントの振る舞いの変化

資源量の配置を図5に変更した実験を行った。この実験では、単体のマシンで  $180 \times 180$  のシミュレーションを実行することができなかった。理由とし

エージェント数 ①:OA ②:VC ③:LO

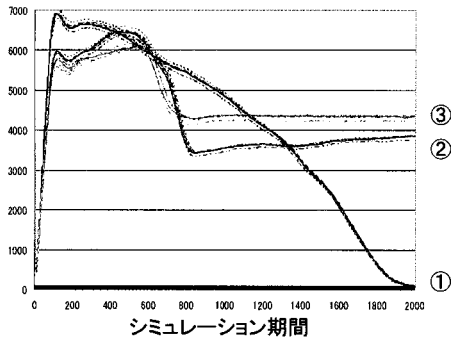


図6 エージェントの推移 (エージェントの推移)

実行時間

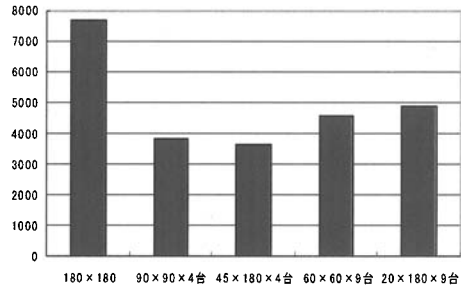


図7 実行時間(エージェントの推移)

エージェント数 ①:OA ②:VC ③:LO

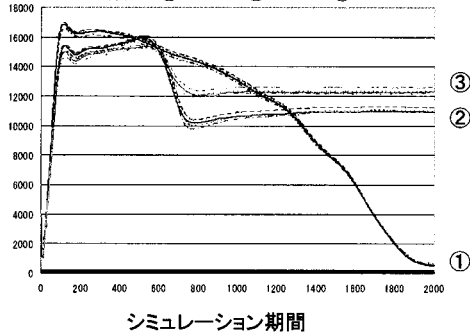


図8 エージェントの推移 (エージェントの振る舞い)

実行時間

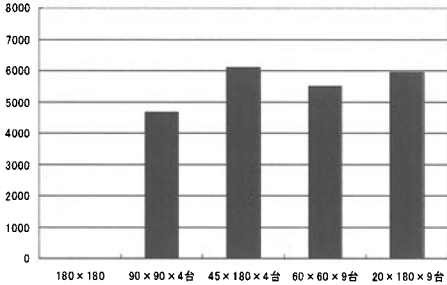


図9 実行時間 (エージェントの振る舞い)

では、エージェントの数が急激に増えたために、メモリ容量が不足してしまったためである。

エージェントの推移を図8に示す。図8より、資源量を大きくし、エージェントの振る舞いを变化させた場合にも、エージェントの推移にほとんど誤差がないと推測できる。

並列計算プログラムの各 Sugarscape モデルについての実行時間を図9に示す。図9より、図7の時と比べて、正方形型の方が短冊型に比べて、実行時間が短縮できた。

### 6.3 環境情報量の変化がもたらす影響

実行時間の差をもっと顕著に表すために、6.2.1の実行結果1のときに使用したパラメータよりも、さらに大規模化な実験を行った。大規模化に伴い変更した実験パラメータは、以下の通りである。

Sugarscape 環境については、単体での計算プロ

グラムでは、評価実験1の Sugarscape モデルの2倍の、360×360の Sugarscape モデルに変更した。

また、並列計算プログラムでは、マシン台数と分割方法により、360×360の区画を正方形型に4分割した Sugarscape モデル (180×180×4台)、360×360の区画を正方形型に9分割した Sugarscape モデル (120×120×9台)、360×360の区画を短冊型に4分割した Sugarscape モデル (90×360×4台)、360×360の区画を短冊型に9分割した Sugarscape モデル (40×360×9台)の計4つの Sugarscape モデルに変更した。

初期エージェント数は、単体での計算プログラム、並列計算プログラム共に、評価実験1の2倍の8640に変更した。

エージェントの最大視野もまた、評価実験1の2倍である12に変更した。

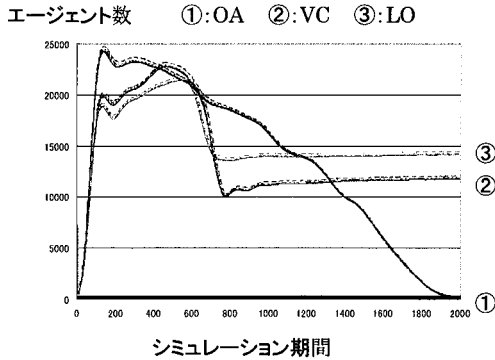


図 10 エージェントの推移 (環境情報の変化)

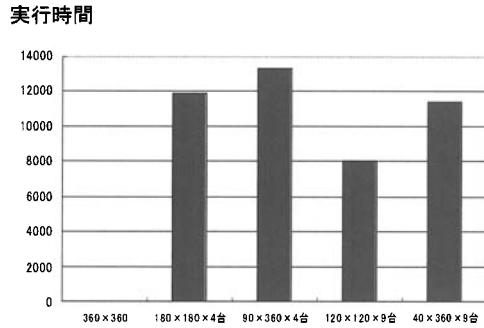


図 11 実行時間 (環境情報の変化)

### 6.3.1 実行結果

MAS を大規模化したため、 $360 \times 360$  の Sugarscape モデルは、単体の計算機であるため、メモリが足りずに実験を行うことができなかった。並列計算プログラムの Sugarscape モデルの実行結果のみを示す。

エージェントの推移を図 10 に示す。図 10 より、MAS の問題を大規模に拡張したときにもほとんど誤差がないことがわかる。

並列計算プログラムの各 Sugarscape モデルについての実行時間を図 11 に示す。図 11 より、正方形の方が短冊型より実行時間の短縮が可能であることがわかる。さらに、4 台の場合より 9 台の場合の方が実行時間の短縮が可能であることもわかる。これは環境が大規模になったことにより、並列処理の効果が得られるためである、と考えられる。

## 7. まとめ

MAS におけるシミュレーションの並列処理方式を比較した。具体的には、大規模 MAS をグリッド・コンピューティングに適応させ、処理を分散させる際に、並列処理方式の違いにより MAS にどのような影響が与えられるかについて、実証実験を行った。その結果、エリア分割方式では、正方形型を用いる方が短冊型を用いるときよりも、大規模 MAS に適していると判断できる。

我々の目標は、環境の大きさをさらに拡大し、マシンの台数を増やすことで、さらに大規模な MAS の実行を可能にすることである。さらに、エリア分

割方式とは異なる分割方式であるエージェント分割方式を、GridRPC に適用させ異なる情報の分割に基づく実装上の違いについて明らかにする。

## 謝辞

本研究の一部は、文部科学省社会連携研究推進事業(平成 17 年度～平成 21 年度)による私学助成を得て行われた。

## 参考文献

- [1] Joshua M. Epstein, Robert Axtell (服部 正太, 木村 香代子訳), 人工社会 - 複雑系とマルチエージェント・シミュレーション -, 共立出版 (1999).
- [2] 西崎 一郎, 上田 良文, 佐々木 智彦, “慈善くじによるグローバル・commons の保全のための資金調達と人口社会モデルを用いたシミュレーション分析”, システム制御情報学会論文誌, Vol.17, No.7, pp.288-296 (2004).
- [3] 中島 悠, 椎名 宏徳, 山根 昇平, 八槇 博史, 石田 亨, “大規模マルチエージェントシミュレーションにおけるプロトコル記述と実行基盤”, 電子情報通信学会論文誌, Vol.J89-D, No.10, pp.2229-2236 (2006).
- [4] Ian Foster, Carl Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, Inc (1999).
- [5] 森下 仙一, 蟻川 浩, 村田 忠彦, “環境改善資金調達マルチエージェントシミュレーションの MPI による並列計算”, 第 22 回 フェジシステムシンポジウム講演論文集 (CD-ROM, ISSN 1341-9080), pp.107-112, September (2006).
- [6] Message Passing Interface Forum, <http://www.mpi-forum.org>
- [7] LAM/MPI Parallel Computing, <http://www.lam-mpi.org>
- [8] MPICH, <http://www-unix.mcs.anl.gov/mpi/mpich1>
- [9] GridMPI, <http://www.gridmpi.org/index.jsp>