PG Lab ディスカッションペーパーシリーズ
第13号　2007年3月

ISSN 1880-9146

Discussion Paper Series
No.13 March, 2007

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

文部科学省私立大学社会連携研究推進拠点
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
Suita, Osaka 564-8680 Japan
URL : http://www.pglab.kansai-u.ac.jp/
e-mail : pglab@jm.kansai-u.ac.jp
tel. 06-6368-1228
fax. 06-6330-3304

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からのお願い

本ディスカッションペーパーシリーズを転載，引用，参照されたい場合には、
ご面倒ですが、弊センター（pglab@jm.kansai-u.ac.jp）宛に
ご連絡いただきますようお願い申しあげます。

Attention from Policy Grid Computing Laboratory, Kansai University

Please reprint, cite or quote WITH consulting Kansai University Policy Grid Computing Laboratory (pglab@jm.kansai-u.ac.jp).

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[1]，蟻川 浩[2]，村田 忠彦[1 2]

Comparisons of parallel processing method in Multi-Agent Simulation

Sen-ichi Morishita[1], Hiroshi Arikawa[2], Tadahiko Murata[1 2]

概要

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.

Abstract

This paper describes a distributed multi-agent simulation using Message Passing Interface. The data distribution method can be categorized into two types: square type and 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 evaluate 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.

キーワード：大規模マルチエージェントシミュレーション，並列処理，メッセージ・パッシング・インターフェイス，分割方法

Keyword: Large Scale Multi-Agent Simulation, Parallel Processing, Message Passing Interface, Distribution Method

1 関西大学　総合情報学部

2 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

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．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 Intarface 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，複数台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 のようになる．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を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

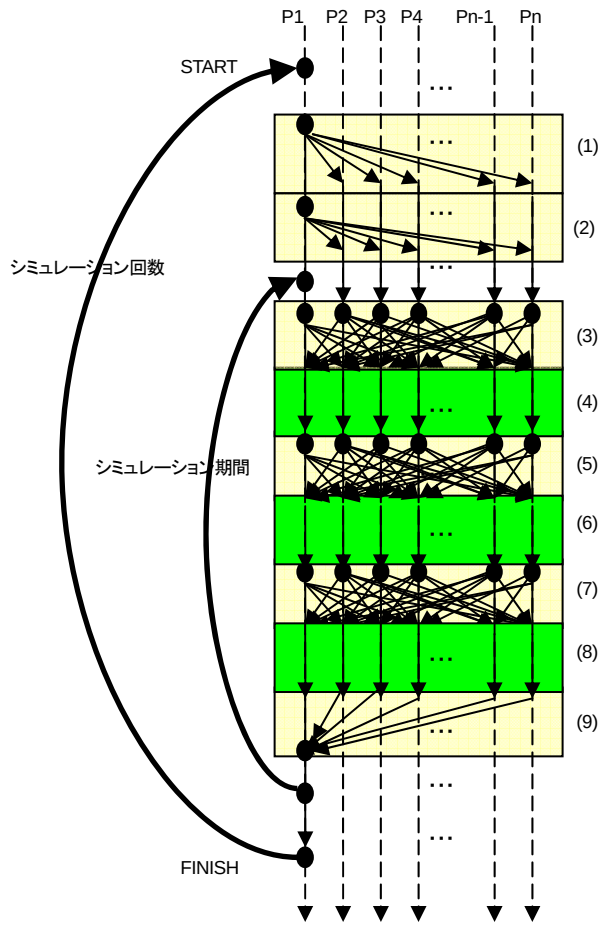

図1　MPI を用いた並列計算の動作フロー

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量の削減を行い，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．

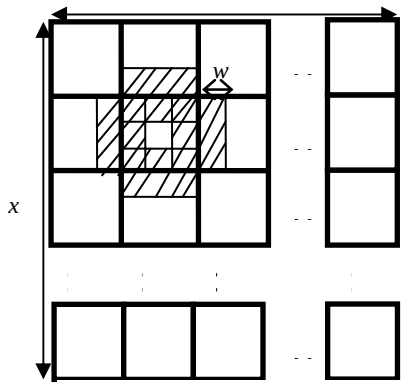

図 2　正方形型の通信

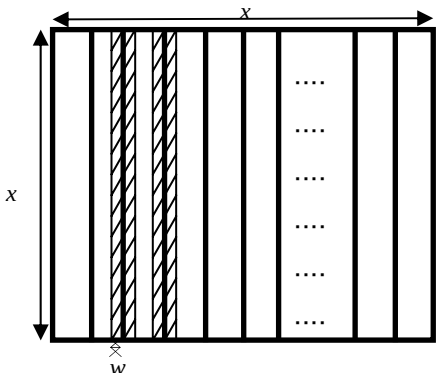

図 3　短冊型の通信例

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，全プロセッサ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 および 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)のように説明できる．

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

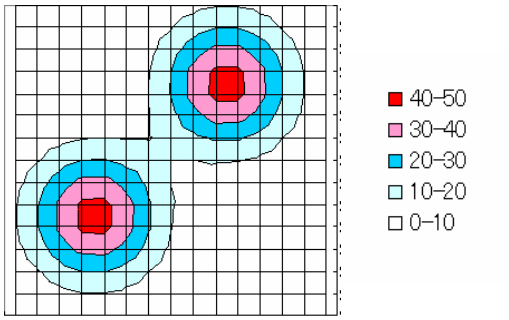

図 4　資源量の分布 1

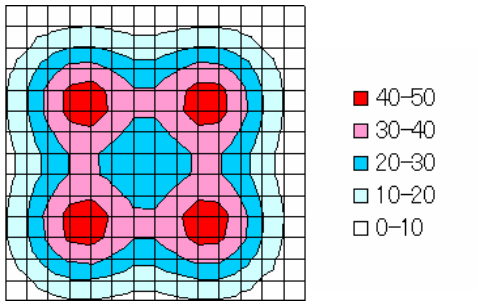

図 5　資源量の分布 2

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×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)のように説明できる．

$$T_r \cong 2c_1 + 4wxc_2 + 2a'c_3 \tag{2}$$

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，データ転送量が少ないため，通信確立時間が短い短冊型の通信時間が短くなる．一方，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．

表 1　平均 2 乗誤差(エージェントの推移)

	OA	VC	LO
4台(短冊型)	299.32	1244.41	2640.16
9台(短冊型)	8113.17	8478.92	3949.73
4台(正方形型)	1227.76	1934.51	2231.47
9台(正方形型)	9726.32	9489.23	17475.06

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，180×180 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，180×180 の区画を正方形型に 4 分割した Sugarscape モデル (90×90×4 台)，180×180 の区画を正方形型に 9 分割した Sugarscape モデル (60×60×9 台)，180×180 の区画を短冊型に 4 分割した Sugarscape モデル (45×180×4 台)，180×180 の区間を短冊型に 9 分割した Sugarscape モデル (20×180×9 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

資源量の配置が図 4 におけるエージェントの推移を図 6 に示す．また，表 1 に並列処理によるシミュレーション結果と単体の計算機によるシミュレーション結果との平均 2 乗誤差を示す．表 1 より，許容できる範囲の誤差であり，かつ図 6 より同様の軌跡を描くことを確認したため，シミュレーション結果に大きな影響はない.

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．この条件においては，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 180×180 のシミュレーションを実行することができなかった．理由としては，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変化させた場合にも，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.

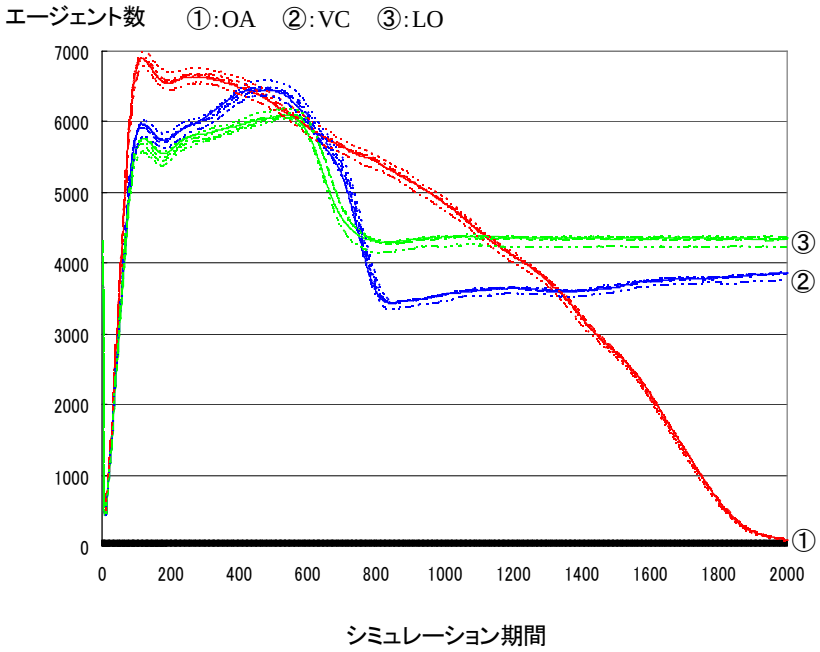

図６　エージェントの推移 (エージェントの推移)

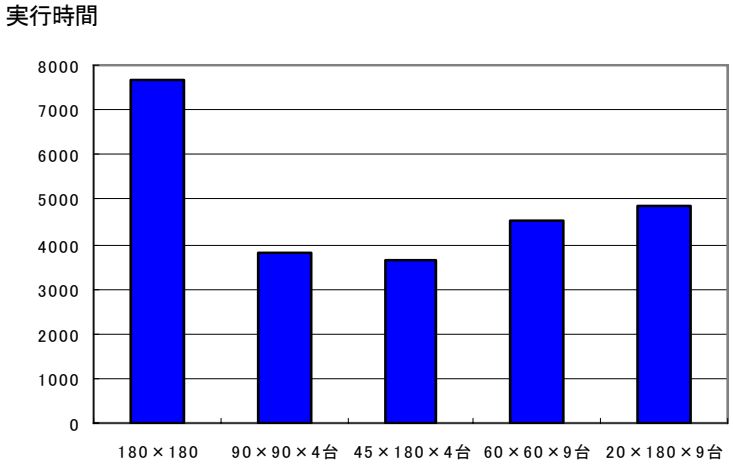

図７　実行時間(エージェントの推移)

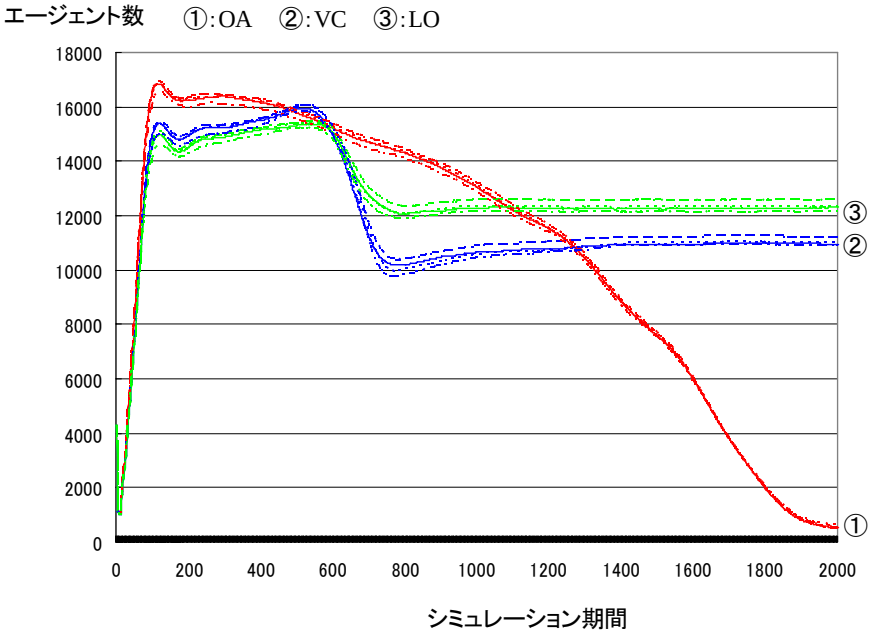

図 8　エージェントの推移 (エージェントの振る舞い)

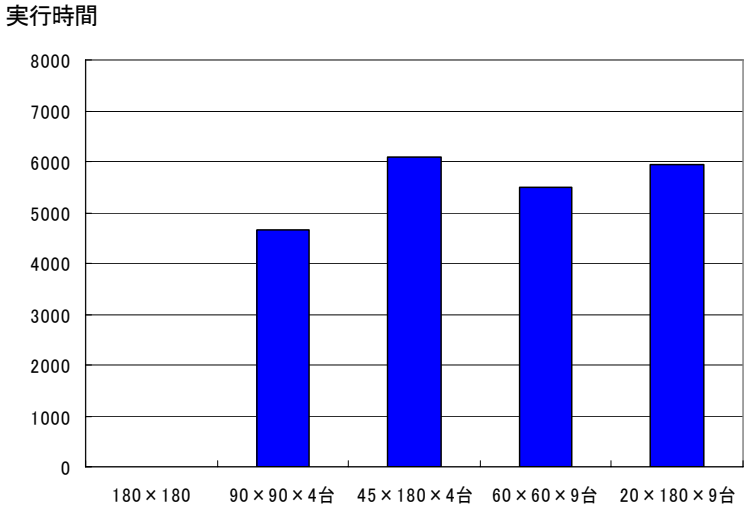

図 9　実行時間 (エージェントの振る舞い)

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

実行時間の差を顕著に表すために，6.2.1 での実験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 の区間を短冊型に 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.

MAS のパラメータを変更した際の実験結果を図 10 および図 11 に示す。360×360 の Sugarscape モデルは，単体の計算機ではメモリ不足によりシミュレーションプログラムが実行できなかった. よって，並列計算プログラム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 のパラメータを大規模問題に拡張して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，正方形型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，エージェント数が増加し，エージェントの意思決定にかかる処理時間が大きくなったこと，そしてその分散処理効果が台数の効果として表れた，と考えられる. 特に，正方形型かつ 9 台の計算機を用いた場合における実行時間の短縮が顕著に表れている. より精細な実験が必要であるが，シミュレーションモデルが大規模になるほど正方形型の分割方式がよいといえる.

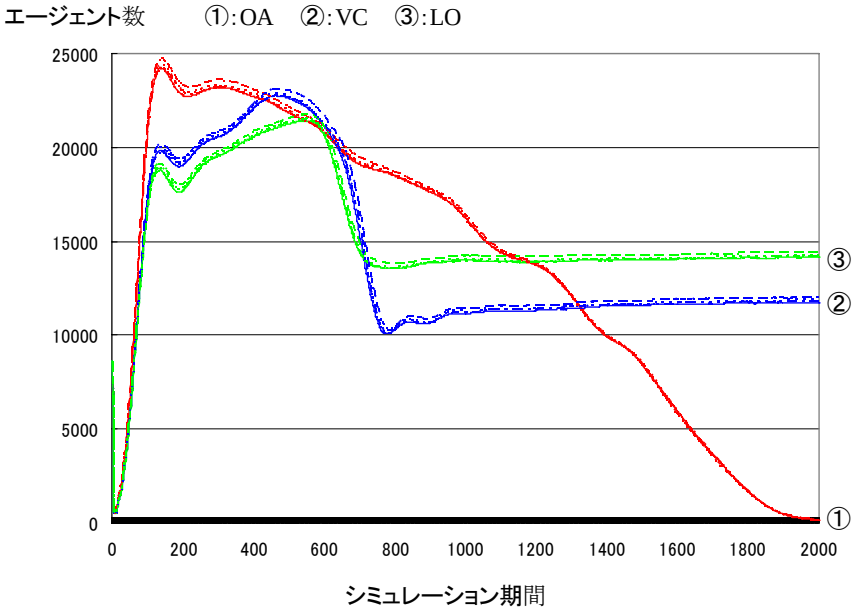

図 10　エージェントの推移 (環境情報の変化)

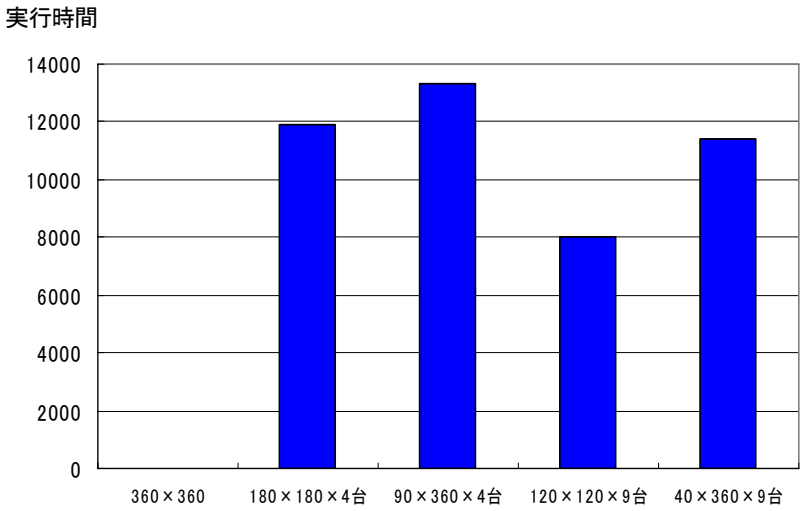

図 11　実行時間 (環境情報の変化)

7. 今後の課題

今回は環境情報に対して正方形型に分割する場合と短冊型に分割する場合についての比較を行った．環境情報の分割による並列処理の場合，各計算機におけるエージェント数が均一にならず，エージェントがまったく存在しない，すなわち演算処理がない計算機もある．すべての計算機の処理負荷が均一であり，かつ並列処理部分の演算が同時刻に終わることが理想である．

そこで並列処理の新たな分割方法として，エージェント数を均一にするための方法を検討する必要がある．具体的には，各計算機の演算領域の再分割を行い，各マシンに割り当てる領域の大きさを変更する方法(領域変更型)が考えられる．今後は領域変更型と正方形型を比較し，どちらの分割方法が大規模シミュレーションに適しているのかを検討する．

環境分割方式においてどの分割方法が適しているかは，資源の配置などのシミュレーション条件に影響される．シミュレーション条件に応じて適切な並列分割方法を判断するための指標を検討することを考えている．

8. 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．

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・コモンズ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 (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 ファジィシステムシンポジウム講演論文集 (CD-ROM, ISSN 1341-9080)，pp.107-112 (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