関西大学学術リポジトリ
Kansai University Institutional Repository

Title	グリッドコンピューティングの政策立案支援への応用
Author(s)	村田，忠彦
Citation	セミナー年報，2007：99-112
Issue Date	2008-03-31
URL	http://hdl.handle.net/10112/531
Rights	
Type	Article
Textversion	publisher

Kansai University
http://kuir.jm.kansai-u.ac.jp/dspace/

グリッドコンピューティングの政策立案支援への応用

村　田　忠　彦
政策グリッドコンピューティング実験センター長
総合情報学部准教授

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、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もっている。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]。

MASを単一の計算資源で処理する場合、エージェント数が増加するにつれて、エージェントの意思決定にかかる時間が長くなる。より現実的な環境をシミュレーションするためにはシミュレーション時間の短縮が必要であり、グリッド・コンピューティングの環境で実行できることが望ましい。

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として仮想的に構成するための枠組みである[3]。この枠組みに基づいてシミュレーションを行う場合、エージェントの意思決定処理に関してどのように分散処理させるかを考慮しなければならない。

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では、MASによる大規模シミュレーションの実現と計算処理の高速化について研究開発を進めている。本稿では、MASを実現するモデルとして知られているSugarscapeモデルを取り上げ、MASによる並列処理の実現とMessage Passing Interface（MPI）を用いたMASツールの実装上の課題について述べる。

2．Sugarscapeモデルに基づく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、西﨑らによって考案されたSugarscape上の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Joshua M. EpsteinとRobert Axtellが開発したSugarscapeモデル[1] と呼ばれる人工社会モデルを環境改善資金調達シミュレーションに

応用したものである。

2.1　Sugarscapeモデル

Sugarscapeモデルは多数のエージェントが存在し、各エージェントは内部状態と行動ルールをもっている。エージェントの内部状態は、生涯を通じて不変である視力、代謝率、性別、初期財産、寿命、交配可能年齢と、状況によって変動する現在位置、財産、年齢とがある。代謝率とは、1期間毎にエージェントが消費する資源量のことである。初期状態では、性別は男女数がほぼ等しくなるように定められ、各エージェントには初期財産が与えられる。エージェントは資源収集を行うと同時に代謝も行い、代謝量を差し引いた分の資源が財産として蓄積されていく。

また、Sugarscape環境は格子状の2次元空間で表現され、上下と左右がそれぞれ連続するトーラス空間になっている。この2次元空間上にはエージェントが生きるために必要な資源が配置されており、エージェントに収集されても再生する。2次元空間上の各地点は、資源の現在量と最大容量と、エージェントの存在有無を要素としてもつ。

エージェントと環境は、それぞれルールをもっており、それらのルールに従ってエージェントは自律的に行動する。各エージェントは1期間毎に2次元空間上の直交する4方向を視力に基づいて見渡し、最も資源が多く、他のエージェントがいない地点に移動する。そして、その地点にある資源を全て収集し、代謝しながら財産の増減を行う。エージェントの財産が0以下となった場合は、寿命に達していなくても死亡する。また、エージェントの移動後、そのエージェントが交配可能年齢であること、その時点での財産が初期財産より多いことを交配の条件とし、その条件を満たしているエージェントは、交配条件を満たした近隣のエージェントと交配を行う。交配時には両親の初期財産の平均を子供に与える。環境におけるルールとして、2次元空間上の各地点の資源は、最大容量に達していない場合、単位期間あたり一定量ずつ再生される、と設定する。

2.2　環境改善資金調達モデル

2.2.1　汚染物質の発生

環境改善資金調達モデル[2]ではSugarscape環境上に汚染物質を発生させることにより公害を表現する。エージェントの資源収集を生産活動とみなし、資源収集によって汚染物質が生じるものとする。資源がm単位収集される時、汚染物質がλm生成されるとする。ある地点の時刻tにおける汚染物質の合計量P_tを式(1)に示す。

$$P_t = P_{t-1} + \lambda m \qquad (1)$$

汚染物質は資源の再生率に影響を与える。蓄積された汚染物質量に応じて、各地点の資源再生率が変化する。したがって、汚染物質が増えるにつれ、エージェントの生存に必要な資源が

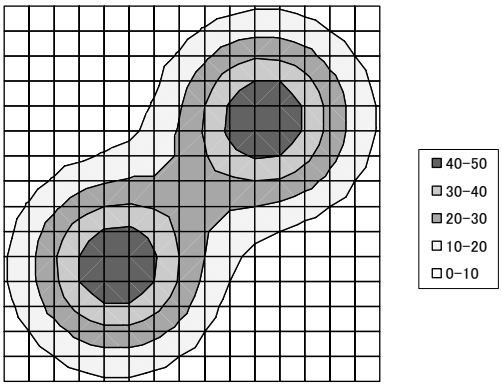

図１　資源量の分布

表１　公害発生率

Accumulated pollutants P_t	Generation rate λ
Under 50 units	0.001
Over 50 units	0.0005

再生されなくなる。

現実社会では環境汚染が著しい場合、汚染を軽減するような措置をとる。これを考慮し、汚染物質量に応じて表１のような公害発生率λを設定する。ここで、汚染物質量がある一定量を超えると公害発生率が半分になっているのは、ある一定量以上の公害が発生した場合、なんらかの特別の処置がとられるものとしているためである[3]。

各エージェントの環境問題に対する関心度を長さ20で０と１から構成されているタグ文字列を用いて表現する。タグ文字列中に１のタグ数が多いエージェントほど環境問題に対する関心度が高いことを意味し、初期状態ではエージェントの各タグ位置にランダムに値を割り当てる。このタグ文字列中の１のタグの割合に応じて、環境問題に対して高い関心があるエージェント（タイプ１）、環境問題に対して通常の関心をもつエージェント（タイプ２）、環境問題に対してあまり興味のないエージェント（タイプ３）として分類する。

環境改善資金調達モデルでは、エージェント間の相互影響を実現するため、環境に対する意識をエージェント間で伝播させる。そこで、環境に対する意識が高いタイプ１のエージェントは他のエージェントに対して環境改善活動を勧める行動をとる。これを実現するために以下のタグフリップルールをエージェントの行動ルールに取り入れる。タイプ１のエージェントは近隣のエージェントの中からランダムに１人選択し、タグ位置をランダムに選択する。もし選択したタグ位置でタイプ１のエージェントが１で相手のタグが０ならば、相手のタグを１に変更する。

2.2.2　募金システム

募金システムでは、タイプ１は３口（15資源）、タイプ２は２口（10資源）、タイプ３は１口（５資源）を寄付し、集まった資源は全て環境改善費用に割り当てられる。募金システムを導入した場合、以下のような手順でシミュレーションを行う。

Step 1　各エージェントの募金することの効用値を計算する。
Step 2　各エージェントは募金するかどうかを決定する。
Step 3　実際に集まった資源が環境改善費用となり、それに応じてSugarscape環境上から汚染物質を取り除く。各エージェントは自身が募金したにも関わらず、募金への参加率が低い場合はフリーライダー問題に直面し、環境問題に対する関心度を低下させる。

2.2.3　慈善くじシステム

慈善くじシステムでは、エージェントはくじに当せんすることに対する期待と環境問題に対する関心度に従って確率的に慈善くじの購入を行い、集まった資源の一部をくじの当せん金として投資者に与え、残りを環境改善費用に割り当てる。慈善くじシステムを導入した場合、以下のような手順でシミュレーションを行う。

Step 1　各エージェントの慈善くじを購入することの効用値を計算する。
Step 2　各エージェントは慈善くじを購入するかどうかを決定する。
Step 3　慈善くじを実施する。今期に購入された総口数をS_t、１口あたりの当せん確率をδとし、実際に集まった資源X_tの$100q$%を当せんしたエージェントに均等配分する。つまり、１口あたりの当せん額を$qX_t/\delta S_t$とする。
Step 4　次回の予定くじ購入額x_tを変動させる。予定くじ購入額の初期値は募金の場合と同じである。
Step 5　実際に集まった資源X_tの$100-(1-q)$%が環境改善費用となり、それに応じてSugarscape上から汚染物質を取り除く。
Step 6　各エージェントがフリーライダー問題に直面した場合は環境問題に対する関心度を低下させる。

3．環境改善資金調達シミュレーションの並列処理への拡張

環境改善資金調達シミュレーションをグリッドコンピューティングの環境に適用するにあたり、並列処理可能なかたちにアルゴリズムを改良しなければならない。ここでは、Sugarscape環境の分割および隣接区域との情報交換、エージェントの衝突回避に関して説明する。

3.1　Sugarscape環境の分割

MASでは、エージェントの数の増加により、エージェントの意思決定にかかるシミュレーション時間が増加し、シミュレーション全体としての実行時間が増加する。グリッド・コンピューティングの環境でシミュレーションを行う目的はシミュレーション時間の短縮であるため、エージェントの意思決定にかかる時間を短縮することが要求される。そこで、Sugarscape環境を複数の区域に分割し、各区域におけるエージェント意思決定の処理を各コンピュータに行わせる。

3.2　隣接区域との情報交換

Sugarscape環境を複数の区域に分割した場合、区域間の移動を定義する必要がある。これは、エージェントが初期区域だけでしか行動できなくなるためである。Sugarscape環境の分割数が増加するにしたがい、エージェントの移動範囲が狭くなるため、環境を分割しない場合と比べエージェントの分布が大きく異なる。よって、従来のSugarscapeモデルとの結果と大幅にかけ離れてしまうことになる。そこで、各区域でエージェントが移動する前に隣接区域と通信を行い、隣接区域の環境情報を付加する。その後、各区域でエージェントの移動を行う。区域を出たエージェントの情報は蓄積しておき、全てのエージェントの行動後、蓄積されたエージェント情報を該当する隣接区域に送信し、移動前の区域のエージェント情報を削除する。これにより、各エージェントの移動毎に通信を行う場合に比べ通信回数が少なくなる。

図2に上記の隣接区域のエージェント情報交換の一例を示す。この図では、1つのSugarscape環境を9区域に分割した。Sugarscape環境はトーラス状の2次元空間なので、たとえば区域7においては、斜線部分の環境情報を必要とするので、近隣の区域を担当するプロセッサから環境情報を取得する。全エージェントの移動後は、斜線部分に入ったエージェント

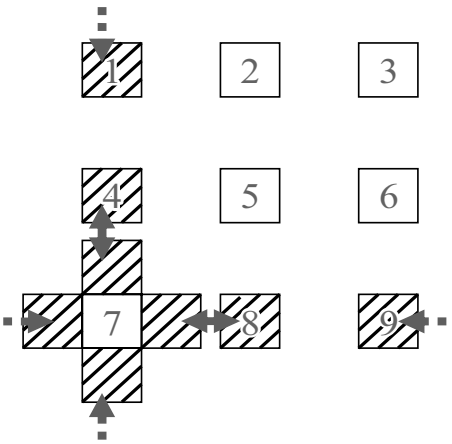

図2　隣接区域との通信例

の情報を該当区域に送信する。そして、隣接区域に出たエージェントの情報を区域７から削除する。

3.3　エージェント衝突時の処理

隣接区域との通信により、区域を移動してきたエージェント情報を受信し、その情報に基づいて各エージェントを予定移動地点に配置する。この場合エージェント同士が衝突するという問題が発生する。１台の計算機上で行う場合、同一区域内（同一計算機内）でエージェントが移動する時には必ず他のエージェントがいない地点を選択する。しかし、複数台の計算機に分散処理させる場合、区域を移動してきたエージェントを配置する前に、移動先の区域に存在するエージェントが先に選択する状態が発生する。この状態が衝突である。

エージェントの衝突に関して、並列処理型MASに拡張するにあたり、先にその区域にいたエージェントを優先するルールを採用する。衝突時において他の区域から移動してきたエージェントは移動予定地点の近隣をランダムに選択して移動する。もちろん、元の区域に戻すことも考えられるが、エージェントの移動のために、再度衝突判定を行う必要があるため、区域を移動してきたエージェントは元の区域に戻るという選択はしない。

4．MPIを用いた並列処理の実現

MASの並列処理を可能にするための方法について説明する。具体的には、MPIライブラリを用いた場合のMASの動作フロー、MPIライブラリを利用して実装する際に工夫した点について述べる。

4.1　MPI

MPI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。1992年に並列計算機ベンダーと学術組織など40の組織で構成されたMessage Passing Intarface Forum[4] にて通信ライブラリの標準インターフェイス規格（API）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[5]、LAM[6]、GridMPI[7] がある。PCクラスタを代表とする並列計算環境が容易に手に入れられる昨今、並列処理プログラミングを実現するための１インターフェイスとして事実上の標準となっている。

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」の関数で構成される。これらの関数を組み合わせて並列プログラムを実装する。

4.2　MPIを用いたMASの動作の流れ

MPIを用いたMASでは、複数のプロセッサを用いてシミュレーションを行う。本稿では、Sugarscape環境を分割する方式を採用するため、プロセッサとプロセスの実行との関係は図3のようになる。図3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P9はプロセッサを意味する。以下では、1試行における各番号の計算処理内容を説明する。

処理（1）：プロセッサ1によって分割したSugarscape環境情報（各区域の情報）とエージェント初期情報をプロセッサ2からプロセッサ9に分配する。

処理（2）：各プロセッサがエージェント意思決定処理をするのに必要となる隣接区域の情報を交換する。

処理（3）：エージェントの意思決定処理を行う。ここでの処理は各プロセッサにおいて独立に実行される。

処理（4）：各プロセッサがもつSugarscape環境情報の領域外に移動するエージェントの情報を移動させる。エージェント毎に移動先の領域を担当するプロセッサにエージェント情報を通信する。

処理（5）：エージェントの衝突判定を行う。3.3で示したルールに従ってすべてのエージェントの位置を確定する。ここでの処理は各プロセッサで独立に実行される。

処理（6）：各領域における資源回復処理を行う。

処理（7）：各領域のデータをプロセッサ1に収集し、プロセッサ1はステップごとのデータ集計を行う。

ここで、処理（2）～処理（7）の繰り返しをシミュレーション期間とする。また、シミュレーション実施者が予め決定したシミューレション期間を1試行とし、統計的に十分な数の試行を行うことでシミュレーション結果を得る。

図3において、プロセス間通信が必要となる箇所は処理（1）、処理（2）、処理（4）、処理（7）の部分である。これらの処理においてMPIライブラリが用いている。以下で具体的に説明する。

処理（1）におけるプロセス間通信の処理：

処理（1）ではプロセス毎に異なるデータを送信する。MPIではあるプロセスから他のプロセスに送信することになるため、送信側となるプロセッサ1ではMPI_Send()を、受信側となるプロセッサ2からプロセッサ9はMPI_Recv()を用いる。

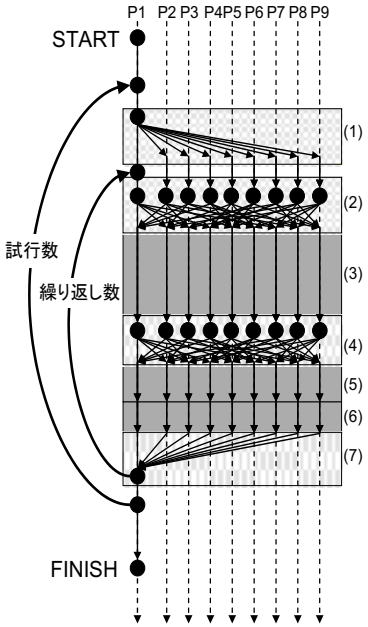

図３　MPIを用いた並列計算の動作フロー

処理（2）と処理（4）におけるプロセス間通信の処理：

処理（2）と処理（4）では隣接した区域の情報を必要とする。例えば、プロセッサ１はプロセッサ２、３、４、７にというように、隣接した情報をもつプロセッサからの情報を送受信する必要がある。送信側と受信側が対応しているため、MPI_Sendrecv()を用いる。

処理（7）におけるプロセス間通信の処理：

処理（7）では各プロセッサで独立に行った結果を集計する処理である。環境改善資金調達MASでは、各プロセッサで得られた結果を合算する必要がある。MPIではリダクション操作と呼ばれる、通信と演算を同時に行う方法をMPI_Reduce()として提供されている。処理（7）ではMPI_Reduce()を用いて、すべてのプロセッサから得られた結果を合算し、プロセッサ１が得るようにした。

５．比較実験

SugarscapeモデルにおけるMASの並列処理手法について有効性を確認するため、１台のプロセッサで動作するMASプログラムと並列処理版MASプログラム（プロトタイプ）を実装し、PCクラスタを用いて実際にシミュレーションを行った。ここでは、実験結果の類似性を検討する。なお、本稿では資金調達のための社会システムを導入しない場合の結果について示す。

5.1　シミュレーション時のパラメータ

Sugarscape環境情報については、単体での計算プログラムでは150×150のSugarscapeモデルを採用する。また、提案する並列計算プログラムでは図2に示した150×150の区画を9区域に分割したSugarscapeモデル（50×50× 9台）のSugarscape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ともに、初期エージェント数は3600とする。また、資源の現在量が最大容量に満たない場合は1期間に最大容量×再生率分まで回復するというルールを用いる。

5.2　並列計算環境

1試行あたりのシミュレーション期間は2000期間とする。シミュレーションの試行回数は50試行とし、シミュレーション結果はその平均とする。

エージェントに関係するパラメータ設定は以下のとおりである。エージェントの内部状態としては、視力は1から6まで、代謝率は15から35まで、寿命は60から100まで、交配開始年齢は12から15まで、交配終了年齢は男なら40から50まで、女なら30から40までとする。初期エージェントの初期財産は代謝率の1倍から3倍とする。

5.3　シミュレーション結果の比較

図4～図6に、並列処理を行わない場合（Not Distributed: ND）と提案手法による場合（Distributed: D）のエージェント人口、汚染物質量、総資源量の推移を示す。また、図4～図6のグラフにおいて、それぞれ200期間毎の2乗誤差を計算し、50試行に対して得られた平均2乗誤差とその平方根を表3に示す。グラフの推移や平均2乗誤差の結果から、分割を行った場合と行わなかった場合の推移の類似性が高く、並列化によるシミュレーション結果への影響は少ない。

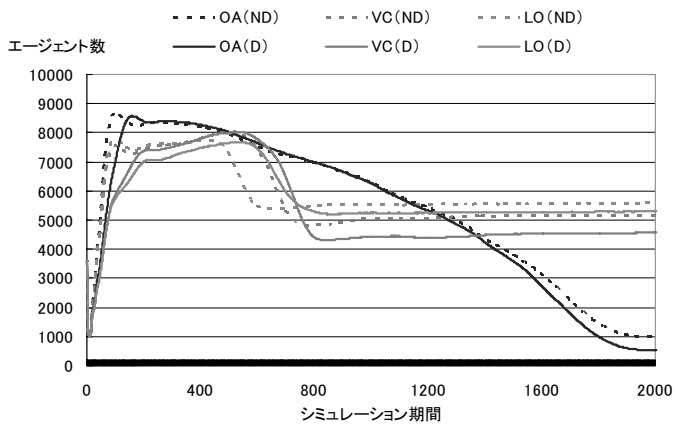

図4　エージェント人口の推移

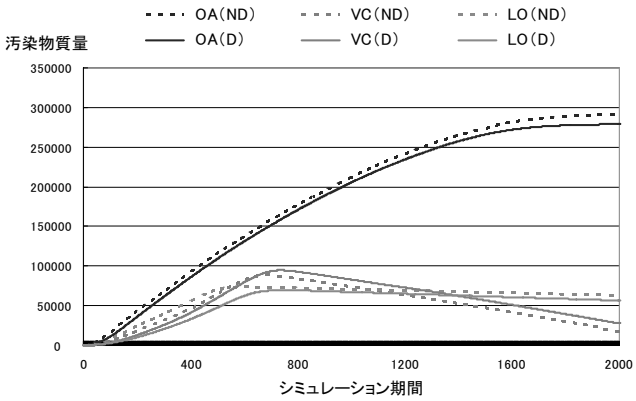

図５　汚染物質量の推移

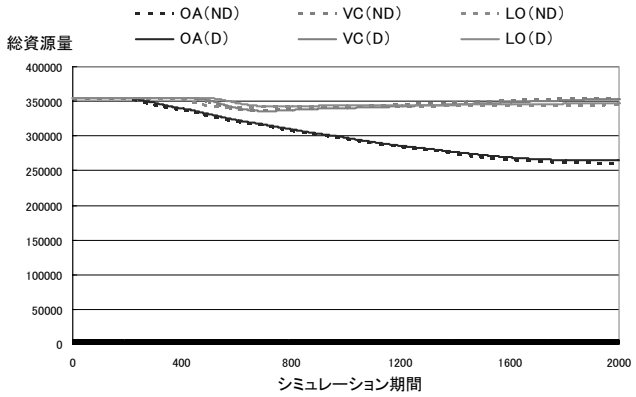

図６　総資源量の推移

表３　シミュレーション結果の誤差

	平均２乗誤差			平方根		
	OA	VC	LO	OA	VC	LO
人口の推移	64840.7	259475.7	483937.2	254.6	509.3	695.6
汚染物質量推移	69687928.6	66554297.9	103686962.3	8347.9	8158.0	10182.6
総資源量の推移	19991624.2	14792641.4	9773633.3	4471.1	3846.1	3126.2

６．シミュレーション実行時間短縮に向けた解決方法

以上において、MPIによるMASの並列処理を実現することができた。これにより、大規模なシミュレーションを可能にする見込みは得られた。並列処理可能なMASプログラムを実装する意義は、大規模なシミュレーションを可能にすることのほかに、シミュレーション時間を短縮することも含まれる。MPIライブラリによるMASプログラムの実装により、我々はシミュレーション時間の短縮が可能になると思っていた。ところが、一般論としてのMPIプログラミン

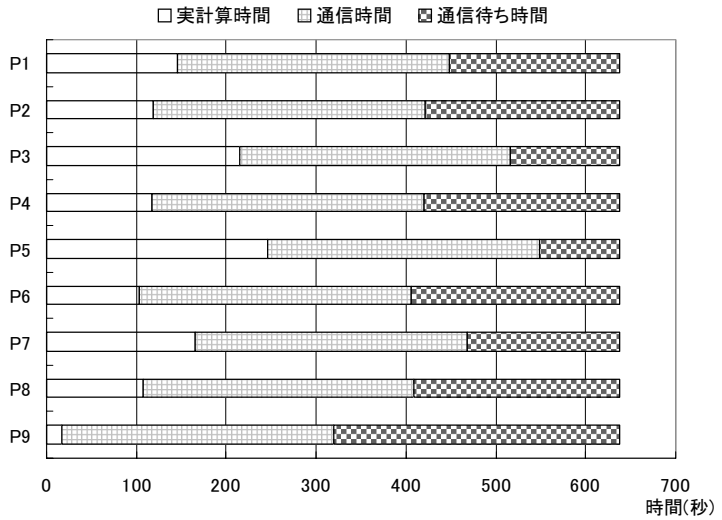

図７　各プロセッサにおける実行時間の内訳

グだけでは時間短縮を可能にするMASプログラムの実装は困難であることをプロトタイプ版のMASプログラムの実装を通じて知ることができた。

図７は政策グリッドコンピューティング実験センターが所有するPCクラスタを用いていくつかの改良を施したMPIライブラリによるMASプログラムの実計算時間、通信時間、通信待ち時間の内訳である。９台のプロセッサエレメントを用いて並列処理を行った結果を示す。PCクラスタを構成するプロセッサエレメントの性能については以下のとおりである。プロセッサはIntel Pentium４　３EGHz、２GByteの主記憶を搭載している。また、プロセッサ間の通信はGigabit　Ethernetによるものである。時間の計測においては、MPIで定義されているMPI_Wtime()という関数を用いて計測した。MPI_Wtime()はマイクロ秒の精度で計測できるライブラリであり、精度としては１メソッドが呼び出される際の実行時間を測定することができる。図７の結果から、通信時間が全体の実行時間の47%占めている。エージェントの意思決定の計算よりもデータの同期待ち、データの転送といった通信にかかる時間のほうが大きいことが判明した。

７．シミュレーション結果の可視化

政策グリッドコンピューティング実験センターでは、政策立案支援に適したツールの開発を最終目標として、グリッドコンピューティングを指向したMASプログラムの開発とともに、シミュレーション結果の可視化についても研究を進めている。ここでは、Sugarscapeモデルに基づくMASプログラムに適した可視化ツールの実装例と、環境改善資金調達シミュレーションの可視化例について示す。

MASを用いたシミュレーションでは個々のエージェントの自律的な振る舞いが環境といったモデル全体にどのように影響を及ぼすかを知ることが目的である。従来は、図4～図6に示したように、モデル全体としての視点、いわゆるマクロレベルでの結果出力に注力されてきた。一方で、エージェントの振る舞いそのものの観察をする、いわゆるミクロレベルでの結果出力には注目されていなかった。本稿で取り上げた環境改善資金調達シミュレーションを例に説明する。資源量の分布があったとき、人口総数の時間的変化はグラフの変化として確認することができるが、資源量の分布に対してエージェントがどのようにアプローチするかという点を確認することはできない。

政策グリッドコンピューティング実験センターでは、社会現象の再現、分析を容易にするためにはミクロレベルでの結果出力が必要であるとの考えから、そのプロトタイプとしてSugarscapeモデル向けMASツール用の可視化ツールを実装した。図8にSugarscapeモデルに基づくMASプログラムの可視化ツールを示す。この可視化ツールでは、Sugarscapeモデルに基づくMASプログラムから出力される情報をもとに、各エージェントの位置を2次元座標系に出力される。また、エージェントの属性によって色を変えて出力することができる。現時点では1試行のシミュレーション結果をアニメーションというかたちで出力することができる。

図9は社会システムを導入しなかった場合のエージェントの分布を可視化ツールによって示したものである。図9(A)は初期状態（step＝0）、図9(B)はエージェント数が最大（step＝103、図4 OAの最高点）のとき、図9(C)はほぼ中間地点（step＝1100）のとき、図9(D)は最終状態（step＝2000）のときのエージェント分布を示している。図9において青色は男性エージェントを、赤色は女性エージェントを表している。エージェントの数が徐々に減少してい

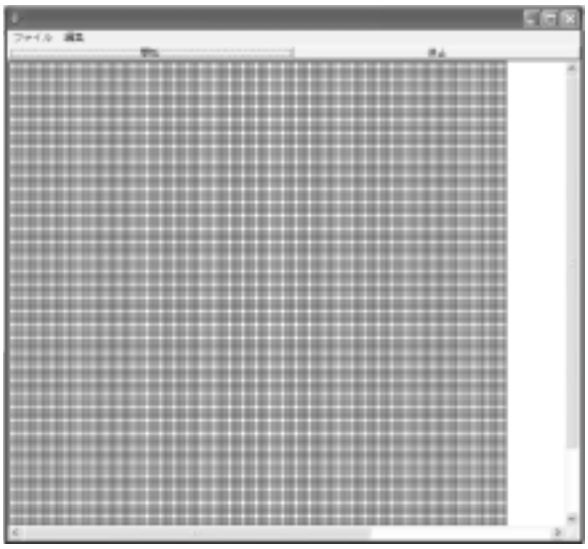

図8　Sugarscapeモデル用可視化ツール

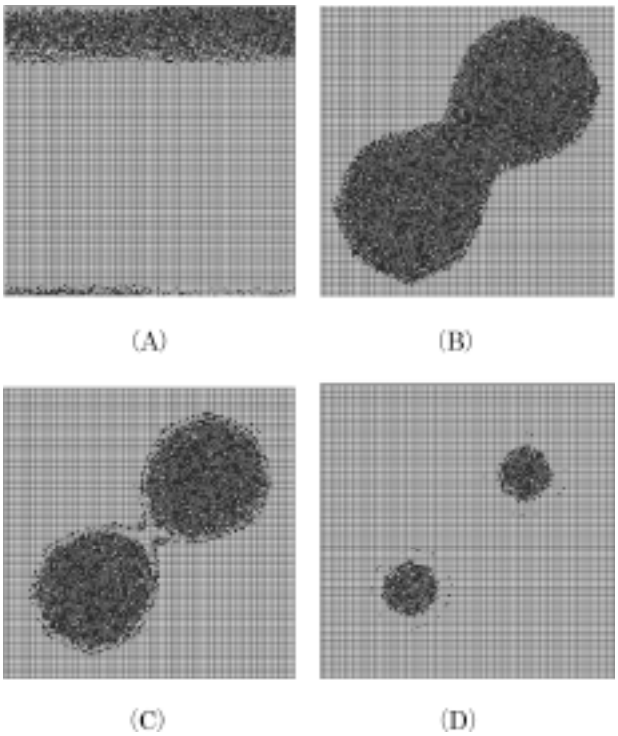

図９　可視化ツールによるエージェントの分布状況

くことが可視化ツールによっても確認できる。

また、可視化ツールの導入によって得られたものとしては、図９(C)のように２つの集中している分布に対して８の字型の輪郭が形成されていることがわかった。このような状況は図４～図６のグラフだけでは得ることのできない情報であり、可視化ツールの必要性を示すことができたといえる。

８．おわりに

本稿では、MASによる大規模シミュレーションの実現と計算処理の高速化について、MASを実現するモデルとして知られているSugarscapeモデルを取り上げ、MASによる並列処理の実現とMessage Passing Interface（MPI）を用いたMASプログラムの実装上の課題について述べた。また、MASの可視化についても触れ、可視化による必要性を示した。

プロトタイプ版MASプログラムの実装を通じて、より大規模なシミュレーションを実現する可能性が見出せた。現在、改良版並列実行型MASプログラムをベースとしたグリッドコンピューティング環境向け並列処理型MASプログラムの実装を進めており、大量の計算機を利用することで実行時間の更なる短縮と大規模シミュレーションの実現を目指している。

本稿では、MPIによるMASプログラムの実装について取り上げたが、グリッドコンピューティングを指向したシミュレーションプログラムの実装方法は他にもある。例えば、GridRPCと呼ばれるサーバ・クライアント型のAPIによる実装方法、Peer-to-Peer型の実装方法といったように、計算機環境に応じた手法がある。今後、計算機環境に応じたMASの実装方法について研究を深めるとともに、政策立案支援ツールとして広く提供できるソフトウェアの開発に力を注いでいくことが大切であると考えている。

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工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］ Ian Foster, Carl Kesselman, “The Grid: Blueprint for a New Computing Infrastructure” Morgan Kaufmann Publishers, Inc.,（1999）.

［4］ Message Passing Interface Forum, http://www.mpi-forum.org

［5］ MPICH, http://www-unix.mcs.anl.gov/mpi/mpich1/

［6］ LAM/MPI Parallel Computing, http://www.lam-mpi.org

［7］ GridMPI, http://www.gridmpi.org/index.jsp