

# マルチエージェントシミュレーションにおける グリッド環境向けアプリケーションの実装方法比較

蟻川 浩, 森下 仙一, 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 : <https://www.pglab.kansai-u.ac.jp/>  
e-mail : [pglab@jm.kansai-u.ac.jp](mailto: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](mailto:pglab@jm.kansai-u.ac.jp)）宛にご連絡いただきますようお願い申し上げます。

## **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](mailto: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 グリッド環境向けアプリケーションの実装方法比較

蟻川 浩<sup>1</sup>, 森下 仙一<sup>2</sup>, 村田 忠彦<sup>1,2</sup>

## Comparisons of Implementation Method for Grid-enabled Multi-Agent Simulation Application

Hiroshi Arikawa<sup>1</sup>, Sen-ichi Morishita<sup>2</sup>, Tadahiko Murata<sup>1,2</sup>

### 概要

グリッド環境を用いた大規模マルチエージェントシミュレーション(MAS)の実行時間短縮を目的として、メッセージ交換型インターフェイス(MPI)およびグリッド向け遠隔実行呼び出し(GridRPC)のそれぞれによる MAS プログラムの実装方法について報告する。また、MPI および GridRPC による MAS プログラムの比較を行い、実装方法の違いによる特徴について論じる。

### Abstract

Multi-Agent Simulation (MAS) is one of methods for social simulations. In order to reduce the computation time for large scale MAS using the Grid, there are two ways of implementing the Grid-enabled MAS applications: Message Passing Interface (MPI) and Remote Procedure Call for Grid (GridRPC). In this paper, we propose a implementation method for Grid-enabled MAS application using MPI and GridRPC. Then, we discuss the simulation result and the processing time by MPI-based MAS program and GridRPC-based 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，メッセージ交換型インターフェイス，グリッド向け遠隔実行呼び出し

Keyword: Large Scale Multi-Agent Simulation, Message Passing Interface, Remote Procedure Call for Grid

---

1 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

2 関西大学 総合情報学部

## 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技法である。MAS は多数のエージェントが相互に作用することで、環境の変化を確認できる特徴を持つ。近年、MAS の枠組みを用いて、コンピュータ上で「社会」(環境)を人工的に構成し、人流や交通流、経済現象、政策立案支援といった社会現象の再現、分析が試みられている[1, 2, 3].

現実社会を MAS で分析する場合、より大規模なシミュレーションを実施する必要がある。例えば、中核市規模における政策立案では、人口規模は 30 万人以上であるから、シミュレーション実施時は少なくとも 30 万のエージェントを定義する必要がある。このため、大規模 MAS の実行基盤が必要になる。大規模 MAS の実行基盤構築に関する研究が、中島ら[4]によって行われているが、大規模 MAS の高速化およびその実現方法を課題として挙げるに留まっている。

そこで我々は、大規模 MAS プログラムをグリッド・コンピューティング環境[5]で動作することを目的に、グリッド向け MAS プログラムの実装方法に関する研究を進めている[6, 7]. 本論文では、グリッド向けプログラムを実装する際に用いられる代表的なアプリケーションインターフェイスである Message Passing Interface (MPI)とグリッド向け Remote Procedure Call (GridRPC)を取り上げ、それぞれの特徴を生かしたグリッド向け MAS プログラムの実装方法について述べる。また、PC クラスタ上で構築した実行環境を利用してそれぞれの MAS プログラムを実行し、計算結果および計算時間の観点から比較を行った。その結果から得られた知見を示す。

## 2. MAS によるグリッド向けプログラムの実装方針

本論文では、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 MAS のモデルとして取り上げる。環境改善資金調達モデルは、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。本論文では、環境改善資金調達モデルによる MAS(環境改善 MAS)をグリッドプログラムとして実装する際の方針を示す。

環境改善 MAS では、エージェントとエージェントが移動する領域情報(環境情報)で構成される。シミュレーションの規模が大きくなるにつれ、エージェント数の増加および環境情報量の大きさが変化する。

エージェント数の増加によりエージェントの意思決定処理にかかる時間が増加するため、シミュレーション実施時間全体に影響を及ぼす。かつ、すべてのエージェントの情報を保持することも必要になる。さらにエージェントの意思決定が複雑になるほど大量の環境情報を保持する必要がある。

まとめると、環境改善 MAS の場合、シミュレーションの規模が大きくなるほど 1 台の計算機ではシミュレーション時間の観点で、またコンピュータの記憶容量の観点で実現困

難であり、グリッドコンピューティングのような並列分散処理技術を適用することで大規模計算が可能になる。具体的には、エージェント意思決定処理については、複数の計算機による分散処理を実現する。また、エージェントの情報を含む環境情報量の規模に対しては、複数の計算機で情報を分散させる。

現在、グリッド・コンピューティング環境で動作するためのプログラムを実装する際、MPICH[8], LAM[9], GridMPI[10]といった MPI ライブラリ[11]を使って実装する方法、NetSolve[12], Ninf-G[13], OmniRPC[14]といったライブラリを使って実装する方法が代表的である。本論文では前者を MPI による方法、後者を GridRPC による方法とする。

前者の場合、プロセス間同士でメッセージ交換をしながら並列計算を進める。環境改善 MAS の場合、複数のプロセッサに環境情報を分散した状態でエージェントの意思決定処理を行うことが可能になる。

一方、後者の場合、遠隔の計算機にあらかじめ用意されているサブプログラムを呼び出す、いわゆる Master-Worker 方式で並列処理を実現する。環境改善 MAS の場合、すべてのエージェントの意思決定処理をプロセッサの台数分で分散して実行することが可能になる。

### 3. MPI による MAS プログラムの実装

図 1 に MPI を用いた MAS の動作フローを示す。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数を意味し、図中 P1 から Pn は各プロセッサ番号を意味する。以下では MPI による MAS の処理内容について説明する。

処理(1) : 環境情報を分割した情報 (各区域の情報)を作成し、自分を含む他のプロセッサに配分する。このとき、MPI\_Scatter()を用いて情報を配分する。

処理(2) : 各区域が担当するエージェントの初期情報を作成し、自分を含む他のプロセッサに配分する。各プロセスで異なる大きさのデータを送信するため、先に MPI\_Scatter() で各プロセッサが担当する情報量を定義し、MPI\_Scatterv()でデータを送信する。

処理(3) : 各プロセッサで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。このとき、データ送信側は MPI\_Bsend()を用いる。また、データ受信側は MPI\_Probe()でデータ受信確認をしてから MPI\_Recv()で受信する。

処理(4) : 各プロセッサでエージェント移動の意思決定を行う。

処理(5) : 各プロセッサが持つ環境情報の領域外に移動するエージェント情報を交換する。データ送信側は MPI\_Bsend()を用いるが、データ受信側では MPI\_Probe()と MPI\_Get\_count()を用いてから MPI\_Recv()によって受信する。

処理(6) : エージェントの衝突判定を行う。

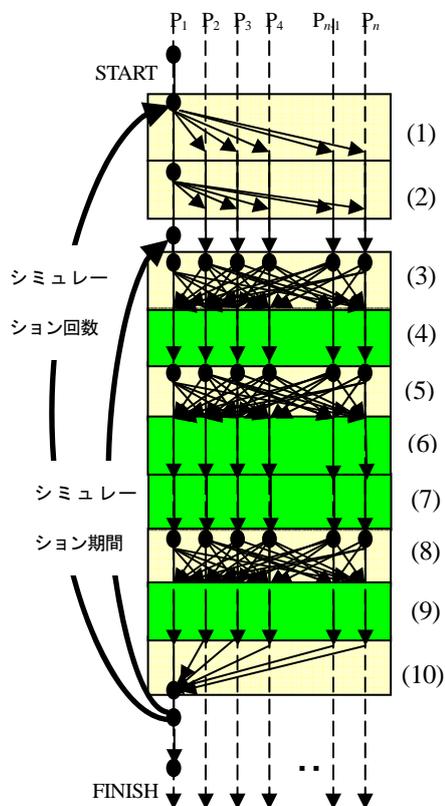


図1 MPIを用いたMASの動作フロー

処理(7) : 各プロセッサで資源の収集と交配を行う。

処理(8) : 環境改善資金調達を行う場合、まず、`MPI_Allreduce()`を使って汚染物質量の削減に必要なデータ(シミュレーション全体でのエージェント数, 投票数など)を収集する。次に、各プロセッサがもつ最大汚染物質量を`MPI_Allgather()`を使ってすべてのプロセッサに対して通知する。最後に、汚染物質量の削減処理を行う。

処理(9) : 各プロセッサで資源回復処理を行う。

処理(10) : 各プロセッサの所持するデータを収集し、ステップ毎のデータ集計を行う。このとき、`MPI_Reduce()`を用いてデータ集計を行う。

処理(3)~(10)を所定のステップ数繰り返し、1 試行の計算が完了する。そして統計的に十分な数の試行を行い、シミュレーション結果を得る。

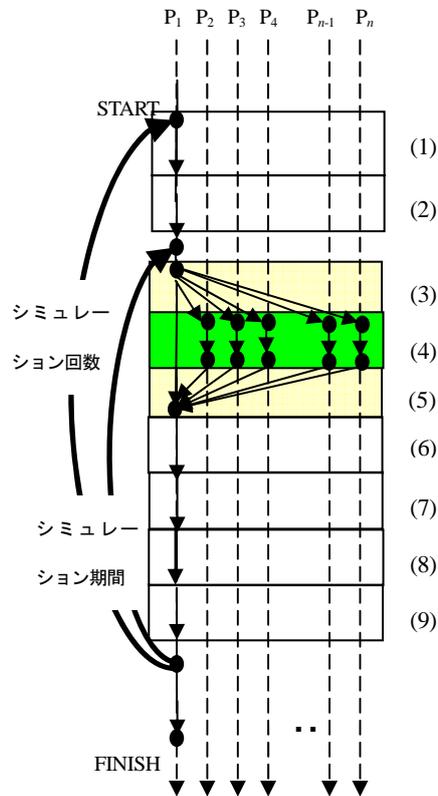


図2 GridRPC を用いた MAS の動作フロー

#### 4. GridRPC による MAS プログラムの実装

図2に GridRPC を用いた MAS の動作フローを示す。以下では各プロセスの処理内容について述べる。

処理(1) : Sugarscape 環境を作成する。

処理(2) : エージェントの初期情報を作成する。

処理(3) : P1 から他の全プロセッサに、演算に必要なデータを送信する。

処理(4) : P1 以外のプロセッサは受信したデータを基にエージェント移動の意思決定を行う。

処理(5) : エージェントの意思決定結果を P1 に送信する

処理(6) : 資源の収集と交配を行う。

処理(7) : 環境改善資金調達を行う場合、汚染物質質量の削減を行う。

処理(8)：資源回復処理を行う

処理(9)：ステップ毎のデータ集計を行う。

処理(3)～(7)を所定のステップ数繰り返し、1 試行の計算が完了する。そして統計的に十分な数の試行を行い、シミュレーション結果を得る。

GridRPC では P1 が他のプロセッサにあるプログラムを起動するかたちになる。環境改善 MAS の場合、処理(3)～(5)の処理が遠隔プログラムとして呼び出される。本論文では OmniRPC を用いて GridRPC による MAS プログラムを実装した。具体的には処理(3)では `OmniRpcCallAsync()` を用いている。また、処理(5)では `OmniRpcWaitAll()` を用いている。

## 5. 実験

MPI と GridRPC で MAS を実装し、比較実験を行った。対象とする MAS は、環境改善資金調達モデルの、OA (Open access), VC (Voluntary contribution), LO (Lotteries) の 3 つのシミュレーションである。これらの違いについては文献[2]に詳細な説明があるため、ここでは省略する。

### 5.1 実験環境

比較実験では、Intel 社 3.0EGHz Pentium 4 プロセッサと 2GByte のメモリを搭載した計算機を 9 台で構成するホモジニアス PC クラスタを用いた。PG クラスタにおけるプロセッサ間の通信速度はすべて 1Gbps である。すべてのプロセッサではオペレーティングシステムとして Windows XP Professional を採用しているため、VMware Server を用いて仮想的に Linux 環境を構成した。

実験環境で用いた MPI ライブラリおよび GridRPC ライブラリは、それぞれ MPICH 1.2.7, OmniRPC 1.0 である。

### 5.2 比較実験

比較実験を行う際の実験パラメータを以下に示す。Sugarscape 環境については、単体の計算プログラムと GridRPC を用いた MAS プログラムでは、 $180 \times 180$  の Sugarscape モデルを採用する。また、MPI を用いた MAS プログラムでは、 $180 \times 180$  の区画を正方形に 9 分割した Sugarscape モデル( $60 \times 60 \times 9$  台)のモデルを採用する。単一計算機上で動作する MAS プログラム, MPI および GridRPC による MAS プログラムすべてにおいて、Sugarscape 環境の資源量の分布は図 3 を用いる。初期エージェント数は 4320 とする。その他については文献[6]で示されているパラメータを用いる。

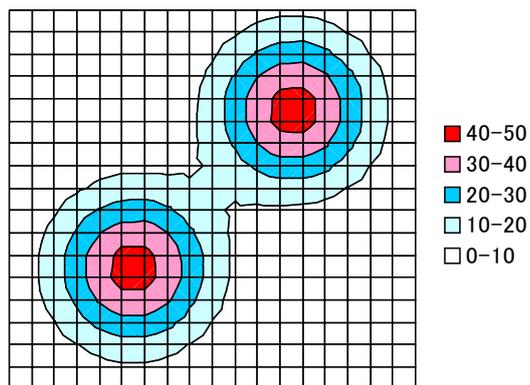


図3：資源量の分布

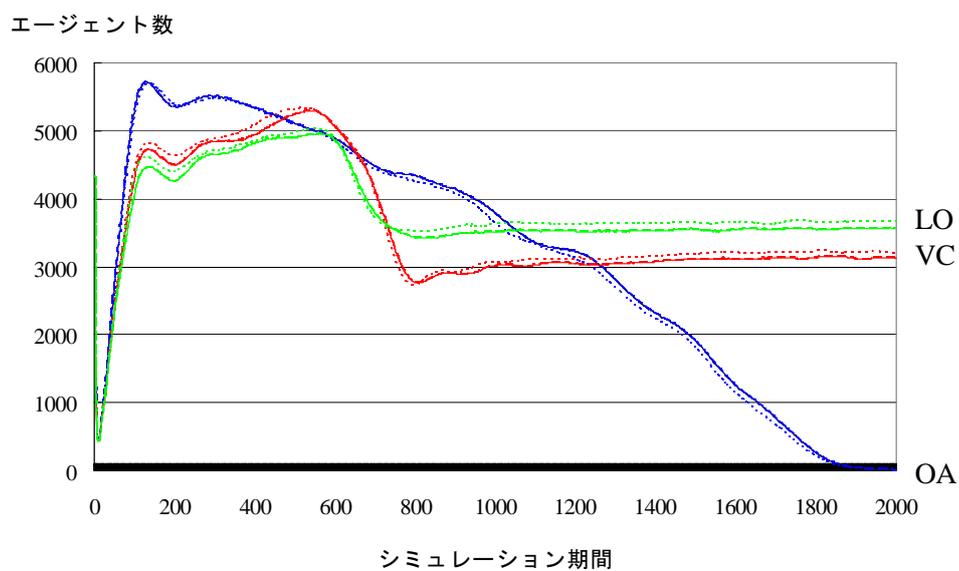


図4：エージェントの推移

表1：平均2乗誤差

	OA	VC	LO
MPI	6422.7	7403.6	9241.8
GridRPC	0	0	0

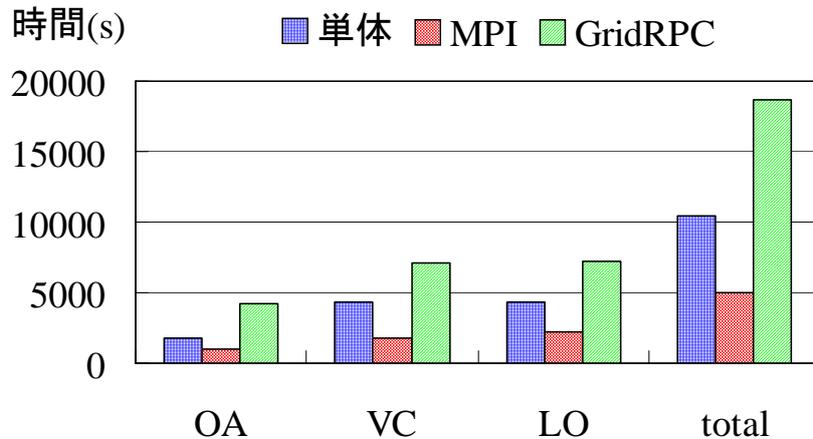


図5：実行時間

### 5.3 実験結果

図4に、単一計算機で動作するMASプログラム、MPIおよびGridRPCによるMPIプログラムのエージェントの推移を表1に、単一計算機で動作するMASプログラムを基準としたMPIおよびGridRPCによるMASプログラムの平均2乗誤差を示す。

図4と表1から、GridRPCによるMASプログラムは単一計算機上で動作するMASプログラムと同一シミュレーション結果が得られた。これは、GridRPCにおける環境情報の管理が1台のプロセッサで行われているため、単一プロセッサで動作する場合と変わらないためである。一方、MPIによるMASプログラムでは環境情報をプロセッサ台数で分散配置するため誤差がある。但し、図4に見られるように、単一計算機上で動作したプログラムと同様の推移を示すため、結果の妥当性には影響がない。

図5に単一計算機で動作するMASプログラム、MPIおよびGridRPCによるMASプログラムの実行時間を示す。MPIによるMASプログラムでは、単体での計算プログラムに比べて大幅な実行時間の短縮が可能である。一方、GridRPCによるMASプログラムの場合、1台あたりのエージェントの意思決定処理が多いこと、かつエージェントの意思決定処理以外の分散処理が実現できないため、単一計算機で動作するMASプログラムよりも実行時間がかかった。本実験では、エージェントの意思決定処理は8台の計算機で行ったため、GridRPCの特徴を生かしきれていないと言える。また、GridRPCの場合、通信時間よりも格段に演算時間がかかる場合には有効であるため、今後、計算機台数を増やした実験および演算時間がかかる場合の実験を行い、効果を確認する。

## 6. おわりに

本論文ではグリッド・コンピューティング向け MAS プログラムの実装方法の違いについて述べた。また、実際にグリッド・コンピューティング向けプログラムを実装するためのライブラリである MPI と GridRPC を用いて MAS プログラムを実装し、PC クラスタを用いた比較実験を行った。MPI による MAS プログラムの場合、シミュレーション実施時間の短縮が可能であるが許容範囲での誤差が発生する。また、GridRPC による MAS プログラムの場合、誤差は発生しないがシミュレーション実施時間はエージェント意思決定処理の処理量に影響を及ぼす。

今後は、プロセッサ台数を増加させた場合の並列効果や問題の規模をさらに大きくした場合の効果について検討するとともに、PC クラスタ以外での計算機環境に適用した場合の MPI および GridRPC による MAS プログラムの特性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نز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] 鶴飼康東, 村田忠彦, 北埜裕子:既婚女性の労働供給における政策グリッドコンピューティング実験, 関西大学経済論集, Vol.55, No.3, pp.421-443, 2005
- [4] 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).
- [5] Ian Foster, Carl Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers (1999) .
- [6] 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).
- [7] 森下仙一, 蟻川浩, 村田忠彦: 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, 情報処理学会研究報告(2007-HPC-109), pp.19-24, 2007
- [8] MPICH, <http://www-unix.mcs.anl.gov/mpi/mpich1>
- [9] LAM/MPI Parallel Computing, <http://www.lam-mpi.org>
- [10]GridMPI, <http://www.gridmpi.org/index.jsp>
- [11]Message Passing Interface Forum, <http://www.mpi-forum.org>
- [12]NetSolve, <http://icl.cs.utk.edu/netsolve/>
- [13]Ninf-G, <http://ninf.apgrid.org/>
- [14]OmniRPC <http://www.omni.hpcc.jp/OmniRPC/>