2007－HPC－111 (24)
2007／8／2

MPIとGridRPCの併用による マルチエージェントシミュレーションプログラムの実装

森下 仙一† 蟻川 浩‡ 村田 忠彦†‡

社会科学の分野では問題解決手法のひとつとしてマルチエージェントシミュレーション(MAS)が注目されている．本稿では，グリッド環境におけるMASの実装方法のひとつとして，MPIとGridRPCを併用した実装方法を提案し，その効果について報告する．本稿で提案するMASプログラムの実装方法では，大規模データの分散化のためにMPIを，エージェントの意思決定処理の高速化のためにGridRPCを用いた．提案手法により，GridRPCと同様の計算結果が得られたとともに，GridRPCで問題となった実行時間の短縮も実現できることを確認した．

Implementation Issues for Multi-Agent Simulation Program using Hybrid of MPI and GridRPC

Sen-ichi Morishita† Hiroshi Arikawa‡ Tadahiko Murata†‡

This paper describes an implementation method for Multi-Agent Simulation Program using a hybrid of MPI and GridRPC. Multi-Agent Simulation (MAS) has obtained increasing attention from social science. But we must solve various problems to achieve a large scale MAS of applying real society. We proposed implementation methods for large scale MAS in the past and implemented it with MPI or GridRPC, We have found a limit of executing a large scale MAS by using our implementation methods. In this paper, we propose a new method which combined implementation methods we proposed and implement it by using MPI and GridRPC. Then, we describe a finding which we obtained by measuring the execution time of the simulation.

1. はじめに

社会科学の分野では，人流や交通流，経済現象などの社会現象を再現・分析できるシミュレーション技法のひとつとして，マルチエージェントシミュレーション(MAS)が注目されている．MASは，環境と自律的な行動をする多数のエージェントから構成されるシミュレーションで，エージェント間の相互作用と意思決定が環境にどのような影響を及ぼすかについてシミュレーションできる[1),2),3)]．

MASによる現実社会のシミュレーションを実施する場合，様々な問題を解決しなければならない．例えば，中核市規模の都市における市民の経済現象のシミュレーションを実現するためには最低でも30万エージェントによるシミュレーションを実現しなければならない．また，エージェントがある活動範囲をもつ場合，活動範囲の特性を環境情報として表現する必要があるが，その大きさはシミュレーションの精度に影響する．さらに，我々が何らかの意思決定をする際には複数の条件に基づいて合理的もしくは非合理的な選択をすることをシミュレーションで実現するとすれば，一エージェントの意思決定にかかる時間が増加し，エージェント数に比例してシミュレーション時間が増加する．このような状況から単一のプロセッサを有する計算機で大規模MASを実行することは時間制約のある場合においては困難であり，大規模な記憶領域と処理時間の短縮を同時に実現できるようなMASプログラムが必要になる．

我々は，グリッド・コンピューティング[4)]を指向した大規模MAS実行環境の構築を目的として，その上で動作するMASプログラムの実装方法について研究を進めている．具体的には，MPIに基づく大規模MASプログラムの実装[5)]，環境情報の分割方法の違いによる大規模MASの基礎性能評価[6)]，GridRPCによるエージェント意思決定処理の短縮を実現するためのMASプログラムの実装方法とMPIによる実装方法の比較[7)]を行ってきた．その過

†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

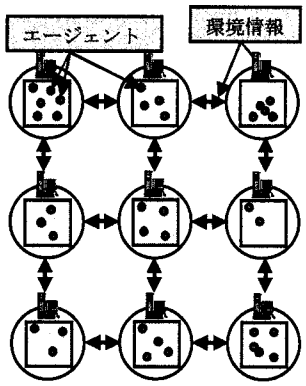

図1　情報分割方式

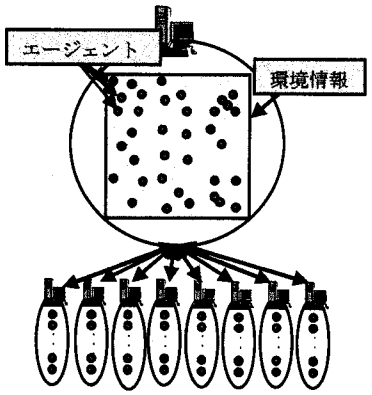

図2　意思決定分割方式

程において，MPIによるMASプログラムの実装ではMASのアルゴリズムに一部改良を加えることでMASプログラムの並列実行とその実行時間の短縮は可能になったが，MASアルゴリズムの一部改良がシミュレーション結果に影響を及ぼすことを確認した[5),6)]．一方，GridRPCを用いたMASプログラムの実装においては，シミュレーション結果への影響は発生しないものの，環境情報の規模拡大が困難であること，またエージェントの意思決定処理よりもサーバとクライアント間の通信確立にかかる時間が影響し，MPIによる実装よりも実行時間の短縮が困難であることを確認した[7)]．

本稿では，シミュレーション結果の影響を最小限にするとともに，実行時間の短縮を考慮に入れたグリッド向けMASプログラムの実装として，MPIによる方法およびGridRPCによる方法を組み合わせたMASプログラムの実装方法を提案する．具体的には，環境情報の拡大の部分についてMPIで取り扱い，エージェントの意思決定の高速化をGridRPCで取り扱う．また，提案手法の有効性についてPCクラスタ環境での動作実験を行い，その結果について述べる．

2.　グリッド向けMASプログラム実装上の課題

MASにはシミュレーションの内容によって様々なモデルが考えられる．そこで，本稿ではSugarscapeモデル[1)]に基づくMASに焦点を当てる．

Sugarscapeモデルに基づくMASは人工社会モデルのひとつであり，エージェントおよび2次元トーラス空間で表現された環境で構成されるシミュレーションモデルである．後者は一般的に環境情報と呼ばれており，エージェントの意思決定に必要となる情報が含まれている．Sugarscapeモデルによる大規模シミュレーションを考えた場合，エージェント数の増加および環境情報の拡大が必要になる．

2.1　MPIによる実装

大規模MASプログラムの実装において，環境情報を記憶するための領域確保を実現する手法としてはMPIによる方法が挙げられる．図1は環境情報を分割した場合の概要である．MPIによるMASプログラムの実装では，環境情報をベースにその環境に存在するエージェント情報を複数のプロセッサに分散配置し，エージェントの意思決定処理を並列処理によって実現する．また，エージェントの意思決定により環境情報が変化するため，隣り合う情報を持つプロセッサ間で情報交換を行う．

MPIによるMASプログラムの場合，環境情報を分散配置させることが可能になるため環境情報の規模が大きくなった場合でも，プロセッサ数を増やすことで対処できる．一方で，隣り合う情報を持つプロセッサ間での情報交換を高速に行うために，エージェントの最大視野分の情報を互いにオーバラップさせる必要があり，かつオーバラップ部分におけるエージェントの衝突処理に工夫しなければならない．その結果，単体のプロセッサによるMASシミュレーションの結果と比較してオーバラップ部分の影響が結果に現れる．

また，MPIによるMASプログラムは実行する計算機環境に制限がある．すなわち，MPIはプロセッサ間で情報交換できる環境が必要になる．したがって，PCクラスタのように同一の処理能力を有するプロセッサで構成された計算機環境の場合は計算時間の短縮は期待できるが，グリッド環境のように異なる性能の計算機上でかつ互いに通信することが難しい環境での実行には適さない．

2.2 GridRPCによる実装

大規模MASプログラムの実装における，もうひとつの方法としてエージェント意思決定処理を複数のプロセッサで分散処理を行うGridRPCによる方法が挙げられる．図2はエージェントの意思決定処理を分割させた場合の概要である．GridRPCによるMASプログラムの実装では，環境情報とエージェント情報は1台のプロセッサが管理し，エージェントの意思決定処理をサーバ/ワーカ方式で複数のプロセッサに分配し，エージェントの意思決定処理にかかる時間を短縮する．

GridRPCによるMASプログラムの場合，エージェント数の増加に対してプロセッサ数の増加で対応できる．一方で，環境情報の規模拡大に対しては1プロセッサの記憶領域を超える環境情報を保持することはできない．場合によってはファイルシステムを活用することで大規模化は可能であるが，ファイルシステムへのアクセスがオーバヘッドとなり，シミュレーション時間の短縮は見込めない．

GridRPCによるMASプログラムの場合，プロセッサの処理能力やマスタ/ワーカ間の通信帯域に応じて1プロセッサあたりに割り当てるエージェント数を柔軟に変更できるため，MPIによるプログラムとは異なり，異なる性能のプロセッサを有する計算機環境で実行できる特徴を有する．

3. MPIとGridRPCの併用によるMASプログラムの実装

前述したように，MPIおよびGridRPCによるMASプログラムにはそれぞれ長所および短所を有する．MPIおよびGridRPCによるMASプログラムの長所および短所はそれぞれが相補の関係になっていることがわかる．すなわち，環境情報の大規模化に対してはMPIを用いることで，また計算機環境に依存しないエージェント意思決定処理時間の短縮化に関してはGridRPCを用いることでグリッド環境に適したMASプログラムの実装が可能になることが見込める．

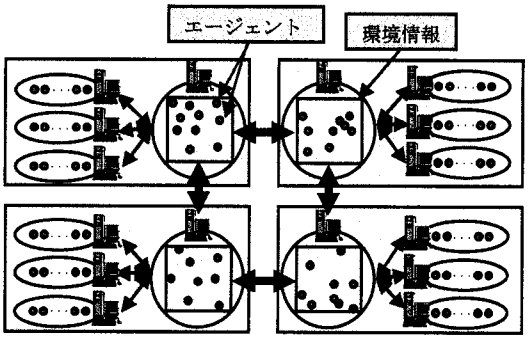

図3　ハイブリッド方式のイメージ

そこで，本稿ではMPIおよびGridRPCを併用したMASプログラムの実装方法について提案する．本稿ではMPIとGridRPCを併用したMASプログラムのことをハイブリッド型MASプログラムと呼ぶことにする．

3.1 ハイブリッド型MASプログラムの実装

まず，ハイブリッド型MASプログラムの概略について説明する．図3はハイブリッド型MASプログラムの実装の概要を示す．ハイブリッド型MASは，MPIによるMASプログラムを基礎として，エージェント意思決定処理部分をGridRPCのAPIを用いて実装し，エージェントの意思決定処理を複数のプロセッサで分散処理する．これにより，エージェント数の増加に対してはGridRPCの枠組みを用いてプロセッサ数の増加で対応できる．特に，複数のPCクラスタで構成されたグリッド環境でも柔軟に対応できる．また，環境情報を規模拡大に対してもMPIの枠組みを用いることで柔軟に対応できる．

図4はハイブリッド型MASプログラムの動作フローを示す．図4のP_{m1}からP_{mn}は環境情報を管理するプロセッサを意味する．またP_{w1}からP_{wk}はエージェント意思決定処理を担当するプロセッサを意味する．以下では，ハイブリッド方式の実装についてMPIとGridRPCのライブラリ名を使いながら説明する．本稿では，説明の都合上GridRPCによる実装の部分はOmniRPC [8]が提供する関数名を用いた．

処理(1)：P_{m1} のプロセッサで，各 P_m プロセッサが担当する環境情報を作成し，担当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)：P_{m1} のプロセッサで，各 P_m プロセッサが担当するエージェントの初期情報を作成し，担当プロセッサに配分する．このとき，MPI_Scatter()を用いて各 P_m プロセッサに配分するエージェントの情報量を送信しておき，その後，エージェントの初期情報を MPI_Scatterv()を用いて送信する．

処理(3)：各 P_m プロセッサ間で，シミュレーションを行う上で必要となる隣接区域の情報を交換し，自分の環境情報に付加する．このとき，データ送信側は MPI_Bsend()を用いてデータ送信を行い，データ受信側は MPI_Probe()を用いてどのプロセッサからデータが送信されてきたかを確認した後，MPI_Recv()を用いてデータを受信する．

処理(4)：各 P_m プロセッサから任意の P_w プロセッサへ，エージェント意思決定処理に必要なデータを送信する．このとき各 P_m プロセッサは，OmniRpcCallAsync()を用いてデータを送信する．

処理(5)：各 P_w プロセッサは，受信したデータを基にエージェント意思決定を行う．

処理(6)：各 P_w プロセッサは，エージェント意思決定の結果を，データ送信を行った P_m プロセッサに送信する．このとき各 P_m プロセッサは，OmniRpcWaitWall()を用いてエージェント意思決定の結果を受信する．

処理(7)：各 P_m 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．このとき，データ送信側は MPI_Bsend()を用いてデータの送信を行い，データ受信側は MPI_Probe()を用いてどのプロセッサからデータが送信されてきたかを確認した後，MPI_Get_count()を用いて受信するエージェントの情報量を取得し，MPI_Recv()を用いてデータを受信する．

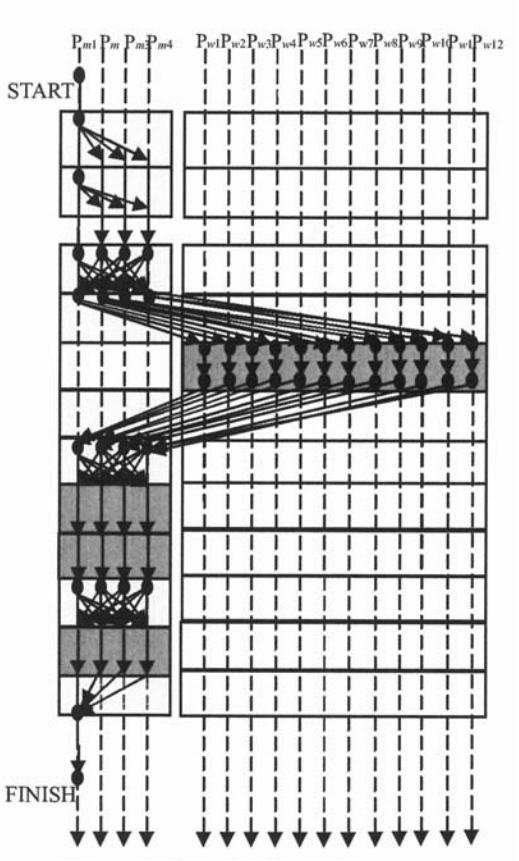

図4　ハイブリッド方式の動作フロー

処理(8)：各 P_m プロセッサで，隣接領域から移動してきたエージェントの衝突判定を行う．

処理(9)：各 P_m プロセッサで資源の収集と交配を行う．

処理(10)：環境改善資金調達を行う場合，P_m プロセッサ間で必要なデータを交換しながら汚染物質量の削減を行う．この処理で用いる MPI 関数は，データ収集に MPI_Allreduce()，削減場所の通知に MPI_Allgather()である．

処理(11)：各 P_m プロセッサで資源回復処理を行う．

処理(12)：各 P_m プロセッサの所持データを，P_{m1} プロセッサに収集し，収集したデータを集計する．その後，データ集計結果を出

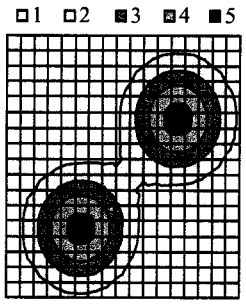

図5　資源量の分布

力する．このとき，MPI_Reduce()を用いて，データの収集と集計を行う．

処理(3)～処理(12)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う．

4. 実験

ハイブリッド方式によるMASの特性について確認すべく，本稿ではSugarscapeモデルを拡張した環境改善資金調達MAS[2)]を用い，MPI, GridRPC, ハイブリッド型それぞれについてMASプログラムを実装した．

4.1 実験環境

実験環境として，Intel社3.0EGHz Pentium4プロセッサ，2GByteのメモリを搭載した20台の計算機で構成されるホモジニアスPCクラスタを用いた．PCクラスタにおけるプロセッサ間の通信速度は全て1Gbpsであり，NFSを用いてプロセッサ間すべてストレージを共有している．各プロセッサノードで用いたOSはLinux 2.6.9-11.ELである．また，MPIおよびGridRPCはそれぞれMPICH 1.2.7，OmniRPC 1.0.0を使用した．実験に際し,OmniRPCはPthreadを有効にした．

4.2 実験方法

本稿では，MPIによる環境情報分割方式，GridRPCによる意思決定分割方式，ハイブリッド方式の3種類を用いた．なお，ハイブリッド方式においてはBARANCEDとUNBRANCEDの2種類の方式を実装した．BARANCEDにおいては環境情報の管理を担当するプロセッサのエージェント数に応じてエージェント意思決定処理を担当するプロセッサ数を割り当てている．一方，UNBARANCEDにおいては環境情報の管理を担当するプロセッサのエージェント数に関係なく均等にプロセッサ数を割り当てている．

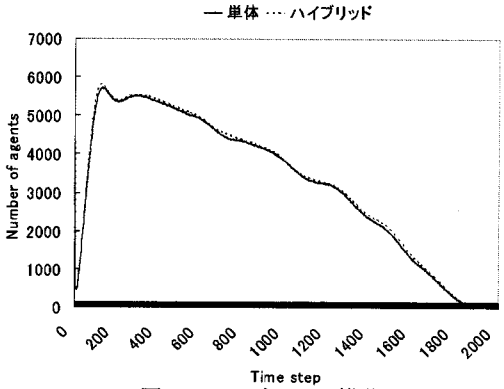

図6　エージェント推移

環境改善資金調達MASのSugarscape環境について，(A) 180×180と(B) 360×360の2種類を採用する．資源量の分布を図5に示す．各環境における初期エージェント数はそれぞれ4320, 17280とし，各エージェントの最大視野はそれぞれ6, 12とする．その他，環境改善資金調達MASを行う上でのシミュレーション条件は文献2)を用いた．

4.3 実験結果および考察

4.3.1 シミュレーション結果への影響

図6は環境情報の大きさが180×180としたときのエージェント数の推移を示す．図6では単体のプロセッサで行った場合および提案手法による場合のシミュレーション結果を示す．図6の結果から実装方法の違いによるシミュレーション結果への影響はほとんどないことが確認できた．

4.3.2 プロセッサ数の違いによるシミュレーション時間の影響

プロセッサ数を変化させた場合のシミュレーション時間への影響を図7と図8に示す．図7は環境情報の大きさが180×180の場合，図8は環境情報の大きさが360×360の場合である．MPIのみ,およびGridRPCのみで実装したMASプログラムの場合，4台,9台,16台と変化させたときの結果を掲載した．また，提案手法については，環境情報を担当するプロセッサを4台にし，エージェント意思決定処理を担当するプロセッサを変化させた結果を示した．提案手法の場合については環境情報を担当するプロ

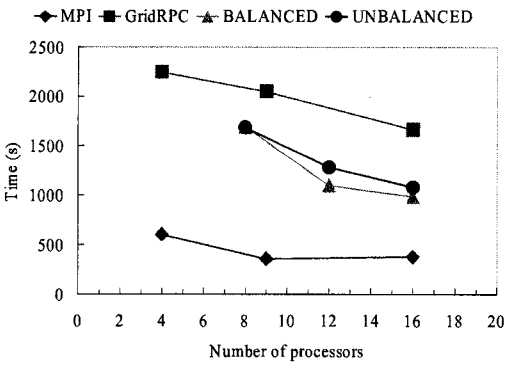

図 7　実行時間 (180×180)

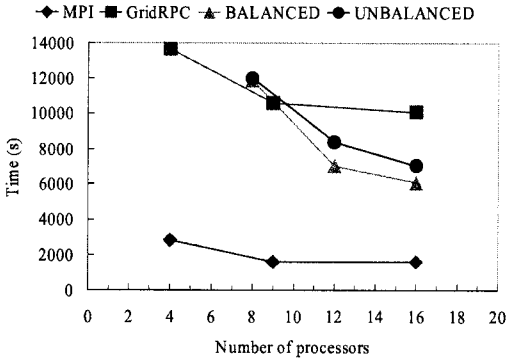

図 8　実行時間 (360×360)

セッサおよびエージェント意思決定処理を担当するプロセッサ数の合計におけるシミュレーション時間の影響を測定した.

図 7 および図 8 から GridRPC のみの場合に比べて実行時間の短縮を実現できた. これにより GridRPC の問題点であったマスタプロセッサに対するデータ交換の負荷が低減された. また, BARANCED および UNBARANCED の結果からプロセッサ数をエージェント数に応じて分配することで実行時間の短縮が可能になることも確認できた.

5. 関連研究

MPI と GridRPC を組み合わせた手法としては, MPI によって並列化されたプログラムを GridRPC で起動し, クライアントプログラムを介して連携する方法を提案している [9]. MPI と GridRPC を組み合わせているという点では似ているが, 提案手法では MPI で並列化されたプログラムに対し, さらに GridRPC で並列化を実現している点が異なる.

6. おわりに

本稿では, MPI と GridRPC によるハイブリッド型 MAS について提案し, PC クラスタでの実験を行った. その結果, ハイブリッド方式により, GridRPC で問題であった計算時間の短縮が可能となった.
今後の課題は, 異なるプロセッサ処理能力で構成されている計算機環境での性能評価を行い, 提案手法の有効性を確認する.

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・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) 鵜飼 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) 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) 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).

7) 森下 仙一, 蟻川 浩, 村田 忠彦："グリッド向けマルチエージェントシミュレーションにおける実装方法の比較", 計算工学講演会論文集, Vol.12, No2, pp.699-702 (2007).

8) M. Sato, T. Boku, D. Takahashi, "OmniRPC: a Grid RPC System for Parallel Programming in Cluster and Grid Environment", *Proceedings of CCGrid2003*, pp. 206-213, (2003).

9) 武宮 博, 田中 良夫, 中田 秀基, 関口 智嗣, MPI と GridRPC を利用した大規模 Grid アプリケーションの開発と実行: Hybrid QM/MD シミュレーション, 情報処理学会論文誌コンピューティングシステム, Vol.46, No. SIG12, pp.384-395, (2005).