Vol.2015-ICS-178 No.13
2015/3/3

マルチエージェントを用いた並列パンデミックシミュレーション

福士　雄太[1,a)]　臼井英之[2,b)]

概要：本研究では，マルチエージェントを用いたインフルエンザの伝播シミュレーションプログラムを開発し，OpenMP を用いたスレッド並列，MPI を用いたプロセス並列，および両者を組み合わせたハイブリッド並列を実装することによりシミュレーションの高速化を行った．また，開発したプログラムを用いて，近畿圏を模倣した 800 万人が生活する仮想的な都市を対象にしたインフルエンザ伝播シミュレーションを行い，結果の妥当性の検討を行うとともに，実行時間の短縮化について検討した．

1. はじめに

近年，世界のグローバル化は進み，人や物が世界中を簡単に行き来するようになった．一方，感染症もこれまでにないスピードで世界中に広がるようになった．そのため，感染症の世界的な拡大（パンデミック）のリスクが高まっている [1]．グローバル化によるパンデミックのリスク増大を受け，パンデミックを防ぐ手法や，パンデミックが起こった時の対策などについて世界各国にて研究が行われている．

パンデミックに関する研究の一つに，感染症伝染のシミュレーションがある．最近の研究では，都市部におけるインフルエンザの伝染の様子を再現する際，人間を模倣したエージェントを定義し，定められたルールに従いエージェントを行動させることで現れる集団的現象を解析するマルチエージェントシミュレーション（以下 MAS と略す）の手法が用いられるようになり，世界各国で研究が進められている [2], [3], [4]．日本においても，MAS を用いたインフルエンザの伝染シミュレーションが行われている [5], [6]．しかし，まだその例は多いとは言えない．

MAS では学校を閉鎖する，外出を自粛するといったシナリオを自由に設定することが出来るが計算コストが高いという欠点がある．一方，インフルエンザの伝搬シミュレーションでは確率を用いるため，実行結果にばらつきが生じる．そのため，複数回のシミュレーション実行が不可欠となる．1 回のシミュレーション実行時間を短縮することで，より多くの回数シミュレーションを実行することが出来れば，より良い精度の結果を得ることが出来る．

本研究ではこれらの背景のもとに大きく以下の 2 つのことを実施した．まず 1 つ目は，マルチエージェントを利用したインフルエンザパンデミックシミュレーションプログラムを開発し，関西圏を対象にインフルエンザの伝染シミュレーションを行った．次に，シミュレーションを高速化するために，MPI（Message Passing interface）によるシミュレーションのプロセス並列化や OpenMP によるシミュレーションのスレッド並列化，両者を組み合わせたハイブリッド並列化といったプログラムの並列化手法を用いることでシミュレーションの高速化に取り組んだ．

2. マルチエージェントによるパンデミックシミュレーション

ここでは，シミュレーションを行うにあたり設定したモデルについて述べる．

2.1 シミュレーションの概要

本研究では，マルチエージェントによるインフルエンザの伝染シミュレーションプログラムを開発した．コンピュータ上に設定された仮想の都市に，住人となるエージェントを定義し，それぞれのエージェントが決められた行動をとることでインフルエンザの伝染していく様子をシミュレーションする．各エージェントはそれぞれ住む家を持ち，自分の家から学校，会社などの場所へと移動する．その際，町と町の間の移動には電車を利用する．そして，

[1] 神戸大学工学部情報知能工学科
Kobe University Faculty of Engineering
Department of Computer and Systems Engineering
[2] 神戸大学システム情報学研究科計算科学専攻
Kobe University Graduate School of System Informatics
Department of Computational Science
[a)] 1155197t@stu.kobe-u.ac.jp
[b)] h-usui@port.kobe-u.ac.jp

エージェントは感染者と接触することで確率的に感染症に感染する．シミュレーションの時間ステップは 1 ステップ 10 分としている．また，プログラムは Fortran90 言語で記述した．シミュレーションの概要図を図 1 に示す．

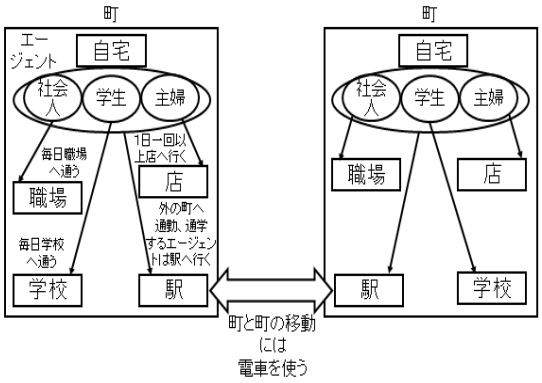

図 1　シミュレーションの概要
Fig. 1　The agent-based model for the pandemic simulation.

2.2　エージェントのモデル

図 1 にあるように，本研究ではエージェントの種類として学生，社会人，主婦の 3 つを用意した．平日の間，学生エージェントは朝家から学校へ通学し，夕方帰宅する．帰宅後はランダムに自分の町にある店へ買い物に出かける．どの町の学校に通うか，何時に通学，帰宅するかはランダムに与えられ，自分の住む町の外の学校に通学するエージェントは電車を利用する．また，社会人エージェントは朝家から会社へ出社し，夕方帰宅する．会社にいる間ランダムに他の会社へ訪問する．帰宅後はランダムに自分の町にある店へ買い物に出かける．学生エージェントと同様，どの町の会社に行くか，何時に通勤，帰宅するかはランダムに与えられ，電車を利用するエージェントも存在する．そして，主婦エージェントは少なくとも 1 回以上，家から自分の住む町の店へ買い物に出かける．

休日は，すべてのエージェントがランダムに店へ買い物に出かける．買い物に出かける場所や買い物に出かける時間はランダムに与えられ，電車を利用するエージェントが存在する．これらのエージェントの行動をまとめたものを図 2 に示す．

2.3　都市のモデル

本研究では図 3 に示すように近畿圏，特に JR 東海道本線および JR 山陽本線沿いの各市を対象とした．それぞれの市には家，学校，会社，駅，店の 5 つの場所が設定されている．ここで，店の数については平成 19 年度商業統計 [7] における商業集積地区（小売業，飲食業及びサービス業を営む事業所が近接して 30 店舗あるもの）を利用した．また，駅はすべての市にて 1 つのみとした．学校，家，職場の数については，各市町村が公表している統計データを利用した．それぞれの市における場所の数を表 1 に示す．

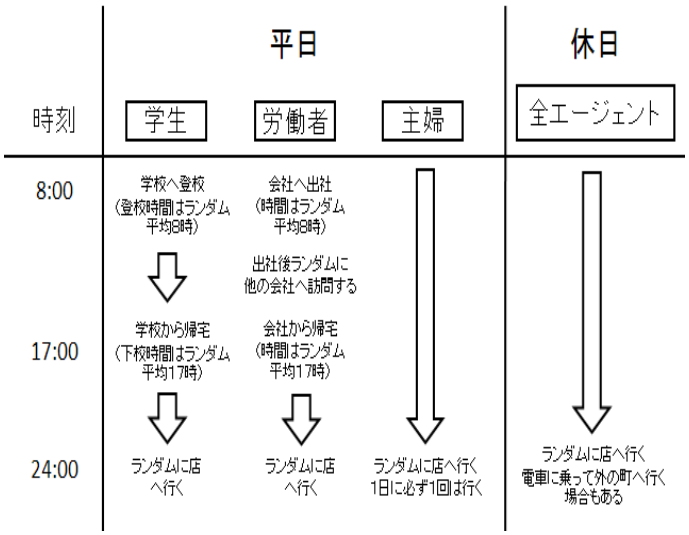

図 2　エージェントの行動パターン
Fig. 2　The types of agent behavioral simulation.

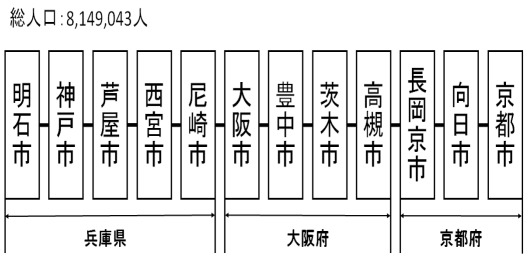

図 3　対象とする都市
Fig. 3　The model of target cities.

表 1　各市における場所の数
Table 1　The numbers of facilities in each city.

市町村名	家	学校	会社	店
明石市	129599	92	9047	29
神戸市	734774	592	67807	143
芦屋市	43882	34	2899	9
西宮市	214419	159	13364	26
尼崎市	224883	274	17878	37
大阪市	1391901	984	189234	503
豊中市	171027	123	14576	50
茨木市	119314	95	9132	17
高槻市	156402	116	9367	22
長岡京市	34668	23	2689	7
向日市	21511	14	1975	2
京都市	693971	613	73391	181

各市におけるエージェントの数については，各市の人口を元に算出した．日本統計年鑑 [8] によると，平成 25 年度

における人口に対する労働者人口比率は約 59.3% である．また，非労働者人口における通学者比率は約 15% である．このデータを元に，各市における人口の 59.3% を社会人エージェントとし，残りの人口のうち 15% を学生エージェント，その他を主婦エージェントとした．それぞれの市における人口は，各市が公表している統計データを利用した．それぞれの市におけるエージェントの数を表 2 に示す．

表 2 各市におけるエージェントの数
Table 2 The numbers of agents in each city.

市町村名	学生	社会人	主婦
明石市	18088	176123	102496
神戸市	94322	918433	534488
芦屋市	5890	57345	33372
西宮市	29385	286127	166513
尼崎市	28482	277331	161394
大阪市	162439	1581713	920488
豊中市	24341	237013	137931
茨木市	16897	164528	95748
高槻市	21723	211519	123095
長岡京市	4886	47567	27682
向日市	3308	32203	18741
京都市	86624	843472	490864

2.4 エージェントの感染状態

本研究では，エージェントを感受性（S:susceptible），潜伏状態（E:Exposed），感染性（I:infectious），回復状態（R:recovered）の 4 状態に分けることでインフルエンザの感染状況を表現する．シミュレーションの初期状態ではすべてのエージェントが状態 S であり，感染状態に応じて状態 S から状態 R へと順に遷移していく．図 4 に状態遷移の様子を示す．

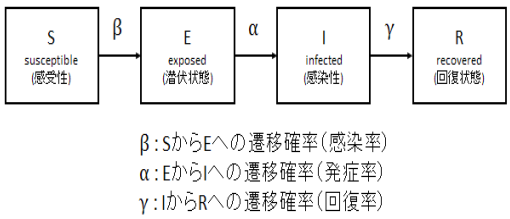

図 4 エージェントの感染状態の遷移
Fig. 4 The transient model of infection states of agents.

ここで，状態 S から状態 E への遷移確率（感染率）は β，状態 E から状態 I への遷移確率（発症率）は α，状態 I から状態 R への遷移確率（回復率）は γ で与えられる．発症率の逆数 α^{-1} と 回復率の逆数 γ^{-1} はそれぞれ感染症の潜伏期間，感染症の感染期間を示す．また，感染率 β を以下の式で定義する．

$$\beta = c\phi\frac{I(t)}{N} \tag{1}$$

定数 c は 1 エージェントが単位時間あたり何人のエージェントに接触するかを示している．また定数 ϕ は感染状態のエージェントと 1 回接触した場合に感染が起こる確率を示している．I(t) および N は，ある場所における感染者の人数と，その場所に存在するエージェントの人数である．

今回のシミュレーションでは，先行研究を参考に以上のパラメータを表 3 のように設定した [9], [10].

表 3 シミュレーションにおけるパラメータの設定
Table 3 Parameter setting used in the simulation.

- 発病率 $\alpha^{-1} = 3.5$[日]
- 回復率 $\gamma^{-1} = 3$[日]
- 各場所における感染率

パラメータ	家	学校	会社	駅	店
c[/10 分]	0.0064	0.2081	0.0256	0.0256	0.2353
ϕ	0.4	0.0575	0.0575	0.0575	0.0575

3. シミュレーションの並列化手法

ここでは，本研究で開発したマルチエージェントによるシミュレーションプログラムの並列化を行う際に用いた手法について述べる．

3.1 シミュレーションの流れ

シミュレーションのフローチャートを図 5 に示す．プログラムが実行されると，まずシミュレーションで利用されるパラメータが書かれた設定ファイルを読み込む．次に，エージェントおよび場所情報の設定を行う．この際，それぞれのエージェントがどの町に住み，どの町の学校や会社へ移動するかを乱数によって決定する．すべてのエージェントと場所情報についての初期化が終了すると，シミュレーションが実行される．

シミュレーションループでは，まず一日の初めに各エージェントの行動についての処理を行い，各エージェントが店に行く回数や場所などが決定する．その後，各エージェントについて，エージェントの状態更新，エージェントの移動が行われ，移動した先にエージェントの情報が伝えられる．そして，各場所について，エージェントから与えられた情報を元に感染率を計算する．1 日が終了した際は，各エージェントの状態を集計し，結果を出力する．以上のシミュレーションループを規定の回数行うことでシミュレーションを実行している．

3.2 MPI によるプロセス並列

MPI を利用することで，シミュレーションをプロセス並列化した．概要およびフローチャートを図 6 に示す．各

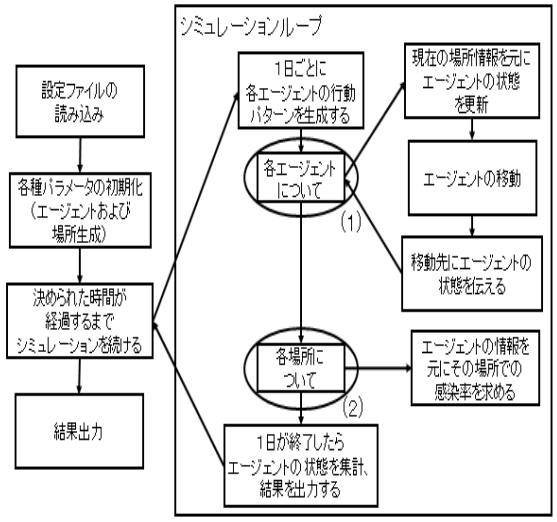

図 5 シミュレーションのフローチャート
Fig. 5 The flow chart of simulation.

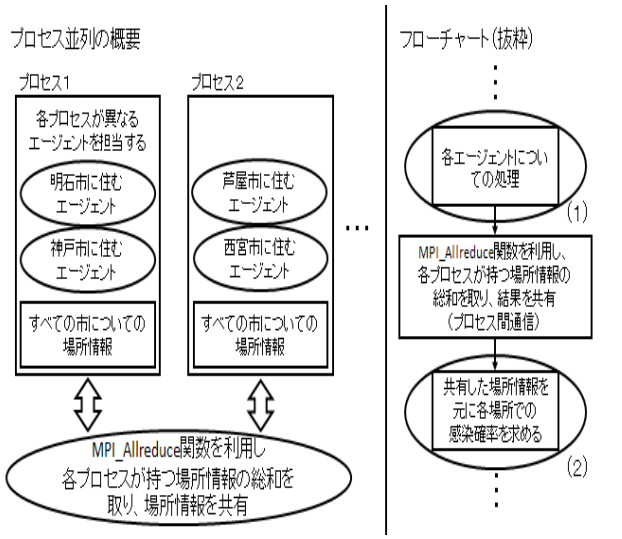

図 6 MPI によるプロセス並列の概要およびフローチャート
Fig. 6 The outline and flow chart of parallel implementation based on MPI.

プロセスは，それぞれ割り当てられたエージェントについてシミュレーションを解き進める．結果集計，初期化などの処理を簡単にするために，基本的に市単位でそこに住むエージェント郡をプロセスに割り当てる．ただし，各プロセスに割り当てるのは各市に住むエージェントのみで，各市での場所情報についてはすべてのプロセスで共有している．すべてのプロセスで場所情報を共有するために，まず各プロセスにて担当するエージェントがどの場所へ移動するかの処理を行い，場所情報を集計する．その後，各プロセスにて集計した場所情報の総和を取り，その結果をすべてのプロセスで共有する．この処理は MPI の機能である MPI_Allreduce 関数を利用し，場所情報を格納している配列について総和を取ることで行っている．MPI_Allreduce 関数を実行するタイミングは，図 6 右に示すように，各エージェントについての処理が終了した直後である．図 5 内シミュレーションループにおける丸印で囲われた（1）と（2）の間に相当する．MPI_Allreduce 関数によって場所情報を共有した後は，取得した情報を元に逐次プログラムと同様各場所について感染率を計算する．

3.3 OpenMP によるスレッド並列

OpenMP を利用することで，シミュレーションをスレッド並列化した．スレッド並列化を行ったのは，図 5 内シミュレーションループにおける丸印で囲われた（1）と（2）の部分の処理である．（1）と（2）の部分は Fortran 言語における do-loop による繰り返しとなっており，この部分を OpenMP の do 指示構文にてスレッド並列化した．シミュレーションループ（1）の部分における移動先にエージェントの状態を伝える処理では，複数のスレッドが 1 つの場所構造体にアクセスすることでデータの衝突が発生する可能性がある．これを防ぐために，場所構造体内に複数の作業用配列を用意した．

3.4 MPI と OpenMP によるハイブリッド並列

先に述べた MPI によるプロセス並列と，OpenMP によるスレッド並列を組み合わせるハイブリッド並列化を行った．OpenMP によるスレッド並列化を行ったプログラムを基本に，MPI 通信を行う処理を追加している．フローチャートとしては，図 6 における丸印で囲われた（1）と（2）の部分の処理をスレッド並列化したものとなっている．

4. シミュレーション結果

ここでは，シミュレーション結果およびシミュレーションの実行速度について述べる．

4.1 シミュレーション結果

先に述べたパラメータを利用し，シミュレーションを実行した．今回は各都市に 20 人の初期感染者を配置し，シミュレーション内における時間が 126 日経過するまでシミュレーションループを実行した．図 7 にシミュレーションの実行結果を示す．図 7 では 12 の市のうち，神戸市，大阪市，京都市に着目した．

図 7 より，感染のピークがシミュレーション開始から約 40 日の地点に存在することがわかる．そして，ピークを迎えた後は感染者数は減少をはじめ，約 80 日の時点でインフルエンザの感染は収束していることがわかる．国立感染症研究所のデータ [11] によると，過去 10 年間の全国のインフルエンザの流行は，大体 2，3 ヶ月の間にピークが現れ，そして収束している．また，各種先行研究 [9], [12] のシミュレーション結果も似たような傾向となっている．今回のシミュレーション結果は感染者数を 20 人とした上

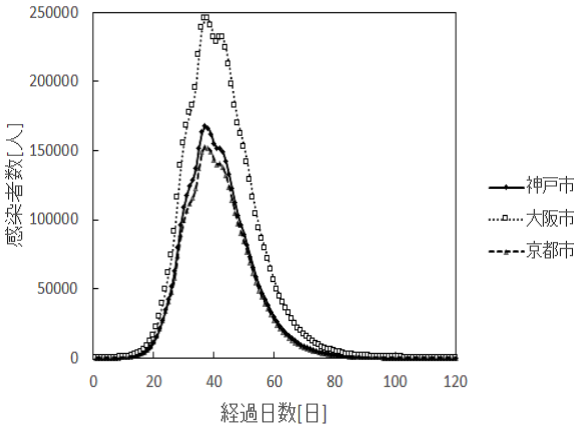

図 7　神戸市，大阪市，京都市における感染者数

Fig. 7　The numbers of infected patients in Kobe-shi, Osaka-shi and Kyoto-shi.

で，シミュレーションを 1 回だけ実行した際の結果ではあるが，感染者がほとんど居ない状況から 2,3 ヶ月の間に感染のピークが現れ，そして収束するというインフルエンザ流行の傾向が再現できている．よって，本研究で開発したシミュレーションにて，インフルエンザの流行の傾向について再現ができていると言える．

次に，市ごとの感染の様子を確認するため，神戸市とその周辺の市に着目する．図 8 に明石市，神戸市，芦屋市，西宮市，尼崎市での人口に対する感染者数の割合を示す．

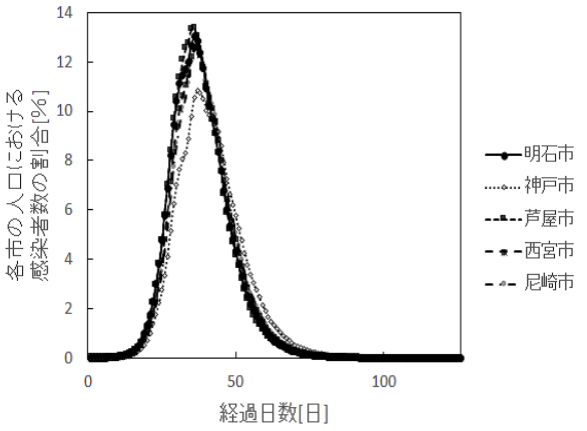

図 8　明石市，神戸市，芦屋市，西宮市，尼崎市での人口に対する感染者数の割合

Fig. 8　The percentage of infected patiens to the population in Akashi-shi, Kobe-shi, Ashiya-shi, Nishinomiya-shi and Amagasaki-shi.

図 8 より，ピーク時における神戸市の人口に対する感染者数の割合は約 10 %，その他の都市では人口に対する感染者数の割合が約 13 %となっており，神戸市に比べその周辺都市の方が感染の様子が激しいことがわかる．これは，周辺都市では神戸市へ通勤，通学するために電車を利用するエージェントが多く，神戸市のエージェントと比べ周辺都市のエージェントは感染者と接触する機会が多くなっているためと考えられる．

4.2　シミュレーションの実行速度

先に述べたシミュレーションの並列化手法について，シミュレーションの実行速度を測定し性能評価を行った．シミュレーション内における時間が 126 日経過するまでシミュレーションループを実行した．時間計測区間は，図 5 に示すシミュレーションループの開始時から終了時までである．実行速度測定においては，京都大学に設置されているスーパーコンピュータ Cray XC30 を利用した．今回利用したノード数は 1 つである．主な緒元を表 4 に示す．

表 4　Cray XC30 の主な緒元

Table 4　The specifivations of Cray XC30 at Kyoto University.

プロセッサ	Intel Xeon E5 系
ノード数	416
プロセッサ数 (コア数)/ノード	2(28)
理論性能/ノード	1030.4 [GFlops]

4.2.1　MPI によるプロセス並列

MPI によるプロセス並列を実装したプログラムについて，シミュレーションの実行速度を測定した．各プロセスにおいて担当するエージェントの数は一定としている．図 9 にプロセス数を 1 から 16 まで変化させたときのシミュレーションの実行速度の変化を示す．この図における加速率は，プロセス数が 1 の状態に比べ，どの程度シミュレーションの実行速度が高速になったかを示す比である．

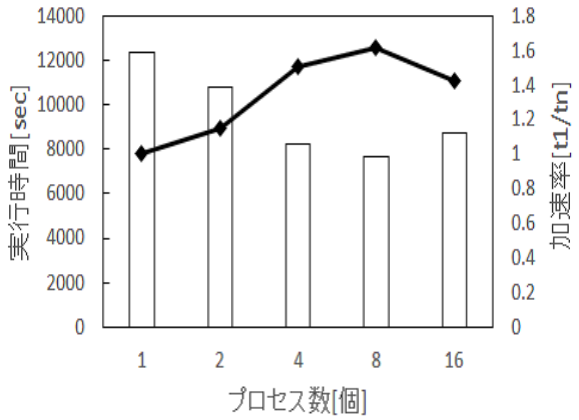

図 9　MPI 並列実装プログラムのプロセス数に対するシミュレーション実行時間および加速率

Fig. 9　The execution time and the acceleration rate of the developed simulator as a function of the number of processes.

図 9 より，プロセス数が 8 の場合においても加速率は 1.6 程度となっており，あまり台数効果が得られていない

ことがわかる．また，プロセス数を 8 から 16 にしたとき加速率が低下していることがわかる．主な原因としては，MPI_Allreduce 関数により全プロセスで場所情報を共有する際に掛かる通信コストであると考えられる．より良い台数効果を得るためには，通信コストを削減する新たなアルゴリズムの考案が必要である．

4.2.2 OpenMP によるスレッド並列

OpenMP によるスレッド並列を実装したプログラムについて，シミュレーションの実行速度を測定した．測定方法は MPI によるプロセス並列の場合と同様である．図 10 にスレッド数を 1 から 16 まで変化させたときのシミュレーションの実行速度の変化を示す．この図における加速率は，スレッド数が 1 の場合に比べ，どの程度シミュレーションの実行速度が高速になったかを示す比である．

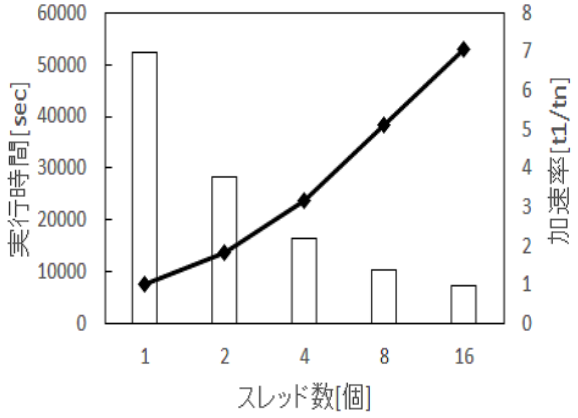

図 10 OpenMP 並列実装プログラムのスレッド数に対するシミュレーション実行時間および加速率

Fig. 10 The execution time and the acceleration rate of the developed simulator as a function of the number of threads.

図 10 より，スレッド数の増加に比例して加速率も上昇していることがわかる．加速率は 16 スレッドの場合に 7 程度となっている．加速率の飽和もあまり見られず，更なるスレッド数の増加による高速化が見込まれる．

4.2.3 MPI と OpenMP によるハイブリッド並列

MPI によるプロセス並列と，OpenMP によるスレッド並列を組み合わせたハイブリッド並列を実装したプログラムについて，シミュレーションの実行速度を測定した．利用する CPU コア数を 16 に固定し，その中でプロセス数とスレッド数を変化させた．図 11 にシミュレーションの実行速度の変化を示す．この図における加速率は，OpenMP によるスレッド並列プログラムにおけるスレッド数 1 の場合の実行速度に比べ，どの程度シミュレーションの実行速度が高速になったかを示す比である．

図 11 より，プロセス数を増加させると加速度は減少していることがわかる．これは，図 9 よりプロセス数を増や

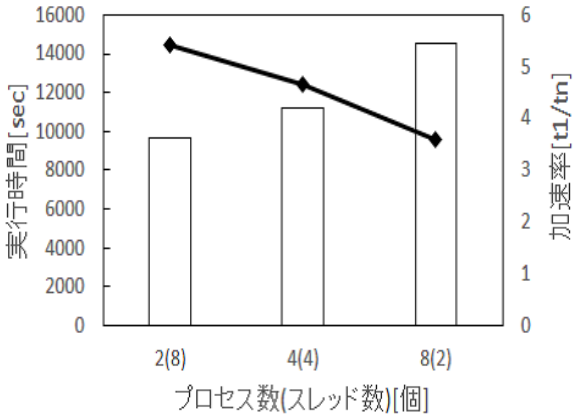

図 11 ハイブリッド並列実装プログラムのプロセス数 (スレッド数) に対するシミュレーションの実行時間および加速率

Fig. 11 The execution time and the acceleration rate of the developed simulator as a function of the number of processes(threads).

した場合の加速率の増加が少なく，スレッド数の減少による速度減少をカバーしきれなかったためと考えられる．また，どのプロセス数においても，OpenMP によるスレッド並列プログラムにおいてスレッド数を 16 にした場合の実行速度よりも遅くなってしまっている．OpenMP によるスレッド並列プログラム以上の速度向上を目指すためには，特に MPI によるプロセス並列プログラムの通信量の削減などの工夫に取り組む必要がある．

5. まとめ

本研究では，まずマルチエージェントを用いたインフルエンザの伝播シミュレーションプログラムを開発した．次にシミュレーションの実行時間を短縮するため，MPI を用いたプロセス並列，OpenMP を用いたスレッド並列，これらを組み合わせたハイブリッド並列を実装し，その性能評価を行った．

本研究にて開発したシミュレーションによって，日本でのインフルエンザの流行の様子や，先行研究と似た傾向を持つ結果が得られた．これにより，本研究にて開発したシミュレーションがインフルエンザの流行の傾向を再現できることを示した．

シミュレーションの実行速度については，OpenMP によるスレッド並列を用いることで，16 スレッドを利用した際に約 7 倍の速度向上が得られた．16 スレッドの段階で加速率の飽和は見られなかったため，更なるスレッドの増加による高速化が期待できる．一方，MPI によるプロセス並列ではプロセスを 8 以上に増やすとプログラムの実行速度が低下してしまった．また，ハイブリッド並列においても期待された速度向上を得ることができなかった．プロセス並列やハイブリッド並列によるシミュレーションの更なる高速化のためには，MPI 通信によるコストを削減する新たな

アルゴリズムの考案に取り組む必要がある．

謝辞 本研究のシミュレーションでは京都大学学術情報メディアセンター全国共同利用スーパーコンピュータシステム（電波科学計算機実験装置:KDK）を利用致しました．ここに感謝の意を表します．

参考文献

[1] Kaufmann, S. H., Schädlich, S. and Wiegandt, K.: *The new plagues: pandemics and poverty in a globalized world*, Haus (2009).
[2] Longini, I. M., Nizam, A., Xu, S., Ungchusak, K., Hanshaoworakul, W., Cummings, D. A. and Halloran, M. E.: Containing pandemic influenza at the source., *Science (New York, N.Y.)*, Vol. 309, No. 5737, pp. 1083–1087 (online), DOI: 10.1126/science.1115717 (2005).
[3] Chao, D. L., Halloran, M. E., Obenchain, V. J. and Longini, Jr, I. M.: FluTE, a Publicly Available Stochastic Influenza Epidemic Simulation Model, *PLoS Comput Biol*, Vol. 6, No. 1, p. e1000656 (online), DOI: 10.1371/journal.pcbi.1000656 (2010).
[4] Mao, L.: Modeling triple-diffusions of infectious diseases, information, andpreventive behaviors through a metropolitan social network —An agent-based simulation，*Applied Geography*, Vol. 50, No. 0, pp. 31 – 39（online），DOI: http://dx.doi.org/10.1016/j.apgeog.2014.02.005 (2014).
[5] Ohkusa, Y. and Sugawara, T.: Application of an individual-based model with real data for transportation mode and location to pandemic influenza, *Journal of Infection and Chemotherapy*, Vol. 13, No. 6, pp. 380–389 (online), DOI: 10.1007/s10156-007-0556-1 (2007).
[6] 祐樹豊坂，英雄廣瀬：MADE によるパンデミックシミュレーション：マルチエージェントと微分方程式モデルの組み合わせ，情報処理学会研究報告，No. 1, pp. 1–6 (2009).
[7] 経済産業省編：平成 19 年度商業統計表（二次加工統計表） 第 10 表 商業集積地区（商店街）の都道府県別、市区町村別の商店街数、事業所数、大店舗数、大店舗内事業所数、従業者数、年間商品販売額及び売場面積 (2009). `www.meti.go.jp/statistics/tyo/syougyo/result-2/h14/xls/niji/ritchi.xls`.
[8] 総務省統計局編：第 64 回日本統計年鑑 16-2 年齢階級，就業状態別労働力人口 (2014). `http://www.stat.go.jp/data/nenkan/zuhyou/y1602000.xls`.
[9] Cooley, P., Brown, S., Cajka, J., Chasteen, B., Ganapathi, L., Grefenstette, J., Hollingsworth, C. R., Lee, B. Y., Levine, B., Wheaton, W. D. and Wagener, D. K.: The role of subway travel in an influenza epidemic: a New York City simulation., *Journal of urban health : bulletin of the New York Academy of Medicine*, Vol. 88, No. 5, pp. 982–95 (online), DOI: 10.1007/s11524-011-9603-4 (2011).
[10] Real-Time, R.: Emergence of a novel swine-origin influenza A (H1N1) virus in humans, *N Engl j Med*, Vol. 360, pp. 2605–2615 (2009).
[11] 国立感染症研究所感染症情報センター：過去 10 年間との比較グラフ（週報）（インフルエンザ）. `http://idsc.nih.go.jp/idwr/kanja/weeklygraph/01flu.html`.
[12] Saito, M. M., Imoto, S., Yamaguchi, R., Tsubokura, M., Kami, M., Nakada, H., Sato, H., Miyano, S. and Higuchi, T.: Enhancement of Collective Immunity in Tokyo Metropolitan Area by Selective Vaccination against an Emerging Influenza Pandemic, *PLoS ONE*, Vol. 8, No. 9, p. e72866 (online), DOI: 10.1371/journal.pone.0072866 (2013).