Implementation of Massive Agent Model Using Repast HPC and GPU

Shinsaku Segawa, Shofuku Kin, Hidenori Kawamura, Keiji Suzuki
Hokkaido University, Sapporo, Hokkaido, Japan

ABSTRACT

Agent Based Model (ABM) is efficient for analysis of various social mechanisms. Recently, there are many studies on massive agent model to explain more complex social phenomena.
Then, we aim for implementation of large scale simulation model using Repast HPC toolkit, a platform for massive agent model. In this article, we build "Schelling Segregation Model" for spatial model using geospatial data provided OpenStreetMap, an open source project creating a free editable map. In this model, agents are located continuous space, not grid in original. When an agent is "unhappy" and migrates to new location, it costs agents some simulation time depending on distance between old location and new one.
This article reports simulation results using Japanese cities and verification result about execution time.

Keywords: Multi Agent Simulation, Social Simulation, High Performance Computing

1. INTRODUCTION

Recently, agent based models (ABMs) become larger, high performance computing systems makes it possible to perform simulations at huge scales only a few years ago [1]. ABMs are efficient for the simulation of objects whose mathematical modeling is difficult [2].
ABMs are seen as attractive approach to reproducing and analyzing diverse social systems including autonomous and heterogeneous decision making entities, i.e., humans [3]. Researchers are possible to explore how through the interaction of many individuals more emergent phenomena arise. Moreover, ABMs allows for practitioners to build models of complex social phenomenon by simulating the interactions of the many actors in such systems. Thus gaining insights that will lead to greater understanding and, in some cases, better management of the behavior of complex social systems [4].
It is reported that in some cases simulation results are significantly different depending on simulation scale [2]. Massive ABMs platform is required for building model of large social systems.
In this article, it is introduced to promote large ABM tool.
This article introduces about implementation of agent model on Repast HPC, ABMs toolkit for large scale.
This article’s aim is to detect bottleneck of simulation through implementation of simple model that is extension of Schelling segregation model using geo-spatial data.
The rest of this article is organized as follows. Sec.2 shows detail of Repast HPC framework, Sec.3 introduces related works for building models on Repast HPC and some toolkit for large scale ABMs. Sec.4 shows the implementation of the model. Sec.5 describes an analysis of simulation. In Sec6, we conclude this paper with future works.

2. Repast HPC

Repast High Performance Computing (Repast HPC) [5] is an agent based modeling and simulation framework for high performance distributed computing platforms written in C++ and using MPI for parallel operations. Most of the other open source ABM software does not support parallelization (See Sec.3.). Repast HPC is designed for parallel environments where many processes are running in parallel and where the agents themselves are distributed across processes. In addition of being a parallel platform, the fact that Repast HPC supports parallelizing the simulation world at prominent modeling methodologies (shared grid, shared network and shared continuous space) independently or in combination, makes it excellent software for high performance computing.
The functions specific to the HPC environment deal with the theoretical challenge of sharing agent information across multiple processes. Repast HPC deals with this through a two-stage process. In the first, agents are 'requested' from other processes; the process that manages the agent will answer a request by providing the other process with a copy of the original agent. In the second, these copies are updated with the borrowed agents' current information. Fig.1 shows an example of sharing agent information across multiple processes. The effect is that an agent on a given process can derive information from and react to the current state of agents on any other process. In the simulation presented here the 'request' for other agents is explicitly scheduled, Repast HPC also provides a way for spatially located and moving agents to interact with agents near themselves in simulation space but across a process boundary; when this functionality is used both the request and synchronization are handled automatically. Fig.2 shows an example.

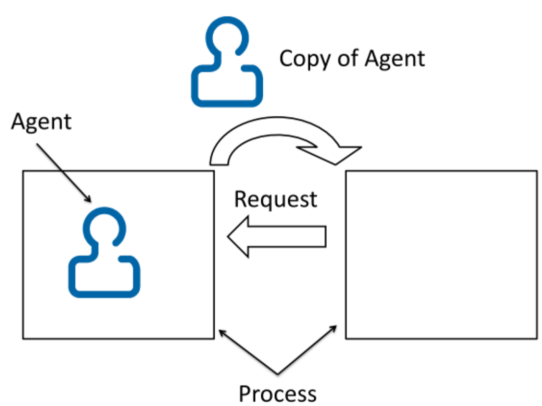


Fig. 1Requesting Agents Across Processes

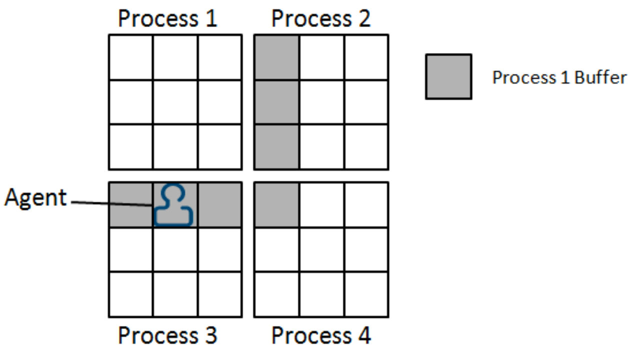


Fig.2 Synchronized Buffer

3. Related Work

In this section, ABM tool kit and the study case about the large scale model using Repast HPC is introduced.

3.1 ABM Toolkit

Many ABM simulation frameworks have been developed and shared among research communities. There are some reports that review ABM tools [6] [7]. Swarm [8] is the first ABM software, launched in 1994 at the Santa Fe Institute. Swarm supports hierarchical modeling approaches whereby agents can be composed of swarms of other agents in nested structures. Swarm provides object oriented libraries of reusable components for building models and analyzing, displaying, and controlling experiments on those models. MASON [9] is a single-process discrete-event simulation core and visualization library written in Java, designed to be flexible enough to be used for a wide range of simple simulations, but with a special emphasis on swarm multi-agent simulations of many (up to millions) of agents. It provides a toolkit to execute high speed simulations with many agents in a single process. ZASE (Zillions of Agents-based Simulation Environment) [Yamamoto 08] is a scalable platform for multi-agent simulations potentially using billions of agents. The purpose of ZASE is to develop ABM applications on multiple general computers such as PCs or workstations where they are connected with a high performance but generally available network such as gigabit Ethernet. ZASE aims at performing even larger-scale simulations by managing many agents in each process and by combining the processes hierarchically. SOARS (Spot Oriented Agent Role Simulator) [10] aims to support a PC grid. It provides a common platform for education and real world applications, and it is easy to develop simulation models with its GUI and also to visualize its simulation results.

3.2 Study Case Using Repast HPC

Only a few examples of Repast HPC are available because it is a new toolkit for ABM, first version has been released in 2010. It is expected that more examples of multi agent simulation on Repast HPC are given in future.

In [1], an example model called "Triangles simulation" in Repast HPC is shown and run at scales up to billions of agents. And practical issues of performance, network analysis, file output and data visualization are illustrated with example. It is reported that execution time scales almost linearly with the number of cores for mainly independent agents, whereas speed-up drops significantly in case of highly interdependent agents.

Recently, ABMs are seen as attractive approach to analysis Computational Social Science (CSS). In [11], models of configurations of social agents at a massive scale are introduced. The paper reports implementation of Cellular Automata (CA) for spatial model, to use urban evacuation analysis.

4. Implementation

In this article, to satisfy three requirements, 1) interaction between agents, 2) agent's movement, 3) agent placement using geographic information, “spatial model” that is extension of Schelling Segregation Model using geospatial information is implemented.

Segregation model, proposed by Thomas Schelling [12], attempts at understanding the phenomenon of residential segregation by considering it as an aggregated result of the decisions of residents in choosing their housings. Residents are represented explicitly as agents in the model. Residents having certain similarities are classified as belonging to the same class. In reality, criteria used to classify resident may be educational level, religion, annual income, skin color, political point of view, etc. Same class residents are represented by same color agents. When the simulation runs, we see groups of nearby same color residents appear. We call these groups emerging structures because they are not explicitly represented in the model as agents. As soon as these emerging structures appear in the simulation, isolated residents tend to be attracted to them. Isolated residents move to join group of other residents similar to them. The forming of such emerging structures is the result of residents’ decision in choosing places where they live in a city. Vice-versa, these emerging structures have certain feedback influences on the behaviors of the residents. Fig.3 shows an example of Segregation Model. In this figure, circle and cross represent same class residents.

In the original model, agents are located on 8×8 grid. In this model, each cell represents “agent’s residents”. In the suggested model, residents are placed along the road, which is continuous space extracted from GIS. It is based on an assumption that typical housing is located along the road.

In spatial model, geospatial data used as GIS is provided by OpenStreetMap (OSM) [13]. OSM is an open source project for creating a free editable map, and geographic information can be obtained easily. OSM provides a file named "planet.osm", an xml format file. This file includes in one file; all the nodes, ways and relations made up by OSM community members.

In this model, road information is extracted from planet.osm and used to make residents. Information about road is explained by "node" elements and "way" elements in osm file. A node defines a single geospatial point using a latitude and longitude. In many case, a node represents a crosspoint. Each node stores its node-ID, longitude and latitude. A way is an ordered list of between 2 and 2000 nodes. A way element describes road, e.g. street, railway and highway. Ways can also represent areas, such as buildings or forests. In this case, the first and last node will be the same - a "closed way". Fig.4 describes each element. Road information is can be obtained by extracting "node" and "way" elements from osm file.

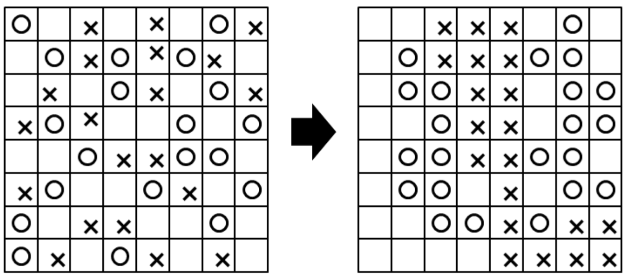


Fig.3 Schelling Segregation Model

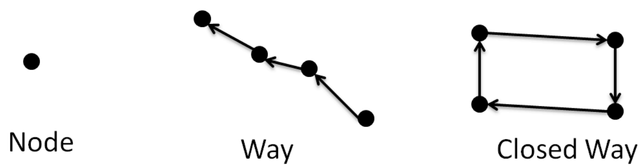


Fig.4 Elements in osm file

5. Simulation and Analysis

In this article, simulation is processed on Japanese cities to test the influence of the geographic information on the simulation. Table shows a list of cities simulated in this article. Four rows, left, bottom, right and top represent longitude and latitude of extracted city. Number of residents depends on total road distance. In the simulation, agents are separated into two groups. Each group consists of 500,000 agents as enough number of large-scale agents. Agents are given their residential randomly when simulation starts. If an agent is in an "unhappy" situation, it starts migration to empty chosen randomly. The simulation continues until agents are happy, and this represents the simulation time distance between new residence and old one. In this simulation, it cost agent 1 simulation time per 5km as real world.

In the original model, an agent refers neighbor agents and decides migration. In the original model, “neighbor” is defined 4cells (Von Neumann Neighborhood in Cellular Automata). In the spatial model, new search method is needed to search continuous space. In this article, as referred Fig.5, city map is converted to 100 × 100 grids and “neighbor” is defined agents on same cells. Fig.6 shows an example. A small example of simulation is shown in Fig.7.

The part of city data is obtained from planet.osm via Osmosis, a command line Java application for processing OSM data. This can extract a rectangle. An example of command extracting the part of data from osm file is described below.

```
$ bzcat planet.osm.bz2 | osmosis --rx - --bb left=141.2035
   bottom=42.9469 right=141.4754 top=43.1413 --wx sapporo-city.osm
```

In this article, ratio that an agent hopes that neighbor agents are same group is called “preference”. Simulation is processed on 3

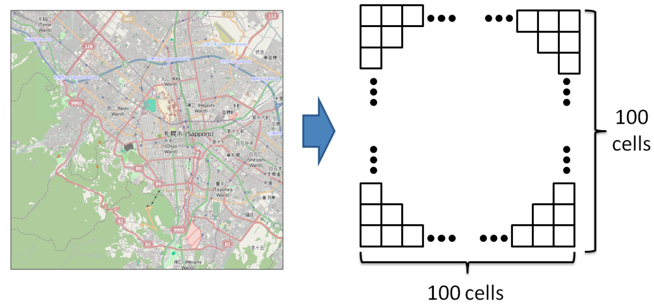


Fig.5 Conversion map data to 100 × 100 grid

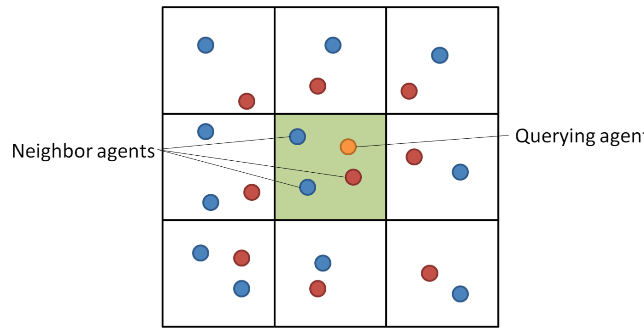


Fig.6 Definition of neighbor agent on suggested model

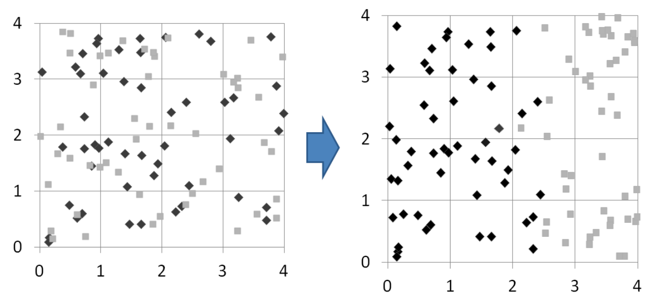


Fig.7 Spatial model

preferences, 30%, 40% and 50% to investigate the relation between preference and execution steps. On each simulation, all agents have same preferences. The graph of the simulation steps for number of “unhappy” agents is shown Fig.8. Simulation cost many steps as preferences increases. Fig.9 shows the graph of the simulation steps for average of moving day of agents. We can see that average of moving day increases as map size is larger.

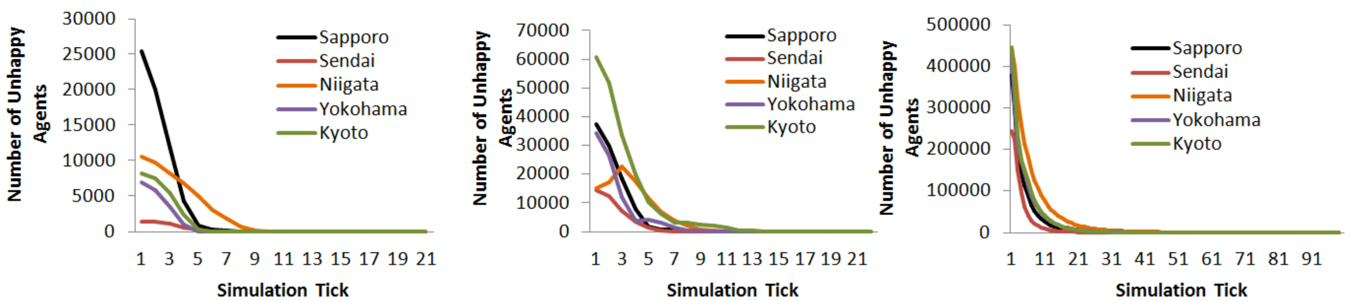
Execution time of simulation is discussed as below. To detect bottleneck of simulation, Simulation step is mainly divided into two phases, initial phase and running phase. Initial phase is a phase initializing simulation, creating agent objects and locating agents into grid or space. Running phase is a main part of the simulation, iteration of searching neighbor agents and migration to other location. Fig.10 and Fig.11 show transition of execution time. Each graph shows execution time increases as number of agents increases. Initial phase is more divided in detail because it costs longer time than run phase. It is shown that a part for giving agents coordinates and locating on grid or space takes most execution time. In this part, Repast HPC API is used. The API is repast::Grid, functions for discrete grids and continuous spaces. It is expected that execution is more efficient by using C++ standard library instead of API.

6. Conclusion

In this article, we verified about large scale ABM using epast HPC which is ABM toolkit. We constructed segregation model which is added the geospatial information in the real world to the Schelling Segregation Model, and in an agent's migration, we added the concept of move cost according to distance. In order to check influence that a geospatial situation has on a simulation, we simulated 1 million agents using the map data which is the city in Japan. We also divided the simulation into two processing parts, and then measured execution time of each division to specify the bottleneck of processing time that is caused by large-scale simulation.

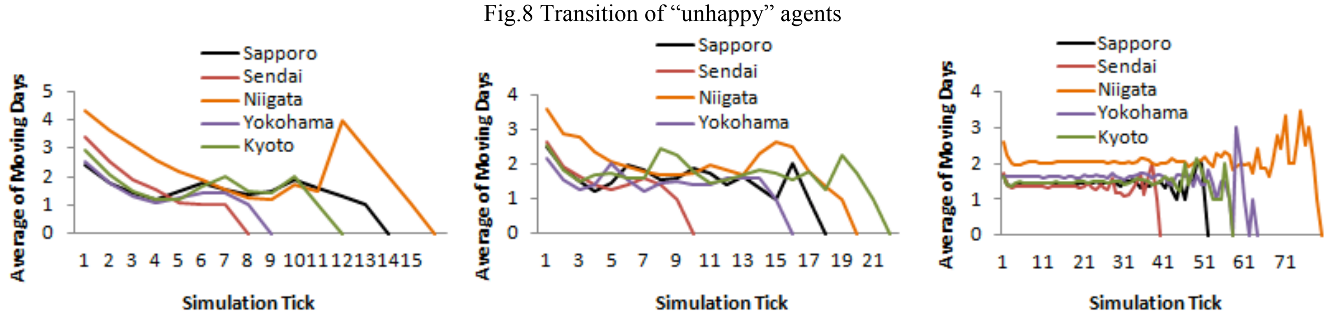
City	Sapporo	Sendai	Niigata	Yokohama	Kyoto
Left	141.2035	140.6595	138.9273	139.5094	135.6496
Bottom	42.9469	38.2192	37.7020	35.3741	34.8859
Right	141.4754	141.0131	139.3873	139.7226	135.8823
Top	43.1413	38.3571	38.0440	35.5136	35.0744
Area()	479.5	474.7	1542.5	300.5	445.9
Number of Residences	1,698,750	1,445,642	1,904,392	1,421,162	1,935,092

Table. 1 List of Simulated Cites



(a) preference 30% (b) preference 40% (c) preference50%

Fig.8 Transition of "unhappy" agents



(a)preference30% (b) preference40% (c) preference50%

Fig.9 Transition of average of moving day

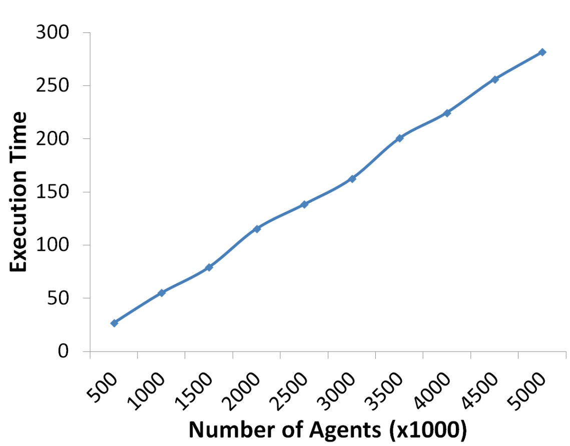


Fig.10 Transition of initial phase

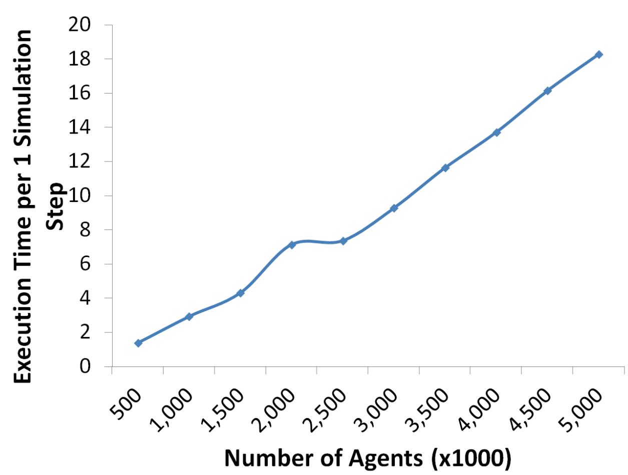


Fig.11 Transition of run phase

7. REFERENCES

[1] John T. Murphy, Computational Social Science and High Performance Computing, A Case Study of a Simple Model at Large Scales, The Computational Social Science Society of the Americas, (2011)

[2] Gaku Yamamoto, Hideki Tai, and Hideyuki Mizuta, A Platform for Massive Agent-Based Simulation and Its Evaluation, Massively Multi-Agent Technology, pp. 1 - 12, (2008)

[3] Yuu Nakajima, Shohei Yamane, and Hiromitsu Hattori, Multi-model Based Simulation Platform for Urban Traffic Simulation, Principles and Practice of Multi-Agent Systems, pp. 228 - 241, (2012)

[4] A.J.Heppenstall, A.T.Crooks, L.M.See & M.Batty, The integration of agent-based modelling and geographical information for geospatial simulation, Agent- based models of geographical systems, pp. 219 - 252, (2012)

[5] http://sourceforge.net/projects/repast/files/Repast \\HPC/Repast HPC 1.0.1/repasthpc-1.0.1.tar.gz/\\download

[6] Steven F. Railsback, Agent-based Simulation Platforms - Review and Development Recommendations,

Fig.10 Transition of initial phase

Society for Computer Simulation International, Vol. 82, Issue 9, pp. 609 - 623, (2006)

[7] CM Macal, MJ North, Agent-based modeling and simulation, WSC '09 Winter Simulation Conference, pp. 96 -98, (2009)

[8] Nelson Minar, Roger Burkhart, Chris Langton, The swarm simulation system, A toolkit for building multi-agent simulations, pp. 96-06-42, (1996)

[9] Sean Luke, Claudio Cioffi-revilla, Liviu Panait, Keith Sullivan, MASON, A Multiagent Simulation Environment, Simulation, Vol. 81, No. 7, (2005)

[10]Hideki Tanuma, Hiroshi Deguchi, Tetsuo Shimizu, SOARS, Spot Oriented Agent Role Simulator- Design and Implementation, Agent-Based Social Systems, Vol.1, pp. 1-15, (2005)

[11]Zia, K. Riener, A. Farrahi, K. Ferscha, A New Opportunity to Urban Evacuation Analysis - Very Large Scale Simulations of Social Agent Systems in Repast HPC, Principles of Advanced and Distributed Simulation (PADS), 2012 ACM/IEEE/SCS 26th Workshop on, pp. 233 - 242, (2012)

[12]Thomas C. Schelling, Dynamic Models of Segregation, The Journal of Mathematical Sociology, Volume 1, Issue 2, (1971)

[13]http://www.openstreetmap.org/