17th World Conference on Earthquake Engineering, 17WCEE
Sendai, Japan - September 13th to 18th 2020

Development of High-performance Computing Extension for an Agent Based Economic Model for Fine-grained Post-disaster Economy Simulations

A. Gill[(1)], M. Lalith[(2)], S. Poledna[(3)], M. Hori[(4)], T. Ichimura[(5)], K. Fujita[(6)]

[(1)] *PhD candidate, Dept. of Civil Engineering, The University of Tokyo, Bunkyo-ku, Tokyo, Japan. Email: gill@eri.u-tokyo.ac.jp*
[(2)] *Assoc. Prof., Earthquake Research Institute, The University of Tokyo, Bunkyo-ku, Tokyo, Japan. Email: lalith@eri.u-tokyo.ac.jp*
[(3)] *Research Scholar, International Institute for Applied Systems Analysis, Laxenburg, Austria. Email: poledna@iiasa.ac.at*
[(4)] *Director-General, Research Institute for Value-Added-Information Generation, Japan Agency for Marine Earth Science and Technology, Yokohama, Japan. Email: horimune@jamstec.go.jp*
[(5)] *Prof., Earthquake Research Institute, The University of Tokyo, Bunkyo-ku, Tokyo, Japan. Email: ichimura@eri.u-tokyo.ac.jp*
[(6)] *Assis. Prof., Earthquake Research Institute, The University of Tokyo, Bunkyo-ku, Tokyo, Japan. Email: fujita@eri.u-tokyo.ac.jp*

Abstract

The heavy dependency of an economic entity on other economic entities, lifeline utilities and other infrastructures makes a national or even regional economy vulnerable to localized natural disasters like major earthquakes since this heavy dependency can cause the local economic impacts to cascade to the whole nation over many coming years. When making recovery plans following a major disaster, it is vital to take economic recovery into account. With widely used standard economic models, it is not possible to relate each physical entity to be restored, while taking interdependencies of economic entities into account, to the performance of the national economy. That can only be achieved using a fine-grained economic model which can accommodate all the complex economic interactions of millions of heterogeneous economic entities present in an economy. Agent-based economic models (ABEMs) simulate an economy as a time-step-driven system of interacting heterogeneous and autonomous agents, and therefore, are best suited for studying economic conditions like post-disaster economy. Although various ABEMs capable of including the complex interactions of real-world economic entities are available, meeting the computational demand of simulating 1:1 scale model of a major economy with hundreds of millions of agents is the major hurdle in utilizing those in disaster recovery. To overcome this hurdle, we developed a distributed memory parallel extension for ABEMs based on Message Passing Interface (MPI). The agents interact over several topological graphs some of those, like banks-firms interaction, are centralized graphs, whereas others like goods market are scale-free graphs. To distribute balanced workload among MPI processes, we partitioned the agents based on a representative employer-employee interaction graph while making all the other graphs available at a minimum communication cost. Using a number of techniques, like mimicking real world solutions, organizing events to maximize communication hiding, etc., we minimized the communication overhead. According to our literature surveys, our code has not only a several orders of magnitudes higher performance and strong scalability as compared to other large-scale agent-based models, but also is the only known code capable of simulating more than several tens of million agents. Details of our parallel implementation, and a demonstrative simulation of hypothetical post-disaster economy simulations of Japanese economy having 150 million agents, are presented in this paper. The developed system is a useful tool in identifying problematic areas of the economy, which can be invaluable in making efficient recovery plans and achieving fast recovery from the disaster.

Keywords: Agent-based Economic Models (ABEMs), Economic recovery, High-performance computing (HPC), MPI.

1. Introduction

Recovery plans generated by traditional methods are not suitable for mitigating big disasters like the anticipated Nankai Trough earthquake in Japan. These methods are highly coarse grained as they don't include interdependency of various infrastructure components and don't consider interdependency of infrastructure components and economic activities. This interdependency of economic activities and the infrastructures makes an economy highly vulnerable to damages to infrastructure caused by major disasters. Moreover, the local economic impacts induced by disasters can cascade to the whole nation and to many coming years because of the interdependency, thereby pushing a nation into harsh economic conditions. It is unwise to focus only on recovering the infrastructures, ignoring how the recovery of each infrastructural component contributes to the economic recovery both in short and long terms. To increase the resilience to

disasters, as well as to find efficient recovery plans for faster recovery of economy and the infrastructure, economy and infrastructure must be simulated together including all their interdependencies. With widely used standard economic models, it is difficult to relate each physical entity to be restored, while taking interdependencies into account, to the performance of the national economy. Therefore, we need a fine-grained economic model which can accommodate all the complex economic interactions of millions of heterogeneous economic entities present in an economy. Agent-based economic models (ABEMs) simulate an economy as a time-step-driven system of interacting heterogeneous agents, and therefore, are best candidate for studying economic conditions like post-disaster economy.

Many ABEMs have been developed for studying various economic phenomenon ranging from studying the impacts of credit networks on the growth rate [2], effects of capital and credit available to consumer goods producing firms [1]. The most complete ABEMs that simulate all the economic entities of a national economy are EURACE developed by Diessenberg et al. [6] and Poledna et al. [7]; where the former model is developed for simulating the Eurozone economy and the latter is more generalized model which can be applied to any economy of any size. The agents in these models follow similar behavior rules. We choose Poledna et al. [7] for our ABEM since it has several advantages: the initial data for agents is derived from the available economic data as per the System of National Accounts (SNA) of the economy, the agents follow the same accounting procedures as laid down in SNA, credit based goods market, and most importantly the model has reproduced past events [12].

Although Poledna et al. [7] provides all the necessary economic entities and their behavior rules, meeting the computational demand of simulating 1:1 scale model of an economy with hundreds of millions of agents is the major hurdle in using it to perform fine-grained economic simulations. Deissenberg et al. [5] tries to simulate the Eurozone economy having hundreds of millions of agents using parallel computers. However, their implementation describes only a small hypothetical labor market, and no literature is available to assess their complete implementation.

Motivated by the need of a High-Performance Computing (HPC) enhanced ABEM to have fine-grained economic simulations, we developed a distributed memory parallel extension for the ABEM based on Message Passing Interface (MPI). Although our current implementation is based on ABEM by Poledna et al.[7], the developed system can easily accommodate other ABEMs since most ABEMs share a common structure. Balanced distribution of computation workload among various MPI-processes is the first and the most important task in any parallel implementation. A major difficulty in assigning balanced workload is the agents' random and bi-directional interactions over several topological graphs. Some of those interactions are over centralized graphs like bank-firms, whereas others are scale-free graphs like goods market. To distribute balanced workload among MPI processes while ensuring the partitions to be geographically non-overlapping, we partitioned the agents based on a representative employer-employee interaction graph while making all the other graphs available at a minimum communication cost. When working with fine grained information of a disaster struck area, the condition that the partitions are geographically non-overlapping enables to efficiently access large volume of input data, related to the level of damages of millions of structures, disruptions to the supply chain networks, etc., of the damaged area. To minimize the number of communications, we mimicked real-life solutions like introduction of recruitment agencies, sales outlets, local banks, and local governments in each process. To efficiently communicate among processes, we used MPI derived data structures and latest MPI functions. Non-blocking MPI-functions were used to fully utilize the communication time by overlapping computation and communication. We demonstrate the computational performance of our implementation by simulating two economies: one having 10 million agents and the other having 330 million agents. The two examples show that our implementation is capable of exceeding 60% strong scalability, irrespective of the highly random interactions among agents. Further, we present a simulation of a hypothetical disaster, with a model of Japanese economy consisting of 125 million agents, as an early demonstration of potential application of the developed system.

The rest of the paper is organized as follows. Section-2 discusses the ABEM with a perspective of HPC implementation. The challenges in implementing HPC have been explained in detail. Section-3 explains our implementation scheme starting from domain decomposition to the solutions of challenges discussed in section-2. The performance of our implementation has been presented in section-4. The final section demonstrates one of the possible applications of the developed tool.

2. ABEM: An HPC Point of View

A typical ABEM simulates the economy as a time-step-driven system of interacting agents. The agents are autonomous and heterogeneous in nature; and interact with other agents in various markets. An agent may be active on more than one market at a time and takes various decisions to satisfy its needs and desires. The rules defining the agents' behavior may vary depending on the ABEM, however in most of the ABEMs, the agents' behavior is defined by simple, pre-defined rules. The actions/decisions of the agents are governed by their current state as well as their experience. Every interaction between two agents in a market results in an exchange of some data between the two. Most of the markets evolve with time depending on the satisfaction or dissatisfaction of the agents with their current counterparts.

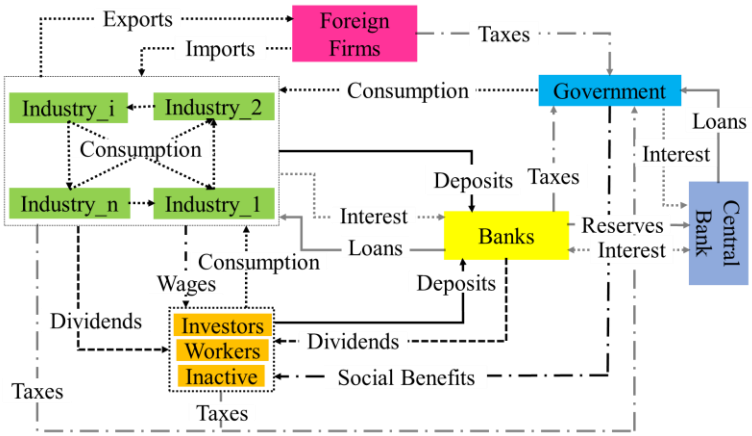


Fig. 1 – Schematic diagram of ABEM

Fig.1 presents a schematic diagram of a typical ABEM, based on which our HPC extension is developed. The model consists of various agents: firms as part of industries, foreign firms, households consisting of investors, workers and inactive households, bank, central government, and central bank. All the agents have a physical location from where they operate. The firms interact with workers in labor market, with consumers in goods market, with bank in credit market, and with government for paying/receiving taxes/subsidies. The workers interact with firms in labor market and goods market, with bank in credit market, and with government to pay tax or receive social benefits. The interaction graph of agents in various markets can be categorized as centralized graphs like credit market and scale free graphs like labor market. In a parallel implementation, each interaction between two agents located in different processes generates a bi-directional communication between the involved processes. The following sections describe the interaction graphs in more details and explain the challenges in their scalable HPC implementation.

2.1 Interactions over centralized graphs

The interactions of bank and government with other agents take place over centralized graphs. Some of the interactions, like paying tax to the government, are unidirectional while others, like applying for loans to the bank, are bidirectional. Unidirectional interactions involve some data transfer from the source agent to the target agents; and are easy to parallelize as all such interactions emanating from a process can be clubbed into one interaction which requires only one communication. On the other hand, bidirectional interactions consist of three stages: data transfer from the source agent to the target agent, some decision making by the target agent based on the data received and a reply to the source agent. Thus, these interactions require two or more communications and cause huge load imbalance because of serialized decision making. Fig. 2 shows a typical interaction scheme of bank and its customers.

2.1.1 Interaction of firms, households, and bank with government over centralized graphs

All local firms, foreign firms, workers, investors, inactive households, and bank pay various kinds of taxes to the central government. The government pays social-benefits and subsidies to the households, and subsidies to some of the industrial sectors. Tax paying is unidirectional and can be easily parallelized by making all the ranks to collect the tax from all their agents and then send this collected tax to the rank containing government agent. Social-benefits payment by government is bidirectional as the amount of social benefit to be paid to a household depends on its economic status. For its parallel implementation, the government can first collect the economic status of households using *MPI_Gatherv*(), and then sequentially calculate the amount of social benefits payable to each households, and finally communicate this amount to the households using *MPI_Scaterv*(). However, this makes the rank containing the government agent do all the computation regarding social benefits to be paid which introduces load imbalance and is not good for scalable implementation.

2.1.2 Interaction of firms, and households with bank

All firms and households interact with bank to deposit their surplus money, withdraw their deposited money, or take loans depending on their financial conditions. Here, depositing is unidirectional whereas withdrawing and taking loan are bidirectional. Similar to the social benefit transfers by the government, withdrawing and taking loan require three steps: first, gathering the loan applications using *MPI_Gatherv*() and then, processing the applications one-by-one, and lastly scattering the reply to the applicants using *MPI_Scatter*(). This gives high computational load to the rank containing bank. It is also important to note here that all the agents may not apply for loans at the same point of time, and even all agents may not be applying for loans in each period. Therefore, the interaction graph between loan applicants and the bank is dynamic in nature and keeps evolving with time which makes its parallel implementation more challenging.

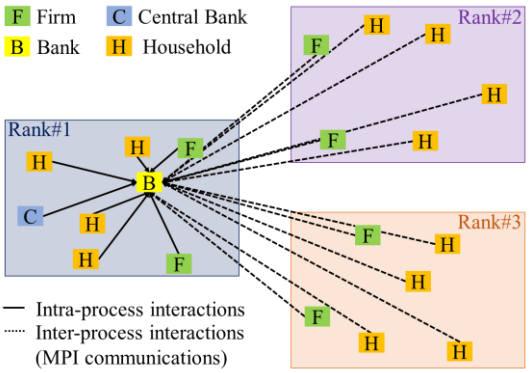


Fig. 2 – Interactions of customers with bank on centralized graphs

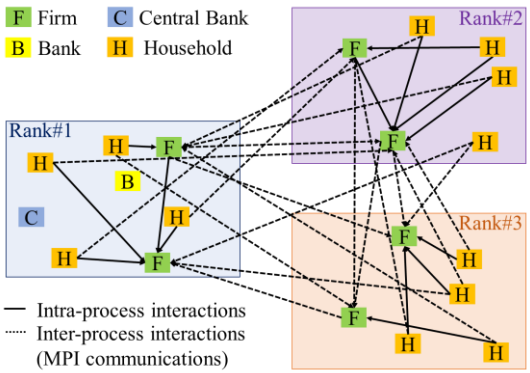


Fig. 3 – Buyer-seller interactions on scale free graphs

2.2 Interactions over scale free graphs

Interactions in goods market as well as labor market take place over scale free graphs. All the interactions in these graphs are bidirectional, random, and dynamic in nature. A typical buyer-seller interaction graph is presented in Fig. 3. Because of the randomness, it is impossible to predict all the interactions and plan their parallelization. Moreover, an implementation for one simulation period may not work in the next period because of the dynamic nature of the graphs. All these properties make their parallel implementation a very challenging task.

2.2.1 Interactions of consumers with sellers in the goods market

Each period, all consumers (i.e., firms, foreign firms, households, and government agencies) visit a random set of sellers (i.e., local firms and foreign firms) of each industry to buy capital goods and consumption goods. All the sellers of the economy are accessible to all the buyers. The probability of a seller being chosen by a buyer is based on its size (i.e., quantity of goods produced) and the price of its products. Sellers with large produce and lower prices have a higher probability of being visited by the buyers. To give a fair chance

to all buyers, the order in which they go shopping is randomized each period. Each visit of a buyer to a seller present in another rank needs two MPI communications between their home ranks because of the bidirectional nature of the interaction. The number of sellers visited by a buyer is unpredictable and depends on its budget and the amount available for sale with the seller at the time of its visit. Therefore, there will be many random, one-to-one, and bidirectional communications between ranks as shown in Fig. 3. Implementing a scalable implementation for such a scenario is extremely challenging.

A simple solution may be the use of request queues by each seller where buyers can submit their buy requests and the seller can process each request sequentially. However, this will make a lot of buyers wait for the reply of their current requests which will slow down the code. The ranks containing sellers with higher selling probability will have a larger number of requests which will introduce load imbalance. The overall performance may be even worse than the serial code.

2.2.2 Interactions of employees with employers in the labor market

Each period, firms hire workers according to their labor demand or fire extra workers; unemployed workers look for jobs until they get employed. The rank with more workers than available jobs will have many unemployed workers who will look for jobs in labor deficient ranks. Whether a rank is labor deficient, or labor surplus is known only at the beginning of labor market in each period. A job application by a worker to a firm in another rank involves two point-to-point communications: one for the application by the worker and the other for firm's reply to the worker. If the worker can secure such a job, the firm needs to pay his wages also which involves one more communication from the firm to the worker. Since a worker may get fired and the fired worker may get job at another firm, all the interactions in the labor market are dynamic. Like the goods market, the number of firms visited by a worker in the search of job is unpredictable and depends on the number of vacancies at the firm. Obviously, implementing a scalable solution for such a scenario is very challenging.

3. Implementation of a Scalable Parallel Computing Extension

This section presents the proposed parallel computing scheme in detail: the partitioning of agents to assign balanced workload to all MPI ranks; various techniques adopted to counter the challenges discussed in the previous section.

3.1 Partitioning

Balanced distribution of workload among the processes is the first and most important ingredient for parallel scalability. As discussed in the previous section, the agents interact over various topological graphs in which some are centralized graphs, and some are scale-free graphs; and most of these graphs are dynamic. We choose a representative employer-employee interaction graph for partitioning the agents and adopt various techniques to ensure complete availability of remaining interaction graphs.

We construct the representative employer-employee interaction graph by joining all the firms with n nearest number of workers, where n is the initial labor demand of the firms. Out of the remaining agents, inactive households, and foreign firms are included in the graph by connecting them with nearest one or two firms, whereas investors, banks, government, and central bank are not included in the graph but assigned to the partitions later. To ensure that the graph sufficiently reflects the employer-employee interactions, the firms-inactive households and firms-foreign firms links are assigned a low link weight. Further, the firms are also linked with neighboring firms, with low link weights, to reduce the possibility of generating a disconnected graph. The nodes of the graph are assigned a nodal weight according to the computational workload associated with the represented agent. The generated graph must be partitioned in such a way that all the partitions have almost equal sum of nodal weights and the partition boundaries cut the minimum number of edges. All the agents are spatially distributed according to their physical location of the simulated region. This enables us to efficiently access large volume of input data related to the disaster, like the damages of each building, road, and lifeline etc.

After partitioning the graph with METIS[9], investors are assigned to partitions according to the number of firms in the partitions. Bank, government and central bank are assigned to the master rank (i.e. rank 0) so that they can interact among themselves without any communications.

3.2 Scalable solutions for the centralized graphs

The centralized graph can be decomposed into local centralized graphs for each rank by introducing representative agents for the bank and the government in each rank. The representative agents will communicate with their parent agents (i.e. the bank and the government), requiring only one communication per rank. Thus, a very large number of communications as well as load imbalance due to the centralized graphs can be eliminated.

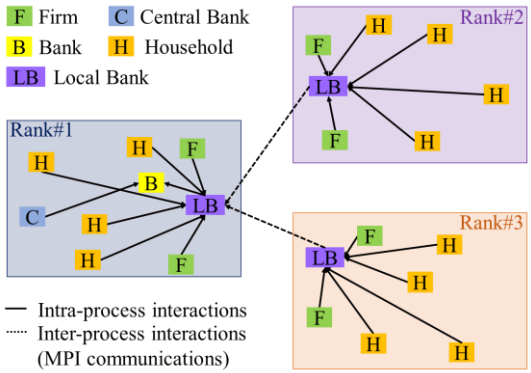


Fig. 4 – Introduction of local banks

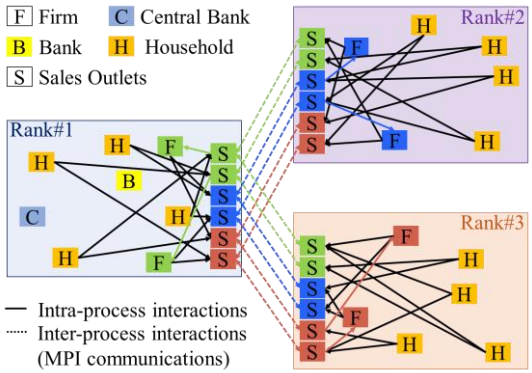


Fig. 5 – Introduction of sales outlets

3.2.1 Interactions with the government on centralized graphs

The simple solution discussed in section 2.1.1 has poor scalability because the rank having the government agent will have to calculate the amount of social benefits to be paid to each household and then communicate the calculated amounts back to the households. A scalable solution is to introduce a new agent –local government– in each rank which will serve the purpose of the government agent. This agent collects tax from all other agents, calculates and pays social benefits to households, and buys a portion of government budget. At the end of the financial period, the local government agent communicates the amount of tax collected, social benefits paid, and the amount bought to its parent agent –the government– present in the master rank using just one communication. As it is evident, this is a scalable solution as it can evenly distribute the computational workload, eliminate all bi-directional communications, and drastically reduce the overall number of communications.

3.2.2 Interactions with the bank on centralized graphs

Bi-directional interactions with the bank are a big challenge for a scalable implementation. Simple solution discussed in section 2.1.2 is not scalable because of the associated load-imbalance and a large number of communications. However, the problem can be solved by introducing a new agent –local bank– a representative of the bank, in each rank which can locally process all the deposit and withdrawal requests from the customers. Also, this ensures a balanced load distribution as all ranks are processing deposit and withdrawal requests. At the end of the financial period, the main bank present in the master rank collects the financial reports i.e. total deposits, loans, interests received etc. of all the local banks using an *MPI_Igather*() communication. The proposed solution evenly distributes the computational workload and eliminates a large number of communications that make it scalable. The scheme is illustrated in Fig

. 4.

3.3 Scalable solution for scale free graphs

As discussed in section 2.2, implementing a scalable solution for scale free graphs seems impossible because of their random, dense and dynamic nature. However, in real world goods market, a firm has many sales

outlets spread over the country for selling its products. Therefore, a firm is able to sell its products everywhere irrespective of its location. Similarly, there are recruitment agencies which hire labor for firms from various regions. Our scalable solution for scale free graphs is to closely mimic the real-world functioning of goods market and labor market, and thereby introduce new agents: sales outlets and recruitment agencies.

3.3.1 Interactions on goods market

As shown in Fig. 5, new agents: sales outlets, are introduced for each firm (i.e. seller) in each rank. Thus, each rank has sales outlets for all the firms present in the economy. The consumers visit these local sales outlets to buy various capital goods and consumption goods. Obviously, these visits don't need any communications, therefore, a very large number of random communications are eliminated. However, the sales outlets still need to communicate with their owner firms to get the quantity to sell and finally sending back the sales record. The firms have to make an important decision on how much quantity be sent to their sales outlets in various ranks. The firms make this decision based on the estimated demands of the products of their industry in the ranks. The firms assign the quantity of goods to be sold to a sales outlet based on the ratio of demand from its rank and total global demand. To implement this decision, first, each rank gathers the sectoral demands, of all the ranks using *MPI_Igather*(), and then the firms scatter their production to their sales outlets using *MPI_Iscatter*(). After receiving the quantity to sell, sales outlets sell their products to local consumers. At the end of buying-selling, these sales outlets report their sales record to their owners. Reporting is done in two steps; first, the master rank collects the sales record of all the sales outlets using *MPI_Ireduce*() and then it scatters the reduced results to the ranks having owner firms using *MPI_Iscatter*().

3.3.2 Distribution of Big Buyers

Because of the heterogeneity of the agents, some agents are economically quite bigger in size as compared to others. For example, in an industry, some firms have only one labor whereas there are firms with thousands of workers in the same industry, creating a huge difference between the total production, requirement of capital and intermediate goods for the production between smaller firms and bigger firms. Similarly, there is income disparity among households as some of them are investors of big firms whereas for some, government subsidies are the only source of income. Since the consumption budget of the households depends on their income, there is huge difference between the consumption budget of households. The total time taken by an agent to buy is directly proportional to its consumption budget.

Since the agents have an associated physical location and are situated in a process depending on their location, some processes may have larger number of big buyers compared to others which is causing a huge load imbalance. In order to balance the workload, the big buyers: firms with more than 100 labors, and the investors: investors of big firms (i.e. firms with more than 100 labor), and bank investors; are made to buy equally from all the processes. Consequently, total demand gets distributed equally among all the processes and the load-imbalance reduces drastically. Minimum number of labors to designate a buyer as big buyer is dependent on the economy. For the economy simulations presented in this paper, using 100 as the minimum limit provides the best computational performance.

3.3.3 Interactions on labor market

Even though the agents were partitioned based on a representative interaction graph of labor market by minimizing the number of inter-process links, a large number of inter-process links are still present because the workers cannot be distributed exactly according to labor demand of firms since they have a fixed physical location. Moreover, the number of such links may increase as the interaction graph evolves with time. In order to get rid of these large number of random and inter-process interactions between workers and firms, an intermediate agent –recruitment agency– is introduced in each rank which acts as labor supplier for the firms and job provider for the job seekers. All the firms send their labor demand in each period to the recruitment agency present in their rank, and the unemployed workers visit recruitment agency to find jobs. Depending on the availability of jobs and unemployed workers, a recruitment agency may have surplus jobs or surplus labor. After finishing the local job market of their rank, the recruitment agency in master rank first

collects the number of surplus jobs/labor of all recruitment agencies using *MPI_Igather*(), and then decides on which rank should send jobs/labor to which so that all the recruitment agencies can satisfy their labor demands. This decision is scattered back to the recruitment agencies using *MPI_Iscatter*(). Depending on the instructions received from the master recruitment agency, the recruitment agencies communicate among themselves using point-to-point communications. Thus, the introduction of recruitment agency has eliminated the random interactions on labor market; moreover, their functioning is not affected by the dynamic nature of the graph. All these factors make this a scalable solution.

3.4 Communication hiding

All the communications required for the proposed implementation are overlapped with computation in order to utilize the communication time and attain higher scalability. A non-blocking message is posted as soon as the data to be sent become available and the message is finalized just before the requirement of that data by the receiver rank.

4. Computational Performance

The ABEM developed Poledna et. al [7] is adopted for this research and has been implemented using C++ and MPI-3. The code has been developed based on various serial and parallel performance enhancing techniques presented in the previous section. This section examines the effectiveness of those techniques in improving the performance of the code.

4.1 Problems Setting

To illustrate the performance improvements, two simulations are presented: one for a small economy having a total of 9.8 million agents (herein referred to as Small-Economy), and the other for a large economy having a total of 331 million agents (herein referred to as Large-Economy). Small-Economy and Large-Economy represent the Austrian economy and the Euro Zone economy respectively.

4.2 Load Imbalance

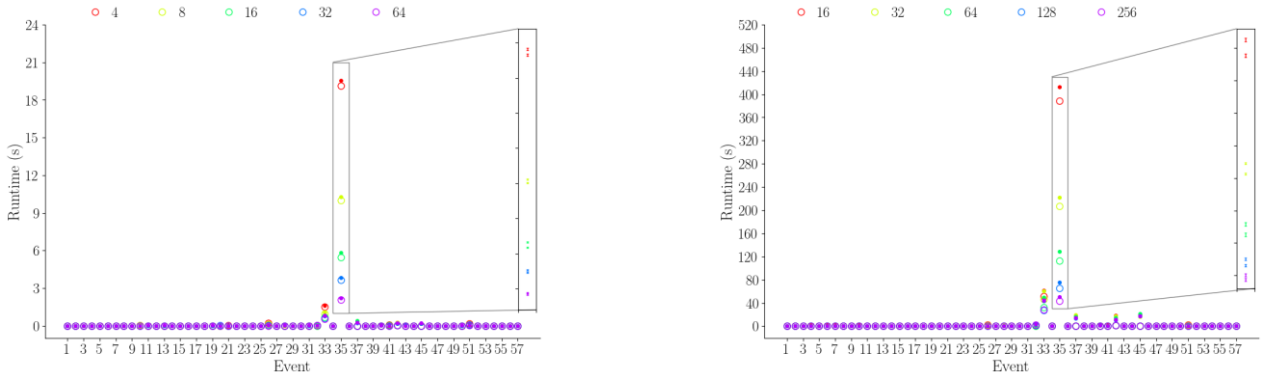
This section studies the performance of the whole code against load imbalance. The simulation results of Small-Economy and Large-Economy are presented in Fig 6. Performance of various techniques proposed in section 3.3 is discussed below one by one.

The interactions among local governments is represented by three events: event 2 denotes the initialization of *MPI_Ibcast*() to broadcast the government budget to all local governments by the main government, event 16 finalizes this communication, and event 51 which is year-end financial accounting. The local governments post *MPI_Ireduce*() to reduce financial data at the main government and finalize it inside event 51. The local governments interact with other agents in event 41 as well as inside 51. The events involving local governments take a very small amount of time and introduce no load imbalance as is evident from Fig. 6.

Events 39, 49, 51, and 57 represent the interactions among local banks. Event 39 posts *MPI_Ireduce*() to collect the budget bought for the bank's investor (distributed buyer) and event 49 finalizes this communication. The collection of local banks business data is done inside event 51 by using *MPI_Ireduce*(). Interactions of local banks with other agents is represented by events 12 and 44. None of these events introduce any load imbalance in both simulations.

The implementation of goods market involves the largest number of events. Event 33 and 35 represent firms' *buy*() and other consumers' *buy*() respectively. Before the start of *buy*(), budget to buy is sent to non-local buyers by parent buyers and the quantity to sell is sent to the sales outlets by the sellers. After the *buy*(), bought quantities by non-local buyers and the sold quantities by the sales outlets are to be collected. Non-local buyers are segregated into non-local firm buyers and non-local household buyers. The interactions among non-local buyers and sales outlets follow the same implementation scheme. Event 11, 20, and 27 post *MPI_Ialltoallw*() to initialize the budget distribution of firm buyers, budget distribution of household buyers,

and quantity distribution of sales outlets respectively. Event number 22, 32, and 28 finalize these communications. Collection of quantities bought by non-local buyers and quantities sold by sales outlets involves two steps as discussed in section 3.3.1. Events 34, 38, and 40 post the *MPI_Ireduce*() for



(a) Small-Economy simulation (b) Large-Economy simulation

Fig. 6 – Mean runtimes (of 13 iterations) for each event. The difference between the center of dot and circle corresponding to an event denotes the load imbalance. The zoomed values show the standard deviations.

collecting the non-local firm buyers bought quantities, non-local household buyers bought quantities and sales outlets sold quantities respectively. Event 37, 47, and 42 finalize the *MPI_Ireduce*() and post *MPI_Iscatter*(). In the last step, events 48, 50, and 45 finalize the *MPI_Iscatter*(). Fig. 6 indicates that there is load imbalance in events 33, 35, 37, 42, and 45. Even after distributing the big buyers, events 33 and 35 still has load imbalance. The reason for this imbalance is the random selection of sellers by a buyer. Computation resources required by a visit to a seller by a buyer are independent of the seller's size. If a big buyer selects big sellers, it may satisfy its demand in a few visits. On the other hand, if it selects small sellers, a large number of visits are required, and the corresponding computation time is longer. Similarly, if a seller gets visited by big buyers, it will sell it products in a short time. The load imbalance of events 33, and 35 is carried over to 37, 42, and 45. However, it should be noted that all the communications communicating a large amount of data take a significantly small amount of time and don't introduce any kind of load imbalance.

Lastly, the events representing the labor market are discussed. Labor market starts with the event 4 which determines the fired workers and initializes *MPI_Isend*(), *MPI_Irecv*() between two labor exchanging recruitment agencies to intimate the fired workers. Event 6 finalizes these communications and completes local hiring. In event 7, the main rank agency collects excess labor/jobs from all the other agencies using *MPI_Gather*() and allocates labor from labor rich agencies to labor deficient agencies and send this decision back using *MPI_Scatter*(). Event 9 posts *MPI_Isend*() and *MPI_Irecv*() between two agencies for sending and receiving jobs as determined by the main agency, and event 13 finalizes these communications. Event 15 again posts *MPI_Isend*() and *MPI_Irecv*() for intimating the job types to the labor providing agencies. These communications are finalized in event 23. The final communications between the agencies are done to send and receive the wages of the workers exchanged. Event 24 and 25 represents the initialization and finalization of these communications. Clearly, the events related to labor market don't introduce any load imbalance in both the cases.

4.3 Scalability

Table 1 and Table 2 show the runtime and strong scalability of the code for Small-Economy simulation and the Large-Economy simulation respectively. Runtime is the average runtime taken by 13 iterations of the code in each case and strong scalability is defined as $(T_n/T_m)/(m/n)$; T_n is the average runtime taken by n number of processes, and $m \geq 2n$. The implementation has a high strong scalability for both the cases.

Table 1 – Runtime and Strong Scalability for Small-Economy

Number of MPI processes	Runtime(s)	Strong Scalability(%)
4	21.98	
8	11.85	92.74
16	7.09	83.57
32	5.02	70.62
64	3.41	73.61

Table 2 – Runtime and Strong Scalability for Large-Economy

Number of MPI processes	Runtime(s)	Strong Scalability(%)
16	492.00	
32	294.13	83.64
64	187.04	78.63
128	129.00	72.50
256	107.32	60.10

5. Numerical Experiment

To demonstrate the application of the developed tool, simulation results of Japanese economy under three scenarios are presented. The first scenario represents normal economic conditions (herein referred to as Normal Scenario) and the other two corresponds to post-disaster economic conditions resulting from two hypothetical disaster scenarios: disaster causing 1% loss of capital (herein referred to as Disaster Scenario-1) and disaster causing 10% loss of capital (herein referred to as Disaster Scenario-2). The disasters are assumed to occur at the end of the 1st period of simulation. Each period represents a quarter of a year.

5.1 Model parameters and agents' behavior rules

The model parameters are derived from the Japanese economic data of quarter 4, 2012. Data sources are input-output table[1], national accounts[2], and OECD[3] quarterly national accounts. The economy has about 120 million households and about 1.5 million local firms spread over 51 industrial sectors. The economy interacts with the rest of the world through foreign firms which are assumed to be about 0.75 million in number. One simulation period of the model represents one financial quarter and the economic simulations were conducted for 12 quarters.

The behavior rules of agents under normal economic conditions are provided in Poldena et. al [7]. For the disaster scenario, the agents' behavior is assumed as below:

a) Following the System of National Accounts procedures, the firms subtract their damaged capital from their profits.

b) The firms try to buy the lost capital starting from the next period after the disaster. However, the availability of loans from the bank and the availability of capital goods in the goods market determine the amount of time it takes for them to buy the lost capital.

c) The households too start buying their lost capital from the next period after the disaster. We assumed that the households get full damage coverage in the form of government aid to buy their lost capital since this is a standard practice in many developed countries. The availability of capital goods in the market is the only constraint for households.

5.2 Results and discussion

The simulation results of all three cases are plotted together and presented in Fig. 7. Among various economic parameters nominal capital formation, operating surplus of firms, wages paid to workers, and

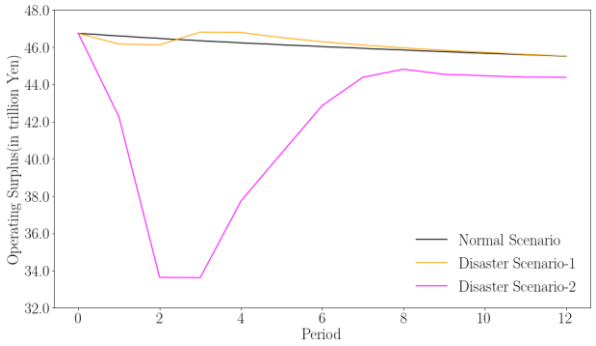
[1] https://www.meti.go.jp/english/statistics/tyo/entyoio/index.html

[2] https://www.esri.cao.go.jp/en/sna/data/kakuhou/files/2017/2017annual_report_e.html

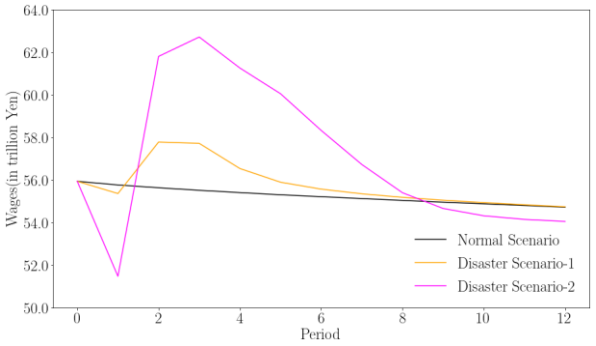
[3] https://stats.oecd.org/Index.aspx?DataSetCode=QNA_ARCHIVE#

nominal gross value added are illustrated. The results show that in the short-term, both disaster scenarios cause negative impact on the economy. The economy recovers completely and approaches the normal levels in case of Disaster Scernaio-1 but under Disaster Scenario-2, it settles after a loss of approximately 1.5% (in gross value added).

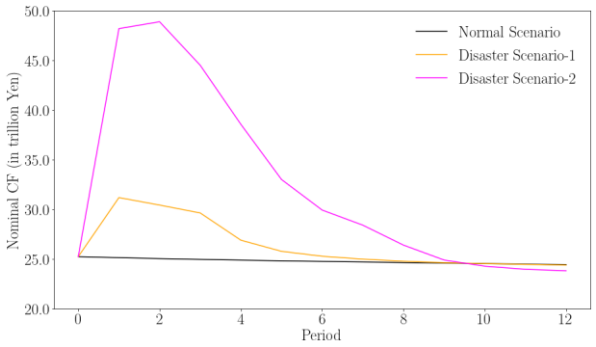
Immediately after the disaster, the operating surplus of firms plunges because of the loss of capital. The capital loss decreases the firms output which forces firms to fire workers. This is depicted as decrease in total wages paid to workers in Fig. 7(b). In later periods, the firms and households try to buy their lost capital which causes a short-term demand in the economy. So, the firms hire more workers to produce more to satisfy the demands. Consequently, all the economic indices show positive signs in the medium-term. The economy can even attain a higher level than the normal scenario. However, the growth is short-lived, once the lost capital of all the agents is recovered, the economy tries to stabilize and reaches an equilibrium. As the Fig. 7 shows, the quarterly gross value added in the economy stabilizes at the normal scenario levels, in case of Disaster Scenario-1 whereas it sets at a state which is 1.5% lower than normal scenario state, in case of Disaster Scenario-2.



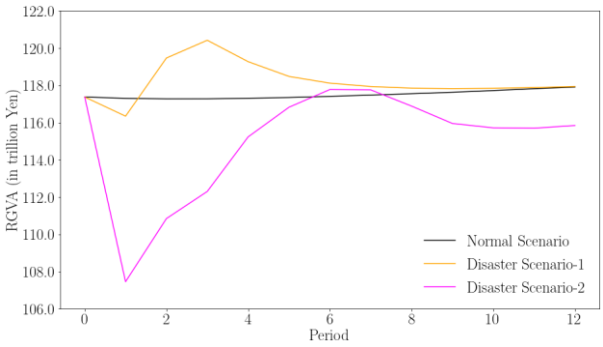
(a) Operating Surplus of firms



(b) Wages paid to workers



(c) Nominal capital formation



(d) Real Gross Value Added in economy

Fig. 7 – Comparison of various economic indices under three economic conditions

6. Conclusion

Large disasters occurring in an industrial region of a nation may take a toll on the national economy. A fine understanding of the behavior of economy in the aftermath of disasters can play a major role in improving the resilience to disasters. This paper is an attempt to develop a fine-grained ABEM that can simulate all the complex economic interactions of hundreds of millions of heterogeneous economic entities of a national economy, with the aim of applications in disaster management. The ABEM has been computationally enhanced with a distributed memory parallel implementation to make it possible to simulate 1:1 scale model

of a large country or even an economic zone. The agents were partitioned based on a representative employer-employee interactions graph. Real life solutions like introduction of sales outlets, recruitment agencies, local banks, and local governments were mimicked to ensure the availability of rest of interaction graphs with a small number of communications. Non-blocking communications were used to hide the communications and utilize the communication time. It is shown that the implementation has a sufficiently high strong scalability and short runtime for simulating an economic zone like Europe with modest computational resources. An illustrative application of the model in simulating post-disaster economy is also presented. The model shows that a large disaster destroying 10% capital of Japanese economy causes a permanent loss of approximately 1.5% to the quarterly gross value added in the economy. The simulation results can be further improved by using more realistic post-disaster behavior rules of the economic entities. The developed system can be a potential candidate for quantitatively assessing the economic performance of recovery plans, or even as a component of a system to find optimal recovery plans to achieve quick economic recovery.

6. References

[1] Assenza T, Gatti DD, Grazzini J (2015): Emergent dynamics of a macroeconomic agent-based model with capital and credit. *Journal of Economic Dynamics and Control*, 50, 5-28

[2] Gatti DD, Gallegati M, Greenwald B, Russo A Stiglitz J (2010): The financial accelerator in an evolving credit network. *Journal of Economics Dynamics and Control* 34, 1627-1650

[3] Caiani A, Godin A, Caverzasi E, Gallegati M, Kinsella S, Stiglitz JE (2016): Agent based-stock flow consistent macroeconomics: Towards a benchmark model. *Journal of Economics Dynamics and Control*, 69, 375-408.

[4] Haldane, AG, Turrell AE (2018): An interdisciplinary model for macroeconomics. *Oxford Review of Economic Policy*, 34(1-2) 219-251.

[5] Holcombe M, Chin S, Cincotti S, Teglio A, Deissenberg C, van der Hoog S, David H, Gemkow S, Harting P, Neugart M (2013): Large-scale Modelling of Economics Systems. *Complex Systems*, 22(2):175-191

[6] Deissenberg C, van der Hoog S, Dawid H (2008): EURACE: A massively parallel agent-based model of the European economy. *Applied Mathematics and Computation*, 204 541-552

[7] Poledna S, Hochrainer-Stigler S, Miess MG, Klimek P, Schmelzer S, Sorger J, Shchekinova E, Rovenskaya E, Linnerooth-Bayer J, Dieckmann U, Thurner S (2018): When does a disaster become a systematic event? Estimating indirect economic losses from natural disasters. arXiv:1801.09740

[8] Caiani A, Russo A, Palestrini A, Gallegati M (2016): Economics with Heterogeneous Interacting Agents. Springer. Print

[9] Karypis G, Kumar V (2019): A fast and highly quality multilevel scheme for partitioning irregular graphs. *Journal on Scientific Computing*, 20(1), 359–392

[11] Lalith M, Gill A, Poledna S, Hori M, Hikaru I, Tomoyuki N, Koyo T, Ichimura T (2019): Distributed Memory Parallel Implementation of Agent-Based Economic Models. *Lecture Notes in Computer Science*, pp. 419-433, Portugal: Springer. DOI:10.1007/978-3-030-22741-8_30.

[12] Poledna S, Miess Michael G, Hommes Cars H (2019): Economic Forecasting with an Agent-Based Model. http://dx.doi.org/10.2139/ssrn.3484768

