Future Generation Computer Systems 47 (2015) 48–59

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

An optimization-based adaptive resource management framework for economic Grids: A switching mechanism

Aminul Haque [a,*], Saadat M. Alhashmi [b], Rajendran Parthiban [c]

[a] Department of Computer Science & Engineering, Daffodil International University, Dhaka, Bangladesh

[b] College of Engineering and Computer Science, Abu Dhabi University, United Arab Emirates

[c] School of Engineering, Monash University Malaysia, Jalan Lagoon Selatan, Bandar Sunway, 46150, Selangor Darul Ehsan, Malaysia

HIGHLIGHTS

- Economic-based approaches are relevant for Grid resource management.
- Different economic models are found suitable for different scenarios in Grid computing.
- A quantitative analysis has been carried out to identify the domains of strengths of major economic models in Grid resource management.
- The opportunities and challenges of developing an optimization framework by utilizing the potentials of different models in different scenarios have been discussed.
- The development of the optimization framework in the context of dynamic and distributed computing environment has been elaborated.

ARTICLE INFO

Article history:
Received 30 June 2012
Received in revised form
9 October 2014
Accepted 23 October 2014
Available online 4 November 2014

Keywords:
Grid computing
Economic models
Domains of strengths
Switching
Optimization

ABSTRACT

The application of Grid computing has been broadening day by day. An increasing number of users has led to the requirement of a job scheduling process, which can benefit them through optimizing their utility functions. On the other hand, resource providers are exploring strategies suitable for economically efficient resource allocation so that they can maximize their profit through satisfying more users. In such a scenario, economic-based resource management strategies (economic models) have been found to be compelling to satisfy both communities. However, existing research has identified that different economic models are suitable for different scenarios in Grid computing. The Grid application and resource models are typically very dynamic, making it challenging for a particular model for delivering stable performance all the time. In this work, our focus is to develop an adaptive resource management architecture capable of dealing with multiple models based on the models' domains of strengths (DOS). Our preliminary results show promising outcomes if we consider multiple models rather than relying on a single model throughout the life cycle of a Grid.

© 2014 Elsevier B.V. All rights reserved.

1. Background and introduction

The journey of Grid computing was officially introduced in mid 1990s with the vision of collaborating distributed computational resources to meet high computational demand posed by scientific communities [1]. While many researchers started developing suitable tools for application development and scheduling for distributed resources; parts of researchers suggest anatomical features [2,3,1]. Methods appeared included decentralized, hierarchical, decentralized-hierarchical, and economical strategies [3]. Soon, it was realized that economic-based resource management techniques could provide significant benefit to Grid resource management [4,5]. The potential of economic-based distributed resource management has not been identified recently, rather the realization was made much earlier than the Grid [6,7]. In Grid computing, the potential was mainly emphasized in 2002 through an intuitive demonstration about the need of economic-based resource management models [8]. The demonstration incorporates detail on the scope and motivation for Grid resource providers in building a worldwide Grid computing platform. This demonstration about different economic models was, however, only based

* Corresponding author. Tel.: +880 1712285212.
E-mail addresses: aminul.cse@daffodilvarsity.edu.bd (A. Haque), saadat.alhashmi@adu.ac.ae (S.M. Alhashmi), rajendran.parthiban@monash.edu (R. Parthiban).

http://dx.doi.org/10.1016/j.future.2014.10.022
0167-739X/© 2014 Elsevier B.V. All rights reserved.

on their hypothetical suitability for the Grid, not based on any experimental evidence.

Since the introduction of various economic models in the Grid, extensive research has been conducted to understand the effectiveness of the models for distributed resource collaboration [9,10]. However, the uniqueness in working principles and pricing methodologies of individual models seem unable to deal with the dynamic and distributed nature of the Grid. Our survey on existing economic models has investigated the similar fact that different models are suitable for different scenarios in Grid computing [11]. For example, English Auction Model is found to be suitable for maximizing revenue for providers; however, the model generates huge communication cost. Commodity Market Model is found suitable for maintaining equilibrium between supply and demand for resources. Therefore, deciding a particular economic model by Grid resource providers is a challenging task due to the following reasons.

- The performance stability of a particular model in a dynamic and distributed environment is hard to establish.
- A particular model may not be suitable to optimize a particular objective function all the time.

The potential of different models in different scenarios has inspired us to develop a research hypothesis—"*The diversity of different economic models can add value in solving distributed resource management problem in Grid Computing*". How one would be able to employ this diversity in a dynamic Grid environment to obtain the expected benefit (optimization) is, however, a key issue. In a previous work, the authors have already identified which models would be suitable for which scenarios [12]. The focus of this work is to develop an optimization framework based on this identification of the domains of strengths of the models.

In our work, we consider only five most widely proposed economic models in the Grid—Commodity Market, Bargaining, English Auction, Continuous Double Auction and Contract-Net-Protocol [11]. We would like to explore the possibility of using models such as Brownian motion from financial markets in future [13]. The paper is structured as follows:

Section 2 presents a discussion of economic-based resource management models. The user and resource models are illustrated in Section 3 following with the motivation for this work in Section 4. Section 5 presents a prototype design for the optimization framework. Section 6 incorporates the development of the optimization model. Section 7 presents the experimental evaluation of the model. Section 8 presents some future works following with conclusions in Section 9.

2. Related work

This work is an extension of previous works [11,12]. In this section, we briefly describe our previous findings and present some research gaps and opportunities in economic-based Grid resource management.

The Commodity Market Model (CMM) has the ability to maintain equilibrium between resource supply and demand, which is crucial for any market-oriented Grid environment. Maintaining supply and demand by regulating pricing behavior ensures a higher probability to deliver requested QoS (Quality of Service) to users as well as to increase system performance. This implies that, in CMM, the probability of optimal resource allocation increases, due to supply and demand equilibrium. This optimal allocation helps increase the probability of satisfying the market-wide users with their requested QoS. The main principle behind this model is to determine an equilibrium/spot price at which the market can diminish its aggregated supply and demand. For example, if demand for a resource exceeds its supply at a particular state, the price of that resource increases in such a way so that the demand function shifts to a point closer to the available supply. Various techniques have been used to determine the equilibrium price in the literature [14]. Wolski et al. identify the suitability of CMM for maintaining market equilibrium and minimizing communication cost compared to that of the English Auction [15]. On the other hand, Tan and Gurd criticize CMM due to its system-oriented approach rather than being incentive-oriented [16]. They argue that in CMM, price formation process considering global information on supply and demand does not account for individual's preference value; thus becomes undesirable by the participants.

Using Bargaining Model (BAR), users and providers can optimize their individual preference functions (time/cost). The model allows participants to negotiate on their preferences and finally to construct a satisfactory SLA (Service Level Agreement) [17,18]. In Grid computing, the preferences could be for budget/job-execution-cost, deadline/job-execution-time or any such criteria. However, successful negotiation also depends on preference functions imposed by the participants. For example, if a user and a provider start negotiating on the same preference function (e.g., deadline and job-execution-time), the negotiation will be ended up with either minimum optimization or it will be failed. The model requires a high communication demand due to its multi-round negotiation process, which may not be suitable in terms of dealing with a large number of participants in the environment.

Auction protocols describe the behavior of the participants in the auctions and analyze the properties of the auction markets. It also studies competitive bidding strategies and market throughput such as revenue. English Auction (ascending-bid) is one of the interesting auction models in the Grid [19]. In this model, an auctioneer seeks to obtain the true market value of resources through auction. Usually, users are free to increase their bids exceeding others for the resource they are competing. When no bidder is willing to increase its bids anymore, the auction ends, and the auctioneer checks its reservation price[1] with the last highest bid and determines the winner. This model is found to be suitable for increasing revenue for providers; because it lets the users compete for resources and finally selects the user who bids the highest through an iterative bidding policy. However, the model is likely to be profitable when there is a lot of demand for resources. In addition, English Auction, by nature, is an iterative model and therefore creates a huge amount of communication overhead during the auction process [20].

Continuous Double Auction, on the other hand, is a suitable model for the Grid due to its decentralized working mode and the ability to handle a large number of users [20,16]. In Grid computing, users and providers are typically regarded as self-interested entities. The model supports sorting the valuations of the entities and thus accelerates the trading phase while maintaining a good level of optimal resource allocation. The sorting process is typically conducted over the budgets of users and costs of resources whereas; the users are sorted in descending order in terms of their budgets and resources are in ascending order in terms of their costs. The model is proposed to be suitable for retaining market equilibrium and price stability compared to that of the English Auction.

Contract-Net-Protocol is popular in Grid computing, especially because of its ability to support for meta-scheduling and resource cooperation, which can deliver a strong QoS to the users. A broker and a resource-node are known as manager and contractor respectively in this scenario [8]. In such a market scenario, a manager tries to optimize its scheduling process (meta-scheduling) by selecting

[1] The minimum price a user must pay to get access by a resource. This is basically the job execution cost computed by a particular node.

one or more suitable contractors from available contractors in the market. The selection process is typically conducted according to the manager's preference values. In terms of task execution in a distributed environment such as Grid, the manager could prefer either time or budget for optimization. CNP has been found to be suitable for utility-based resource allocation and scalability [21]. It is also suitable for solving distributed cooperation problem and for supporting meta-scheduling process. The model is specially designed to understand user requirements and schedule resources accordingly. Providers obtain limited chance to optimize their utility functions using this model.

Grid computing shares resources across geographical boundaries and possesses users from around the world. Expecting constant performance all the time by a particular model in such a dynamic and large-scale computing environment is, therefore, subjected. An extensive economic-based research has been conducted to deal with this issue from the Grid's inception until now. Various optimization techniques have been applied to deliver a sustainable computing platform. However, a few researches have focused on the strengths of the economic models and their ability to cope with the Grid. In this paper, we focus on an optimization mechanism that works at the elementary level of the models. Our focus is not limited to a particular model rather it seeks for the opportunity of utilizing the potential of different models in different scenarios. The following section briefly describes the fundamental properties of users and resources in the market scenario.

3. User and resource modeling

Grid entities are typically regulated depending on their own objectives. Economically inspired Grid entities are, however, more complex and more self-interested compared to non-economic entities. As such, simulating such individually rational entities requires comprehensive definition of their properties. These properties will then play a key role in defining their characteristics in the simulation. We use GridSim tool—a widely used Grid simulation toolkit for our development [22].

Grid *users* can be characterized in terms of their applications (such as protein folding) that need to be executed on Grid resources. A Grid application can again be composed of several Gridlets (also known as tasks). Based on the relationship and dependency among Gridlets, Grid applications can be categorized into three types; Bag of Tasks[2] [23], MPI (Message Passing Interface) and Workflow. Currently, our work is suitable for Bag of Tasks type applications, that is, Gridlets do not require to communicate with each other for execution. Again Grid applications can be computationally and/or data intensive. Our work, at present, supports computationally intensive applications.

Grid *resources* on the other hand, are typically referred to as resource-nodes or only nodes. The nodes can be standalone computational and/or storage systems. Multiple resources can then join to form a network of nodes to service the requirement of a scientific application. One node varies from another in terms of capability, performance and access constraint. Currently, our work does not support for reservation for resources. The models are based on queue-based systems.

A *broker* (also known as job-scheduler) performs all the crucial tasks on behalf of a user. A complete life cycle of a broker is presented in Fig. 1.

Each broker has unique ID. There are as many brokers as Gridlets, which implies that brokers are independent in performing their tasks. A broker meets a resource at random time. This helps

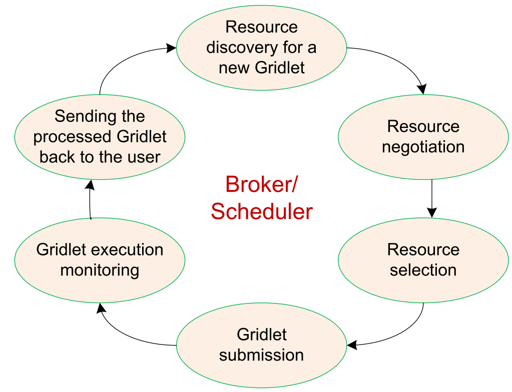


Fig. 1. A broker's life cycle on Gridlet Execution.

to generate a picture closer to the real world. At the first phase of its life-cycle, it collects the IDs of the available resources from GIS[3] (Grid Information Service). Afterwards, it starts communicating with resources and negotiating based on constraints defined by the user (Fig. 1). For example, it consults with a resource whether the resource can process the Gridlet within its available budget and deadline. However, the negotiation process varies over economic models. It is worth mentioning here that in our context, negotiation and bargaining are different terms. In negotiation, it is not necessary to have iteration; however, in bargaining it must have. For BAR, definitely there are multiple iterations and different requests are made at each iteration. However, for other models such as CMM and CNP, the communication between a broker and a resource is regarded as negotiation in general.

The broker may succeed in finding its suitable resource among the available resources or may fail. In terms of a failure, it sends the unprocessed Gridlet to its user. If the broker finds a suitable resource, it selects the resource to submit its Gridlet. A single resource cannot process multiple Gridlets, due to the GridSim's current limitation. Once the Gridlet has finished processing, the resource sends back the Gridlet to the broker along with the processing outcomes. The broker then sends the processed Gridlet back to its user. We have assumed that if the execution of a Gridlet is canceled or the Gridlet is rejected from a resource in any case, the broker will resend the Gridlet to the remaining resources. Therefore, a broker remains involved until it ensures that the execution of the Gridlet is finished. Nimrod is an example of real world Grid broker [24]. From our comparative analysis among the economic models, it is observed that different models are suitable for different scenarios [12]. We briefly present the result of Communication overhead here (Fig. 2), for motivation for the optimization model.

4. Motivation for an optimization framework

This section provides motivation for optimizing various performance metrics in a dynamic Grid computing environment based on the strengths of different models in different scenarios. In the supply–demand plot presented in Fig. 2, it is observed that CDA performs better in Region-1, CNP in Region-2 and CMM does better in Region-3. Now, let us consider the following example.

Example. Let us imagine a Grid network that would like to minimize its communication overhead. The communication service for

[2] This kind of applications consist of multiple independent tasks requiring no communication among the tasks.

[3] A service designed to assist Grid brokers/other entities by providing available resource information. Whenever a new resource entity is created, it must register with the GIS.

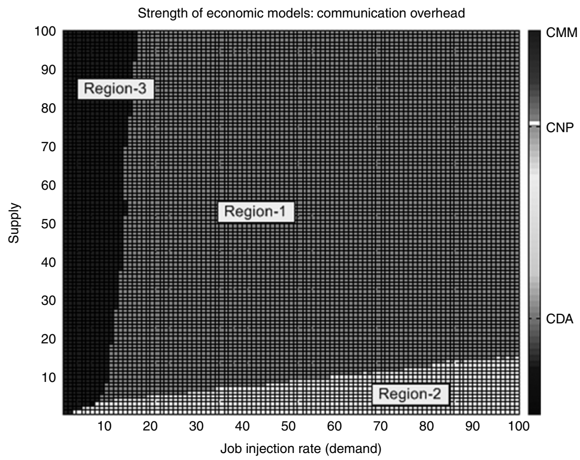


Fig. 2. Communication overhead comparison.

a network is not free in general. In addition, communication overhead has become a key topic by many researchers in distributed resource collaboration in the Grid [25,8,26]. Through a comparative analysis, we have identified that three different economic models perform better compared to each other in three different regions in terms of communication overhead performance metric (Fig. 3). Now, the possibility of minimizing communication overhead gets higher if the network employs the three different economic models in their respective domains of strengths.

For instance, if the network notices that its current demand is low regardless of its supply, it can employ CMM as CMM has been identified more strong in this case (Fig. 2). Likewise, if sometimes the network notices that its supply has been decreased moderately and demand has been increased, it can switch to CNP. As a result, the network would be able to utilize the potential of different models in different scenarios, thus optimizing the defined function in general. Similar optimization is also possible for other metrics where different models perform better compared to each other at different scenarios. Such optimization procedure would provide incentives to Grid providers, which in turn would help to construct a viable market mechanism. However, to be consistent with the nature of the Grid, such a multi-model architecture gives rise to the following questions;

1. What motivates a Grid to follow such an optimization procedure?
2. How one can utilize the potential of different models in a highly dynamic environment?
3. Who will keep track of the scenarios?
4. Who will decide which model to be used and when to be used?
5. Who is responsible for the corresponding consequences of changing models?

To deliver the answers of these questions, we propose a switching framework that dynamically switches from one economic model to another and is able to adapt with its consequences in the environment. To facilitate the switching process, we design a switching agent, which automatically decides which model to be used when and for what purpose. A prototype of the switching framework is presented in the following section.

5. Prototype design

To understand the working behavior of our switching framework, we design a prototype that describes the key functionalities performed by the framework. The key functionalities are based on the following two principles.

– The switching decision must be carried out and broadcast into the environment without any considerable delay.
– The network entities must be able to interpret the decision message and must know how to adapt with changing circumstances.

The first principle describes the problem of maximizing the utilization of the switching framework. The sooner the decision can be made and broadcast across the network, the better it can maximize the potential of the switching model. If the time taken to make the decision or the time from the point of decision broadcasting until the entities receive the message (latency) is higher, it might be hard to obtain maximum profit. The second principle states about the adaptive capabilities of the network entities, that is, the entities must be able to interpret and understand the decision message and alter their behavior accordingly.

A prototype of the switching framework is illustrated in Fig. 3. There are two parts—first, the Grid environment with adaptive management capabilities and second is the switching agent. The roles (broker, resource) in such an environment must have dynamic capabilities to deal with different models. Therefore, we need to design the parameters and the organization of a broker in a way so that it can adapt with changing behaviors in the environment. A resource model, on the other hand, also needs to have generic capabilities to deal with multiple models. The auctioneers, in this case, must be prepared so that they can start processing as soon as they are invoked by the system. In addition, every broker, resource and auctioneer should be able to interpret the language of the agent. Interpreting agent's message is crucial; because the agent informs whether to continue with the current model or to prepare a new model.

The most crucial role played in the environment is by the switching agent. There are six different stages in the agent's life

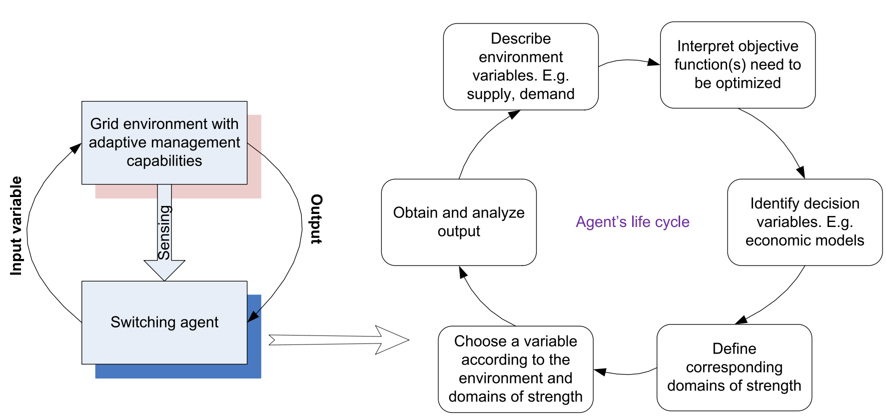


Fig. 3. Agent-driven switching framework for Grid resource management.

cycle (right part of Fig. 3). The agent can sense (keep track of supply and demand in the environment) the Grid in two different ways—time-based and event-based. In a time-based mode, agent senses the environment at every specific period whereas in an event-based approach, the agent senses the environment whenever there is a change in supply and/or demand in the environment. We have considered the event-based approach in our current work, and we would like to explore the time-based scenario in the future. Upon receiving an event, the agent starts its first activity, which is, analyzing the output (Fig. 3). The logic behind analyzing the output is to ensure that whether the model is making any optimization.

In an event-based system, whenever there is a change in supply or demand, the agent is invoked to provide decision on whether to continue with existing model or to switch to another model. The agent, at first, interprets the current supply and demand, and the objective function for optimization. Based on the objective function, the agent then selects the available economic models suitable to make the optimization. These models are considered as decision variables. For example, in terms of revenue considering communication overhead, there are three different decision variables—BAR, CDA and CNP (third row of Table A.1). Each of these models shows the strength over other models in its respective domains. The next job by the agent is to identify the respective DOS of the decision variables. Depending on the values of current supply and demand, the agent then selects the model respective to that region. If the decided model is different from the existing one, the new model works as an input to the environment (Fig. 3). Otherwise, the system continues with the existing model. The process continues until and unless the supply or demand becomes unavailable from the environment. In the following section, we describe the development process of our switching framework.

6. Switching framework

The rising demand of Grid computing has led to the requirement of a framework that supports dynamic organization of different models. This section focuses on developing a framework that can dynamically decide a best-suited model to specific scenarios and can switch to that model to ensure improved performance. First, we describe the user, broker and resource models to fit into the framework. Then, we describe some dynamic parameters those can adapt with changing circumstances when switching from one economic model to another. Finally, to facilitate the switching decision and autonomous switching, the role of the switching agent is explained.

6.1. Adaptive entities: behavior description

The user, in this case, must be defined with parameters required to deal with all the models in the framework (Fig. 4). This helps the broker to be adaptive with the environment. The behavior of the broker is regulated according to the current model in the environment. As mentioned earlier, the broker must be able to interpret and understand the message of the agent to know about the recommended model and to behave accordingly.

The performance of job execution in Grid environment typically depends on an efficient brokering mechanism [24]. Therefore, it is crucial to characterize a broker with parameters and capabilities so that it can successfully survive in real computing environment. We have described the broker model suitable for the switching framework in Pseudocode-1. There are two major parts in this code. The first part deals with the interpretation of the message received from the environment about the recommended model and subsequent procedures to get ready for the new model (*line3* through *line17*). Once the message has been disclosed and found that a different model from the current model has been suggested, the first step by the broker would be to stop sending queries to any further resources (*line7*). Please note that the sequence of commands and the delay parameter in the pseudo-code are crucial to support the design principles discussed in Section 6. The delay parameter is defined as the control over delay to send a message/event from one node to another node in the network. Following the termination of the query sending process, the broker takes a while to finish its tasks (e.g. negotiation) with external entities (*line8*). This is to ensure that the switching process does not interrupt any ongoing communication. If the broker has not found its suitable resource(s), it sets the recommended model as the current model and suggests itself to prepare according to the model (*line7* and *line14*).

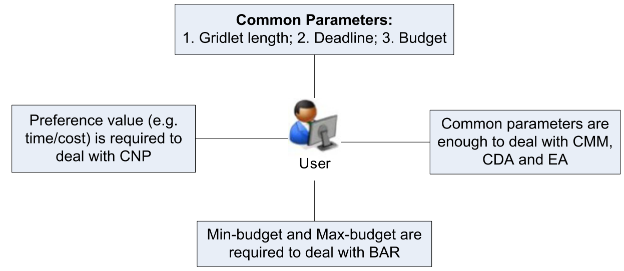


Fig. 4. User describes parameters suitable for switching framework.

Section 2 of the pseudo-code corresponds to the construction of the query according to the new model (*line19* through *line22*). The Pseudocode-1 is in action until the simulation finishes or the broker is terminated.

The agent also notifies the resources about any change of model in the environment. Therefore, the resources also must have capabilities to deal with all the five models dynamically. Pseudocode-2 describes the extended resource model. Pseudocode-2 is also based on the similar ideology to Pseudocode-1, that is, the sequence of commands and delay values are crucial to support the switching framework. After the interpretation that a different model has been suggested by the environment, the resource starts denying any further queries for which the resource has not made any commitments yet (e.g. negotiation) (*line7*). Once the existing negotiation processes have been finished and the resource has not been committed for any broker (*line10*), the resource starts preparing for the new model (*line11* through *line14*). Once the preparation is done, it starts processing new queries according to the suggested model (*line20* and *line21*).

If the new model is either of the auctions (e.g. EA or CDA), the environment first notifies it to the auctioneer. Afterwards, the brokers and resources in the environment are notified about the model, because in order to accept the query/call/proposal from brokers or resources, the auctioneer must be ready beforehand. Now, we discuss the characteristics of our framework when it switches from one economic model to another. As we are dealing with five different models, there will be 20 possible ways of switching from one model to another model. Each of these ways is crucial to consider due to the distinct characteristics of individual models. The following sub-section describes the framework in terms of an autonomous switching environment.

6.2. Autonomous switching

The rising complexity in computing systems has led to the significance of a feasible understanding about the future of those systems to deal with dynamic and conflicting demands. The concept of "autonomic computing" is introduced to deal with this challenge [27]. In this subsection, we describe our framework in terms of an autonomous computing environment. An autonomic computing system is typically defined as a system that can evolve by

Pseudocode-1: Adaptive Broker Model

```
Switch (Track messages using Tag values)
 {
  Case: Message Received from Environment //scheduled by Pseudocode-3
        Cast the message and identify the Recommended_model
        if (Current model != Recommended_model)
        {
              Stop sending request from this broker //to stop the current model
              Wait for a while //this is important if the broker is currently busy talking to any resources
              De-activate the parameters related to Current model
              if(The broker is still looking for the suitable resource)
              {     let delay = 0;
                    Current model = Recommended_model
                    Send a message to itself to prepare for the new model
                    sendMessage(this.Broker, delay, Tag.Current model);
              }else terminate the broker
        }else keep continue with the Current model
  Break;

  Case: Request Send
        Prepare the job specification according to the model
        Start sending the request to the resources one by one
  Break;
 }
```

Pseudocode-2: Adaptive Resource Model

```
 Switch (Track messages using Tag values)
  {
  Case: Message Received from Environment //scheduled by Pseudocode-3
        Cast the message and identify the Recommended_model
        if (Current model != Recommended_model)
        {
              Start rejecting request from any broker with the Current model
              Wait for a while //this is important if the resource is currently busy talking to any broker
              De-activate the parameters related to Current model
              if(The resource is not committed to any broker)
              {     let delay = 0;
                    Current model = Recommended_model
                    Ask itself to prepare for the new model
                    sendMessage(this.Resource, delay, Tag.Current model);
              }else remove the resource from the available resource list
        }else keep continue with the Current model
  Break;

  Case: Request Evaluation
        Prepare for the evaluation process as per the new model
        Start processing requests according to the new model
  Break;
  }
```

self-management process in order to meet the system administrator's goal. In this sub-section, we aim to discuss about such an autonomous system that is able to deal with dynamic integration of different economic models in the Grid environment. We discuss our switching framework in terms of the main objectives of an autonomous system. The objectives are:

Self-configuration

This describes the ability of a system to adapt with unknown scenarios automatically. This may include installation, configuration or integration. Individual components must know how to adapt with the new configuration and rest of the system should accept their presence without any disturbance in the system. We have already presented such a broker and resource models in the earlier sections. We described them in terms of self-organization process to deal with multiple models seamlessly. For example, when either a broker or a resource is notified about a new model, it starts configuring its parameters to deal with the new model, thus to adapt with the new environment. Along with the participants, the system itself must also realize the rationality behind the changing behavior of the participants. The role performed by the system in such an environment is explained in Pseudocode-3.

Self-optimization

An autonomous system must know how to improve the overall performance of the system. To achieve this, it monitors, experiment, or tune its parameters; overall, it tries to learn from the environment. In this paper, we discussed the DOS of different economic models in Grid computing. We then identified the possibility of optimization by dynamically tuning between different models based on relative DOS. To this respect, we developed the switching framework, which is able to switch from one model to another automatically in order to optimize predefined objective functions. To facilitate the optimization process, we develop a switching agent, which we will describe in the following subsection. The other two objectives of an autonomous system—self-healing and self-protection that deal with detection, diagnosis, and repairing from bugs or malicious programs are beyond the scope of this paper. We would like to explore these objectives in future.

Upon receiving the message of a new model from the agent, the environment starts processing its subsequent procedures. The procedures include dealing with parameters to control the behavior of the current model and to inform the participants in the environment about the new model. Pseudocode-3 describes these contributions in detail. The environment is the first entity that receives

Pseudocode-3: Environment Management – Broadcast Switching Decision – Switch from model-1 to model-2

```
1.   Message Received from Agent  //scheduled by Pseudocode-4
2.   The agent suggests model-2 for current network scenario
3.   Stop initiating new auction instances if the model-1 is an auction
4.   Wait for a while as some processes related to model-1 might be still in execution
5.   De-activate the global parameters related to model-1
6.   let delay = 0;
7.
8.  //Send a message about the model-2 to every broker in the network
9.  Foreach(broker in the broker-List)    //the broker-List is updated continuously as soon as a broker is
10.                                          terminated from the environment
11.   {
12.          Send a message about model-2 to the broker without any delay
13.          sendMessage(this.Broker, delay, Tag.Recommended_model);
14.   }
15.
16. //Send a message about the model-2 to every resource in the network
17. Foreach(resource in the resource-List)   //the resource-List is updated continuously as soon as a
18.                                             resource is committed to a broker
19.   {
20.          Send a message about model-2 to the resource without any delay
21.          sendMessage(this.Resource, delay, Tag.Recommended_model);
22.   }
```

the notification of switching. The entity has been constructed with general components (e.g. *broker-list*, *resource-list*), managing auction entities (e.g. creating auction instances, recording winner details, etc.), and informing network entities about the new model before the model starts its action. Once the environment notices that the agent (*line2*) has suggested a new model, it waits for a while to neutralize the effect of existing model (*line4*) and quickly broadcast the decision to existing brokers and resources (*line9* through *line22*) through the network.

The notification of a new model to the participants must occur without any considerable delay. Otherwise, the application of the new model at desired scenario might not be possible. However, in reality, the dynamic nature of network latency might lead to some inefficiency in receiving the decision of a new model by the participants. We would like to study such behavior in future. We have, however, observed some undesirable situations during the simulations when switching from CDA to other models. This may happen due to the transaction rapidness of CDA. The moment CDA starts processing, the resources start generating asks continuously. Therefore, by the time the resources are informed about a new model, CDA have already generated a large number of asks for which few acceptances might occur. This hampers to apply the new model at a more desirable scenario. The deviation from the desired scenario becomes more significant if CDA operates longer in the environment, and then the switching happens. This characteristic, when switching from other models is, however, quite negligible.

The main parameter that drives the switching framework is the optimization function defined by the Grid provider. The switching agent must be informed about the optimization function before the simulation starts in order to perform the decision process properly. The following section describes the functions carried out by the agent.

6.3. Switching agent

"Learning"—is an important term in the field of agent technology. It is now a closely related topic between natural and computational systems. We generally use the terms to refer to the improvement of a system based on its experience. However, depending on the nature of a system, learning methods could vary from one another. According to Gerhard Weiss [28], the definition of learning is stated as below.

"*The acquisition of new knowledge and skills and the incorporation of the acquired knowledge and skills in future system activities provided that this acquisition and incorporation is conducted by the system itself and leads to an improvement in its performance*."

The objective of autonomous system and learning is closely coupled. However, learning is particular in dealing with individuals. We develop an agent that is able to apply its knowledge based on previous identification (Table A.1) towards improving the performance of the system. Depending on the degree of freedom in taking decision, learning can be classified into two principle categories:

- *Centralized learning*: This is a kind of learning where an agent is independent in processing the learning activities and decision. The system's overall performance solely depends on this agent's learning and decision-making capabilities.
- *Decentralized learning*: In this case, individual agents are designed to carry out their respective activities towards improving the system performance.

As we are building a single agent to improve the performance of the system through deciding which economic model to use when, our approach goes under the centralized learning. The agent now needs to apply this knowledge and to prove its effectiveness in terms of optimizing different performance metrics in a dynamic Grid environment. To the knowledge here, we mean that the DOS identified in Table A.1. Based on this knowledge, the agent now needs to decide which economic model to consider at a specific scenario in a dynamic environment. We have presented an example of the agent's knowledge representation below.

The agent's knowledge representation

In our framework, the agent uses the economic models as input variables for optimization in the environment. As we are dealing with five different models, the problem space defined by the agent will be,

$$\textit{Problem Space} = \{CMM, BAR, EA, CDA, CNP\}.$$

Now, based on the function that needs to be optimized, the agent manipulates its decision space. To do this, the agent uses our identification of DOS. For example, if the optimization function is the minimization of communication overhead, the decision space will be described as,

$$\textit{Decision Space} = \{CMM, CDA, CNP\}.$$

As these three economic models have been identified suitable for minimizing communication overhead in their respective domains compared to other models (Fig. 2), these are considered as decision variables. However, at a specific time in the environment, the agent can choose only one decision variable. To facilitate the decision process, the agent then converts the decision space in terms of their respective domains. From Table A.1, we can write,

$$\begin{aligned} \textit{Decision Space} = \{&(d \geq 1 \textit{ and } ((s+27.7)/(d-2.69)^2) \geq 0.65),\\ &(((s+27.7)/(d-2.69)^2) < 0.65 \textit{ and}\\ &((d-1.24)^2/(s+2.34)) \geq 0.32),\\ &(((d-1.24)^2/(s+2.34)) < 0.32 \textit{ and } s \geq 1)\} \end{aligned}$$

where, s and d refer to the supply and demand respectively at any state in the environment. Now for a specific time, either one of the domains in the decision space would be true. As the agent is notified about every occurrence of supply or demand change in the environment, it can determine the true domain respective to s and d of any given state. Thereafter, the agent chooses the economic model corresponding to that domain as an input to the environment. The nature of s and d in the decision space ensures the scalability of the domains; that is, d can have any value greater than or equal to one.

The most crucial role played in the environment is by the agent. The agent keeps track of supply and demand in the environment and based on this, it makes decision on which economic model to choose at a specific scenario. Pseudocode-4 provides the pseudo code that describes the major actions performed by the agent.

Whenever there is a change in supply and/or demand, the code is invoked by the environment to update its supply and/or demand (s, d) function (*line5* and *line6*). The optimization metric and DOS are imported to the agent before the simulation starts. As mentioned earlier, we are using event-driven mechanism; therefore, whenever there is a change in supply or demand, it attempts to determine the suitable model. If the model is different from the current model, the agent notifies about the new model to the environment (*line16* through *line21*). Based on the optimization metric, the agent first computes its decision space (Domain—$1, 2, \ldots, i$) (*line10*). For example, if the optimization metric is minimization of communication overhead, there will be three different domains (regions) dominated by three different models. Afterwards, the agent determines that for current supply and demand values, which domain is true (*line12* and *line13*). Once the true domain is identified, the model corresponding to the domain is selected as the recommended model (*line14*). The last line of Pseudocode-4 triggers the Pseudocode-3 to be initiated and Pseudocode-3 is then used to let the participants know about the new model. An overview of the workflow in the switching framework has been illustrated in Fig. 5. The mathematical models of the DOS of different economic models help the agent to facilitate the decision process in the following ways,

- As mentioned earlier, the switching decision must be conducted as quick as possible to maximize the utilization of the switching model. Because of the formalization of the DOS, the agent now requires to consume only a little computational power and can make decision very quickly, which otherwise would require to import the respective data sets (the whole matrices) to make relative decisions.
- The second case is about the scalability of our model. As we have obtained the clear trends of the models' strengths, the formalization models further ensure the feasibility of accounting the strength for extended scenarios. We will see the proof for extended scenario in the following section.

The following section describes a simulation study carried out to measure the effectiveness of our framework.

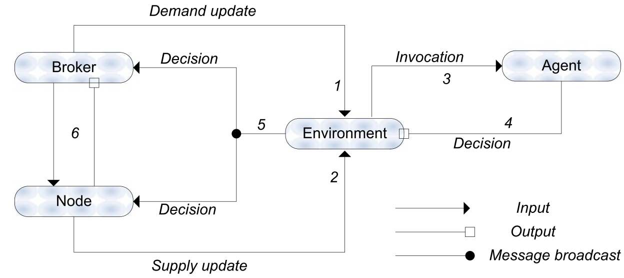


Fig. 5. A workflow diagram for switching framework.

7. Performance analysis

This section evaluates our switching framework and discusses the simulation space in support of the evaluation. We evaluate our switching model in terms of optimizing five different performance metrics individually. For our switching model and for a particular optimization case, we use a constant number for the total number of resources and continuously inject different number of users (Gridlets) in order to obtain a dynamic variation of supply and demand in the environment. For the injection of users, some predefined ranges have been used (fourth column of Table 1). A new set of users is injected when the previous set is finished. This scenario continues until the total number of resources is allocated in the environment. For statistical purpose, *we run all the simulations with five different samples and present only their averages*.

Table 1 presents the number of users and resources used for different simulations aimed for different optimization scenarios. The ranges used for the "number-of-users" column have been set deliberately so that we can obtain possible scenarios to enable the execution of switching framework. For example, for the second simulation, the range of (1, 32) gives higher probability for obtaining different regions those are dominated by different models (Fig. 2). The number-of-resources has been extended to 200 for the first two functions to measure the extensibility of our mathematical models summarized in Table A.1. From now on, we will use the terms user and Gridlet interchangeably.

Optimization metric: revenue over communication overhead

In this case, the Grid would like to maximize its overall revenue while minimizing the communication overhead. Therefore, we let the agent know about the optimization function and the agent dynamically decides which model to be used for what scenario based on the relative DOS.

Fig. 6 illustrates the results obtained during the optimization procedure for revenue over communication overhead. The upper plot compares the performances obtained for different models. The X-axis of the plot shows different supply–demand ratios at different times. The supply starts at 200 and keeps decreasing as occupied by randomly injected users. For injecting users, we use the range (1, 20) in Table 1. The Y-axis represents the revenue over communication overhead ratio for different models. Due to different units, we normalize both parameters and then compute the ratios. We can observe that overall; the switching model performs better compared to any other individual models. The optimization could be further enhanced by utilizing all the three decision variables—CDA, CNP and BAR (Table A.1). However, for this plot, our switching model only uses CDA and BAR, and switches between them based on their relative DOS. For each different scenario (along X-axis), the agent determines which model will be suitable to optimize the function. For example, when the environment notices that the current supply is 160 units and demand is 11 units, the agent computes the suitable protocol for this particular scenario. The down left part of Fig. 6 shows this

Pseudocode-4: Switching Decision

```
Input: Demand, Supply, Optimization-metric, DOS (Table 2), current_model, gradient
Output: Recommended_model

//Keep track of supply and demand
If a broker is terminated, update the demand function
If a resource is committed, update the supply function

If(Optimization-metric = metric-n)  // where n is one of the 11 metrics in Table 2
      {
            Domain-i: Assuming that there are i number of different domains dominated by different
                      economic models for this particular optimization
            compute the gradient using the current Demand and Supply for Domain-i
            if the gradient sits in Domain-i, select the corresponding model of the domain
            Recommended_model = model corresponding to the Domain-i
      }
If(current_model != Recommended_model)
      {
            Let the environment know about the Recommended_model
            let delay = 0;
            sendMessage(environment, delay, Tag.Recommended_model);
      }
```

Table 1
Resource configuration for switching model.

Simulation no:	Optimization metric	Number of resources	Number of users
1	Revenue over communication overhead	200	(1, 20)
2	Communication overhead	200	(1, 32)
3	Social welfare	100	(10, 50)
4	Average turn-around time per Gridlet	100	(20, 50)

Table A.1
Domains of strengths of economic models in Grid computing.

Performance metric	Economic model (DOS) space (s, d)
Revenue	**EA** (whole space)
Revenue over communication overhead	**BAR**$\{d \geq 1$ and $((s + 46.06)/d) \geq 13.01\}$, **CDA**$\{((s + 46.06)/d) < 13.01$ and $\{((s - 1.93)/d) \geq 0.07\}$, **CNP**$\{((s - 1.93)/d) < 0.07$ and $s \geq 1\}$
Communication overhead	**CMM**$\{d \geq 1$ and $((s + 27.7)/(d - 2.69)^2) \geq 0.65\}$, **CNP**$\{((d - 1.24)^2/(s + 2.34)) < 0.32$ and $s \geq 1\}$, **CDA**$\{((s + 27.7)/(d - 2.69)^2) < 0.65$ and $((d - 1.24)^2/(s + 2.34)) \geq 0.32\}$
Success rate	**All Equal**$\{1 \leq d \leq 8 \Vert 1 \leq s \leq 4\}$, **CDA or CMM**$\{d > 8$ and $((s + 15)/d) \geq 4\}$, **CDA or BAR**$\{((s + 15)/d) < 4$ and $s/d \geq 0.83\}$, **CDA or CNP**$\{0.66 \leq s/d < 0.83\}$, **CDA or CNP or CMM or BAR**$\{s/d < 0.66$ and $s \geq 4\}$
Average turn-around time per Gridlet	**CMM**$\{d \geq 1$ and $((125 - s)/d) \geq 6.3 \Vert ((s - 22)/d^2) < 0.003$ and $s \geq 1\}$, **CDA**$\{((125 - s)/d) < 6.3$ and $((s - 22)/d^2) \geq 0.003\}$
Total simulation time	**CMM**$\{1 \leq d \leq 4 \Vert ((s - 9.63)/d) < 0.37$ and $s \geq 1\}$, **CDA**$\{d > 4$ and $((s - 9.63)/d) \geq 0.37\}$
Resource utilization	Similar to the success rate
User utility	**CNP** (approx. whole space) **CMM** (negligible)
Resource utility	**EA** (whole space)
Resource utility over communication overhead	**BAR**$\{d \geq 1, s \geq 20$ and $((s + 90.91)/d) \geq 22.73\}$, **CDA**$s \geq 1, ((s + 90.91)/d) < 22.73 \Vert s < 20\}$
Social welfare	**CDA**$\{d \geq 1$ and $s/d \geq 0.91\}$, **CNP**$\{s/d < 0.91$ and $s \geq 1\}$

computation process. The agent verifies all the decision variables in terms of their relative DOS using the current supply and demand parameters. After the domain test, the agent determines that BAR is the suitable protocol for the scenario. The downright part of Fig. 6 shows the screenshot that the system is switching from CDA to BAR due to changing scenario. Because of switching between the models, the system is able to better utilize the relative strengths of the models and ultimately to optimize the function. Thus, the switching model outperforms all the five economic models individually.

The EA and CNP show the lowest performance due to their highest communication overhead. Because of more supply than demand (Table 1), even to satisfy a single Gridlet, CNP produces a huge amount of messages. The sizes of the groups of potential resources in this case is higher [29]. This prevents CNP to outperform even the EA (Fig. 7). On the other hand, CDA, CMM and BAR show the quite competitive performance. As we have extended the total number of resources up to 200 and still it is doing the optimization, the scalability of the formalization models is satisfied.

Optimization metric: minimizing communication overhead

In this particular case, the Grid would like to minimize its overall communication overhead. Fig. 7 depicts the results obtained from different economic models. Overall, the switching model produces minimum communication overhead. Once again, the optimization could be enhanced by utilizing all the decision variables. In this case, we use only two variables (CDA and CMM), whereas, there are three decision variables for this metric—CDA, CMM and CNP. Because of switching between the models based on their respective DOS, switching model outperforms all the others. The CMM shows the second best performance in this case. Due to additional supply (Table 1), the spot prices in CMM are low, which causes most of the users to be accepted very quickly. Thus, it prevents the model from exchanging a huge amount of messages. The CDA shows competitive performance with CMM at the end whereas; due to multiple rounds in BAR, it shows the third best performance. As mentioned earlier, because of more supply, the sizes of the groups of potential resources in CNP are larger. Therefore, to provision a single Gridlet, it requires exchanging a huge amount of messages. On the other hand, even EA uses multiple

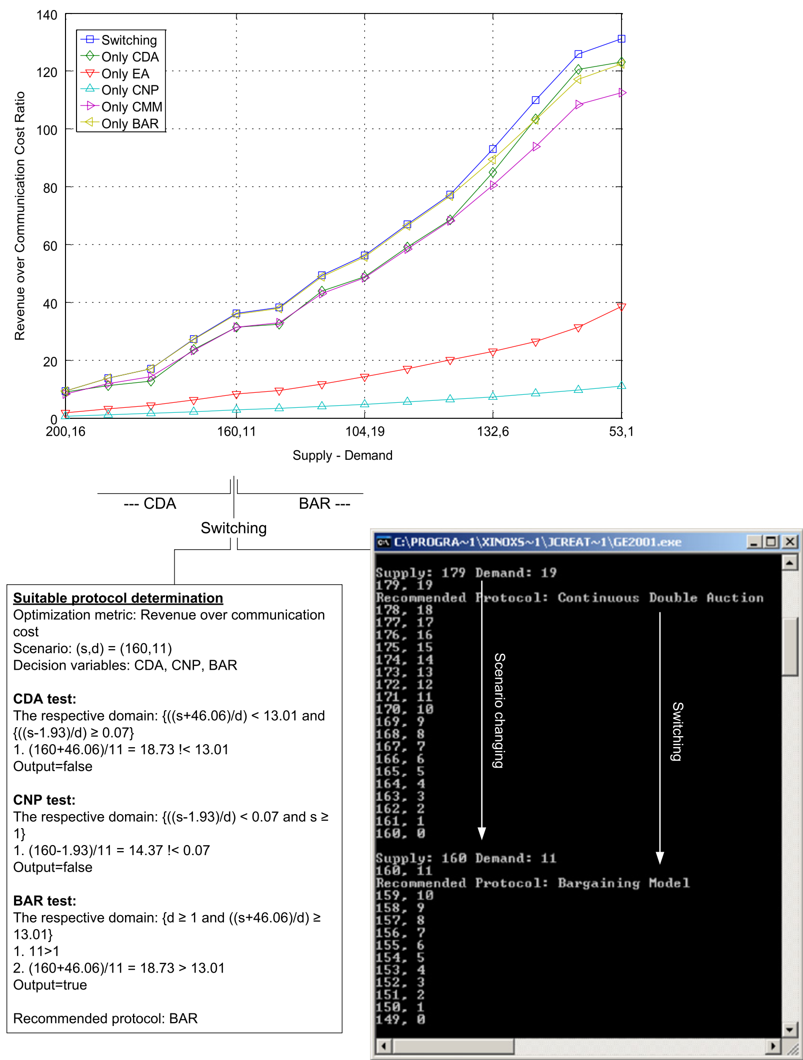


Fig. 6. Scenario illustrated for revenue over communication overhead optimization.

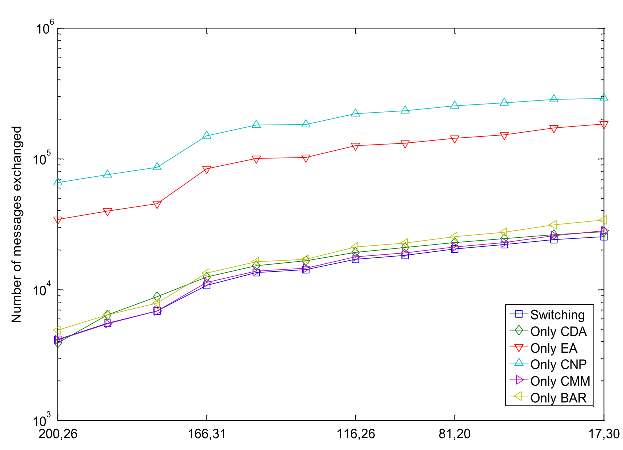


Fig. 7. Scenario illustrated for minimizing communication overhead.

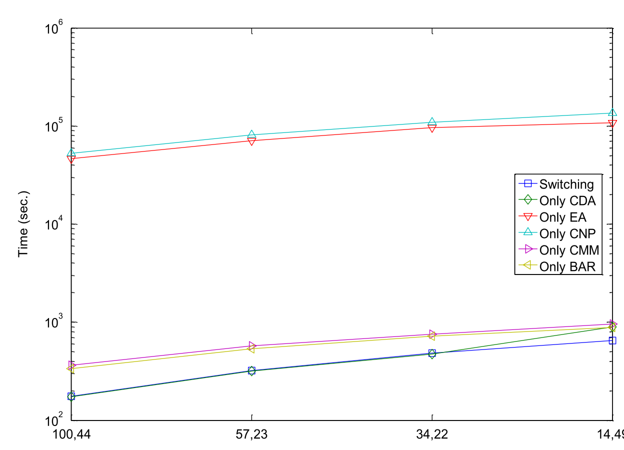


Fig. 8. Scenario illustrated for minimizing average turn-around time per Gridlet.

rounds, as the total number of users in the environment is low and the auction groups are only formed with interested users [29], it produces a lower number of messages for a single auction. Therefore, EA performs better than the CNP in this case.

Optimization metric: average turn-around time per Gridlet

This is the average time required for a particular Gridlet to know about its ultimate acceptance or rejection notification. If the Grid would like to minimize the average turn-around time per Gridlet, this optimization procedure is computed. Fig. 8 shows the results obtained for the metric in terms of different models. Overall, the switching model performs better than the others do.

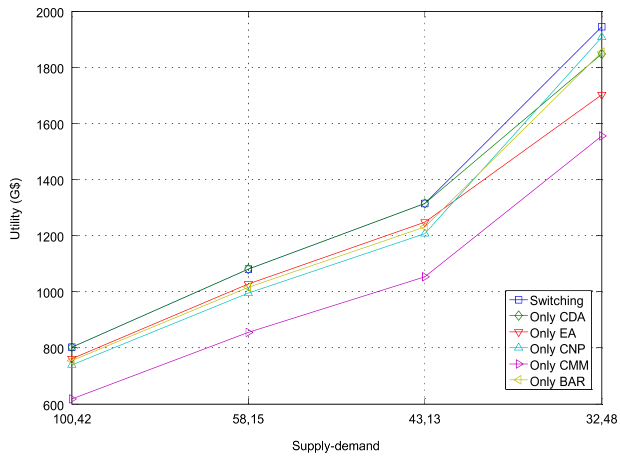


Fig. 9. Scenario illustrated for optimizing social welfare.

There are two decision variables for this function—CDA and CMM. It can be observed from Fig. 8, for a certain scenarios (until (34, 22)), switching model shows equal performance with CDA. This indicates that until (34, 20), switching model uses CDA. At (34, 20), the model switches to CMM, because CMM has been determined as the suitable protocol for this scenario. Staying longer in the environment and keep switching between the models would enhance the optimization further. The BAR shows slightly better performance than the CMM. Due to better Pareto-optimality[4]in resource allocation, BAR consumes lower time. Due to different working principles of EA and CNP, their performances are much low.

Optimization metric: social welfare

In this case, the Grid would like to increase the utilities for both users and resources in the environment. Fig. 9 demonstrates the overall performance achieved by different models for social welfare. The switching model, overall, outperforms all the other models. Until the model switches to CNP at (43, 13), its performance coincides with that of the CDA. The moment it switches to CNP at (43, 13), the performance starts increasing; because CNP has been determined to be suitable by the agent for this scenario. Due to the same reason, CNP alone, in this region, shows improved performance. The BAR overcomes EA at (43, 13). Due to higher supply than the demand, the users in BAR can quickly obtain resources without much competition making it unable to utilize users' budgets. This helps the users to maximize their utilities, thus contributing to maximize the overall welfare. The CMM shows the lowest performance as usual.

8. Future research directions

The opportunity of utilizing the potential of different economic models in different Grid scenarios is examined in this paper. Consequently, the paper gave rise to some challenges such as practical-ability that need to be evaluated with importance. This paper has opened up a new door for economic-based distributed resource management in Grid computing and may be extended to the following works.

Supporting Cloud computing infrastructure

Cloud is another recently emerged distributed computing paradigm, which can be distinguished from other conventional computing platforms for its focus on additional scalability, dynamic configuration and virtualized services. In spite of the additional focus of Cloud computing, Grid and Cloud basically share the similar concepts and therefore, face similar challenges such as dynamic resource configuration, resource utilization, scalability, and data management [26]. Therefore, we would like to explore the possibility of implementing the switching framework in Cloud infrastructure as well. One challenge to implement our model in the Cloud that we see at present is defining the supply model as resources are scaled up and down virtually. Therefore, perceiving actual resource availability at a particular time is crucial to ensure Pareto optimal resource allocation in the environment. The commercialized approach of Cloud computing further encourages us to evaluate the effectiveness of our framework in that particular environment. We believe, our findings in distributed economic-based resource management would help to make a benchmark by cloud providers around the world in constructing their business strategies.

Supporting multiple application types

The application of Grid computing has been widening day by day. Therefore, the execution model must support multiple application types. Currently, our framework is suitable only for Bag-Of-Tasks type applications, such as data mining, design exploration or parameter sweep. This kind of applications does not require the tasks/jobs/Gridlets to communicate among each other. However, there are other types of applications, such as Message Passing Interface (MPI) and workflow require communication among the tasks. Simulations in Bio-informatics and weather forecasting are examples of such kind of applications. We aim to design our framework to support MPI or workflow type and to evaluate the effectiveness of different economic models in those application domains.

The aforementioned applications are mainly computationally intensive. However, a large community is also seeking the possibility of utilizing Grid resources for data-intensive applications such as investigating material properties or climate modeling. In data Grids, thousands or millions of data sets are stored and replicated across a Grid network. These data sets are then invoked and processed to generate meaningful results for users worldwide. A range of parameters, such as bandwidth for data replication, data value to replicate to a particular site, data storage capacity, computational requirements, and data security could be successfully managed and regulated using economic models. Therefore, we would like to investigate the performance of different economic models for data-intensive applications as well.

Supporting multi-criteria optimization and improving the switching agent

Currently, our switching framework supports optimization only for a single performance metric. We aim to improve the switching agent so that it can provide decisions suitable for optimizing multiple performance metrics. In such a scenario, a Grid provider might want to optimize a combination of metrics, such as revenue, average turnaround time per job and social welfare. There are several methods including Genetic and Evolutionary algorithms in the literature to solve such problems. One of the suitable approaches within these algorithms that we plan to explore is the ranking method. According to this method, a number of objective functions that need to be optimized are ranked in terms of the provider's preference values. This preference relation is then applied over the solution space that consists of the suitable models identified for those optimization functions. The agent will then dynamically decide which model to use based on the solution space.

The evaluation of our switching framework in real Grid computing networks might lead to some inefficiency in terms of reflecting on a desired supply and demand ratio due to the random nature of network latency. Therefore, we aim to design our agent using fuzzy intelligence so that it can conduct switching process depending on some approximate values obtaining from the network behavior.

[4] A special resource allocation process, in which allocating of a particular resource is not supposed to affect other resources that are currently being allocated or executed which ultimately results in economically efficient resource allocation [30].

9. Conclusions

The growing and dynamic interest of Grid computing has led to the deployment of different market mechanisms. To be adaptive with future computing environment, dynamic and reliable organization of different market mechanisms are essential. Therefore, understanding the languages and values of these market mechanisms would help to evolve the Grid's vision of a worldwide virtual computing platform. This paper discussed the possibilities and challenges for utilizing the potential of different models in different Grid scenarios. We have developed an adaptive economic-based switching framework in this regard. The framework showed its effectiveness in dynamically switching from one economic model to another depending on the models' domains of strengths. It further proved the suitability in optimizing various objective functions in an autonomous computing environment. To facilitate the optimization process, an agent dynamically made decision and decided which model to be used and when and for what purpose to be used. The paper contributed a new dimension for analyzing the computational economy for future Grids. Our findings, in this paper, would inspire economically inspired Grid organizations in shaping their business models and towards developing an incentive-oriented computing platform.

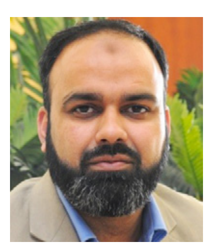
Appendix

See Table A.1.

References

[1] David De Roure, Mark A. Baker, Nicholas R. Jennings, The evolution of the grid, in: F. Berman, G. Fox, T. Hey (Eds.), Grid Computing: Making the Global Infrastructure a Reality, John Wiley & Sons Ltd., 2003, pp. 65–100.
[2] Ian Foster, Carl Kesselman, Steven Tuecke, The anatomy of the grid: enabling scalable virtual organizations, Int. J. High Perform. Comput. Appl. 15 (3) (2001) 200–222. http://dx.doi.org/10.1177/109434200101500302.
[3] Klaus Krauter, et al., A taxonomy and survey of grid resource management systems for distributed computing, Softw. - Pract. Exp. 32 (2) (2002) 135–164.
[4] W.H. Bell, D.G. Cameron, R. Carvajal-Schiaffino, A.P. Millar, K. Stockinger, F. Zini, Evaluation of an economy-based file replication strategy for a data grid. Paper Presented at the Proceedings, in: CCGrid 2003. 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003, 12–15 May.
[5] Rajkumar Buyya, David Abramson, Jonathan Giddy, An economy driven resource management architecture for global computational power grids, in: Paper Presented at the In proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA 2000, Las Vegas, USA, 2000.
[6] Ralph Cassady, Auctions and Auctioneering, University of California Press, 1967.
[7] R.G. Smith, The contract net protocol: high-level communication and control in a distributed problem solver, IEEE Trans. Comput. C-29 (12) (1980) 1104–1113.
[8] Rajkumar Buyya, David Abramson, Jonathan Giddy, Heinz Stockinger, Economic models for resource management and scheduling in Grid computing, Concurr. Comput.: Pract. Exper. 14 (13–15) (2002) 1507–1542.
[9] James Broberg, Srikumar Venugopal, Rajkumar Buyya, Market-oriented grids and utility computing: the state-of-the-art and future directions, J. Grid Comput. 6 (3) (2008) 255–276. http://dx.doi.org/10.1007/s10723-007-9095-3.
[10] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, Future Gener. Comput. Syst. 25 (6) (2009) 599–616.
[11] Aminul Haque, Saadat M. Alhashmi, Rajendran Parthiban, A survey of economic models in grid computing, Future Gener. Comput. Syst. 27 (8) (2011) 1056–1069. http://dx.doi.org/10.1016/j.future.2011.04.009.
[12] Aminul Haque, Saadat M. Alhashmi, Rajendran Parthiban, Identifying and modeling the strengths and weaknesses of major economic models in grid resource management, J. Grid Comput. 12 (2) (2014) 285–302. http://dx.doi.org/10.1007/s10723-013-9289-9.
[13] Rafael Moreno, Ana B. Alonso-Conde, Job scheduling and resource management techniques in economic grid environments, in: Paper Presented at the Across Grids, 2004.
[14] Gunther Stuer, Kurt Vanmechelen, Jan Broeckhove, A commodity market algorithm for pricing substitutable Grid resources, Future Gener. Comput. Syst. 23 (5) (2007) 688–701.
[15] Richard Wolski, James S. Plank, John Brevik, Todd Bryan, G-commerce: market formulations controlling resource allocation on the computational grid, in: Paper Presented at the Proceedings 15th International Parallel and Distributed Processing Symposium, 2001.
[16] Zhu Tan, John R. Gurd, Market-based grid resource allocation using a stable continuous double auction, in: Paper Presented at the 8th IEEE/ACM International Conference on Grid Computing, 2007.
[17] Kwang Mong Sim, A survey of bargaining models for grid resource allocation, ACM SIGecom Exch. 5 (2005) 22–32.
[18] Kwang Mong Sim, Relaxed-criteria G-negotiation for Grid resource co-allocation, ACM SIGECOM: E-Comm. Exch. 6 (2) (2007) 37–46. http://doi.acm.org/10.1145/1228621.1228625.
[19] Rajkumar Buyya, David Abramson, Srikumar Venugopal, The grid economy, Proc. IEEE 93 (3) (2005) 698–714.
[20] Marcos Assuncao, Rajkumar Buyya, An evaluation of communication demand of auction protocols in grid environments, in: Paper Presented at the Proceedings of the 3rd International Workshop on Grid Economics and Business, GECON 2006, 2006.
[21] Massimiliano Caramia, Stefano Giordani, Resource allocation in grid computing: an economic model, J. WSEAS Trans. Comput. 3 (1) (2008) 19–27.
[22] Rajkumar Buyya, Manjur Murshed, GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing, Concurr. Comput.: Pract. Exper. 14 (13–15) (2002) 1175–1220. http://dx.doi.org/10.1002/cpe.710.
[23] Walfredo Cirne, Francisco Brasileiro, Jacques Sauvé, Nazareno Andrade, Daniel Paranhos, Elizeu Santos-neto, Federal Campina, Grid computing for bag of tasks applications, in: Paper Presented at the Proceedings of the 3rd IFIP Conference on E-Commerce, E-Business and E-Government, Brazil, 2003.
[24] Rajkumar Buyya, David Abramson, Jonathan Giddy, Nimrod/G: an architecture for a resource management and scheduling system in a global computational grid, in: Paper Presented at the The Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region, 2000.
[25] A. Andrzejak, Xu. Zhichen, Scalable, efficient range queries for grid information services, in: Paper presented at the In Proceedings of Second International Conference on Peer-to-Peer Computing, P2P 2002, 2002.
[26] Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu, Cloud computing and grid computing 360-degree compared, in: Paper Presented at the Grid Computing Environments Workshop, 2008, GCE'08, 2008.
[27] J.O. Kephart, D.M. Chess, The vision of autonomic computing, Computer 36 (1) (2003) 41–50.
[28] Gerhard Weiss, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, Cambridge, Massachusetts, 2000.
[29] Aminul Haque, Saadat Alhashmi, Rajendran Parthiban, An inspiration for solving grid resource management problems using multiple economic models, in: K. Vanmechelen, J. Altmann, O. Rana (Eds.), Economics of Grids, Clouds, Systems, and Services. Vol. 7150, Springer, Berlin, Heidelberg, 2012, pp. 1–16.
[30] P. Gradwell, J. Padget, Distributed combinatorial resource scheduling, In Proceedings of the 1st International Workshop on Smart Grid Technologies (SGT-2005), 2005.

Aminul Haque is an Assistant Professor at the Department of Computer Science & Engineering, Daffodil International University, Bangladesh. He earned his Ph.D. degree from the School of Information Technology, Monash University, Malaysia. He completed his B.Sc. in Physics from Shah-Jalal University of Science and Technology, Bangladesh. His area of research interests are: Grid/Cloud Computing, Economic-driven distributed resource management and Agent technology.



Saadat M Alhashmi is an Associate Professor at the College of Engineering and Computer Science, Abu Dhabi University, UAE. He holds a B.Sc. in Engineering from AMU, India, an M.Sc. in Automatic Controls and Systems Engineering from The University of Sheffield and a Ph.D. from Sheffield Hallam University, UK. His area of research interests are: Supply Chain Management, Fuzzy Logic, Grid/Cloud Computing, Multimedia Information Retrieval and Multi-agent Methodology.

Rajendran Parthiban is an Associate Professor at the School of Engineering, Monash University, Malaysia. He completed his B.E. (Hons) in 1997 and Ph.D. in 2004 both from the University of Melbourne, Australia. His research interests are in cost and energy comparisons of optical network architectures, visible light communication and positioning, and grid/cloud computing.