

# Towards Better Understanding of Two Economic Models: A Grid Perspective

Aminul Haque<sup>1</sup>, Saadat M Alhashmi<sup>1</sup>, and Rajendran Parthiban<sup>2</sup>

<sup>1</sup>School of IT, Monash University, Malaysia

<sup>2</sup>School of Engineering, Monash University, Malaysia

---

## Abstract

Economic models play a significant role in performance of grid implementation. In this paper, we studied Double auction and Contract-Net-Protocol, which are two of the widely used economic models in grid computing. Economic models facilitate harnessing grid resources across distributed ownerships. However, dynamic nature of these resources imposes further challenge in seamless collaboration. Agent technology is efficient in grid computing due to their autonomous, distributed and collaborative nature. This work models an agent-based economic architecture that supports dynamic management of distributed resources. A simulation environment is designed and implemented for different grid scenarios using the two models and they are later evaluated. We compare our results in terms of job rejection rate, total revenue gained and utilization of idle resources by the market providers. The experimental results predict the effectiveness of using more than one economic model due to various reasons. Our findings would help grid resource providers to decide which model to use at which scenario in order to optimize their designed objectives.

**Keywords:** Grid; Agent; Double auction; Contract-Net-Protocol; Revenue; Utilization

---

## Introduction

Grid computing is an emerging field and distinguished from traditional computing by its focus on large-scale resource sharing, innovative applications and high performance orientation (Foster et al. 2001). In general, grid computing is a special kind of computing infrastructure that is not under centralized control and is able to gather geographically distributed heterogeneous resources over the internet to solve certain complex problems (such as drug design, protein analysis, and investigate material properties). Coordination of distributed, heterogeneous computing resources creates

virtual organizations that supports for utilization of idle resources (Neumann et al. 2008).

Foster and Kesselman (1999) termed grid computing as a promising platform due to three reasons, – (a) It is more cost-effective than traditional computing, (b) It provides a significant amount of computing power to solve complex problems and (c) Dispersed resources can be shared cooperatively toward a common objective. In spite of the potential that grid computing offers, dynamic nature and distributed ownerships of these resources impose further challenge in

seamless collaboration. The efficiency of agent technology to meet such a challenge has already been discovered, since both of them aim to achieve a large scale open distributed system Foster and Kesselman (1999).

Wooldridge (2002) defined agents as *“a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives”*. Due to the autonomous characteristics of agent technology, it could benefit the grid in dynamically negotiating and managing large-scale heterogeneous resources. In the grid computing, this negotiation could be based on several attributes such as price, quality of service (QoS) and deadline.

Pricing is an important factor for accessing distributed resources which are owned by different owners. Buyya (2002) argues that the grid's heterogeneity and decentralization is similar to the present standard human economies, where market-based mechanisms can be used successfully to manage this environment. There are many problems in managing distributed resources identified by Czajkowski et al. (1998) such as site autonomy problem, heterogeneous substrate problem and cost management problem. Carsten and Ramin (2004) propose market-oriented programming to solve these problems.

Commodity market and bargaining models, in combination, perform better than individually in terms of job rejection rate and total revenue, which is proposed by Haque et al. (2010). In this paper, we focus on Double auction (DA) and Contract-Net-Protocol (CNP) models and identifying their individual strengths and weaknesses in computing grid. DA is one of the widely proposed models in grid computing (Zhu and John 2007, Pourebrahimi et al. 2006, Placek and Buyya 2006), Marcos and Buyya (2006)]. Ouelhadj et al. (2005) found CNP efficient in scheduling distributed grid resources and suitable for utility-based negotiation

(Paurobally 2007). This paper focuses on investigating different strengths and weaknesses of the models in various scenarios by changing the total number of users and providers in the market. We find that different models are suitable for different scenarios and we perceive the significance of using two models in combination. In this paper, we are focusing on provider-side. Hence, our findings only assist to develop providers' strategy. From hereafter, we use the word “grid”, to refer to the word “grid computing”. The rest of the paper is organized as follows:

The following Section presents some works related to DA and CNP in the grid. Then we illustrate our proposed framework. After that, we describe the implementation of DA and CNP suitable for conducting large scale simulation. Then we present experimental results conducted under three different scenarios. Finally we conclude with some future research directions.

## Related Work

This section briefly introduces the DA and CNP and describes the state of the art of the models in grid computing.

### Double Auction

In DA, users post their requests and budgets and service providers their availability at anytime during the trading period. If a user's request matches with a provider's availability, the service-level-agreement (SLA) is established. In this model, service providers are sorted in descending order in terms of their demands and resource users are sorted in ascending order in terms of their budgets. As mentioned earlier, DA is a widely proposed model in the grid. It is well known in this context due to its efficiency for handling large number of users, market equilibrium and least communication overhead (Marcos and Buyya 2006, Buyya et al. 2005).

Zhu Tan and John Gurd compare SCDA (Stable Continuous Double Auction) with CDA (Continuous Double Auction) (Zhu and John 2007). A Compulsory Bidding Adjustment Layer (CBAL) to identify an appropriate bid for a current market is added to CDA. Hence, it is called SCDA. They have shown SCDA as superior to the CDA in terms of economic and scheduling efficiency. However, they have explored the experiments with a limited number of requests and offers, which somehow unable to predict economic efficiency in a large scale grid scenario.

Pourebrahimi et al. (2006) propose CDA against traditional non-market based resource allocation and finds market-based allocation more efficient than simple FCFS (First Come First Serve) mechanism in terms of resource and task utilization. Their proposed pricing function is able to reflect market price based on supply and demand. The focus of our work is to investigate the pros and cons of DA and CNP in a large grid market.

A sealed DA with a Storage Exchange (SX) platform has been applied by Placek and Buyya (2006) as a mechanism to maximize surplus and optimize utilization. They have developed some clearing algorithms that periodically allocate the trades amongst requested bids. They have gone through a limited variation of different parameters (such as Storage Request Bids, Storage Request Capacity and Storage Service Asks). In terms of communication overhead, DA performs better than other auction models such as First-price-sealed, English, Dutch (Marcos and Buyya 2006). However, we present a comparative analysis of DA and CNP to benefit providers.

#### Contract-Net-Protocol

According to this model, a user is called the manager and providers are called the contractors. Here, the manager declares his/her request and invites bids from service providers. Potential contractors evaluate the request and respond by submitting their

bids. The manager evaluates the bids and selects a contractor (usually the shortest bid provider) to proceed. Once the bid is accepted, user is allowed to use the resources. If the selected provider is unable to deliver the committed services, the manager might seek for other providers. Along with DA, multi-agent based CNP is found to be efficient for scheduling and utility-based negotiation (Paurobally 2007) in the grid.

Kunal and Arobinda (2008) adopts CNP to deal with resource heterogeneity and proposes two resource selection policies. One is K-time optimization policy, in which users are sorted in ascending order in terms of their proposed deadlines of finishing respective jobs. Another one is, K-cost optimization policy, in which users are sorted in terms of their budgets, they are willing to pay. The value of K refers to whether to switch from K-time to K-cost or not and vice versa. The drawback of this system is, though the failed users have the chance to re-announce/revise their call for proposals (*cfp*), they still resubmit their *cfp* without changing anything (e.g. increase budget or reduce QoS). Hence, the probability of accepting revised *cfp* would be decreasing and produce high communication overhead.

Paurobally (2007) proposes an iterative CNP for web services that support utility-based negotiation. The paper talks about the similarities between the grid and agent technologies, specially the similarity in collaborating large scale distributed resource. In addition, they have a negotiation mechanism for preferences through which a user can optimize his/her preferences. However, their evaluation considers a limited num of scenarios with a few agents and compares the negotiation mechanism with no negotiation scenario. In addition, they focus on user side whereas our focus is on provider side.

An efficient job scheduling mechanism on the grid using multi-agent system has been proposed by Ouelhadj et al. (2005). They have studied most common and distributed

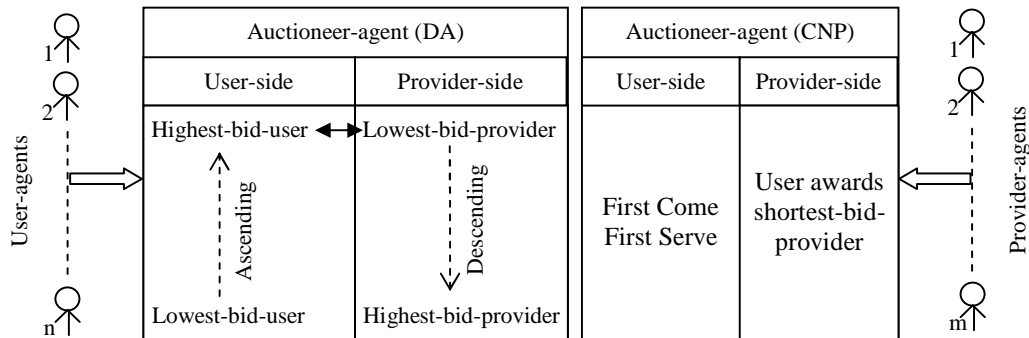
task allocation mechanism using CNP. They have used three types of agents in their proposed framework; User Agent (UA) that acts for a user, Local Scheduler Agent (LSA) which acts for a provider and Super Scheduler Agent (SSA) that acts as a mediator between UA and LSA. Their proposed hierarchical SLA negotiation protocol is for dynamically allocating tasks, balancing resource load and recovering from failures. They have not shown any experimental results, without which one is unable to measure the effectiveness of their proposed scheduling mechanism in real grid scenarios.

The DA and CNP are widely discussed in the literature due to their individual strengths in

the grid. However, it is hard to argue that a single model is suitable to cope with all the scenarios that dynamically changes over times. Hence, we are motivated to focus on both models to find out their individual strengths and weaknesses in different grid scenarios (by varying the number of users and providers). In addition, agent technologies are found efficient for autonomous management of the resources and can be used in grid to automate price negotiation, scheduling decisions, and traffic management for the grid participants (users and providers). Agent technology is also suitable to deal with a large number of grid participants autonomously while carrying distinct goals and objectives.

**Proposed Framework**

In order to work with DA and CNP in an autonomous large scale environment, we propose the following agent-based framework:



**Fig 1. Proposed Framework of Agent-based Auction Mechanism**

Figure 1 presents our proposed framework that deals with agent-based auction mechanism. In our framework, a user-agent (UA) works on behalf of a user and a provider-agent (PA) works on behalf of a provider autonomously. A user can define his/her resources requirements and budget (*cfp*) he/she is willing to pay. Similarly, a provider can define his/her resource availability as well as profit margin to the corresponding agent. In our framework, one can accommodate as many UAs (up to *n*) and PAs (up to *m*), where  $|n, m| > 0$ . For this reason, the framework allows to choose a

range of different values (such as storage, number of CPUs and budget) so that thousands of UAs and PAs can be set autonomously within these bands of values. In our model, auctioneer-agent plays an important role to conduct auction protocol among the users and providers. There are different auctioneers for different auction protocols. The auctioneer for DA is assigned to ascending users based on their budgets and descending providers according to their demands. Then auction starts with the highest-bid-user and lowest-bid-provider. In terms of CNP, the auctioneer allows users to

randomly proceed with the providers and finally awards the job to the shortest-bid-provider. After getting the acceptance, the PA reserves resources for the requested *cfp* and follows its commitment.

### Implementation

We implement the grid simulation model using a cross-platform multi-agent programmable modeling environment known as Netlogo (Sallez et al. 2004, Damaceanu 2008). We choose Netlogo because:

- Netlogo is a FIPA (Foundation for Intelligent Physical Agent) conformant platform (Michael et al. 2006)
- It has extensive built-in models to deal with multi-agent
- It can work as a 'simulated parallel' environment (Sallez et al. 2004)
- It is platform (Mac, Windows, and Linux) independent (Damaceanu 2008)

### Agent-interaction

In the Netlogo framework, three different results can be obtained based on the UA and PA interaction. The first result describes the job rejection rate for a provider. Job rejection could occur due to scenarios such as, disagreement of resource prices or unavailability of resources. This rate is calculated using two parameters - the total number of rejected jobs ( $n_{rejected}$ ) and the total

number of requested jobs ( $n_{requested}$ ). The job rejection rate is assumed to range from 0 to 1. The job rejection rate,  $R_{rate}$ , is given by:

$$R_{rate} = \frac{n_{rejected}}{n_{requested}}$$

The second output demonstrates the total revenue earned by a provider. It sums only the prices of the executed jobs. Hence, the total revenue,  $T_{rev}$ , is given by:

$$T_{rev} = \sum_{i=1}^j M_i$$

Where  $i$  denotes the executed job number,  $j$  denotes total number of executed jobs and  $M_i$  defines mutual price (between a user and a provider) for the  $i^{th}$  executed job. In terms of DA, we consider  $M$  as,

$$M_i = \frac{\text{user's bid} + \text{provider's bid}}{2}$$

The mutual price,  $M$  for DA used here is consistent as proposed by Marcos and Buyya (2006). In terms of CNP, the value of  $M$  is the bid of the shortest bid provider. Since, we deal with simulations, we consider accepted jobs to manipulate  $T_{rev}$  rather than executed jobs. The third output presents how the resources on provider side are utilized. In this paper, we consider the utilization of storage and CPU. The percentage of utilization,  $U$  for a particular resource type,  $res$ , can be formalized by the following equation:

$$U_{res} = \frac{\text{initial number of res} - \text{available number of res}}{\text{initial number res}} * 100 \quad (3)$$

### Implementation of Economic Models

This section illustrates the implementation of the two economic models, DA and CNP. In our system, a user will be considered as a failed job if it cannot establish his/her SLA with any of the potential providers in the market. A provider will be taken out from the market if he/she is no longer able to supply resources for at least one user.

#### Double Auction (DA)

In our model, a UA works on behalf of a user and a PA works on behalf of a provider

autonomously. There are two types of DA in the literature; Continuous DA (auction continues until there is at least one user and one provider) and Periodic DA (auction happens periodically) and both of them have multiple rounds for bidding with users. We apply the former type in our model, since Continuous DA allows receiving users and providers at any time during trading phase. Providers in our model are sorted based on their unit demands on the resources as

presented in Table 1. We sort users based on their budgets obtained using the unit budgets as mentioned in Table 1.

**Contract-Net-Protocol (CNP)**

The communication in CNP happens between a manager and a contractor in our model. Hence, we assume a UA like a manager and a PA like a contractor. The manager initiates his/her *cfp* and invites bids from potential contractors. The contractors then evaluate the *cfp* in terms of resources and proposed budget. The potential contractors would be able to provide the required resources and satisfy themselves with the proposed budget. After that, potential contractors propose their interests on accepting the job. After receiving the proposals, the user selects shortest-bid-provider among the potential providers while ensuring the requested QoS and SLA commitment.

**Bidding Policy**

In our model, each user receives with a budget, which the user is willing to pay. If a UA is unable to establish its SLA even after negotiating with all providers, it will be considered as a failed job. The cost associated with a UA's request is manipulated by the provider, PA using unit price of each particular resource. The cost *C* of a job, *J* requested by a particular user, *UA* can be formalized as follows;

$$C(UA) = \sum_{res=1}^r Req_{res} \times P_{res}(PA) \tag{4}$$

Where, *res* refers to the resource type requested by the user, *UA*. Typically, this can be storage, CPU, and memory. However, we conduct our experiment only with storage and CPU

*r* is the total number of resource types

*Req<sub>res</sub>* means required resource amount of type, *res*

*P<sub>res</sub>* is the unit price (e.g. price/GB storage) for type, *res*. This is a function of a particular provider, *PA*. In our model, *C* is also called the demand by a particular provider, *PA* for the job, *J*. User-agents also use Equation 4 to manipulate their respective budgets.

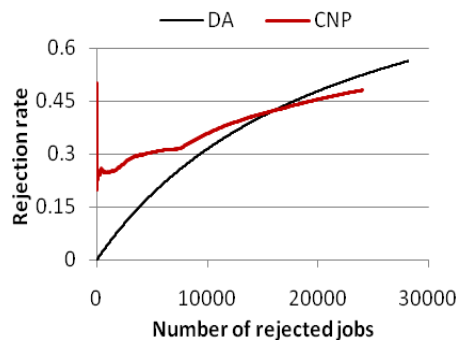
Table 1 shows the resource configuration we have used to conduct our experiments. Column 1 of Table 1 represents different parameters that a user and a provider are supposed to use to set their agents. Since the total number of users and providers in our model is large, we have used the ranges of values (see column 2 and 3) to set a value from the respective ranges automatically. To set the demand on unit storage by a particular provider, range [1-3] is used. For unit processor, it is [7-9]. We use the similar kind of ranges to assign users' budgets in order to maintain the consistency between requests and budgets. This consistency is crucial in case of experimenting with large number of users, since much variation in resource amounts and respective budgets impact on experimental results and hence true evaluation might not be possible.

**Table 1: Resource configuration**

User/provider-level parameter	User-level-range	Provider-level-range
Storage/disk space (GB)	400-450	46500-48500
Number of CPUs (MIPS per CPU)	20-24	2100-2300
Budget/demand (\$)	1-3 (/GB), 3-7 (/MIPS)	1-3 (/GB), 7-9 (/MIPS)

We conduct our experiments by varying supply and demand so that performance patterns for various scenarios can be distinguished and analyzed. For all scenarios, we use the same configuration as presented in Table 1.

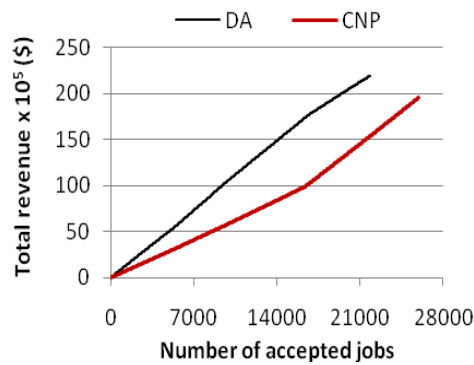
Scenario 1. Supply = Demand: The experiments are evaluated in terms of job rejection rate, total revenue and utilization of



**Fig 2. Job rejection rate comparison**

The trend obtained for DA is smoother than that of CNP. The reason for this is, in DA, users are sorted in ascending order in terms of their budgets and providers are in descending order in terms of their demands. In such a condition, auction starts with the highest budgeted user and lowest demanded provider and moves forward. Hence, in the beginning of auction, the chance of rejecting jobs is very low and the chance gradually increases over time. For CNP, basically auction occurs among one user and multiple providers, where the reception of a particular user by the providers is unknown. Hence, the chance of that particular user to be accepted might be less due to budget constraint even though the job is accepted by the lowest demanded provider. Due to this random nature of receiving jobs, the trend for CNP is also random. We could explain this scenario in terms of how a particular auction supports to prevent from rejecting jobs. In terms of DA, the acceptance of jobs is being supported

idle resources. In our first scenario, there are 50,000 users and 500 providers. The number of users and providers are set in a way so that the respective requests and offers are approximately equal. Figure 2 shows the comparison of job rejection rate between DA and CNP. Trend of a particular model in the figure explains the total rejection rate over the experiment by 500 providers. In terms of overall rejection rate, CNP performs better than the DA.



**Fig 3. Revenue comparison**

from two perspectives, which are, user perspective and provider perspective whereas for CNP, it is only from provider perspective. However, this can be applied only to interpret the former parts of the trends. In terms of latter parts of the trends, CNP outperforms the DA. We could explain this by using the supportive idea inversely. In both cases, lower demanded providers are utilized quickly, which leaves the providers with higher demands. Hence, the rejection rate becomes high at the end. At the end, CNP performs better than DA, since the chance of accepting jobs by DA becomes harder over times than that by CNP. For DA, both users and providers start off opposing jobs from being accepted whereas for CNP, only providers do the same work. However, in terms of overall rejection rate, CNP performs better than DA, since eventually, the intensity of two-sided support in rejecting jobs becomes high than one-sided under such a scenario.

In terms of revenue, DA performs better than CNP, though DA accepts less jobs than CNP (Figure 3). Theoretically, DA is suitable for revenue, since it sorts users in way such that highest budgeted users receive the chance of dealing providers at first. It further sorts providers in a way (i.e. provider are sorted in descending order according to their demands) so that the chance of rejecting jobs

is minimized. The combination of these two approaches helps DA in maximizing revenue. On the other hand, CNP accepts a user after dealing with multiple providers and lowest demanded provider wins the user. Hence, chance of maximizing revenue decreases. This is the reason for DA in gaining more revenue even rejecting more jobs than CNP does.

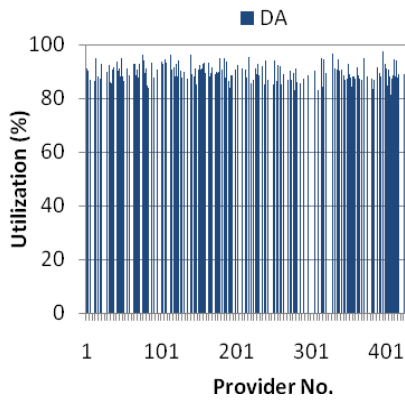


Fig 4. Resource utilization for Double auction

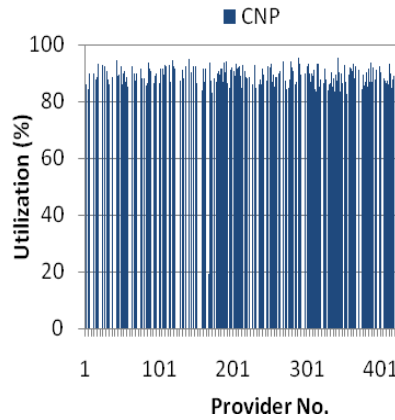


Fig 5. Resource utilization for Contract-net

In terms of resource utilization, we obtain similar pattern for storage and processors. Hence, we only submit the solution for storage. Resource utilization is obtained by using Equation 3. Figure 4 illustrates the utilization for DA and Figure 5 for CNP. The X-axes represent provider number and there are 500 providers in each case. Because of

accepting more users, CNP utilizes more resources than that of DA. The white spaces in between the solid colors refer to the zero utilization by the respective providers. These providers are basically the highest demanded providers and hence utilizing resources become impossible by them.

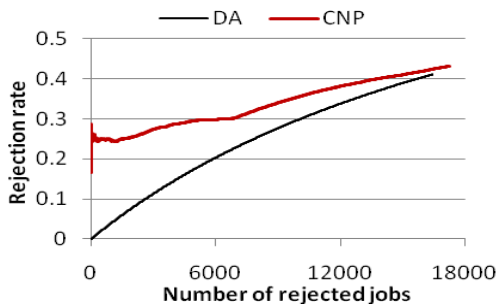


Fig 6. Job rejection rate comparison

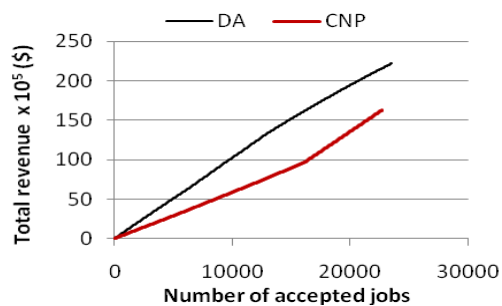
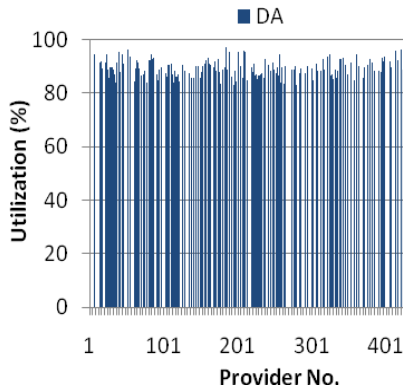


Fig 7. Revenue comparison

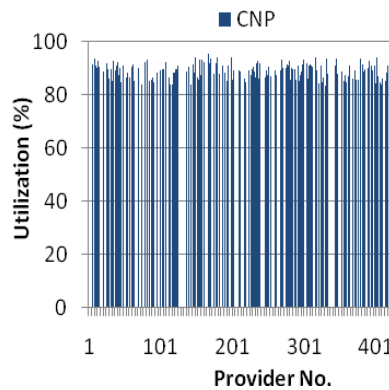


Scenario 2. Supply > Demand: We conduct our second experiment with 40,000 users and 500 providers in a way so that supply is greater than demand. Due to more supply, the chance of rejecting jobs decreases. Hence, the overall rejection rate is less (Figure 6) than that in Scenario 1 (Figure 2). However, in this case, DA performs slightly better than CNP. This indicates that DA is able to utilize the advantage of additional supply whereas the CNP is not able to do that. Because of having less number of users than the providers compared to Scenario 1, in terms of DA, the acceptance rate by the lowest demanded providers increases. For CNP, because of additional supply, though the

chance of rejecting jobs decreases, due to the randomness in receiving jobs, CNP underperforms the DA. Hence, we could predict that as long as number of users is kept decreasing while the number of providers is constant, DA starts off performing better than CNP. In terms of revenue, DA brings significant revenue to the market compared to CNP (Figure 7). In spite of the slight improvement in rejection rate, DA shows dramatic performance in revenue. There are two reasons for this. One reason is already mentioned to interpret Scenario 1 (Figure 3). In addition, high acceptance rate assists in maximizing revenue for DA.



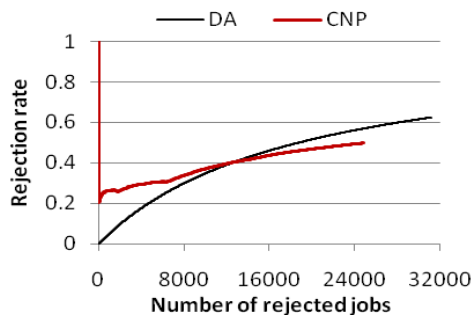
**Fig 8. Resource utilization for Double auction**



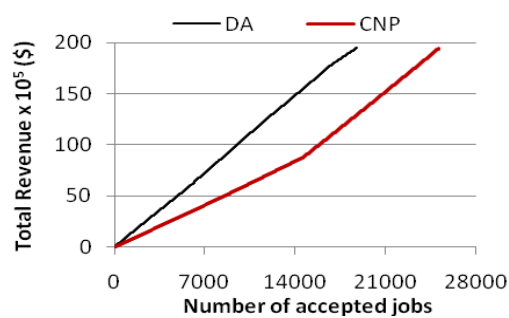
**Fig 9. Resource utilization for Contract-net**

The explanation provided to describe the rejection rate can also be used to understand the utilization patterns in Figure 8 and Figure 9. Since the number of accepted jobs in DA is slightly higher than CNP (Figure 7), the utilization for DA (Figure 8) is slightly better than CNP (Figure 9). It is worth mentioning

here that because of additional supply, in terms of DA, the number of providers with zero utilization decreases by 17 whereas in terms of CNP, it increases by 39 compared to Scenario 1. This again indicates the continuous improvement in utilization by DA and declination by CNP as long as the number of users is kept decreasing while the number of providers is constant.



**Fig 10. Job rejection rate comparison**

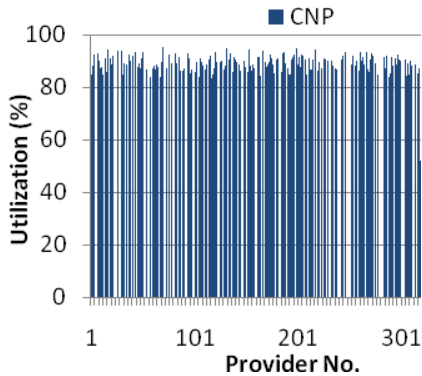


**Fig 11. Revenue comparison**

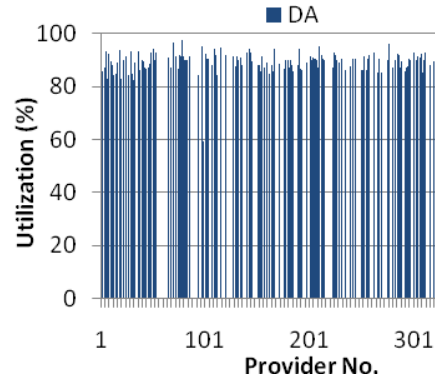
Scenario 3. Supply < Demand: In our third experiment, there are 50,000 users and 400 providers. The same configuration as in Table 1 is used such that supply is lower than demand. Under such a scenario, in terms of job rejection rate, CNP performs better than DA (Figure 10). Because of deduced supply (400 providers) compared to the earlier scenarios (500 providers), the overall job rejection rate is greater than earlier. The geometry of the trends can be explained in a similar way as explained for Scenario 1. In this scenario, CNP rejects fewer jobs than that by DA. Since DA employs two-sided auction and due to fewer number of providers, number of lowest demanded providers also becomes fewer. Hence, accommodate more jobs becomes hard due to resource constraint. In terms of CNP, since auction happens only at provider side and users arbitrarily deal with providers, the chance of resource utilization by maximum providers increases. Unlike DA, CNP does not depend on the number of providers. Hence, it becomes easier to accommodate more jobs by CNP.

Figure 11 illustrates the comparison of revenue between DA and CNP. Competitive revenue is achieved in this case. As mentioned earlier, DA is suitable at

maximizing revenue, since it filters the highest budgeted users and proceeds the auction. Hence, even rejecting more jobs, DA receives significant revenue. On the other hand, due to accepting more jobs compared to DA, it is possible for CNP to gain competitive revenue. It also indicates that if the number of providers further decreases, CNP might perform better than DA. Because of the high number of accepted jobs, CNP utilizes more resources (Figure 12) than that of DA (Figure 13). The average revenue achieved by CNP is 56.17% whereas for DA, it is 42.24%. For CNP, the number of providers with zero utilization is 148 and for DA, it is 211. It is worth mentioning that the reason for the large number of rejected jobs and hence the large number of providers with zero utilization in our model is because of single round auction. In a single round auction, users do not have any chance to revise their jobs and re-negotiate with providers. Hence, failed users after the first round cannot counter bids on their jobs, which eventually treat them as rejected jobs. However, in future, we would like to make it iterative.



**Fig 12. Resource utilization for Double auction**



**Fig 13. Resource utilization for Contract-net**

Table 2 presents the solutions obtained from the experiments conducted with three different scenarios. The resource utilization presented here is the average utilization of storage and processor. As can be seen from the table, a single model is not suitable in all the scenarios and at all criteria. This inspires us to use more than one economic model in the grid environment where scenario changes very frequently. This aspires of using dynamic switching between the models based on a given scenario. However, before commencing such a market based computing, providers are supposed to be able to define their preferences on economic models for a particular scenario and set their agents accordingly so that agents on behalf of them can dynamically switch between the models. For example, during Scenario 1, if a

provider’s preference is on revenue, he/she chooses DA and if the preference is on satisfying more users, CNP is preferable. Some providers might want to increase their individual welfare by considering some economic efficiency. During Scenario 2, DA performs better than CNP in all cases except the time consumption. However, CNP does not perform significantly better than DA in this regard. Hence, we recommend DA as a preference from provider perspective. In terms of Scenario 3, in all cases, CNP outperforms DA except revenue, though CNP receives competitive revenue. Hence, we recommend providers to choose CNP as their preference. Agents in such an environment need to be proactive so that they can sense a particular scenario and switch to a model that suits the defined preferences best.

**Table 2: Summary results for the three different scenarios**

Scenario	Number of providers	Number of users	Number of rejected jobs		Total revenue × 10 <sup>5</sup> (\$)		Resource utilization (%)		Net simulation time (sec)	
			DA	CNP	DA	CNP	DA	CNP	DA	CNP
Supply = Demand	500	50,000	28,179	24,045	219.12	195.76	41.58	49.13	30056.43	44261.70
Supply > Demand	500	40,000	16,445	17,260	222.29	163.19	44.79	43.04	21166.46	19137.59
Supply < Demand	400	50,000	31,224	24,891	194.86	194.31	44.60	59.42	27939.21	22514.74

## Conclusions

Grid computing aims to build world-wide virtual organization through collaborating resources across multiple administrations. Economic models are feasible in collaborating distributed resources over distinct rules and policies. Selection of a particular economic model in grid environment also depends on providers' individual preferences, optimal allocation, and system performance. However, one model is different from another in terms of resource allocation policy, optimizing users' preferences, and satisfying providers' demands. In this paper, we studied with two of the widely proposed models in grid computing, which are, Double auction and Contract-net-protocol. We attempted to investigate the individual strengths and weaknesses of the models in three different scenarios. After a comprehensive analysis with the experimental results, we found that a single model is not suitable at all the times and all the cases. For the first two scenarios, Contract-net-protocol model performs better than Double auction model in terms of revenue. However, for the first scenario, Contract-net-protocol model is suitable in satisfying more users. For the third scenario, we recommend Contract-net-protocol to providers.

The model presented in this paper is single round. In future, we would like to make it iterative so that failed users can revise their jobs and renegotiate with providers. This would reduce the job rejection rate by a particular provider. Future work needs to be done on dynamic changes of resource prices based on supply and demand. We further would like to dynamically changes market scenarios and develop agents working on behalf of providers would be able to switch between the models so that providers' individual preferences can be maximized.

## References

Buyya, R. (2002), 'Economic-based distributed resource management and

scheduling for Grid computing,' *Doctoral thesis*, Monash University.

Buyya, R. Abramson, D., and Venugopal, S. (2005), 'The Grid Economy' Proceedings of the IEEE, 93 (3), 698-714.

Carsten, E. and Ramin, Y. (2004) Applying economic scheduling methods to Grid environments, Grid resource management: state of the art and future trends, Kluwer Academic Publishers, 491-506.

Czajkowski, K. Foster, I. Karonis, N. Kesselman, C. Martin, S. Smith, W., and Tuecke, S. (1998) A resource management Architecture for metacomputing systems, Springer, Berlin, Germany, 62-82.

Damaceanu, R.-C. (2008) 'An agent-based computational study of wealth distribution in function of resource growth interval using NetLogo,' *Applied Mathematics and Computation*, 201 (1-2), 371-377.

Foster, I. and Kesselman, C. (1999) The grid blueprint for a future computing infrastructure, Morgan Kaufmann.

Foster, I. Kesselman, C., and Tuecke, S. (2001), 'The Anatomy of the Grid: Enabling Scalable Virtual Organizations,' *International Journal of High Performance Computing Applications*, 15 (3), 200-222.

Haque, A. Alhashmi, S. M., and Parthiban, R. (2010), 'Implementation of Autonomous Pricing Mechanism in Grid Computing' The 12th International Asia-Pacific Web Conference (APWeb), 6-8 April 2010, Busan, Korea, DOI=<http://doi.ieeecomputersociety.org/10.1109/APWeb.2010.68>

Kunal, G. Arobinda, G. (2008) 'Resource Selection in Grids Using Contract Net' In Proceedings of the 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2008), IEEE Computer Society.

- Marcos, A. and Buyya, R. (2006), 'An evaluation of communication demand of auction protocols in grid environments' In Proceedings of the 3rd International Workshop on Grid Economics and Business (GECON), World Scientific Press.
- Michael, J. N. Nicholson, T. C., and Jerry, R. V. (2006) 'Experiences creating three implementations of the repast agent modeling toolkit,' *ACM Trans. Model. Comput. Simul.*, 16 (1), 1-25, DOI=<http://doi.acm.org/10.1145/1122012.1122013>
- Neumann, D. Stößer, J. Weinhardt, C., and Nimis, J. (2008) 'A Framework for Commercial Grids—Economic and Technical Challenges,' *Journal of Grid Computing*, 6 (3), 325-347.
- Ouelhadj, D. Garibaldi, J. MacLaren, J. Sakellariou, R., and Krishnakumar, K. (2005) 'A Multi-agent Infrastructure and a Service Level Agreement Negotiation Protocol for Robust Scheduling in Grid Computing,' *Advances in Grid Computing-EGC 2005*, Springer-Verlag, 651-660.
- Placek, M. and Buyya, R. (2006), 'Storage Exchange: A Global Trading Platform for Storage Services' 12th International European Parallel Computing Conference (EuroPar), Springer-Verlag, Berlin, Germany, 425-436.
- Paurobally, S. (2007) 'A Framework for Web service negotiation,' *ACM Transaction of Autonomous and Adaptive System*, 2 (4), 14, DOI=<http://doi.acm.org/10.1145/1293731.1293734>
- Pourebrahimi, B. Bertels, K. Kandru, G. M., and Vassiliadis, S. (2006), 'Market-Based Resource Allocation in Grids' Second IEEE International Conference on e-Science and Grid Computing. (e-Science '06), 80-80.
- Sallez, Y. Berger, T., and Tahon, C. (2004), 'Simulating intelligent routing in flexible manufacturing systems using NetLogo,' IEEE International Conference on Industrial Technology, 2, 1072-1077.
- Wooldridge, M. (2002) *An Introduction to MultiAgent Systems*, John Wiley & Sons Ltd., Chichester, England.
- Zhu, T. and John, R. G. (2007), 'Market-based grid resource allocation using a stable continuous double auction,' In Proceedings of the 8th IEEE/ACM International Conference on Grid Computing, IEEE Computer Society, DOI=<http://dx.doi.org/10.1109/GRID.2007.4354144>