

Analysis the Performance of CDS Algorithm with Constrained Price and Deadline in Grid Computing

Deepti Gupta

AmbalaCollegeofEngineering and Applied Science
Mithapur,AmbalaCantt.Haryana, India

Verender Singh Madra

AmbalaCollegeofEngineering and Applied Science
Mithapur,AmbalaCantt.Haryana, India

ABSTRACT

Clusters, grids and P2P networks enable aggregation of resources and creation of virtual enterprises for solving large scale problems. Resource planning in Grid computing is an advanced task owing to the heterogeneous and dynamic nature of the resources. Trade based scheduling is more attractive from point of view of business. This paper describes the design, implementation and evaluation of an economic strategy based Grid resource scheduling mechanism. It takes into account the architectural features, special requirements of computational Grids with ensuring economic efficiency. The design is concentrated on two goals. Mainly, the Grid computing environment is regarded as a distributed two-sided trade market; competition occurs on both sides of the market, users and resource providers simultaneously. Next, it needs to provide an effective scheduling environment that must have the ability to offer resources with minimal delay to jobs. In this dissertation, a framework for economic strategy based Grid scheduling is proposed. The framework entities such as users, broker and resources employ Continuous Double Scheduling Algorithm to decide the final values of prices and deadlines. We evaluated the performance of the different strategies of CDS with respect to each other and with the Offer Based Scheduling.

General Terms

Economic Scheduling, Algorithms, Distributed Computing

Keywords

Offer based Scheduling, Grid Computing, Continuous double scheduling

1. INTRODUCTION

In large scale computing systems, there are often large amounts of resources to be used for computing jobs. Since these resources can cost up to thousands of dollars but important problem is to maximize their utilization.

Scheduling in a large computing system as grid computing is not as easy as scheduling on a multi-processor machine because of several factors. These factors include the fact that grid resources are mostly used by paying buyers who have interest in how their jobs are being scheduled. Also, grid computing systems usually operate in remote locations so scheduling jobs for the clusters may be occurring over a network is also a challenging task. [18]. Due to these reasons that looking at scheduling in grid computing is an interesting and important problem to examine.

1.1 Scheduling Categories

Scheduling is divided into two main categories: conventional and economic scheduling. These scheduling categories are discussed below.

1.1.1 Conventional scheduling

Conventional (traditional) scheduling considers the overall performance of a system as a metric for determining the scheduling quality. For example, the time it takes to schedule all jobs (makespan). Additionally, it doesn't take the cost as a factor for scheduling jobs on resources.

1.1.2 Economic scheduling

In economic scheduling, cost is considered an essential factor for scheduling jobs. For a scheduling to take place, the parties must positively value this scheduling. [28,29,30] The participation in the Grid must be cheaper for the users than purchasing their own resources, and must satisfy their requirements. On the other hand, resource providers must know if it is worth to provide their resources for usage by users.

2. RELATED WORK

Many researchers have already worked on many scheduling techniques used in allocation of resources. In reference to [23] they introduced a co-allocation policy for composing resource offers from multiple resources providers to co-allocate a grid user's jobs. When the metascheduler receives offers to meet user requirements, it can decide how to submit the job among the resource providers.

They compared the results with FCFS, EBF, FPFS and simple co-allocation algorithms (SCOAL) and draw the conclusion that co-allocation algorithm presented in paper[23] is better than algorithms FCFS, EBF, FPFS and simple co-allocation (SCOAL). So we decide to work on offer-based scheduling because the comparison shows that this co-allocation algorithm reduce the total time to release user jobs and waiting time in the global queue, maximize the resources utilization rate and load the balance among the resources providers

In reference to [24], they introduced policy for composing resource offers from multiple providers to schedule deadline constrained BoT applications. When the metascheduler receives enough offers to meet user deadlines, it can decide how to balance the tasks among the resource providers according to the information it has access, such as resource providers' total computing power and their local loads. From their experiments they observed that when using offer-based policies, more BoTs can meet deadlines and the delays between the user deadline and the new deadline assigned by the system is much lower (in some cases 50% lower) in comparison to the policy that uses free time slots (Free Time Slots).

3. PROBLEM FORMULATION AND OBJECTIVES

Pricing is important factor because it gives resource providers an incentive to supply their resources to the Grid and also because it enforces the users to use the resources just when they need them because they have to pay for their use. But the main problem is that researchers have worked on this algorithm on the factor of deadline only and totally ignore the concept of cost which can be also be a very effective factor in evaluation criterion of performance of any scheduling algorithm. The other problem is that there is a fixed window size of time in which offers should received to the broker.

1. Our main objective is to work on scheduling the jobs of users to the distributed resources in accordance to the best offer received from number of resources. We start the work with aim of analyzing the performance of Continuous Double Scheduling Algorithm with two constrained parameters i.e. deadline and price. We improve the strategy of broker to improve the performance and try to make it cost beneficial to both the users and resources.

By implementing this scheduling we choose the best resource from the available resources and get cheap and best resource for executing the user's jobs under the constrained deadline and price given by the user. We analyze the benefits of users and resources in terms of price and deadline for the execution of job in comparison to OBS.

2. Our other objective is to compare the performance of continuous double scheduling by considering three different strategies-

- **Deadline strategy**
- **Price strategy**
- **Deadline and Price strategy**

4. PROPOSED WORK

4.1 Sequence Diagram of Continuous Double Scheduling

The sequence diagram of the CDS is shown in Figure 4.1. The following four steps are involved:

1. Resource Providers participate in a CDS pretty much in the same way that they participate in an offer-based scheduling. A user submits a MessageAsk to the broker to indicate a willingness to buy. Broker sends the message MessageCallforproposal to all the resource providers.
2. A resource provider submits a message MessageBid to the broker to indicate a willingness to sell.
3. At the time of receiving an bid message from a resource provider, the broker will query the current status of the CDS.
4. Once a deal is made (the ask matches with a bid), the matching ask and bid will be removed and a contract will be generated. The corresponding resource provider and user will then be informed about the contract.

4.2 Continuous Double Scheduling Algorithm

In continuous double scheduling, there is no clearing-time frame. Bids and asks are continuously received and matched. Trades can occur at any time, i.e. there is continuous matching and clearance. The trades consist of bilateral transactions triggered by an acceptance of the best bid or ask. The matching bid and ask will be removed from the trade to form a

transaction. Many such individual transactions are carried out and trading does not stop as transactions are concluded.

Brokers have their strategies and they aim to maximize their own utilities. The priority of broker is to maximize the profit to users, they generate from trade on behalf of users and resources by the use of continuous double scheduling algorithm. The broker is an entity that receives bid and ask parameter and maintains a lists of the current bids and asks and matches two offers when the bid is higher or equal to the lowest ask. The trade occurs at the average of matching ask and bid prices.

CDS= (r, R, U, ASKS, BIDS, A_{low}, B_{high}) where:

1. **r** is the type of resource scheduled by the CDS.
2. **R = {R1, R2....Rm}** is a finite set of identifiers of resource providers, where m is the number of resource providers.
3. **U = {u1, u2 ...un}** is a finite set of identifiers of users; where n is the number of users.
4. **ASKS = {A1, ..., Ak}** is a finite queue of asks that are the ask parameters submitted by users in ascending order, where k is the number of asks.
5. **BIDS = {B1, ..., Bi}** is a finite queue of bids that are the bid parameters submitted by resource providers in descending order, where i is the number of bids.
6. **A_{low}** is the current lowest ask of ASKS.
7. **B_{high}** is the current highest bid of BIDS.

A CDS consists of the following steps:

1. A CDS starts with r, ASKS, BIDS, A_{low} and B_{high}
2. The following situations occur:
 3. When a Resource Provider submits an ask with value 'A',
 - A. if $A \geq A_{low}$ then the ask is inserted into the appropriate place in ASKS.
 - B. if $B_{high} < A < A_{low}$ then $A_{low} = A$ and A is inserted into the appropriate place in ASKS;
 - C. if $A \leq B_{high}$ then this provider makes a deal at value ($0.5 \times A + 0.5 \times B_{high}$) with the consumer that submitted B_{high}.
 4. When a User submits a bid with value 'B',
 - A. if $B \leq B_{high}$ then B is inserted into the appropriate place in BIDS;
 - B. if $B_{high} < B < A_{low}$ then $B_{high} = B$ and B is inserted into the appropriate place in BIDS;
 - C. if $B \geq A_{low}$ then this user makes a deal at value ($0.5 \times A_{low} + 0.5 \times B$) with the provider that submitted A_{low}.

The three different strategies are based on different parameters. These strategies are designed to check the performance of system by applying three different parameters from user and resource.

4.2.1. Price Strategy

In this strategy, the broker receives ask and bid parameters with price parameter and occurs trade on the basis of price parameter received from users and resources. When user sends bid (price) or resource sends ask (price) to the broker and broker finds the best deal for both user and resource at average value of price ($0.5 \times a + 0.5 \times b$) and informs the outcome of the price to the bidder and the seller when a match is made.

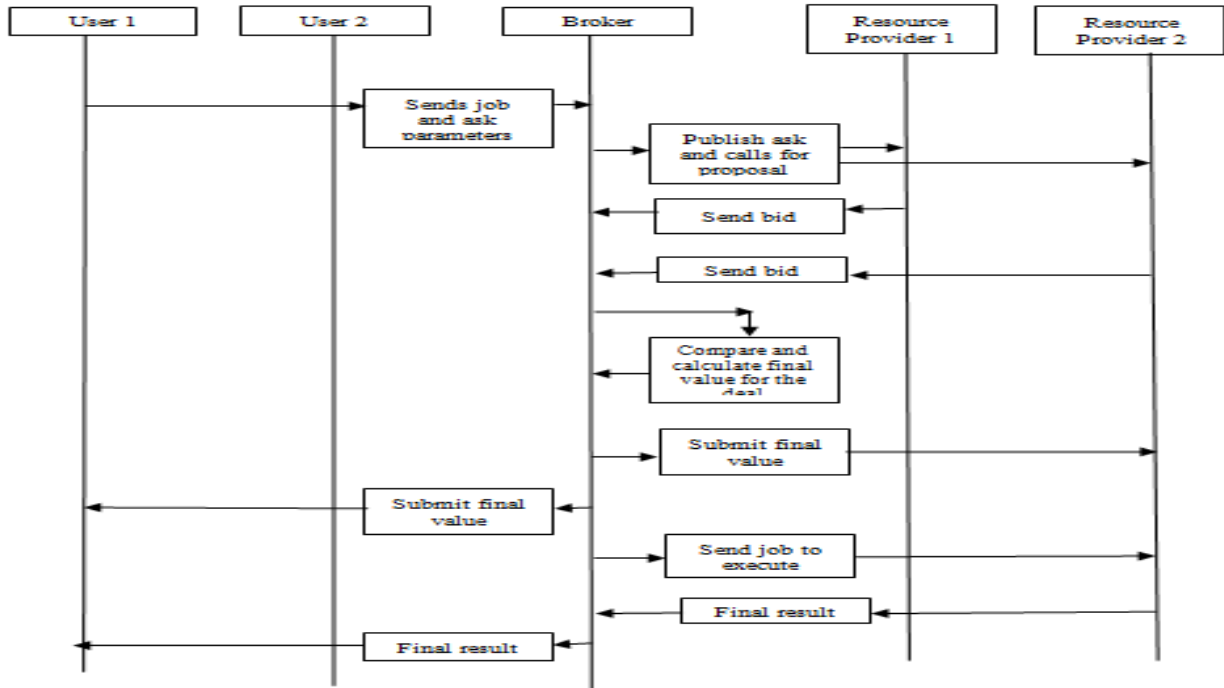


Figure 1: Sequence Diagram of the Continuous Double Scheduling

4.2.2. Deadline Strategy

In this strategy, the broker receives ask and bid parameters with deadline parameter and occurs trade on the basis of deadline parameter received from users and resources. When user sends bid (deadline) or resource sends ask (deadline) to the broker and broker finds the best deal for both user and resource at average value of deadline ($0.5*a+0.5*b$) and informs the outcome of the price to the bidder and the seller when a match is made.

In this strategy, the broker receives ask and bid parameters with price and deadline parameter and occurs trade on the basis of price and deadline parameter received from users and resources. When user sends bid (price and deadline) or resource sends ask (price and deadline) to the broker and broker finds the best deal either according to deadline or price for both user and resource at average value of price ($0.5*a+0.5*b$) and deadline ($0.5*a+0.5*b$) and informs the outcome of the price to the bidder and the seller when a match is made.

4.2.3. Price and Deadline Strategy

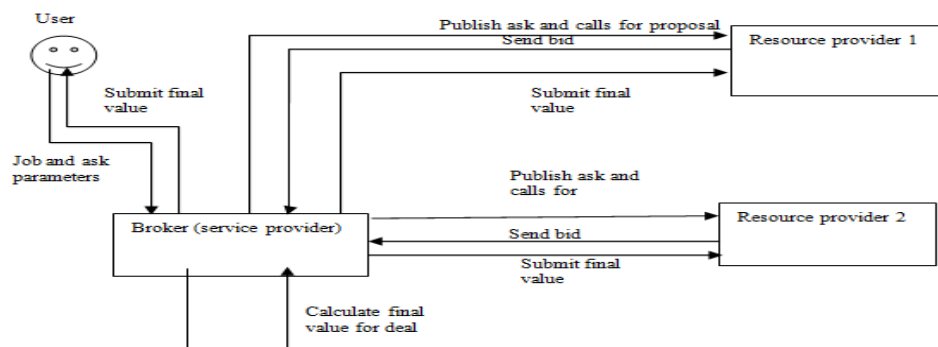


Figure 2: Interaction between the components in CDS

4.3 Simulated Topology

Our simulated topology consists of 2 routers, 5 users and 5 resources along with 1 broker. The simulated network topology is shown in Figure 5.1. Each resource has 10 machines; processing speed of each resource can be from 300 MIPS to 400 MIPS and each machine of resource contains 4 Processing Elements (PE). Each resource is characterized by following parameters:

- Baud Rate of links =1000 bits/sec
- The Propagation Delay=10 millisecond

- Maximum Transfer Unit =1500 byte
- Total Processing Elements =4
- Total Machine =10
- Spaceshared Scheduling Algorithm

In GridSim, total processing capability of a resource's CPU is modeled in the form of its Million Instructions per Second (MIPS) rating as per SPEC (Standard Performance Evaluation Corporation).

There are five users, sharing the same characteristics:

- TotalGridlet = 100
- Gridlet Length = 42000000 MI
- Baud Rate of links =1000 bits/sec
- The Propagation Delay =10 millisecond
- Maximum Transmission Unit = 1500 bytes

To simplify the experiment setup, some parameters are identical for all network elements, such as the Maximum Transfer Unit (MTU) of links is 1,500 bytes. The user and resource uses a Continuous Generator and Sim_uniform_obj distribution for generating the value of price and deadline in CDS between the given minimum and maximum range of values.

In this simulation there are 5 users, 5 resources, and 2 routers. User1, User2, User 3, User 4 and User 5 are connected to Router 1 and Res 1, Res 2, Res 3, Res 4 and Res 5 are connected to Router 2. In this experiment we take up to 100 gridlets allocated to the users. The baud-rate between users and router 1 is 1Mb/sec, between router1 and router 2 is 10Mb/sec and between router2 and resources is 1Mb/sec.

5. ANALYSIS OF SIMULATED RESULT

In the first result as shown in Figure 3, we measure how the users are benefited in execution of their jobs. Two proposed policies (OBS and CDS) are tested in the same situation. It is clearly shown that the jobs sent by the users are completed faster by the resources in CDS compared to OBS. This is possible only because of the better strategy used in broker to select the best resource and make deal at lower value than the OBS. so we can definitely say that CDS as compared to OBS gives better result in time parameter to the users.

Similarly in the second result as shown in Figure 4, we measure how the users are benefited in spending cost to the

resources for the execution of their job. The users profit is increased due to the CDS strategy used in broker as it lowers the value of price by making the better deal with the resources. So it is clearly shown that CDS gives the better result in price parameter also than the OBS to the users.

In the third result as shown in Figure 5, we tried to compare the two different strategies of CDS Algorithm and tested both in same situation. In the graph it is clearly shown that in comparison to price strategy, price and deadline strategy gives more optimal result in terms of cost. It is clearly shown that there is much difference in spending cost by the user in the two simulated strategies. CDS with price and deadline strategy gives more benefit to the user in terms of cost.

In the fourth result as shown in Figure 6, we can see that when the number of jobs increases, the Makespan increases and by comparing the two curves of Figure 6, we see that we have obtained a gain in Makespan by using our CDS algorithm compared to the OBS scheduling algorithms. This is due to the broker strategy of CDS that continuously matches and clears after allocation. But in OBS, the job will remain in the global queue of the Metascheduler, which will delay the job's execution start time if broker does not find the resource to schedule the jobs.

In the fifth result as shown in Figure 7, we tried to show the effect on resource utilization rate of both the policies separately, and we noticed that the resource utilization rate increases in the CDS strategy as compared to OBS strategy. This is due to the broker strategy used in CDS in which there is no clearing-time frame. i.e. trade can occur at any time. The trades consist of bilateral transactions triggered by an acceptance of the best bid or ask. But in the OBS algorithm there is a time boundary and consists only unilateral transactions.

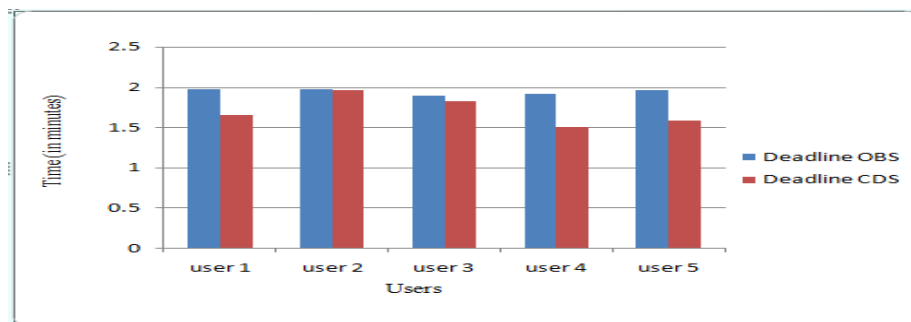


Figure 3: User's Benefit in Deadline

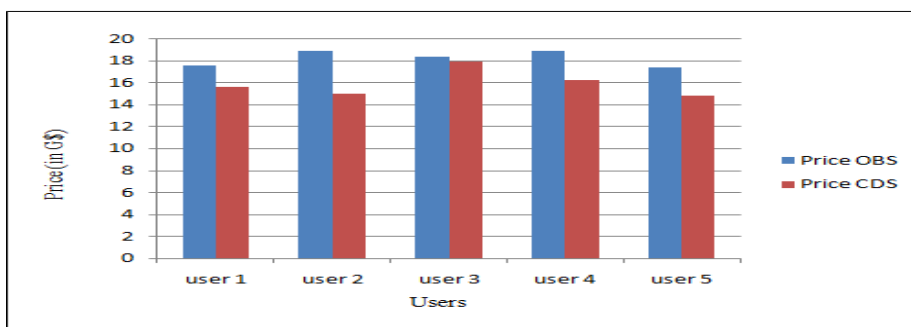


Figure 4: User's Benefit in Price

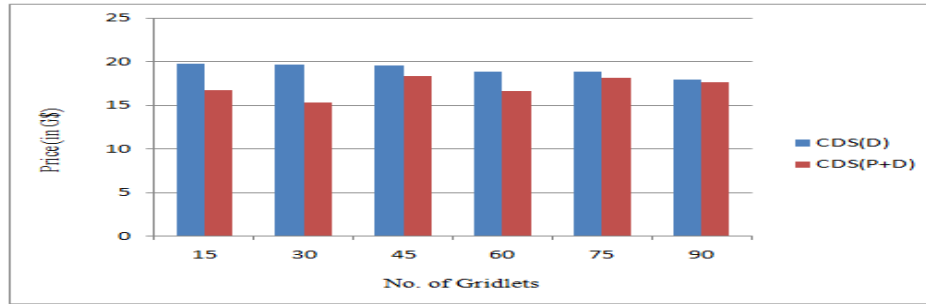


Figure 5: Difference in spending cost by users

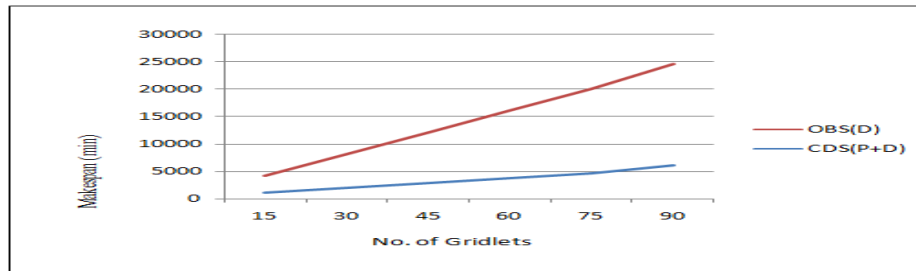


Figure 6: Total makespan to execute all the jobs

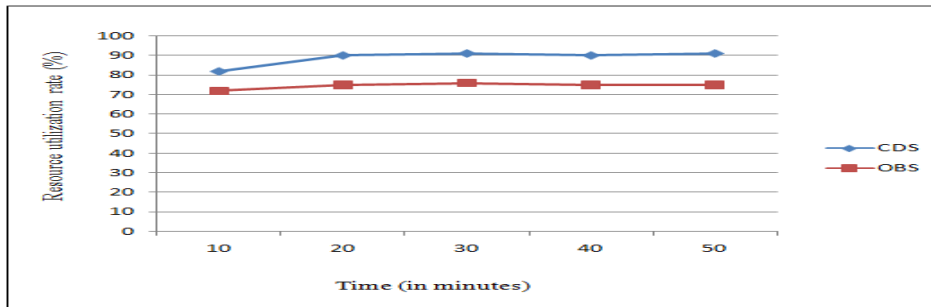


Figure 7: Resource Utilization Rate

6. CONCLUSION

Conventional (traditional) scheduling considers the overall system performance to evaluate the scheduling quality such as utilisation and schedule length. Furthermore, conventional scheduling doesn't consider pricing of resource usage. However, pricing is important because it gives resource providers an incentive to supply their resources to the Grid. Moreover, pricing is important because it enforces the users to utilise the resources just when they need them because they have to pay for their use. Thus, economic scheduling needs to be considered for Grid computing environment. To support this in our dissertation work, a framework for economic scheduling in Grid computing using CDS Method has been developed. In our dissertation work, we evaluated the performance of the different strategies of CDS with respect to each other and with the OBS. The evaluation shows that the price strategy is the best strategy for the user to employ it, if he wants to pay less for executing its jobs. But, the user has to employ deadline or Price-Deadline strategy, if he cares about how long its jobs will take to complete. The broker gives better results if it employs User-Resource average strategy as in CDS, in comparison to broker strategy of OBS. The Results shows that CDS is better than OBS because there is no fixed time window to worry about. Participants in a CDS can offer/acquire resources at the time they want.

7. REFERENCES

- [1] Rajkumar Buyya and srikumar venugopal. A Gentle introduction of grid computing and technologies, 2005.
- [2] R. Buyya and M. Murshed. GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. Concurrency & Computation: Practice and Experience, Dec. 2002.
- [3] Anthony Sulistio, Uros Cibej, Srikumar Venugopal, Borut Robic and Rajkumar Buyya. A Toolkit for Modelling and Simulating Data Grids: An Extension to GridSim, Concurrency and Computation: Practice and Experience (CCPE), vol. 20, Wiley Press, New York, USA, Sep. 2008.
- [4] Anthony Sulistio and Rajkumar Buyya. The GridSim Toolkit, Poster for the 3th International Conference on e-Science and Grid Computing (e-Science'07), Dec. 10-13, 2007.
- [5] R. Buyya. Economic-based distributed resource management and scheduling for grid computing. PhD thesis, Monash university, Melbourne, Australia, 2002.
- [6] H. Chen and M. Maheswaran. Distributed dynamic scheduling of composite tasks on grid computing systems. In IPDPS, 2002.

- [7] Rajkumar Buyya and Anthony Sulistio. Service and Utility Oriented Data Centers and Grid Computing Environments: Challenges and Opportunities for Modeling and Simulation Communities. In Proceeding of the 41st Annual Simulation Symposium (ANSS'08), Ottawa, Canada, Apr. 13-16, 2008.
- [8] Krepska, T. Kielmann, R. Sirvent and R. M. Badia. A service for reliable execution of grid applications. In Achievements in European Research on Grid Systems. Springer Verlag, 2007.
- [9] M. Parashar and C.Lee. Proceeding of IEEE: specific issue on grid computing, volume 93, issue 3, IEEE Press, New York, USA, March 2005.
- [10] Rajiv Ranjan, Rajkumar Buyya and Aaron Harwood. A Case for Decentralized Grid Resource Indexing.
- [11] Schmidt and M. Parashar. Flexible information discovery in decentralized distributed systems. In the Twelfth International Symposium on High Performance Distributed Computing (HPDC-12), June, 2003.
- [12] Iamnitchi and I. Foster. On fully decentralized resource discovery in grid environments. International Workshop on Grid Computing, Denver, CO, 2001.
- [13] Iamnitchi and I. Foster. A peer-to-peer approach to resource location in grid Environments, 2004.
- [14] Czajkowski K., Foster I., Kesselman C., "Resource coallocation in computational grids", Proceedings of Inter Symp on High Performance Distributed Computing, California IEEE Computer Society Press, 219–228, 1999.
- [15] Goswami K. and Gupta A... Resource Selection in Grids using Contract Net. 16th Euromicro Conference on Parallel, Distributed and Network- Based Processing (PDP), Feb. 2008.
- [16] Stylianos Ziko and Helen D. Karatza. Resource Allocation Strategies in a 2-level Hierarchical Grid System", 41st Annual Simulation Symposium, 2007.
- [17] Ravish Mahajan and Arobinda Gupta. Scalable Contract Net Based Resource Allocation Strategies for Grids. Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2008.
- [18] Rajkumar Buyya and Anthony Sulistio. GridSim: Java-based Modelling and Simulation of Computational Economy-based Scheduling for Grid Computing, Poster Exhibit @ CCGrid 2001: The First IEEE/ACM International Symposium on Cluster Computing and the Grid, Brisbane, Australia, May 15-18, 2001.
- [19] Rajkumar Buyya, David Abramson, Jonathan Giddy, An Economy Driven Resource Management Architecture for Global Computational Power Grids, The 2000 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2000), Las Vegas, USA, June 26-29, 2000.
- [20] Sid Ahmed MAKHLOUF (1), Belabbas YAGOUBI, "Co-allocation in Grid Computing using Resources Offers and Advance Reservation Planning" Courier du Savoir, 2012.
- [21] Marco A. S. Netto, "Offer-based scheduling of deadline-constrained Bag-of-Tasks applications for utility computing systems", *IPDPS*, 2009, IEEE International Symposium on Parallel & Distributed Processing (IPDPS), pp. 1-11
- [22] L. Chunlin and L. Layuan. An agent-based approach for grid computing, 2003
- [23] A.Iamnitchi and I. Foster. On fully decentralized resource discovery in grid environments. In International Workshop on Grid Computing, 2001.
- [24] M. Maheswaran and K. Krauter. A parameter-based approach to resource discovery in grid computing systems. In Proceedings of the First International Workshop in Grid Computing, pages 181–190, 2000.
- [25] Berman and R. Wolski. The apples project: A status report. In Proceedings of the 8th NEC Research Symposium,, 1997.
- [26] E. Sutherland. A futures market in computer time. *Commun. ACM*, 11(6):449–451, 1968.
- [27] Nakai. Pricing computing resources: Reading between the lines and beyond. Technical report, National Aeronautics and Space Administration, 2002.
- [28] A.Takefusa, S. Matsuoka, H. Casanova, and F. Berman. A study of dead-line scheduling for client-server systems on the computational grid. In HPDC'01: Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10'01), pages 406–415. IEEE Computer Society, 2001.
- [29] R. Montero, E. Huedo, and I. M. Llorente. Grid Scheduling Infrastructures with the GridWay Metascheduler, 2006.
- [30] S. Venugopal, K. Nadiminti, H. Gibbins, and R. Buyya. Designing a resource broker for heterogeneous grids. *Software-Practice and Experience*, 38(8):793–826, 2008.
- [31] S. Song, K. Hwang, and M. Macwan. Fuzzy trust integration for security enforcement in grid computing. In NPC, pages 9–21, 2004.
- [32] A.Galstyan, K. Czajkowski, and K. Lerman. Resource allocation in the grid using reinforcement learning. In International Conference on Autonomous Agents and Multiagent Systems, 2004.
- [33] M. Bsoul, I. Phillips, and C. Hinde. A framework for economic scheduling in grid computing using tender/contract-net model. In pgnnet, 2006.
- [34] X. He, X. Sun, and G. V. Laszewski. A qos guided scheduling algorithm for grid scheduling, 2003
- [35] F. Howell and R. McNab, SimJava: A Discrete Event Simulation Package For Java With Applications In Computer Systems Modelling, First International Conference on Web-based Modelling and Simulation, San Diego, CA, Society for Computer Simulation, January 1998.