

A Novel Probabilistic Strategy for Delay Corrected Allocation in Shared Resource Systems

G. Arun Kumar¹, Aravind Sundaresan¹, Snehanshu Saha¹, Bidisha Goswami¹, Shakti Mishra²

¹PES Institute of Technology, Department of Computer Science and Engineering, Bangalore South Campus, Bangalore 560100, India

²Sir MVIT, Department of Computer Science and Engineering, Bangalore, Karnataka, India

E-mails: garunk0703@gmail.com sundaresan.aravind2@gmail.com snehanshusaha@pes.edu bidishagoswami@pes.edu mishra_shakti@yahoo.com

Abstract: Cloud computing offers scalable services to the user where computing resources are owned by a cloud provider. The resources are offered to clients on pay-per-use basis. However, since multiple clients share the cloud's resources, they could potentially interfere with each others' task during peak load instances. The environment changes every instant of time with a new set of job requests demanding resource while another set of jobs relieving another set of resources. A major challenge among the service providers is to maintain a balance without compromising Service Level Agreement (SLA). In case of peak load, when each client strives for a particular resource in minimal time, the resource allocation problem becomes more challenging. The important issue is to fulfil the SLA criterion without delaying the resource allocation.

The paper proposes a n -player game-based Machine learning strategy that would forecast outcome using a priori information available and measure/estimate existing parameters such as utilization and delay in an optimal load-balanced paradigm. The simulation validates the conclusion of the theorem by showing that average delay is low and stays in that range as the number of job requests increase. In future, we shall extend this work to multi-resource, multi-user environment.

Keywords: Load balancing, Service Computing, Logistic Regression, Probabilistic Estimation, Machine Learning.

1. Introduction

Cloud Computing is an emerging paradigm that offers enormous resources to clients on demand. Cloud providers rent their resources to multiples clients concurrently and charge their customers on the basis of the resource usage.

This is usually achieved through virtualization of resources. The resources are virtually partitioned and VMs are hosted on the physical servers.

As per the demand, resources are dynamically provisioned to clients. On the basis of application requirement and number of users, too many requests from clients may vary in nature and hence virtual machines of different sizes are required to fulfil these requests. These VMs are spread across data centres on different physical servers to balance the load.

Once the server is selected, the server is probed for accessing the service. Prior to avail such services, a typical business agreement takes place between these parties termed as Service Level Agreement. Service Level Agreement (SLA) is the only term between customer and service provider that governs the conditions to be fulfilled at the time of service delivery. A cloud service provider creates a schedule for the bundle of services as requested by the user. Every provider promises enormous elasticity of resource provisioning. Addressing the resource delivery within the time line as per the metrics in SLA is a major challenge in any dynamic environment. The nature of the network is completely vulnerable and system strategy of servicing is heterogeneous. Resource allocation with an optimally low delay and balanced loading is one of the major challenges in cloud service provisioning.

Several research groups have worked in the domain of load balancing and traffic management. Bio inspired model [12] is one of the most discussed solutions. Paper [4] formulated a problem as “multi facility resource allocation problem”. The objective of their work is to maximize the social welfare cost (utility-actual cost) by distributing n resources to N users. The load balancing and optimization of existing resource is one of the most pursued topics in the area of distributed computing. Paper [12] have proposed a predator-prey relation to model the relationship between demand and supply of resources. There are very popular scheduling schemes in literature such as FCFS, round-robin, Largest job first, etc., including some special scheduling algorithms such as Compromised Time-Cost Scheduling [7], Particle Swarm Optimization-based Heuristic Scheduling [8], Transaction Intensive Cost-constrained scheduling [9], Elastic Scheduling [10], QoS constrained Scheduling [11], and Dynamic Job Grouping based Scheduling [12]. It has been found out that all of these scheduling algorithms suffer from disk space management issue [6]. Also, the existing pricing and scheduling schemes do not take into account the service performance factors. The clients are interested in completing their jobs in the least possible time with the least possible total cost which is the amount of money they pay for the resources used. The cloud provider, on the other hand, is interested in maximizing the resource utilization of the cloud and thus its revenue. These result in unintended client-client and client-provider interactions which are not captured by existing pricing and resource allocation mechanisms [13]. Moreover, the state-space information creates small delay which affects overall service time span. This is the reason why randomization in resource availability might be a better approach. Game theory is one of the frequently used approaches for solving several optimization problems. In grid computing, an agent based cooperative, semi-cooperative and non-cooperative strategy has been examined and

it has been concluded that cooperation among the player is extremely important when multiple factors have to be considered [1] and [2] proposed a cloud bank model with bidding game strategy. They considered the situation of uncertainty of internet availability and credit is assigned based on the quality and constant online availability of services. Authors of [3] argued that cooperation among several cloud providers makes better profitability and optimization by proposing a stochastic linear programming game model. Architecture for spectrum access in small cell network aiming to address heavy computational complexity has been presented in [5]. Dynamic resource scaling in service computing domain is one of the fundamental needs to obtain the optimality. The virtual networking based methodology has been proposed in [16]. This virtual network is composed of distributed virtual router's consumes heavy processing power, memory and bandwidth. The model proposes service level agreement in terms of low penalty as functions of congestion and virtual network request rejection. The paper discussed a new mapping strategy of unused allocated bandwidth with minimal rejection. This is an excellent idea while we consider a dynamic scenario where thousands of VM moves around the network for placement of application. The authors of [17] introduce an approximation algorithm which adjusts the availability factors of partitions with some boundary sequence. Here output shows some improvement in resource utilization using single source partitioning. However, the partition migration issue has not been taken into consideration. The scenario of service computing explores dynamism. The features, elasticity and dynamism need to be added in the service computing platform. Next, an optimized Ant Colony Algorithm is used to list strategy and greedy algorithms. The performance of the algorithm is promising. The algorithm takes care about the multi-user dynamic environment like the target scenario in service computing. Multi resource dynamism is experimented in [19]. This paper proposed a model based on virtualization. This paper focuses on the benefits of resource utilization through virtual machine migration [20]. This paper discussed on dynamic balance of multiple resources for provisioning of response time in virtualized clusters. The authors propose a methodology to balance multiple resources for virtual machines under the constraints of service level objective. The authors propose a quality of service aware framework that dynamically balances multiple virtualized resources based upon application level service level objectives. The proposed framework integrates two separate components, model predictive control and adaptive proportional integral control. The paper [14] proposed an adaptive resource allocation algorithm based on game theory for multi-resource environment. In this work, each physical server providing resource is treated as a game player and knows the utility information of other players. The resources are allocated in a time-slot paradigm. The dynamically arriving user requests of current time slots are recorded and served for resource allocation at the start of next time slot. Each start of a time slot is called a decision moment. In order to support elastic multi-source consumption, this work presents a Fairness Utilization trade off Game Algorithm (FUGA). A new class of games called Cloud Resource Allocation Games (GRAGs) has been proposed in [12]. In this work, resource allocation problem is considered a non-cooperative game where

clients selfishly try to maximize their utility. Also, the various conditions have been determined to achieve Nash equilibrium and Price of Anarchy results in CRAG and Stackelberg CRAG. The authors in [15] modelled cloud as multi-agent system composed of agents (machines) with varied capabilities. The resource requirement that is beyond the capabilities of single agent is accomplished through agent coalitions. This coalition is modelled as a game and exploited the uncertainty principle.

In this paper, we design a new game based strategy to allocate a resource in multi-user environment. In case of peak load, when each client strives for a particular resource in minimal time, the resource allocation problem becomes more challenging. The important issue is to fulfil the SLA criterion without delaying the resource allocation. Hence, we have devised a new approach that is based on creating equivalence between granting a resource and winning a request.

The major contribution is summarized as follows:

- a. To fulfil the SLA specification without compromising the load balance of the entire system.
- b. The client's requests are considered as players and the game strategy is based on the heuristics.
- c. To forecast the outcome of the game and awarding the resource to the winner.

The overall work represents an extension of the existing structure of open-stack. The Remainder of the paper is organized as follows. Section 2 explains the problem statement. Section 3 details the complete workflow analysis and background of the proposed work. Section 4 elaborates the analytical and computational strategy. Section 5 discussed the implementation followed by the results in Section 6.

2. Problem statement

Dynamic resource utilization is one of the major challenges in service computing domain. The environment changes every instant of time with a new set of job requests demanding resource while another set of jobs relieving other set of resources. The challenge is to keep a balance without compromising Service Level Agreement. Maintaining a balance among the available resources is very crucial as imbalanced configuration can lead the physical host in the state, over provisioned or under provisioned. We discussed a game base strategy while bids for resources. The players are qualitative and quantitative representations of the tentative resource. We assume a set of information already available about the qualitative and quantitative parameters of a priori games. The winner of the game would be awarded the resource. We restrict the discussion to single resource multi-player game. The result is promising and may be extended as a multi-resource multi player game in the future.

2.1. Our approach

Our model is inspired by Machine Learning paradigms. Machine learning has emerged as one of the most talked about and used areas in computing exploited to solve several challenging problems in Natural and Physical Sciences [28-30]. Several interesting solutions in Informatics have also been proposed using machine learning [24-27]. Lately, communications and distributed systems have picked up the concept. Hybrid machine learning algorithms are fused with meta-heuristics to improve performance of traffic intensive networks [23]. Solutions in Data centre optimization and congestion avoidance in cloud have also been proposed using machine learning [22]. The game design proposed here is based on machine learning and significantly different from the ones mentioned in the literature. The model is conceived to forecast the probability of the expected winner and casting a solution accordingly. Forecasting the outcome of the game depends on prior information for classification and the primitive that there exists a single fixed resource. The workflow/demands are the players for this game. The approach is to design the game based on previous outcomes. Since the resource is valuable, the proposed model will allocate the resource only to the player who wins the game. The model, designed as a set of classes, contains the information about the components of the RI's (Resource Indicators) and exploits the history of winning the same game. The model is leveraged on winning the resource to be allocated by considering the present scenario of the RI. The model strives to predict the outcome of the game in the process.

2.2. Workflow analysis

The scenario consists of several sets of workflows demanding to possess the same resource. This is conceptualized as a competition among k players bidding in a game. The winner would be granted the resource. The intention of the game is to forecast the winner with *a priori* information available in the training set. The a-priori information is the vector bundle indicating the outcome of the bidding results between the players in the previous n encounters. The game strategy computes the probability of winning or losing the game by extracting the features of the existing training set. The model highlights are as follows:

- The game is designed to grab a single resource. If the resource is allocated, then the player would be the winner, else the player would lose.
- RI is Resource Indicator is representative of an individual player. A random variable is allocated to represent RI. $RI = 1$ if player p_1 wins and
 - $RI = 2$ if player p_2 wins and so on.
- The a priori information about the previous bidding results is represented as a feature vector $X = [x_1, x_2, \dots, x_n]$.

This implies the p players have already played the game n times. X is the vector which contains the information about the outcome of the bidding between two players in the last n encounters. So, $x_3 = 1$ implies player p_1 wins the 3rd bidding game.

- We define a classifier function $\text{clf}(X)$ to calculate the value of RI (to predict the outcome of a future bid for the same resource).

3. Our strategy

The problem consists of p players bidding to be allocated a single fixed resource. The a priori information is given in the form of the results of previous bidding encounters. In this case, the a priori information is a set of 10 results, each denoting a winner of the corresponding bid. The solution proposed in this model is: The a priori information is considered in the form of a feature vector which consists of the results of the last 10 encounters between the p players denoted by X . The classifier defined in the paper is based on a multivariate logistic regression classification model. The output of this classifier is $\text{RI} \in \{1, 2, \dots, p\}$. So $\text{RI} = 1$ if a player p_1 wins, and $\text{RI} = 2$ if the player p_2 wins, and so on.

3.1. Solution scheme

Regression Analysis is one of the most widely used techniques for analysing multi-factor data. The reason behind using Regression Analysis is that it presents an equation that represents the relationship between a variable of interest (response variable) and a set of related predictor variables.

Given a response variable y and predictor variables x_1, x_2, \dots, x_n we define the following relationship:

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n.$$

This is called a multi-variable linear regression model which is parameterized by $\theta_1, \theta_2, \dots, \theta_n$.

In linear regression, the output given in the form of a response variable is continuous. The problem presented here, however, requires the output in the form of a categorical variable. This justifies the need to use Logistic Regression.

Logistic Regression is a statistical method for analyzing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous/binary variable. Logistic regression deals with this problem by using a logarithmic transformation on the outcome variable. It expresses the linear regression equation in logarithmic terms (called *logit*).

Given a binary output variable y , i.e., $y \in 0,1$ (Note: This is a 2-class problem) and $X = [x_1, x_2, \dots, x_n]$ with predictor variables x_1, x_2, \dots, x_n , we model the conditional probability $P(y = 1|X)$, modelled as a function of X , i.e., $p(X)$. Formally, the univariate logistic regression model is

$$\log \frac{p(X)}{1 - p(X)} = \theta_0 + \theta x.$$

Therefore, for the multivariate case we have,

$$\log \frac{p(X)}{1 - p(X)} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n.$$

Solving for p ,

$$p(X) = \frac{1}{1 + \exp\{-(\theta_0 + \sum_{i=1}^n \theta_i x_i)\}}.$$

Based on the above definition the value to output variable y is assigned as follows:

$$(1) \quad y = \begin{cases} 1 & \text{if } p(X) \geq 0.5, \\ 0 & \text{otherwise.} \end{cases}$$

The 2-class problem can be extended to the multi-class problem by using one-vs-all strategy. This involves training a single classifier per class with samples of that class as positive and all the other samples as negative. This implies for k class problem we have probabilities $p_1(X), p_2(X), \dots, p_k(X)$. Each of the aforementioned probabilities can be rewritten concisely as $P(y=i | X, \theta)$, $i=1, 2, \dots, k$. After evaluating the above probability, the predicted class is determined by

$$(2) \quad y = \arg \max_i P(y=i | X, \theta).$$

This is the mathematical formulation used to build the classifier for our scenario.

4. Implementation

Consider k players, p_1, p_2, \dots, p_k , competing for a single resource. The a priori information about the results of the previous n encounters is represented as a feature vector, $X = [x_1, x_2, \dots, x_n]$, where each $x_i \in \{1, 2, \dots, k\}$ and $x_i = j$ implies player p_j had won the previous i -th bidding.

This k -class classification problem uses the one-vs-all strategy as defined earlier. The winner of the current workflow, represented by RI, is determined by the following rule,

$$(3) \quad \text{RI} = \text{clf}(X) = \arg \max_i h_{\theta}^{(i)}(X),$$

where

$$(4) \quad h_{\theta}^{(i)}(X) = P(\text{RI} = i | X, \theta).$$

That equation is determining the binary output variable denoting the winner of the future bid. It is the same as Equation (2). The equation is rewritten for brevity. The hypothesis $h_{\theta}^{(i)}(X)$ gives the probability of the player p_i winning the current workflow. Therefore, for a k -class problem there would be k -hypothesis. Equation (4) is parameterized by

$$(5) \quad \theta^{(i)} = [\theta_0, \theta_1, \theta_2, \dots, \theta_n].$$

4.1. Parameter estimation

Given a hypothesis to estimate, we define a cost function as

$$(6) \quad J^{(i)}(\theta) = \begin{cases} -\log(h_{\theta}^{(i)}(X)), & \text{when } \text{RI} = i, \\ -\log(1 - h_{\theta}^{(i)}(X)), & \text{when } \text{RI} \neq i. \end{cases}$$

Consider m samples of training data, denoted by

$$[(X_1, \text{RI}_1), (X_2, \text{RI}_2), \dots, (X_m, \text{RI}_m)].$$

For a two-class problem Equation (6) can be modified as

$$(7) \quad j^{(i)}(\theta) = -\frac{1}{m} [\sum_{j=1}^m \text{RI}_j \log(h_{\theta}^{(i)}(X_j)) + (1 - \text{RI}_j)(1 - \log(h_{\theta}^{(i)}(X_j)))].$$

Optimal $\theta^{(i)}$ is determined by $\theta^{(i)} = \arg \min_{\theta} J^{(i)}(\theta)$, which completely defines

the hypotheses $h_{\theta}^{(i)}(X)$.

In order to improve the accuracy of the classifier/hypothesis, the following transformation of feature vector X is performed. Given a hypothesis $h_{\theta}^{(i)}(X)$ for player i , the new feature vector is defined as

$$(8) \quad \hat{X} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n],$$

where

$$\hat{x}_j = \begin{cases} 1, & \text{when } x_j = i, \\ 0, & \text{when } x_j \neq i. \end{cases}$$

4.2. Prediction

After the training phase, the parameters $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(k)}$ are used by the classifier function $\text{clf}(X)$ to assign the workflow RI for an unlabelled sample X with the feature transformation performed as explained above.

5. Result and discussion

The classifier described above was trained with the following parameters:

$p = 4$, number of players competing,

$m = 1000$, training set size,

$n = 10$, number of previous encounters considered, i.e., *a priori* information.

After training the classifier, the problem was simulated with $n = 1000, 5000, 10000$, where n is the number of job requests. The output of the classifier is a vector of labels, where each label denotes the winner of the current bidding encounter.

Comparing the output of the classifier with the actual class labels of the job requests, we obtain the accuracy of the classifier. This denotes how reliable our classifier is in predicting the outcome of future job requests. The following accuracy was achieved after the simulation:

- For $n = 1000$ Accuracy = 98.5%,
- For $n = 5000$, Accuracy = 98.58%,

- For $n = 10\,000$, Accuracy = 98.55%.

The output of the classifier for a test case denotes the number of victories attained by each workflow. This information is then used to compute the resource utilization. Resource utilization is computed to determine the amount of resource required by each workflow competing for it. The values of resource utilization tabulated below depict the percentage of the resource that must be allocated by the vendor for each of the workflows requesting the resource. The resource utilization results are tabulated in Table 1, and also shown in Fig. 1.

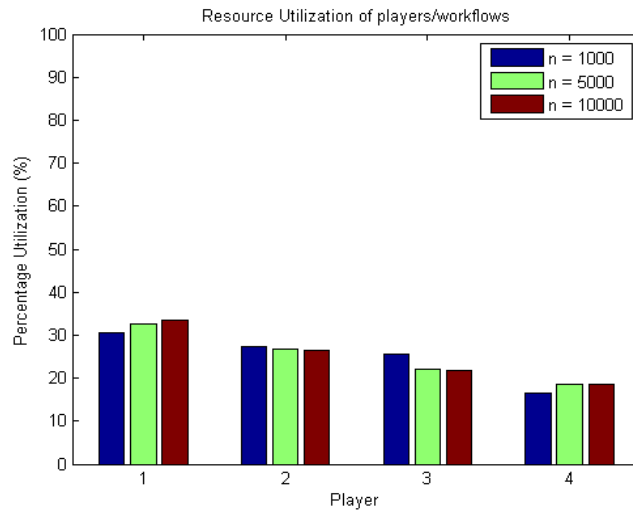


Fig. 1. Resource utilization of players/workflows

The simulation results are tabulated in Table 1.

Table 1. Resource utilization for each workflow

n	Resource Utilization, %			
	p_1	p_2	p_3	p_4
1000	30.70	27.20	25.60	16.50
5000	32.62	26.62	22.14	18.62
10000	33.49	26.35	21.66	18.50

The resource utilization obtained for each workflow can be used by the service provider to formulate the Service Level Agreement (SLA) as it gives a rough estimate of the actual requirement of the resource by a customer.

There are existing scheduling algorithms like FCFS, Round-Robin and SJF that are used to manage resource allocation. An issue with these algorithms is that they require state-space information which creates a small delay which affects the overall time span. Our probabilistic approach eliminates the need for such prior information and henceforth the delay incurred is significantly reduced. Time delay comparisons of our model were made with FCFS, Round-Robin and SJF with test set size $n = 100, 500, 1000, 5000, 10\,000, 50\,000, 100\,000$ and the results obtained are tabulated in Table 2 and Table 3, and also shown in Fig. 2 and Fig. 3.

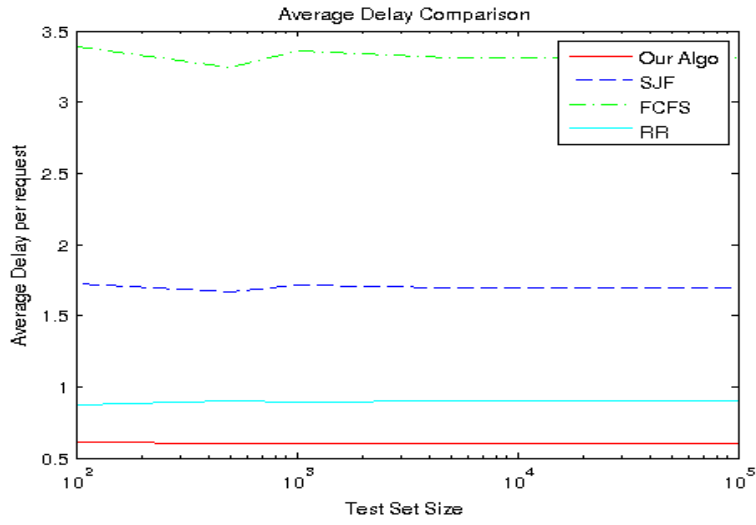


Fig. 2. Average time delay per job request

Table 2. Average time delay per job request

n	Time, ms			
	Ours	SJF	FCFS	RR
100	0.5840	1.5980	3.0990	0.8340
500	0.5980	1.6898	3.2947	0.8859
1000	0.6106	1.7271	3.3773	0.9090
5000	0.6048	1.7139	3.3466	0.9068
10 000	0.5961	1.6933	3.3024	0.8956
50 000	0.5998	1.6951	3.3060	0.8993
100 000	0.6019	1.7002	3.3165	0.9022

The model proposed in the paper certainly outperforms the existing algorithms by not using the state-space information and using a probabilistic approach for allocating resources.

Table 3. Total time delay per job request

n	Time, ms $\times 10^5$			
	Ours	SJF	FCFS	RR
100	0.0006	0.0016	0.0031	0.0008
500	0.0030	0.0084	0.0165	0.0044
1000	0.0061	0.0173	0.0338	0.0091
5000	0.0302	0.0857	0.1673	0.0453
10 000	0.0596	0.1693	0.3302	0.0896
50 000	0.2999	0.8476	1.6530	0.4496
100 000	0.6019	1.7002	3.3165	0.9022

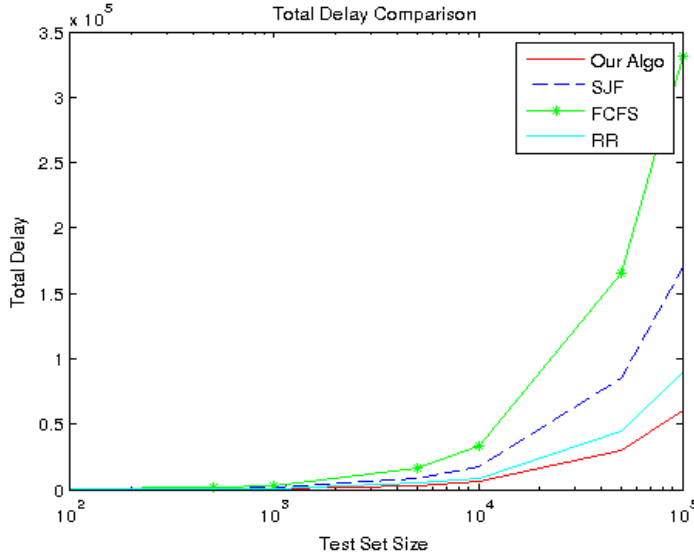


Fig. 3. Total time delay per test set size

We represent the average delay per Job Request by a non-negative random variable D . The value D_{\max} is defined which denotes a threshold on the random variable D .

We propose

Delay Estimation Theorem.

$$(9) \quad P(D \geq D_{\max}) \leq \frac{E[D]}{D_{\max}}, \text{ when } D \geq 0, D_{\max} \geq 0.$$

Proof: Define a probability space

$$A = \{d \in \Omega \mid D(d) \geq D_{\max}\},$$

where $\Omega \equiv$ Probability space that contains A .

The mean value of D ,

$$(10) \quad E[D] = \sum_{d \in \Omega} P(d)D(d),$$

$$E[D] = \sum_{d \in A} P(d)D(d) + \sum_{d \notin A} P(d)D(d).$$

The second term $\sum_{d \notin A} P(d)D(d) \geq 0$ by definition, therefore Equation (10)

becomes to

$$(11) \quad E[D] \geq \sum_{d \in A} P(d).D(d).$$

We know that

$$(12) \quad E[D] \geq D_{\max} \sum_{d \in A} P(d) = D_{\max} P(D \geq D_{\max}),$$

therefore Equation (12) becomes to

$$(13) \quad P(D \geq D_{\max}) \leq \frac{E[D]}{D_{\max}}.$$

If $D_{\max} \geq D$, then $P(D \geq D_{\max})$ will be bounded by a small number. Hence the system will function within an acceptable delay. In fact, this band may be further tightened upon accumulation of considerable historical data.

Overall delay is increasing in piecewise linear fashion, making the problem tractable and the delay of finite bounded variation.

Mean average delay per job request is

$$E[D] = \frac{(0.5840 + 0.5980 + 0.6105 + 0.6048 + 0.5961 + 0.5998 + 0.6019)}{7} \approx 0.6 \text{ ms.}$$

Therefore, Equation (9) becomes to

$$P(D \geq 1 \text{ s}) \leq \frac{0.6 \times 10^{-3}}{1} = 0.6 \times 10^{-3}.$$

We conclude that the probability of average time delay per job request being greater than $D_{\max} = 1 \text{ s}$ is 0.6×10^{-3} which is very small.

The average delay estimation theorem concludes that delay being greater than a threshold value is minimal implying that the average delay is finitely bounded above. The simulation validates the conclusion of the theorem by showing that average delay is low and stays in that range as the amount of job requests increase.

6. Conclusion and future work

In this paper, we have introduced a single resource multi-user game that forecasts the probability of the expected game winner. The demands of clients are equivalent to players in the games and the forecasting of the outcome of the game depends upon the a priori information about the games. The a priori information is given in the form of the results of previous bidding encounters. The resource will be granted to the player who wins the game. The a priori information is taken in the form of a feature vector which consists of the results of the last 10 encounters between the players denoted by the classifier. The strategy defined in the paper is based on a multivariate logistic regression classification model. The classifier training has been carried out with 4 players, and 10 previous encounters with training set size as 1000. After initial training of the classifier, we have simulated the problem with $n = 1000, 5000$ and $10\ 000$ and found the accuracy as good as 98.5%. Also, we have calculated resource utilization for each workflow, the average time delay per job request and total time delay per test set size. The simulation validates the conclusion of the theorem by showing that average delay is low and stays in that range as the number of job requests increase.

In the future, we shall extend this work to multi-resource, multi-user environment. Another possible dimension that can be added to the problem is the heterogeneity of the job requests. Multiple random variables can be defined to represent different kinds of job requests.

References

1. Khan, S. U., I. Ahmad. Non-Cooperative, Semi-Cooperative, and Cooperative Games-Based Grid Resource Allocation. – In: Proc. of 20th International Parallel and Distributed Processing Symposium IPDPS'2006, 2006.
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=163938>
2. Niyato, D., A. V. Vasilakos, Z. Kun. Resource and Revenue Sharing with Coalition Formation of Cloud Providers: Game Theoretic Approach. – In: Proc. of 11th IEEE/ACM International Symposium on Cluster, Grid and Cloud Computing, May 2011, pp. 215-224.
<http://dx.doi.org/10.1109/CCGrid.2011.30>
3. Hong, M. An Alternating Direction Method Approach to Cloud Traffic Management. December 2014.
<http://arxiv.org/pdf/1407.8309.pdf>
4. Xu, Y., J. Chen, J. Wang, Y. Xu, Q. Wu, A. Anpalagan. Centralized-Distributed Spectrum Access for Small Cell Networks: A Cloud-Based Game Solution. February 2015.
<http://arxiv.org/pdf/1502.06670.pdf>
5. Bala, A., I. Chena. A Survey of Various Scheduling Algorithms in Cloud Environment. – International Journal of Engineering Inventions, Vol. 1, 2011, No 2.
<http://www.ijejournal.com/papers/v1i2/F0123639.pdf>
6. Liu, K., Y. Yang, J. Chen, X. Liu, D. Yuan, H. Jin. A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost-Constrained Workflows on Cloud Computing Platform. – International Journal of High Performance Computing Applications, Vol. 24, May 2010, No 4, pp. 445-456.
7. Pandey, S., L. Wu, S. M. Guru, R. Buyya. A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments. – In: Proc. of 24th IEEE International Conference on Advanced Information Networking and Applications (AINA'2010), April 2010, pp. 400-407.
8. Yang, Y., K. Liu, J. Chen, X. Liu, D. Yuan, H. Jin. An Algorithm in SwinDeW-C for Scheduling Transaction Intensive Cost-Constrained Cloud Workflows. – In: Proc. of 4th IEEE International Conference on e-Science, Indianapolis, USA, December 2008, pp. 374-375.
9. Lin, C., S. Lu. Scheduling Scientific Workflows Elastically for Cloud Computing. – In: Proc. of IEEE 4th International Conference on Cloud Computing, July 2011, pp. 746-747.
10. Xu, M., L. Cui, H. Wang, Y. Bi. A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing. – In: Proc. of IEEE International Symposium on Parallel and Distributed Processing, August 2009, pp. 629-634.
11. Venticinque, S., R. Aversa, B. Di Martino, M. Rak, D. Petcu. A Cloud Agency for SLA Negotiation and Management. – In: M. R. Guarracino et al., Eds. Parallel Processing Workshops, Euro-Par'2010. – Lecture Notes in Computer Science. Vol. 6586. Berlin, Heidelberg, Springer, 2011.
12. Goswami, B., S. Saha. Article: Resource Allocation Modeling in Abstraction Using Predator-Prey Dynamics: A Qualitative Analysis. – International Journal of Computer Applications, Vol. 61, 6-13 January 2013, No 6.
13. Dressler, F., O. B. Akan. A Survey on Bio-Inspired Networking. – Journal of Computer Networks, Elsevier, Vol. 54, 2010, No 6, pp. 281-290.
14. Jalaparti, V., G. Nguyen, I. Gupta, M. Caeser. Cloud Resource Allocation Games. Technical Report, University of Illinois.
<http://hdl.handle.net/2142/17427>
15. Xu, X., H. Yu. A Game Theory Approach to Fair and Efficient Resource Allocation in Cloud Computing. – Mathematical Problems in Engineering, Vol. 2014, 2014, Article ID 915878, 14 pages. doi:10.1155/2014/915878.
16. Pillai, P. S., S. Rao. Resource Allocation in Cloud Computing Using the Uncertainty Principle of Game Theory. – IEEE System Journal, Vol. PP, May 2014, No 99, pp. 1-12.
17. Li, Y., A. M. K. Cheng. Static Approximation Algorithms for Regularity-Based Resource Partitioning. – In: 33rd Real-Time Systems Symposium (RTSS'12), 2012, IEEE, pp. 137-148.

18. Rajkumar, S. R. Resource Optimization Using Virtual Machine Swapping Circuits, Power and Computing Technologies. – In: International Conferences on Circuits, Power and Computing Technologies (ICCPCT'13), 2013.
19. Wei, Y., C.-Z. Xu. Dynamic Balance Configuration of Multi Resource in Virtual Cluster. – In: Proc. of 21st IEEE International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS'13), 2013, pp. 60-69.
20. Yu, H., W. Shi, T. Bai. An Open-Stack Based Resource Optimization Scheduling Framework. – In: 6th International Symposium on Computational Intelligence and Design, 2013, IEEE.
21. Saha, S., J. Sarkar, M. N. Anand, A. Dwivedi, N. Dwivedi, R. Roy, S. Rao. A Novel Revenue Optimization Model to Address the Operation and Maintenance Cost of a Data Center. – Journal of Cloud Computing: Advances, Systems and Applications, Springer, Vol. 5, 2016, Issue 1, pp 1-23.
22. Sarasvathi, V. N., S. N. Iyengar, S. Saha. QoS Guaranteed Intelligent Routing Using Hybrid PSO-GA in Wireless Mesh Networks. – Cybernetics and Information Technologies, Vol. 15, 2015, No 1, pp 69-83.
23. Mohanchandra, K., S. Saha, S. K. Murthy, G. M. Lingaraju. Distinct Adoption of k-Nearest Neighbor and Support Vector Machine in Classifying EEG Signals of Mental Tasks. – International Journal of Intelligent Engineering Informatics, Vol. 3, 2015, No 3.
24. Mohanchandra, K., S. Saha, G. M. Lingaraju. EEG Based Brain Computer Interface for Speech Communication: Principles and Applications. – In: Brain-Computer Interfaces: Springer International Publishing, 2015, pp. 273-293.
25. Kumar, D., S. K. Meher. Granular Neural Networks Models with Class-Belonging Granulation. – In: Proc. of International IEEE Conference on Contemporary Computing and Informatics (IC3I'14), 2014, pp. 1198-1202.
26. Mohanchandra, K., S. Saha. Optimal Channel Selection for Robust EEG Single-Trial Analysis. – AASRI Procedia, Vol. 9, 2014, Elsevier, pp. 64-71.
27. Bora, K., S. Saha, S. Agrawal, M. Safonova, S. Routh, A. Narasimhamurthy. CD-HPF: New Habitability Score via Data Analytic Modeling Elsevier. – Astronomy and Computing, Vol. 17, 2016, pp. 129-143.
28. Khaidem, L., S. Saha, S. R. Dey. Predicting the Direction of Stock Market Prices Using Random Forest arXiv Preprint arXiv:1605.00003.
29. Saha, S., N. Jangid, A. Mathur, A. M. Narasimhamurthy. DSRS: Estimation and Forecasting of Journal Influence in the Science and Technology Domain via a Lightweight Quantitative Approach COLLNET. – Journal of Scientometrics and Information Management, Vol. 10, 2016, No 1, pp. 41-70.