DOI: http://dx.doi.org/10.26483/ijarcs.v9i3.6007

ISSN No. 0976-5697

Volume 9, No. 3, May-June 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

TIME WINDOW BASED AUTO-REGRESSIVE HYBRID PSO FOR OPTIMAL CLOUD PACKAGE SELECTION

K.Mani
Associate Professor, Department of Computer Science,
Nehru Memorial College,
Tiruchirappalli, Puthanampatti-621 007, India

R.Mohana Krishnan
Research Scholar, Department of Computer Science,
Nehru Memorial College
Tiruchirappalli, Puthanampatti-621 007, India

Abstract: Rapid expansion of cloud technologies were mainly due to the increased requirements of cloud users. However, increased requests also laden with increased resource requirements especially due to the elastic nature of the cloud. This mandates the need for effective resource provisioning model. This paper presents a Time Window based Auto-Regressive Hybrid PSO (TWARP) model that provides faster and more appropriate resource allocations. The TWARP model is composed of a temporal data grouping model to create training data, an auto-regression model to predict future requirements, a PSO-SA based optimal package selection mechanism and a final request handling mechanism that allocates the actual resource to a user. Experiments indicate low time requirements and effective allocation levels. Comparison with recent literature works also indicates highly effective performances of the proposed model.

Keywords: Cloud Provisioning; Auto-regression; PSO; Simulated Annealing; Package selection

I. INTRODUCTION

Cloud technologies exhibit rapid improvements in the past decades due to the high adoption levels opted by customers. Several commercial cloud providers have emerged providing various solutions like PaaS, IaaS, SaaS and so on. The reason behind this is that several enterprises and commercial organizations have opted for cloud based services rather than opting for a dedicated infrastructure [1]. However, this mode of operation tends to create a huge expectation from the provider's stand point. This is due to the fact that cloud resources are elastic in nature [2]. The cloud providers are unaware of the accurate requirements due to the auto upscaling facilities available in cloud as a part of the architecture [3]. Effective resource management is needed to avoid overutilization and underutilization of resources [4].

The major goal of cloud environments is to satisfy maximum number of resource requests. Higher amount of requests satisfied automatically leads to higher user satisfaction. High satisfaction levels leads to high reliability levels and hence low churn rates. To enable this scenario, cloud providers should meticulously handle the available resources and the ways they are provisioned. Resource provisioning plays a vital role in enabling higher reliability levels. Major goal is to provide an effective tradeoff to enable high transaction rates and low turnaround times.

This paper proposes an effective Time Window based Auto-Regressive Hybrid PSO (TWARP) to initially predict future requirements, enabling the providers to be request ready. Time window based training data selection enables effective handling of periodical data patterns. PSO-SA based prediction model enables effective resource selections. Package predictions are performed prior to the user's actual request, hence provisioning takes very low time, as it eliminates the need for resource selection. The only time requirement is to transfer resources to the customer. Experimental results indicate low time requirements and most optimal resource allocations in terms of QoS levels.

II. RELATED WORKS

Resource provisioning is one of the major requirements of a cloud provisioning system [5]. Further, dynamic resource provisioning is also expected without human intervention [6]. This section presents some of the recent contributions in literature in the domain of resource provisioning.

An algorithm handling data intensive applications was proposed by Toosi et al. in [7]. This is considered to be a data-aware provisioning algorithm, proposed to meet the deadline constraints of the users. The model also proposes to handle the deadline requirements of the user, and operates on Aneka platform. A generic resource provisioning model was proposed by Arani et al. in [8]. It is based on the MAPE architecture, however, its impact on network latency, bandwidth and data location have not been analyzed. A hybrid high performance based computing model was proposed by Mateescu et al. in [9]. This model is developed for performing scientific computations. The major contribution of this model is that it provides an elastic cluster that provides an expandable cloud resource environment. A failure aware resource provisioning model was proposed by Javadi et al. [10]. This model aims to effectively handle failures in hybrid cloud environments. A privacy aware model, aimed to provide location awareness was proposed by Xu et al. in [11]. This model proposes a tagging mechanism for location aware data, enabling effective provisioning that provides low data transfer time. An event driven framework to enable QoS guaranteed resource provisioning was proposed by Xu et al. in [12]. It abstracts the cloud based computationally intensive MapReduce computations as a dynamic optimization problem and proposes an event driven solution model as a solution by proposing two resource scaling algorithms. Other similar models operating to

improve execution time include works by Ibrahim et al. [13] and Lin et al. [14], models that improves fault tolerance includes models by Tang et al. [15] and Wang et al. [16] and models that aim to handle deadline constraints are by Anyanwu et al. [17], Polo et al. [18] and W.Zhang et al. [19].

III. TIME WINDOW BASED AUTO-REGRESSIVE HYBRID PSO (TWARP)

Selecting the appropriate package from the list of packages available with the cloud service providers is complex. The complexity is induced due to the varying user requirements that do not really confirm to the specified requirements and the availability of large number of packages with the service provider. Although initial requirements define the packages that are to be selected, a drift could be observed in the usage levels in due course of time. Although increase in requirements are handled by upscaling in cloud environments, these drifts occur in long terms and not as short bursts, hence require upscaling for long periods leading to monetary losses in the part of customer and inability to predict resource requirements in the part of service provider. This paper presents a Time Window based Auto-Regressive Hybrid PSO (TWARP) package selection model that effectively captures drifts to provide resource predictions for effective resource management. The proposed TWARP model is composed of four major phases namely; time based data grouping for training, auto-regression based requirement prediction, package selection using PSO-SA and the final package selection.

A. Temporal Data Grouping

Initial input requirements are usually provided by the customers. Packages pertaining to these initial requirements are assigned for the customers. However, no constraints are imposed on the resource utilization levels, hence the customers can use any amount of resources required, which is the major advantage of migrating to cloud environments. However, this becomes challenging for the cloud resource providers. The providers are required to maintain backup resources to cater to the dynamic upscale requirements. But the providers are oblivious towards the level of scaling required.

This work uses a temporal grouping mechanism that can effectively identify the future resource requirements of a user. Temporally grouping the data and maintaining a time window can clearly reveal the repeating patterns and evolutions occurring in the resource requirement levels. This phase maintains a sliding time window and records within the time window are considered as the training data for the prediction model. Shifting the window continually results in capturing the recent patterns effectively. This work uses a window size of 12 months, so as to enable effective capture of the seasonal changes.

B. Requirement Prediction using Auto-Regression

Auto-regression is a time series based model that predicts by utilizing observations from previous time as an input to the current prediction [20]. Auto regressive models are linearly dependent on their own previous values. The process of auto regression is given by

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

Where φ_i is the model parameter, *c* is the constant and ε_t is the white noise.

Auto-regression can be effectively used in areas of time-series based analysis [21]. This work utilizes auto-regression to predict the QoS requirements of the next request.

Data contained in the temporal window is used as the training instances. The time window is constrained to records for 12 months. Every record contains resource request levels for a single month. The data is composed of 14 features. Features and their descriptions are provided in table 1.

TABLE 1: QoS Parameters Considered for Analysis

QoS Parameters Considered for Evaluation	*Formula*
Bandwidth (Bw)	*Bandwidth (Bw)=Bits/ second (B/S)*
Computation Capability (CC)	*Computation Capability (CC)=Actual Usage time of the Resource/Expected Usage time of the Resource*
Availability (Av)	*Availability (Av)=mean time to failure/(mean time to failure + mean time to repair)*
Correctness (Cr)	*Correctness (Cr) =total number of failed transmissions/ (total number of failed transmissions + total number of successful transmissions)*
Usability (Us)	*Usability (Us) =no of successful operations in a workload/ (total operations available in the workload)*
Reliability (Re)	*Reliability (Re)= mean time to failure + mean time to repair*
Variable computation load (Vc)	-
Serviceability (Se)	*Serviceability (Se) = Service Uptime/*

	(Service Uptime+ Service Downtime)
Latency (l)	*Latency (l) = Time of output produced with respect to that Cloud workload - Time of input in a Cloud workload*
Security (S)	-
Portability (P)	-
Reliable storage (Rs)	-
Data Backup (Db)	-
Customization (Cu)	-

The 14 dimensional data, along with the temporal dimension is passed to the auto-regression model. Prediction P_{dt} for each dimension *d* for time *t* is given by

$$P_{dt} = c + \sum_{i=1}^{n} \varphi_i X_{dt-i} \qquad \forall\, d = 1,2,\ldots m$$

Where *n* is the number of instances and *m* is the total number of dimensions.

The prediction constitutes the level of changes that would have probably occurred in the course of time in the input requests. This, when integrated with the most recent transaction provides the prediction for the next request. This is given by

$$X_{dt} = X_{dt-1} + P_{dt} \qquad \forall\, d = 1,2,\ldots m$$

Where X_{dt-1} is the most recent resource request.

C. *Multiple Package Selection using PSO-SA*

Particle Swarm Optimization (PSO) is a metaheuristic swarm based model operating based on the movement of agents over the search space [22, 23]. The agents are termed as particles. Data pertaining to packages are used to construct the swarm. The particles are then distributed in the swarm in-random and their particle best (*pbest*) and global best (*gbest*) values are determined by their current positions. A random initial velocity is assigned to the particles using the eqn. (1)

$$V_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$$

where b_{up} and b_{lo} are the upper and lower bounds of the search space.

This initiates the particle acceleration. Current position of the particles combined with the velocity determines the magnitude and direction of movement of the particles. The particles are designed to move in a continuous space. However, the current problem requires discrete solutions. Hence the continuous movement is discretized to a nearby node. Fitness of each particle is identified and it corresponds to the *pbest.*

Particle fitness is identified by the absolute difference between the predicted QoS and the QoS pertaining to the current particle. QoS of a resource is calculated using the below equation

$$X_{QoS} = Ns_{bw} + Ns_{cc} + Ns_{Av} + Ns_{Cr} + Ns_{Us} + Ns_{Re} + Ns_{Vc} + Ns_{Se} + Ns_S + Ns_P + Ns_{Rs} + Ns_{Db} + Ns_{Cu} + (-Ns_L)$$

Where Ns_x is the normalized value of the resource parameter.

The existing *pbest* values are used to identify the *gbest.* In regular PSO, a local search is performed for this process. However, it was identified that this mechanism is greedy and leads to the particles getting stuck in local optima. Hence the proposed model replaces the regular local search with Simulated Annealing (SA) based local search.

Simulated Annealing [24] is another metaheuristic algorithm that exhibits faster convergence levels. All the *pbest* values are passed to SA, and the optimal value provided by SA is identified to be the potential *gbest.* The derived value from SA replaces the existing *gbest* if it exhibits a better fitness. The process of particle movement and *gbest* identification is continued till stagnation and the *gbest* value existing during stagnation is considered as the optimal package for the current requirement.

This process is repeated and multiple packages are selected. The number of packages to be selected is user and domain dependent. QoS pertaining to each of these packages is identified and a package list is created.

D. *Requirement Analysis and Final Package Selection*

The previous three phases are performed prior to the actual requirement request. When the actual requirement arises, the QoS pertaining to it is identified and is compared with the QoS pertaining to the predicted requirement. The absolute difference is identified and if it exceeds the threshold boundary, PSO-SA is used to identify the optimal package. If the difference falls within the boundary, the package list is analyzed for the best matching package and it is recommended to the user.

IV. RESULTS AND DISCUSSION

Experiments were conducted by implementing PSO-SA in C#.NET. Package dataset used in [25] is used for analysis. The request data is composed of data pertaining to 4 years of request. Requirement for first year is used as the initial time window. As time progresses, the time window are shifted representing the proceeding months. Results pertaining to optimization were analyzed in terms of time taken for processing and in terms of QoS values and the results pertaining to predictions were analyzed in terms of RMSE and MAE.

A comparison between the requested and provided QoS levels is shown in figure 1. It could be observed that in most requirements, the difference between the provided QoS is almost equal to the required QoS. This indicates the efficiency of the prediction model.

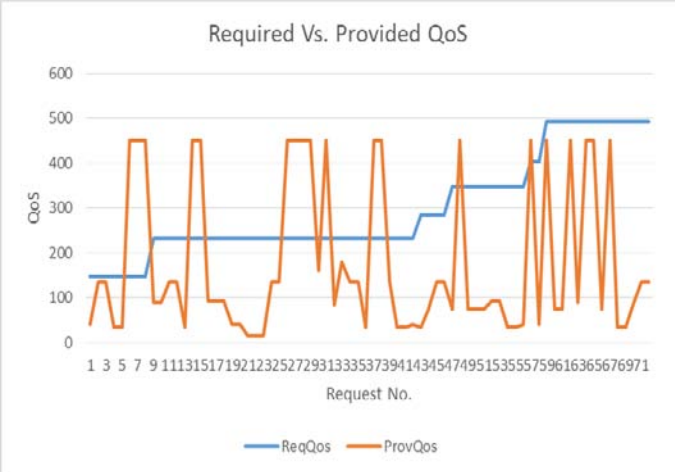

Figure 1: Comparison between Required and Provided QoS

QoS difference between the provided and the required levels is shown in figure 2. It could be observed that except for a few requests, most of the requests exhibit very low QoS difference, hence depicting effective allocations.

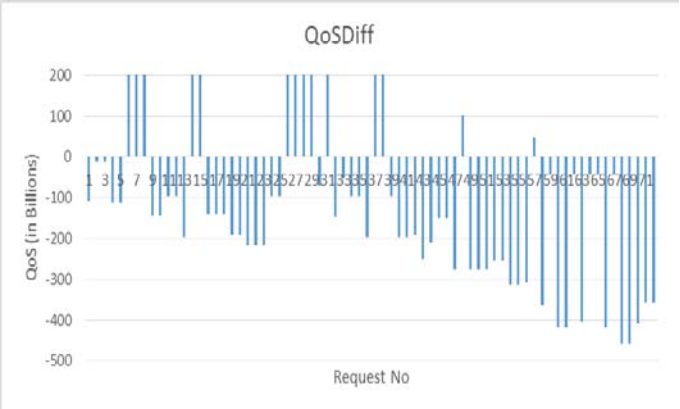

Figure 2: QoS Difference

Time taken for processing the requirements is measured beginning from window creation, prediction and identification of the package (figure 3). It was observed that most requests took ~2ms for processing. A few spikes measuring up to 6ms and a few low requirements measuring to 1ms were also observed during the allocation process. An average identification time of 2.1ms exhibits that the proposed architecture exhibits high correlation to real-time processing speeds.

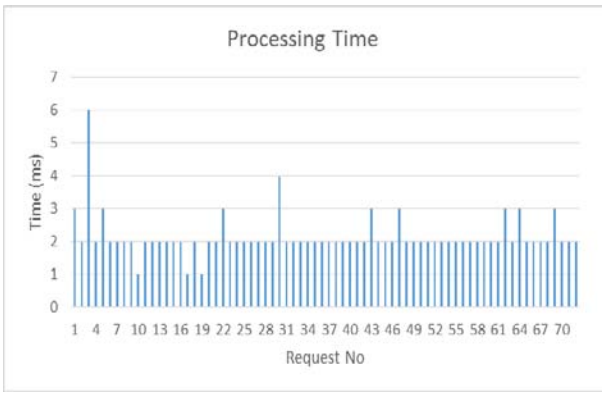

Figure 3: Processing Time (ms)

The effectiveness of the Auto-regression based prediction is measured in terms of the error levels exhibited by them. Error levels are usually measured in terms of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) [26, 27]. Mean Absolute Error measures the effectiveness of the predictions, and is given by

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|y_i - y'_i|$$

Root Mean Square Error indicates the variability of the predictions, and is given by

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(y_i - y'_i)^2}$$

Where y_i and y_i' are the actual and the predicted ratings for the *N* test reviews.

The error metric levels obtained from TWARP model is shown in figure 4. It could be observed that the proposed model exhibits very low error levels of MAE at 0.19 and RMSE at 0.02.

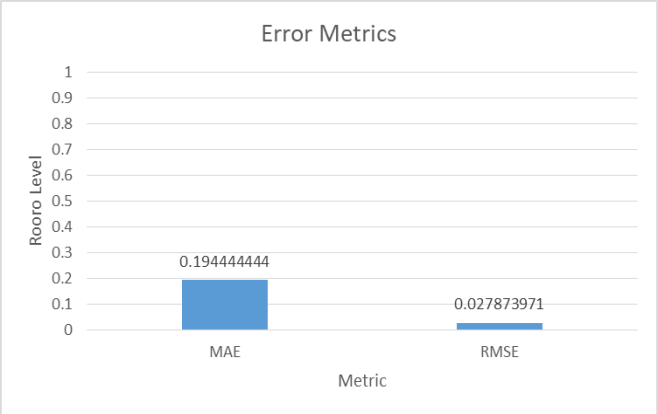

Figure 4: Error Metrics for Prediction Analysis

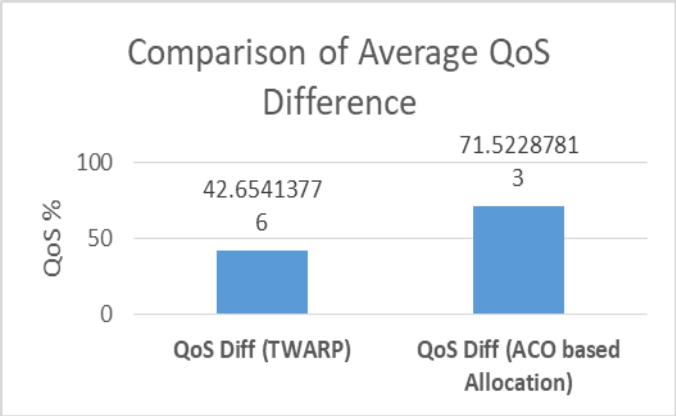

Figure 6: Average QoS Comparison

A comparison of the proposed TWARP with the technique proposed by Madhumathi et al. [25] is shown in figures 5 and 6. Madhumathi et al. proposed a cloud provisioning technique based on modified ACO as the base metaheuristic algorithm. Comparison is carried out in terms of time taken for the optimization process and the average QoS difference levels exhibited by the techniques.

A time comparison of the proposed model with the ACO based allocation scheme is shown in figure 5. It could be observed that the proposed model exhibits an average time requirement of 2.15ms, while the ACO based allocation scheme requires 85.6ms. the proposed TWARP model was observed to exhibit ~42X reduced time requirements.

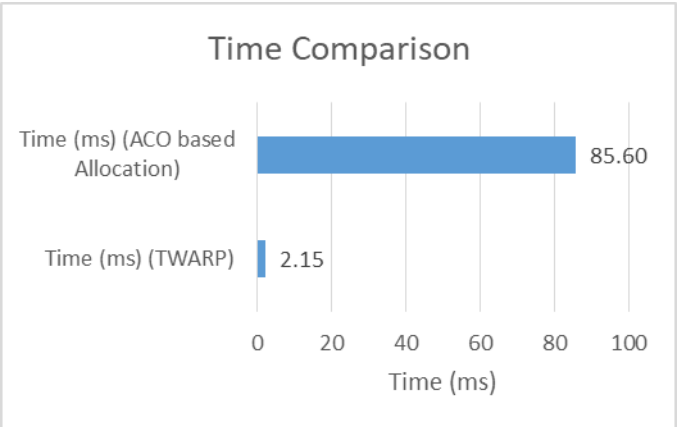

Figure 5: Time Comparison

A comparison of the overall average difference between the QoS values is shown in figure 6. It could be observed that the proposed TWARP model exhibits a QoS difference of 42%, while the ACO based allocation model exhibits an average difference of 71%. The proposed model exhibits a reduction level of 29%, depicting the highly effective selections exhibited by the proposed model.

V. CONCLUSION

Resource provisioning in cloud environments is one of the major requirements of the current resource intensive environment. Incorporating dynamic resource requirements is one of the major concepts to be handled to convert it to an autonomic process. This paper presents a Time Window based Auto-Regressive Hybrid PSO (TWARP) model that aims to effectively handle the dynamic service requirements of the user. The model has been designed to capture long time drifts, hence a time window of 12 months is considered. The major advantage of this approach is that it provides faster allocation, enabling optimal resource management for the provider. The major limitations of the proposed model are that perfect allocation was never possible. Instead, the package that optimally meets the requirements is assigned. The proposed model can be extended to service providers to recommend granular changes in the package requirements to enable most effective allocations. Future directions of the proposed model can also include creating a package list pool that can be accessed by multiple users. Extending the model to create an autonomic computing environment that performs automatic upscaling and downscaling. Enabling the time window to be elastic can effectively capture the nuances contained in the resource requests.

VI. REFERENCES

[1] A.B. Grant and O.T. Eluwole, “Cloud resource management virtual machines competing for limited resources,” In: 2013 55th international symposium ELMAR. IEEE; 2013, pp. 269–274.

[2] A. Gulati, G. Shanmuganathan, A. Holler and I. Ahmad, “Cloud-scale resource management:challenges and techniques,” In: Proceedings of the 3rd USENIX conference on Hot topics in cloud computing. USENIX Association; 2011, pp. 3-3.

[3] PT. Endo, A.V. de Almeida Palhares, N.N. Pereira, G.E. Goncalves, D. Sadok and J. Kelner, “et al.Resource allocation for distributed cloud: concepts and research challenges,” IEEE Netw vol.25(4), 2011, pp.42–46.

[4] A. Tchernykh, U. Schwiegelsohn, V. Alexandrov and E.G. Talbi, “Towards understanding uncertainty in cloud

omputing resource provisioning," Procedia Computer Science, 51, 2015, pp.1772-1781.

[5] M. Amiri and L. Mohammad-Khanli, "Survey on prediction models of applications for resources provisioning in cloud," J. Netw. Comput. Appl. 82. 2017, pp. 93–113. http://dx.doi.org/10.1016/j.jnca.2017.01.016.

[6] S. Singh, I. Chana and R. Buyya, "STAR: SLA-aware autonomic management of cloud resources," IEEE Trans. Cloud Comput. http://dx.doi.org/10.1109/TCC. 2017.2648788. 2017.

[7] A.N. Toosi, R.O. Sinnott, and R. Buyya, "Resource provisioning for data-intensive applications with deadline constraints on hybrid clouds using Aneka," Future Generation Computer Systems, 79, 2018, pp.765-775.

[8] M. Ghobaei-Arani, S. Jabbehdari and M.A. Pourmina, "An autonomic resource provisioning approach for service-based cloud applications: a hybrid approach," Future Gener. Comput. Syst. http://dx.doi.org/10.1016/j.future.2017. 02.022. 2017.

[9] G. Mateescu, W. Gentzsch and C.J. Ribbens, "Hybrid computing-where HPC meets grid and cloud computing," Future Gener, Comput. Syst. 27 (5). 2011, pp.440–453. http://dx.doi.org/10.1016/j.future.2010.11.003.

[10] B. Javadi, J. Abawajy and R. Buyya, "Failure-aware resource provisioning for hybrid cloud infrastructure," J. Parallel Distrib, Comput. vol. 72 (10). 2012, pp. 1318–1331. http://dx.doi.org/10.1016/j.jpdc.2012.06.012

[11] X. Xu and X. Zhao, "A framework for privacy-aware computing on hybrid clouds with mixed-sensitivity data in: Proceedings of the IEEE International Symposium on Big Data Security on Cloud," 2015, pp. 1344–1349 http://doi.org/10 1109/HPCC-CSS-ICESS.2015.110.

[12] X. Xu, M. Tang, and Y.C. Tian, "QoS-guaranteed resource provisioning for cloud-based MapReduce in dynamical environments," Future Generation Computer Systems, 78, 2018, pp.18-30.

[13] S. Ibrahim, H. Jin, L. Lu, S. Wu, B. He and L. Qi, "LEEN: Locality/fairness-aware key partitioning for MapReduce in the Cloud, in: Proceedings of IEEE 2nd International Conference on Cloud Computing Technology and Science," CloudCom, IEEE, New York, Indianapolis, IN, 2010, pp. 17–24.

[14] H. Lin, X. Ma, J. Archuleta, W.-c. Feng, M. Gardner and Z. Zhang, "Moon: MapReduce on opportunistic environments," in: Proceedings of ACM 19th International Symposium on High Performance Distributed Computing, ACM, New York, Chicago, IL, 2010, pp. 95–106.

[15] B. Tang, M. Moca, S. Chevalier, H. He and G. Fedak, "Towards MapReduce for Desktop Grid Computing," in: Proceedings of 2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC, IEEE, New York, Fukuoka, Japan, 2010, pp. 193–200.

[16] L. Wang, J. Tao, R. Ranjan, H. Marten, A. Streit, J. Chen and D. Chen, "G-Hadoop: MapReduce across distributed data centers for data-intensive computing," Future Gener. Comput. Syst. Vol. 29 (3). 2013, pp.739–750.

[17] K. Kc and K. Anyanwu, "Scheduling hadoop jobs to meet deadlines, in: Proceedings of IEEE 2nd International Conference on Cloud Computing Technology and Science," CloudCom, IEEE, New York, Indianapolis, IN, 2010, pp. 388–392.

[18] J. Polo, Y. Becerra, D. Carrera, M. Steinder, I. Whalley, J. Torres and E. Ayguade, "Deadline-Based MapReduce Workload Management," IEEE Trans. Netw. Serv. Manag, vol. 10 (2). 2013, pp.231–244.

[19] W. Zhang, S. Rajasekaran, T. Wood and M. Zhu, "Mimp: Deadline and interference aware scheduling of hadoop virtual machines," in: Proceedings of IEEE/ACM 14th International Symposium on Cluster, Cloud and Grid Computing, CCGrid, IEEE/ACM, New York, Chicago, IL, 2014, pp. 394–403.

[20] P.C. Phillips, Towards a unified asymptotic theory for autoregression. Biometrika, vol.74(3), 1987, pp.535-547.

[21] C. Han, P.C. Phillips, and D. Sul, "Lag length selection in panel autoregression," Econometric Reviews, vol. 36(1-3), 2017, pp.225-240.

[22] J.Kennedy, and R. Eberhart, "PSO optimization. In Proc," IEEE Int. Conf. Neural Networks Vol. 4, 1995, pp. 1941-1948. IEEE Service Center, Piscataway, NJ.

[23] Y. Shi, and R. Eberhart, "A modified particle swarm optimizer," In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on May, 1998, pp. 69-73. IEEE.

[24] P. J. Van Laarhoven, and E. H. Aarts, "Simulated annealing," In *Simulated annealing: Theory and applications* 1987, pp. 7-15. Springer, Dordrecht.

[25] C. Madhumathi, and G. Ganapathy, "Cloud Package Selection for Academic Requirements using Multi Criteria Decision Making based Modified Ant Colony Optimization Technique," International Journal of Engineering and Technology (IJET), 8, 2016, pp. 1205-1211

[26] S. Dooms, T. De Pessemier and L. Martens, "Offline optimization for user-specific hybrid recommender systems," *Multimedia Tools and Applications*, vol. 74(9), 2015, pp.3053-3076.

[27] X. Ge, J. Liu, Q. Qi, and Z. Chen, "A new prediction approach based on linear regression for collaborative filtering," IEEE *Eighth International Conference* In *Fuzzy Systems and Knowledge Discovery (FSKD)*, Vol. 4, 2011, pp. 2586-2590

BIOGRAPHY

Mani. K received his MCA and M.Tech from the Bharathidasan University, Trichy, India in Computer Applications and Advanced Information Technology respectively. Since 1989, he has been with the Department of Computer Science at the Nehru Memorial College, affiliated to Bharathidasan University where he is currently working as an Associate Professor. He completed his PhD in Cryptography with primary emphasis on evolution of framework for enhancing the security and optimizing the run time in cryptographic algorithms. He published and presented round 15 research papers at international journals and conferences.

R.Mohana Krishnan received his MCA from the Bharathidasan University, Trichy India in Computer Applications and Advanced Information Technology respectively and MBA from Great Lakes Institute of Management with Association with Stuart School of Business, illinois institute of technology, has 22 years of IT & Strategic Management experience at Various Corporates in India and abroad IBM, CSC, GEMINI, U.S.A and HCR,USA in the field of IT Delivery and Complex Solutions Provider, Client Engagement, Pre Sales and Technology solutions for various domains. Expertise in delivering enterprise level solutions with prime focus in creating maximum value through strategic planning, Leadership in Solutions. Currently associated with HCL India as AGM, and leading the Application and Solutions.