International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 14 (2017) pp. 4448-4454

Fast and Effective Fuzzy Parameters based cloud provisioning by means of Modified PSO

A.Inbarajan[1] and Dr. N. P. Gopalan[2]

[1]Research Scholar, Computer Science and Engineering, PRIST University, Tanjore, India.

[2]Professor, Department of Computer Applications, National Institute of Technology, Trichy, India.

Abstract

Cloud providers are on the increase and cloud users are on an increase too, due to the low demands and high delivery rates. Provisioning of cloud resources have been the topic of interest for a long while. But current cloud providers do not operate by provisioning resources in a standalone manner. Instead, plan based resource provisioning is carried out to enhance the usability of the systems creating a win-win situation for both the customer and the provider. This paper presents a method that uses fuzzy quality parameters to enable users to select best plans that caters to their needs perfectly. Migrators or first time users do not have any knowledge on how the quality parameters impact the cloud services. This paper uses past log information of the migrators and application details of the first time users to provide them with the best plans. After obtaining the fuzzy requirements, cloud plans are analyzed and optimal plans are identified using metaheuristic techniques.

Keywords: Cloud Provisioning; Cloud Plan Selection; Particle Swarm Optimization; Usage Logs.

INTRODUCTION

Cloud based architectures have become major resource utilities, due to their flexible nature and the large variety of services provided by them [3]. Clouds support IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service). The advantage of using cloud architecture is that it requires minimal resources from the part of the customer [4]. Hence the usage cost is very low compared to dedicated models [5,6]. Further, the flexibility of cloud as a pay per use model and automatic upward scalability makes it a more desirable model. Several cloud provisioning systems have been proposed in literature providing insights on methodologies that can be used for effective cloud provisioning.

An IaaS cloud model, designs specifically for scientific workflows is presented in [1]. This method takes cost and deadlines as the most important constraints for cloud provisioning. Due to the scientific nature of the working problem, this algorithm proposes both static and dynamic strategies for cloud and resource provisioning. A resource provisioning model for mobile cloud based architectures is proposed in [2]. This method is developed for independent resource provisioning, and it operates by predicting and storing resource usage logs in a 2D matrix. This matrix is then used to predict future resource needs. Similar mobile cloud provisioning methods include [9, 10, 11]. Several critical applications have now been deployed in cloud; hence failure awareness has become one of the major necessities of any cloud provider. Failure aware provisioning of resources are in the raise due to the reliability associated with them. Some failure aware techniques include [19, 20, 21]. These methods incorporate a failure handling by analyzing the real time threats and constructing allocation policies accordingly. Apart from the previously mentioned services from cloud (IaaS, PaaS and SaaS), several other services has been currently included by cloud providers, such as XaaS (All as a Service) and NaaS (Network as a Service). Methods to facilitate effective provisioning for a NaaS system is presented in [22].

A platform named Aneka [7, 8] was developed to collect resource requirements from environments such as grids, desktops, public and private clouds, in order to perform effective provisioning. Quality based resource provisioning methods have also been proposed, that allots resources on the basis of QoS metrics [12]. Several workloads [13] were analyzed and these workloads are used to identify the best resource that suits the current requirements. Methods considering the workloads alone and not SLA agreements have also been in use, such as [14, 15]. Other methods strictly considering SLA agreements but do not consider time include [16, 17, 18].

It can be observed from the literature that several resource provisioning techniques have been proposed, but cloud architectures in the current scenario do not provide individual resources, instead they provide pre-formulated plans that could be utilized by the consumers. Further, most of the techniques are feedback based, that allots random plans and

fine tunes them later, according to the user. Users tend to have their past logs (web or cluster), that can be used to predict optimal resource requirements. This paper presents methods that can be used to formulate cloud plans on the basis of their past usage data. This data is usually in the form of web logs or access logs. User's quality requirements are extracted from these documents and PSO is used to identify the optimal plan pertaining to the user's needs.

OUR APPROACH

Cloud Architectures have become the major areas of utility due to their flexible nature and lesser demands when compared to dedicated systems. This has led to a huge migration of users from dedicated environments to cloud architectures. Though flexibility cannot be provided by cloud services from all aspects, cloud providers offer several plans catering to the demands of almost all the requirements of the users. The users are required to select plans catering to their minimal demands [23, 24].

Due to the availability of several plans, it becomes difficult for the user to select a particular plan. Another major problem lies in the user understanding their requirements [25]. Most of the users are unaware of their own requirements, which lead to selection of inappropriate plans. Selecting a plan that is underutilized would lead to unnecessary cost for the user, while selecting a plan that is insufficient leads to frequent demands from the cloud provider, which can prove to be costly. Since cloud services are scalable, if an increase in the requirements is observed, the plans are automatically shifted to the next higher level. But this service tends to be costlier when used from outside the plan. Hence selecting an appropriate plan is necessary for cost efficient working of the system.

Migrating users are usually unaware of the Quality of Service parameters available in the cloud plans. If the user has had an experience in cluster or web, their usage logs provide a good insight on the requirements of the user. This paper proposes a fuzzy based technique that analyzes the user's logs to identify the quality requirements of the user.

The quality parameters [12] considered in our algorithm are described below.

- ***Bandwidth:*** Network bandwidth refers to the number of bits transferred (sent/ received) in a particular workload in one second.

Network Bandwidth=Bits/ second (B/S) *(1)*

Usage logs contain information about the size of data transferred from/ to the node along with the timestamp. This data can be aggregated to identify the average data transferred, which can in turn be used to identify the bandwidth requirement.

- ***Availability:*** Availability refers to the level of recovery achieved by the system in case of failure.

Availability=mean time to failure/(mean time to failure + mean time to repair) *(2)*

Failures are usually represented by status codes representing 4xx and 5xx, where 4xx refers to client error and 5xx refers to server errors. Successful transactions are usually represented by 2xx. Time taken between a 4xx or 5xx status code and 2xx status code is considered as the recovery time.

- ***Computational Capacity:*** Computational capacity refers to the ratio between the actual time of usage of a resource to the expected usage time.

Computing Capacity = Actual Usage time of the Resource/Expected Usage time of the Resource. *(3)*

- ***Usability:*** Usability refers to the ability of the system to perform successful operations. The higher the usability ratio, the better the system's operation.

Usability=no of successful operations in a workload/ (total operations available in the workload) *(4)*

- ***Correctness:*** Correctness defines the degree to which the cloud service will be provided accurately to the cloud customers.

Correctness=total number of failed transmissions/ (total number of failed transmissions + total number of successful transmissions) *(5)*

- ***Variable computing load:*** It is the change in the load balance with respect to time. This can be calculated identifying the workload variance in each transmission.

- ***Reliability:*** Reliability refers to the time taken for the system to recover after failure.

Reliability=mean time to failure + mean time to repair *(6)*

- ***Latency:*** Latency is the time taken to process a single workload,

Latency = Time of input a Cloud workload –Time of output produced with respect to that Cloud workload (7)

- ***Serviceability:*** Serviceability is the probability of the service being up and running

Serviceability= Service Uptime/ (Service Uptime+ Service Downtime) (8)

The logs generally contain timestamps, transaction status and payload sizes, which can be used for most of the interpretations. This direct interpretation tends to provide absolute values for the quality parameters, which could not be utilized directly to perform recommendations. Hence these values are fuzzified and used for further computations.

Some users tend to adopt cloud services in the first generation of their processing phase. Hence log records pertaining to their usage scenario might not be present. In such cases, the user's needs are roughly formulated and are subject to clustering. This groups similar user and the quality parameters are extracted from these users to identify the probable parameters for the current user. The proposed architecture of the cloud provisioning system is presented in Fig 1.

Service plans were constructed using fuzzified values divided into three classes; low medium and high. Each quality parameter is provided a value of low, medium or high depending on the requirement of the service. Our application uses 11 service plans, each with its own individual requirements. The service plans are designed considering real time application scenarios. The service plans correspond to websites, technological computing, online transaction processing, e-commerce, financial services, productivity applications, software/ project development, graphics oriented, critical internet applications and mobile computing services.

The final phase of this paper deals with identifying the corresponding service plan depending on the user's requirements. There is no straight forward approach available to perform this operation. A best plan catering perfectly to the user's needs might not be available. Hence this becomes an optimization problem that has its own set of tradeoffs to be incorporated to obtain the final solution. This tradeoff is either in terms of cost or quality. Our approach uses a modified form of Particle Swarm Optimization (PSO) [26, 27,28] algorithm to perform the optimization process. Service plans form the search space of PSO and all the particles are dispersed from a single point, describing the user's requirements.

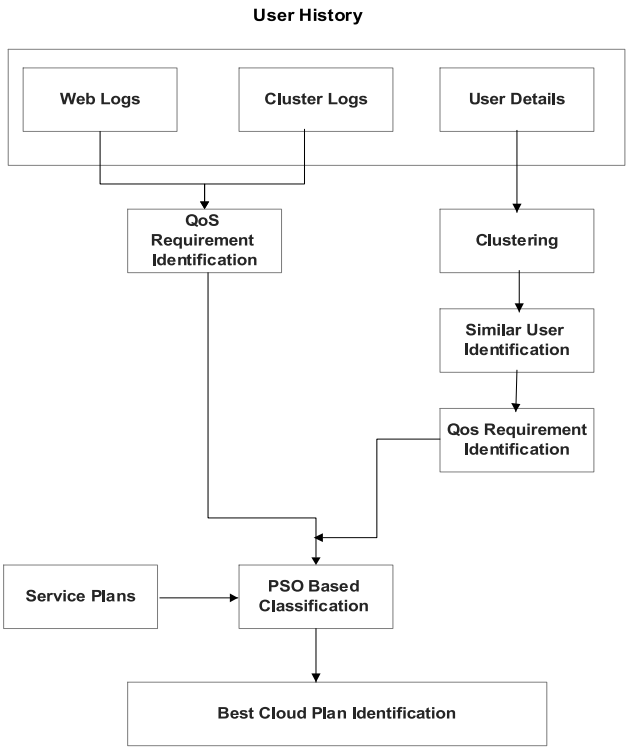

Figure 1. Architecture of the Cloud Provisioning System

The first phase of PSO is the initialization of the search space and the particles. The search space in our application corresponds to the services plans from the cloud service provider. All the particles are set to an initial velocity making sure the co-ordinates fall within the bounds of the search space. The initial velocity is calculated using

$$V_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|) \quad (9)$$

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

The particle best (*pbest*) and global best (*gbest*) values are maintained so as to keep track of the best solution obtained from a single particle and the best solution pertaining to the entire swarm. The velocity and direction of a particle are dictated by the *pbest* and the *gbest* values obtained in the previous iterations.

The velocity and direction of a particle are calculated using

$$V_{i,d} \leftarrow \omega V_{i,d} + \varphi_p r_p (P_{i,d} - X_{i,d}) + \varphi_p r_p (g_d - X_{i,d}) \quad (10)$$

Where r_p and r_g are the random numbers, $P_{i,d}$ and g_d are the parameter best and the global best values, $X_{i,d}$ is the value current particle position, and the parameters ω, φ_p, and φ_g are selected by the practitioner.

Though PSO operates on continuous domain, PSO in its discrete form is used in this application. After a particle accelerates to its new location, the position of the particle is used to identify the nearest discrete point and it is moved there. The new location is calculated using

$$P' = \min\left(\sum_{j=1}^{n}\left(\sum_{k=1}^{d}\sqrt{(P_{ik}-N_{jk})^2}\right)\forall i = 1\ to\ p\right) \quad (11)$$

Where P_{ik} refers to the particle i's current location corresponding to dimension k, N_{jk} refers to the k th dimension of node Ni.

This process is repeated for a specified time interval or until the same solution is repeated for defined number of times. The global best obtained at the end of the iteration is considered as the final solution. In this application, it corresponds to a service plan that best suits the user's requirement.

RESULTS AND DISCUSSION

Experiments were conducted using the access log dataset, containing approximately 4.4 million entries (4,477,823) representing transactions of 333,924 individual users on various timelines. Data fuzzification and quality identification are performed using this data by aggregating them on the basis of the IP address.

A ratio between the required QoS and the QoS provided by the system are presented in Figure. The x axis represents the user's requirements and y axis represents the plan allotments. It could be observed that the plan allotments are kept to the nearest possible extent with respect to the requirements

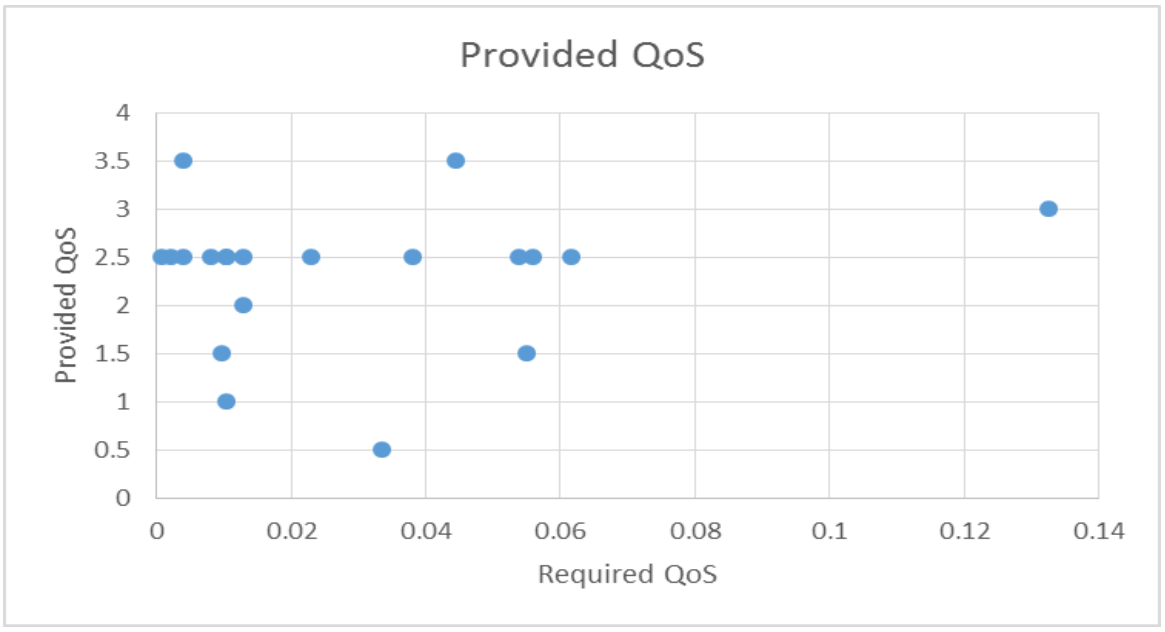

Figure 2. Provided QoS

The difference between requested QoS and provided QoS is presented in Figure. It shows that the points plotted all occupy the positive quadrant, showing that none of the requirements were services with plans exhibiting requirements below the requested quality. Further the quality deviations are maintained to the minimum levels exhibiting the efficiency of the system.

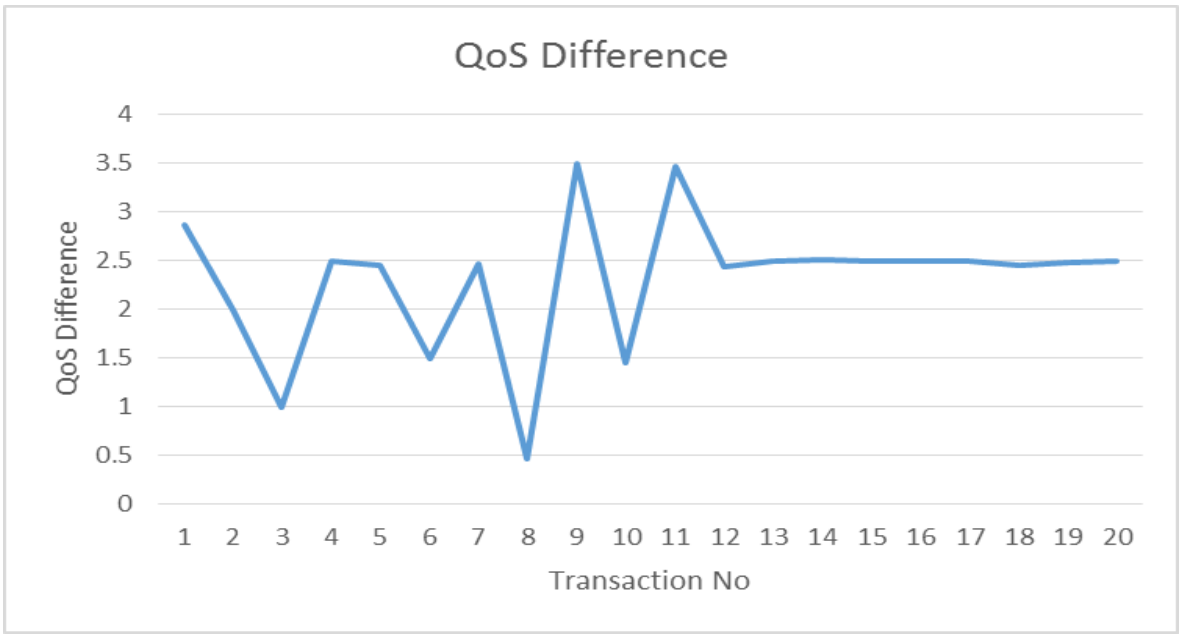

Figure 3. QoS Difference

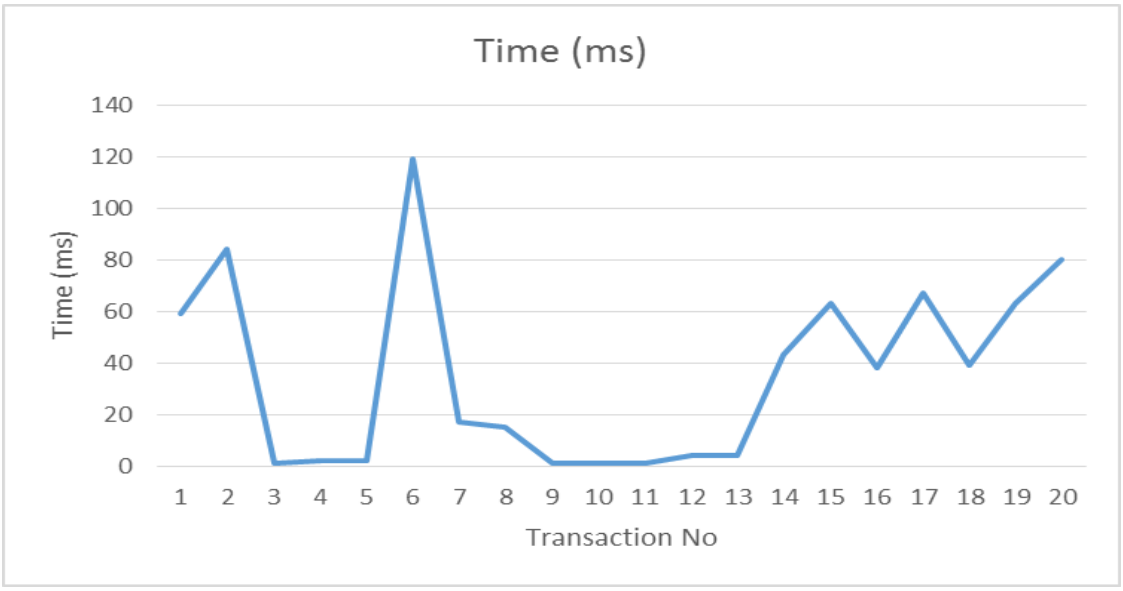

Figure 4. Processing Time for PSO

Time taken for allotments using the PSO algorithm is presented in Figure. It could be observed that the maximum time consumed for identifying the appropriate service is 119ms, which is approximately 0.1 sec, while the minimum time consumption can be observed to be <10ms.

CONCLUSION

A novel and fast method to identify optimal cloud plan using user logs and user requirements is presented in this paper. Usage of a metaheuristic technique (PSO) has ensured the plan selection is performed faster. It could be observed from the results that the QoS provided by the current algorithm meets the requirements of the users and is never under delivered. Since, the cloud plans are pre-defined, they have rigid QoS properties, and hence a perfect match of quality parameters cannot be achieved. The time taken for provisioning is also acceptable, hence our method is considered to be one of the most appropriate methods for provisioning cloud plans. Our approach can be further extended by incorporating effective local search techniques. Fault tolerance is one of the major issues that are to be concentrated in the cloud environment, due to the critical

nature of the applications running in it. Our future researches will also concentrate on incorporating fault tolerance into the provisioning system.

REFERENCES

[1] M. Malawski, G. Juve, E. Deelman and J. Nabrzyski, "Algorithms for cost- and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds ", Future Generation Computer Systems, Volume 48, Pages 1–18, July 2015.

[2] S.K. Sood and R. Sandhu, "Matrix based proactive resource provisioning in mobile cloud environment", Simulation Modelling Practice and Theory, Volume 50, Pages 83-95, January 2015.

[3] M. Mezmaz, N. Maleb, Y. Kessaci, Y.C. Lee, E.G. Talbi, A.Y. Zomaya and D. Tuyttens, "A parallel bi-objective hybrid meta-heuristic for energy aware scheduling for cloud computing systems", J. Parallel Distrib. Comput. 71 (11)-1497–1508, 2011.

[4] S. Islam, J. Keung, K. Lee and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud", Future Gener. Computer Syst. 28 (1)-155–162, 2012.

[5] E. Caron, F. Desprez and A. Muresan, "Forecasting for cloud computing on-demand resources based on pattern matching, J. Grid Comput. 9 (1)-49–64, 2011.

[6] S.K. Sood, "A combined approach to ensure data security in cloud computing", J. Network Computer Appl. 35 (6)-1831–1838, 2012.

[7] R.N. Calheiros, C. Vechiola, D. Karunamoorthy and R. Buyya, "The Aneka platform and QoS driven resource provisioning for elastic applications on hybrid clouds", Future Gener. Comput. Syst. 28 (6)-861–870, 2012.

[8] R.N. Calheiros, C. Vecchiola, D. Karunamoorthy and R. Buyya, "Deadline-driven provisioning of resources for scientific application in hybrid clouds with Aneka", Future Gener. Computer Syst. 28 (8)-58–65, 2012.

[9] L. Deboosere, P. Simoens, J.D. Wachter, B. Vankeirsbilck, F.D. Turck, B. Dhoedt and P. Demeester, "Grid design for mobile thin client computing", Future Gener. Computer Syst. 27 (6)-681–693, 2011.

[10] J. Park, H. Kim, Y.S. Jeong and E. Lee, "Two-phase grouping-based resource management for big data processing in mobile cloud", Int. J. Commun. Syst., http://dx.doi.org/10.1002/dac.2627, 2013.

[11] M. Shiraz, A. Gani, R.H. Khokhar and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing", IEEE Commun. Surveys Tutorials 15 (3)-1294–1313, 2013

[12] S. Singh and I. Chana, "Q-aware: Quality of service based cloud resource provisioning, Computers & Electrical Engineering", In Press, Corrected Proof, Available online 25 February 2015.

[13] S. Sukhpal and C. Inderveer, "QRSF: QoS-aware resource scheduling framework in cloud computing". J Supercomput; 71(1):241–92, 2015.

[14] B. Mauricio, L. Keith, C. Anton, K. Patryk and P. Leonardo. "Cloud workload analysis with SWAT". In: IEEE 24th international symposium on, computer architecture and high performance computing (SBAC-PAD). IEEE; p. 92–99, 2012.

[15] D. Christina and K. Christos. "iBench: quantifying interference for datacenter applications". In: 2013 IEEE international symposium on, workload characterization (IISWC). IEEE; p. 23–33, 2013.

[16] S. Seokho, J. Gihun and J.Chan. "An SLA-based cloud computing that facilitates resource allocation in the distributed data centers of a cloud provider". J Supercomput; 64(2):606–37, 2013.

[17] L. Leigh, "Application workload prediction and placement in cloud computing systems [PhD Dissertation]". Massachusetts Institute of Technology; 2014.

[18] C. Chung, C. Shiung and C. Feng-Wei. "A predictive method for workload forecasting in the cloud environ ment". In: Advanced technologies, embedded and multimedia for human-centric computing. Lecture notes in electrical engineering", vol. 260. Netherlands: Springer p. 577–85, 2014.

[19] B. Javadi, J. Abawajy and R. Buyya, "Failure-aware resource provisioning for hybrid Cloud infrastructure", Journal of Parallel and Distributed Computing, Volume 72, Issue 10, Pages 1318-1331, 2012.

[20] J.H. Abawajy, "Determining service trustworthiness in Intercloud computing environments", in: The 10th International Symposium on Pervasive Systems Algorithms, and Networks, ISPAN, pp. 784–788, 2009.

[21] T. Mather, S. Kumaraswamy and S. Latif, "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O'Reilly Media, Inc., 2009.

[22] J. Huang, G. Liu, and Q. Duan,"On modeling and optimization for composite network–Cloud service provisioning", Journal of Network and Computer Applications, Volume 45, Pages 35-43, 2014.

[23] Tchernykh, U. Schwiegelsohn, V. Alexandrov and El-ghazali Talbi,"Towards Understanding Uncertainty in Cloud Computing Resource Provisioning",Procedia Computer Science, Volume 51, Pages 1772-1781, 2015.

[24] D. Hu, N. Chen, S. Dong and Y. Wan, "A User Preference and Service Time Mix-aware Resource

Provisioning Strategy for Multi-tier Cloud Services",AASRI Procedia, Volume 5, Pages 235-242, 2013

[25] R. Mian, P. Martin and J. Vazquez-Poletti, "Towards effective science cloud provisioning for a large-scale high-throughput computing", Future Generation Computer Systems, Volume 29, Issue 6, Pages 1452-1458, 2013.

[26] De Falco, I., Della Cioppa , A. and Tarantino, E. “Facing classification problems with Particle Swarm Optimization”. Applied Soft Computing, Volume 7, Issue 3,Pages 652–658, 2007.

[27] Kennedy, J. and Eberhart, R. “Particle Swarm Optimization”. Proceedings of IEEE International Conference on Neural Networks IV. pp. 1942–1948.doi:10.1109/ICNN.1995.488968, 1995.

[28] M. Devi and R. Sukumar, "Metaheuristic Based Noise Identification and Image Denoising Using Adaptive Block Selection Based Filtering", Circuits and Systems, 7, 2729-2751, 2016.