International Journal of Grid and Distributed Computing
Vol. 9, No. 1 (2016), pp.99-104
http://dx.doi.org/10.14257/ijgdc.2016.9.1.10

Trust Based Cloud Service Composition Framework

N. Sasikaladevi

School of Computing, SASTRA University
sasikalade@gmail.com

Abstract

Cloud Computing plays an important role in academic and industry. Business process are composed and executed in the distributed loosely coupled environment. Business processes are implemented as the composite service which involves many services and these services connected by different workflow patterns. Numerous functionally similar services are available in the cloud. As it is an on-demand service, cloud service consumer needs some Quality of Service factors in services. Trust plays an import role in service selection. Selecting the trust worthy service for service composition with the cost and other constraints is a tedious task. In this paper, service trust estimation method based on Beta distribution is proposed. Trust based Cloud service selection framework for service composition is proposed and it is proved that the proposed framework provides better performance as compared to the well-known service composition methods based on the execution time and optimality value.

Keywords*: Cloud Computing, MMKP, QoS, Trust, Bayesian Inference.*

1. Introduction

Cloud Computing (CC) has emerged as a significantly important research area attracting much attention from both the research and industry. Cloud Services are available across the world in a loosely coupled distributed environment. Service consumer an able select the services based on their own Quality of Service requirements [1]. Cloud Service may either refer to the functional atomic service or it may be the composite service. As it is a online service and pay and use service, consumer needs trust worthy services from the provider. Whenever there is a functionally similar set of services, consumer has a choice to choose the best service based on the trust value of the service. But it is not possible to maintain only trust worthy services in the service registry [2]. Trust is the probability by which person A expects that other person B performs a required task perfectly and it is between the range o to 1 [3].

Trust management is a serious issue in CC environments. Business process consists of numerous services and connected by different workflow patterns. Each service may need to invoke another service to form the composite service to represent the business process. To satisfy the specified functionality requirement, a service may have to invoke other services forming composite Web services with complex invocations and trust dependencies among services and service providers [4]. Although these surely enhance the service provision, they really raise the computation complexity and thus make trust based service selection is a tedious task.

In this paper, the Trust based composite service composition framework is presented. Then, for the atomic service using Beta Distribution method is proposed. Trust based service selection model is proposed. Final section shows the status of the result and the compassion between the proposed model and well known service composition model. Trust based Cloud Service Selection Framework (TCSSF) is a framework used to select the services for service composition. It selects trust worthy service for composition. Trust plays a vital role in this composite service selection model. Cloud provides on-demand

ISSN: 2005-4262 IJGDC

service[5,6]. Whenever online services are used by the service consumer, he/she requires the cost effective service with low delay. Cost and response time are the predominant Quality of Service (QoS) parameter today. Service consumers need the trust worthy service for interaction. TCSS framework enables the user to select trustworthy services for composition with the constraints such as cost and response time.

2. Related Work

Lie [7,8] proposed service invocation graph and service invocation matrix for composite service representation. In addition, we propose a trust evaluation method for composite services based on Bayesian inference. He proposed trust based service selection and discovery algorithm based on Monte Carlo method. Zainab [8] proposed trust based mediator framework based on trust metrics. They used whitewashing, cold start and trust bootstrapping methods. Zhenziang [10] proposed trust model based on trust behavior. They used trust gravitation method and analyze the trust relationship between pervasive entities.

Diana Comes *et al*. [11] proposed multi objective based heuristic algorithm for the selection of services in service orchestrations. The OPTIM_HWeight algorithm is based on weight factors and the OPTIM_PRO algorithm is based on priority factors for performing the service selection. The OPTIM_HWeight make use of a specific heuristic function called HWeight. Diana Comes *et al*. [11] proposed another multi objective based heuristic service selection algorithm OPTIM_PRO.

Rong Wang *et al*. [12] proposed the fast heuristic algorithm for SSP. In this work, the utility function is described to reflect the QoS parameters of the service selection model. And then, the service selection model is mapped into MMKP form. It is the Simple Heuristic that is used to solve the MMKP. Tao Yu *et al*. [13] proposed WS-HEU algorithm for web service selection for composite web service. WS-HEU intended to find solutions for MMKP. The search is time-consuming.

Diana's Models OPTIM_HWeight and OPTIM_PRO incorporate reliability, response time, cost and availability as the QoS factors. OPTIM_HWeight uses the gradient descent approach. It is used to deliver a score for the candidate service selection versus the current service selection; a higher value is ranked higher. This approach is referred as a gradient ascent. OPTIM_PRO is based on Priority approach. Candidates are ranked based on the utility value. Higher priority is given to service candidates with highest utility value. Highly ranked candidates are having higher probability of selection. F-HEU incorporated response time and cost as the QoS factors. F-HEU is penalty based approach. Penalty is assigned to infeasible candidates. Probability of selecting the infeasible candidates is low. WS-HEU incorporates response time, cost and availability as the QoS factors. WS-HEU is based on Khan's Approach. Khan's approach is based on the concept of one upgrade and one or more downgrades.

3. Trust based Cloud Service Selection Framework

Trust based Cloud Service Selection Framework (TCSSF) is a framework used to select the services for service composition and it is shown in Figure 1. It selects trust worthy service for composition. Trust plays a vital role in this composite service selection model. Cloud provides on-demand service. Whenever online services are used by the service consumer, he/she requires the cost effective service with low delay. Cost and response time are the predominant Quality of Service (QoS) parameter today. Service consumers need the trust worthy service for interaction. TCSS framework enables the user to select trustworthy services for composition with the constraints such as cost and response time. Trust estimator is a third party service which calculates the trust of the candidate services for composite Service.

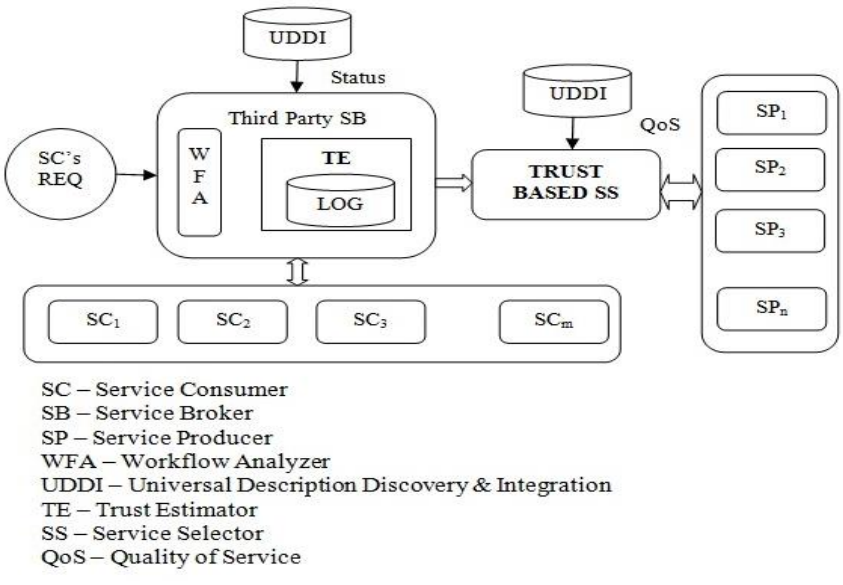

Figure 1. Trust based Cloud Service Selection Framework

Trust value of the service in estimated based on 3 methods such as direct trust, referral trust and weighted trust. Direct trust is estimated by the specific consumer based on the observations on the service. Referral trust value is estimated by getting the information from other service consumer who have already used their service. Weighted trust is estimated by adding direct trust and referral trust with different weight to each one.

Trust Estimator (TE) service uses the local log registry to maintain the status of each and every service. The structure of the log registry is given in the table. And it is updated whenever there is a service request. The status is classified as Successful and Failure. This information is maintained in the log registry as counters. If the service provides the satisfied result then its SI is incremented by one else FI is incremented by one.

3.1 Beta Distribution for Trust Estimation

Beta distribution is normally used to estimate the trust value of different system [14][15][16][17]. In the similar manner in this paper Beta distribution is used to estimate the Direct Trust Value (DTV), Referral Trust Value (RTV) and Weighted Trust Value (WTV). Beta distribution is the used to estimate the probability based on independent binary events.

Let x be a random variable of continuous type and let α and β be other positive constants such that $\alpha>0$ and $\beta>0$ be the real constants. Then, x is said to follow the beta distribution [22].

Probability distribution function $f(x)$ is defined as,

$$f_x(x) = \begin{cases} \dfrac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}, if\ 0 < x < 1 \\ 0, \qquad\qquad Otherwise \end{cases}$$

$$f_x(x) \geq 0\ , \forall x\ \in R\ and$$

$$\int_0^\infty f_x(x)dx = \frac{1}{B(\alpha,\beta)} \int_0^1 x^{\alpha-1}\,(1-x)^{\beta-1}dx = 1$$

The expected value (mean) (μ) of a Beta distribution random variable x with two parameters α and β is a function of only the ratio β/α of these parameters,

$$\mu = E(x) = \int_0^1 x\, f(x;\, \alpha,\beta)\, dx$$

$$= \int_0^1 x \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} dx$$

$$= \frac{\alpha}{\alpha+\beta}$$

TE search through the log and calculates the SI and FI value of the service given by the service consumer and then calculates the direct trust value (DTV) by using the equation given below,

$$DTV = \frac{\alpha}{\alpha+\beta} \; where \; \alpha = SI \; and \; \beta = FI \quad (1)$$

Direct trust estimation method has certain limitations such that Ballot-stuffing attack, Bad-mouthing attack, Newcomer and Sybil attacks, Sleeper and on–off attacks and Conflicting behavior attack [14].

In order to overcome the limitations of direct trust approach, Referral trust approach is proposed. In referral trust approach, Trust value is estimated by getting the information from server service observers (Service consumer) who have already used that specific service. Based on the inspiration from Denko Sun's approach [16], in this paper new method is defined to calculate the trust based on the direct and referral approach. In addition to get the status information of the service from the specific service consumer, the status information also collected from other service consumers, who have already used the service. That information is kept in the Log registry to estimate the referral trust RTV.

$$RTV = \frac{\alpha}{\alpha+\beta} \text{ where } \alpha = \sum_{i=1}^{n} SI \,,\; \beta = \sum_{i=1}^{n} FI \,, \quad (2)$$

where, n is the number of service consumers.

CC provides pay –and-use infrastructure for service consumer. As because they are paying, service consumer's observations on the service plays an important role. In Direct trust approach, individual's observations only considered, in referral approach others observations are considered. Still, there is a need to propose another method in which, the service consumer's observation should be weighted high as compared to others observations. Hence, new approach is proposed in order to remove the bias by other service consumers. The equation shows the weighted trust estimation for a service by considering both direct trust and referral trust.

$$WTV = .75 * DTV + .25 * RTV \quad (3)$$

In order to give more importance on the trust value estimated by the service consumer, .75 weight factors is assigned for DTV and .25 weight factors is assigned for RTV. In TCSS Framework the TE is used to estimate the trust value of the services. The Algorithm to TE is given below,

Algorithm1: TE

1. TE receives the service request from the service consumer.
2. TE searches the log registry by using service consumer ID and service request ID.
3. If there is any match found, then compute $DTV = \frac{\alpha}{\alpha+\beta}$ $where \; \alpha = SI \; and \; \beta = FI$
4. If the match not found, then it sets DTV= 0;

5. TE searches the log registry by using service request ID. If there is any match found, then compute $RTV = \frac{\alpha}{\alpha+\beta}$ where $\alpha = \sum_{i=1}^{n} SI$, $\beta = \sum_{i=1}^{n} FI$, Wher n is the number of service consumers.
6. If the match not found, then it sets RTV= 0;
7. TE calculate the WTV by using $WTV = .75 * DTV + .25 * RTV$
8. Return WTV

4. Trust Based Cloud Service Selection

Composite service consists of the several atomic services. All these services are connected by basis workflow patterns such as sequential, parallel, conditional, simple loop and dual loop. Figure shows the representation of each of the basic workflow pattern. In the web, functional similar services are available. Functionally similar service have different non-functional characteristic such as reliability, availability, response time, cost and trust. TCSS is used to select the trust worthy service from the functionally similar services for each service class. TCSS problem is represented in 0-1 MMKP form.

The Multi-dimensional Multi-Choice Knapsack Problem is one variation in 0-1 Knapsack problem. In 0-1 MMKP, there are n groups and each group contains m items. Here, group represents service classes and items represent functionally similar service candidates in each service class. Each service candidate has trust value and it has 2 QoS values such as cost, response time. The objective of the 0-1 MMKP is to pick exactly one service candidate from each service class for maximum total trust value of the collected items, subject to 2 QoS constraints of the knapsack[21].

TCSS problem is represented as,

$$maximize \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}T_{ij}$$

$$Subject\ to \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}\,Q_{ijk} \leq C_k, Where\ k = 1,2$$

$$\sum_{j=1}^{m} x_{ij} = 1, \quad x_{ij} \epsilon \{0,1\}$$

- Trust value of j^{th} service candidate of i^{th} service class
- Cost of j^{th} service candidate of i^{th} service class if k=1, Response time of j^{th} service candidate of i^{th} service class if k=2;
- Maximum cost and maximum response time for the composite service (this values are taken from Service Level Agreement)

This paper proposes the new heuristic algorithm to solve TCSS in 0-1 MMKP form. It is based on Khan [20] and Hifi Approach [19].

Algorithm 2: TCSS

Input parameter: SG,SC, n,m, T, C, RT, Max_C, Max RT
SG: Vector of Service classes
SC: Vector of Service Candidates in each service class
N: Number of Service classes
M: Number of Service Candidates
T: Trust matrix of size n*m

C: Cost vector of service candidates
RT: Vector of Response time of service candidates
Max_C: maximum cost value
Max_RT: Maximun response time
Output parameter: SS
SS: vector of selected service candidate in each service class.

Procedure:
Step 1: Sort SC of each SG based on T. So that minimum feasible candidates are on to the top.
Step 2: Initial Solution
Select top sc from SC of each SG and refer that as SS
Sum the C of each candidate in SS and RT of each candidate in SS as sum_c and sum_rt
Sum the T of each candidate in SS as sum_t
If sum_c >Max_C or sum_rt<=Mac_RT then
Return "No feasible output" and exit else proceed with step 3
Step 3: Iterative Up gradation
Randomly select one candidate from each SG and refer it as SS_new
Sum the T value of each candidate in SS_new as sum_t
Sum the C of each candidate in SS_new and RT of each candidate in SS_new as sum_c and sum_rt
If sum_c <=Max_C or sum_rt<=Max_RT then
If sum_t of SS_new>sum_t of SS then
Replace SS with SS_new
Repeat step 3 for Max_iteration times.
If number of iteration reaches Max_iteration then return SS and sum_t.

Service selection problem is a combinatorial optimization problem. As it is a online program, there is need to design a algorithm in such as way that it should d produce optimal solution or near optimal solution in the faster manner. Initially all the service candidates of each service group are sorted based on their trust value in the non-decreasing order. So that, the best service candidate in term on the trust value will be in the end of the list.

At first, first service candidate in each and every group is selected to form the result. The need for doing this is to avoid the local minima problem. The TCSS algorithm starts with the candidates which has lowest trust value. Hence, it is the candidate with lowest trust value, possibility of the constraint violation is minimal. If the selected candidates not satisfy the constraints such as maximum cost and maximum tolerable response time, then it quit with "No solution". Otherwise, it will go to the iterative step to linearly upgrade its value.

The import step of this TCSS is iteratively improving the result. In this step, every time, one candidate is selected randomly from each group. Then, the cumulative trust value is calculated. Cost of the selected candidates is summed together and response time of the selected candidates is summed together. Then, these values are compared with maximum cost and maximum response time of the service composition (mentioned in Service level agreement). These QoS values are referred as constraints. If the constraints are satisfied, then new list of the candidate is referred as the selected candidate. The same process is repeated with the selected candidate list in order to improve the optimality of the result. As it is a combinatorial optimization problem, it is hard to solve the problem in the linear time. This algorithm provides heuristic solution to solve the problem in the linear time.

5. Experiment and Analysis

The optimization objective is to maximize the trust with respect to the QoS constraints cost, execution. The execution platform of the proposed TCSS algorithm is a HP Pavilion G6 System with Microsoft Windows 7 Home Premium Operating System, Intel Core i3-380M Processor with the speed of 2.53 GHz, Dual Core System, L3 cache-3 MB, 64 bit computing, 8 GB RAM, DDR3 SDRAM and with the Hard disk with the capacity of 500 GB HDD/5400 rpm. The algorithm is implemented using C#.NET and executed in the Microsoft Visual Studio 2013.

This TE is implemented in C#.NET. TE estimates the direct trust value (DTV), Referral trust value (RTV) and Weighted Trust value (WTV) based on the Algorithm 1.

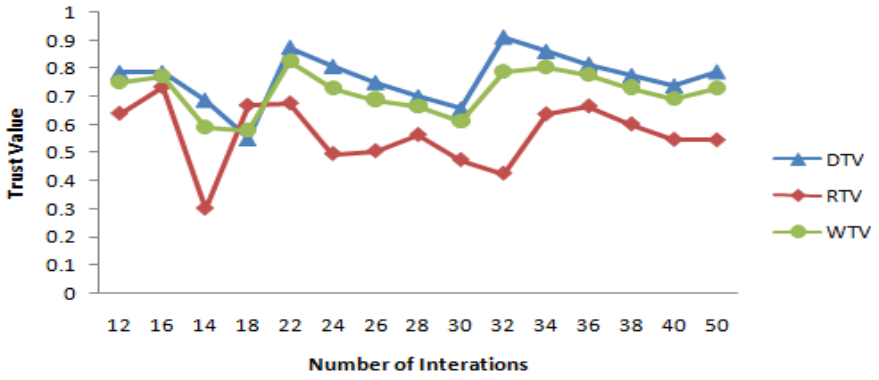

Figure. 2 Comparison of DTV, RTV and WTV

For performance evaluation, 2 QoS attributes (cost and response time for each service are randomly generated with a uniform distribution between [1,10]. I also generate the trust value of each candidate service is a uniform distribution between [0.0, 1.0]. Algorithm is tested with varying number of service classes and service candidates. Figure. 3 shows that if the number of service candidate in the service classes increases then the average trust value will get increased. The best optimum value is .99 is achieved by using TCSS Framework.

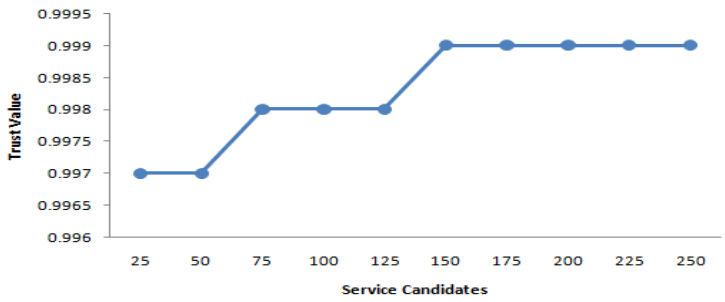

Figure 3 Trust Value vs Service Candidates

Figure. 4 shows that if the number of service classes in the composite service increases then the average trust value will get increased. The best optimum value is .99 is achieved by using TCSS Framework.

Figure. 4 Trust Value vs Service Classes

As the Service selection is the online optimization problem, the execution time of the TCSS should be minimum as compared to other service selection algorithms. TCSS algorithm is testes with different samples of service candidates and service classes. Estimated trust values and time taken to execute the TCSS is given in table.

Table 1 TCSS with Respect to Execution Time and Trust value

Service Groups	Trust Value	Execution Time
5	0.997	0.019
10	0.997	0.09
15	0.998	0.15
20	0.998	0.31
25	0.998	0.55

Table 1 and Figure 5 shows that the if the service groups increases then the execution time and optimality value get increases

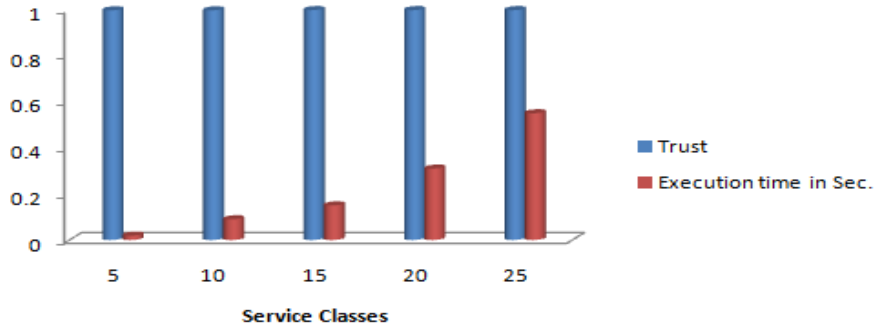

Figure 5 Service Classes vs Trust and Execution Time

6. Comparison with WS-HEU

TCSS is compared with existing well known heuristic based service selection algorithm WS-HEU proposed by Tao Yu [13]. WS-HEU is implemented by using MMKP approach. The proposed TCSS outperforms the existing algorithm based on the execution time and optimality value. Execution time of TCSS is lesser as compared to the execution time of WS-HEU. The optimality rate of LASA-HEU is approximately .006% better than the WS-HEU.

This experiment is carried out with different data sets such as service groups ranges from 5 to 50 with the step size of 5. Service Candidates ranges from 25 to 500 with the step size of 25. TCSS provides .006% better trust value as compared to WS-HEU. It is shown in the figure.

Figure 6. Performance Comparison of TCSS with WS-HEU

Figure 6 shows that WS-HEU shows rapid concave ups and concave downs in the graph. This is because of the downgrades and upgrades in the WS-HEU algorithm. This is not the case with TCSS. TCSS algorithm is implemented with feasible upgrades; there are no downgrades in the TCSS. So that TCSS provides the consistently better trust value as compared to WS-HEU. Table III shows the selected service groups, candidates and the performance results of the TCSS and WS-HEU.

Table 2. Performance Comparison

Service Groups	Service Candidates	WS-HEU
5	25	0.986
10	50	0.988
15	75	0.989
20	100	0.991
25	125	0.992
30	150	0.99
35	175	0.994
40	200	0.996
45	225	0.997
50	250	0.996

Table 2 shows that the proposed TCSS outperforms the WS-HEU based on the optimality and execution time. This is because of the elimination of the downgrades in TCSS. Figure 7 shows the execution time of WS-HEU and TCSS. Execution time for TCSS increases rapidly when the number of service classes increases. Similarly, execution time for TCSS increases rapidly when the number of service candidates increases. But, Execution time of WS-HEU decreases linearly when the service classes increases.

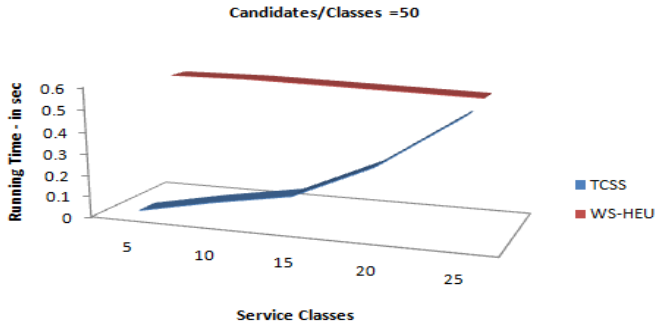

Figure 7 Execution Time of TCSS vs WS-HEU

7. Conclusions

Choosing the best service based on the QoS requirement of the consumer is a tedious task. Among the different QoS requirements, trust plays the vital role. In this paper, trust enforced service selection approach is proposed. The results show that the convergence time of TCSS is 2 times lower than the existing heuristic based service selection algorithm. The average trust is high as compared to the well known heuristic service selection algorithms. In this paper, Beta distribution is used to estimate the individual service trust value the trust based cloud service selection problem is represented in MMKP and new heuristic is developed to solve the trust based service selection problem. But, Composite services consist of several atomic services and theses services are connected using different workflow patterns. In this paper, workflow patterns are not considered for the service selection. Trust based cloud service selection problem can be represented as graph. It can be solve by using Multi constraint optimal path selection problem (MCOP) in future.

References

[1] M.P. Papazoglou, P. Traverso, S. Dustdar, Leymann, "Service-oriented computing: a research roadmap", International journal of Cooperative Information Systems, vol. 17, no. 2, **(2008)**, pp.223–255.
[2] L.-H. Vu, Hauswirth, M., Aberer, K, "QoS-based service selection and ranking with trust and reputation management", Springer LNCS, vol. 3760, **(2005)**, pp. 466–483.
[3] A. Josang, R. Ismail, C. Boyd, "A survey of trust and reputation systems for online service provision", Decision Support Systems, vol. 43, no. 2, **(2007)**, pp.618–644.
[4] D.A. Menasc´e, "Composing web services: A QoS view", IEEE Internet Computing, vol.8, no. 6, **(2004)**, pp.88–90.
[5] Wira D. Mulia, Naresh Sehgal, Sohum Sohoni, John M. Acken, C. Lucas Stanberry and David J. Fritz, "Cloud Workload Characterization", IETE Technical Review, vol. 30, no. 5, **(2013)**.
[6] M. B. Bashir, M. S. A. Latiff, A. A. Ahmed, A. Y. and M. E. Eltayee, "Content-based Information Retrieval Techniques Based on Grid Computing: A Review", IETE Technical Review, volume 30, no. 3, **(2013)**, pp.223-232.
[7] L. Li, Y. Wang, E. P. Lim, "Trust oriented Composite service selection and discovery", ICSOC Service Wave 2009, LNCS 5900, **(2009)**, pp. 50-67.
[8] L. Li, Y. Wnag, E. P. Lim, "Trust oriented Composite service selection with QoS constraints", Journal of Universal Computer Science, vol. 16, no. 13, **(2010)**, pp. 1720-1744.
[9] Z. M. Aljazzaf, "Trust based Service Selection", Dissertation, University of Western Ontario, **(2011)**.
[10] Z. Chen, L. Ge, H. Wang, X. Huang, J. Lin, "A trust based service evaluation and selection model in pervasive computing environment", International symposium on pervasive computing and application, **(2006)**.
[11] C. Diana, B. Harun, R. Roland, Z. Michael and G. Kurt, "Heuristic Approaches for QoS based Service Selection", Springer LNCS, **(2009)**, pp.441-455.
[12] R. Wang, Chi-Hung Chi, Jianming Deng, "A fast heuristic algorithm for the composite web service selection", Springer LNCS, **(2009)**, pp. 506-518.
[13] T. Yu, Y. Zhang, K. A. Lin, "Efficient Algorithms for Web Service Selection with end-to-end QoS constraints", ACM Transactions on the Web, vol. 1, no. 1, **(2007).**
[14] K. Thirunarayan, P. Anantharam, C. Henson, A. Sheth, "Comparative trust management with applications: Bayesian approaches emphasis", Future Generation Computer Systems, vol. 31, **(2014)**, pp. 182-199.
[15] F.K. Hussain, E.Chang, O.K. Hussain, "State of the art review of the existing Bayesian-network based approaches to trust and reputation computation", International conference on Internet Monitoring and Prediction, **(2007)**, pp.4.
[16] Y.L Sun, W.Yu, Z Han, Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks", IEEE Journal on Selected areas in Communications, vol. 24, **(2006)**, pp.305-316.
[17] D. Quercia, S.Hailes, L.Capra, "B-Trust: Bayesian trust framework for pervasive computing", Springer LNCS, vol. 3986, **(2006)**, pp.298-312.
[18] M.K Denko, T. Sun, "Probabilistic trust management in pervasive computing ",IEEE International Conference on Embedded and Ubiquitous Computing, **(2008)**, pp. 610-615,
[19] M.Hifi, M.Michrafy, A Sbihi, "A reactive local search baed algorithm for the multiple choice multi dimensional knapsack problem", Journal on computational optimization and applications, Springer, vol. 33, **(2006)**, pp. 271-285.
[20] M. Akbar, Manning, Shoja, Khan, "Heuristic Solutions for the Multiple Choice Multi Dimension Knapsack Problem", LNCS, vol. 2074, **(2001)**, pp.659-668.
[21] H. Kellerer, D. Pisinger, "Knapsack Problems", Springer Book, **(2004)**.
[22] Arjun K. Gupta, Nadarajah, "Hand book of Beta distribution and its applications", CRC press, **(2001)**.

