Trust Driven Workflow Scheduling By Composition of Cloud Services under Fuzzy Preferences of Users

Pavithra.M ME,*1 S.K.Mahalingam ME,Ph.D.,*2

*1 PG Scholar , Department Of Computer Science And Engineering, J.K.K.Nataraja College Of Engineering And Technology, Namakkal,Tamilnadu-63818. Mail id: pavithra.m79@gmail.com
*2 Research Scholar, Department Of Computer Science and Engineering, Anna University, Coimbatore. Mail id: skmahalingam1986@gmail.com

Abstract—The main objective of the current work is to implement Trust based multi-workflow scheduling strategy for composition of cloud services. The compatibility checking will be done whenever the different cloud services are integrated for improved performance. In this ontology-based approach is proposed to analyse Cloud service compatibility by applying reasoning on the expert knowledge. In addition, to minimize effort of users in expressing their preferences, the combination of evolutionary algorithms and fuzzy logic for composition optimization is applied.

Keyword: cloud service, trust, fuzzy logic

I INTRODUCTION

Cloud computing security or, more simply, cloud security is an evolving sub-domain of computer security, network security, and, more broadly, information security. It refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing.

On the cloud provider side, trust domains are differentiated according to the current existing single cloud platforms and providers rely on their trust agents to manage trust relationship. While on the cloud customer side, they will judge independently or depend on the credible trust intermediary institutions to help them manage trust relationship.

Workflow mechanism offers an effective method for realizing resource sharing and cooperation between the multiple dynamic changing virtual organizations. Resource allocation and task scheduling is one of the key problems in realizing workflow since it directly influences workflow’s execution efficiency.

Workflow can be described by directed acyclic graph (DAG) diagram or non-DAG method.

Workflow scheduling quality of service (QoS) is one of the most important factors influencing cloud service quality. In cloud, users’ QoS requirements generally can be divided into the performance QoS requirements and the trust QoS requirements.

The latest emergence of Cloud computing is a significant step towards realizing this utility computing model since it is heavily driven by industry vendors. Cloud computing promises to deliver reliable services through next-generation data centers built on virtualized compute and storage technologies. Users will be able to access applications and data from a “Cloud” anywhere in the world on demand and pay based on what they use.

Many high-performance computing (HPC) and scientific workloads (i.e., the set of computations to be completed) in cloud environment, such as those in bioinformatics, biomedical informatics, chem. informatics and geo-informatics, are complex workflows of individual jobs. The workflow is usually organized as a directed acyclic graph (DAG), in which the constituent jobs (i.e., nodes) are either control or data dependent (i.e., edges). Control-flow dependency specifies that one job must be completed before other jobs can start. In contrast, dataflow dependency specifies that a job cannot start until all its input data (typically created by previously completed jobs) is available. Control-flow is the more commonly used abstraction to reason about the relationship between different jobs, but we show how dataflow information is more valuable to effectively utilize the storage.

The performance QoS requirements often include the completion time (make span), consumption fees and implementation accuracy, etc. And the trust QoS requirements are referred to the security of providing services and the possibility or reliability of access to services.

The efficient workflow scheduling can lead to the trusted domain where the users will be satisfied with the high security and less execution time. By constructing the workflow...
efficiently the user demands like QOS requirement and Pricing strategies can also satisfied.

II LITERATURE SURVEY

Huankai Chen, Frank Wang, Na Helian[1] cellular automaton is used for modeling the complex cloud project scheduling system. Additionally, a method is presented to analysis the reliability of cloud scheduling system by measuring the average resource entropy (ARE). Furthermore, a new cost-efficient and reliable resource allocation (CERRA) model is proposed based on cellular automaton entropy to aid decision maker for planning projects on the cloud. Simply, here represent the cloud scheduling system behavior as a cellular automaton, specifically as a one-dimension cellular automata network. The algorithm used in this work is cost-efficient and reliable resource allocation (CERRA). The input given is number of allocated resources and the output achieved is Average Resource Entropy, project completion time, cost of allocated source, cost-efficiency and reliability rate. The experiments help demonstrate how the CERRA model can be implemented and interpreted, and how the CA Entropy-based solution can be introduced in a project manager's decision-making process. The experiment result shows that the proposed model is able to achieve both cost-efficient and reliable resource allocation solution for running project on the cloud.

Anupam Das and Mohammad Mahfuzul Islam[2] propose a feedback-based dynamic trust computation model named SecuredTrust which can effectively detect sudden strategic alteration in malicious behavior with the additional feature of balancing workload among service providers. Secured-Trust considers variety of factors in determining the trust of an agent such as satisfaction, similarity, feedback credibility, recent trust, historical trust, sudden deviation of trust, and decay of trust. Here used a novel policy of utilizing exponential averaging function to reduce storage overhead in computing the trust of agents. Here also proposed a new load-balancing algorithm based on approximate calculation of workload present at different service providers. The algorithm used in this work is SecuredTrust. The input submitted is number of agents and the output achieved are Trust computation error, average STR, number of times malicious agents are selected as service providers, computational time, workload, Scalability. Simulation results indicate, compared to other existing trust models, Secured-Trust is more robust and effective against attacks from opportunistic malicious agents while capable of balancing load among service providers.

Xiaoyong Li, Feng Zhou, and Xudong Yang[3] proposed a scalable feedback aggregating (SFA) overlay for large-scale P2P trust evaluation. First, the local trust rating method is defined based on the time attenuation function, which can satisfy the two dynamic properties of trust. The SFA overlay is then proposed from a scalable perspective. Not only can the SFA overlay strengthen the scalability of the feedback aggregation mechanism for large-scale P2P applications, but it can also reduce networking risk and improve system efficiency. More importantly, based on the SFA overlay, an adaptive trustworthiness computing method can be defined. This method surpasses the limitations of traditional weighting methods for trust factors, in which weights are assigned subjectively. The algorithm used in this work is scalable feedback aggregating (SFA) overlay. The input submitted is numbers of collusive malicious peers, P2P system size and the corresponding output is Global convergence time (GCT), Average messaging overhead (AMO), RMS error. Through theoretical and experimental analysis, the SFA-based trust model shows remarkable enhancement in scalability for large scale P2P computing, as well as has greater adaptability and accuracy in handling various dynamic behaviors of peers. Here shown that our system yields very good results in many typical cases and that the proposed mechanism is robust against various complicated environments.

Guanfeng Liu, Yan Wang, Mehmet A. Orgun, and Ee-Peng Lim[4] present a novel complex social network structure incorporating trust, social relationships and recommendation roles, and introduce a new concept, Quality of Trust (QoT), containing the above social information as attributes. Here then model the optimal social trust path selection problem with multiple end-to-end QoT constraints as a Multiconstrained Optimal Path (MCOP) selection problem, which is shown to be NP-Complete. To deal with this challenging problem, here propose a novel Multiple Foreseen Path-Based Heuristic algorithm MFPB-HOSTP for the Optimal Social Trust Path selection, where multiple backward local social trust paths (BLPs) are identified and concatenated with one Forward Local Path (FLP), forming multiple foreseen paths. This strategy could not only help avoid failed feasibility estimation in path selection in certain cases, but also increase the chances of delivering a near-optimal solution with high quality. The algorithm introduced in this work is Multiple Foreseen Path-Based Heuristic algorithm, MFPBHOSTP. The input given is Enron e-mail corpus dataset and the corresponding output is execution time, Path Utility. Here conducted extensive experiments on a real online social network data set, Enron e-mail corpus, which is formed by sending and receiving e-mails between participants. Experimental results have demonstrated the good performance of our proposed algorithm MFPB-HOSTP.

Zheng Yan, and Christian Prehofer[5] adopt a holistic notion of trust, which includes availability, reliability, integrity, safety, maintainability, and confidentiality, depending on the requirements of a trust or. Based on this, here propose an autonomic trust management solution for component-based software systems focusing mainly on system runtime. Applying the proposed trust control model, here predict trustworthiness and select suitable control mechanisms for managing trust in an autonomic approach. Here further design a number of algorithms, adoptable by a trust management framework, for autonomic trust management during component execution. The trust model presented here differs from prior work as it considers the trust control mechanisms’ influence in order to support autonomic trust management according to the system’s competence. It is not a model only for the purpose of trust
evaluation and decision support, but also autonomic trust maintenance. The methodology used in this work is adaptive trust control model, Trustworthiness Prediction and Control Mode Selection, Trust Assessment at Runtime, Adaptive Trust Control Model Adjustment algorithm. The input submitted and output retrieved are as follows control mode selection and Runtime, Trustworthiness prediction. Here applied the FCM theory into trust modeling and showed its practical effectiveness for autonomic trust management through case study. The solution’s implementation is realized on the basis of a number of algorithms for trustworthiness prediction and control mode selection, trust assessment, and trust control model adjustment. These algorithms can be adopted by the trust management framework that works as a delegate to manage the trustworthiness of the trustee entity based on the criteria set by the trust or entity. Here reported simulation-based experimental results to verify the proposed algorithms and demonstrated both the effectiveness and benefits.

Yan Wang and Lei Li[6] present a novel two dimensional aggregation approach consisting of both vertical and horizontal aggregations of trust ratings. The vertical aggregation calculates the aggregated rating representing the trust level for the services delivered in a small time period. The horizontal aggregation applies our proposed optimal algorithm to determine the minimal number of time intervals, within each of which a trust vector with three values can be calculated to represent all the ratings in that time interval and retain the trust features well. Hence, a small set of trust vectors can represent a large set of trust ratings. This is significant for large-scale trust rating transmission and trust evaluation. The algorithm used in this work is Optimal MTI algorithm, Greedy MTI algorithm, and Vertical rating aggregation algorithm. Incorporating the results in both Experiments 4 and 5, the greedy algorithm is useful and more efficient when processing rating data with less frequently changing trust trend. In contrast, the optimal algorithm can outperform when processing large-scale rating data with a high VSPCL threshold.

Alejandro Martínez, Francisco J. Alfaro, Jose´ L. Sa´nchez, Francisco J. Quiles, and Jose´ Duato[7] shown that it is possible to achieve a more than acceptable QoS performance with only two VCs. Here reuse at the switches the scheduling decisions made at the network interfaces. This VC reduction opens up the possibility of using the remaining VCs for other concerns like adaptive routing or fault tolerance. Furthermore, it is also possible to reduce the number of VCs supported at the switches, thereby simplifying the design or increasing the number of ports. Here proposed a switch design that benefits from that proposal. The algorithm used in this work is switch design, Cost-Effective Technique. The inputs submitted are normalized load, throughput and the output collected are latency, cumulative distribution function (CDF) of latency. Here shown that this proposal is able to scale properly. The results with large network sizes are almost as good as with smaller networks and the difference keeps constant. Here also examined the power consumption for different interconnects, ranging from 64 to 512 end nodes. For all of them, we cut half of the power consumed by the inter connect.

Nirmalya Roy and Sajal K. Das[8] use of the Grid as a candidate for provisioning computational services to applications in ubiquitous computing environments. In particular, here present a competitive model that describes the possible interaction between the competing resources in the Grid Infrastructure as service providers and ubiquitous applications as subscribers. The competition takes place in terms of quality of service (QoS) and cost offered by different Grid Service Providers (GSPs). Here also investigate the job allocation of different GSPs by exploiting the non cooperativeness among the strategies. Here present the equilibrium behavior of the model facing global competition under stochastic demand and estimate guaranteed QoS assurance level by efficiently satisfying the requirement of ubiquitous application the algorithm used in this work are Grid Service Provider algorithm and Local Participating Players algorithm. The input submitted is number of tasks and the corresponding output is TimeSpan, Response time, Delay ratio, Rejection rate, Utilization, Loss probability. Here performed extensive experiments over Distributed Parallel Computing Cluster (DPCC) and studied overall job execution performance of different GSPs under a wide range of QoS parameters using different strategies. This model and performance evaluation results can serve as a valuable reference for designing appropriate strategies in a practical grid environment. The mathematical and performance results in this work offer the theory and practice toward the design and evaluation of practical computational and business grid for ubiquitous computing applications.

Zhiqiang Wei, Wei Zhou, Mijun Kang, Michael Collins, Paddy Nixon[9] model the trust based interaction decision process as a public decision process. Each recommender give its opinion, which reflects its own trust value on the service provider, on the decision about whether the requestor should interact with the service provider and receive some credits from the requestor as incentive. A VCG (Vickrey-Clarke- Groves) based strategy-proof trust mechanism is proposed to address the problems described above. In this mechanism, the user will maximize its profit only when it offers a truthful recommendation. This trust mechanism can utilize the graded trust value as well as a binary trust value system and is effective with all kinds of strategic manipulation. Moreover, the proposed mechanism is efficient because the recommender will receive payments before the actual outcome of the interaction with the service provider can be observed. This property is essential for the incentive mechanism in the mobile environment; since sometimes the desired observation may take such a long time that the mobile user cannot wait to get the payments. The simulation results show that the mechanism is effective regarding both positive and negative deviations as well as strategic manipulations. The evaluation indicates that it is impossible for the lying node to gain more utilities by adopting some deviation type and manipulating its opinion in one time recommendation.
III COMPOSITION OF CLOUD SERVICES

In proposed system Composition based trust driven scheduling algorithm is presented in order to achieve the trust driven and QoS aware scheduling for the composition of cloud services.

The composition of cloud services is done when the given single cloud service is not enough to satisfy the user requirement like memory storage, QoS requirement etc. The composition of cloud services will be done effectively without affecting each other. i.e. without affecting process of one cloud service with process of another cloud service with any of its properties by compatibility checking.

Trust-based multi-workflow scheduling algorithm uses trust mechanism to filter dishonest providers and then binds workflow to the provider whose service capability is closest to their demand. Assume that providers’ number is m and workflows’ number is n.

Compatibility checking in service provider for each workflow with multiple constraints will be done by finding pareto front list. Finally ranking will be given to each pareto front solution by using fuzzy ranking approach.

Composition based Trust-driven scheduling algorithm:

Input: workflowList (workflows requesting for cloud services), providerList (cloud providers), tLevel (trust level in the transaction).
Output: Composition of cloud service scheduling results.
Main steps:

1. Delete unqualified providers from providerList according to current trust level tLevel and generate reliable suppliers list named trustProviderList;
2. Calculate the QoS and price factor’s for given workflow
   a) Calculate each workflow’s service tolerate price QTP (QoS tolerate price) in workflowList according to DQoS (Described QoS) vector;
   b) Calculate each reliable provider’s resource provision price QPP (QoS Provision price) in trustProviderList according to PQoS vector;
   c) Calculate SDS (Service demand similarity) using the Euclidean distance method and generate demand service similar matrix V.
3. Search for each workflow, its set of highest similarity provider
4. Evaluate valid composition by checking compatibility against the parameters Composition (c), Constraint List (cl).
   a) if CompositionValidity (c, cl) Exists in cache
      i. ValidComposition = GetCompositionValidityFromCache (c, cl)
   b) ValidComposition = True;
   c) Foreach workflow w and service provider s in c do
      i. Foreach Constraint t in cl do
         1. If Compatibility (t, w, s) Exists in Cache then
            a. ValidComposition = GetCompatibilityFromCache (t, w, s);
         2. End
         3. Else
            a. ValidComposition = CheckCompatibilityByReasoning (t, w, s)
         4. End
         5. InsertCompatibilityToCache (t, w, s)
      ii. End
      iii. If ValidComposition = False then
         1. Break;
      iv. end
   e) end
   f) InsertCompositionValidityToCache (c, cl);
   g) Return ValidComposition;
5. Bind the workflow with the provider’s resource set if the provider’s trust is higher than trust threshold and if it is a valid composition.

A. Cloud Setup

In this module, cloud setup will be done by using the cloudSim toolkit. The number of cloud service providers will be created who will provide the various services to the users. And then the user requirement gathering module will be created
through which one will gather the information about the tasks for which they want to utilize the cloud environment. Then the needed virtual environment will be created for processing the user queries through the network.

B. Trust Evaluation

In this module trust value will be calculated. Since trust relationships are classified into direct trust and recommended trust, the computation of trust should also be treated differently according to its obtained method direct or indirect. For cloud entities, they should calculate integrated trust also. The computation method can be abstracted as follows:

\[
IT = \alpha \times DT + (1 - \alpha)RT
\]

Here DT is the direct trust degree, RT is recommended trust degree, \(\alpha\) and \((1-\alpha)\) is the weight of DT and RT respectively.

The computation of direct trust

The direct trust is obtained by direct transaction history between cloud entities. The computation method of direct trust (cloud entity A–B in transaction context tc).

\[
DT(A, B, tc) = \frac{\text{SuccessTradeNum}}{\text{TotalTradeNum}}
\]

Here SuccessTradeNum represents the success transaction times and TotalTradeNum means the total transaction times.

The computation of recommended trust

When cloud entities want to trade with unfamiliar ones, they should use or combine recommended trust to aid the trust decision. The method of compute recommended trust is.

\[
RT(A, B, tc) = \frac{\sum_{\Omega} DT(A, B, tc)}{\text{Length}(\Omega)}
\]

Here \(\Omega\) represents the transaction recommendation set that entity trust.

C. Multi-objective evaluation of workflow

In our work we consider three user objectives, the lowest cost, quickest deployment time, and the highest reliability. This makes it infeasible to find an optimal composition as these objectives can conflict with each other. One way to address this problem is to convert the appliance composition problem to a single-objective problem by asking users to give weights for all the objectives. However, this approach is error-prone and impractical, as not all the users have the knowledge to accurately assign weights to objectives. Furthermore, since the composition solutions will depend on the capability of users to assign proper weight to the objectives, additionally we have to find a way to evaluate the knowledge of users about each objective to ensure the accuracy of the approach.

D. Fuzzy ranking for the reliable cloud services

Our proposed fuzzy inference engine includes three inputs and one output. Inputs of the system are normalized deployment time, deployment cost, and reliability of composition, which are all described based on the same membership functions. Output of the fuzzy engine represents how desirable the current set of inputs are based on the fuzzy rule-based indication. It shows the membership function for output by which we allow the gradual assessment of the membership of elements in a set. For example, the value “0” in output means the solution is highly undesirable whereas the value “1” shows that the solution is highly desirable. Fuzzy rules should be defined by the user to describe their preferences. For example a rule can be defined as: if DT is low and DC is low and Reliability is high, composition is highly desirable.

E. Scheduling workflows

When more than one workflow (belonging to different cloud users) ask for cloud services at the same time, it binds the user (workflow) to the most similar provider’s resource set in SDS to achieve the target of providing on-demand services. Trust-based multi-targets QoS workflow scheduling algorithm named C Multi-QoS Schedule is as follows:

Input: workflow List (workflows requesting for cloud services), provider List (cloud providers), t-Level (trust level in the transaction).

Output: scheduling results.

Main steps:

- Delete unqualified providers from provider List according to current trust level tLevel and generate reliable suppliers list named trust Provider List;
- Calculate each workflow’s service tolerate price QTP in workflow List according to D QoS vector;
- Calculate each reliable provider’s resource provision price QPP in trust Provider List according to P QoS vector;
- Calculate SDS using the Euclidean distance method and generate demand service similar matrix V.
- Search for each workflow its highest similarity provider and bind the workflow with the provider’s resource set if the provider’s trust is higher than trust threshold.

Trust-based multi-workflow scheduling algorithm uses trust mechanism to filter dishonest providers and then binds workflow to the provider whose service capability is closest to their demand.

F. Performance evaluation
In this module, performance evaluation of our work is done by comparing the proposed methodology with the existing methodology by using the performance metrics of trust value, QoS satisfaction.

IV CONCLUSION

In order to provide the better cloud service with the agreed QoS level, in this work, the Trust value based workflow scheduling by using the cloud service composition is introduced. The cloud service composition is nothing but the integrating the services from the more than two cloud service providers for satisfying the cloud users requirements. To ensure the QoS satisfaction level, in this work, trust value calculation is introduced which ensures the execution success rate of critical tasks and the introduction of two-level based scheduling mode reduces the problem scale of workflow scheduling. Multi-workflow scheduling helps to realize the QoS requirements analysis and service customization as the unit of user. The experimental results prove that the proposed methodology can provide the better result than the existing methodology.

V FUTURE WORK

However, the testing and application of this program has still been in the simulation experimental stage till now. So in the future, it will be feasible to build a small cloud prototype system and realize the deployment of our strategy in the real environment to test its efficiency and effectiveness.

REFERENCE

10. WenAn Tan, Yong Sun, Ling Xia Li, GuangZhen Lu, and Tong Wang, “A Trust Service-Oriented Scheduling Model for Workflow Applications in Cloud Computing”, Systems Journal, IEEE (Volume:8, Issue: 3 ), 26 June 2013