Ensuring Cloud Service Guarantees Via Service Level Agreement (SLA)-based Resource Allocation

Kaiqi Xiong
Department of Computing Security
College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14623, USA

Xiao Chen
Department of Computer Science
College of Science and Engineering
Texas State University
San Marcos, TX 78666, USA

***Abstract*— This paper studies the problem of resource management and placement for high performance clouds. It is concerned with the three most important performance metrics: response time, throughput, and utilization as Quality of Service (QoS) metrics defined in a Service Level Agreement (SLA). We propose SLA-based approaches for resource management in clouds. Specifically, we first quantify the metrics of trustworthiness, a percentile of response time, and availability. Then, this paper provides the formulation of cloud resource management as a nonlinear optimization problem subject to SLA requirements and further gives the solution of the problem. Finally, we give a solution of this nonlinear optimization problem and demonstrate the effectiveness of proposed solutions through illustrative examples.**

I. Introduction

Cloud computing has been one of the most successful technologies in computer science and engineering for the past few years. It has recently played in an extremely important role in the development of science and engineering that is closely and directly related to our daily life. Cloud computing provides applications, platform and infrastructure as a service and these services are referred to as infrastructure-as-a-service (IaaS), software-as-a-service (SaaS), and platform-as-a-service (PaaS) [34]. Cloud computing is used to increase the sharing of computing resources and reduce the cost of e-business services. It is done on a pay-per-use or charge-per-use basis. More and more companies such as Amazon, Google, Microsoft, and IBM are offering clouds to serve a variety of services for business, research and education nowadays [12].

Virtualization technology in cloud computing significantly improves the performance and flexibility of computing systems but dramatically increases the vulnerability of cloud services [31] and [32]. Cloud computing deploys services in highly distributed environments. Services from different cloud service providers at different locations are often federated into composite services subject to a Service-Level-Agreement (SLA) [20]. The SLA defines QoS requirements that cloud providers promise to offer and a price that cloud users are willing to pay for received services. QoS usually includes security, performance, and availability. These QoS requirements are studied as follows.

In this paper, we consider typical SLA metrics (called *QoS metrics* alternatively) and give their definitions according to security, performance, and availability requirements.

Security can be categorized as identity security and behavior security. Identity security includes data confidentiality, data integrity, authentication, and authorization between a customer and a service provider. Behavior security describes the trustworthiness among multiple virtual and physical resource sites and the trustworthiness of these resource sites by customers including the trustworthiness of cloud computing services or results provided by these sites. The trustworthiness of cloud services is usually characterized as *trust*. It is defined as a belief of cloud services that cloud users received. It is often quantified as values ranging from 0 to 1. "0" means that cloud users do not trust cloud services they received. Conversely, "1" means that cloud users 100% trust cloud services they received

Furthermore, performance requirements are often associated with response time, throughput, and utilization, which are defined as follows:

- *Response time* is the time for a service request to be satisfied. That is, this is the time it takes for a service request to be executed on the service provider's multiple resource sites [14], [28].
- *Throughput* is the service rate that a service provider can offer. It is defined by the maximum throughput or by the undergoing change of throughput with service intensity.
- *Utilization* is the percentage of time that cloud resources are utilized.

Finally, *availability* means that the information of a cloud computing system is accessible to those who have been authorized to have access at appropriate times. It is defined as the percentage of time that cloud providers are able to offer cloud services to cloud users. Certainly, there may be many other requirements, for example, usability, adaptability, scalability, and survivability. They will be similarly studied like availability.

In this paper, we consider how to allocate sufficient computing resources but not to over-provision these resources to process and analyze audit logs for ensuring the guarantee of an SLA, referred to as *the SLA-based resource allocation problem* for high-performance cloud auditing. An SLA is

a contract negotiated and agreed upon between cloud users and cloud providers. It defines Quality of Services (QoS) that cloud providers promise to offer and a price that cloud users are willing to pay for received services [36].

This paper gives a study of the SLA-based approaches for resource management in high-performance cloud auditing. We present our solution for the SLA-based resource allocation problem in high-performance cloud auditing. This paper is organized as follows. The SLA-based resource management problem is given in Section II. Section III gives an overview of the existing approaches for SLA-based resource management. Section IV presents the solutions of SLA-based resource management for cloud auditing. Finally, we conclude our discussion and give future work in Section V.

II. The SLA-based Resource Management Problem for Cloud Auditing

In this section, we consider resource management for high-performance cloud auditing subject to SLA metrics as discussed in Section I, referred to as the *SLA-based resource management* for high-performance auditing [29]. The resulting SLA-based resource management problem is defined later.

To ensure the security of cloud resources and the secure delivery of cloud services, cloud auditing has become very important and necessary. Audit logs are a chronological sequence of computing system records. Auditing data will be collected at different locations among multiple organizations across different cloud service providers. Thus, it is necessary to allocate resources for processing those cloud auditing data efficiently and effectively [10].

In the SLA-based resource management [6] for high-performance cloud auditing, we consider a cloud-computing environment in which multiple cloud-service providers work together to process cloud auditing data distributed in their domains for cloud security and risk analysis. While collected data can be from different locations, cloud service providers are better and must collaborate each other to analyze those data and maximize the findings of security and risk analysis. As shown in Figure 1, there are m cloud sites or locations in which cloud auditing data is collected and stored by cloud service providers. In the SLA-based resource management, these cloud service providers work together to offer composite services for security and risk analysis under the predefined QoS requirements and a price defined in an SLA. Assume that these cloud service providers own N_1, $N_2, \cdots$, and N_m physical, virtual, or both servers to offer services, where integer $m > 0$. In *this SLA-based resource management problem* for cloud service composition is to seek for positive integers n_1, $n_2, \cdots$, and n_m from these cloud service providers at m sites to ensure the guarantee of QoS requirements for cloud services, where n_j is less than or equal to N_j for $j = 1, 2, \cdots, m$.

In this paper, we will specifically consider the problem of how to manage cloud resources for analyzing audit logs in a cloud computing system where VM audit logs are distributed in multilevel security servers across multiple security domains as shown in Figure 1. In this figure, each security domain has multiple VM clusters consisting of VM audit logs that are distributed in multilevel security servers. Our research question is how to allocate sufficient computing resources but no over-provision for analyzing cloud audit logs so that the SLA agreed between cloud providers and users can be guaranteed. The above resource management problem is called *the SLA-based cloud/virtual resource management problem* (simply referred to as *the SLA-based resource management*) for high-performance cloud auditing. It will address the challenge of finding the minimal number of *virtual servers or machines* (VMs) for ensuring QoS requirements in high-performance cloud auditing.

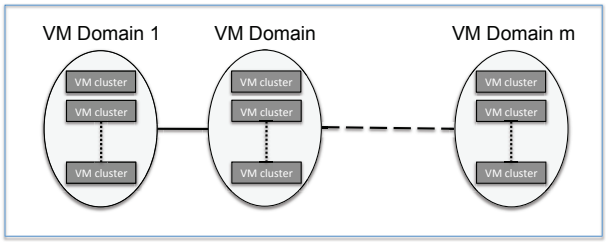

Fig. 1. Virtual Machine (VM) Resources for High-performance Cloud Auditing

III. Related Work

In this section, we give an overview of existing approaches for SLA-based resource management in a variety of computing systems including a cloud computing system.

Cloud computing is used to increase the sharing of computing resources and reduce the cost of e-business services. It is done on a pay-per-use or charge-per-use basis. The SLA-based resource management for high-performance cloud auditing will play a key role in ensuring the security and successful delivery of cloud services since more and more companies such as Amazon, Google, Microsoft, and IBM are offering clouds to serve a variety of services for business, research and education nowadays.

As stated above, the SLA-based resource management for high-performance cloud auditing will address the challenge of finding the minimal virtual resources to ensure QoS requirements in cloud auditing data analysis and risk management. QoS-based resource allocation has been extensively studied (e.g., see [1], [3], [7], [10], [17], [21], [24], [35]) where QoS metrics include response time, bandwidth, cluster utilization, availability, or packet loss rate. In [3], the resource allocation of the networks between a user and data centers has been studied for the assurance of QoS requirements. The end-to-end QoS includes an end-to-end delay (i.e., response time). Chassot et al. only discussed and measured the maximal, minimal and average values of the response time. Optimizing static workload allocation based on mean response time and mean miss rate has been studied for multi-cluster computing systems in [17]. A price-based job allocation scheme was proposed for grid systems in [8].

Moreover, while Martin and Nilsson [6] only measured the average response time, Menasce and Casalicchio in [21] considered the average response time in their model as well. These QoS metrics can be estimated by using measurement techniques [2], [3], [9], [18]. However, the SLA-based cloud resource management problem is very new in the field. It has been proven a very challenge and important problem [2], [3].

Among these QoS metrics, the calculation of response time often becomes critical in solving the SLA-based cloud resource management problem. Typically, response time is taken into account through its mean in the literature. However, this may not be a meaningful quality of service as far as the customer is concerned, who may be more interested in a statistical bound of the response time. For instance, a customer can request that 95% of the time its response time should be less than a desired value. Martin and Nilsson [6] measured the average response time of a service request. A framework for service management in grid computing was defined in [3], [21], but they did not provide a method for calculating the probability distribution of the response time. In order to compute a percentile of the response time one has to first find the probability distribution of the response time. This is not an easy task in cloud computing where there are many virtual resources and there are many different types of users.

As is seen above, the calculation of end-to-end QoS metrics including a percentile of response time, bandwidth, and service availability is very complicated. Hence, the above SLA-based cloud resource management problem has been proven very challenging. Our methodology for solving the SLA-based resource management problem will be discussed for high-performance cloud auditing in next section.

IV. The Solutions of the SLA-based Resource Management Problem and the Placement Problem for Cloud Auditing

In this section, we propose approaches for SLA-based resource management for high-performance cloud auditing.

As is seen above, we have presented an SLA-based resource management problem in high-performance cloud auditing where we are required to calculate the number of service resources required to ensure that QoS requirements including performance, availability, and security are met subject to a given fee. For example, the response time of a service request meets the requirement of a predefined percentile response time under a given fee. The SLA-based resource optimization problem can be constructed by minimizing the total cost of service providers while satisfying SLA guarantees. It will be formulated as a mathematical minimization problem later.

Before a service is processed in cloud computing, a trust server is responsible for selecting m clusters to execute the service from a pool of clusters. In this section, we first use a queueing network to model m VM clusters (called *sites* alternatively in this paper since different clusters are typically located in different sites) and formulate the SLA-based resource management as a resource optimization problem. Then, we demonstrate how to develop an algorithm to solve the SLA-based resource optimization problem. Without any confusion, we reuse m ($0 < j \leq m$) as the number of VM clusters necessary for processing a customer's service job. We only consider the case of single customer services in this paper. Our discussion below can be easily extended to the case of multiple priority customer services.

The objective of this research is to find the number of servers n_j from the pool of VMs in each VM cluster such that QoS can be guaranteed. That is, the problem can be formulated as:

$$\min_{n_1,\cdots,n_m} \sum_{j=1}^{m} c_j n_j \tag{1}$$

subject to QoS requirements where c_j and n_j are the operation cost of a VM and the number of VMs required to ensure QoS guarantee in high-performance cloud auditing, respectively.

As we see, the objective function of the above problem is linear, but QoS requirements are somewhat complicated. Typically, these requirements are implicit and nonlinear with respect to the number of VMs n_j ($j = 1, 2, ..., m$). That is, the above problem is usually nonlinear constrained optimization, so it is difficult to solve. As an example, we consider trustworthiness, the percentile of response time and availability in this paper since these two metrics are very important in SLAs nowadays.

Trustworthiness of virtual resources and services: As discussed before, "Trust" is used to deal with the notion of the trustworthiness in behavior security. In this paper, trust is defined as a firm belief in the competence of a virtual resource such as a VM that acts as expected. It has been extensively studied for a variety of computing systems in the literature (e.g., see [4], [15], [25], [26]).

In this paper, we are only interested in the trustworthiness of virtual resources from a customer's perspective. Assume that a trust manager is a trusted agent who represents customers. The trust manager uses the collected trustworthy information regarding the clusters to evaluate their security behavior. We consider security behavior by modeling the behavior trusts of all clusters, and quantify the trustworthiness of these clusters using a rank and a threshold-based approach. This approach is based on previous job completion experience assessed by the trust manager and customers. The assessment may also include the opinion of other trust managers besides its own customer's feedback in the case of multiple security domains in which multiple trust managers are needed. The domain of feedback is assumed to be the interval: [0, 1]. Let $r_j(k)$ be the trust index of Cluster j at time $t = k$.

The trust function describes the trustworthiness of clusters monitored and updated by the trust manager. Therefore, the trust function reflects a probabilistic security behavior of the resource clusters from a customer's perspective.

The percentile of response time. The response time is the time it takes for a service request job to be executed on the service provider's cluster nodes and then sent its

completed job back to the customer. Let T be a random variable representing the response time, and let $f_T(t)$ and $F_T(t)$ be its probability and cumulative distributions pdf and CDF, respectively. Also, let T^D be the desired target response time that a customer requests and agrees with its service provider based on a fee paid by the customer. The statistical bound on the response time can be expressed by

$$F_T(t)|_{t=T^D} = \int_0^{T^D} f_T(t)\,dt \geq \gamma \;\; (0 \leq \gamma \leq 1) \qquad (2)$$

which is called *percentile response time*. This means that $\gamma \times 100\%$ of the time a service request job will be executed in less than T^D.

In the SLA-based cloud resource management problem, we can model those m VM clusters as a queueing network. Without loss of generality, we assume that the m VM clusters are modeled as a tandem queueing network consists of m stations numbered sequentially from 1 to m, each station representing a VM cluster. Moreover, each cluster j is modeled as a single FIFO queue served by n_j identical servers, each providing a service at the rate μ_j. An infinite server is used to model the propagation delay of the networks among servers and users. Let Λ be the external arrival rate to the infinite server, and let λ and λ_j be the effective arrival rates to the infinite server and station j ($j = 1, 2, ..., m$). The notion of server here is defined as a service resource at each VM cluster that processes users' jobs. We assume that all service times are exponentially distributed and the external arrival to the trust server occurs in a Poisson fashion. The trust server provides a service at the rate μ. Therefore, a percentile of end-to-end response time for cloud services can be subsequently derived by using the Laplace-Stieltjes transform (simply referred to as the Laplace transform. See [35] for details).

Availability: It is a critical metric in today's computer design (see Hennessy [11]). Brown and Patterson [5] used an availability metric to describe variations in system quality of service over time. It is defined by the response time of a request service and the number of failures that can be tolerated by a system. The former was discussed as above. So, we only need to study the latter in this section. We consider the percentage of time that a resource is "up" or "down" as a metric, which is a traditional way to define service availability.

Network availability data may be also found over the Internet. For example, the University of Houston maintains current and historical network availability, see [19].

The Solution of the SLA-based Resource Management Problem with an Illustrative Example:

In this research, we adapt a decompose strategy to solve the above SLA-based resource management problem for high-performance cloud auditing. Specifically, we propose the following procedure for obtaining the solution:

1) Select m resource sites within a predefined trust index at time $t = t_k$.
2) Solve for n_j in the m-dimensional integer optimization problem of (1) under the constraint of a percentile response time of (2), and the constraint of service availability:
$$P_j(n_j, N_j) \geq \delta_j\% \qquad (3)$$
where T^D is a desired response time defined by a customer and δ_j is a desired percentage of service availability at site j. Furthermore, $P_j(n_j, N_j)$ is the service availability of cluster j in which n_j out of N_j servers are in operation.
3) Update the trust indices of all m sites based on the activity during the time interval $[t_k, t_k + T^D]$. Then, the trust manager decides if a completed job is accepted. If each trust index at those selected sites which complete the service job meet a predefined index value, then the completed job is accepted. Otherwise, then the completed job is discarded and the trust manager needs to resubmit the job.

The above procedure is called the *SLA-based approach*. As is seen, the procedure ensures the trustworthiness of virtual resources, the percentile of response time, and service availability step-by-step. Specifically, in this procedure, Steps 1 and 3 guarantee that cloud services are obtained from reliable VMs and Step 2 ensures the guarantee of the percentile of response time and service availability. It is a key step among them. The constrained nonlinear optimization problem in this step can be converted into non-constrained nonlinear optimization based on the interior method.

Example: We study a case of ten VM cloud clusters belonging to ten different cloud providers. We assume that all cloud services need to be sequentially processed in seven of ten VM clusters only, as depicted in Figure IV where let us recall that Λ represents an arrival rate of cloud services. That is, $m = 7$.

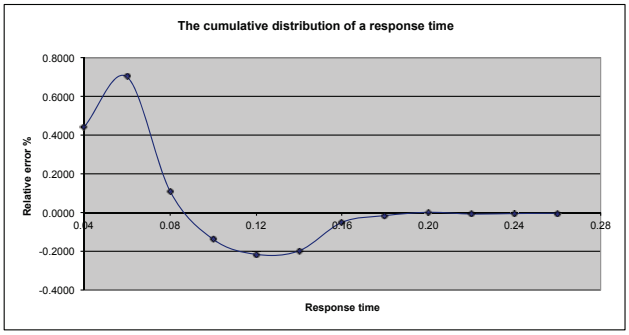

Fig. 2. A Model Consisting of a Trust Server and Multiple Cloud Clusters for Cloud Service Processing

In Steps 1 and 3 of the above procedure, we need to constantly update the trust index of each service cluster. We consider the discrete time t_1, t_2, ..., t_k,... in an increasing order ($k = 1, 2, \cdots,$). For each cluster j, its trust index at time t_k is defined as a weighted sum of its trust index at time t_{k-1} and a satisfactory ratio for services completed at time between t_{k-1} and t_k. The satisfactory ratio is equal to the number

of satisfactory services assessed by customers (i.e., through customers' feedback) divided by the total number of service submissions from time t_{k-1} and t_k.

Let us choose $\xi = 0.6$ and the trust index at time t_1 is listed in Table I.

Assume that $r_j(k)$ is uniformly distributed in [0.75, 1]. r_i and r_j are independent for any $i \neq j$ $(i, j = 1, \cdots, 10)$. Furthermore, we choose $\lambda = 100$, $T^D = 0.16$, $\gamma = 0.975$, $\mu = 200$, and the service rates of these ten clusters are given in Table II.

As discussed above, each VM cloud cluster is modeled as a single queue in this proposed SLA-based approach. This means that all low-capacity servers of each cluster is regarded as a single high-capacity server in the queue. In this example, we assume that the server scaling factor of cloud cluster j is $\psi(n_j) = 1.5^{\log_2 n_j}$, that is, the service processing capacity of cloud cluster j with a server service rate of μ_j is approximated as that of the single server with a service rate of $\psi(n_j)\mu_j$. We also choose $c_j = 2$, $N_j = 500$, $\delta_j = 99.999$, $\hat{I}_j = 0.95$ $(j = 1, \cdots, 10)$, and the server unavailable rates of these ten clusters are listed in Table III.

A customer submits a service job at time t_8 with $t_{k+1} - t_k = 0.01$ $(k = 1, 2 \cdots)$ and it requires two of these 5 resource sites satisfying the predefined $\hat{I}_j$ for processing the job.

First, we generated I^{t_k} $(k = 2, \cdots, 5, \cdots, 21)$ in Matlab. As we see, sites 1, 2, 3, 4, 6, 7, 8, and 10 meet the trust requirement at $t = t_8$. Thus sites 2, 3, 4, 6, 7, 8, and 10 are selected because they have the highest seven trust indices.

Then, we calculated a percentile of end-to-end response time for cloud services by using the Laplace transform (see [35]). Figure IV gives the cumulative distribution of response time obtained from the proposed SLA-based approach for cloud resource management. We further simulated the model consisting of seven cloud clusters in Figure IV by using Arena 7.01. The simulation results are considered as "exact" since the simulation model is an exact representation of the queueing network under study.

Moreover, we obtained that $\hat{a}^{(1)} = 100$, and the optimal number of servers required for 99% of the response time to be less than $T^D = 0.18$ is shown in Table V. The exact optimal number of servers, obtained by exhaustive search using the simulation model, and assuming that each cluster has the same utilization, or balanced utilization, is consistent with the ones shown in Table V. We validated that they are consistent with the result obtained by a brute force search using the simulation model in Arena, and assuming that each cluster has the same utilization, or balanced utilization.

Furthermore, we obtained the number of servers required for 99.999% service availability in these seven clusters using Step 2 of the above proposed procedure, as shown in Table V. By doing the calculation in Step 3 of of the above proposed procedure, we obtained the numbers of servers required for the response time and service availability guarantees at these seven clusters, respectively.

Finally, the trust manager needs to determine whether to accept the job completed by sites 2, 3, 4, 6, 7, 8, and 10 when it receives the completed job at $t = t_8 + T^D = t_5 + 0.18 = t_{26}$.

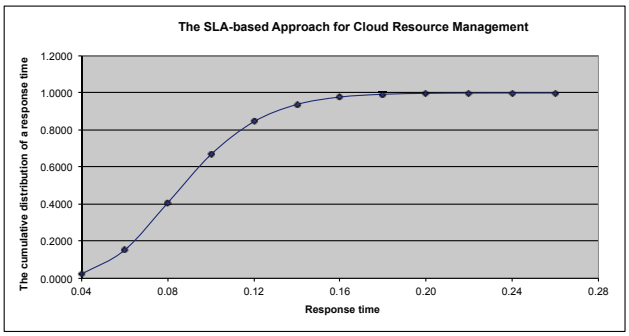

Fig. 3. The Cumulative Distribution of Response Time by the Proposed SLA-based Approach

As is seen, these clusters meet the trust requirement at $t = t_{26}$, and consequently the job is accepted.

We have used the above ten-site example to demonstrate how to apply our approach to solving the trust-based resource provisioning problem.

As we know, most existing clouds do support SLA-based services [8], [16], [22], [23], [30]. However, the solution of cloud resource management has not been addressed very well in existing literature [13], [27], [33]. It is not easy to solve a resource management problem when we consider all these constraints: trustworthiness, an end-to-end response time, and service availability. This section demonstrated how to apply our efficient algorithm to solve the trust-based resource provisioning problem by using the above illustrative example.

V. Conclusions and Future Work

We have studied SLA-based resource management in cloud computing where trustworthiness, percentile response time, and availability are considered as our QoS metrics. We have first proposed an approach for SLA-based resource management and provided an illustrative example to demonstrate how the proposed approach is used for solving the SLA-based resource management problem in high-performance cloud auditing. We have solved the SLA-based resource management problem using an efficient numerical procedure. Our numerical validations have showed that our proposed algorithm has reached a good accuracy.

VI. Acknowledgment

Kaiqi Xiong gratefully acknowledges the partial support from National Science Foundation (NSF) under grants #10656665, #1450854, #1303382, and #1431265, and NSF/BBN under grants #1125515 for project #1895 and #1346688 for project #1936. He is also thankful to Rochester Institute of Technology for its seed fund. Moreover, Xiao Chen gratefully acknowledges the partial support of NSF under grants #1305302 and #1440637.

References

[1] Aib, I., Agoulmine, N., Pujolle, G.: The generalized service level agreement model and its application to the SLA driven management of wireless environments. In: International Arab Conference on Information Technology (2004)

TABLE I

THE INITIAL TRUST INDEX OF THE TEN CLUSTERS

Cluster	1	2	3	4	5	6	7	8	9	10
I^{t_1}	0.8	0.6	0.3	0.8	0.9	0.5	0.9	0.7	0.8	0.9

TABLE II

THE SERVICE RATES OF THE TEN CLUSTERS

Service Rates	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}
Values	88	18	80	35	45	41	15	25	98	35

[2] Allman, M., Eddy, W., Ostermann, S.: Estimating loss rates with TCP. ACM Performance Evaluation Review **31**(3) (2003)

[3] Ballani, H., Costa, P., Karagiannis, T., Rowstron, A.: Towards predictable datacenter networks. In: Proceedings of SIGCOMM (2011)

[4] Bishop, M.: Comoputer Security. Addison Wesley, Boston, MA (2002)

[5] Brown, A., Patterson, D.: Towards availability benchmarks: A case study of software RAID systems. In: Proceedings of 2000 USENIX Annual Technical Conference. USENIX (2000)

[6] Chandra, A., Gong, W., Shenoy, P.: Dynamic resource allocation for shared data centers using online measurements. In: Proceedings of Eleventh International Conference on Quality of Service (IWQoS) (2003)

[7] Chassot, C., Garcia, F., Auriol, G., Lozes, A., Lochin, E., Anelli, P.: Performance analysis for an IP differentiated services network. In: Proceedings of IEEE International Conference on Communication (ICC'02) (2002)

[8] Du, L.: Pricing and resource allocation in a cloud computing market. In: Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012), CCGRID, pp. 817–822 (2012)

[9] Gummadi, K., Saroiu, S., Gribble, S.: King: Estimating latency between arbitrary internet end hosts. In: Proceedings of the ACM SIGCOMM Internet Measurement Workshop (2002)

[10] He, L., Jarvis, S., Spooner, D., Nudd, G.: Optimising static workload allocation in multiclusters. In: Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS) (2004)

[11] Hennessy, J.: The future of systems research. IEEE Computer **32**(8), 27–33 (1999)

[12] Hummer, W., Gaubatz, P., Strembeck, M., Zdun, U., Dustdar, S.: Enforcement of entailment constraints in distributed service-based business processes. Inf. Softw. Technol. **55**(11), 1884–1903 (2013)

[13] Iturrioz, J., Azpeitia, I., Díaz, O.: Generalizing the "like" button: Empowering websites with monitoring capabilities. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing, SAC, pp. 743–750. ACM, New York, NY, USA (2014)

[14] Keller, M., Karl, H.: Response time-optimized distributed cloud resource allocation. In: Proceedings of the 2014 ACM SIGCOMM Workshop on Distributed Cloud Computing, DCC '14, pp. 47–52. ACM, New York, NY, USA (2014)

[15] Kim, Y., Perrig, A., Tsudik, G.: Simple and fault-tolerant key agreement for dynamic collaborative groups. In: Proceedings of the 7th ACM Conference on Computer and Communications Security (ACM CCS 2000), pp. 235 – 244. ACM (2000)

[16] Lin, W.Y., Lin, G.Y., Wei, H.Y.: Dynamic auction mechanism for cloud resource allocation. In: Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGRID, pp. 591–592. IEEE Computer Society, Washington, DC, USA (2010)

[17] Martin, J., Nilsson, A.: On service level agreements for IP networks. In: Proceedings of the IEEE INFOCOM (2002)

[18] Matthys, C., Bari, P., Lieurain, E., Salomon, D., Winkelbauer, L., Jacob, B., Mui, S., Pannu, J., Park, S., Raguet, H., Schneider, J., Vanel, L.: On Demand Operating Environment: Managing the Infrastructure (Virtualization Engine Update). IBM Redbooks (2005)

[19] University of Houston: Network availability. In: http://www.telecomm.uh.edu/stats/Net Status.html

[20] Wikipedia: Service level agreement. In: `http://en.wikipedia.org/wiki/Service-level-agreement`

[21] Menasce, D., Casalicchio, E.: A framework for resource allocation in grid computing. In: Proceedings of the 12th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS) (2004)

[22] Oddi, G., Panfili, M., Pietrabissa, A., Zuccaro, L., Suraci, V.: A resource allocation algorithm of multi-cloud resources based on markov decision process. In: Proceedings of the 2013 IEEE International Conference on Cloud Computing Technology and Science - Volume 01, CLOUDCOM, pp. 130–135. IEEE Computer Society, Washington, DC, USA (2013)

[23] Palanisamy, B., Singh, A., Liu, L., Jain, B.: Purlieus: Locality-aware resource allocation for mapreduce in a cloud. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, SC, pp. 58:1–58:11. ACM, New York, NY, USA (2011)

[24] Penmatsa, S., Chronopoulos, A.: Price-based user-optimal job allocation scheme for grid systems. In: Proceedings of the 20th International Parallel and Distributed Processing Symposium (IPDPS) (2006)

[25] Perrig, A., Canetti, R., Song, D., Tygar, D.: Efficient and secure source authentication for multicast. In: Proceedings of Network and Distributed System Security Symposium (2001)

[26] Rosenberg, J., Remy, D.: Securing Web services with WS-Security: demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption. SAMS, Indianapolis, IN (2004)

[27] Sagbo, K.A.R., Houngue, P.: Quality architecture for resource allocation in cloud computing. In: Proceedings of the First European Conference on Service-Oriented and Cloud Computing, ESOCC, pp. 154–168. Springer-Verlag, Berlin, Heidelberg (2012)

[28] Satoh, I.: Self-adaptive resource allocation in cloud applications. In: Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, UCC, pp. 179–186. IEEE Computer Society, Washington, DC, USA (2013)

[29] Stoner, S.: AF cloud computing architecture. In: `http://www.dodenterprisearchitecture.org/pastmeetings/Documents/stevenstoner.pdf`

[30] Tsai, J.T., Fang, J.C., Chou, J.H.: Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. Comput. Oper. Res. **40**(12), 3045–3055 (2013)

[31] VirtualPC: VirtualPC vulnerabilities. In: `http://www.microsoft.com/echnet/security/bulltin/ms07-049.mspx`

[32] VMware: VMWare vulnerabilities. In: `http://securitytracker.com/alerts/2008/Feb/1019493.html`

[33] Wang, H., Tianfield, H., Mair, Q.: Auction based resource allocation in cloud computing. Multiagent Grid Syst. **10**(1), 51–66 (2014)

[34] Wuhib, F., Stadler, R., Lindgren, H.: Dynamic resource allocation with management objectives: Implementation for an openstack cloud. In: Proceedings of the 8th International Conference on Network and Service Management, CNSM, pp. 309–315. International Federation for Information Processing, Laxenburg, Austria, Austria (2013)

[35] Xiong, K., Perros, H.: SLA-based resource allocation in cluster computing systems. In: IEEE International Parallel and Distributed Processing Symposium (IPDPS) (2008)

[36] Zhang, Y., Niyato, D., Wang, P.: An auction mechanism for resource allocation in mobile cloud computing systems. In: Proceedings of the 8th International Conference on Wireless Algorithms, Systems, and Applications, WASA, pp. 76–87. Springer-Verlag, Berlin, Heidelberg (2013)

TABLE III

THE SERVER UNAVAILABILITY RATES OF THE TEN CLUSTERS

Unavailable Rates	η_1	η_2	η_3	η_4	η_5	η_6	η_7	η_8	η_9	η_{10}
Values	0.08	0.01	0.005	0.04	0.011	0.03	0.02	0.045	0.045	0.01

TABLE IV

THE TRUST INDICES OF THE TEN CLUSTERS AT TIMES $t = t_2, \cdots, t_5, \cdots, t_{20}, t_{21}$

Cluster	1	2	3	4	5	6	7	8	9	10
I^{t_2}	0.8345	0.7442	0.6786	0.9558	0.7945	0.6668	0.8543	0.7983	0.5405	0.7885
...	...	...	...	...	...	...	...	...	...	...
I^{t_6}	0.8863	0.8893	0.8965	0.8701	0.9235	0.8998	0.9183	0.5936	0.6845	0.9565
I^{t_7}	0.8428	0.9335	0.9409	0.9514	0.8563	0.9455	0.9603	0.9465	0.7992	0.9435
I^{t_8}	0.9305	0.9589	0.9656	0.9748	0.8854	0.9758	0.9621	0.9688	0.5858	0.9688
...	...	...	...	...	...	...	...	...	...	...
$I^{t_{25}}$	0.8728	0.9215	0.9338	0.9182	0.8736	0.9326	0.9419	0.9546	0.6223	0.9235
$I^{t_{26}}$	0.8923	0.9585	0.9802	0.9508	0.9206	0.9624	0.9798	0.9896	0.8888	0.9545

TABLE V

THE OPTIMAL NUMBER OF SERVERS

Cluster	2	3	4	6	7	8	10
#Servers necessary to ensure 99% response time	62	5	20	16	84	35	20
#Servers necessary to ensure 99.999% service availability	10	7	22	18	15	23	10
#Servers necessary to ensure these two SLA metrics	62	7	22	18	84	35	20