Melanie Siebenhaar, Ulrich Lampe, Dieter Schuller, Ralf Steinmetz:
Robust Cloud Monitor Placement for Availability Verification.
In: Proceedings of the 4th International Conference on Cloud Computing and Services Science (CLOSER 2014), April 2014.

Robust Cloud Monitor Placement for Availability Verification

Melanie Siebenhaar, Ulrich Lampe, Dieter Schuller, and Ralf Steinmetz
Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Germany
Email: firstname.lastname@KOM.tu-darmstadt.de

Keywords: Cloud Computing, Service Level Agreements, Verification, Monitoring, Placement, Performance, Availability

Abstract: While cloud computing provides a high level of flexibility, it also implies a shift of responsibility to the cloud provider and thus, a loss of control for cloud consumers. Although existing means such as service level agreements or monitoring solutions offered by cloud providers aim to address this issue, there is still a low degree of trust on consumer side that cloud providers properly measure compliance against SLAs. A solution lies in designing reliable means for monitoring cloud-based services from a consumer's perspective. We already proposed such a monitoring approach in our former work. However, our experiments revealed that our approach is sensitive to network impairments. Hence, in the work at hand, we introduce the Robust Cloud Monitor Placement Problem and present a formal optimization model. Based on the model, we propose an initial optimization approach, that allows to obtain an exact solution using off-the-shelf algorithms.

1 INTRODUCTION

Cloud computing provides highly configurable computing resources on-demand and with minimal management effort via the Internet (Mell and Grance, 2011). Hence, it promises a high level of flexibility similar to utilities like electricity or water (Buyya et al., 2009). However, using services from the cloud implies a shift of responsibility to the cloud provider and thus, a loss of control for cloud consumers. A negotiation of so-called service level agreements (SLAs) between a cloud consumer and a cloud provider addresses this issue by providing quality guarantees to the cloud consumer and specifying corresponding penalties for the cloud provider in case of violating a part of the contract. The quality guarantees typically refer to certain quality levels such as thresholds for performance parameters like availability. Although a negotiation of SLAs provides a certain level of control to cloud consumers, this solution does not seem to be sufficient. Still, there is only a low degree of trust on consumer side that providers conduct sufficient measurements of performance against SLAs (CSA and ISACA, 2012). However, cloud providers often additionally provide some means for monitoring to their customers. Nevertheless, solely relying on these solutions cannot be perceived as a reliable evidence base for detecting and documenting SLA violations from a consumer's perspective, particularly in the context that providers often put the burden of SLA violation reporting on their customers (Patel et al., 2009).

A solution lies in the design and provision of reliable means to enable consumers to verify compliance with SLAs from their perspective. Such an approach has already been proposed in our former work in (Siebenhaar et al., 2013). Our approach involves the placement of monitoring units within both, the cloud provider's and consumer's infrastructure in order to verify the availability of cloud applications. However, our experiments revealed that our solution is very sensitive to network impairments. Hence, we examine the robust placement of monitoring units depending on current network reliability in the work at hand. This paper introduces the *Robust Cloud Monitor Placement Problem* (RCMPP) as a new research problem and presents a formal optimization model. Furthermore, we propose transformations that can be applied to this nonlinear, multi-objective optimization problem in order to obtain an exact solution using off-the-shelf optimization algorithms.

The remainder of this paper is structured as follows: Section 2 gives an overview of related approaches to our work. Section 3 briefly summarizes our former approach for availability verification. In Section 4, we describe the RCMPP in detail and introduce a formal optimization model. Based on the formal model, we propose an optimization approach in Section 5. The paper closes with a summary and directions for future work in Section 6.

2 RELATED WORK

Only a few approaches have been proposed so far for monitoring the performance of cloud resources from a consumer's perspective. For example, (Sharma et al., 2013) propose a network monitoring framework that enables tenants to monitor the connectivity between their allocated compute nodes. Other related publications focus on robust placement problems in networks in general. For example, (Natu and Sethi, 2008) try to minimize the number of monitor locations while accounting for a maximum of k node/edge failures, and (Bin et al., 2011) consider relocatable VMs in the presence of up to k host failures. In the field of Operations Research, the fault-tolerant facility location problem, first studied by (Kamal and Vazirani, 2000), where each city must at least be connected to a certain number of distinct facilities, is related to the work at hand. But none of the related approaches addresses robust monitor placement by jointly considering current network reliability, monitor redundancy, and resource as well as location constraints.

3 AVAILABILITY VERIFICATION OF CLOUD APPLICATIONS

In this section, we give an overview of our former work for verifying the availability of cloud applications introduced in (Siebenhaar et al., 2013), which serves as a foundation for the work at hand. Our former work proposed a hybrid monitoring approach for verifying the availability of cloud applications from a consumer's perspective, since availability is one of the very few performance parameters that are part of the SLAs of today's cloud providers (e.g., Elastic Compute Cloud (EC2) by (Amazon, 2008)). However, further performance parameters can be easily incorporated in our approach as well. Basically, our former approach combines consumer- and cloud-side monitoring. In doing so, consumers not only obtain means to assess the status of a cloud application independently from a cloud provider, but also obtain visibility of the end-to-end performance of a cloud-based service. Hence, our approach enables consumers to attribute downtimes to causes either on consumer or provider side. Three different component types are considered in our design (cf. Figure 1): Monitoring units placed on VMs on consumer- and cloud-side observe the availability of a cloud application and the VM, where the application is running on. For our approach we make the assumption that a consumer has access to the VM where a cloud application is hosted. However, a monitoring unit on cloud-side observing a specific application is never placed on the VM where the application is hosted. In this case, a monitoring unit would not be able to report any downtimes if the underlying VM crashes. Nevertheless, a lightweight software component, a VM monitor, is placed on each VM where a cloud application is hosted in order to gain access to predefined processes that are essential for running the cloud application. A periodical pull model is applied by the monitoring units in order to invoke predefined services of the cloud application or request the status of essential processes from VM monitors. The set of monitored services and essential processes of a cloud application can be defined in advance, e.g., as part of the SLAs. Finally, a broker component is introduced which collects and aggregates the monitoring data of all monitoring units. The monitoring units apply an event-based push model to send data to the broker. The broker maintains a list of the downtimes reported by each monitoring unit and periodically computes the overall availability. For more information on the computation of the overall availability, the interested reader is referred to (Siebenhaar et al., 2013). Since failures in the network between a consumer and provider could prevent monitoring units on consumer-side from detecting a downtime of a cloud application, monitoring units should be placed on different network domains. We assume that an enterprise running several data centers at different branches could assign independent consumer-side monitors to different locations.

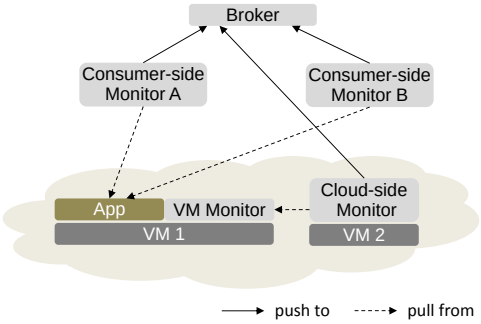

Figure 1: Overview of the monitoring framework

4 ROBUST CLOUD MONITOR PLACEMENT PROBLEM

4.1 Problem Statement

Our work focuses on a scenario, where an enterprise cloud user has several regional branches worldwide, each running a private data center, and makes use of applications running in the data centers of a cloud

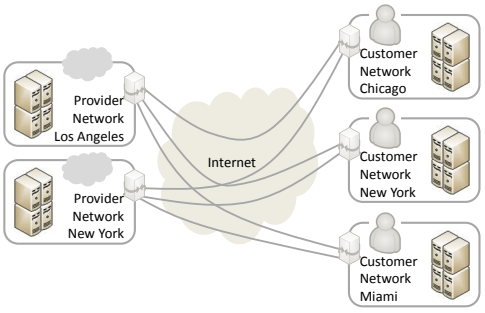

Figure 2: Robust Cloud Monitor Placement scenario[1]

provider (cf. Figure 2). We take the perspective of the enterprise cloud user, who wants to verify the availability of the running cloud applications. For this purpose, the enterprise cloud user applies our hybrid monitoring approach as described in the previous section. In doing so, s/he places monitoring units for each cloud application on consumer- and provider-side. However, since the experiments in (Siebenhaar et al., 2013) revealed that our approach is sensitive to network impairments, the placement of the monitoring units must be conducted in such a way that the robustness of our monitoring framework is maximized. Table 1 gives an overview of the symbols introduced in the following and used in the formal model. The upper part of the table describes the basic entities, whereas the lower part contains the parameters.

The scenario considered in this work consists of a set of data center sites, formally denoted as $S = \{1,...,n\}$. This set S is a union of the set $S'' = \{1,...,d\}$) of data center sites of the enterprise cloud user and the set $S' = \{d+1,...,n\}$) of data center sites of the cloud provider. A set $V_s = \{1,...,i\}$ of VMs is running on each data center site $s \in S$. All VMs V_s are potential candidates for monitor placement. On the data center sites $s' \in S'$ of the cloud provider, each VM $v' \in V_{s'}$ is provided to the enterprise cloud user, e.g., as part of a virtual private network, running a set of cloud applications $C_{s'v'} = \{1,...,j\}$ to be monitored. The VMs $v \in V_s$ representing potential monitor candidates are interconnected with the VMs $v' \in V_{s'}$ of the cloud applications $C_{s'v'}$ via a set of links $L = \{l(sv \rightleftharpoons s'v')\}$. Since the enterprise cloud user is not aware of the underlying network topologies of the Internet service provider and the cloud provider when connecting to a cloud application, we assume that the enterprise cloud user is only able to measure the end-to-end performance between a given pair of VMs represented by a single link $l \in L$ between this pair of VMs in the model. Each monitoring unit for a cloud application $c \in C_{s'v'}$ exhibits a certain resource demand of $rd_{s'v'cr} \in \mathbb{R}^+$ for a specific resource type $r \in R = \{1,...,k\}$ such as, e.g., CPU power or memory. Furthermore, each VM $v \in V_s$ on a site $s \in S$ is able to provide a specific resource supply of $rs_{svr} \in \mathbb{R}^+$. For reasons of fault-tolerance, the enterprise cloud user specifies a redundancy factor $rf_{s'v'c} \in \mathbb{N}_{>1}$ for each cloud application $c \in C_{s'v'}$ indicating that application c has to be monitored by $rf_{s'v'c}$ different monitoring units. We further assume that the enterprise cloud user is able to utilize traditional network measurement tools in order to estimate the reliability of a given VM $v \in V_s$ on a site $s \in S$ or a link $l \in L$ between a given pair of VMs. With respect to these measurements, $p_{sv} \in \mathbb{R}^+$ denotes the observed reliability for a VM $v \in V_s$ on a site $s \in S$ and $p_{l(sv \rightleftharpoons s'v')} \in \mathbb{R}^+$ denotes the observed reliability for a given link $l \in L$.

The challenge for the enterprise cloud user now consists in assigning the monitoring units of all cloud applications to VMs on consumer and provider sites. In doing so, the objective of the enterprise cloud user is to maximize the reliability of the monitoring framework expressed by the probability that at least one of the redundant monitoring units for each cloud application is working. The resource supplies of all VMs must not be exceeded by the assigned moni-

Table 1: Used symbols in the formal model

Symbol	Description
$S = \{1,...,n\}$	set of n data center sites
$S'' = \{1,...,d\})$	consumer sites, $S'' \subset S$
$S' = \{d+1,...,n\})$	provider sites, $S' \subset S$
$V_s = \{1,...,i\}$	VM candidates for monitor placement on site $s \in S$
$C_{s'v'} = \{1,...,j\}$	cloud applications to monitor on VM $v' \in V_{s'}, s' \in S'$
$L = \{l(sv \rightleftharpoons s'v')\}$	links interconnecting VM monitor candidates V_s and VMs of applications $C_{s'v'}$
$R = \{1,...,k\}$	set of k considered VM resource types
$rd_{s'v'cr} \in \mathbb{R}^+$	resource demand for monitoring application $c \in C_{s'v'}$ for resource $r \in R$
$rs_{svr} \in \mathbb{R}^+$	resource supply of VM $v \in V_s$ for resource $r \in R$
$rf_{s'v'c} \in \mathbb{N}_{>1}$	redundancy factor for monitoring application $c \in C_{s'v'}$
$p_{l(sv \rightleftharpoons s'v')} \in \mathbb{R}^+$	observed reliability for each link $l \in L$
$p_{sv} \in \mathbb{R}^+$	observed reliability for each VM $v \in V_s$

[1]based on http://docs.aws.amazon.com/AmazonVPC/latest/NetworkAdminGuide/images/Branch_Offices.png

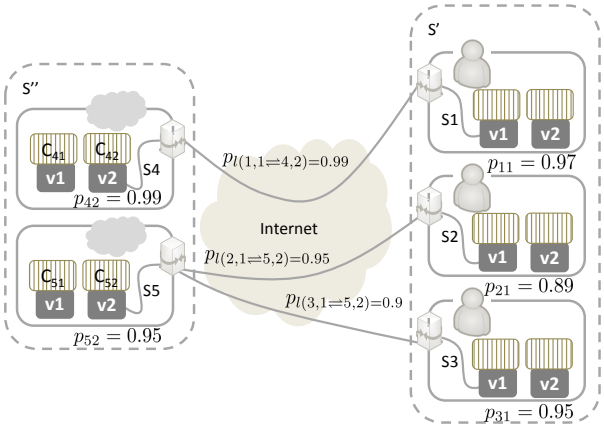

Figure 3: Robust Cloud Monitor Placement example

toring units. Furthermore, all monitoring units for a specific cloud application must be placed on different sites and, following our hybrid monitoring approach, at least one monitor must be placed on consumer and provider side, respectively. In addition, a monitoring unit for a specific cloud application is not allowed to be placed on the VM where the cloud application is running. As already mentioned, this is due to the fact that a monitoring unit is not able to report a downtime when the underlying VM is currently down.

We denote this problem as the *Robust Cloud Monitor Placement Problem* (RCMPP). A simplified example for a RCMPP instance neglecting resource demands and supplies for ease of clarity is depicted in Figure 3. The instance exhibits two provider sites S_4, S_5 and three consumer sites S_1, S_2, S_3 each running two VMs. A set of applications is running on each VM. For example, C_{42} denotes the set of applications running on VM no. 2 on site S_4. The corresponding link between this VM and VM no. 1 on consumer site S_1 exhibits a reliability of $p_{l(1,1 \rightleftharpoons 4,2)} = 0.99$ and VM no. 1 exhibits a reliability of $p_{11} = 0.97$. In our work, we are only interested in monitoring the cloud applications on provider side. However, the applications running on consumer side are also considered in the model in the form of remaining resource supplies of their VMs. If we assume that only one application is part of each set $C_{s'v'}$ on provider side in the example and that the enterprise cloud user specifies a redundancy factor $rf_{s'v'c} = 2$ for each application, 8 monitoring units will be assigned to VMs in total, thereby placing one monitoring unit on consumer side and one on provider side for each application.

4.2 Formal Model

In order to develop strategies to solve the RCMPP, we transform the problem into an optimization model.

Model 1 Robust Cloud Monitor Placement Problem

$$Maximize \quad \{p^{mon}_{s'v'c}(x) | s' \in S', v' \in V_{s'}, c \in C_{s'v'}\} \quad (1)$$

$$p^{mon}_{s'v'c}(x) = 1 - \prod_{s \in S, v \in V_s} (q^{path}_{svs'v'})^{x_{svs'v'c}} \quad (2)$$

$$q^{path}_{svs'v'} = [(1 - p_{sv}) + (1 - p_{l(sv \rightleftharpoons s'v')}) - (1 - p_{sv})\ (1 - p_{l(sv \rightleftharpoons s'v')})] \quad (3)$$

subject to

$$\sum_{s \in S, v \in V_s} x_{svs'v'c} = rf_{s'v'c} \quad (4)$$
$$\forall s' \in S', v' \in V_{s'}, c \in C_{s'v'}, rf_{s'v'c} \geq 2$$

$$\sum_{s' \in S', v' \in V_{s'}, c \in C_{s'v'}} rd_{s'v'cr}\ x_{svs'v'c} \leq rs_{svr} \quad (5)$$
$$\forall s \in S, v \in V_s, r \in R$$

$$\sum_{v \in V_s} x_{svs'v'c} \leq 1 \quad (6)$$
$$\forall s \in S, s' \in S', v' \in V_{s'}, c \in C_{s'v'}$$

$$\sum_{s \in S, v \in V_s} x_{svs'v'c} \geq 1 \quad (7)$$
$$\forall s' \in S', v' \in V_{s'}, c \in C_{s'v'}, s = \{d+1, ..., n\}$$

$$\sum_{s \in S, v \in V_s} x_{svs'v'c} \geq 1 \quad (8)$$
$$\forall s' \in S', v' \in V_{s'}, c \in C_{s'v'}, s = \{1, ..., d\}$$

$$x_{svs'v'c} = 0 \quad (9)$$
$$\forall c \in C_{s'v'},\ s = s' \text{ and } v = v'$$

$$x_{svs'v'c} \in \{0, 1\} \quad (10)$$
$$\forall s \in S, v \in V_s, s' \in S', v' \in V_{s'}, c \in C_{s'v'}$$

The resulting optimization model is depicted in Model 1 and will be described in the following.

As already stated, the objective of the RCMPP is to assign the monitoring units of all cloud applications to VMs on consumer and provider side so that the resulting placement maximizes the probability that at least one of the monitoring units for each cloud application is working. The objective can be described by a set of multiple potentially conflicting objective functions that we want to maximize simultaneously (cf. Equation 1). Each objective function expresses the probability $p^{mon}_{s'v'c}(x)$ to reliably monitor a specific application, i.e., at least one of the redundant monitoring units is working properly. $p^{mon}_{s'v'c}(x)$ equals 1 minus the probability that all monitors for a specific cloud

application $c \in C_{s'v'}$ fail (cf. Equation 2). The decision variable $x_{svs'v'c}$ is defined in Equation 10 as binary and indicates whether VM $v \in V_s$ running on site $s \in S$ is selected to monitor cloud application $c \in C_{s'v'}$ running on VM $v' \in V_{s'}$ on site $s' \in S'$. x denotes the vector of all decision variables $x_{svs'v'c}$ (cf. Equation 1). Equation 3 determines the probability $q^{path}_{svs'v'}$ for a monitoring unit of cloud application $c \in C_{s'v'}$ placed on VM $v \in V_s$ on site $s \in S$ to fail, thereby considering the reliability of the VM $v \in V_s$ and the reliability of the path interconnecting VM $v \in V_s$ and the VM of the cloud application. Equation 4 ensures that a cloud application $c \in C_{s'v'}$ is monitored by $rf_{s'v'c}$ redundant monitoring units. For each application, at least two monitoring units must be assigned to VMs, one on consumer side and one on provider side (cf. Equations 7 and 8). Equation 5 defines that each VM $v \in V_s$ offers sufficient resource supplies to serve the assigned monitoring units and Equation 6 makes sure that all monitoring units for a specific cloud application $c \in C_{s'v'}$ are placed on different sites. Finally, Equation 9 ensures that a monitoring unit for a specific cloud application $c \in C_{s'v'}$ is not assigned to the VM $v' \in V_{s'}$ on site $s' \in S'$ where the cloud application is also running on.

As can be seen from Model 1, the RCMPP constitutes a multi-objective optimization problem. Hence, no unique solution of the problem exists. Furthermore, the RCMPP is nonlinear due to Equation 2. However, the problem can be linearized and transformed into a single-objective optimization problem as described in the next section.

Model 2 Optimization of Complementary Objectives

$$Minimize \quad \{q^{mon}_{s'v'c}(x) | s' \in S', v' \in V_{s'}, c \in C_{s'v'}\} \quad (11)$$

$$q^{mon}_{s'v'c}(x) = \prod_{s \in S, v \in V_s} (q^{path}_{svs'v'})^{x_{svs'v'c}} \quad (12)$$

$$q^{path}_{svs'v'} = [(1 - p_{sv}) + (1 - p_{l(sv \rightleftharpoons s'v')}) - (1 - p_{sv})\ (1 - p_{l(sv \rightleftharpoons s'v')})] \quad (13)$$

Model 3 Linearization of RCMPP

$$log(q^{mon}_{s'v'c}(x)) = log(\prod_{s \in S, v \in V_s} (q^{path}_{svs'v'})^{x_{svs'v'c}}) \quad (14)$$

leads to

$$Minimize \quad \{q^{log}_{s'v'c}(x) | s' \in S', v' \in V_{s'}, c \in C_{s'v'}\} \quad (15)$$

$$q^{log}_{s'v'c}(x) = \sum_{s \in S, v \in V_s} x_{svs'v'c}\ log(q^{path}_{svs'v'}) \quad (16)$$

$$q^{path}_{svs'v'} = [(1 - p_{sv}) + (1 - p_{l(sv \rightleftharpoons s'v')}) - (1 - p_{sv})\ (1 - p_{l(sv \rightleftharpoons s'v')})] \quad (17)$$

$$q^{path}_{svs'v'} > 0\ \forall s \in S, v \in V_s, s' \in S', v' \in V_{s'} \quad (18)$$

5 Optimization Approach

We propose an initial solution approach based on linearization and transformation of the RCMPP into a single-objective optimization problem. In doing so, off-the-shelf optimization algorithms, such as branch-and-bound (Hillier and Liebermann, 2005), can be applied in order to obtain an exact solution.

For linearization of the RCMPP, we follow a two-step approach: In the first step, we build a new set of objective functions representing the complementary objectives of the former Model 1 and turn the maximization problem into a minimization problem (cf. Model 2). $q^{mon}_{s'v'c}(x)$ in Model 2 determines the probability that all monitors for a specific cloud application $c \in C_{s'v'}$ fail and thus, represents the complementary probability to $p^{mon}_{s'v'c}(x)$ in the former Model 1. In the second step, we can now linearize the problem by taking the logarithm of both sides (cf. Equation 14), an approach that is also followed by (Andreas and Smith, 2008). Model 3 shows the result of the linearization. Since no system is without failure, we assume $q^{path}_{svs'v'} > 0$. Please note, that the Equations 4 to 10 of Model 1 also belong to Model 2 and Model 3, but have been neglected due to lack of space. Finally, we have to transform Model 3 into a single-objective optimization problem. For this purpose, we apply a so-called *minimax strategy* (Jensen and Bard, 2003), i.e., our transformation is based on a worst-case analysis, where we aim to minimize the worst possible outcome. Hence, the former objective of minimizing the probability $q^{log}_{s'v'c}(x)$ that all monitors fail for all cloud applications simultaneously can be transformed into minimizing the maximum probability of all $q^{log}_{s'v'c}(x)$ (cf. Equation 19 in Model 4). In doing so, the maximum (i.e., worst) probability of all $q^{log}_{s'v'c}(x)$ can be represented by a new decision variable $z \in \mathbb{R}$. Furthermore, we have to add $|C_{s'v'}|$ new constraints $\forall s' \in S', v' \in V_{s'}$ (cf. Equation 20 in Model 4) to the set of former constraints already introduced in Model 1. As can be seen from Model 4, the resulting problem constitutes a mixed-integer linear programming problem that can be solved using branch-and-bound (Hillier and Liebermann, 2005).

Model 4 Transformation to Single-Objective RCMPP

$$Minimize \quad z \tag{19}$$

subject to

$$q^{log}_{s'v'c}(x) \leq z \tag{20}$$

$$\forall s' \in S', v' \in V_{s'}, c \in C_{s'v'}, z \in \mathbb{R}$$

6 SUMMARY AND OUTLOOK

Although cloud-based service delivery is a very flexible and convenient way for consumers to obtain computing resources, it is attended with a shift of responsibility to the cloud provider and thus, with a loss of control for consumers. Negotiating SLAs with providers and using their provisioned monitoring solutions to verify SLA compliance later on is not considered as sufficient by consumers. Therefore, we have designed a hybrid monitoring approach for availability verification of cloud applications from a consumer's perspective in our former work in (Siebenhaar et al., 2013). However, since our experiments revealed that our approach is sensitive to network impairments, we examined the robust placement of monitoring units in the work at hand. In this paper, we introduced the *Robust Cloud Monitor Placement Problem* (RCMPP) and a corresponding, formal optimization model. Furthermore, we proposed an initial solution approach based on transformations turning the nonlinear, multi-objective RCMPP into a mixed-integer linear programming problem. An exact solution can then be obtained using the branch-and-bound optimization algorithm.

In future work, we will implement and evaluate our proposed optimization approach. Furthermore, we plan to extend the proposed model to consider other objectives such as the total monitoring costs.

ACKNOWLEDGEMENTS

This work was supported in part by the German Federal Ministry of Education and Research (BMBF) under grant no. "01|C12S01V" in the context of the Software-Cluster project SINNODIUM (www.software-cluster.org), E-Finance Lab Frankfurt am Main e.V. (http://www.efinancelab.com), and the German Research Foundation (DFG) in the Collaborative Research Center (SFB) 1053 – MAKI. The authors assume responsibility for the content.

REFERENCES

Amazon (2008). Amazon ec2 service level agreement. http://aws.amazon.com/ec2-sla/, [last access: 3 December 2012].

Andreas, A. K. and Smith, J. C. (2008). Mathematical Programming Algorithms for Two-Path Routing Problems with Reliability Considerations. *INFORMS Journal on Computing*, 20(4):553–564.

Bin, E., Biran, O., Boni, O., Hadad, E., Kolodner, E., Moatti, Y., and Lorenz, D. (2011). Guaranteeing High Availability Goals for Virtual Machine Placement. In *31st International Conference on Distributed Computing Systems (ICDCS)*, pages 700–709.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. *Future Generation Computer Systems*, 25(6):599–616.

CSA and ISACA (2012). Cloud Computing Market Maturity. Study Results. Cloud Security Alliance and ISACA. http://www.isaca.org/Knowledge-Center/Research/Documents/2012-Cloud-Computing-Market-Maturity-Study-Results.pdf, [last access: 28 November 2012].

Hillier, F. S. and Liebermann, G. J. (2005). *Inroduction to Operations Research*. McGraw-Hill, 8th edition.

Jensen, P. A. and Bard, J. F. (2003). Appendix A: Equivalent Linear Programs. In *Supplements to Operations Research Models and Methods*. John Wiley and Sons. http://www.me.utexas.edu/ jensen/ORMM/supplements/units/lp_models/equivalent.pdf [last access: 12 January 2014].

Kamal, J. and Vazirani, V. V. (2000). An Approximation Algorithm for the Fault Tolerant Metric Facility Location Problem. In Jansen, K. and Khuller, S., editors, *Approximation Algorithms for Combinatorial Optimization*, volume 1913 of *Lecture Notes in Computer Science*, pages 177–182. Springer.

Mell, P. and Grance, T. (2011). The NIST Definition of Cloud Computing. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, [Last access: 28 November 2012].

Natu, M. and Sethi, A. S. (2008). Probe Station Placement for Robust Monitoring of Networks. *Journal of Network and Systems Management*, 16(4):351–374.

Patel, P., Ranabahu, A., and Sheth, A. (2009). Service Level Agreement in Cloud Computing. Technical report, Knoesis Center, Wright State University, USA.

Sharma, P., Chatterjee, S., and Sharma, D. (2013). CloudView: Enabling Tenants to Monitor and Control their Cloud Instantiations. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 443–449.

Siebenhaar, M., Wenge, O., Hans, R., Tercan, H., and Steinmetz, R. (2013). Verifying the Availability of Cloud Applications. In Jarke, M. and Helfert, M., editors, *Proceedings of the 3rd International Conference on Cloud Computing and Services Science (CLOSER 2013)*, pages 489–494. SciTe Press.