The Interplay Between Timeliness and Scalability in Cloud Monitoring Systems

Guilherme da Cunha Rodrigues[1,2], Rodrigo N. Calheiros[2], Marcio Barbosa de Carvalho[1],
Carlos Raniery Paula dos Santos[3], Lisandro Zambenedetti Granville[1], Liane Tarouco[1], Rajkumar Buyya[2]

[1]Computer Networks Group, Institute of Informatics
Federal University of Rio Grande do Sul, Porto Alegre, Brazil
Email: {gcrodrigues,mbcarvalho,granville,liane}@inf.ufrgs.br

[2]**Clou**d Computing and **D**istributed **S**ystems (CLOUDS) Laboratory
Department of Computing and Information Systems
The University of Melbourne, Australia
Email: rncalheiros@ieee.org, rbuyya@unimelb.edu.au

[3]Department of Applied Computing
Federal University of Santa Maria, Santa Maria, Brazil
Email: carlos.santos@ufsm.br

Abstract— **Cloud computing is a groundbreaking solution to acquire computational resources on demand. To deliver high quality cloud services and provide features such as reduced costs and availability to customers, a cloud, like any other computational system, needs to be properly managed in accordance with its characteristics (*e.g.,* scalability, elasticity, timeliness). In this scenario, cloud monitoring is a key to achieve it. To properly work, cloud monitoring systems need to meet several requirements such as scalability, accuracy, and timeliness. This paper aims to unveil the trade-off between timeliness and scalability. Evaluations demonstrate the mutual influence between scalability and timeliness based on monitoring parameters (*e.g.,* monitoring topologies, frequency sampling). Results show that non-deep monitoring topologies and decreasing the frequency sampling assist to reduce the mutual influence between timeliness and scalability.**

I. Introduction

Cloud computing is a groundbreaking solution to acquire computational resources on demand [1] [2]. Nowadays, cloud computing has been standing out by some advantages such as reduced costs, accessibility, and flexibility. In accordance to NIST (National Institute of Standards and Technology) [3], there are five essential characteristics of a cloud, namely, on demand self-service, broad network access, resources pooling, rapid elasticity, and measured service. Measured service demands that a cloud automatically controls the resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service provided (*e.g.,* processing, bandwidth, active user accounts). Thus, resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of services.

To deliver high quality cloud services and provide features such as reduced costs and availability to customers, a cloud, like any other computational system, needs to be properly managed in accordance with its characteristics (*e.g.,* scalability, elasticity) [4]. In this context, cloud monitoring becomes a challenging and important issue because it provides to cloud operators (*e.g.,* service providers, infrastructure providers) means to suitably manage (*e.g.,* analyse, control) a cloud computing environment [5].

Cloud monitoring serves as support to management activities. It presents information from multiple resources (*e.g.,* network, processing) and services (*e.g.,* analyses, notifications), enabling cloud operators to perform control activities, which allows for the cloud to offer predictable performance to customers [6] [7]. In order to properly meet such performance goals, cloud monitoring systems need to meet several requirements such as scalability, comprehensiveness, adaptability, accuracy, elasticity, and timeliness [8] [9].

Nevertheless, when a cloud monitoring system tries to accomplish a specific requirement, other requirement is usually negatively or positively affected [8]. For this reason, the development of cloud monitoring systems is focused on specific aspects of cloud operation, providing only partial solutions for cloud monitoring [10]. In addition, the lack of knowledge about the exact influence among cloud monitoring requirements restricts the capacity of integrating cloud monitoring systems. Therefore, the balance among cloud monitoring requirements becomes a challenging and important task to cloud monitoring systems. To solve this problem, we need to firstly understand the relationship among different cloud monitoring requirements, *i.e,* how much a specific cloud monitoring requirement have influence over other. Previous research investigated the mutual influence between scalability, adaptability and accuracy [11]. The goal of this paper is investigate the interplay between timeliness and scalability.

Timeliness is an ability in which a cloud monitoring system is able to supply information in the time that users (*e.g.,* service providers, customers) need to access it [8] [12]. Cloud monitor-

978-1-4673-7194-0/15/$31.00 ©2015 IEEE

ing systems need to be timely in order to collect, synchronize, organize, and present the amount of monitoring data generated from distributed probes. Timeliness is an important task to cloud monitoring systems because its activities (*e.g.,* collection, synchronization) need to be opportunely accomplished to avoid breaches in SLAs and consequent financial penalties caused by late detections of problems in the infrastructure. Additionally, timeliness is affected by other cloud monitoring requirements such as accuracy, elasticity and scalability [11].

Scalability is the capacity in which a cloud monitoring system is able to increase the number of probes in order to ensure that all resources that compose a cloud are properly monitored [13] [14]. To enable scalability, cloud monitoring systems must be developed to cope with huge amount of monitoring data that will be collected and transferred from distributed probes [10]. Scalability is a difficult task to cloud monitoring systems due the necessity to manage a large number of probes and data. In addition, huge amount of monitoring data impairs cloud monitoring requirements such as adaptability, accuracy and timeliness [8].

The interplay between timeliness and scalability is an important issue because, to supply information in time to users (timeliness), a cloud monitoring system needs to be capable to grow in amount of probes to monitor all resources in a cloud (scalability). On the other side, the amount of probes impairs the capacity of the system to be timely because it negatively impacts activities such as collection and synchronization. Furthermore, these requirements are directly related to other requirements such as accuracy, adaptability, and elasticity [8] [11]. The acquaintance of the real trade-off between timeliness and scalability will assist the design and development of comprehensive cloud monitoring systems that can improve clouds in issues such as increase profits to infrastructure and service providers and techniques for self configuration. Additionally, it will assist in the evaluation of other requirements in the future, allowing the enhancement of the integration among cloud monitoring systems.

The remainder of this paper is organized as follows. Section II presents the related work. Section III introduces and discusses relevant issues about the mutual influence between timeliness and scalability. Section IV presents a quantitative evaluation between the two requirements. Section V presents conclusions and future work.

II. Related Work

The monitoring activity is essential for Cloud Service Providers (CSPs) in order to guarantee the proper functioning of the cloud infrastructure. Several research groups have investigated the requirements that should be fulfilled by comprehensive cloud monitoring systems. In this section, we highlight the most recent and relevant works in this area.

Aceto *et al.* [8] presented an exhaustive study on cloud monitoring. The authors defined a set of properties that cloud monitoring systems should support, difficulties in supporting those properties, and the related solutions currently available in the literature. Examples of the monitoring properties discussed by Aceto *et al.* include: timeliness, scalability, elasticity, and adaptability. Finally, the authors presented the current platforms and services available for monitoring cloud environments and discussed which properties are tackled by each platform.

Montes *et al.* [10] proposed a cloud monitoring solution based on levels (*e.g.,* SaaS, PaaS, IaaS, physical) called GMonE. By using GMonE, both Cloud Service Providers and customers are able to visualize monitoring data. In addition, Montes *et al.* presented an evaluation about the mutual relation between scalability and elasticity. The results demonstrated that monitoring solutions can be improved when monitoring requirements are considered together.

Clayman *et al.* [9] presented Lattice, a cloud monitoring framework developed to monitor both resources and services in virtualized environments. The design and development of the framework, allowed Clayman *et al.* to identified the main requirements for cloud monitoring systems, *i.e.,* scalability, elasticity, migration, adaptability, autonomy, and federation. The authors have also discussed federation problems and its impact over the monitoring activity.

Despite the importance of the presented works, they lack a depth discussion on the mutual influence among the monitoring requirements. Such analysis is essential once each individual requirement is highly affected by the others [10]. We have ourselves initiated an effort to evaluate multiple requirements in conjunction [11]. Initially, we investigated the mutual influence among scalability, adaptability, and accuracy. The results allowed us to identify proper monitoring methods to reduce the impact of scalability over adaptability. In this paper, we expand the evaluation to determine the interplay between timeliness and scalability.

III. Scalability and Timeliness

Previous research provided a large discussion about cloud monitoring requirements [8] [9] [10] [11]. However, there are several issues that require further investigation. In this paper, we address the mutual influence between timeliness and scalability, which we define below.

- *Timeliness*: Timeliness is the competence that a monitoring system has to detect events on time to assist users to obtain information at the moment in which they need to use it. Timeliness is important to cloud monitoring systems because cloud systems are based on Service Level Agreements (SLAs) that regulate the deal among infrastructure providers, service providers, and customers. In this scenario, if monitoring data is not timely, an action correcting violation in the SLA cannot be accomplished in time, resulting in penalties (costs) to a service provider, for example.
- *Scalability*: Scalability is the competence to increase the amount of probes in a monitoring system to cope with resource increase in the system. Scalability is important to cloud monitoring systems because the cloud business model provides resources on demand. Traditional monitoring systems are static and then cannot easily handle cloud system characteristics that are directly related to scalability such as dynamicity and autonomicity.

Scalable systems, like clouds, have capacity to quickly increase the amount of resources on demand. However, it implies in challenges to cloud computing environments. For example, it can be challenging to assure that a cloud monitoring system will detect and respond within a previously agreed time interval (*e.g.,* 10ms, 50ms, 200ms) a virtual machine failure in a cloud computing environment with 10,000 virtual machines (Timeliness). In this context, other aspects that can be suitably explored include: how to define SLAs in accordance with the size of a cloud, or how to provide support to a service provider in order to define SLAs based on its response time capacity.

Cloud monitoring systems currently have two methods to handle the mutual influence between timeliness and scalability. In the first method, the cloud monitoring system works to accomplish a specific requirement in detriment to another. In other words, it aims to accomplish scalability without concerning its impact over timeliness or vice-versa. This method is widely used [10] [15], although it is not efficient because it restrains the cloud monitoring system capacity to attend a specific requirement, creating incomplete monitoring solutions. In the second method, the cloud monitoring system aims to provide both requirements in a balanced way. This method is more complex and non trivial and, to be achieved, the mutual influence between these two requirements has to be unveiled.

The acknowledgement of mutual influence between timeliness and scalability assists to improve monitoring in clouds. Additionally, the mutual influence between both is important because, at the same time that a cloud monitoring system grows to monitor all resources in the system, the amount of probes and monitoring data also increase, impairing activities such as data collection and synchronization.

Data collection is impaired in this situation because there are more monitoring data to be gathered. Synchronization is impaired because there are more monitoring data to be analysed together. Thus, to fulfil data collection and synchronization, the cloud monitoring system spends more time. Therefore, the amount of monitoring data makes it difficult to handle it timely, causing delay between event occurrence and notification. In this context, if scalability is fulfilled, the cloud monitoring system has more monitoring information to be managed and it impairs timeliness because it induces communication delay [8], as demonstrated in Section IV-B.

Usually, filtering and aggregation are implemented in cloud monitoring solutions to reduce the amount of monitoring data. They reduce the communication delay and, as a result, timeliness and scalability are improved. To apply filtering, there are different methods (*e.g.,* by resource type, statistics), techniques (*e.g.,* compress, reduce), and monitoring architectures [12] [16] [17]. However, regardless the method, technique, or architecture, filtering and aggregation were shown to be harmful for others requirements such as accuracy [11].

Moreover, there are characteristics, such as frequency sampling and resources placement, that need to be considered, because in a scalable cloud monitoring system they have influence over timeliness.

Frequency sampling depends on the resource type (*e.g.,* CPU, memory) that is monitored. For instance, to monitor CPU utilization, a monitoring system has to obtain samples in shorter intervals because this is a resource whose utilization constantly changes in tiny intervals of time. Sampling in higher frequency increases the amount of monitoring data in a network, causing communication delay as verified in Section IV-C.

Resource placement is a characteristic that contributes to increase the communication delay in a scalable cloud monitoring system. It happens because the distance between resources and managers contributes to the time spent in communication, as show in Section IV-A. In addition, resource placement is crucial to activities such as synchronization because the managers placement contributes to communication delay.

Besides the above considerations, timeliness and scalability have issues that must be analysed to accomplish other cloud monitoring requirements such as accuracy, adaptability and elasticity. For example:

- Frequency sampling impairs both *timeliness* and *accuracy*, as shown by Park *et at.* [18];
- The method used to increase the amount of probes is important to reduce the influence of *scalability* over *adaptability* [11];
- The dynamic changes (*elasticity*) of monitored resources must be timely (*timeliness*) reported [8].

From the above, we conclude that mutual influence evaluation between both timeliness and scalability can assist in future works about the influence among other cloud monitoring requirements.

IV. Quantitative Evaluation of Timeliness and Scalability

The experimental evaluation presented in this section aims to unmask the trade-off between timeliness and scalability. It provides results such as how monitoring topologies are affected by both requirements and how a specific monitoring topology is impaired by monitoring parameters (*e.g.,* amount of monitoring data, frequency sampling).

The experimental environment is build using Mininet system [19] to simulate monitoring topologies. The evaluation is performed on an Intel 2.20Ghz Pentium 4 core 2 duo T6600 CPUs, 4GB of RAM, running Ubuntu Server 12.04 LTS.

The monitoring parameters which we investigate are amount of monitoring information per sampling (*i.e.,*120 and 150 bytes), frequency sampling (*i.e.,*1 and 10 seconds) and amount of samples per experiment (*i.e.,*100 samples). Network links have 1Gbps and two monitoring topologies are evaluated. Response time is the output metric, and it is measured between probes in the edge hosts and core manager.

Monitoring topologies are based on typical architectures for cloud environments consisting in two and three levels trees of routers or switches [20] [21]. To each switch is added one aggregator, thereby, the amount of aggregators depends on the topology. Topologies are depicts in Figure 1 and Figure 2, namely, Topology 1 and Topology 2. Monitoring topologies are evaluated to timeliness based on addition of hosts as well as aggregators in order to analyse influence of scalability over timeliness. Topology 1 is expanded to 64, 256, 576 and 1296 hosts, and respectively to 10, 18, 26 and 38 aggregators. Topology 2 is expanded to 64, 216, 512 and 1331 hosts, and respectively to 21, 43, 73 and 133 aggregators.

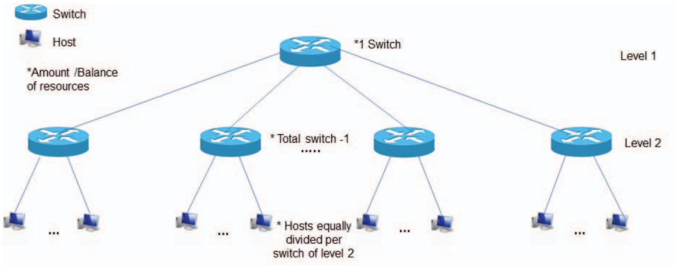

Switch
Host
*Amount /Balance of resources
*1 Switch
Level 1
* Total switch -1
.....
Level 2
...
* Hosts equally divided per switch of level 2

Figure 1. Topology 1, evaluated topology for until level 2 and expanded to 64, 256, 576 and 1296 hosts.

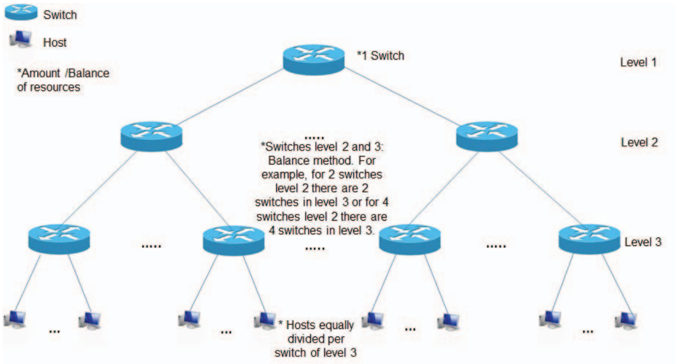

Switch
Host
*Amount /Balance of resources
*1 Switch
Level 1
Level 2
Level 3
*Switches level 2 and 3: Balance method. For example, for 2 switches level 2 there are 2 switches in level 3 or for 4 switches level 2 there are 4 switches in level 3.
.....
...
* Hosts equally divided per switch of level 3

Figure 2. Topology 2, evaluated topology for until level 3 and expanded to 64, 216, 512 and 1331 hosts.

In the next subsections, the evaluations are presented in accordance with monitoring parameters, namely, monitoring topology, frequency sampling and amount of monitoring data.

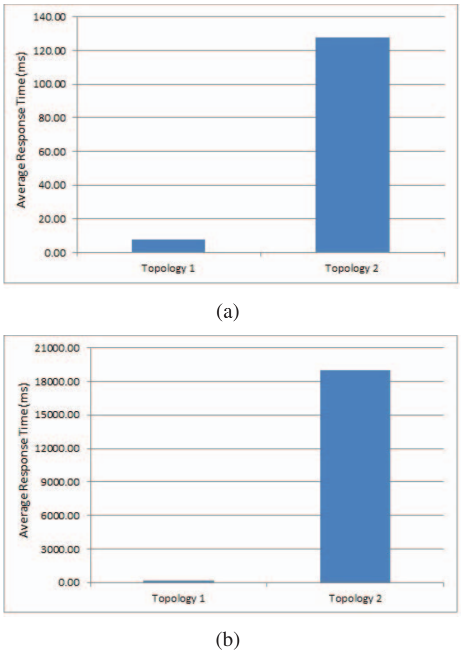
A. *Monitoring topology*

Monitoring topologies are structures used as a support to the collection and transferring of monitoring data. In our first set of experiments, we investigate the influence of monitoring topologies over the performance of cloud monitoring system.

Evaluations are made to 64 hosts and 1300 hosts. Figure 3 and Figure 4 present evaluations to response time per topology to 120 bytes and 150 bytes of monitoring data. Evaluations to monitoring topologies show that non-deep topologies (*i.e.,* Topology 1) present shorter response time than deep topologies (*i.e.,* Topology 2). For instance, Figure 3 show that response time is lower for Topology 1, regardless the amount of hosts. Figure 3 and Figure 4 show that response time is lower for Topology 1, regardless the amount of monitoring data per sampling.

Non-deep topologies reduce replication of monitoring data in a network because core managers are closer to edge agents as well as close to intermediate managers. In addition, non-deep topologies reduce hops between edge agents and core managers.

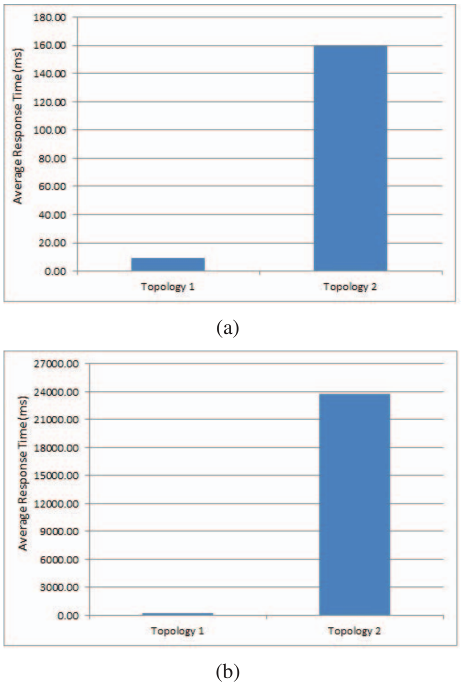
Monitoring topologies present a predictable increase for response time based on amount of monitoring data per sampling and amount of hosts. For example, Figure 3 and Figure 4 show that response time increase in 1,678.55% between Topology 1 and Topology 2 with 120 bytes and 64 hosts, virtually the same increased in response time (*i.e.,* 1,675.21%) to Topology 1 and Topology 2 with 150 bytes and 64 hosts.



(a)

(b)

Figure 3. Average response time per topology with 120 bytes of monitoring data for 64 and 1300 hosts. (a) 64 Hosts. (b) 1300 Hosts.



(a)

(b)

Figure 4. Average response time per topology with 150 bytes of monitoring data for 64 and 1300 hosts. (a) 64 Hosts. (b) 1300 Hosts.

B. *Amount of monitoring data*

In this subsection, evaluations aim to unveil how the amount of monitoring data impairs response time in an isolate way. Evaluations aims to verify the influence of monitoring data increase over timeliness without taking into account other parameters and, as a result, evaluations shows how the monitoring data scalability impairs timeliness. Frequency sampling is not utilized because it impairs the evaluation to

Table I. AVERAGE RESPONSE TIME FOR 120 BYTES OF MONITORING DATA WITHOUT FREQUENCY SAMPLING.

Topology	Hosts	Aggregators	RT Average (ms)
Topology 1	64	10	7.60
Topology 1	256	18	30.11
Topology 1	576	26	67.62
Topology 1	1296	38	151.99
Topology 2	64	21	127.57
Topology 2	216	43	936.61
Topology 2	512	73	3900.08
Topology 2	1331	133	19029.25

Table II. AVERAGE RESPONSE TIME FOR 150 BYTES OF MONITORING DATA WITHOUT FREQUENCY SAMPLING.

Topology	Hosts	Aggregators	RT Average (ms)
Topology 1	64	10	9.52
Topology 1	256	18	37.64
Topology 1	576	26	84.51
Topology 1	1296	38	189.99
Topology 2	64	21	159.48
Topology 2	216	43	1170.84
Topology 2	512	73	4875.12
Topology 2	1331	133	23786.56

amount of monitoring data as we explain in the Section IV-C.

Table I and Table II present the results in terms of average response time for amount of monitoring data for 120 bytes and 150 bytes without frequency sampling. Results show that response time is affected by the amount of monitoring data in accordance with the growing of a cloud monitoring system. For example, the average response time is 151.99 ms for 120 bytes of 189.99 ms for 150 bytes of Topology 1 with 1296 hosts. In other words, response time increased 25.01%, virtually the same difference of amount of data between 120 and 150 bytes (25%). The behaviour is practically the same to all topologies and scenarios evaluated, being the worst case to Topology 1 with 64 hosts where response time increased in 25.26%.

C. Frequency sampling

Evaluations for frequency sampling aim to unmask issues such as: how the interval between data collection and response time impairs timeliness in accordance with the scalability of cloud monitoring systems. Frequency sampling is an important parameter because, depending on the frequency of data collection, the response time increases and, as a consequence, timeliness is impaired.

Table III, Table IV, Table V, and Table VI present results in terms of average response time for variation of frequency sampling based on 120 bytes and 150 bytes. Evaluation results show that frequency sampling impairs response time when the frequency sampling is shorter than the response time. It happens when a new process of data collection starts, and the former data collection was not finished. For instance, Table III (*i.e.,* Topology 2, 512 and 1331 Hosts) shows that response time is increased in 390% and 1903% respectively, comparing response time with the same configuration in Table I. On the other hand, when the frequency sampling is bigger than the response time, the monitoring data in a network does not increase at the same time and, as a consequence, the response time practically remain constant. For example, Table III (*i.e.,* Topology 2, 64 Hosts) has virtually the same response time (*i.e.,* 127.93ms) that the equivalent configuration in Table I (*i.e.,* 127.57ms).

Table III. AVERAGE RESPONSE TIME FOR 120 BYTES OF MONITORING DATA WITH FREQUENCY SAMPLING OF 1 SECOND.

Topology	Hosts	Aggregators	RT Average (ms)
Topology 1	64	10	7.72
Topology 1	256	18	30.07
Topology 1	576	26	67.77
Topology 1	1296	38	151.54
Topology 2	64	21	127.93
Topology 2	216	43	936.33
Topology 2	512	73	15210.46
Topology 2	1331	133	362127.72

Table IV. AVERAGE RESPONSE TIME FOR 120 BYTES OF MONITORING DATA WITH FREQUENCY SAMPLING OF 10 SECONDS.

Topology	Hosts	Aggregators	RT Average (ms)
Topology 1	64	10	7.67
Topology 1	256	18	30.19
Topology 1	576	26	67.54
Topology 1	1296	38	151.81
Topology 2	64	21	128.02
Topology 2	216	43	936.32
Topology 2	512	73	3900.22
Topology 2	1331	133	36155.62

Table V. AVERAGE RESPONSE TIME FOR 150 BYTES OF MONITORING DATA WITH FREQUENCY SAMPLING OF 1 SECOND.

Topology	Hosts	Aggregators	RT Average (ms)
Topology 1	64	10	9.44
Topology 1	256	18	37.71
Topology 1	576	26	84.38
Topology 1	1296	38	189.67
Topology 2	64	21	159.77
Topology 2	216	43	1369.91
Topology 2	512	73	23766.35
Topology 2	1331	133	565787.41

Table VI. AVERAGE RESPONSE TIME FOR 150 BYTES OF MONITORING DATA WITH FREQUENCY SAMPLING OF 10 SECONDS.

Topology	Hosts	Aggregators	RT Average (ms)
Topology 1	64	10	9.56
Topology 1	256	18	37.51
Topology 1	576	26	84.59
Topology 1	1296	38	189.91
Topology 2	64	21	159.11
Topology 2	216	43	1170.45
Topology 2	512	73	4875.58
Topology 2	1331	133	56374.18

Table IV presents results when frequency sampling is set to 10 seconds. In Table IV (*i.e.,* Topology 2, 512 Hosts), we noticed that response time (*i.e.,* 3900.22 ms) is virtually the same in Table I (*i.e.,* 3900.08 ms) because the response time is shorter than the frequency sampling. In Table IV (*i.e.,*Topology 2, 1331 Hosts), we observe that response time (*i.e.,*36155.62 ms) increases because the frequency sampling is shorter than response time (*i.e.,*19029.25 ms) in Table I. In addition, comparing Table IV (*i.e.,* Topology 2, 1331 hosts) with the equivalent configuration in Table III, we realized that response time (*i.e.,* 362127.72 ms) is bigger in Table III because the frequency sampling is 10 times short (*i.e.,* 1 second in Table III and 10 seconds in Table IV).

These results demonstrated that shorter intervals for frequency sampling impairs timeliness. The impact of frequency sampling over response time grows with the increase of number of probes. It happens because, when the monitoring system grows, the amount of resources to be monitored increases. Hence, we observed more clearly the frequency sampling influence in large topologies.

D. Discussion

Results for monitoring parameters presented in this paper demonstrated that cloud providers could reduce the mutual influence between timeliness and scalability via different methods. In the first method, cloud providers could reduce as much as possible the frequency sampling in accordance with SLAs as presented in Section IV-C. In the second method, when negotiating SLAs, cloud providers could take into account the depth of the infrastructure to set the response time. For example, to increase profit, a cloud provider could own different infrastructures with different topologies to support different and more restricts SLAs for top customers, thereby, providing to them better response time and accomplishing timeliness.

V. Conclusions and Future Works

This paper presented an investigation about the mutual influence between timeliness and scalability. The acquaintance of the interplay between both requirements support important issues such as assisting the design and development of comprehensive and integrated cloud monitoring systems, supporting future evaluations among others requirements (*e.g.,* adaptability, elasticity), increasing profits to infrastructure and service providers based on predictions to resources usage in monitoring, and assisting cloud providers to satisfy SLAs based on timeliness.

Evaluations demonstrated the mutual influence between scalability and timeliness in regard to different monitoring parameters (*e.g.,* monitoring topologies, frequency sampling). However, cloud or infrastructure providers could reduce the mutual influence between timeliness and scalability through methods proposed in Section IV-D.

Additionally, the results showed that mutual influence between scalability and timeliness can be quantified based on monitoring parameters such as monitoring topologies, amount of monitoring data, and frequency sampling. Therefore, the mutual influence between timeliness and scalability is liable to be mathematically represented.

As future works, we plan to develop and evaluate a mathematical method to predict the mutual influence between timeliness and scalability. Moreover, we will investigate the trade-off between timeliness and adaptability.

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, Jun. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.future.2008.12.001

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, apr 2010.

[3] P. M. Mell and T. Grance, "Sp 800-145. the nist definition of cloud computing," Gaithersburg, MD, United States, Tech. Rep., 2011.

[4] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[5] G. Rodrigues, V. Guimaraes, G. Santos, L. Tarouco, and L. Granville, "Network and services monitoring: A survey in cloud computing environments," in *Proceedings of the 11th International Conference on Networks*, ser. ICN '12, 2012, pp. 7–13.

[6] Amazon, "Amazon CloudWatch," 2014, available at: http://aws.amazon.com/en/cloudwatch/. access in: Jan 2014.

[7] S. De Chaves, R. Uriarte, and C. Westphall, "Toward an architecture for monitoring private clouds," *Communications Magazine, IEEE*, vol. 49, no. 12, pp. 130 –137, december 2011.

[8] G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Survey cloud monitoring: A survey," *Comput. Netw.*, vol. 57, no. 9, pp. 2093–2115, Jun. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2013.04.001

[9] S. Clayman, A. Galis, C. Chapman, G. Toffetti, L. Rodero-Merino, L. M. Vaquero, K. Nagin, and B. Rochwerger, "Monitoring service clouds in the future internet." in *Future Internet Assembly*, G. Tselentis, A. Galis, A. Gavras, S. Krco, V. Lotz, E. P. B. Simperl, B. Stiller, and T. Zahariadis, Eds. IOS Press, 2010, pp. 115–126. [Online]. Available: http://dblp.uni-trier.de/db/conf/fia/fia2010.html#ClaymanGCTRVNR10

[10] J. Montes, A. Sánchez, B. Memishi, M. S. Pérez, and G. Antoniu, "Gmone: A complete approach to cloud monitoring," *Future Gener. Comput. Syst.*, vol. 29, no. 8, pp. 2026–2040, Oct. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.future.2013.02.011

[11] G. Rodrigues, G. Santos, V. Guimaraes, L. Granville, and L. Tarouco, "An architecture to evaluate scalability, adaptability and accuracy in cloud monitoring systems," in *Proceedings of the 28th International Conference on Information Networking*, ser. ICOIN '14. IEEE, 2014, pp. 46–51.

[12] C. Wang, K. Schwan, V. Talwar, G. Eisenhauer, L. Hu, and M. Wolf, "A flexible architecture integrating monitoring and analytics for managing large-scale data centers," in *Proceedings of the 8th ACM international conference on Autonomic computing*, ser. ICAC '11. New York, NY, USA: ACM, 2011, pp. 141–150. [Online]. Available: http://doi.acm.org/10.1145/1998582.1998605

[13] E. Feller, L. Rilling, and C. Morin, "Snooze: A scalable and autonomic virtual machine management framework for private clouds," in *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, ser. CCGRID '12, 2012, pp. 482–489.

[14] R. Mian, P. Martin, and J. L. Vazquez-Poletti, "Provisioning data analytic workloads in a cloud," *Future Gener. Comput. Syst.*, vol. 29, no. 6, pp. 1452–1458, Aug. 2013.

[15] S. Meng and L. Liu, "Enhanced monitoring-as-a-service for effective cloud management," *IEEE Trans. Comput.*, vol. 62, no. 9, pp. 1705–1720, 2013. [Online]. Available: http://dx.doi.org/10.1109/TC.2012.165

[16] P. Hasselmeyer and N. d'Heureuse, "Towards holistic multi-tenant monitoring for virtual data centers," in *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, 2010, pp. 350–356.

[17] M. Kutare, G. Eisenhauer, C. Wang, K. Schwan, V. Talwar, and M. Wolf, "Monalytics: Online monitoring and analytics for managing large scale data centers," in *Proceedings of the 7th International Conference on Autonomic Computing*, ser. ICAC '10. New York, NY, USA: ACM, 2010, pp. 141–150. [Online]. Available: http://doi.acm.org/10.1145/1809049.1809073

[18] J. Park, H. Yu, K. Chung, and E. Lee, "Markov chain based monitoring service for fault tolerance in mobile cloud computing," in *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on*, March 2011, pp. 520–525.

[19] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets-IX. New York, NY, USA: ACM, 2010, pp. 19:1–19:6. [Online]. Available: http://doi.acm.org/10.1145/1868447.1868466

[20] M. Andreolini, M. Colajanni, and S. Tosi, "A software architecture for the analysis of large sets of data streams in cloud infrastructures," in *Computer and Information Technology (CIT), 2011 IEEE 11th International Conference on*, Aug 2011, pp. 389–394.

[21] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, Aug. 2008. [Online]. Available: http://doi.acm.org/10.1145/1402946.1402967