International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 5 Issue 3, March - 2021, Pages: 156-165

Samsara architecture: Exploring situation awareness in cloud computing management

Shadiyev Usmon Ramazanovich[1], Abdirofiyev Norbek Abdirofi o'g'li[1], Ruslan Malikov[2*], Nigora Abdiyeva[2], Jurabek Abdiyev[3]

[1]Samarkand State University, Samarkand 140104, Uzbekistan
[2]Samarkand State Institute of Foreign Languages. Samarkand, 140104, street Bustonsaroy 93
[3]Physical-technical Institute of NPO "Physics – Sun" of Uzbekistan Academy of Sciences Uzbekistan, Tashkent, Chingiz Aitmatov street 2B.
Author: ruslan.malikov.1987@gmail.com (R. Malikov)

Abstract: *Issues related to energy consumption and its efficiency in large-scale computing environments have emerged as critical points in the development of modern computer systems. This article presents the Samsara architecture, which aims to manage energy-efficient computational clouds. The architecture was developed exploring situation awareness strategies, operating autonomously and with minimal human intervention, essential aspects in massive data processing centers. Samsara has been implemented as one of the modules of the OpenStack cloud platform and considers the maximum allocation capacity of each physical machine, to consolidate the workloads. In the evaluations carried out with synthetic loads, reductions of up to 12.3% in the energy consumption in the managed computational infrastructure were achieved, demonstrating the potential of Samsara for the operation of computational clouds.*

Keywords: Cloud Computing, Energy Efficiency, Situation Awareness.

1. INTRODUCTION.

Cloud Computing has gained a significant space among technologies that support the current scenario of distributed and parallel computing. However, energy consumption of underutilized computing resources, especially in large-scale computing environments, can represent a substantial waste of electrical energy. Also, there is a growing concern to promote a reduction in carbon dioxide emissions, which searches for energy efficiency essential to ensure that the growth of Cloud Computing occurs in a sustainable manner [11]. However, Cloud Computing is exposed to heterogeneous workloads that reflect the diversity of applications supported by it, and that increases the complexity of its structures, searching for more efficient use of its resources and energy consumed in a research challenge, mainly given the scenario exposed above. Considering this scenario, this article presents the design of an architecture, called *Samsara*, which aims *to manage computational clouds with a focus on optimizing the use of resources for energy efficiency*. This architecture was developed using strategies based on Situation Awareness, operating autonomously, and with minimal human intervention – important aspects considering the typical size of computing cloud data centers. The optimization sought explores the maximum allocation capacity of each physical machine, performing the consolidation of workloads. The *Samsara* architecture was evaluated experimentally and the results obtained showed that it achieved a reduction in energy consumption of around 12.31% for the evaluated workloads. This article is organized into seven sections. In Sections 2 and 3 the aspects of energy efficiency in a computational environment and those of Situation Awareness are discussed. Section 4 presents the *Samsara* architecture and its features. Section 5 discusses the results obtained from the Samsara assessment. Section 6 deals with related work. Finally, final considerations are presented in Section 7.

2. ENERGY EFFICIENCY IN COMPUTATIONAL ENVIRONMENTS.

Attention to energy efficiency issues has increased with the growth of the computing industry. As a consequence, there has also been a high demand for energy with this growth. To serve as a starting point, energy consumption in computing from 2005 to 2010 grew by 56%, representing 1.1% to 1.5% of the total energy consumed globally and 2% of CO2 emissions [11]. Recently a study developed in [9] projected that greenhouse gas emissions from information and communication industry (ICT), which includes telephones, computers, and data centers, may correspond to up to 4% of global greenhouse gas emissions by 2020 with an increased forecast for subsequent years. The same study projects that between 2020–2025 this figure may reach 8%. In [2] it points out that this figure could exceed 14% in 2040. Recent hardware development initiatives, which have been focusing their efforts on improving technologies that deal better with energy consumption of components and devices, have managed to increase energy efficiency to a certain degree. The main approaches to increase energy efficiency of computing devices attempt to make energy consumption follow the applications load. An example of this is the use of the technique of voltage and frequency control (Dynamic Voltage and Frequency Scaling – DVFS), which allows reducing the consumption of processors to the cube [5]. Furthermore, industry initiatives aiming at developing standardized techniques and methods for reducing energy consumption, especially consumption associated with large computational structures, have gained impulse and increasingly encourage the idea of

Green Computing or *Sustainable Computing*. Initiatives like *The Green Grid*1 have focused their attention on the issue of energy efficiency and are working to improve the efficiency of information technology resources and data centers around the world. In large-scale environments, issues related to energy consumption are more strongly linked to how available resources are used. In expectation that these environments will have to deal with worst-case situations when providing computing services, there is a tendency to oversize their resources. As a consequence of this strategy, in times of lower demand, resources available in these environments may not reach an ideal efficiency range, resulting in a waste of energy consumed as a side effect [5]. In cloud environments, the use of virtualization allows us to use workload consolidation as an approach capable of promoting greater energy efficiency when migrating running virtual machines to a smaller set of servers, exploring the strategy of using to the maximum the allocation capacity of each physical machine, thus amortizing idle energy costs more efficiently [16].

3. SITUATION AWARENESS.

The development of adaptive applications and their increasing use impose that applications gain characteristics that allow them to deal with the complexity resulting from this process, making them dynamic to adapt their behavior according to the changes that occur around them. This ability to understand the surrounding environment, its changes, and the ability to adapt accordingly, are related to researches related to Situation Awareness [17]. Information from sensors, in general, is raw data with minimal processing. One way to deal with the limitation of working with raw data is to derive that information from a low-level context to a higher-level layer. The notion of situation can be used as a high-level concept for representing a state [13]. A *situation* consists of *the interpretation of a set of instantiated contextual elements, coming from sensors, relating each one to provide some valid information in a specific time interval* [10]. It is a subjective concept that depends on a set of elements to gain meaning. For example, you can depend on the sensors available in the system to determine which contexts can be used in a specification [1]. A situation is distinguished from other information in that it includes structural aspects of temporality – reflecting the instant of an event, duration, frequency and sequence – and of other natures that can be a simple abstract state of a certain entity (e.g. server disconnected). It also depends on the environment where the system works, which determines the domain of knowledge that can be applied and the needs required by the applications, determining which states are interesting. Therefore, the information provided by the same sensor can be interpreted for different situations according to the needs of the applications.

Moreover, a situation can also be composed of other situations abstracted from low-level data and reused in different environments and applications, so that this relationship obtained between situations allows to increase the abstraction of these states to decrease their complexity. The great advantage of using contextual information to promote situation awareness is the ability to provide human beings with an understandable representation of the data captured by sensors for the applications, whereas this uses abstracts from them the complexity of it's reading, simplifying the inference activities [15].

4. SAMSARA: PROPOSED ARCHITECTURE AND FEATURES.

In the proposed modeling for use in *Samsara*, the cloud environment is represented by three entities: *Cell*, *Physical Node*, and *Instance*. The *Cell* represents the cloud environment managed by the *Samsara* architecture. Within the *Cell*, there is a set of *Physical Nodes*, which represent servers or hosts responsible for hosting and running the *Instances*. Each *Instance*, in turn, represents a virtual machine running on a *Physical Node*. Fig. 1 shows the *Samsara* architecture, organized in three subsystems: (i) *Contexts Management Subsystem*, responsible for collection and storage of contextual information; (ii) *Situation Identification Subsystem*, responsible for processing and analysis of contextual information, making adjustments and conversions of the information obtained and stored (processing) and, from them, carry out the identification of situations and decision making from them (analyze); and finally, the (iii) *Adaptation Subsystem*, responsible for executing actions that make changes in the managed environment. Each subsystem has services that run on two instances of the architecture: (a) *Nodal Managers*, installed in Compute Nodes1, are responsible for interacting directly with the physical node, collecting and storing their information through a set of sensors, and by carrying out local analysis of contextual information; and (b) the *Cellular Manager*, installed on Controller Node1, is responsible for managing the cloud environment (*Cell*), using situational context information from Nodal Managers, to analyze and infer situations in which the cell is and trigger actions to modify these situations when necessary, according to the defined rules.2 The *Samsara* architecture implements functionalities of three natures: 3 *Collection of Contextual Information* – In physical node scope, information about the capacity of resources are collected, as well as their identification within the cloud environment and the contextual information regarding their use. This information is collected by the *Collection Service*, which periodically consults the sensors3 to produce contextual information and stores it in the *Nodal Contexts*. This information is used by the *Nodal Interpretation Service* and sent to the Cellular Manager (Fig. 2). 4 *Interpretation of Contexts and Identification of Situations* – The interpretation of contexts and the identification of situations in the physical node are in charge of the Nodal Contexts Interpretation Service, a reasoning component of the Nodal Manager. For this identification, the Nodal Contexts of Repositories is consulted (see Fig. 3). The contextual information recovered is analyzed through the application of rules that will identify the situation of the physical node.

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 5 Issue 3, March - 2021, Pages: 156-165

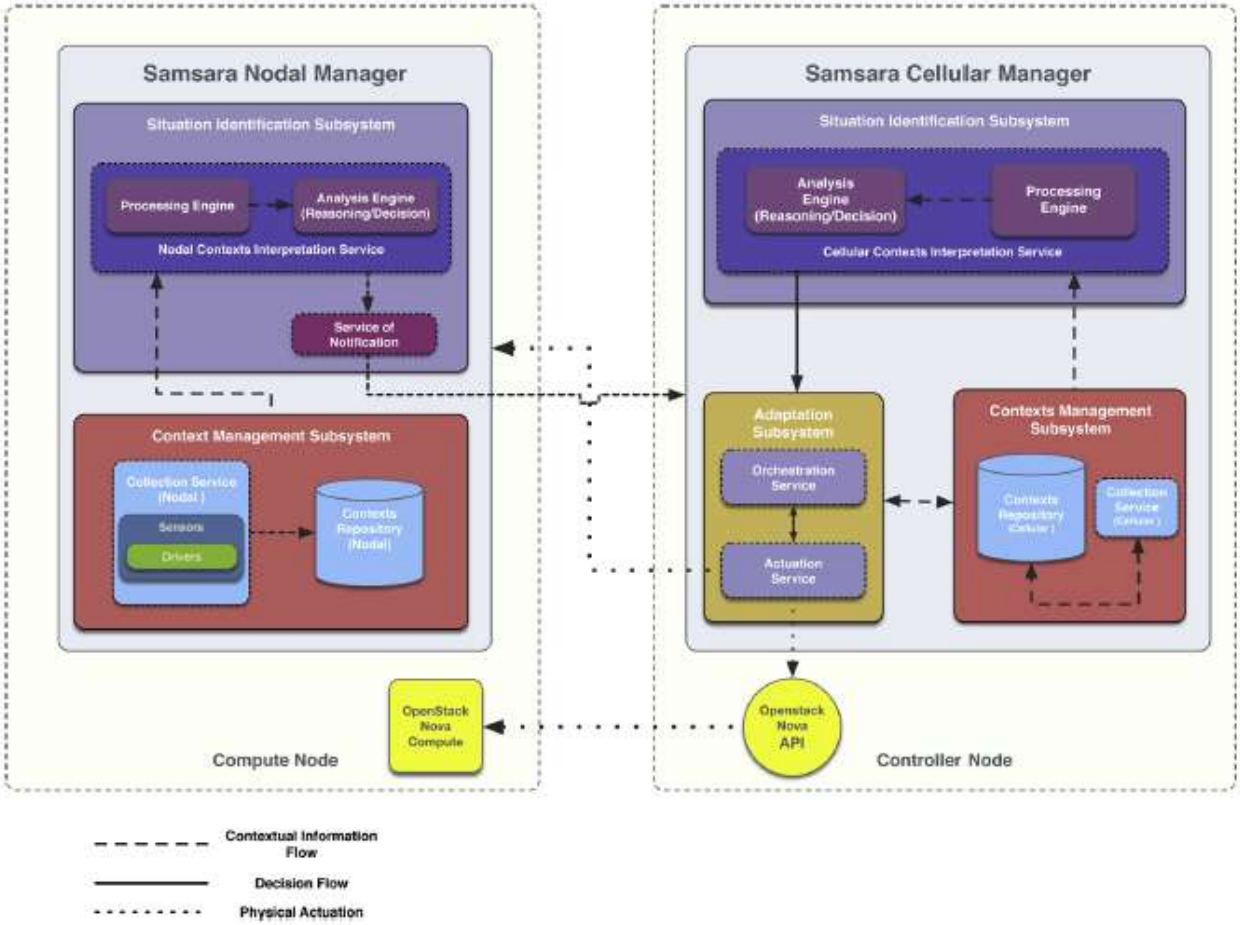

Fig. 1. Samsara architecture organization.

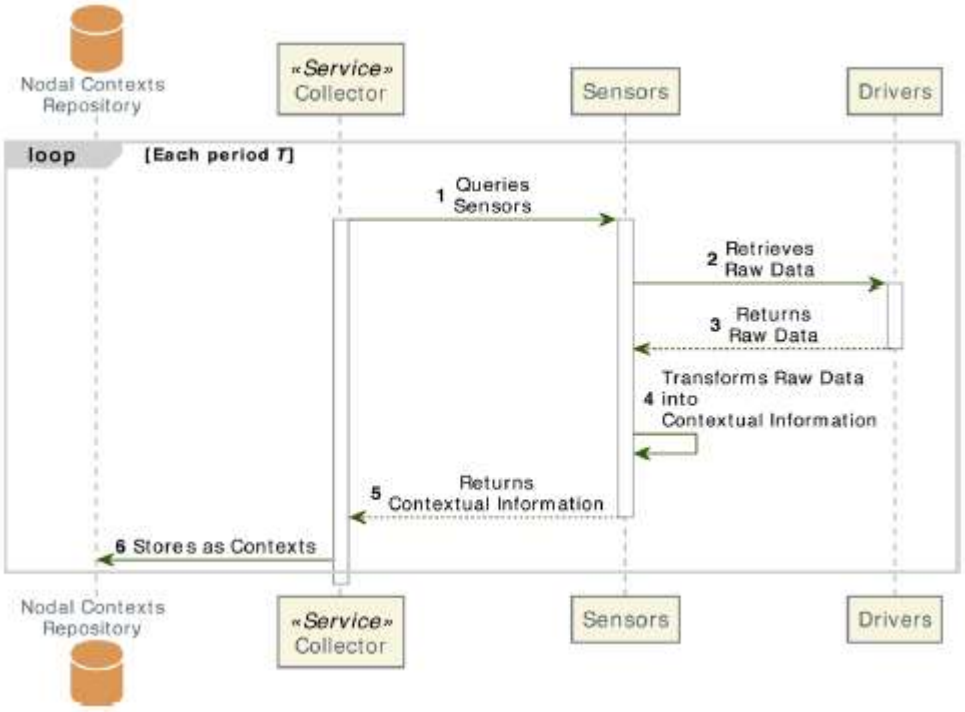

Fig. 2. Collection of contextual information: sequence diagram.

The *situation* at the cellular level is constructed from the notification of the situation of the nodes to the *Situation Identification Subsystem* of the *Cellular Manager*. Once a *situation* that requires changes in the cell environment is identified, an adaptation event is triggered corresponding to the identified *situation*, which will then be handled by the *Adaptation Subsystem* (Fig. 4).

5 *Environment Adaptation* – The process of cell adaptation aims to make changes in the organization of the cloud environment so that it reaches an objective defined by its administrators or reacts to events triggered by the detection of a certain *situation*. During its execution, the *Orchestration Service* analyzes the state of the physical nodes that make up the cell environment, performs the

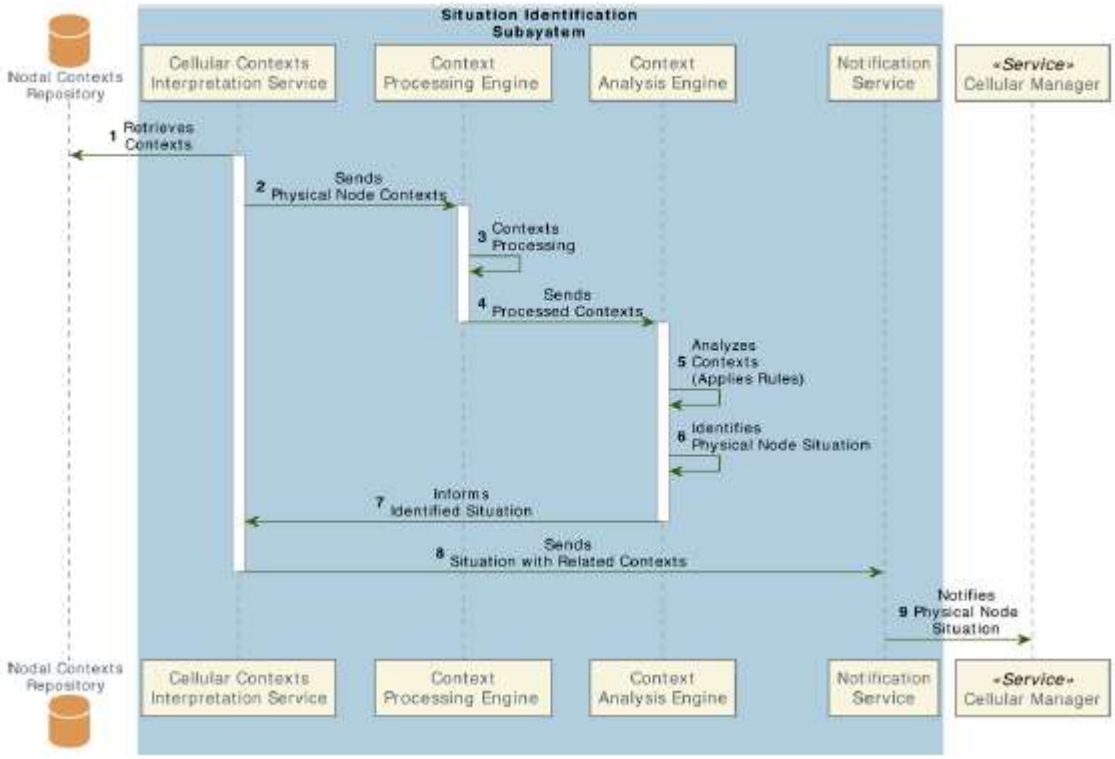

Fig. 3. Interpretation of contexts in nodal manager: sequence diagram.

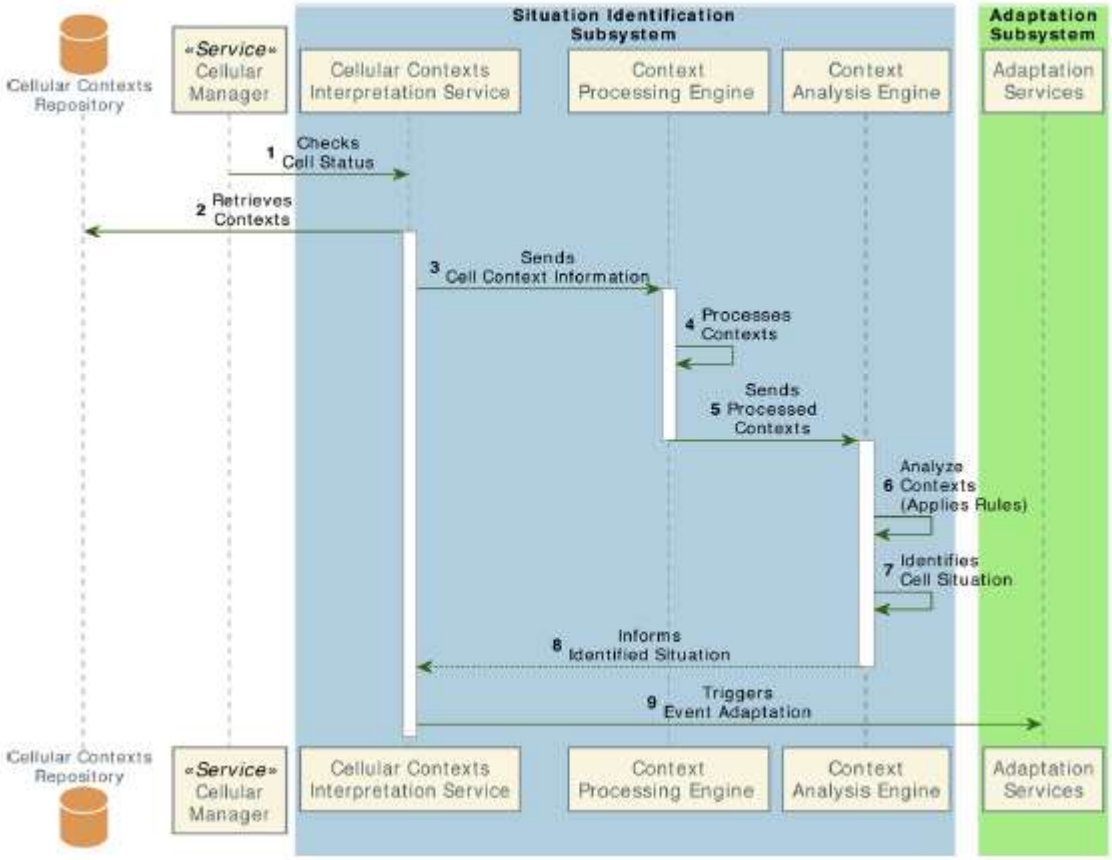

Fig. 4. Interpretation of contexts in the cellular manager: sequence diagram.

application of Adaptation Policies, and generates an *Adaptation Plan*. Then, it activates the *Actuaction Service*, whose set of actuators will execute actions on the *Cell* by the generated *Adaptation Plan*. The architecture is currently prepared to perform two types of adaptation: (i) Adaptation to *Optimize Energy Consumption* (Fig. 5), through the process of consolidating loads and deactivating idle physical nodes; and (ii) Adaptation to *Optimize Load Balancing* (Fig. 6), between physical nodes.

5. EXPERIMENTAL EVALUATION AND RESULTS.

During the evaluation of the *Samsara* architecture, it was assumed that it would be able to identify periods of underutilization or overload of the physical nodes and make adaptations to the *Cell* according to its general state. The objective here is that the Cell's energy consumption corresponds to the workload to which it is subjected. For the execution of the evaluation set, 5 Dell PowerEdge T430 servers were used, with the following description each: Intel (R) Xeon

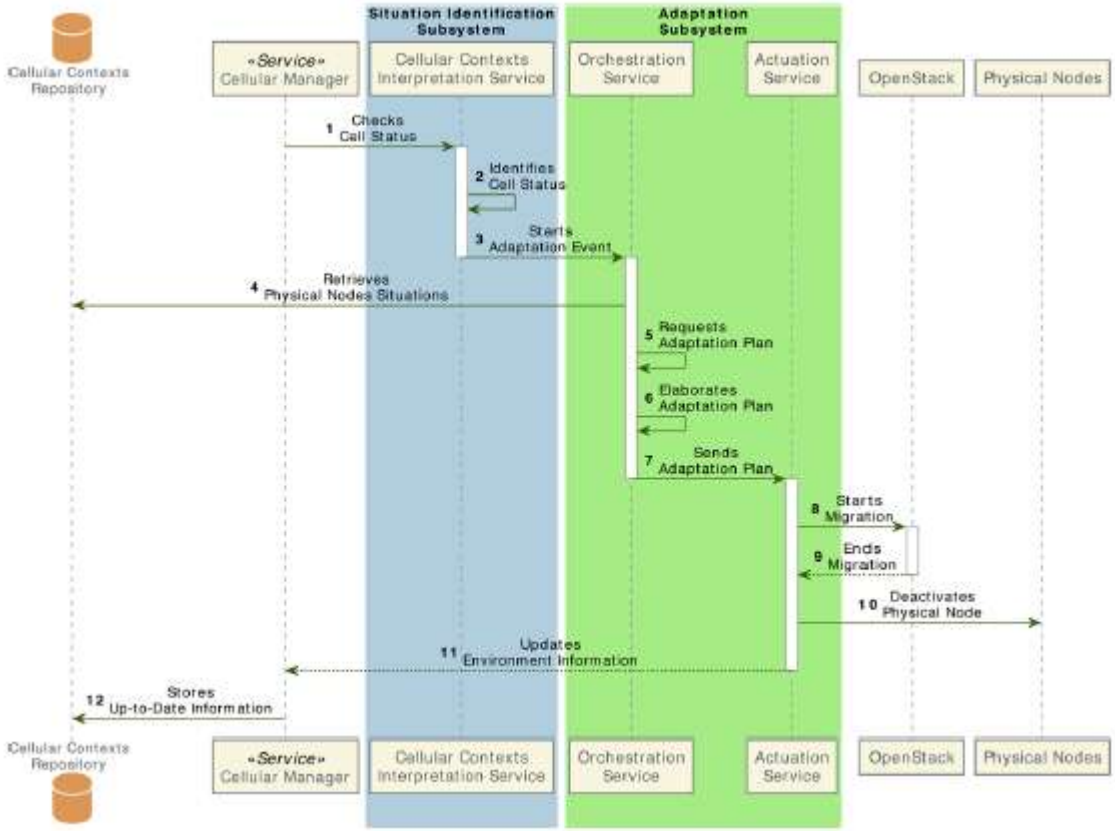

Fig. 5. Process of adaptation to optimize energy consumption.

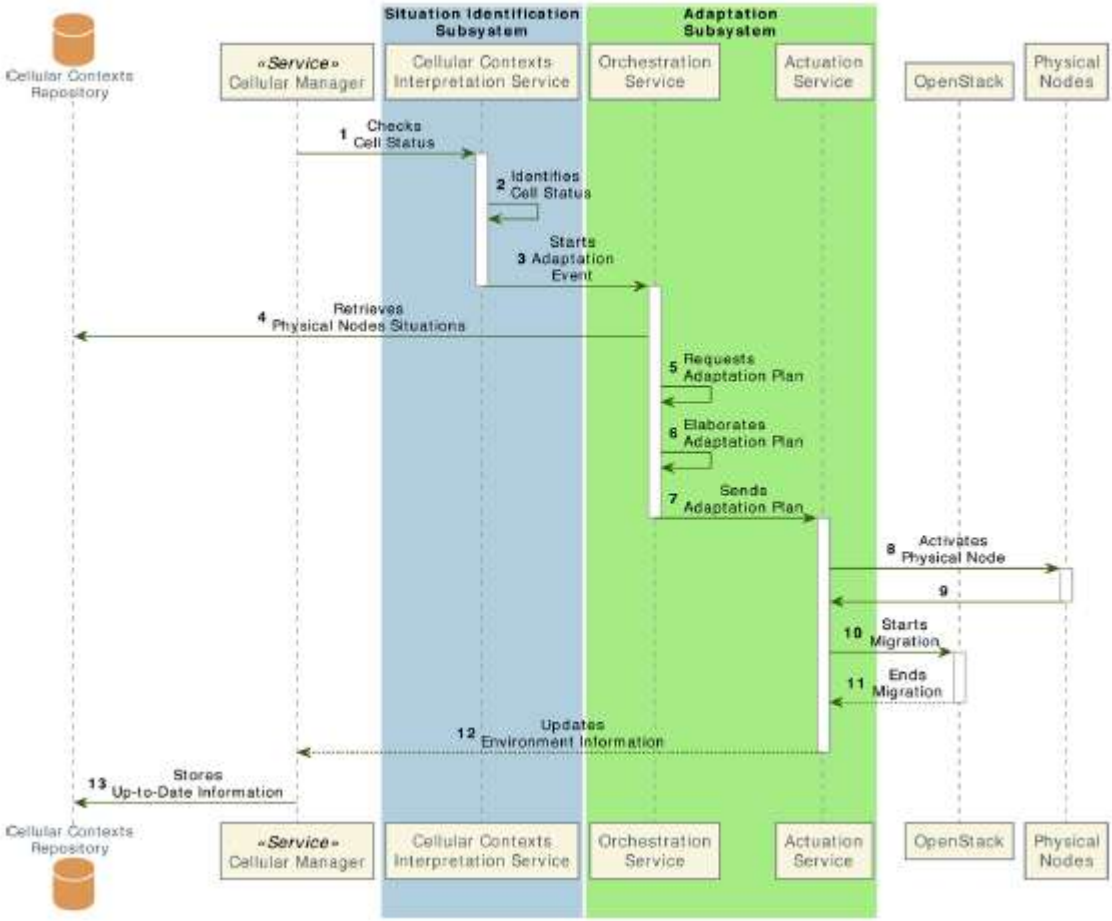

Fig. 6. Process of adaptation to optimizing load balancing.

(R) CPU E5-2420 @ 1.90 GHz (6 cores, 12 threads), 8 GB DDR3 Synchronous DIMM 1600 MHz, running Ubuntu 14.04 LTS – Trusty x86_64, stable version given the characteristic of having long-term support. The cloud environment was implemented on *OpenStack* version 2015.2 (Liberty) and its servers were organized as follows: (i) a server chosen as Controller Node, responsible for hosting and executing Samsara's Cell Controller; (ii) and the other four servers such as Compute Nodes responsible for executing the Samsara Nodal Controller and hosting virtual machines. In the evaluation of the *Samsara* architecture, two scenarios were defined: (i) the Cell's adaptation mechanisms are not activated, seeking to build a basis for evaluating the mechanisms of the developed architecture; and (ii) Samsara's adaptation mechanisms are activated, setting values for detecting physical node (PN) situations to work in the range between 30–70% CPU utilization (normal situation), identifying values below that as underutilization and above as overhead, considering in all of them the existence of at least one active virtual machine (VM), according to Table 1.4 In both scenarios, 30 virtual machines are created and distributed among the 4 physical nodes according to the *OpenStack* engines. The cell settings have been adjusted to identify energy inefficiency when at least one machine has been underutilized in the last 5 min and to identify overload if there is at least one machine overloaded in the last 90 s. These rules are seen in Table 23. During the execution of experiments based on the second scenario, the *Samsara* architecture, in addition to detecting the situations defined in its rules, must adapt according to the generated Adaptation Plan. In each scenario, the Cell is subjected to a set of preset workloads that attempt to exploit CPU usage, organized through the following steps: (i) in the first 3 min, nothing is executed, this period is used to achieve system stabilization; (ii) in the next 55 min, values between 10 to 100% of CPU utilization are generated, executed in descending order, starting at 100% of CPU utilization, reducing 10% at each execution period until reaching 10% of CPU utilization. Each value generated is executed for approximately 5 min and 30 s; (iii) at the end of the first stage of execution, the process is reversed, from then on, values are generated in ascending order.

5.1. RESULTS EVALUATION.

The results evaluation obtained at the end of the experiments characterized that *Samsara* when activated, managed to achieve through its mechanisms the objective of reducing energy consumption, promoting better use of the physical resources of the

computational cloud. Table 3 shows the information related to the *average energy consumption* recorded during the experiments. Comparisons between the two scenarios showed that, by enabling the adaptation mechanisms of the *Samsara* architecture, the environment average energy consumption was reduced by approximately 12.31%. The average consumption was 2.04 MJ in the test scenario where the adaptation mechanisms are kept disabled and 1.79 MJ in the scenario where they are activated, with a reduction of 0.25 MJ. The reduction achieved when converted to watt-hour (Wh^{-1}), shows that it was possible to save approximately 69.44 Wh^{-1}. The average energy consumption values discussed above were obtained with a 95% confidence interval (CI).

Table 1
Physical node rules.

Situation	Context of interest	Rules	Action
Overloaded	CPU Usage and Active Virtual Machines	$CPU_{AverageUse} \geq 70\%\ e$ $VM_{Active} > 0$	Notifies Situation
Underutilized	CPU Usage and Active Virtual Machines	$CPU_{AverageUse} \leq 30\%\ e$ $VM_{Active} > 0$	Notifies Situation
Idle	CPU Usage and Active Virtual Machines	$CPU_{AverageUse} \leq 30\%\ e$ $VM_{Active} = 0$	Notifies Situation

Table 2 Cell rules.

Situation	Context of interest	Rules	Action
Energetically inefficient	Physical Nodes Situation	$PN_{underutilized} \geq 2 e$ $Period_{underutilized} \geq 300 s$	Triggers Event (Consolidation)
Energetically optimized	Physical Nodes Situation	$(PN_{underutilized} < 1 e$ $Period_{underutilized} \geq 300 s)$ *ou* $(PN_{overloaded} < 1\ e$ $Period_{overloaded} \geq 90 s)$	-
Overloaded	Physical Nodes Situation	$PN_{overloaded} \geq 1\ e$ $Period_{overloaded} \geq 90 s$	Triggers Event (Balancing)

Table 3
Mean energy consumption.

Adaptation disabled (MJ)	Adaptation enabled (MJ)	Reduction (MJ)	Reduction (W h^{-1})	Reduction (%)
2.04	1.79	0.25	69.44	12.31

To observe the environment behavior concerning energy consumption, 30 rounds of tests were carried out for each scenario, selecting among those that represented the highest and lowest average consumption in their respective scenarios. To validate the values, the local regression model (LOESS) was used. Fig. 7 shows the variation of the instantaneous power of the two scenarios that correspond to the variation curve of the CPU utilization during the experiment. It is possible to notice that the curve that represents the variation of the instantaneous power over 120 min when the adaptation is enabled *(AE)*, for several moments, remains below the curve that represents the scenario in which the adaptation is disabled *(AD)*, both in the comparison between the rounds of higher consumption *(AD4 and AE3)* and in the comparison between the rounds of lower consumption *(AD13 and AE5)*.

This variation in instantaneous power also corresponds to the moments when the physical nodes are deactivated or reactivated, generating a more significant reduction or increase in instantaneous power.

6. RELATED WORKS.

The work developed in [3,4] presents a distributed framework to perform the dynamic consolidation of virtual machines with a characterized agnostic approach regarding the type of workload and the applications executed. It addresses the problem of performing dynamic consolidation of virtual machines by dividing it into four subproblems: (i) detecting periods of underutilization of host resources; (ii) the detection of periods of overload of hosts resources; (iii) selection of virtual machines to be migrated; and (iv) their allocation on the hosts through live-migration. In [6,7] Snooze is presented, a framework for managing virtual machines, aware of energy consumption and with a holistic approach to the use of resources, with a practical application of serving to efficiently manage a data center in production through the use of dynamic consolidation of virtual machines. Its architecture is organized through the implementation of hierarchical and layered management, responsible for collecting information about the use of resources, carrying out their estimation, and determining the scheduling of virtual machines and energy management actions. In [8] is presented an intelligent QoS-aware autonomic resource management approach named CHOPPER (Configuring, Healing, Optimizing, and Protecting Policy for Efficient Resource management). It aims efficiently to schedule provisioned cloud resources automatically and maintains the SLA based on the user's QoS requirements to reduce human intervention and improves user satisfaction. It uses three different phases (monitoring, analyses, and plan and execution) of self-management which have been developed by focusing on important aspects of self-configuration, self-healing, self-protection, and

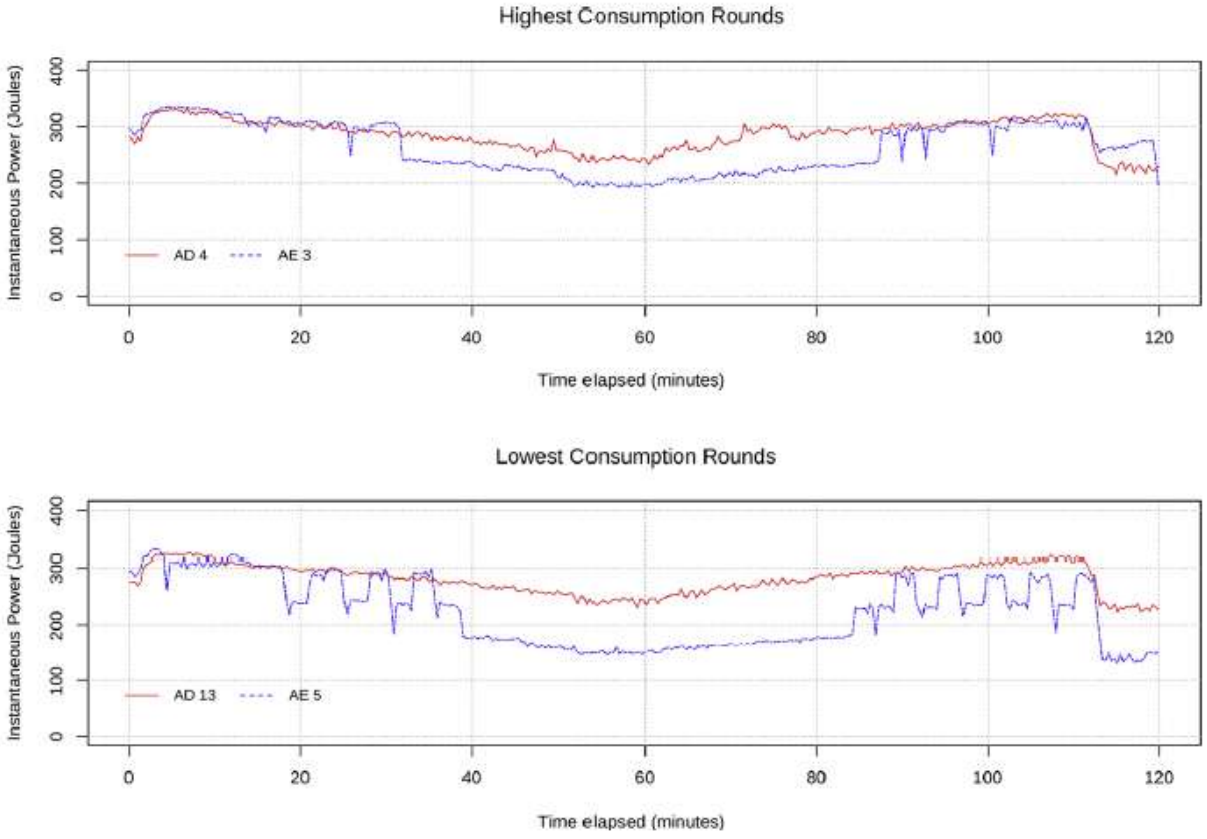

Fig. 7. Instantaneous power over 120 min.

self-optimization. The work of [12] proposed the Energy-Efficient Cloud Orchestrator (e-eco), an orchestrator of energy-saving techniques on a cloud environment that aims to improve the trade-off between power savings and application performance through smart management of a set of power-saving techniques. It allows on-the-fly management of what techniques should be applied based on the application behavior, reducing the impact of these techniques on application performance. A prototype has been implemented in real and simulated cloud environments. The related works presented seek approaches to increase efficiency in the use of resources in large-scale computing environments with a focus on energy efficiency. The architecture presented in this article addresses this issue by proposing to raise the level of abstraction for modeling resource management of a computational cloud, using Situation Awareness to deal with the complexity related to the management of this type of environment.

7. FINAL CONSIDERATIONS.

The main objective of this work was to design an architecture that contributes to the qualification of the mechanisms responsible for managing resources in a cloud environment. It was developed to manage this environment autonomously and with minimal human intervention, seeking to optimize the use of resources in terms of energy efficiency without causing a significant impact on the services provided. Therefore, *Samsara* explores mechanisms for Situation Awareness, both to determine the current state of the environment and to act on it. During its design, aspects that contributed to the flexibility and extensibility of Samsara's mechanisms were considered, allowing the addition of several mechanisms and strategies throughout its architecture. The approach used to achieve a balance between performance and energy consumption consists of exploring the consolidation of workloads, seeking to increase the number of nodes that can be placed in a state of minimizing energy consumption. During periods of increased demand for processing, when identifying situations of overload in the environment, Samsara triggers the adaptation event that balances workloads between physical nodes, which, if necessary, will reactivate nodes at rest. Samsara architecture achieved a reduction in energy consumption of around 12.31% for the evaluated workloads. These experimental results proved to be promising and point to further research. Among the aspects raised to continue the work, the following st and out: (i) explore the use of different algorithms for detecting under-utilization and overload events; (ii) explore support for other mechanisms of action on the environment, such as the resizing of virtual machines and the techniques of CPU pinning and CPU capping; (iii) employ fuzzy logic in the mechanisms of interpretation and decision making; (iv) develop and make available an API for Samsara, as well as a management interface accessible to administrators via a browser; and (v) perform tests subjecting Samsara to real workloads as the objective of evaluating its behavior and performance and allowing its improvement.

REFERENCES

[1] J. Augusto, A. Aztiria, D. Kramer, U. Alegre, A survey on the evolution of the notion of context-awareness, Appl. Artif. Intell. 31 (2017) 613–642.

[2] L. Belkhir, A. Elmeligi, Assessing ict global emissions footprint: trends to 2040 & recommendations, J. Clean. Prod. 177 (2018) 448–463, https://doi.org/10.1016/j. jclepro.2017.12.239. http://www.sciencedirect.com/science/article/pii/ S095965261733233X.

[3] A. Beloglazov, Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing, Ph.D. Thesis, The University of Melbourne, 2013.

[4] A. Beloglazov, R. Buyya, Openstack neat: a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds, Concurr. Comput.: Pract. Exp. 27 (2015) 1310–1333.

[5] A. Beloglazov, R. Buyya, Y.C. Lee, A. Zomaya, et al., A taxonomy and survey of energy-efficient data centers and cloud computing systems, Adv. Comput. 82 (2011) 47–111.

[6] E. Feller, L. Rilling, 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), IEEE Computer Society, 2012, pp. 482–489.

[7] E. Feller, M. Simonin, Y. J´egou, A.C. Orgerie, D. Margery, C. Morin, Snooze: A Scalable and Autonomic Cloud Management System. Ph.D. Thesis, INRIA, Inria Rennes, 2014.

[8] S.S. Gill, I. Chana, M. Singh, R. Buyya, Chopper: an intelligent qos-aware autonomic resource management approach for cloud computing, Cluster Comput. 21 (2018) 1203–1241, https://doi.org/10.1007/s10586-017-1040-z.

[9] I. Lean, Towards Digital Sobriety, 31, 2019, p. 2019. Retrieved.

[10] P. Liu, S. Jajodia, C. Wang, Theory and Models for Cyber Situation Awareness, vol. 10030, Springer, 2017.

[11] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.M. Pierson, A.V. Vasilakos, Cloud computing: survey on energy efficiency, ACM Comput. Surv. (CSUR) 47 (2015) 33.

[12] F.D. Rossi, M.G. Xavier, C.A. De Rose, R.N. Calheiros, R. Buyya, E-eco: performance-aware energy-efficient cloud data center orchestration, J. Netw. Comput. Appl. 78 (2017) 83–96, https://doi.org/10.1016/j.jnca.2016.10.024.

[13] L. Russell, Sensory Substitution: Situational Awareness and Resilience using Available Sensors, Ph.D. Thesis, Carleton University, 2019.

[14] S. Srikantaiah, A. Kansal, F. Zhao, Energy aware consolidation for cloud computing, Proceedings of the 2008 conference on Power Aware Computing and Systems 10 (2008).

[15] H. Vahdat-Nejad, S. Izadpanah, S. Ostadi-Eilaki, Context-aware cloud-based systems: design aspects, Cluster Comput. 22 (2019) 11601–11617.

[16] A. Varasteh, M. Goudarzi, Server consolidation techniques in virtualized data centers: a survey, IEEE Syst. J. 11 (2017) 772–783, https://doi.org/10.1109/ JSYST.2015.2458273. http://ieeexplore.ieee.org/document/7217792/.

[17] J. Wang, Z. Li, H. Zhang, Y. Yi, A study of situation awareness-based resource management scheme in cloud environment, Int. J. Commun. Netw. Distrib. Syst. 24 (2020) 214–232.