Journal of Network and Computer Applications 47 (2015) 99–106

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Cloud resource management: A survey on forecasting and profiling models

Rafael Weingärtner *, Gabriel Beims Bräscher, Carlos Becker Westphall

Networks and Management Laboratory, Federal University of Santa Catarina Florianpolis, Santa Catarina, Brazil

ARTICLE INFO

Article history:
Received 11 March 2014
Received in revised form
5 August 2014
Accepted 26 September 2014
Available online 13 October 2014

Keywords:
Application profiling
Forecasting
Cloud computing
Cloud management

ABSTRACT

With the rise of cloud computing, a huge complexity growth of the structure that is the base of the Cloud happens. Thus, to effectively manage applications and resources it is crucial the use of models and tools that create an application profile which is used to apply forecasting models to determine the most suitable amount of resource for each workload. There are models and tools that address the creation of an application profile to later apply some forecasting technique and estimate the amount of resource needed for a workload. Therefore, this paper aims to present a taxonomy for application profiling models and tools, presenting its main characteristics, challenges, describing and comparing such models and tool. At the end this work presents a discussion about the use of application profiling and its future research trends.

Contents

1. Introduction

Cloud computing is being largely used to deliver services through the Internet mainly because of economic and technical reasons (Urgaonkar et al., 2009; Hall, 2012; Forum, 2013). In this context, there is an issue that comes with the enormous growth of services being delivered by the Cloud, the base structure needed to host the Cloud grows in complexity, which impacts directly on the management costs as stated by Geronimo in (Geronimo et al., 2013).

There are many tools that can create profiles of applications resource usage, each one of them has its own peculiarities and different analysis and predictive models. This way to achieve the optimum management of the Cloud the provider has not to choose the most accurate tools and models but also the one that best suits its needs.

In order to ease future researches on cloud computing management, this paper presents a survey, condensing application profiling models and techniques published known by the authors until

* Corresponding author.
E-mail addresses: weingartner@lrg.ufsc.br (R. Weingärtner), brascher@lrg.ufsc.br (G.B. Bräscher), westphal@lrg.ufsc.br (C.B. Westphall).

http://dx.doi.org/10.1016/j.jnca.2014.09.018
1084-8045/

the writing of this paper. The techniques are going to be described, compared and discussed. Thus, at the end this paper we present open research challenges in this area.

This paper is structured as follows. Section 2 describes the role of application profiling on the cloud management, presenting needs to perform it, its main characteristics and the challenges faced when applying it. Section 3 presents techniques and models published until the moment of the writing of this paper known by the authors, presenting its characteristics, pros and cons and at the end comparing each one of them. Section 4 opens a discussion about the use of application profiling on cloud based applications, pointing open challenges to be researched in this field. Section 5 concludes the paper, wrapping up everything that has been discussed so far.

2. Application profiling

Application profiling is a technique used to describe the use of computing resources by an application and its expected behaviors. It should be used by cloud providers to better understand and manage applications and resources. Figure 1 shows the 9 main characteristics of application profiling, 3 reasons to use such tools or models and the 4 challenges faced up when applying them.

2.1. The need for profiling

Considering the growth of cloud computing and that resource utilization impacts directly in costs, it can be pointed out three main reasons to perform application profiling.

- Application management – environments in which applications share resources have to predict needs for resource properly. This way, it can be allocated the amount needed for each workload to perform its job as expected by its end users. Therefore, in order estimate the amount of resources that should be allocated it is necessary an accurate tool to predict applications' need. Thus, preventing service degradation generated by resource contention that occurs when applications compete for resources;
- Resource management – in order to optimize resource utilization it is essential a model that predicts the amount of resource that best suits each workload. Enabling cloud providers to consolidate workloads while maintaining service level agreements (SLAs);
- Cost management – in a cloud environment the costs are directly bound to the amount of resource used to provide an application/service. Therefore, using accurate models it is possible to consolidate workloads causing little or no impact on application performance while reducing the costs with management and provisioning.

Moreover, the global understanding of the application needs of resource and how each resource affects its behaviors is fundamental to effectively manage applications and services in the Cloud.

2.2. Profiling characteristics

The creation of an application profiling involves the collection, processing and analysis of different data sets. These sets can be traces of resource usage, such as CPU, memory, network bandwidth or metrics related to provided applications/services such as number of requests that is being served, the application's architecture, etc.

Hereby we present a set of characteristics that should be considered into cloud forecasting, management and profiling models in order, not just to optimize resource usage, but also to provide quality of experience (QoE) and Quality of service (QoS). On one hand, the first one is related to the experience that end users have when accessing services/resources. On the other hand, the second one is related to SLAs, assuring that providers comply with the defined set of rules.

Therefore, it can be said that the model that represents the state of the art on cloud application profiling, forecasting and management should have the following characteristics:

- Accuracy – when it is collected traces of resource usage to create a historic database, the tool/model used to collect this data should be accurate, not just to account the amount of resource that is being used directly by an application, but also the amount of resource used to manage the application itself. That extra resource should be taken into consideration by a profile, management and forecasting model. Hence, physical nodes that may be elected to host the workload have to have the amount of resource needed by the application plus the amount required to manage it.
- Application design – today, most of applications are developed binding together different components such as database server, application server and front-end server. When the application is deployed in a cloud each one of those components is normally configured into distinct virtual machines. This way, if we scale up the amount of resource available for one of those components to avoid service degradation due to a sudden increase on requests, it is also needed to scale up the amount of resource proportionally on layers that are interdependent. Otherwise, we solve a problem in one of the layers pushing it to a dependent one.
- Background workload – it is needed to monitor the background workload that the physical host has, in order to identify interferences that one application can have on another. Thus, enabling the identification of incompatible applications, which cannot share the same physical server, hence they compete for the same resource.
- Historic data – it is essential the collection and store of resource usage traces. Every single resource that could affect the application behavior should be monitored and stored. These traces can be used to detect patterns of loads that may happen over time.
- Migration – monitoring future needs of resource is vital. This way, it is possible to identify that a physical host is running out of resource in time to activate the migration process before the server gets flooded and the application suffers from service

Fig. 1. Mind map characteristics, challenges and needs.

degradation. The migration process has high computing costs, therefore, it should be triggered before the server gets flooded, otherwise applications may suffer from resource contention caused not just by applications competing for resource, but also from resource contention generated by the migration process itself.

- Networking affinity – workloads deployed in different hosts communicate with each other using physical networking structures. Therefore, workloads that most communicate with each other should be placed into the same physical node or in the nearest one, in order to avoid networking hops. Hence, applications may suffer service degradation due to flooded network. Moreover, multi-tier applications could benefit from this characteristic, speeding up the communication between layers, hence their packages will be exchanged in memory.
- Overhead – the application profiling models need to constant monitor a variety of characteristics of applications and physical servers. It also needs to process and analyze those data in a real time manner to support the cloud management processes. Hence, those models should be aware of the overhead that is caused by them, and try to minimize the impact that it has on services provided by the Cloud.
- Request types – it is related to the collection and classification of request types that the application is serving to future correlation and analysis with traces of computing usage. This way, it could be identified different groups of request ranging from most sensitive ones that need priority to the ones that can suffer some delay in times that servers are loaded.
- Service Level Agreements (SLA) – by monitoring constantly the SLA, providers have means to tune up the amount of resource allocated to workloads. SLAs are the guidelines to be followed toward quality of service (QoS) assurance. Moreover, there is the quality of experience (QoE) that is the behavior perceived by end users in which SLA agreements play an important role. Therefore, providers should carefully manage SLAs, hence they impact on QoS and QoE.

Namely, QoE stands in the perspective of the end-user, different from QoS, which is focused on the provider perspective. According to Hobfeld in (Hofeld et al., 2012) QoE does not merely rely on QoS metrics, it has the unique and subjective experience that each end user had when she/he uses the system. Moreover, as stated by Zapater and Bressan in (Zapater and Bressan, 2007), QoE gives a way to understand end users' needs and desires in a system. Thus, Zapater divided QoE and QoS in different domains, as presented in Fig. 2, QoS is related to transport, network and application layers, while, QoE is comprised by service and its end users.

After all, if SLAs are not fulfilled, QoE hardly would be accomplished, hence the second one relies on QoS achieved through SLAs agreements. In contrast, if QoS is guaranteed, it is a step toward QoE assurance, QoE is subjective, relying also on users experiences, therefore, even though if QoS is assured by providers, it does not mean that the QoE would be achieved, hence, users may not like some UI features or they may miss some functions.

Even though playing an important role to an optimum cloud management, none of the application profile techniques or models that will be presented here fully apply all of the characteristics shown here. Furthermore, most of the paper that will be presented treated the cloud as a static environment running its tests in a single box, however, the cloud is a rather dynamic environment with workloads being deployed and destroyed all the time. Therefore, those models and techniques are not yet ready to properly manage a dynamic environment such a Cloud.

2.3. Profiling phases and its challenges

Application profiling has four (4) phases shown in Fig. 3 which are data granularity definition, monitoring, processing and storing. Each one of those phases has its own challenges to be dealt with ranging from data definition to storing, therefore we have drawn a few points that should be considered into each one of those phases:

- Data granularity definition – this is the bootstrap phase in which specialists have to define which metrics are going to be monitored and take into consideration to further phases, it is a vital step in which further phases rely on. On one hand, an overly granularity could make the model inapplicable, because data collection, processing and management would be costly. On the other hand, a sparse granularity could make models loosely, which could make it hard to uncover workload patterns.
- Data monitoring – as stated by Aceto in (Aceto et al., 2013), data collection should neither affect provided services nor be intrusive. Monitoring tools should cause less overhead possible, in order not to compete for computing resource with applications. Moreover, they cannot be deployed into users workloads, those tools should monitor resource usage from providers' perspective, monitoring platforms in which users' workloads are deployed.
- Data storing – this phase is affected directly by data granularity definitions and the processing step. Hence, every output of each step is stored to further analysis and correlation. As all of the previously presented phases, it cannot have impact on provided services.
- Data processing – this is the phase in which data collected and stored are served as input to forecasting and management models. Therefore, considering the dynamic nature of the cloud, the processing should be capable to be executed without impacting on provided services. Hence, it is frequently recalculating the amount of resource needed for each workload of the cloud.

Figure 3 pictures how each phase of application profiling interacts with each other. Data granularity definition is the kickoff, in which specialists define which metrics are going to be monitored and how granular its monitoring will be. Having defined the set of metrics that

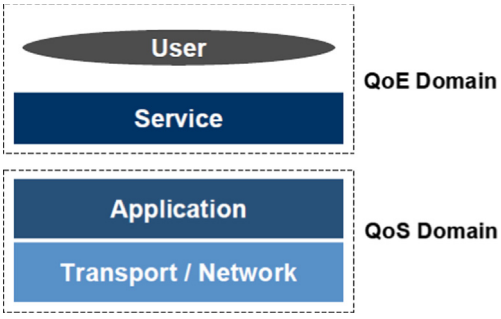

Fig. 2. QoE/QoS layered model, taken from Zapater and Bressan (2007).

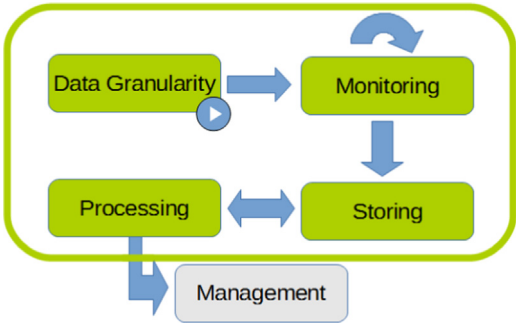

Fig. 3. Application profiling phases.

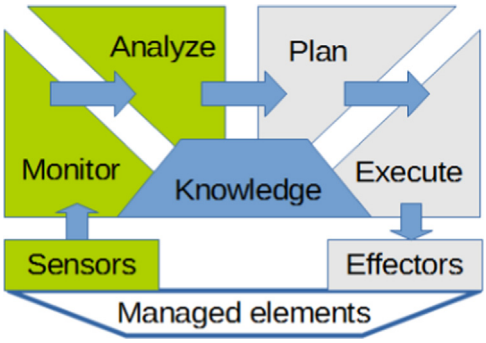

Fig. 4. MAPE-K autonomic loop.

will be monitored, we can start the monitoring loop in which is constantly gathered data to store and feed processing. The storing phase is fed by monitoring and processing phases, the output of the forecasting models is stored to further analysis and derivation of forecasting errors factors that can be used to tune up the forecasting model. The processing step gets stored data to feed some analysis and forecasting model which will output the workloads' future needs for computing resource. Those predictions are then stored and sent to management, in order to take some action and maintain QoS and QoE.

Those four steps above presented in this paper are the backbone for application profiling in which the main goal is to analyze and understand the cloud environment. Therefore, application profiles play an important role in QoS and QoE management, and resource optimization.

Additionally, the presented profiling and forecasting phases are a part of MAPE-K autonomic loop (Monitor, Analyze, Plan, Execute, Knowledge) (Kephart et al., 2003), pictured in Fig. 4. Thus, they comprise the highlight green processes presented in Fig. 4. Furthermore, the monitoring phase goal is to capture the cloud state in a given point in time while the next phase goal is to analyze that data and predict the computing resource needed for each application/service in order to keep performing its jobs properly.

Generally speaking, after profiling the managed elements and forecasted the computing resources needed by workloads, the MAPE-K process shown in Fig. 4 continues with the planning and execution phases. On one hand, the planning phase is responsible to create a path of actions to be taken in order to optimize resource utilization, maintain QoS and QoE. On the other hand, the execution phase deals with the execution of actions into the managed environment. Moreover, according to the description created by Manvi and Shyam in (Manvi and Shyam, 2014), the resource management tools and techniques must embrace resource provisioning, allocation, mapping and adaptation. Thus, in Manvi and Shyam (2014) is found a survey on resource management tools and models.

The phases and its challenges presented above are the main difficulties faced up when one tries to create models to manage a Cloud. Most of those problems that arise when developing profiling and forecasting models are due the dynamic nature of the Cloud. Therefore, any work that may claim to develop a novel solution to manage a cloud shall consider the dynamism that is naturally present in a cloud environment.

3. Profiling and forecasting models

Cloud computing is being more and more used to deliver on demand content; this movement toward the Cloud creates new management challenges. Hence, a cloud environment is more complex than a conventional data center. We hereby present papers that have made some progress developing techniques and models for the application and resource forecasting and management that could be applied in a Cloud.

Urgaonkar in (Urgaonkar et al., 2002) and subsequently in a more comprehensive work in Urgaonkar et al. (2009) presented models to create an application profiling without the previous knowledge of the application. He placed applications in nodes, making sure that the application does not suffer any type of background interferences of concurrent applications. Thus, he introduced the concept of a percent (%) of overbooking, as airline companies do when selling tickets to allocated more resource to applications then the provider actually has. It is mainly based on historic traces of resource usage; it does not make any correlation or analysis of the collected data to predict future needs for computing resource. Therefore, it takes applications observed resource usage as static, which does not fit to cloud application, given their dynamic and elastic nature.

Geronimo in (Geronimo et al., 2013) proposed a model to manage resources on a green cloud, using public clouds to deal with unexpected peaks of demand. It satisfied SLAs while being energetically more efficient than traditional approaches. Thus, optimizing the physical resource usage and reducing costs. It used traces of resource usage and request load to find usage patterns and manage virtual machines from a private cloud.

Dawoud in (Dawoud et al., 2012) pointed out that the use of simple historic traces can lead to ineffectively decisions in management. Especially when dealing with web applications which commonly use multi-tier architectures. This work proposed the correlation of historic traces of resource usage with factors such as workload type, request types, and also the resource contention factor that exist in public clouds. Despite that, it does not correlate the load that each tier creates on subsequent one. In other words, this work would predict the resource contention in one tier of the application, it would scale up the amount of resource for that tier, but the problem would still persist. Hence, the problem of resource contention was not properly fixed, but changed its location to another tier of the application.

Paper (Do et al., 2011) proposed a model that takes into account the interferences created by background workloads such as concurrent services and operating system programs. Hence, every single one of those applications is competing for computing resources, therefore, their interferences should be taken into account when placing a workload on a cloud. At the end it proposed the use of a canonical correlation analysis to identify the resources that most affect the workload behavior; it later took that output into a management algorithm that would find a suitable host that has the resources most needed available.

Gong in (Gong et al., 2010) presented a model called PRESS (PRedictive Elastic ReSource Scaling for cloud systems) that addressed the prediction of future needs of resource that an application can have. This is a lightweight model that does not require any advanced technique/tool to collect resource usage traces to create an application profile. It used two approaches to apply its prediction technique; on one hand, it uses Fast Fourier Transform (FFT) to discover the dominant frequencies and identifies patterns of resource usage; on the other hand, when it does not discover a sufficient amount of usage patterns to apply its prediction algorithm, it uses Markov Chain with a finite number of states that the application can reach to predict the need of resource for a short period in the future.

Furthermore, Shen in (Shen et al., 2011) extended the work developed by Gong in (Gong et al., 2010), in order to make the scale up and down of resources with a smarter approach, trying to lower the SLAs violations. It applies paddings as a security measure, on the amount of resources predicted by the PRESS model. In addition, it uses statistics of SLAs violations to tune paddings. Moreover, it added a forecasting model that predicts when a workload (virtual machine)

needs to migrate to a more suitable host to avoid service degradation. Thus, the migration process could be triggered before the physical host runs out of available resource and the application suffers service degradation.

Ren in (Ren et al., 2010) extended the Digital Continuous Profiling Infrastructure(DCPI) model. The DCPI provides means to continuously monitor a datacenter structure with less overhead possible. The paper presents an extension to make the monitoring of datacenter's applications in a non-intrusive way with almost no overhead. Thus, it says that we do not need to monitor the whole datacenter, we just need to monitor a portion of it. Hence, workloads running in the datacenter are most probably similar if not equal; we would not need to monitor every single one of them. Therefore, it decreases the amount of resources needed to monitor, store and analyze the data generate by the monitoring. However, a cloud is not as heterogeneous as a datacenter, this way, taking a snapshot of a small portion of the cloud and considering it as the whole cloud state to manage applications and resources could lead to mistaken actions.

Wood in (Wood et al., 2008) proposed an autonomous model that has the ability to correlate the resource usage on native environment (environments in which the application would run on physical hosts) to a virtualized environment. Thus, automatizing the work of estimating resources needed to an application that is migrating from physical to virtual environment, the paper achieved a higher precision at the same time that it eased the job of such migrations. This work creates a profile of the application needs in terms of hardware resource, mapping needs in physical environments to needs that the application can have in virtual environments.

Etchevers in (Etchevers et al., 2011) addressed the lack of automation that management solutions have when dealing with applications that are published in the Cloud. The paper proposed a model to describe multi-tier applications. Therefore, it creates a formal way to describe every requirement that an application can have such as operating systems, middleware, third party software, and every connection that can exist between each layer of the application.

Di Cosmo in (Di Cosmo et al., 2012) as the work developed in Etchevers et al. (2011) tackled the lack of automation to manage cloud application. However, this paper proposed a formal way to describe the layers needs in a quantitative way instead of qualitative as Etchevers et al. (2011). It describes a way to write down a specification of each layer in quantitative terms. E.g. a multi-tier application that is composed of a web front-end and a back-end server, it may require one instance of the front-end layer to three instances of the back-end server. This way when the scale up and down of resources happens on the front-end, it also should happen at the back-end layer proportionally.

Hulkury and Doomun in (Hulkury and Doomun, 2012) presented an integrated Green Cloud Computing Architecture that addresses the workload placement problem, finding the best place to deploy workloads based on their theoretical energy consumption. A manager (cloud client side) would have to provide workload SLAs description, network and server specifications, to calculate the energy consumption of it in each cloud scenario (local, private or public Cloud). As proposed in Werner et al. (2011), it suggests the use of public clouds as an extension of private Clouds, routing workloads between them when it can be profitable. Sadly, it depends on some information that, in most cases, the client does not have access, like the energy consumption of the public Cloud elements in order to fully satisfy the model requirements. It also mentions the use of XML to store SLAs and QoS constraints in the Cloud Manager; however it does not define any standard to do that.

Vondra and Sedivy in (Vondra and Sedivy, 2013) based on paper (Vondra and Sedivy, 2012) in which is presented an ongoing work that aim the maximization of the cloud computing structure filling the gaps created by web servers workloads with batch processing workloads. These gaps are periods in which there is either a lower request or no requests at all to web servers. Thus, it used time series forecasting techniques to predict web servers' load and then decide whether or not to deploy a batch workload. Hence, it can be found two distinct workloads in a Cloud, interactive and batch ones. On one hand, interactive workloads are the ones such as web servers and Web systems. On the other hand, batch workloads are related to scientific computation, data mining tools, etc. The best way to increase the private cloud optimization is the mix of interactive workloads with non-interactive ones. Therefore, batch processes would be used to fill the gaps (resource availability) in interactive workloads. Thus, it is an ongoing work and probably is going to be improved and tuned by its authors; it has some problems that need and probably will be addressed such as:

- If a sudden spike on request happens, and there is already some batch workloads being processed they have to be stopped in order to free up resource to the interactive workload. Thus, it would be interesting not to stop the batch workloads and loose hours of work, but instead suspend the virtual machine that is running it to latter resume its job;
- The prediction technique used requires a rather high amount of computing resource to perform its task and predict the interactive workload. As already stated, it is needed a prediction tools that does its job while consuming the minimum amount of CPU, RAM and storage;
- The prediction tool was just based on historical traces; it would be interesting to add some more data into those models in order to create a better prediction, given the dynamic nature of the Cloud.

Bankole and Ajila in (Bankole and Ajila, 2013) stated that to efficiently meet the SLAs, it is needed to pro-actively predict and provision future VM needs of computing resource. The paper developed and compared three different machine learning techniques to predict needs of resources; Support Vector Machine (SVM), Neural Networks (NN) and Linear Regression (LR). Thus, it was not just used traces of resource usage like CPU, memory and network, but also SLA metrics for response time and throughput. Thus, it achieved the best prediction overall performance with SVM, which was considered to provide the best prediction model. Although it was said that the author is developing prediction models for multi-tier applications, it was not considered the load that an increase of resource in one tier could impose on another. At the end it seemed to have just tried to predict and adjust the CPU needs, letting aside metric like memory and networking that are as vital as CPU given the dynamism of the cloud.

Elprince in (Elprince, 2013) discussed that the cloud infrastructure has been growing over the year, and thus, its complexity is making management high costly. It proposed an autonomous cloud management model that would predict the workload needs and automatically adjust its resource. The client would initially provide to the cloud provider the desired response time and type of workload it is going to run. However, it was not presented how it would be written. Despite that, the work is interesting, predicting not just CPU needs, but also memory and I/O disk latency. It was also proposed a prediction model that would be able to auto tune up its prediction using a Fuzzy Inference System. Thus, he implemented and tested his model with different machine learning (ML) techniques, spotting the most suitable one to work as the core of his prediction model. At the end, he compared different ML techniques and showed the differences between them, some could be more accurate with a higher cost, while others could not be as accurate, but had a low cost. Furthermore, his experiments had

shown that the best results were obtained applying Model trees via Bagging technique. It also has showed that his Fuzzy Tuner, which is responsible to tune the predictions, can be an excellent approach to offer service differentiation among clients.

Tak in (Tak et al., 2013) designed a way to simplify the migration process of legacy applications into the Cloud. It presented a technique called PseudoApp, which benchmarks the application in native environment and then reproduces the captured behaviors of resource usage in the targeted Cloud. All request types are mapped and stored, and at the end it is possible to identify the resource that a specific request is demanding more or less. This work is remarkable, mapping in a fine granular way all the application requests and latter mapping them with its respective resource usage, enabling them to be reproduced in public clouds, spotting the one with less resource contention.

The Organization for the Advancement of Structured Information Standards (OASIS) proposed (Standard, 2013), Topology and Orchestration Specification for Cloud Applications (TOSCA). It provides a language to describe service components and their relationships, enabling the description of management procedures that create or modify services using orchestration processes. It also has the ability to specify the operational behavior of applications, how servers are deployed and connected. However, it does not provide means to specify how applications can be modified at run time.

Han in (Han et al., 2013) introduced a framework called Elastic-TOSCA that extended (Standard, 2013). It improved the TOSCA framework with the ability to monitor running applications, describe quality of service (QoS) and depict plans to scale up or down applications. Thus, it became possible to understand the behavior of the application and then use SLA constraints to manage the scale up or down of resources according to the load.

Du in (Du et al., 2013) presented a model capable of predicting virtual machines performance and interferences between VMs. It aimed to develop accurate and functional management architecture; it used artificial neural network to provide ability to predict performance of all VMs and their workloads. Thus, It embraced accuracy, considering resources used by the Xen domain0 to perform its management tasks, the domain0 is the management domain that has the native host drivers and performs I/O operations on behalf of all guest domains. It also used the relationship between VMs, their load and their interferences when competing for resources into the proposes forecasting model.

Sonnek in (Sonnek et al., 2010) developed a migration technique based in network affinity. It aims to reduce the communication overhead between two virtual machines by placing them in the same host. The main goal is to avoid scenarios in which a pair of VMs is connected by a slow link; otherwise the network can become a performance bottleneck. It was proposed an affinity-based virtual machine placement system, focused not in understanding computing resources usage (as CPU and memory), but the existing communication dependencies among workloads.

Keller in (Keller et al., 2012) suggested network affinity as an important aspect to be considered when migrating virtual machines. It dealt with problems of applications that are developed in a multi-tier way and the need to place those tiers in the same host to increase performance and reduce bandwidth usage. Therefore it proposed LIME (Live Migration of Ensembles), an algorithm to perform migration of applications' layers and their virtual network structure, guaranteeing affinity placement as well as keeping configurations and behaviors. In addition, it allows cloud providers to manage their resources, provide live maintenance, or perform simpler and safer tests, maintaining the cloud environment reliable and dynamic.

Chen in (Chen et al., 2013) proposed a VM allocation method based in network affinity. Basically, it creates sets of VMs, combining virtual machines that have some dependence. By creating distinct sets of machines with network affinity, it is possible to manage their allocation in order to maintain every element of the same set as near as possible. Consequently the traffic between physical machines will decrease ensuring a better usage of the network. It saves network bandwidth and increases services performance.

Akula and Potluri in (Akula and Potluri, 2014) presented a working in progress in which an algorithm for dynamic consolidation of virtual machines is shown. It consolidates VMs into physical servers that are not fully loaded, in order to power off the idle ones. Thus, for the placement of virtual machines it builds a communication graph that maps the relationship between VMs, which enabled them to re-allocate VMs in bulks. The work (Akula and Potluri, 2014) has discussed some vital points of cloud computing, such as the relationship between VMs, resource optimization (consolidation) and the necessity of migration. However, it has not been implemented, which makes it hard to measure its efficiency. Nevertheless, Akula stated that they intent to develop his proposal and integrate it with LIME architecture (Keller et al., 2012).

Table 1 matches the core characteristics of application profiling presented previously with the properties of presented works. Thus, visually exposing the gaps that each work has, and that can be considered as challenges to be addressed in future papers that aim to improve application profiling, forecasting and management for the cloud.

Going through Table 1, we noticed that almost none of the presented publications correctly measure the amount of resource that is effectively used by a workload, which can lead to mistaken actions when managing Cloud's applications and resources. As pointed out before, to correctly measure the resources needed by a workload it is crucial to account the resources used by it directly and the amount used indirectly to manage the workload itself as Du does in (Du et al., 2013).

Furthermore, most publications claim to manage a cloud environment, optimizing physical resource usage while not impacting in provided services. However, just a few of them take into consideration the dynamic nature of the cloud. Most of the presented papers developed their models and techniques and tested them in a single box environment with well know workloads. Therefore, those models were not built to deal with the intrinsic dynamism of the cloud in which new services and applications are deployed and destroyed all the time.

4. Challenges and directions for cloud based application profiling

With the rapid adoption of cloud computing and its use to deliver complex and critical applications, we foresee the following challenges to be tackled by future researches.

4.1. Minimum performance assurance

It is needed not just to guarantee the availability of applications, but also to assure minimum performance level as proposed by Urgaonkar and Schad respectively in Urgaonkar et al. (2009); Schad et al. (2010). Furthermore, without a minimum level of performance applications can have unexpected behaviors which impacts directly in QoE (end users experience when using provided services).

Toward a model that guarantees minimum performance, it is needed to understand better the cloud application. As Geronimo and Hullkury proposed respectively in Geronimo et al. (2013); Hulkury and Doomun (2012). It is needed to create a standard to write down the SLAs in a more comprehensive way, enabling the description of applications and their tiers relationships and computing resource needs. The application description would be created by the developer or cloud sponsor who is responsible to develop and maintain the

Table 1
proposed models versus profiling core characteristics.

Publications	Characteristics								
	Accuracy	Application design	Background workload	Historic data	Migration	Networking affinity	Overhead	Request types	SLA
Urgaonkar et al. (2002)				X					
Wood et al. (2008)				X					
Urgaonkar et al. (2009)				X					
Gong et al. (2010)				X			X		X
Ren et al. (2010)							X		
Sonnek et al. (2010)				X		X	X		
Do et al. (2011)			X	X					X
Shen et al. (2011)				X	X		X		X
Etchevers et al. (2011)		X/2							
Dawoud et al. (2012)				X				X	
Di Cosmo et al. (2012)		X/2							
Hulkury and Doomun (2012)		X/2		X					X
Keller et al. (2012)		X				X			
Geronimo et al. (2013)				X					X
Vondra and Sedivy (2013)				X					
Bankole and Ajila (2013)				X					X
Elprince (2013)				X			X		X
Tak et al. (2013)								X	
OASIS Standard (2013)		X							
Han et al. (2013)		X							X
Du et al. (2013)	X		X	X		X			
Chen et al. (2013)				X		X			
Akula and Potluri (2014)						X			

application. Thus, it would be a clear way to describe the application needs and behaviors to the cloud provider.

Moreover, the language developed by OASIS in Standard (2013) and its extension proposed by Han in (Han et al., 2013) match all the requirements that (Geronimo et al., 2013; Hulkury and Doomun, 2012) pointed out. However, it is needed some work to take those documents generate using Tosca (Standard, 2013) or Elasctic-Tosca (Han et al., 2013) into a cloud management model.

Therefore, a formal application description would ease the task to assure minimum performance to cloud applications which is essential to guarantee QoS and QoS while improving resource usage.

4.2. Autonomic cloud computing

Cloud computing provides resources in a reliable, secure and cost efficient manner. Therefore, it requires optimization in multiple layers such as infrastructure, platform and application (Buyya et al., 2012). Thus, clouds are heterogeneous environments which are growing in complexity and size, and resource management is a vital aspect of it. Moreover, Mendes in (Mendes et al., 2014) said that autonomic cloud is one of the solutions for the management issues that arise with Cloud growths. He said that it is needed models that give autonomicity to cloud management tools in order to cope with the dynamic and elastic nature of clouds.

It is humanly impossible to manage a cloud that is growing in complexity exponentially; there are too many variables to be considered when managing application and resources in a Cloud, especially a public one which has multiple interests and clients (Mendes et al., 2014). Thus, to feed an autonomic model, an accurate model is needed to predict services computing needs.

As presented by Buyya in (Buyya et al., 2012), management tools need to be automated and improved to dynamic provisioning of resources. Autonomic systems have characteristics such as self-optimizing, self-monitoring and self-healing which could benefit the cloud environment. However, without a proper profiling and forecasting model to feed the autonomic management model, it may not achieve its true potential.

4.3. Standard test beds

During our research it was noticed that each one of the presented proposals was validated with workloads that have a well-known behavior by their author, which can lead to models that work just on specific scenarios, but not in a cloud environment with all of its peculiarities.

Furthermore, none of the workloads used properly represented the dynamism naturally found in a Cloud environment. Therefore, the proposals were not actually tested with a cloud environment and may not match the peculiarities found in a cloud such as multiple actors, decisions and interests, competing for the same amount of computing resource at the same time.

The fact that each proposal used a different workload also made it impossible to compare them directly. Hence, each one was applied in unlikely scenarios with different variables and over distinct structures. As already stated in Aceto et al. (2013), we foresee the need to create standard test beds to be used to validate cloud management models. Therefore, it would be possible to compare proposals' results directly without the need to re-code them and re-run the tests every time a new model is shown.

Furthermore, those standards test beds should be created using distinct scenarios mixing the most different types of workloads, striving to simulate the dynamic nature found on clouds in which different services are deployed, updated and destroyed all the time.

4.4. Forecasting and profiling models optimization

We noticed that most papers tested their models against small workloads that do not simulate the dynamic nature of the cloud. Those models may fall apart when applied in a large and complex environment such as a cloud; hence, the amount of data needed to be monitored and processed will be larger than what was used in theirs experimentation.

Profiling and forecasting models must deal with large amount of data generated by cloud element such as hosts, VMs, platforms, applications, networking elements, storage, etc. As already discussed those models are essential to maintain services' QoS and QoE, therefore, if they are not properly designed to deal with the

peculiarities of the cloud, applications and services may suffer resource contention and service degradation.

In addition, Manvi and Shyam in (Manvi and Shyam, 2014) surveys the cloud resource management scenario, and concluded that, due to its complexity, a more distributed and scalable approach is necessary to support all peculiarities found in that environment. Thus, Monitoring and forecasting systems must be built to deal with millions of elements in a rather dynamic environment such as a cloud. After all, approaches that can manage cloud challenges are needed in order to keep its competitiveness.

5. Conclusion

The paper presented an application profiling and forecasting taxonomy, describing reasons to perform it as well as their main characteristics and challenges. Thus, we also introduced it was a part of the MAPE-K autonomic loop in which it is responsible to monitor and analyses collected data to feed the planning and execution phases of the loop.

We also described, compared and discussed profiling and forecasting models and techniques published known by the authors until the writing of this paper. Thus, we classified each of the discussed and presented paper into the taxonomy that was presented.

In conclusion, we presented and discussed open challenges to be faced up on the area of profiling and forecasting models and techniques to cloud environments in which the cloud dynamism is one of the greatest barriers to be dealt with. Thus, we discussed that autonomic computing may come in hand to deal with the dynamic nature of the cloud and its growths in complexity and size.

References

Aceto G, Botta A, De Donato W, Pescapè A. Cloud monitoring: a survey. Comput Netw 2013;57(9):2093–115.

Akula G, Potluri A. Heuristics for migration with consolidation of ensembles of virtual machines. In: Sixth international conference on communication systems and networks (COMSNETS), 2014; 2014, p. 1–4. doi: http://dx.doi.org/10.1109/COMSNETS.2014.6734927.

Bankole A, Ajila S, 2013. Cloud client prediction models for cloud resource provisioning in a multitier web application environment. In: IEEE seventh international symposium on service oriented system engineering (SOSE), 2013; 2013, p. 156–61. doi: http://dx.doi.org/10.1109/SOSE.2013.40.

Buyya R, Calheiros RN, Li X. Autonomic cloud computing: open challenges and architectural elements. In: Third international conference on emerging applications of information technology (EAIT), 2012, IEEE; 2012, p. 3–10.

Chen J, Chiew K, Ye D, Zhu L, Chen W. Aaga: Affinity-aware grouping for allocation of virtual machines. In: IEEE twenty-seventh international conference on advanced information networking and applications (AINA), 2013, IEEE; 2013, p. 235-242.

Dawoud W, Takouna I, Meinel C. Dynamic scalability and contention prediction in public infrastructure using internet application profiling. In: IEEE fourth international conference on cloud computing technology and science (CloudCom), 2012, IEEE; 2012, p. 208–16.

Di Cosmo R, Zacchiroli S, Zavattaro G. Towards a formal componentmodel for the cloud. In: Eleftherakis G, Hinchey M, Holcombe M, editors. Software Engineering and Formal Methods, Vol. 7504 of Lecture Notes in Computer Science. Berlin Heidelberg: Springer; 2012w. p. 156–71. http://dx.doi.org/10.1007/978-3-642-33826-7_11.

Do AV, Chen J, Wang C, Lee YC, Zomaya AY, Zhou BB. Profiling applications for virtual machine placement in clouds. In: IEEE international conference on cloud computing (CLOUD), 2011, IEEE, 2011; p. 660–67.

Du G, He H, Meng F. Performance modeling based on artificial neural network in virtualized environments. Sens. Transducers 2013;153:37–44.

Elprince, N. Autonomous resource provision in virtual data centers. In: IFIP/IEEE international symposium on integrated network management (IM 2013), 2013; 2013, p. 1365–71.

Etchevers X, Coupaye T, Boyer F, DePalma N. Self-configuration of distributed applications in the cloud. In: IEEE international conference on cloud computing (CLOUD), 2011, IEEE, 2011; p. 668–75.

Forum CI. Uk cloud adoption and trends for 2013, Technical report, Cloud industry forum (October 2013).

Geronimo G, Werner J, Westphall C, Westphall C, Defenti L. Provisioning and resource allocation for green clouds. In: ICN 2013, The twelfth international conference on networks; 2013, p. 81–86.

Gong Z, Gu X, Wilkes J. Press: Predictive elastic resource scaling for cloud systems. In: International Conference on Network and Service Management (CNSM), 2010, IEEE, 2010; p. 9–16.

Hall P. Opportunities for csps in enterprise-grade public cloud computing, Technical report, OVUM (May 2012).

Han R, Ghanem MM, Guo Y. Elastic-tosca: supporting elasticity of cloud application in tosca. In: CLOUD COMPUTING 2013, The fourth international conference on cloud computing, GRIDs, and virtualization, 2013; p. 93–100.

Hobfeld T, Schatz R, Varela M, Timmerer C. Challenges of qoe management for cloud applications. IEEE Commun. Mag. 2012;50(4):28–36.

Hulkury MN, Doomun MR. Integrated green cloud computing architecture. In: International conference on advanced computer science applications and technologies (ACSAT), 2012, IEEE, 2012; p. 269–74.

Keller E, Ghorbani S, Caesar M, Rexford J. Live migration of an entire network (and its hosts). In: Proceedings of the Eleventh ACM Workshop on Hot Topics in Networks, ACM, 2012, pp. 109–114.

Kephart J, Chess D. The vision of autonomic computing. Computer 2003;36(1):41–50. http://dx.doi.org/10.1109/MC.2003.1160055.

Manvi SS, Shyam GK. Resource management for infrastructure as a service (iaas) in cloud computing: a survey. J Netw Comput Appl 2014;41(0):424–40 doi: doi:10.1016/j.jnca.2013.10.004 url: ⟨http://www.sciencedirect.com/science/article/pii/S1084804513002099⟩.

Mendes R, Weingartner R, Geronimo G, Bräscher G, Flores A, Westphall C, et al. Decision-theoretic planning for cloud computing. In: The thirteenth international conference on networks—ICN, 2014; 2014.

Ren G, Tune E, Moseley T, Shi Y, Rus S, Hundt R. Google-wide profiling: a continuous profiling infrastructure for data centers. IEEE Micro 2010;30(4):65–79.

Schad J, Dittrich J, Quiané-Ruiz J-A. Runtime measurements in the cloud: observing, analyzing, and reducing variance. Proce VLDB Endow 2010;3(1–2):460–71.

Shen Z, Subbiah S, Gu X, Wilkes J. Cloudscale: elastic resource scaling for multi-tenant cloud systems. In: Proceedings of the second ACM symposium on cloud computing, ACM, 2011; 2011, p. 5.

Sonnek J, Greensky J, Reutiman R, Chandra A. Starling: minimizing communication overhead in virtualized computing platforms using decentralized affinity-aware migration. In: thirty-ninth international conference on parallel processing (ICPP), 2010, IEEE; 2010, p. 228-237.

Standard O. Topology and orchestration specification for cloud applications version 1.0, Tech. rep., OASIS Standard (November 2013). url: ⟨http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html⟩.

Tak BC, Tang C, Huang H, Wang L. Pseudoapp: performance prediction for application migration to cloud. In: IFIP/IEEE international symposium on integrated network management (IM 2013), 2013, IEEE, 2013, pp. 303–310.

Urgaonkar B, Shenoy P, Roscoe T. Resource overbooking and application profiling in a shared internet hosting platform. ACM Trans Internet Technol (TOIT) 2009;9(1).

Urgaonkar B, Shenoy P, Roscoe T. Resource overbooking and application profiling in shared hosting platforms. ACM SIGOPS Oper Syst Rev 2002;36(SI):239–54.

Vondra T, Sedivy J. Maximizing utilization in private iaas clouds with heterogenous load. In: CLOUD COMPUTING 2012, The third international conference on cloud computing, GRIDs, and virtualization, 2012, p. 169-173.

Vondra T, Sedivy J. Maximizing utilization in private iaas clouds with heterogenous load through time series forecasting. Int J Adv Syst Meas 2013;6(1 and 2):149–65.

Werner J, Geronimo GA, Westphall CB, Koch FL, Freitas RR. Um modelo integrado de gestão de recursos para as nuvens verdes. In: CLEI 2011, vol. 1; 2011, p. 1–15.

Wood T, Cherkasova L, Ozonat K, Shenoy P. Profiling and modeling resource usage of virtualized applications. In: Proceedings of the ninth ACM/IFIP/USENIX international conference on middleware, Springer-Verlag New York, Inc. 2008, p. 366–87.

Zapater M, Bressan G. A proposed approach for quality of experience assurance of iptv. In: Digital Society, 2007. ICDS '07. First international conference on the, 2007, p. 25–25. doi: http://dx.doi.org/10.1109/ICDS.2007.4.