Comparative Analysis of Frameworks for the Performance Evaluation of Multi-tier Cloud Applications[1]

[1]G.R. Garay <godofredo.garay@reduc.edu.cu>
[2]A. Tchernykh <chernykh@cicese.mx>
[3]A.Yu. Drozdov < alexander.y.drozdov@gmail.com>
[1]University of Camaguey,
Carretera de Circunvalacíon, km 5, 74650 Camagüey, Cuba
[2]CICESE Research Center, Carretera Ensenada-Tijuana No. 3918, Zona Playitas, Código Postal 22860, Apdo. Postal 360, Ensenada, B.C. México
[3] Moscow Institute of Physics and Technology (State University)
9 Institutskiy per., Dolgoprudny, Moscow Region, 141700, Russia

Abstract. In early stages of a hardware design, when a lot of options need to be consideredquickly, analytic modeling is used. It allows the performance evaluation of proposed systems without requiring complexand costly detailed simulations. Analytical approaches for the performance evaluation of cloud computing environments include Queuing Theory and Control Theory models. Real-Time Calculus (RTC) is a high-level analysistechnique previously proposed for stream-processing hard real-time systems and frequently used toevaluate trade-offs in packet stream processing architectures.The central idea of theModular Performance Analysis with RTC (MPA-RTC) is to build an abstract performance model of a system that bundles all information needed for performance analysis with RTC. In this paper, we address the performance evaluation of multi-tier clouds applications, and compare a Real-Time Calculus-based framework with two classical analytical approaches such as queuing theoretic approaches and control theoretic approaches. We focus on the capabilities of these alternatives for estimating the key Quality of Service parameter - the application response-time. In addition, we discuss the capabilities of each analytical approach for modeling other aspects of cloud computingenvironment such as workload models, task processing models, virtual machine (VM) provisioning, VMs performance interference, autonomic resource management, server consolidation, and cloud scaling strategies (horizontal and/or vertical). The capabilities of MPA-RTCas a valuable tool for the performance evaluation of cloud computing platforms are exposed.

Keywords: Real-Time Calculus, queuing, control, QoS, response-time, cloud computing

DOI: 10.15514/ISPRAS-2015-27(6)-14

[1] The work is partially supported by the Ministry of Education and Science of Russian Federation under contract No02.G25.31.0061 12/02/2013 (Government Regulation No 218 from 09/04/2010).

For citation: Garay G.R., Tchernykh A., Drozdov A.Yu. Comparative Analysis of Frameworks for the Performance Evaluation of Multi-tier Cloud Applications. *Trudy ISP RAN/Proc. ISP RAS*, vol. 27, issue 6, 2015, pp. 199-224 (in Russian). DOI: 10.15514/ISPRAS-2015-27(6)-14

1. Introduction

Virtualization-based resource management in cloud computing environments is usually related to performance improvement, including QoS guaranteeing, energy saving, and others parameters specified in the SLAs.

A number of researchers have focused on SLA (Service Level Agreement)-based objectives (e.g., client-perceived response time, throughput, dependability, reliability, availability, costs, security, confidentiality, etc.).

In order to optimize the system performance, some methods have to be exploited to estimate the possible metrics based on the input of the system. To this end, analytical performance models can be established for the examined applications running upon the virtualized environment.

After the objectives and proper performance estimation approaches are determined (e.g., analytical frameworks), performance analysis need to figure out the best configuration for the placement of virtual machines [3].

In a previous work [32], we discussed a Real-Time Calculus-based approach for the performance evaluation of multi-tier cloud applications, where we only focused on the capabilities of RTC for estimating the Quality of Service parameters such as response time.

In this considerably extended version of the paper, we compare the previously proposed analytical framework with two classical analytical approaches commonly used for the performance evaluation of multi-tier cloud Web applications (see [3-5]) such as queuing theoretic approaches and control theoretic approaches. In particular, we focus on the capabilities of these alternatives that can be employed for estimating Web application response-time. In addition, specific VMs management issues are also analyzed.

The paper is organized as follow. In Section 2, we present the motivation of the work, and give some background information. Existing analytical approaches are presented in Section 3, and the main features of Real-Time Calculus are presented in Section 4. A discussion of the principal findings is presented in Section 5. The paper is concluded in Section 6.

2. Motivation

As a motivation example (Fig. 1), let us consider a system under test (SUT) consisting a three-tier web application [4, 6]. The three-tiers include presentation-tier, application (business)-tier and data-tier, implemented in actual systems as a web server process (P), application server process (B), and database server process (D), respectively (Fig. 2).

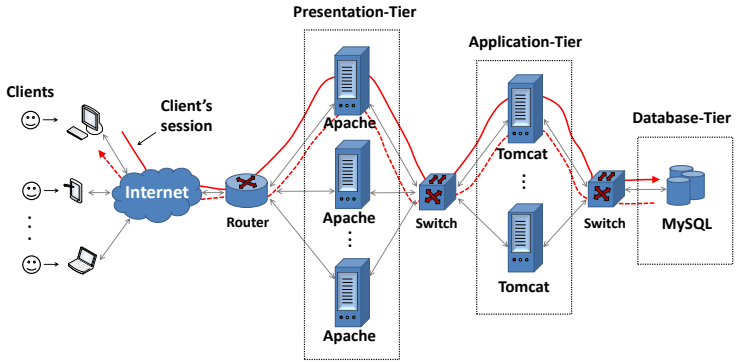

Fig. 1. Imaginary example of a client session on a basic multi-tier application architecture (note that in virtualized cloud platforms, each software server, i.e., Apache, Tomcat, and MySQL, is run inside of a virtual machine).

The first tier named presentation-tier consists of Web server. It displays what is presented to the user on the client side within their Web browsers. For the Web server-tier, it mainly has three functions: (1) Admitting/denying requests from the clients and services Web requests; (2) Passing requests to the application server; and finally, (3) receiving response from application server and sending it back to clients. In this paper, all these tiers will be modeled as software servers.

In our SUT (Fig. 1), a state-full web application is considered. For this reason, the session-based data-access client requests and responses are processed by the same virtual machines (VMs) instances (see Fig. 2).

In practice, multiple deployment scenarios of VMs on physical machines (PMs) may exist. In this paper, we want to answer the following question: can we predict whether the application's response time will violate (or surpass) a pre-specified deadline when application's characteristics at each single tier in isolation are known in advance with certain levels of confidence?

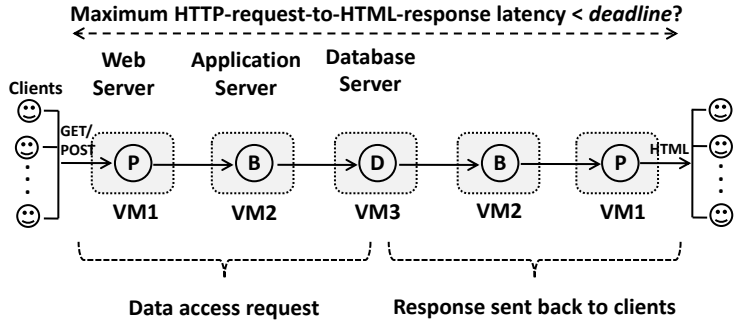

Fig. 2. Focus of attention: Predicting Web-application response-time in cloud computing platform, e.g., does maximum request-to-response latency of a client request will not exceed application deadline (with 95% confidence interval)

3. Existing Approaches

3.1 Queuing models

One of the most popular analytical approaches for the performance evaluation of cloud computing environments [4, 5] is Queuing Theory (QT) [7]. Here, we present a short introduction to QT [8], which summarizes the most important issues of this analytical approach.

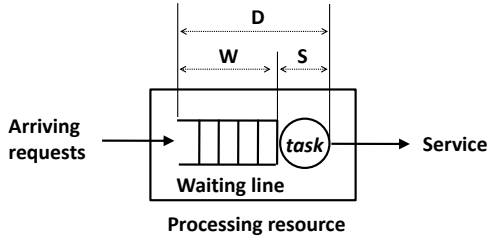

Fig 3. Single queue parameters in the context of the classical QT: mean waiting time (W), mean service time (S), mean request-response delay time (D).

QT can be seen as a branch of probability theory applied to different fields, e.g., communication networks, computer systems, and so forth. QT tries to estimate parameters like e.g., the mean system response time (waiting time in the queue plus service times), distribution of the number of customers in the queue, distribution of the number of customers in the system, and so forth. This analysis is mainly studied in stochastic scenarios (Fig. 3).

Queuing systems may not only be different in distributions of the inter-arrival and service times, but also in the number of servers, size of the waiting queues (infinite or finite), service discipline, and so on.

To analyze multi-tier web applications, one can represent web applications as a network of queuing systems. One basic classification of queuing networks is the distinction between open and closed queuing networks.

In an open network, new customers may arrive from outside of the system (coming from a conceptually infinite population) and, later on, leave the system. In a closed queuing network, the number of customers is fixed, and no customers enter or leave the system. Examples of queuing models that could be used to capture and analyze the behavior of cloud systems and their applications are M/M/1, M/G/1, M/M/m, M/G/m/K, M/M/c, or Erlang formulas (Fig. 4; see [4, 5] for references).

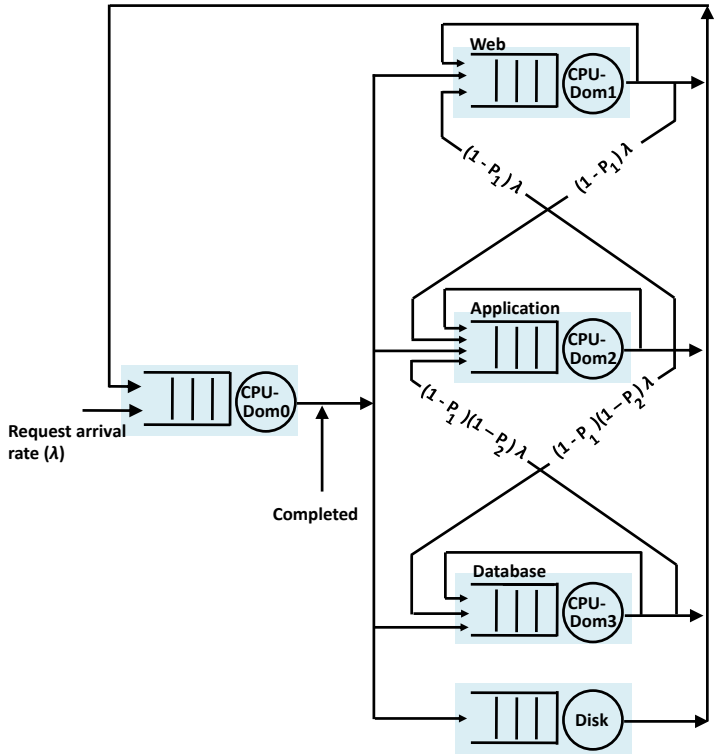

Fig 4. Example of a closed-queueing system based on M/G/1queuing modeling for virtualized three-tier applications, as shown in Fig. 1 (Adapted from [2]).

3.2 Control theory models

Control theory (CT) is another popular technique [4, 5]. It provides a systematic approach for designing closed-loop systems that are one of the basic type of control system, which uses feedback signals to control itself. They are designed to automatically achieve and maintain the desired output condition by comparing it with the actual condition. Such systems are designed to be stable by trying to avoid wild oscillations, accurate by achieving the desired outputs (e.g., response time), and settle quickly to steady state values (e.g., to adjust the workload dynamics) [9] (Fig. 5).

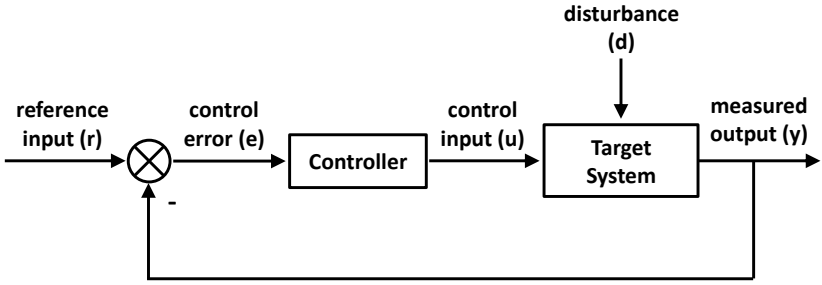

Fig 5. Standard feedback control loop (Adapted from [1]).

The target system provides a set of performance variables referred to as measured outputs or simply outputs.

Sensors monitor the outputs of the target system, and actuators can adjust control inputs, or simply inputs, to change the system behavior.

The feedback controller is the decision-making unit of the control system. The main objective of the controller is to maintain the outputs of the system sufficiently close to the desired values by adjusting the inputs under disturbances. This desired value is translated by the control system to the set point signals, which gives the option for the control system designer to specify the goals or values of the outputs that have to be maintained at runtime.

The feedback control system is a reactive decision making mechanism, because it waits until a disturbance affects the outputs of the system to make the necessary decisions.

Another type of control systems is feed-forward control system (considered as a proactive control mechanism).

Also, it is used a combination of the two previous types, i.e., feedback and feed-forward control system (which addresses the limitations of both schemes) [10].

Recently, CT has been used in the analysis of many aspects of cloud computing environments [4, 5] (Fig. 6).

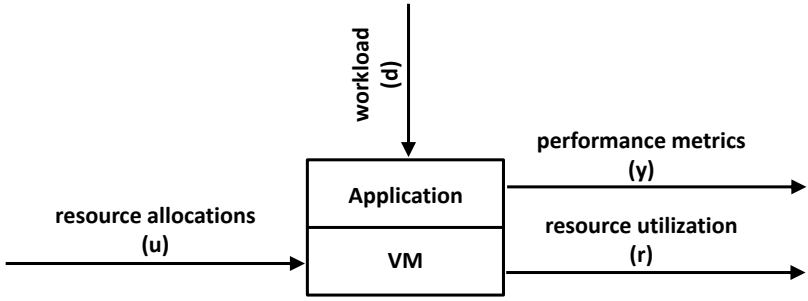

Fig 6. Example of the application of control theory to automated resource and service level management in shared virtualized infrastructures with three nodes hosting multiple multi-tier applications (Adapted from [1]).

4. Modular Performance Analysis with RTC

In addition to the analytical approaches described in the previous section, in this paper, we analyze the features provided by RTC.

The central idea of "Modular Performance Analysis with RTC" (MPA-RTC) [11] is to build an abstract performance model of a system that bundles all information needed for performance analysis with RTC.

The abstract performance model unifies essential information about the environment, about the available computation and communication resources, about the application tasks (or dedicated HW/SW components), as well as about the system architecture itself.

For performance analysis by using MPA-RTC, a real system (e.g., a multi-tier web application) can be decomposed into abstract performance analysis components (i.e., RTC components) whose behavior can be deterministic or non-deterministic. For instance, Fig. 2 shows that the system can be decomposed into five concatenated queuing subsystems, which can be analytically modeled as RTC components with non-deterministic behavior.

4.1 Deterministic analysis

RTC is a formal method developed in embedded systems domain [12-14]. In [15], RTC is compared with the analytical approaches commonly used for the performance evaluation of network interfaces. A case study of the applicability of RTC in the context of performance evaluation of network interfaces is presented in [16].

Basically, the RTC framework primary consists of a task model, resource model, and calculus (i.e., Real-Time Calculus) that allows reasoning about event streams and their processing.

In this work, we consider the problem of the evaluation of cloud computing environments. In the mentioned framework, the input event stream might be composed by a finite number of different event types, e.g., HTTP requests issued by clients, service requests issued the web server to the application server, or service requests issued the application server to the database server.

On the other hand, the processing resources that we model are the virtual machines in which the application tiers are deployed, and the task model, considered in this work, consists of software servers.

In RTC, the resource model captures the information about the available processing capacity of different hardwares involved in the processing of requests, and the possible mappings of processing functions to these resources (e.g., mapping application tiers to virtual machines).

The analytical framework also considers characteristics of the event stream entering the system (e.g., clients requests in Fig. 2), which are specified by using their arrival curves.

Thus, given the infrastructure of a data center, the calculus associated with the RTC-based framework can be used to analytically determine properties such as the maximum delay (latency) experienced by an event stream, and take into consideration the underlying scheduling disciplines at the different processing resources.

In this paper, we estimate the impact of the data center resource pool parameters (e.g., servers speed), and stochastic behavior of both web applications workload and application tiers processing time on the application response time by analytical methods.

Other specific VMs management issues are also analyzed and discussed (Section 5).

In RTC, the basic model is characterized by a processing resource that receives incoming requests and executes them using the available resource (processing or communication) capacity. To this end, some non-decreasing functions of resource provisioning are introduced.

Definition 1 (Arrival and Service Function). An event stream can be described by an arrival function R, where R(t) denotes the number of events that have arrived in the interval [0, t).

A computing or communication resource can be described by a service function C, where C(t) denotes the number of events that could have been served in the interval [0, t).

Definition 2 (Arrival and Service Curves). The upper and lower arrival curves, $\alpha^u(\Delta),\ \alpha^l(\Delta)\ \in \mathbb{R}\geq 0$ of an arrival function R(t) satisfy the following inequality:

$$\alpha^l(t-s) \leq R(t) - R(s) \leq \alpha^u(t-s), \forall\, s,t : 0 \leq s \leq t$$

The upper and lower service curves,

$$\beta^u(\Delta),\ \beta^l(\Delta) \in \mathbb{R}^{\geq 0}$$

of a service function C(t) satisfy

$$\beta^l(t-s) \leq C(t) - C(s) \leq \beta^u(t-s)\ \forall\, s,t : 0 \leq s \leq t$$

As described in [12], α_f^u and β_r^l bounding-functions can be defined using a piecewise linear approximation (Fig. 7).

For example, given a trace representing the processing capabilities of a VM running an application tier, two-slopes piecewise linear functions (i.e., LR functions, Section 4.2) can be used for describing a lower bound of the processing service at VMs over any time interval of length Δ (Fig. 7a).

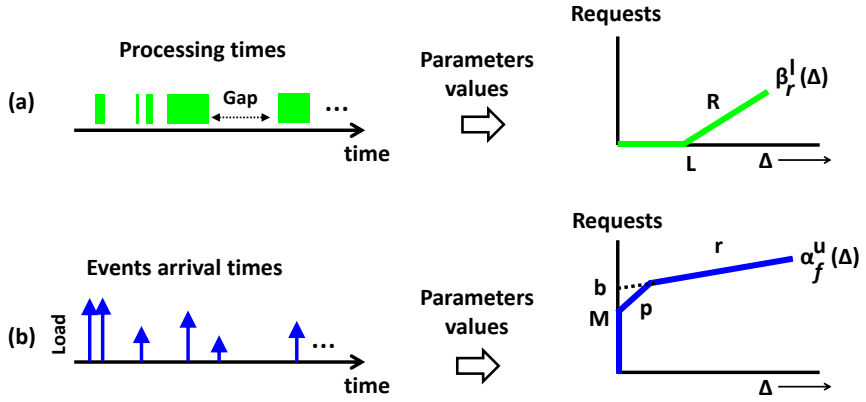

Fig 7. Obtaining the parameters values required for constructing the straight line segments of the upper and lower bounding-curves by using a software server trace and an arrival trace, respectively. In (a), the slope L represents the latency (i.e., longest gap in the trace), and the slope R can be interpreted as the average (long-term) processing rate. In (b), M represents the maximum possible load (measured e.g. in time units) on a resource for processing one token (i.e., one request); the slope p of the middle segment can be interpreted as the (load on a resource due to short-term) peak/burst rate, the slope r as the (load on a resource due to the) long-term request arrival rate, and the value b, as the burst tolerance of events stream.

Similarly, arrival curves defined by using piecewise linear segments with three pieces (three slopes) can be used for expressing an upper bound of the number of events that may arrive over any time interval of length Δ. This allows us to model an arrival curve in the form of a T-SPEC specification (p, r, M, b). For instance, a token bucket is used to specify event streams (i.e., traffic), which is widely used in the area of communication networks [17] (Fig. 7b).

Then, by using the RTC-based analytical framework, we can compute the maximum delay experienced by an event stream passing through a single resource processing the flow (e.g., a single application tier), and passing through a multiple processing resources (e.g., the entire application tiers).

When α_f^l and α_f^u describe the arrival curves of an event stream f, and if, β_r^l and β_r^u, describe the processing capability of r in terms of the same units, then, the maximum delay suffered by the event stream f at the resource r can be given by the following inequality:

$$\text{delay} \leq \sup_{t \geq 0}\{\inf\{\tau \geq 0 : \alpha_f^u(t) \leq \beta_r^l(t + \tau)\}\}$$

A physical interpretation of this inequality can be given as follows: the maximum delay experienced by an event stream (e.g., client data access requests in multi-tier cloud web applications) waiting to be served by r (e.g., a web, application, or database server) can be bounded by the maximum horizontal distance between the bounding-functions α_f^u and β_r^l (Fig. 8).

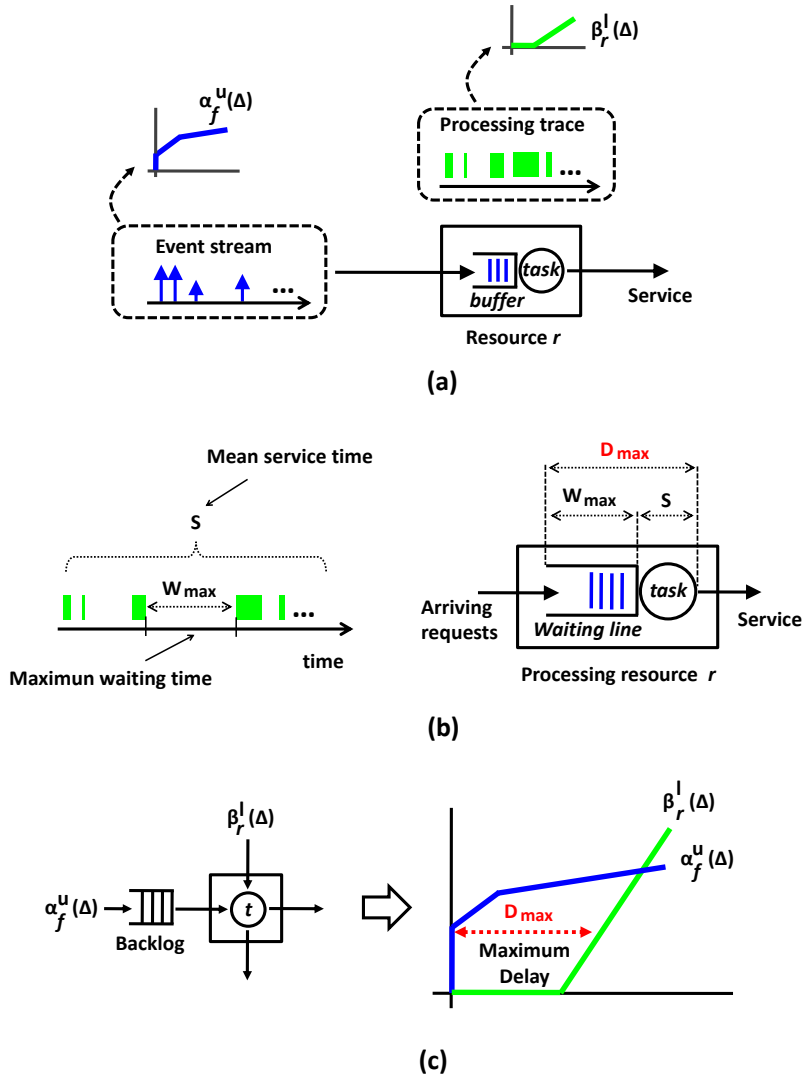

Fig 8. (a) Deriving the α_f^u and β_r^l bounding-functions of the processing resource r. (b) RTC model parameters and our metric of interest (D_{max}). (c) Modeling the resource r and obtaining its maximum request-response delay time (D_{max}) by using RTC.

According to [12], if the event stream passes through multiple resources, such as a tandem of software servers involved in processing incoming event stream using a FIFO discipline (Fig. 2), which have their input lower service curves equal to β_1^l, β_2^l, β_3^l, ..., β_n^l, then, an accumulated lower service curve β^l for serving this event

stream can be computed through an iterated convolution (as defined in the network calculus domain [18] (Fig. 9):

$$\beta^l = (((\beta_1^l \otimes \beta_2^l) \otimes \beta_3^l) \otimes \dots) \otimes \beta_n^l \qquad (1)$$

Thus, the maximum delay experienced by this stream can be given by

$$delay \leq sup_{t \geq 0}\{\inf\{\tau \geq 0 : \alpha_f^u(t) \leq \beta^l(t + \tau)\}\}$$

In the analytical framework, depending on the context, in which these bounding-functions are used, the delay can be computed in terms of different time units, e.g., cycles, seconds, etc.

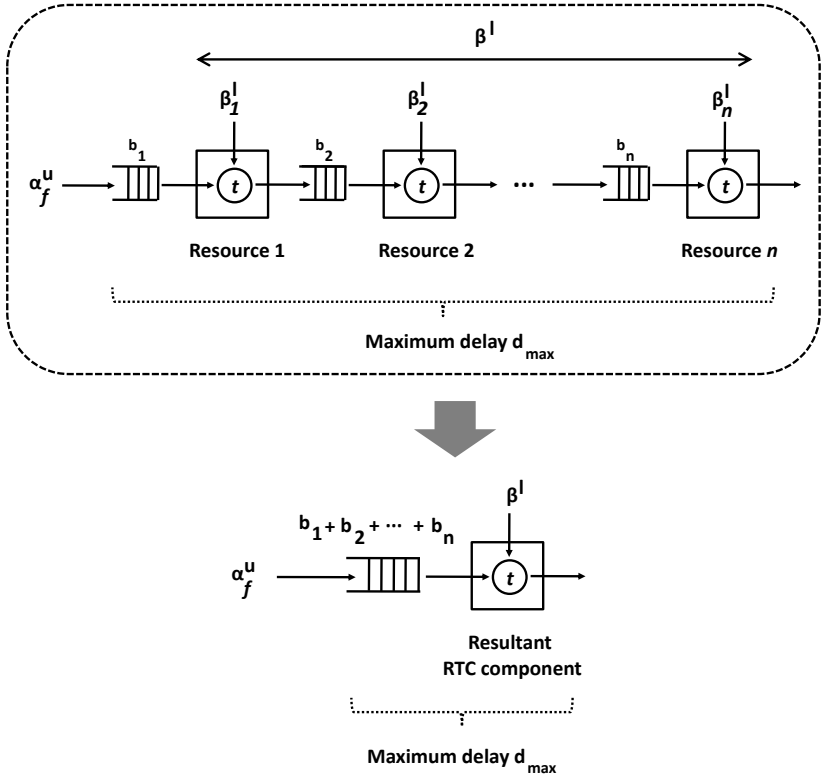

Fig 9. A tandem of processing resources (as in Fig. 2) modeled each one of them by means of an RTC component (upper part), and representation of a resultant RTC component for the system (bottom), which uses as input the β^l accumulated lower service curve computed for the tandem of processing resources using the equation (1).

In general RTC-based analysis, components are specified as transformers of input arrival and service curves into output arrival and service curves through a set of equations (Fig. 10; see [11]). Thus, RTC-based analytical approaches are compositional in the sense that they use local parameters about processing resources

(such as the arrival rate of event stream, long-term average service rate, longest gap in a trace of processing availability), which can be determined without taking into account any interference with other resources.

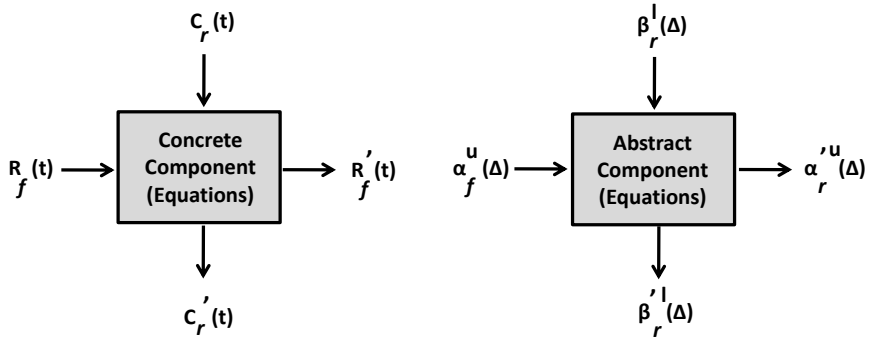

Fig 10. Transforming input functions into output functions. (a) Specific arrival and service functions, $R_f(t)$ and $C_r(t)$, enter into a concrete processing resource and are transformed into the $R'_f(\Delta)$ and $C'_r(\Delta)$ output-functions. (b) Abstract arrival and service curves, α_f^u and β_r^l, enter into an abstract component (RTC component), and are transformed into the $\alpha_f'^u$ and $\beta_r'^l$ output service curves.

Hence, by using this local information, we can predict how global parameters (such as end-to-end latency) will behave in a given system that combines the analytical models (RTC components) of these individual processing resources. This approach shows how to reduce the complexity of the system by combining the analysis of single components.

4.2 Stochastic analysis

The analytical framework described in the previous sections allows us to obtain hard real-time guarantees on delays and backlog. To this end, a finite trace of an event stream and a sliding window approach are applied to derive the arrival and service curves [14].

Contrary to the classical MPA-RTC, the RTC-based probabilistic analysis presented in [16] provides soft real-time guarantees, i.e., guarantees on delays and backlogs that are valid up to a certain level of confidence, as opposed to the hard guarantees commonly derived by formal methods.

In [16], the α_f^u and β_r^l bounding-curves are not deduced by sliding a window of length Δ over the trace and recording the minimum and maximum number of events lying within the window. Stochastic models for the service and arrival curves are considered. These models are stochastic in the sense that they consider uncertainties in the estimation of the parameters required for constructing the pieces of line for α_f^u and β_r^l.

This approach is most suitable in the context of our work (Fig. 2). For example, processing tasks at presentation, application and data layers could be modeled as latency-rate servers (LR servers). In such a case, the β_r^l lower service curve can be represented as a $\beta_{L,R}(t)$ latency-rate function (LR function). In the network calculus domain, it is defined as [18]:

$$\beta_{L,R}(t) = \begin{cases} R(t-L), & if\ t > L \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

for some $L \geq 0$ ("latency") and $R \geq 0$ ("rate").

4.3 RTC model calibration

In general, an RTC model for multi-tier cloud web applications can be calibrated (parameterized) using different alternatives. For example, the value of the input parameters of analytical model, which are needed for constructing the pieces of line of the arrival and service curves (mathematical functions), can be obtained from direct measurement on real systems [19], simulation results [20] e.g., by using trace/model-based simulations, or by synthetic models [21].

It should be noted that deriving the parameters for constructing the $\beta_{r_i}^l$ lower service curve of a concrete system component with non-deterministic behavior (e.g., a web, application or database server) from simulations or real traces may give the case where the following assumption holds (see [16]).

$$\exists\, i, \Delta :\ \beta_{r_i}^l(\Delta) < \beta_{\{r_i, reality\}}^l(\Delta) \quad (3)$$

where $i \in (1, 2, 3, \ldots)$, and $\beta_{r_i}^l$ is a resultant lower service curve derived from a set of lower service curves.

The elements of this set are a family of service curves of the component obtained by using alternatives for model calibration described above. Notice that the value of the L and R are parameters of an aggregated (resultant) bounding-curve.

Let us say that $\beta_{r_i}^l$, can be computed using aggregation functions like "AVERAGE", "MINIMUM", or "MAXIMUM", given a list of parameter values (Fig. 11; see [16] for details).

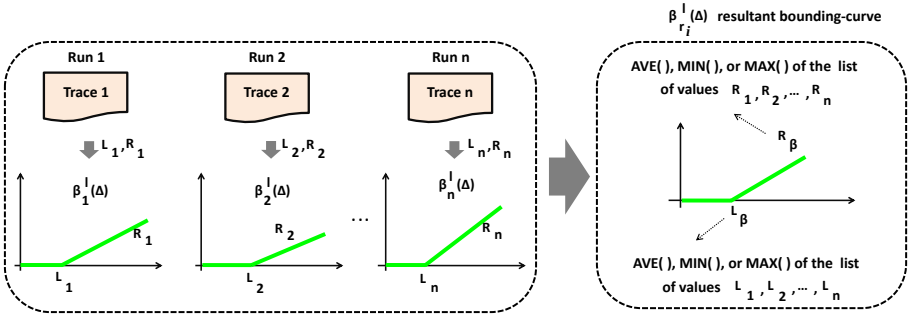

Fig 11. Family of service curves corresponding to a system component with non-deterministic behavior (left part), and procedure for obtaining its resultant bounding-curve (right part).

Lastly, $\beta^{l}_{\{r_i,\text{reality}\}}(\Delta)$ in (3) is an unknown lower bounding-curve of the SUT for the stochastic component being considered.

Indeed, note that as (3) may occasionally hold, the analytically computed results are invalid. For this reason, in [16], statistical methods are used in order to demonstrate that the values of the L and R parameters of $\beta^{l}_{r_i}$ have an adequate level of predictability, and, hence, results are valid up to certain level of confidence.

5. Discussion

In this work, we are interested in the capabilities of each analytical approach for modeling the following aspects of cloud computing: multi-tier cloud web applications, response time guarantees (hard and/or soft), workload models, task processing models, VM provisioning, VMs performance interference, autonomic resource management, server consolidation, and cloud scaling strategies (horizontal and/or vertical).

Table 1 summarizes all these issues. Moreover, to support our comparison, references to analytical studies based on queuing theory (QT) and control theory (CT) are given.

Multi-tier cloud Web application. Several authors have addressed the issue of modeling multi-tier cloud Web application by analytical approach such as QT and CT with varying degree of success (see the review in [4]).

Table 1. Comparison of analytical approaches

Modeling capabilities	MPA-RTC	Queuing Theory (QT)	Control Theory (CT)
Multi-tier cloud Web application	Yes	Yes	Yes
Hard/Soft response time guarantees	Both	No	Soft guarantees

Workload models	Real and/or synthetic	Synthetic	Real or synthetic
Task processing models	Real and/or synthetic	Synthetic	Real or synthetic
VM provisioning	Yes	Yes	Yes
VMs performance interference effect	Yes	Yes	Yes
Autonomic resource management	Yes	Yes	Yes
Server consolidation	Yes	Yes	Yes
Horizontal/Vertical scaling	Both	Both	Both

Based on the ideas exposed in Section 4, we consider that MPA-RTC is also a suitable approach for modeling multi-tier cloud Web applications. Nevertheless, it should be noted that there are differences in the scope of each approach.

RTC belongs to the class of so-called deterministic queuing theories. It is deterministic in the sense that hard upper and lower bounds of the performance metrics (such as latency) can be always found.

This distinguishes it from the class of non-deterministic analysis techniques such as QT and CT for which this guarantee cannot be provided (in general).

Deterministic queuing theories such as MPA-RTC are well-suited for studying hard performance bounds since they ensure that all requirements are met by the system during all the time.

In contrast, RTC does not allow us to model the average response time of web applications. For this purpose, stochastic approaches such as QT are better suited.

Specifically, the RTC-based probabilistic analysis described in Section 4.2 might be useful for obtaining soft real-time guarantees in the context of cloud computing environments.

Response time guarantees. In principle, RTC models allow performance analysts to derive hard and soft response time guarantees in the context of cloud computing systems.

In particular, the end-to-end latency quantity in RTC allows us to evaluate worst case scenario, i.e., the maximum delay experienced by an event stream at a given individual software server (or at a tandem of them).

On the contrary to RTC, the mean delay quantity used in QT-based analysis does not allow to obtain QoS guarantees such as response time.

Regarding CT, this methodology provides only soft performance guarantees. It is to be noted that due to inherent sources of instability in control systems (e.g., latency to get the stationary values of observable variables after applying a control action) under unpredictable disturbances, the deadlines of some tasks could be violated; hence, hard real-time guarantees cannot be obtained at all.

Nevertheless, we consider that an RTC-based stochastic analysis (Section 4.2) would be more suitable from the perspective of performance evaluation of cloud computing environments due to the dynamic nature of incoming requests and server-side processing (Fig. 2). Below we consider our workload and task processing models.

Workload models. The workload model can be analytically evaluated by using any of the following four alternatives:

(1) Real workload traces (data gathered from a production platform);
(2) Naive synthetic workload models that use probability distributions to generate workload data (based on little or no knowledge of real trace characteristics);
(3) Realistic synthetic workload models in which the model and its parameters have been abstracted through careful analysis of real workloads data from production servers;
(4) Combinations of the previous alternatives (in particular, MPA-RTC allows this approach).

Both real and realistic synthetic workloads have been considered in studies based on CT (see [5]). On the other hand, most of QT-based studies use synthetic workload models based on Poisson process [5].

In [22], the authors show that one can reasonably accept that this assumption is valid.

With respect to RTC, it supports a flexible workload model. For example, workload can be expressed by any type of service units per unit time arriving at processing resources (e.g., instructions/s, requests/s, transactions/s, etc.). It has a highly flexible workload granularity level. Besides, we can construct arrival curves from realistic event arrival traces or synthetic traffic models (constant, bursty, Poisson, etc). Also, different workload sizes (fixed or variable) can be modeled.

Task processing models. In [5], a variety of experimental platforms for modeling the processing of tasks in CT-based studies (e.g., real testbeds, simulators) are reviewed.

On the contrary, most QT-based studies only consider synthetic task processing models (e.g., processing times which follow exponential distribution [23]).

Using MPA-RTC, software servers can be modeled by means of RTC components (LR servers). To calibrate these components in isolation, the processing characteristics of software servers in terms of computational work performed by them (e.g., measured in requests/second) can be used.

VM provisioning. The process of provisioning VMs in IaaS clouds includes partial delays caused by queuing, provisioning decision, VMs instantiation and deployment.

In MPA-RTC, these delays can be modeled as non-processing intervals (variable latency periods) in a server trace in terms of processing availability (Fig. 12).

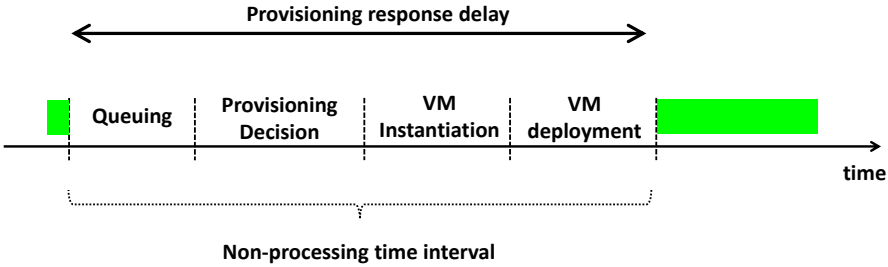

Fig 12. Modeling provisioning response delay: Non-processing intervals in a trace of execution time of software servers.

VM provisioning has been modeled analytically either by using QT [24] or CT [25].

VMs performance interference effect. In a virtualized system, performance interference is caused by sharing physical resources (mainly, I/O [26]) among VMs and virtual machine monitor scheduling (Fig. 13).

VM performance interference has been analytically modeled by using QT [27] and CT [28]. To model the performance degradation due to resource contention by using MPA-RTC, an extra logical performance component (i.e., a non-deterministic RTC component) can be added to the RTC model of the SUT.

Particularly, the service curve of this RTC component would allow us to model the non-deterministic access to shared resources in virtualized environments.

For performance analysis, this abstract component should be properly calibrated in order to achieve realistic results (Section 4.3).

Autonomic resource management. We consider that RTC can be a proper alternative for this purpose. Instead of an offline trace-based calibration approach, online methods could be employed.

To this end, all the desired parameters values of analytical model could be collected through physical infrastructure monitoring [19]. Then, collected data could be incorporated into an RTC-based autonomic control loop, aiming at achieving business objectives.

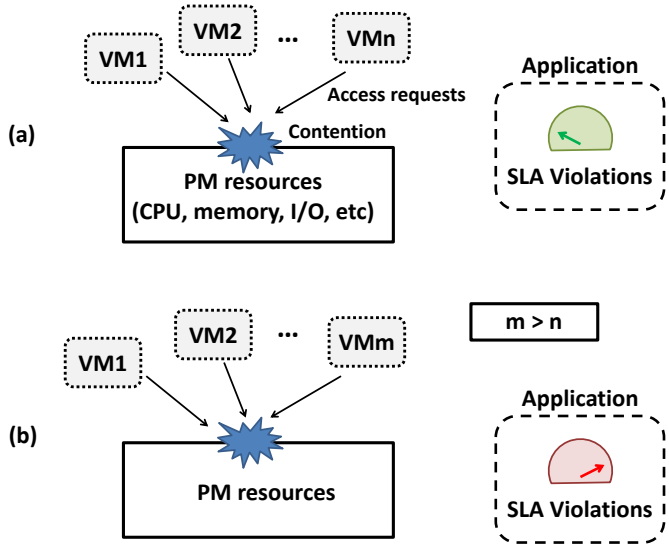

Fig 13. Imaginary examples of VMs performance interference effect due to resource contention on virtualized environments: (a) For a total of n virtual machines deployed, application performance in terms of SLA violations is acceptable. (b) For m > n, performance degrades ostensibly.

This way, cloud systems could dynamically adapt themselves to the changing environment, and, based on management strategies, control actions (e.g., live VMs migration) could be triggered (Fig. 14).

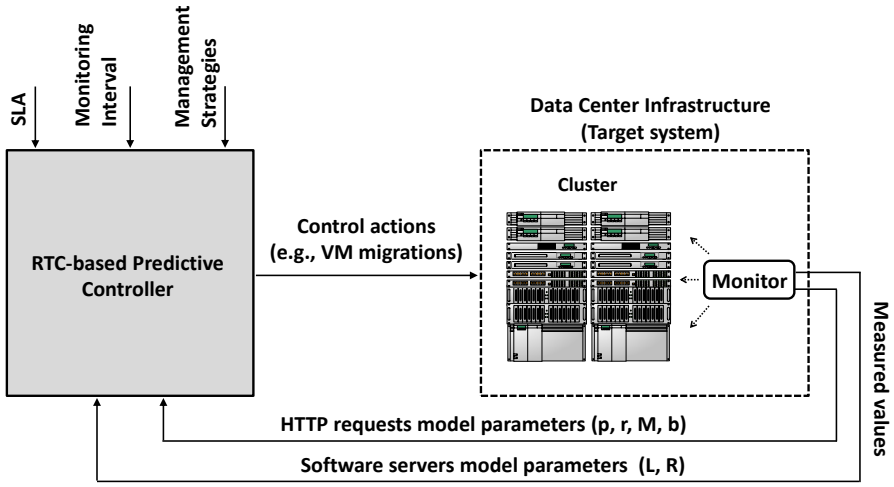

Fig 14. Cloud monitoring through online instrumentation for RTC-based autonomic resource management.

Various typical papers covering autonomic resource management by using QT and CT are surveyed in [3].

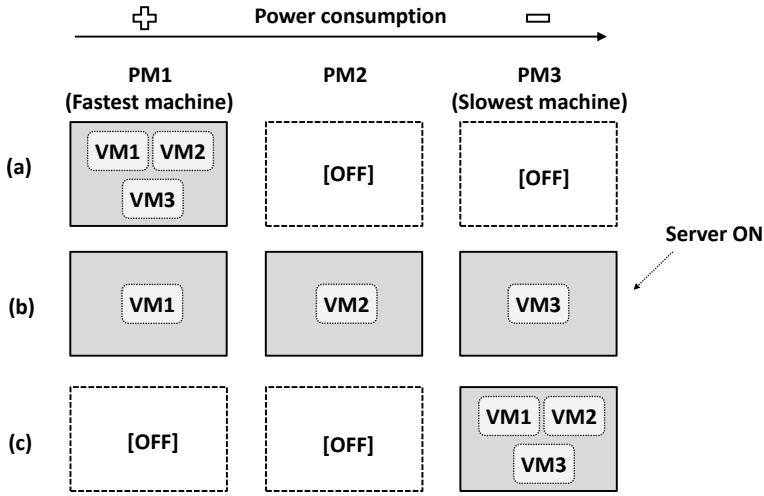

Fig 15. Imaginary examples of VMs deployment scenarios for our SUT (Fig. 2): (a) Speed-oriented server consolidation. (b) Non-consolidated scenario. (c) Energy-efficient server consolidation.

Server consolidation. The consolidation of servers is an energy-aware resource allocation technique for cloud computing systems.

In real scenarios, IaaS providers need to evaluate many VM combinations to find the optimal consolidation of VMs on the physical servers taking into account QoS (Fig. 15). We consider that the RTC-based interference model as well as autonomic resource management issues described above could be precisely incorporated into VM consolidation performance analysis.

In [29], CT is used to deal with the problem of achieving the best consolidation level that can be attained without violating application SLAs.

In [30], server consolidation is analyzed by using QT.

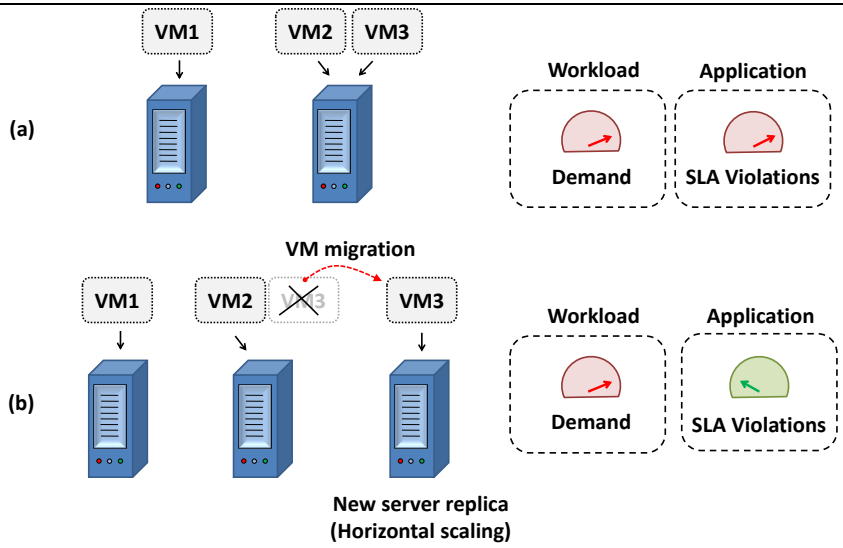

Fig 16. Effect of horizontal scaling on application performance for our SUT (Fig. 2): (a) In this imaginary example, for the baseline VMs deployment scenario considered, a high workload demand leads to a high number of SLA violations. (b) For the same baseline scenario, after adding a new server replica for migration purposes, the number of SLA violations is low.

Horizontal/Vertical scaling. Approaches to scaling cloud infrastructure to meet client workload requirements can be classified as vertical scaling type, e.g., adding larger and more powerful physical machines to accommodate the demand, and horizontal scaling type, e.g., adding new server replicas (i.e., PMs) and load balancers to distribute load among all available replicas (Fig. 16).

We would expect that using a higher speed server (vertical scaling) or adding a new server replica for VMs migration purposes (horizontal scaling) have to be reflected in the shape of the service curves (LR function) characterizing the task processing of the software servers deployed on the VMs being migrated.

For this reason, we consider that MPA-RTC allows us to model both vertical and horizontal scaling strategies. In [5], various examples are reviewed in which vertical scaling strategies are evaluated using QT. Several examples of application of control theory for the performance evaluation of both vertical and horizontal scaling can be found in [5]. In [31], horizontal scaling by using QT is evaluated.

6. Conclusion

In this paper, we discuss different approaches for modeling cloud-based systems. Based on the results of their comparison, we conclude that RTC is suitable framework for estimating statistical response time guarantees, which is an important

quality attribute for Web applications from the user point of view. In addition, other contemporary issues in cloud computing research could be analyzed by using MPA-RTC.

References

[1]. X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, P. Padala, and K. Shin, "What does control theory bring to systems research?," ACM SIGOPS Operating Systems Review, vol. 43, pp. 62-69, 2009.

[2]. K. RahimiZadeh, M. AnaLoui, P. Kabiri, and B. Javadi, "Performance modeling and analysis of virtualized multi-tier applications under dynamic workloads," Journal of Network and Computer Applications, 2015.

[3]. X.-Y. Wang, L.-H. Fan, X.-H. Jia, and W.-T. Huang, "A Survey of Virtualization-based Resource Management in Cloud Computing Environments," Journal of Convergence Information Technology, vol. 8, 2013.

[4]. D. Huang, B. He, and C. Miao, "A survey of resource management in multi-tier web applications," IEEE Communications Surveys & Tutorials, vol. 16, pp. 1574--1590, 2014.

[5]. T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments," Journal of Grid Computing, vol. 12, pp. 559-592, 2014.

[6]. N. Grozev and R. Buyya, "Performance modelling and simulation of three-tier applications in cloud and multi-cloud environments," The Computer Journal, vol. 58, pp. 1-22, 2015.

[7]. L. Kleinrock, Theory, Volume 1, Queueing Systems: Wiley-Interscience, 1975.

[8]. . Willig, "A short introduction to queueing theory," Technical University Berlin, Telecommunication Networks Group, vol. 21, 1999.

[9]. T. Abdelzaher, Y. Diao, J. L. Hellerstein, C. Lu, and X. Zhu, "Introduction to control theory and its application to computing systems," in Performance Modeling and Engineering: Springer, 2008, pp. 185-215.

[10]. T. Patikirikorala, A. Colman, J. Han, and L. Wang, "A systematic survey on the design of self-adaptive software systems using control engineering approaches," in Proceedings of the 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2012, pp. 33-42.

[11]. E. Wandeler, L. Thiele, M. Verhoef, and P. Lieverse, "System architecture evaluation using modular performance analysis: a case study," Int. J. Softw. Tools Technol. Transf., vol. 8, pp. 649-667, 2006.

[12]. S. Chakraborty, S. Künzli, L. Thiele, A. Herkersdorf, and P. Sagmeister, "Performance evaluation of network processor architectures: combining simulation with analytical estimation," Comput. Netw., vol. 41, pp. 641-665, 2003.

[13]. L. Thiele, S. Chakraborty, M. Gries, and S. Künzli, "A framework for evaluating design tradeoffs in packet processing architectures," in Proceedings of the 39th conference on Design automation New Orleans, Louisiana, USA: ACM, 2002.

[14]. S. Chakraborty, S. Künzli, and L. Thiele, "A General Framework for Analysing System Properties in Platform-Based Embedded System Designs," in Proceedings of the conference on Design, Automation and Test in Europe - Volume 1: IEEE Computer Society, 2003.

[15]. G. R. Garay, J. Ortega, and V. Alarcón-Aquino, "Comparing Real-Time Calculus with the Existing Analytical Approaches for the Performance Evaluation of Network Interfaces," in Proceedings of the 21st IEEE International Conference on Electronics, Communications and Computers (CONIELECOMP 2011) Cholula, Puebla, México: IEEE, 2011, pp. 119-124.

[16]. G. R. Garay, J. Ortega, A. F. Díaz, L. Corrales, and V. Alarcón-Aquino, "System performance evaluation by combining RTC and VHDL simulation: A case study on NICs," Journal of Systems Architecture, vol. 59, pp. 1277-1298, 2013.

[17]. S. Shenker and J. Wroclawski, "General characterization parameters for integrated service network elements. RFC 2215.," IETF, 1997.

[18]. J.-Y. L. Boudec and P. Thiran, Network calculus: a theory of deterministic queuing systems for the internet: Springer-Verlag New York, Inc., 2001.

[19]. G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Cloud monitoring: A survey," Computer Networks, vol. 57, pp. 2093-2115, 2013.

[20]. A. Ahmed and A. S. Sabyasachi, "Cloud computing simulators: A detailed survey and future direction," in Advance Computing Conference (IACC), 2014 IEEE International, 2014, pp. 866-872.

[21]. A. Bahga and V. K. Madisetti, "Performance evaluation approach for multi-tier cloud applications," Journal of Software Engineering and Applications, vol. 6, p. 74, 2013.

[22]. D. Villela, P. Pradhan, and D. Rubenstein, "Provisioning servers in the application tier for e-commerce systems," ACM Transactions on Internet Technology (TOIT), vol. 7, p. 7, 2007.

[23]. Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," The Journal of Supercomputing, vol. 60, pp. 268-280, 2012.

[24]. H. Khazaei, J. Misic, and V. B. Misic, "A fine-grained performance model of cloud computing centers," Parallel and Distributed Systems, IEEE Transactions on, vol. 24, pp. 2138-2147, 2013.

[25]. P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, and K. Salem, "Adaptive control of virtualized resources in utility computing environments," in ACM SIGOPS Operating Systems Review, 2007, pp. 289-302.

[26]. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," Communications of the ACM, vol. 53, pp. 50-58, 2010.

[27]. F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions," Proceedings of the IEEE, vol. 102, pp. 11-31, 2014.

[28]. R. Nathuji, A. Kansal, and A. Ghaffarkhah, "Q-clouds: managing performance interference effects for qos-aware clouds," in Proceedings of the 5th European conference on Computer systems, 2010, pp. 237-250.

[29]. C. Anglano, M. Canonico, and M. Guazzone, "FC2Q: exploiting fuzzy control in server consolidation for cloud applications with SLA constraints," Concurrency and Computation: Practice and Experience, pp. n/a-n/a, 2014.

[30]. Z. Luo and Z. Qian, "Burstiness-aware Server Consolidation via Queuing Theory Approach in a Computing Cloud," in Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on, 2013, pp. 332-341.

[31]. A. Gandhi, P. Dube, A. Karve, A. Kochut, and L. Zhang, "Modeling the Impact of Workload on Cloud Resource Scaling," in Computer Architecture and High Performance Computing (SBAC-PAD), 2014 IEEE 26th International Symposium on, 2014, pp. 310-317.

[32]. Godofredo R. Garay, Andrei Tchernykh, Alexander Yu. Drozdov. An Approach for the Performance Evaluation of Multi-Tier Cloud Applications. II International Conference «Engineering & Telecommunication –En&T 2015», November 18–19, IEEE Computer Society, 2015

Сравнительный анализ методов оценки производительности многоуровневых облачных приложений[2]

[1]Г.Р. Гарай <godofredo.garay@reduc.edu.cu>
[2]А. Черных <chernykh@cicese.mx>
[3]А.Ю. Дроздов < alexander.y.drozdov@gmail.com>
[1]Университет Камагуэй, Куба
[2]Исследовательский центр CICESE, Мексика
[3]Московский физико-технический институт (государственный университет), 141700, Россия, Московская область, г. Долгопрудный, Институтский пер., 9

Аннотация. Аналитическое моделирование используется на ранних этапах проектирования аппаратуры, когда достаточно быстро должны быть рассмотрены многочисленные варианты. Оно позволяет провести оценку производительности предлагаемых систем без сложного и затратного моделирования. Наиболее популярные аналитические подходы к оценке производительности облачных вычислений включают модели теории очередей и теории контроля.Исчисление реального времени (Real-TimeCalculus- RTC) это аналитическая техника высокого уровня, первоначально разработанная для систем обработки потоков в режиме жесткого реального времени и часто используемая для нахождения баланса параметров в архитектурах обработки потока пакетов. Центральная идея модулярного анализа производительности с RTC (MPA-RTC) заключается в построении абстрактной модели производительности, которая связывает всю информацию, необходимую для анализа с исчислением реального времени.В этой статье мы рассматриваем оценку эффективности многоуровневых приложений для облачных вычислений, и сравниваем RTC с двумя классическими аналитическими подходами, такими как модели теории очередей и теории контроля.Мы сосредотачиваемся на возможностях этих альтернатив для оценки ключевого параметра качества обслуживания - времени ответа приложений.Кроме того, мы обсуждаем возможности каждого аналитического подхода для моделирования других аспектов среды облачных вычислений, таких как модели рабочей нагрузки, модели обработки задач, выделение ресурсов для виртуальных машин (VM), помех производительности виртуальных машин, автономное управление ресурсами, консолидацию серверов, а также стратегии масштабирования облачных вычислений (по горизонтали и / или по вертикали).

[2] Работы выполнены при финансовой поддержке Минобрнауки России (Соглашение № 02.G25.31.0061 12/02/2013).

Ключевые слова: исчисление реального времени, теория очередей, теория управления, QoS, облачные вычисления

DOI: 10.15514/ISPRAS-2015-27(6)-14

Для цитирования: Гарай Г.Р., Черных А., Дроздов А.Ю. Сравнительный анализ методов оценки производительности многоуровневых облачных приложений. Труды ИСП РАН, том 27, вып. 6, 2015 г., стр. 199-224. DOI: 10.15514/ISPRAS-2015-27(6)-14.

Список литературы

[1]. X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, P. Padala, and K. Shin, "What does control theory bring to systems research?," ACM SIGOPS Operating Systems Review, vol. 43, pp. 62-69, 2009.

[2]. K. RahimiZadeh, M. AnaLoui, P. Kabiri, and B. Javadi, "Performance modeling and analysis of virtualized multi-tier applications under dynamic workloads," Journal of Network and Computer Applications, 2015.

[3]. X.-Y. Wang, L.-H. Fan, X.-H. Jia, and W.-T. Huang, "A Survey of Virtualization-based Resource Management in Cloud Computing Environments," Journal of Convergence Information Technology, vol. 8, 2013.

[4]. D. Huang, B. He, and C. Miao, "A survey of resource management in multi-tier web applications," IEEE Communications Surveys & Tutorials, vol. 16, pp. 1574--1590, 2014.

[5]. T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments," Journal of Grid Computing, vol. 12, pp. 559-592, 2014.

[6]. N. Grozev and R. Buyya, "Performance modelling and simulation of three-tier applications in cloud and multi-cloud environments," The Computer Journal, vol. 58, pp. 1-22, 2015.

[7]. L. Kleinrock, Theory, Volume 1, Queueing Systems: Wiley-Interscience, 1975.

[8]. A. Willig, "A short introduction to queueing theory," Technical University Berlin, Telecommunication Networks Group, vol. 21, 1999.

[9]. T. Abdelzaher, Y. Diao, J. L. Hellerstein, C. Lu, and X. Zhu, "Introduction to control theory and its application to computing systems," in Performance Modeling and Engineering: Springer, 2008, pp. 185-215.

[10]. T. Patikirikorala, A. Colman, J. Han, and L. Wang, "A systematic survey on the design of self-adaptive software systems using control engineering approaches," in Proceedings of the 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2012, pp. 33-42.

[11]. E. Wandeler, L. Thiele, M. Verhoef, and P. Lieverse, "System architecture evaluation using modular performance analysis: a case study," Int. J. Softw. Tools Technol. Transf., vol. 8, pp. 649-667, 2006.

[12]. S. Chakraborty, S. Künzli, L. Thiele, A. Herkersdorf, and P. Sagmeister, "Performance evaluation of network processor architectures: combining simulation with analytical estimation," Comput. Netw., vol. 41, pp. 641-665, 2003.

[13]. L. Thiele, S. Chakraborty, M. Gries, and S. Künzli, "A framework for evaluating design tradeoffs in packet processing architectures," in Proceedings of the 39th conference on Design automation New Orleans, Louisiana, USA: ACM, 2002.

[14]. S. Chakraborty, S. Künzli, and L. Thiele, "A General Framework for Analysing System Properties in Platform-Based Embedded System Designs," in Proceedings of the conference on Design, Automation and Test in Europe - Volume 1: IEEE Computer Society, 2003.

[15]. G. R. Garay, J. Ortega, and V. Alarcón-Aquino, "Comparing Real-Time Calculus with the Existing Analytical Approaches for the Performance Evaluation of Network Interfaces," in Proceedings of the 21st IEEE International Conference on Electronics, Communications and Computers (CONIELECOMP 2011) Cholula, Puebla, México: IEEE, 2011, pp. 119-124.

[16]. G. R. Garay, J. Ortega, A. F. Díaz, L. Corrales, and V. Alarcón-Aquino, "System performance evaluation by combining RTC and VHDL simulation: A case study on NICs," Journal of Systems Architecture, vol. 59, pp. 1277-1298, 2013.

[17]. S. Shenker and J. Wroclawski, "General characterization parameters for integrated service network elements. RFC 2215.," IETF, 1997.

[18]. J.-Y. L. Boudec and P. Thiran, Network calculus: a theory of deterministic queuing systems for the internet: Springer-Verlag New York, Inc., 2001.

[19]. G. Aceto, A. Botta, W. De Donato, and A. Pescapè, "Cloud monitoring: A survey," Computer Networks, vol. 57, pp. 2093-2115, 2013.

[20]. A. Ahmed and A. S. Sabyasachi, "Cloud computing simulators: A detailed survey and future direction," in Advance Computing Conference (IACC), 2014 IEEE International, 2014, pp. 866-872.

[21]. A. Bahga and V. K. Madisetti, "Performance evaluation approach for multi-tier cloud applications," Journal of Software Engineering and Applications, vol. 6, p. 74, 2013.

[22]. D. Villela, P. Pradhan, and D. Rubenstein, "Provisioning servers in the application tier for e-commerce systems," ACM Transactions on Internet Technology (TOIT), vol. 7, p. 7, 2007.

[23]. Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," The Journal of Supercomputing, vol. 60, pp. 268-280, 2012.

[24]. H. Khazaei, J. Misic, and V. B. Misic, "A fine-grained performance model of cloud computing centers," Parallel and Distributed Systems, IEEE Transactions on, vol. 24, pp. 2138-2147, 2013.

[25]. P. Padala, K. G. Shin, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, and K. Salem, "Adaptive control of virtualized resources in utility computing environments," in ACM SIGOPS Operating Systems Review, 2007, pp. 289-302.

[26]. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," Communications of the ACM, vol. 53, pp. 50-58, 2010.

[27]. F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions," Proceedings of the IEEE, vol. 102, pp. 11-31, 2014.

[28]. R. Nathuji, A. Kansal, and A. Ghaffarkhah, "Q-clouds: managing performance interference effects for qos-aware clouds," in Proceedings of the 5th European conference on Computer systems, 2010, pp. 237-250.

[29]. C. Anglano, M. Canonico, and M. Guazzone, "FC2Q: exploiting fuzzy control in server consolidation for cloud applications with SLA constraints," Concurrency and Computation: Practice and Experience, pp. n/a-n/a, 2014.

[30]. Z. Luo and Z. Qian, "Burstiness-aware Server Consolidation via Queuing Theory Approach in a Computing Cloud," in Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on, 2013, pp. 332-341.

[31]. A. Gandhi, P. Dube, A. Karve, A. Kochut, and L. Zhang, "Modeling the Impact of Workload on Cloud Resource Scaling," in Computer Architecture and High Performance Computing (SBAC-PAD), 2014 IEEE 26th International Symposium on, 2014, pp. 310-317.

[32]. Godofredo R. Garay, Andrei Tchernykh, Alexander Yu. Drozdov. An Approach for the Performance Evaluation of Multi-Tier Cloud Applications. II International Conference «Engineering & Telecommunication –En&T 2015», November 18–19, IEEE Computer Society, 2015