An Approach for the Performance Evaluation of Multi-Tier Cloud Applications

Godofredo R. Garay
University of Camaguey, Camaguey, Cuba
godofredo.garay@reduc.edu.cu

Andrei Tchernykh
CICESE Research Center, Ensenada, México
chernykh@cicese.mx

Alexander Yu. Drozdov
Moscow Institute of Physics and Technology
Moscow, Russia, alexander.y.drozdov@gmail.com

***Abstract*—In this paper, we discuss a Real-Time Calculus-based approach for the performance evaluation of multi-tier cloud applications. We focus on its capabilities for estimating the Quality of Service parameters.**

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

I. 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 [1].

In this paper, we discuss a Real-Time Calculus-based approach for the performance evaluation of multi-tier cloud applications. We focus on the capabilities of the Real-Time Calculus for estimating the Quality of Service parameters. In particular, we focus on the capabilities of these alternatives that can be employed for estimating Web application response-time.

The paper is organized as follow. In Section II, we present the motivation of the work, and give some background information. We discuss the main features of Modular Performance Analysis with Real-Time Calculus in Section III. We conclude the paper in Section IV.

II. MOTIVATION

As a motivation example (Fig. 1), let us consider a system under test (SUT) consisting a three-tier web application [2, 3]. 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.

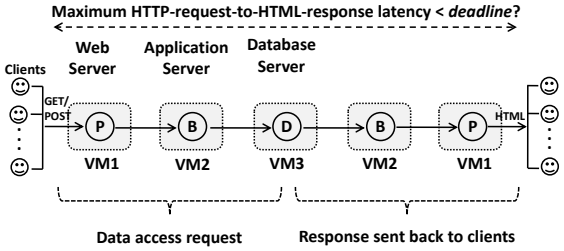

Fig. 1. Focus of attention: Predicting Web-application response-time in cloud computing platform, e.g., does maximum request-to-response latency of a client data access request will not exceed application deadline (with 95% confidence interval)?

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.

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?

III. PERFORMANCE ANALYSIS WITH RTC

Two of the most popular analytical approaches for the performance evaluation of cloud computing environments are Queuing Theory (QT) [4] and Control theory (CT) [5]. Surveys on the application of QT and CT in the context of the

10.1109/EnT.2015.19

performance evaluation of Multi-tier Cloud Applications can be found in [6, 7]. In addition to these analytical approaches, in this section, we analyze the features provided by RTC.

The central idea of "Modular 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.

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. 1 shows that the system can be decomposed into five concatenated queuing subsystems, which can be analytically modeled as RTC components with non-deterministic behavior.

A. Deterministic analysis

RTC is a formal method developed in embedded systems domain [8, 9]. In [10], 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 [11]. 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. 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 hardware 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, 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.

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, $\alpha^u(\Delta),\ \alpha^l(\Delta) \in \mathbb{R}$ of an arrival function R(t) satisfy the following inequality:

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

The upper and lower service curves,

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

of a service function C(t) satisfy

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

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

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 III-B) can be used for describing a lower bound of the processing service at VMs over any time interval of length Δ (see Fig. 2a).

Similarly, arrival curves defined e.g., 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 exactly model an arrival curve in the form of a T-SPEC specification (p, r, M, b), i.e., a token bucket is used to specify event streams (i.e., traffic), which is widely used in the area of communication networks [13] (again, see Fig. 2a).

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:

$$delay \le sup_{t\ge 0}\{\inf\{\tau \ge 0 : \alpha_f^u(t) \le \beta_r^l(t+\tau)\}\}$$

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. 2b).

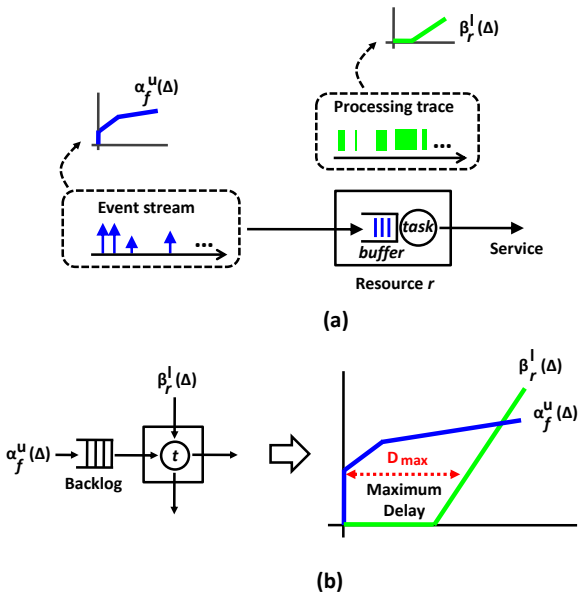

Fig. 2. (a) Deriving the α_f^u and β_r^l bounding-functions for a processing resource r. (b) Modeling the resource r and obtaining its maximum request-response delay time (D_{max}) by using RTC.

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, 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 [14]:

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

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.

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 (see [15]). 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.

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.

B. *Stochastic analysis*

The analytical framework described in the previous section 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 [16].

Contrary to the classical MPA-RTC, the RTC-based probabilistic analysis presented in [11] 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 [11], 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. 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 [14]:

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

for some L ≥ 0 (“latency”) and R ≥ 0 (“rate”).

C. *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 [17], simulation results [18] e.g., by using trace/model-based simulations, or by synthetic models [19].

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 [11]).

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

where $i \in (1, 2, 3, \dots)$, 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 (see [11] for details).

Lastly, $\beta_{\{r_i,reality\}}^l(\Delta)$ 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.

In [11], statistical methods are used in order to demonstrate that the values of the *L* and *R* parameters of $\beta_{r_i}^l$ have an adequate level of predictability, and, hence, results are valid up to certain level of confidence.

IV. Conclusion

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

In particular, we demonstrate that the end-to-end latency quantity in RTC, i.e., the maximum delay experienced by an event stream at given individual software servers, can be used to evaluate worst case scenarios. Hence, the RTC-based stochastic analysis (Section III-B) 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.

We discuss a novel approach for modeling cloud-based systems, and 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.

We consider that other specific VMs management issues such as VM provisioning, VMs performance interference effect, autonomic resource management, etc, could be analyzed by using MPA-RTC.

References

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[15] 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.

[16] 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.

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

[18] 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.

[19] 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.