RoC Prediction for Bi-Objective Cost-QoS Optimization of Cloud VoIP Call Allocations

Jorge M. Cortés-Mendoza[1], Andrei Tchernykh[2✉] Gleb Radchenko[3] Alexander Yu Drozdov[4]
CICESE Research Center, Ensenada, Baja California, México.
South Ural State University, Chelyabinsk, Russia.
Moscow Institute of Physics and Technology, Moscow, Russia.
[1]jcortes@cicese.edu.mx, [2✉]chernykh@cicese.mx
[3]gleb.radchenko@susu.ru
[4]alexander.y.drozdov@gmail.com

***Abstract*—In this paper, we present cloud VoIP scheduling strategies to provide appropriate levels of quality of service to users, and cost to VoIP service providers. This bi-objective problem is reasonable and representative for real installations and applications. We conduct comprehensive simulation on real data of sixty four strategies with dynamic prediction of the load. We show that our prediction rule that consider the number of Virtual Machines (VMs) running in the system improves the efficiency of traditional rate of change algorithm. It provides suitable quality of service and lower cost. Variations of VM startup time delays permit to evaluate the prediction rules under different scenarios and assess the robustness of all strategies.**

Keywords— call allocation; cloud voice over IP; quality of service; bin packing; load prediction.

I. INTRODUCTION

In the last years, companies are moving to cloud model, because it is considered as feasible, innovative, and profitable solution. Main reasons of business migrations are cost savings and scalability. Companies can increase earnings and provide an adequate Quality of Services (QoS) to the users. Additionally, it allows to provide services in several geographical areas without deploying local infrastructure. The use of cloud infrastructure has grown in many services used every days [1, 2, 3]: photos, music, television, webmail, voice over IP, etc.

Voice over IP (VoIP) is considered the next evolutionary step in the telephone systems. It has had a great adoption and fast growing in cloud computing. Cloud VoIP (CVoIP) offers higher flexibility and more features than traditional telephony (PSTN) infrastructure. The main benefits of this technology, over traditional telephony model and VoIP, are cost effectiveness, fewer operational issues, reliability, flexibility, scalability, etc.

In CVoIP solution, providers rent Virtual Machines (VMs) on local clouds to deploy IP Private Branch Exchange (PBX) systems. VMs execute specialized software to emulate the telephone system, where Asterisk [4] is the backbone to build CVoIP system. Asterisk instances provide many features: calls, voice mails, video/audio conferences, interactive phone menus, call distribution, etc. Furthermore, users can transfer images and texts, and they can create new functionalities, opening up a complete new experience in telephonic communication.

The adoption of CVoIP depends on offering competitive prices and maintain adequate QoS levels. CVoIP providers face the QoS problem improving various parameters: the quality of voice, transit time of packets across the Internet, signaling overhead, end-to-end delay, jitter, codec compression technique, among others [5].

In our previous works [6, 7, 8], we formulated the scheduling problem of VoIP services in cloud environments, and proposed a new model with bi-objective optimization. The model considered the special case of the on-line non-clairvoyant dynamic bin packing problem, and debated solutions for provider cost minimization, and QoS optimization. All strategies focus on VMs provisioning, they use information about utilization, rental time, and startup time delay (StUp) of VMs. Our most recent work [9] focuses in prediction of calls arrival, the main objective is to estimate the amount of arrival calls to alleviate the effects generated by StUp on CVoIP.

In this paper, we extend the previous study of dynamic prediction to estimate the amount of calls arrival, reduce the rent of VMs, and minimize the QoS reduction.

We propose and evaluate a new mechanism to request VMs: when the predicted VMs utilization is over the fixed threshold. Our fast dynamic load prediction is calculated based on the speed of utilization increment and the number of VMs running in the system.

We analyze sixty four on-line non-clairvoyant scheduling strategies with dynamic load prediction. The bi-objective scheduling strategies consider billing hours and quality degradation optimization criteria.

We perform wide simulation analysis on real workload of the MIXvoip company [3], and show that our prediction rule improves original rule, it can reduce the amount of calls in waiting queue.

The paper is organized as follows. Next section presents underlined infrastructure and software in VoIP. Section III briefly reviews related works on call load balancing, and load estimation. Section IV provides the problem definition and proposed model. Section V describes VoIP call allocation strategies. Section VI discusses our experimental setup, workload and studied scenario. Section VII presents experimental analysis of the provider cost and quality of service. Section VIII highlights the final conclusions of the paper and future work.

II. INTERNET TELEPHONE

VoIP consists of transmitting encoded voice, into a digital form, over the internet. The signal must be decoded, on the receiving end, to let recipient hear the sender's voice.

CVoIP takes out overprovision and underprovision, which denotes in a reduction of costs and an increment of scalability. The deployment of the virtual infrastructure reduces operational complexity, the main advantages of CVoIP are: easy implementation, faster provisioning, and integration of services dynamically scalable. Further, it adds new features and capabilities for users (data transfer availability, integrity, and security).

Figure 1 shows the architecture of CVoIP. Internally, VMs run specialized software where Voice Nodes (VNs) are the core part of the system. The VMs utilization have to be high in order to optimize the overall system performance and reduce provider cost. However, this reduction degrades the quality of the calls (Section II.B). On the other hand, reduction on the load of the VMs can guarantee the QoS but it increases the idle time of VMs, hence, the expenses for VoIP provider.

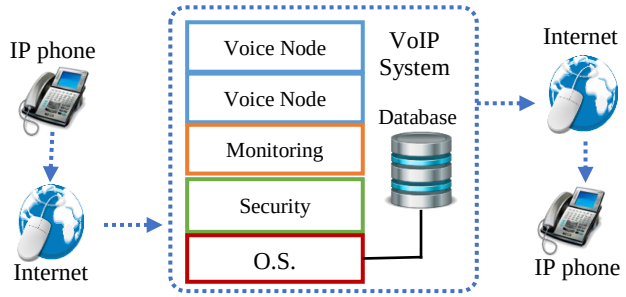

Fig. 1. VoIP architecture.

A. *Infrastructure*

Super Voice Node (SN) and Super Voice Nodes Cluster (SNC) concepts were developed by MIXvoip company [3] to enrich features for telephone exchanges. They combine cloud services, VoIP services, and traditional telephony services. SN model provides redundancy in communication and high voice quality between SNs (distributed in different geographical areas) through the public Internet, it permits short paths between two local users.

The most known telephone software for processing calls and providing a powerful control over call activity is Asterisk [4]. It is a framework, under free license, for building multi-protocol, real-time communication solutions providing a powerful control over call activity. It processes calls, and connects to other telephone services, such as the Public Switched Telephone Network (PSTN) and VoIP services.

The CVoIP system consists of multiple voice nodes that run and handle calls. Each node has Asterisk running process with unique IP address that is used by end users to connect inside and outside the network.

B. *Quality of service*

The VoIP services have stricter constraints and sensitive factors. Call processing and call delivery are two main issues, which determine the QoS (quality of calls). Call processing focuses on the time to set-up and tear-down the call, and on converting the voice portion of the calls into packets transported over the network. Adequate quality of voice is the most important aspect in call processing.

The quality of voice is subjectively perceived by the listener. A common benchmark used to determine the quality of voice is the Mean Opinion Score (MOS). It evaluates the quality of speech provided by a codec. Each codec provides a certain quality of speech only if processor utilization is low enough. Theoretically, processor utilization of 100% provides the best expected performance. However, Eleftheriou [10] showed that CPU cannot handle the stress when utilization is up to 85%, then jitters and broken audio symptoms appear. Additionally, the author did not report any influence of memory on the voice quality reduction.

The codec also increases the bandwidth but it is less significant than the same codec adds to CPU utilization. For instance, 6,500 calls per second not exceed 100 Mbps of bandwidth, and 10,000 calls not reach 400 Mbps [7]. A connection of 100 Mbps can support between 6,000 and 25,000 calls, depending on type of codecs. However, the amount of calls handled by CPU is less than supported by 100 MBs connection. Therefore, call processing is the key feature to guarantee the QoS. A method to ensure QoS is to bound processor utilization [7].

C. *CPU utilization*

Calls have different impact on the processor utilization [6] depending on the operations performed by Asterisk. If transcoding operations are performed, the utilization is higher than when transcoding is not used. In the latter case, Asterisk is in charge of only routing the call. However, depending on the codec, the processor load is influenced as well.

The performance of ATOM processors are analyzed for VoIP [10] considering calls amount, utilization, power consumption, database messaging, registration and call performance. The author concludes that CPU can process from 70 to 500 calls with 100% of utilization.

D. *VoIP provider optimization criteria*

Inefficient resource utilization directly leads to higher costs. VoIP providers should use the resources efficiently to offer competitive prices. Virtualization technologies allow creating VoIP virtual servers hosted in clouds and rented (leased) on a subscription basis to any scale.

In a typical cloud scenario, VoIP provider have certain service guarantees distinguished by the amount of computing power received within a requested time, and a cost per unit of execution time. In this paper, two criteria are considered: the billing hours for VMs to provide a service, and the number of call send to waiting queue due to the lack of resources.

III. RELATED WORK

This section describes the last advances in call load balancing and load estimation, both topics are fundamental in our research.

A. Call load balancing

The main objective of call load balancing is to reduce the infrastructure cost and guarantee that service will be delivery in the best possible way. Several algorithms have been proposed to improve the performance of CVoIP systems.

The Virtual Load-Balanced Call Admission Controller (VLB-CAC) [11] is a strategy to balance calls and provide admission control for Session Initiation Protocol (SIP) servers. It has mechanisms to predict the call number, required resources, and select the most appropriate VM instances considering CPU, memory and bandwidth. The authors propose a model to maximize the resource usage and system throughput.

Mobicents SIP Load Balancer [12] is a SIP-based proxy for VoIP infrastructure with multiple ingress proxy servers. It allows to avoid congestion, bad resource utilization, and overload. Mobicents uses Round Robin algorithm to distribute the traffic between servers, and it contemplates requests and system parameters, like CPU.

B. Load estimation.

Prediction techniques to anticipate the incoming traffic (calls for VoIP) are applied for an efficient distribution of the load in the system. The goals of traffic prediction on cloud computing is to minimize the infrastructure costs and improve the QoS to the end user.

Incoming voice traffic prediction with Interactive Particle Systems (IPS), Gaussian Mixture Model (GMM), and Gaussian Process (GP) is studied on [13]. Authors compare the performance of the algorithms during different time frames in a real VoIP environment. The authors provide flexible modeling approaches, traffic shaping determined by clients, and scalable solutions with good prediction precision. All algorithms were trained and tested under different scenarios.

Rate of change (RoC) [14] is a strategy for load balancing tasks in distributed system. It allows to trigger a dynamic, distributed, and implicit load balancing mechanism. The balancer (Bal) makes job distribution decisions at run-time, locally and asynchronously. Each Bal considers its own load; migration does not depend on the load of other Bals. The migration decision depends on current load, load changes in the time interval (rate of load change), and current load balancing parameters. Difference in Load (DL) is used as an estimation of load, its prediction for the next time slot, and detection of the need for load balancing process.

IV. Model

A. Infrastructure model

The model follows the our previous works [6, 7, 8, 9], where cloud VoIP infrastructure consists of m heterogeneous super node clusters: $SNC_1, SNC_2, \dots, SNC_m$ with relative speeds $s_1, s_2, \dots . s_m$. Each SNC_i, for all $i = 1, \dots, m$ consists of m_i SNs. Each SN_k^i, for all $k = 1, \dots, m_i$, runs $k_i(t)$ VMs at time t. We assume that VMs of one SN are identical and have the same processing capacity. The virtual machine VM_j is described by a tuple $\{st_j, size_j, StUp_j\}$ that consists of its request time $st_j \geq 0$, the processing capacity $size_j$ in MIPS, and startup time delay $StUp_j$.

B. Calls model

We consider n independent calls $J_1, J_2, \dots, J_n$ that must be scheduled on set of SNCs. The call J_j is described by a tuple $\{r_j, p_j, u_j\}$ that consists of its release time $r_j \geq 0$, duration p_j (lifespan), and contribution to the processor utilization u_j due to the used codec. The release time is not available before the call is placed, and its duration is unknown until the call has been completed. The utilization is a constant for a given call that depends on the used codec and VM processing capacity.

C. Criteria

We define the provider cost model by considering a function that depends on the number of rented VMs. We denote the number of Billing Hours in SNC_i by:

$$\bar{b}_i = \int_{t=0}^{C_{max}} k_i(t) \cdot m_i \, dt \tag{1}$$

and run in all SNC by:

$$\bar{b} = \sum_{i=1}^{m} \bar{b}_i \tag{2}$$

Additionally, we analyze the number of calls waiting on the queue, it occurs when the VMs cannot process the arriving calls, so the calls wait on a queue for resources on a VM. The amount of Calls to Queue is defined by:

$$\bar{c} = \sum_{j=1}^{n} \delta(s_j - r_j) \tag{3}$$

where s_j is the initiation time of J_j, and $\delta(\alpha)$ is 1 if $\alpha > 0$ and 0 otherwise.

D. Evaluation method

We use Set Coverage [15] *SC(A,B)* to analyze the bi-objective problem, it is a formal and statistical metric that calculates the proportion of solutions in B, which are dominated by solutions in A:

$$SC(A,B) = \frac{|\{b \in B; \mid \exists a \in A \; ; \; a \leq b\}}{|B|} \tag{4}$$

A metric value *SC(A,B)* = 1 means that all solutions of *B* are dominated by *A*, whereas *SC(A,B)* = 0 means that no member of *B* is dominated by *A*. This way, the larger the value of *SC(A,B)*, the better the Pareto front *A* with respect to *B*. Since the dominance operator is not symmetric, *SC(A,B)* is not necessarily equal to 1–*SC(A,B)*, and both *SC(A,B)* and *SC(B,A)* have to be computed for understanding how many solutions of *A* are covered by *B* and vice versa.

V. Call Allocation

The call allocation problem is similar to a well-known dynamic bin-packing problem, a variation of the classical NP-hard optimization problem with high theoretical relevance and practical importance. In VoIP, the scheduler decides whether the call is placed into one of the currently available VMs or new VM must be run. The scheduler only knows the contribution of the call to the VM utilization u_j. All decisions have to be made without knowledge of duration of the call, call arrival rate, etc.

Temporal existence of the items is the principal novelty of this problem. Call lifespan, and call allocations determine the

state of the VMs. Unlike the standard formulation, bins are always open and dynamic, even completely packed. Items in bins can be terminated (call termination) and utilization can be changed at any moments, then VMs can use free space to processing more calls.

We consider a scenario where the bin size is equals to 0.7 that corresponds to 70% of VM utilization. The scheduler has no information of the calls arrival rate, and it takes decisions depending on the current system state.

A. *Call allocation*

In this paper, we consider call allocation taking into account VM startup time delay. A new VM is requested when the predicted utilization exceeds the fixed threshold. During VM StUp, old VM continues call processing with utilization more than threshold reducing QoS. The worst case appears when the current VM does not have enough resources to process arriving calls. In this case, the system puts the calls into a queue, waiting for available resources.

Table 1 summarizes the call allocation strategies that request new VM when call prediction exceeds the threshold. Each strategy uses four prediction sample interval: 10, 20, 30, and StUp seconds. All strategies consider a prediction model based on Rate of change.

TABLE I. CALL ALLOCATION STRATEGIES WITH PREDICTION.

		Description
KF	Rand	Allocates job *j* to VM randomly using a uniform distribution.
	RR	Allocates job *j* to VM using a Round Robin algorithm.
UA	FFit	Allocates job *j* to the first VM capable to execute it.
	BFit	Allocates job *j* to VM with smallest utilization left.
	WFit	Allocates job *j* to VM with largest utilization left.
RA	MaxFTFit	Allocates job *j* to VM with farthest finish time.
	MidFTFit	Allocates job *j* to VM with shortest time to the half of its rental time.
	MinFTFit	Allocates job *j* to VM with closest finish time.

B. *Prediction mode*

Rate of Change (RoC) [20] is a dynamic distributed load balancing algorithm, it achieves the goal of minimizing processor idling times without incurring into unacceptably high load overheads. Resources calculate the change in their load between two Sample Intervals (*SI*), it is an adaptive parameter and the length may vary. Each resource calculates the Difference in Load (*DL*), and use it as estimation on load for the next *SI*. A finer sampling allows to improve the balance the system, but it increases the overhead.

We use the *DL* as a mechanism to predict requests for new VMs, which can be provided after StUp time. *DL* is used to estimate the number of VMs after *SI*, it permits to initialize VMs before the arriving calls degrade the QoS.

Let $u_i(t)$ be the utilization of SNC_i at time t, the rate of load change during SI=$[t - Si, t]$ is defined by:

$$\Delta_i(t) = (u_i(t) - u_i(t - Si)) \quad (5)$$

Figure 2 shows *DL* changes scenario. Solid lines represent real utilization and dashed lines are estimated utilization. If the utilization is larger than *Ub* then the broker requests for a new VM. *Ub* is an adjustable parameters than depend on the utilization threshold.

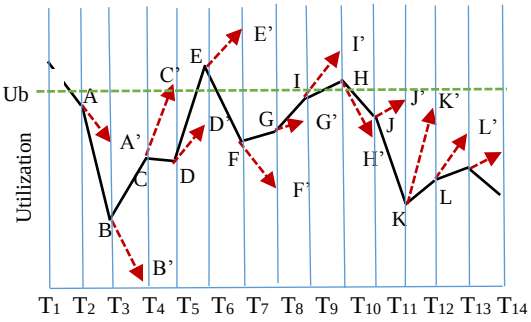

Fig. 2. Calls prediction mechanism.

Our goal is to choose an adequate prediction rule $\Delta_i(t)$. In this work, we propose a new prediction rule for Rate of Change algorithm (nRoC) to estimate the load and evaluate its performance by simulation.

C. *Prediction Rule*

Prediction rule is fundamental to estimate the number of VMs to provide the service in CVoIP. RoC uses a lineal rule than only considers the difference between two sample intervals, see (5). In our previous study, we identify that CVoIP system is more vulnerable when the number of VMs is small, so we propose a prediction rule that consider the number of VMs running in the system. It is defined by (6), where $k_i(t)$ describes the number of VMs running. When $k_i(t)$ is high, then the system can respond better to abrupt changes on the load, due to it has more resources to support high volumes of arriving calls.

$$\Delta_i(t) = (u_i(t) - u_i(t - Si))/k_i(t), \quad (6)$$

VI. PERFORMANCE EVALUATION

We perform experiments using standard trace based simulator CloudSim [16] extended by our algorithms, supporting dynamic calls arrival, VM startup delays, statistical analysis, and prediction model.

A. *Workload*

We use traces of real VoIP service [13] that include several information about phone calls. Table 2 and Table 3 present the number of calls per day and call duration. The histogram [13] shows that the load is typical for business clients with two peaks in 10-12 and 14-16 hours.

TABLE 2. NUMBER OF CALLS PER DAY.

Day	Total	Average
Monday	131,443	21,906
Tuesday	129,379	21,563
Wednesday	131,460	21,910
Thursday	130,439	21,739
Friday	120,999	20,166

TABLE 3. CALL DURATION.

Time (min.)	Number of calls
0 - 1	310,602
1 - 2	136,211
2 - 3	68,988
3 - 4	39,392
4 - 5	23,397
…	…
19 - 20	721

VII. EXPERIMENTAL ANALYSIS

The VoIP providers rent VMs on an hourly base. When the VM rental time is finished, the VM can be turned off only if VM

is not processing calls. In any other case, this VM continue running for one hour more.

We compare two prediction rules for Roc strategies. In order to evaluate their performance, we incorporate seven StUp delays: 45, 90, 135, 180, 225, 270 and 315 sec., and four *SI*: 10, 20, 30, and StUp sec. as test cases [8, 9].

Figure 3 shows an example of real and predicted values of load for 25 min. with 10 sec. of *SI*, we can see that nRoC predictions fit better with real values of load.

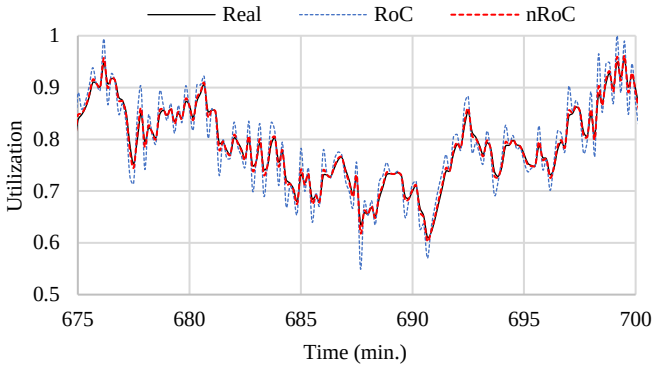

Fig. 3. Load estimation with 10 sec. of sample interval.

Table 4 shows the values *SC(A,B)* for the dominance of strategy RoC over nRoC with four *SI*, we perceive that RoC dominates the solutions of nRoC in the range 8.3 – 37.5%, with 22.2% in average.

TABLE 4. SET COVERAGE OF ROC.

Strategy \ StUp	45	90	135	180	225	270	315
Bfit	0.225	0.142	0.117	0.125	0.125	0.192	0.167
FFit	0.133	**0.083**	0.117	0.117	0.158	0.175	0.117
MaxFTFit	0.142	0.217	0.192	0.108	0.158	0.183	0.100
MidFTFit	0.192	0.242	0.158	0.258	0.217	0.175	0.167
MinFTFit	0.325	0.225	0.308	0.350	**0.375**	0.333	0.167
Rand	0.333	0.208	0.292	0.258	0.300	0.300	0.300
RR	0.250	0.283	0.233	0.333	0.225	0.283	0.358
WFit	0.308	0.267	0.208	0.283	0.250	0.233	0.317

Table 5 shows the values *SC(B,A)* for the dominance of strategy nRoC over RoC, we see that nRoC dominates the solutions of RoC in the range 67.5 – 99.2%, with 84.3% in average. The dominance *SC(B,A)* confirms that nRoC can generate solution 84.3%, in average, better or at least with the same quality of Roc, for different StUp and *SI*.

TABLE 5. SET COVERAGE OF NROC.

Strategy \ StUp	45	90	135	180	225	270	315
BFit	0.908	0.958	0.933	0.933	0.933	0.883	0.858
FFit	0.967	0.983	**0.992**	0.950	0.917	0.900	0.908
MaxFTFit	0.958	0.875	0.942	0.917	0.883	0.833	0.858
MidFTFit	0.867	0.817	0.900	0.792	0.850	0.825	0.842
MinFTFit	0.792	0.858	0.775	0.733	**0.675**	0.700	0.725
Rand	0.775	0.858	0.800	0.833	0.775	0.742	0.742
RR	0.792	0.783	0.842	0.792	0.833	0.800	0.700
WFit	0.775	0.833	0.883	0.833	0.825	0.808	0.750

VIII. CONCLUSIONS

In this paper, we introduce a new prediction rule for Rate of Change algorithm to optimize cloud VoIP call allocation strategies. Its main function is to estimate the number of VMs needed to provide the VoIP service in a cloud infrastructure. Our analysis shows that the new approach can generate better solutions than original one by 84.3%, in average, for different scenarios. However, further study is required to assess their actual performance and effectiveness in a real domain. This will be the subject of future work.

REFERENCES

[1] https://cloud.google.com/customers/, accessed June 20, 2017.

[2] https://aws.amazon.com/es/solutions/case-studies/?nc2=%20h_ql_ny_livestream_blu, accessed June 20, 2017.

[3] https://www.mixvoip.com/, accessed June 20, 2017.

[4] L. Madsen, J. V. Meggelen, and R. Bryant. Asterisk: The definitive guide. O'Reilly Media, Inc., 2011.

[5] H. P. Singh, S. Singh, J. Singh, and S. A. Khan. VoIP: State of art for global connectivity—A critical review. Journal of Network and Computer Applications, 37, 365-379, 2014.

[6] J. M. Cortés-Mendoza, A. Tchernykh, A. M. Simionovici, P. Bouvry, S. Nesmachnow, B. Dorronsoro, and L. Didelot. VoIP service model for multi-objective scheduling in cloud infrastructure. International Journal of Metaheuristics, 4(2), 185-203, 2015.

[7] J. M. Cortés-Mendoza, A. Tchernykh, F. A. Armenta-Cano, P. Bouvry, A. Yu. Drozdov, and L. Didelot. Biobjective VoIP Service Management in Cloud Infrastructure. Scientific Programming, vol. 2016, Article ID 5706790, 14 pages, 2016. http://dx.doi.org/10.1155/2016/5706790

[8] J. M. Cortés-Mendoza, A. Tchernykh, A. Yu. Drozdov, and L. Didelot. Robust cloud VoIP scheduling under VMs startup time delay uncertainty. 9th International Conference on Utility and Cloud Computing, 234-239, 2016.

[9] J. M. Cortés-Mendoza, A. Tchernykh, A. Feoktistov, I. Bychkov, and L. Didelot. Load-Aware Strategies for Cloud-based VoIP Optimization with VM Startup Prediction. 7th IEEE Workshop Parallel / Distributed Computing and Optimization (PDCO 2017), 472- 481, 2017.

[10] 3CX Phone System and ATOM N270 Processor Benchmarking. http://www.3cx.com/blog/voip-howto/atom-processor-n270-benchmarking, accessed June 20, 2017.

[11] A. Montazerolghaem, M. Hossein, A. Leon-Garcia, M. Naghibzadeh, and F. Tashtarian. A Load-Balanced Call Admission Controller for IMS Cloud Computing. IEEE Transactions on Network and Service Management, 2016.

[12] http://www.mobicents.org , accessed June 20, 2017.

[13] A. M. Simionovici, A. A. Tantar, P. Bouvry, A. Tchernykh, J. M. Cortés-Mendoza, and L. Didelot. VoIP traffic modelling using Gaussian mixture models, Gaussian processes and interactive particle algorithms. In 2015 IEEE Globecom Workshops 2015 Dec 6 (pp. 1-6).

[14] L. M. Campos, and I. D. Scherson,. Rate of change load balancing in distributed and parallel systems. Parallel Computing, 26(9), 1213-1230, 2000.

[15] E. Zitzler. Evolutionary algorithms for multiobjective optimization: Methods and applications, PhD thesis, Swiss Federal Institute of Technology. Zurich (1999).

[16] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Software: Practice and Experience (SPE), Volume 41, Number 1, 23-50, 2011.