Robust Cloud VoIP Scheduling under VMs Startup Time Delay Uncertainty.

Jorge M. Cortés-Mendoza
CICESE Research Center.
Ensenada, Baja California, México.
jcortes@cicese.edu.mx

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

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

Loic Didelot
MIXvoip S.A.
Sandweiler, Luxembourg.
ldidelot@mixvoip.com

ABSTRACT
In this paper, we address cloud VoIP service orchestration and scheduling to provide appropriate levels of quality of service to users, and performance to VoIP service providers. We consider voice quality affected by call processing, and cost contributed by billing hours for used VMs in a cloud. We believe that this bi-objective focus is reasonable and representative for real installations and applications. We conduct comprehensive simulation of our calls load balancing strategies on real data and show that not all approaches provide suitable quality of service. We analyze eight on-line dynamic non-clairvoyant scheduling strategies with variations in VM startup time delays to deal with realistic VoIP cloud environments. We show that the proposed strategies outperform currently in use strategies in terms of quality of service and provider cost. The robustness of these strategies is also discussed.

Keywords
Call allocation; Scheduling; Cloud computing; Cloud Voice over IP; Quality of Service; Bin packing.

1. INTRODUCTION
Cloud computing has been widely adopted by many companies as a profitable business model. Voice over IP (VoIP) technology is migrated to the Cloud considering it as a long-term service roadmap. The main benefits behind this technology, over traditional telephony model and VoIP, are cost effectiveness, flexibility, scalability, extended service variety, etc. It can cope with different workloads, and dynamically adapt resources in response to demand.

In Cloud VoIP (CVoIP) solution, calls, voice mails, video/audio conferences, interactive phone menus, call distribution are operated by Asterisk [1] software for a telephone private branch exchange (PBX) executed in Virtual Machines (VMs).

The Quality of Service (QoS) is important for the success of the business. Many factors come into play when considering CVoIP phone system: the quality of voice, transit time of packets across the Internet, queuing delays at the routers, signaling overhead, end-to-end delay, jitter, call set-up and tear-down time, codec compression technique, processing capability, etc. [2].

In our previous works [3, 18], we formulate the problem of scheduling of VoIP services in cloud environments, and propose a new model for bi-objective optimization. We consider the special case of the on-line non-clairvoyant dynamic bin packing problem, and discuss solutions for provider cost and quality of service optimization. We propose twenty call allocation strategies and evaluate their performance by comprehensive simulation analysis on real workload considering six months of the MIXvoip company service [17].

In this paper, we continue this study. We analyze the strategies focusing on one more important factor that affects time sensitive applications and resource auto-scaling mechanisms crucial in CVoIP. We take into account VM time provisioning. The unpredicted changes in VM startup times (StUp) could result in resource under- or over-provisioning, hence, in call quality reduction, or cost increasing. While there are studies of the startup of VMs in clouds, its impact on voice quality in CVoIP is not hardly understood.

In this paper, we consider realistic scenarios, where VMs have startup time delays that depend on the instance type (spot, on-demand), operating system (Linux, Windows), size of the OS image, cloud providers (EC2, Rackspace, and Azure [4]), etc. We propose and evaluate eight on-line dynamic non-clairvoyant scheduling strategies considering billing hours for used VMs and voice quality. We conduct comprehensive simulation on real data of the MIXvoip provider, and show that not all approaches are suitable due to their reduced voice quality and increased amount of calls to queue.

The paper is structured as follow. The next section briefly discusses VoIP service considering underlined infrastructure and software. Section 3 reviews related works on bin packing variants and VM startup time delays. Section 4 provides the problem definition and proposed model. Section 5 describes VoIP call allocation strategies. Section 6 discusses our experimental setup, workload and studied

* Corresponding author.

UCC '16, December 06-09, 2016, Shanghai, China
© 2016 ACM. ISBN 978-1-4503-4616-0/16/12$15.00
DOI: http://dx.doi.org/10.1145/2996890.3007865

scenarios. Section 7 presents experimental analysis of the provider cost and quality of service. Section 8 concludes the paper.

2. INTERNET TELEPHONE

The Internet telephony VoIP refers to the provisioning of voice communication services over the Internet. It reduces the infrastructure and communication cost, therefore, achieves significant call rate reduction.

Voice nodes (VNs) are the core part of the VoIP telephony system (Figure 1). They execute specialized software to emulate a telephone exchange, gateways, interconnection switches, session controllers, firewall, etc. VNs communicate with the database, where all the users are registered, and calls are recorded with details such as: destination, duration, etc. They also process voicemails, call forwarding, music on hold, conference calls, signaling, voice signal digitization, encoding, etc.

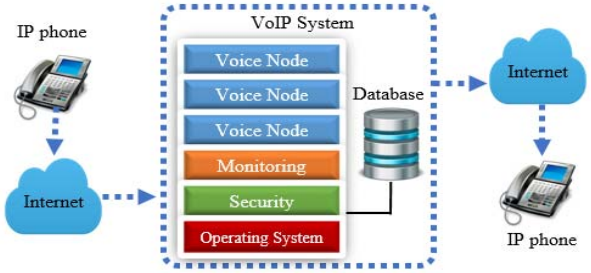

Figure 1. VoIP architecture [18].

Overprovisioning is a common problem of traditional VoIP solutions. The scalability requires the service availability all the time for any number of users. With the increasing number of clients, providers need to invest in a large infrastructure to avoid loss of calls (hence, users). Overrunning cost is not an efficient solution, even with the growing number of the customers, and potential safety of being able to deliver services during peak hours or abnormal system behavior.

A cloud based VoIP can reduce the costs and overprovisioning. Further, it adds new features and capabilities for users (data transfer availability, integrity, and security), and providers (easier implement and integrate services that are dynamically scalable). The virtual infrastructure can be easily scalable and deployed.

In CVoIP model proposed in [18], the voice nodes are operated by VMs. To optimize the overall system performance and reduce provider cost, the VM utilization has to be high. However, it reduces quality of the calls. Hence, load of the VoIP servers should be reduced to guarantee the QoS. On the other hand, the idle time increases the useless expenses of the VoIP provider. We consider two objective problem: minimizing the number of VMs without overloading them to improve both the provider cost and QoS.

2.1 Infrastructure

Mixvoip [17] developed the concept of the Super-Node (SN) and Super Nodes Cluster (SNC) to enrich features for telephone exchanges, combine cloud service with smart business telephony, VoIP and other telephony services [3]. SNC is a set of SNs deployed in a cloud, and interconnected logically at a local level. It provides short path between two local users. This deployment brings redundancy on a given geographical area, but ensures a high voice quality between the SNC nodes through the public Internet. It provides services near ISDN quality in a public IP network.

Asterisk is the most known Private Branch Exchange (PBX) software that includes components necessary to build scalable phone systems, see Madsen et al. 2011 [1]. It is a framework 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 VoIP system consists of multiple heterogeneous 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.

2.2 Quality of service

The VoIP QoS is determined by two factors: call processing (quality of voice, call set-up and tear-down time, etc.) and call delivery (transit time of packets across the Internet, queuing delays at the routers, packet travel time from source to destination, jitter, packet loss, etc.).

The quality of voice is the most important aspect in call processing. A common benchmark used to determine the quality of voice is the Mean Opinion Score (MOS), a subjective evaluation of the listener. 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, 2015 [6] shows that 20 calls with total CPU usage 19% do not produce jitters. With increasing number of calls up to 90, hence, utilization up to 85%, CPU cannot be able to handle the stress anymore, and jitters and broken audio symptoms appear. Additionally, the author shows that memory size does not influence significantly on the voice quality.

Table 1 shows the bandwidth used by different codecs, considering that VoIP calls use audio streams for endpoints (a call between two parties will use double of bandwidth). The number of calls supported in 100 Mbps connection is between 6,000 and 25,000 depending on codec of the calls. Additionally, Montazerolghaem et al. [7] report that the consumed bandwidth of 6,500 calls per second not exceed 100 Mbps, and 10,000 calls not reach 400 Mbps.

Table 1. Codec bandwidth.

Codec	Bit Rate (kbps)	MOS	Bandwidth (kbps)
G.711	64	4.1	87.2
G.729	8	3.92	31.2
G.723.1	6.3	3.9	21.9
G.723.1	5.3	3.8	20.8
G.726	32	3.85	55.2
G.726	24	-	47.2
G.728	16	3.61	31.5
G722_64k	64	4.13	87.2
ilbc_mode_20	15.2	NA	38.4
ilbc_mode_30	13.33	NA	28.8

The call processing is the key feature to guarantee the QoS. Cortés-Mendoza et al. 2015 [3] propose to limit processor utilization in order to ensure it.

2.3 CPU utilization

Cortés-Mendoza et al. 2015 [3] shows that calls have different impact on the processor utilization 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.

Table 2 shows processor utilization for call without transcoding presented by Montoro et al. 2009 [8].

Table 2. Utilization for calls without transcoding

Protocol	Codec	10 Calls	1 Call
SIP/RTP	G.711	2.36%	0.236%
SIP/RTP	G.726	2.13%	0.213%
SIP/RTP	GSM	2.58%	0.258%
SIP/RTP	LPC10	1.92%	0.192%

Eleftheriou, 2015 [6] analyzes the performance of ATOM processors for VoIP 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.

2.4 VoIP provider optimization criteria

To offer competitive prices to customers, VoIP providers should optimize the server costs (compute, storage, software, and associated VoIP components), infrastructure costs (power distribution, cooling equipment, space for facilities, etc.), operational costs (energy, cooling, etc.), network costs (links, transit equipment), etc. Inefficient resource utilization has a direct negative effect on performance and cost. 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, a VoIP provider can select different resources that are available on demand from cloud providers. They 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 voice quality reduction.

This approach is not restricted to find a unique solution but a set of solutions known as a Pareto optimal set. A tradeoff between objectives depends on the VoIP provider's preference.

3. RELATED WORK

The next section describes recent results on call load balancing, bin-packing problem, and VM startup time delays related with VoIP management.

3.1 Call load balancing

The migration of VoIP service to clouds triggers researches on call allocation, load balancing, quality of service, etc. The main objective 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 system.

Montazerolghaem et al. [9] develop a load balancer and admission controller for SIP servers, and propose a model to maximize the resource usage and system throughput. The Virtual Load-Balanced Call Admission Controller implements mechanisms to predict the number of calls, required resources, and selects the most appropriate VM instances considering CPU, memory and bandwidth.

3.2 Bin packing.

Bin packing techniques are widely used to allocate applications into the VMs and/or VMs on the resources.

Song et al, [9] propose a relaxed on-line bin packing algorithm called Variable Item Size Bin Packing. It allocates data center resources using live VM migration. The main goal is to restrict the combinations of tasks in a bin to minimize the amount of wasted space. The authors evaluate the effectiveness of the algorithm, and provide a theoretical proof for the number of used servers and VM migrations. Its multi-dimensional version considers a mix of CPU and network.

Wolke et al. [10] analyze a wide variety of controllers for VMs placement and reallocation using the resources availability (CPU, memory, etc.) and placing incoming VMs on servers as long as their residual capacity allows. Bin packing algorithms are used in this stage. The reallocation controller triggers VM migration in order to optimize the allocation, when underloaded servers are emptied and overloaded ones are relieved. The authors found that combinations of placement controllers and periodic reallocations achieve the highest energy efficiency subject to predefined service levels.

Li et al. [11] consider a variant of Dynamic Bin Packing problem to solve the request dispatching problem arising from cloud gaming. In cloud gaming system, computer games run on cloud servers, where each game instance demands certain amount of resources. In order to provide a good user experiences, requests must be dispatched with enough resources of CPU and GPU. The main objective is to minimize the number of servers (bins) to process the gaming requests (items) due to each resource adds a proportional cost to the duration of its usage. The authors analyze the competition ratio for original and modified versions of Best Fit and First Fit algorithms.

3.3 Startup Time.

Elasticity is defined as the ability to dynamically acquire or release computing resources under demand. Some advantages of cloud elasticity are: avoiding the wasteful practice of over-provisioning, adapting its capacity to unpredictable workload over time, and repercussion on energy consumption. However, elasticity is meaningful if VMs can be deployed in time and be used within the user time expectation. The unexpected VM StUp could result in service quality reduction and resource under-provisioning. In order to reduce the factor that affect the execution of time-critical applications, it is necessary to study the effects of VM StUp delays. An important focus of such a study is robustness as the ability to cope with variations from the nominal operating conditions. In this spirit, a robust load balancer should be able to sustain acceptable performance despite foreseeable StUp variations and call arrival rate.

Mao et al. [4] study the VM StUp in clouds and analyze the relationship between the StUp and different factors (OS image size, instance type, data center location and the number of instances acquired at the same time). VM StUp is the time period that cloud providers need to find a spot in the data centers to provision VMs, allocate resources (e.g. IP addresses) to VMs and copy/boot/configure the OS image. It varies from 44 to 96 seconds for Linux instances, and from 429 to 810 seconds for Windows instances on EC2 and Rack-space. On Azure provider, VM StUp varies from 356 to 406 seconds.

Razavi et al. [5] study the dependence between VM startup time, disk size, and content. The authors develop and evaluate an approach for consolidating VM instances. They reduce the disk size and content by selecting the desired set of services to create VMs disk with the minimal required size. This approach can be applied to all VM images for which the user can express which top-level software packages are required for its desired functionality.

Hoffman et al. [14] measure the speed of VM creation and time to accessing the SSH among cloud providers located in different regions in the world. The fastest cloud vendor is Google Cloud with 31 seconds on average to complete both processes. It is followed by Amazon and Vexxhost with 47 seconds, and Linode with 57 seconds. The cloud providers with worst results are Rackspace with 128 seconds and Microsoft Azure with 138 seconds.

4. MODEL

In this paper, we follow the model proposed in our previous work (Cortés-Mendoza et al. 2015, 2016 [3, 18]), 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.

We contribute with a new version of the model by introducing VM startup delays. In this version, the virtual machine VM_j is described by a tuple $\{st_j, d_j, size_j\}$ that consists of its request time $st_j \geq 0$, startup delay d_j, and the processing capacity $size_j$ in MIPS.

The SNC contains a set of routers and switches that transport traffic between the SNs and to the outside world. A switch connects a redistribution point or computational nodes. The connections of the processors are static but their utilization is changed. The SNC interconnection network is local. The interconnection between SNCs is provided through public Internet.

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 initiation 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 of a call is not available before the call is submitted, 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.

We define the provider cost model by considering a function that depends on the number of VMs and their running time. We denote the number of billing hours in SNC by $\overline{m}_i = \int_{t=0}^{C_{max}} k_i(t) \cdot m_i \, dt$ and run in all SNC by $\overline{m} = \sum_{i=1}^{m} \overline{m}_i$. In addition to $st_j, d_j, size_j$, the VM is characterized by $vmu_i(t)$ the utilization (load) of the VM_i at time t. VM hosts Asterisk running process that handles calls.

We introduce a quality reduction as a function of the VMs utilization (Figure 2), and consider this problem as a special case of dynamic bin packing (on-line and non-clairvoyant) with bi-objective optimization of provider cost and quality of service. Bins represents VMs, and the items height define the call contribution to the VM utilization. This approach allows to adapt to cloud uncertainties such as dynamic elasticity, performance changing, virtualization, loosely coupling application to the infrastructure, parameters such as an effective processor speed, number of running virtual machines and actual bandwidth, among many others.

4.1 Methodology of analysis

To choose a good strategy, we perform an analysis based on the degradation methodology proposed in [13], and applied for scheduling in [3, 18, 19]. It shows how the metric generated by our algorithms gets closer to the best found solution as $(\gamma - 1) \cdot 100$, with $\gamma = \frac{strategy\ metric\ value}{best\ found\ metric\ value}$. The goal of bi-objective analysis is to obtain a set of compromise solutions that represents a good approximation to the Pareto front. A solution is Pareto optimal if no other solution improves it in terms of all objective functions.

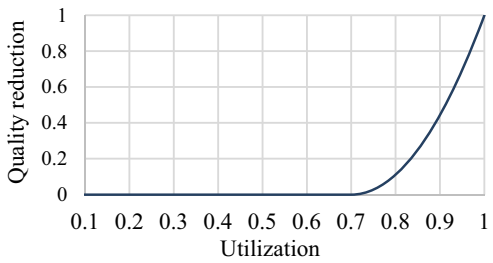

Figure 2. Voice quality reduction versus processor load (utilization).

5. 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. The classic bin packing problem concerns placing items of arbitrary height into a minimum number of bins with fixed capacity (of one-dimensional space) efficiently. Bin-packing is a very active area of research in the algorithms and operations research communities.

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.

5.1 Call allocation with startup time delay

We focus on the call allocation with the VM startup time delay. Figure 3 shows an example of call allocation with threshold of 70% of utilization, three VMs (blue rectangle) with a rental time (width) and a utilization (height).

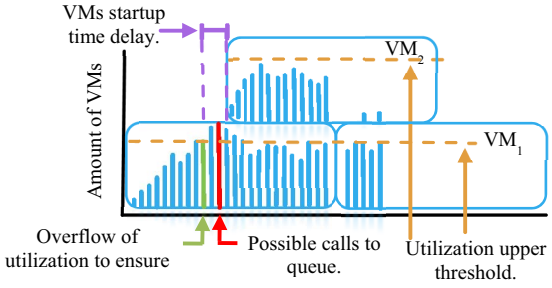

Figure 3. Calls allocation with QoS guaranteed and startup time delay.

A new VM is requested when the utilization is over the threshold (green line). During VM StUp (gap of purple lines), old VM continues call processing with utilization more than 70% and reduced QoS. The worst case appears when the current VM does not have enough resources to process arriving calls (red line). In

this case, the system puts the calls into a queue, waiting for available resources.

Table 3 briefly describes our call allocation strategies. They are grouped by the type and amount of information used for allocation decision: (1) knowledge-free (KF), with no information about applications and resources; (2) utilization-aware (UA) with CPU utilization information; and (3) time-aware (TA) with VM rental time information.

Table 3. Call allocation strategies.

		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.
TA	MaxFTFit	Allocates job *j* to VM with farthest finish time.
	MidFTFit	Allocates job *j* to VM with finish time between farthest and closest.
	MinFTFit	Allocates job *j* to VM with closest finish time.

6. EXPERIMENTAL SETUP

All experiments are performed using standard trace based simulator CloudSim [15], a framework for simulation of cloud computing infrastructures and services. We extend it by our algorithms, support of dynamic calls arrival, VM startup delays, and statistical analysis.

6.1 Workload

We use traces of real VoIP service [16], and Standard Workload Format with four additional fields to process the calls. The workload includes a set of phone calls registered by the VoIP system in the Call–Detail–Record (CDR) database with the following information: Index of the call; ID of the user who makes the call; IP of the phone where the call is placed from; IP of the local phone; Destination of the call; Destination country code; Destination country name; Telecommunications service provider; Beginning of the call (timestamp); Duration of the call (in seconds); Duration of a paid call; Cost per minute; etc. Number of calls per week day and call duration are presented in Table 4 and Table 5. The histogram of the number of calls per hour during a day is typical for business clients with two peaks in 10-12 and 14-16 hours.

Table 4. Number of calls per week 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 5. 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
5 - 6	15,075
6 - 7	10,009
7 - 8	7,256
8 - 9	5,536
9 - 10	4,202
…	…
19 - 20	721

7. 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 evaluate eight strategies: BFit, FFit, MaxFTFit, MidFTFit, MinFTFit, Rand, RR, WFit. In order to evaluate their robustness, we incorporate eight StUp delays: 0, 45, 90, 135, 180, 225, 270, 315 sec. as test cases.

Figure 4 shows the billing hours (BH) degradation versus StUps. We observe that strategies with better performance are BFit and FFit, and the worst strategies are MinFTFit and WFit.

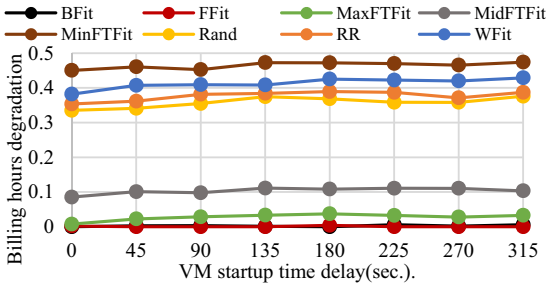

Figure 4. Billing hours degradation (time delay).

We see that our scheduling strategies tend to be robust with respect to BH. The StUp does not affect considerably the number of BHs (for all strategies).

Figure 5 displays the quality reduction degradation. We see that the strategies with better performance are WFit and RR, and the worst strategies are BFit, FFit and MaxFTFit. However, that difference is small. The worst degradation 0.639×10^{-3} is produced by BFit with StUp 315 sec. Similarly, the average Calls to Queue (CQ), see Figure 6, is about 4 for MaxFTFit in a day during 30 days (the worst strategy) with StUp equals to 315 sec.

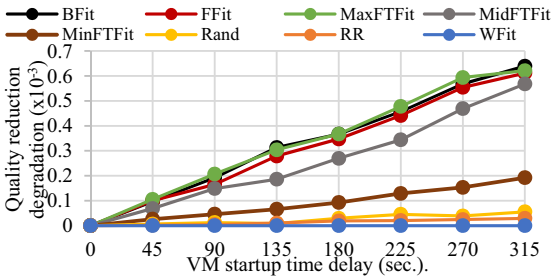

Figure 5. Quality reduction degradation.

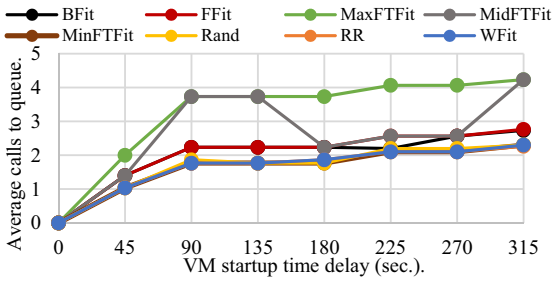

Figure 6. Average calls to queue.

The bi-objective solution space covers a range of values of BH degradation from 0 to 0.5, whereas values of QR degradation are in the range from 0 to 0.0008. We conclude that our strategy can reduce provider cost about 40-50% with only 0.08% of quality degradations. They are robust and address problems of uncertainty due to VM startup time delays.

8. CONCLUSION

In this paper, we formulate and study scheduling problems addressing cloud VoIP service with VM startup time delays. We define bi-objective model with provider cost, contributed by billing hours for used VMs, and quality of service, affected by call processing, optimization criteria.

We analyze eight on-line non-clairvoyant scheduling strategies on real data based on one month of the MIXvoip company service to deal with realistic VoIP cloud environments. All strategies have a high quality of service putting 0.1% of calls to the waiting queue during one month with 0.08% of quality reduction in the worst case. Moreover, they have the low variation in billing hours even with high dispersion of VM startup time delays.

We show that the proposed strategies outperform known ones in terms of quality of service and provider cost including those currently in use. We also evaluate their robustness in face of uncertainty of VM startup time delays variations. The evaluations demonstrate their potential benefits and stability with respect to handling call arrival rates and startup time delays variation. However, further study is required to assess their actual performance and effectiveness in a real domain. This will be the subject of future work.

9. ACKNOWLEDGMENTS

Part of the work was supported by CONACYT, México, grant no. 178415. The work of A. Drozdov is partially supported by the Ministry of Education and Science of Russian Federation under contracts RFMEFI58214X0003 and 02.G25.31.0061/12/02/2013.

10. REFERENCES

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

[2] 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.

[3] 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.

[4] M. Mao and M. Humphrey. A performance study on the vm startup time in the cloud. In Cloud Computing (CLOUD), IEEE 5th International Conference on (pp. 423-430), 2012.

[5] K. Razavi, L. Razorea, and T. Kielmann. Reducing vm startup time and storage costs by vm image content consolidation. In Euro-Par 2013: Parallel Processing Workshops (pp. 75-84). Springer Berlin Heidelberg, 2013.

[6] 3CX Phone System and ATOM N270 Processor Benchmarking. http://www.3cx.com/blog/voip-howto/atom-processor-n270-benchmarking , accessed September 20, 2016.

[7] A. Montazerolghaem, S. Shekofteh, M. Yaghmaee, and M. Naghibzadeh. A load scheduler for SIP proxy servers: design, implementation and evaluation of a history weighted window approach. Int. J. Commun. Syst. (2015).

[8] P. Montoro, and E. Casilari. A Comparative Study of VoIP Standards with Asterisk. In Fourth International Conference on Digital Telecommunications, 2009. ICDT '09 (pp. 1–6).

[9] 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 (in press).

[10] W. Song, Z. Xiao, Q. Chen, and H. Luo. Adaptive resource provisioning for the cloud using online bin packing. Computers, IEEE Transactions on. 63(11):2647-2660, November 2014.

[11] A.Wolke, B. Tsend-Ayush, C. Pfeiffer, and M. Bichler. More than bin packing: Dynamic resource allocation strategies in cloud data centers. Information Systems, 52, pp.83-95, September 2015.

[12] Y. Li, X. Tang, and W. Cai. Dynamic bin packing for on-demand cloud resource allocation. Parallel and Distributed Systems, IEEE Transactions on. 27(1):157-170. January 2016.

[13] D. Tsafrir, Y. Etsion, and D. Feitelson. "Backfilling using system-generated predictions rather than user runtime estimates". IEEE Transactions on Parallel and Distributed Systems 18(6): 789-803, 2007.

[14] http://blog.cloud66.com/ready-steady-go-the-speed-of-vm-creation-and-ssh-key-access-on-aws-digitalocean-linode-vexxhost-google-cloud-rackspace-and-microsoft-azure/, accessed September 20, 2016.

[15] CloudSim: A framework for modeling and simulation of Cloud Computing infrastructures and services. http://www.cloudbus.org/cloudsim/ , accessed September 20, 2016.

[16] 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. . The Fourth IEEE International Workshop on Cloud Computing Systems, Networks, and Applications 2015 (CCSNA'15). In conjunction with IEEE Global Communications Conference (GLOBECOM 2015) San Diego, CA, USA, 6-10 December, 2015.

[17] https://www.mixvoip.com/, accessed September 20, 2016.

[18] 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.

[19] A. Tchernykh, L. Lozano, U. Schwiegelshohn, P. Bouvry, J. E. Pecero, S. Nesmachnow, A. Yu. Drozdov. Online Bi-Objective Scheduling for IaaS Clouds with Ensuring Quality of Service. Journal of Grid Computing, Springer-Verlag, p. 1-18, 201.