Hindawi Publishing Corporation
Scientific Programming
Volume 2016, Article ID 5706790, 14 pages
http://dx.doi.org/10.1155/2016/5706790

Research Article

Biobjective VoIP Service Management in Cloud Infrastructure

Jorge M. Cortés-Mendoza,[1] Andrei Tchernykh,[1] Fermin A. Armenta-Cano,[1] Pascal Bouvry,[2] Alexander Yu. Drozdov,[3] and Loic Didelot[4]

[1] *CICESE Research Center, Ensenada, BC, Mexico*
[2] *University of Luxembourg, Luxembourg City, Luxembourg*
[3] *Moscow Institute of Physics and Technology, Moscow, Russia*
[4] *MIXvoip S.A., Steinsel, Luxembourg*

Correspondence should be addressed to Andrei Tchernykh; chernykh@cicese.mx

Received 22 January 2016; Revised 3 July 2016; Accepted 16 August 2016

Academic Editor: Ligang He

Voice over Internet Protocol (VoIP) allows communication of voice and/or data over the internet in less expensive and reliable manner than traditional ISDN systems. This solution typically allows flexible interconnection between organization and companies on any domains. Cloud VoIP solutions can offer even cheaper and scalable service when virtualized telephone infrastructure is used in the most efficient way. Scheduling and load balancing algorithms are fundamental parts of this approach. Unfortunately, VoIP scheduling techniques do not take into account uncertainty in dynamic and unpredictable cloud environments. In this paper, we formulate the problem of scheduling of VoIP services in distributed cloud environments and propose a new model for biobjective optimization. We consider the special case of the on-line nonclairvoyant 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.

1. Introduction

Voice over Internet Protocol (VoIP) has now become the most popular technology to communicate for long distance calling and is adopted all over the world. Together with general aspects of quality of service (QoS) of the Internet and other networks, like transmission rates, error rates, and other characteristics, VoIP adds new requirements: voice quality, service response time, throughput, loss, interrupts, jitter, latency, resource utilization, and so on. Hypervisor-level scheduling, traffic control, dynamic resource provisioning, and so forth are issues to address for the VoIP providers to ensure QoS and successful end-to-end business solution.

Effective VoIP scheduling involves many important issues: load estimation and prediction, performance analysis, system stability, call resource requirements estimation, routing, bandwidth limitation, resource selection for call allocation, and so forth [1–3].

Businesses provided VoIP systems are always looking for a way to cut down costs. Beloglazov et al. 2012 [4] consider efficiency of resource management deployed on the infrastructure and applications running on the system. One of the ways to reduce a cost is to avoid provisioning of more resources than required by users and QoS.

Cloud VoIP (CVoIP) solutions can offer even cheaper and scalable service by using virtualized telephone infrastructure in the efficient way. However, Tchernykh et al., 2015 [5], show that virtualization in cloud computing adds other complexity dimensions to the problem in terms of parameter variation, system uncertainty, dynamic consolidation of the virtual machines (VMs), and their migrations.

In this paper, we continue study presented by Cortés-Mendoza et al., 2015 [3], where we introduce a VoIP optimization model and study five call allocation strategies. We describe and analyze a model for cloud VoIP services focusing on two important aspects: QoS and VM utilization. We

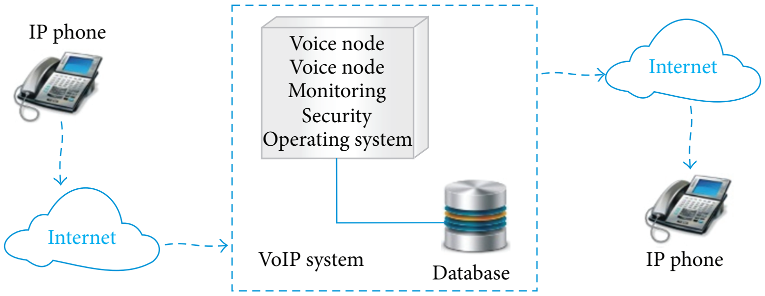


FIGURE 1: VoIP architecture.

take into account two main beneficiaries of the optimization technology: VoIP provider that runs its software on the cloud and end users.

While voice quality in real VoIPs is often seriously affected by the signaling overhead, end-to-end delay, jitter, packet loss, compression technique, hypervisor-level scheduling, and so forth, we restrict ourselves to voice quality affected by CPU usage during call processing. We believe that the focus in our model is reasonable and representative for real installations and applications.

In this paper, we provide a range of monoobjective and biobjective optimization solutions considering billing hours for VM running in a cloud and voice quality. We conduct the comprehensive simulation on real data and show that our scheduling strategies can provide a good compromise between saving money and voice quality. The paper makes the following contributions:

(i) We propose a set of on-line dynamic nonclairvoyant scheduling strategies to deal with VoIP calls in cloud environments. These strategies cover a wide domain of the VoIP biobjective problem, so that VoIP providers can select specific strategies depending on the goals.

(ii) We propose a novel function to ensure the VoIP QoS. This function considers the CPU utilization as a mean to evaluate the voice quality reduction.

(iii) We validate twenty strategies and evaluate their performance by comprehensive simulation analysis on real workload of the MIXvoip provider [6].

The paper is structured as follows. The next section briefly discusses VoIP service considering underlined infrastructure and software. Section 3 reviews related works. Section 4 presents several factors that have an impact on the QoS and provider cost. Section 5 provides the problem definition and corresponding model. Section 6 describes methodology of the analysis. Section 7 describes approaches for VoIP call allocation and corresponding algorithms. Section 8 describes our experimental setup, workload, and studied scenarios. Section 9 presents experimental analysis of the provider cost when quality of service is guaranteed. Section 10 analyzes a general biobjective problem. Section 11 concludes the paper by presenting the main contribution of the work and future research directions.

2. Internet Telephony

The Internet telephony VoIP refers to the provisioning of voice communication services over the Internet rather than via the traditional telephone ISDN network (ISDN (Integrated Services Digital Network) is a set of standards for digital transmission over ordinary telephone copper wire). One of the reasons of its wide acceptance is significant reduction of calling rates.

The scalability requires the service availability all the time for any number of users. To deal with increasing number of clients, providers may invest in a large infrastructure to avoid loss of calls (hence, users). In the case of overprovisioning, the infrastructure is underutilized most of the time.

The clients connect to a voice server, which is the main part of the VoIP telephony system (Figure 1). The voice nodes communicate with the database in the system, where all the users are registered, and calls are recorded with details such as destination and duration. They provide software to emulate a telephone exchange, gateways, interconnection switches, session controllers, firewall, and so forth.

The voice nodes handle calls with different features such as voicemail, call forwarding, music on hold, and conference calls depending on customers. They provide signaling, voice signal digitization, encoding, and so forth. In order to use VoIP services, an Internet connection and an IP hard-phone or soft-phone are needed.

Traditional VoIP solutions are not scalable. Drawbacks arise when the hardware reaches its maximum capacity. To scale it, it is necessary to increase or replace existing hardware. Overprovisioning and, hence, cost overrunning are 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 CVoIP can further reduce costs, add new features and capabilities, provide easier implementations, and integrate services that are dynamically scalable. Other benefits include data transfer availability, integrity, and security.

Cloud-based VoIP solutions allow reducing an importance of a Build-To-Peak approach. The virtual infrastructure can be easily scalable.

In this solution, the voice nodes are operated by VMs. Distributed cloud-based VoIP architecture assumes that voice nodes are distributed geographically; hence, they are grouped in different locations (data centers). To deploy and effectively

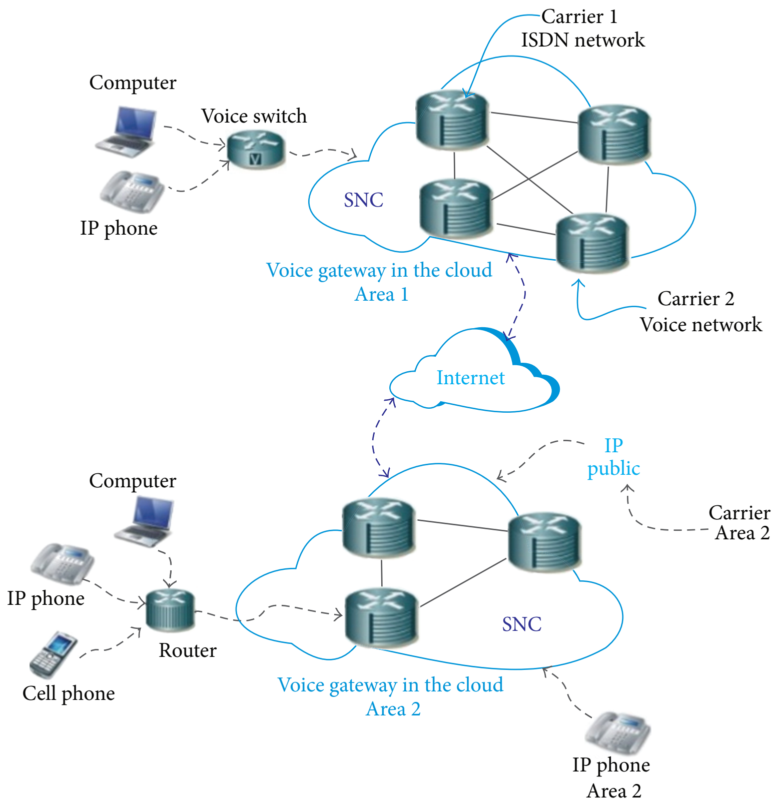


Figure 2: SNC deployment [3].

manage telephones via clouds, different characteristics need to be improved. The most important is the utilization of the virtual infrastructure and voice quality.

The advantage of cloud-based solution is in increased scalability and low cost. However, it has several unsolved problems. To optimize the overall system performance and reduce provider cost, the VM utilization has to be high, but it reduces quality of the calls (see Section 4). 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.

The most important cause of the load imbalance is the dynamic nature of the problem and system. The objective is to maximize VoIP system performance by minimizing the number of processing units without overloading them. It can improve the provider cost and guarantee QoS.

2.1. Infrastructure. MIXvoip company [6] presents telephony combining cloud service with smart business telephony, VoIP, and other telephony services.

It developed the concept of the super node (SN) and super nodes cluster (SNC) to enrichment features for telephone exchanges (Figure 2).

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.

As shown in Figure 2, a user call is allocated to the nearest SN in his area. This deployment allows providing services near ISDN quality in a public IP network.

2.2. Software. Asterisk is the most known Private Branch Exchange (PBX) software that includes all of the components necessary to build scalable phone systems (see Madsen et al. 2011) [10]. It allows making and processing calls and connecting to other telephone services, such as the public switched telephone network (PSTN) and VoIP services. It is a framework for building multiprotocol, real-time communication solutions providing a powerful control over call activity.

Delivering information and transferring data are based on protocols, such as Session Initiation Protocol (SIP) and Real Time Protocol (RTP). SIP is the protocol for signaling, establishing presence, locating users, setting, modifying, and tearing down sessions between end-devices. It is used for controlling communication sessions such as voice and video calls over IP networks. The media transportation is provided via RTP. Codecs are used for converting the voice portion of a call in audio packets to transmit over RTP streams.

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

3. Related Work

Different techniques have been introduced in recent years in order to overcome the challenges of VoIP. However, VoIP scheduling for QoS and provider cost optimization are still insufficiently studied.

So, [11] (2011) analyzes dynamic scheduling and persistent scheduling for VoIP services in wireless orthogonal frequency division multiple access systems. The author develops analytical and simulation models to evaluate the performance of VoIP services in terms of the average throughput and the signaling overhead according to the scheduling schemes.

Lee et al. (2006) [12] analyze the performance of three scheduling algorithms (Unsolicited Grant Service (UGS), real-time Polling Service (rtPS), and extended real-time Polling Service (ertPS)) for IEEE 802.16e standard covering mobile broadband wireless systems. The authors use simulation to show that ertPS algorithm with Enhanced Variable Rate Codec (EVRC) and silence suppression can support more calls compared with the UGS and rtPS algorithms, on 21% and 35%, respectively.

Folke et al. (2007) [13] analyze four scheduling algorithms: Proportional Fair (PF), Maximum Rate (MR), MR with the minimum bit rate ($\mathrm{MR}_{\mathrm{min}}$), and MR with strict delay ($\mathrm{MR}_{\mathrm{delay}}$) for mix of conversational (VoIP) traffic and interactive (web) traffic. All strategies are tested with varying scheduling delay budgets and loads. The authors show that a scheduler that gradually increases the VoIP priority and considers the user's current possible rate performs well. However, a more drastic increase in VoIP priority is needed when the delay budget is short. Furthermore, attempting to uphold quality for both VoIP and web traffic makes the system sensitive to overload situations.

Bayer et al. (2010) [14] analyze on demand scheduling, uncoordinated resource coordination scheme, coordinated resource coordination scheme, and VoIP aware resource coordination scheme for TDMA based access control in mesh networks in the IEEE 802.16 standard. The authors show that the mesh networks are able to support VoIP with good quality when a persistence scheduling is applied. Compared to other resource coordination schemes the VoIP aware scheduler significantly increases the number of supported calls.

Wu et al. (2014) [15] present a real-time scheduling framework in virtualized environment that considers real-time constraints of applications. The authors propose a mechanism called multicore dynamic partitioning to divide physical CPUs into two pools dynamically according to the scheduling parameters of VMs. They use global earliest deadline first strategy to schedule calls on VMs, with an Asterisk instance, to improve CPU usage and guarantee the call quality.

Mazalek et al. (2015) [16] study the impact of the IPSec encryption on the CPU utilization, bandwidth, and voice quality. The authors show a significant effect of voice payload period on the CPU utilization and bandwidth. They save up to 40–60% of bandwidth when the period is chosen properly. The call quality, expressed by mean opinion score (MOS) scale, remains almost constant up to the moment, when the CPU utilization is close to 80%.

Table 1: Processor utilization for 1 call without transcoding [7].

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%

Costa et al. (2015) [17] show that Asterisk PBX server is able to provide VoIP communication capabilities with an acceptable MOS quality. The authors use the blocking probability metric to measure the capacity of the VoIP server and MOS to assess the quality of the voice calls. The experimental results show that the Asterisk PBX using SIP effectively handled more than 160 concurrent voice calls with a blocking probability below 5%.

Cheng et al. (2015) [18] present and compare the SRT-Xen scheduler with other four schedulers (Credit, Credit2, rtglobal, and rtpartition). They focus on real-time-friendly scheduling to improve the management of the virtual CPUs' queueing. They use an instance of Asterisk to evaluate the performance of the strategies and speech quality. SRT-Xen achieves at least 13.61% more sessions with perceptual evaluation of speech quality >4.

4. VoIP Quality of Service

4.1. Utilization. Calls have different impact on the processor utilization depending on the operations performed by Asterisk, when the calls are being established. If transcoding operations are performed, the utilization is higher than that 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 1 shows processor utilization for call without transcoding presented by Montoro and Casilari (2009) [7].

VoIP gateways support a larger number of codecs and DSP modules (Digital Signal Processing): G.711, GSM, LPC10, Speex. G.711 A-law and U-law PCM, G.726 ADPCM, G.728 LD-CELP, G.729 CS-ACELP, G.729a CS-ACELP, G.729 Annex-B, G.729a Annex-B, G.723.1 MP-MLQ, G.723.1 ACELP, G.723.1 Annex-A MP-MLQ, G.723.1 Annex-A ACELP, and so forth. Some codec compression techniques require more processing power than others. Examples of the compression method are presented by Cisco [9]:

PCM: Pulse Code Modulation

ADPCM: Adaptive Differential Pulse Code Modulation

LDCELP: Low-Delay Code Excited Linear Prediction

ACELP: Algebraic-Code-Excited Linear Prediction

MP-MLQ: Multi-Pulse, Multi-Level Quantization

CS-ACELP: Conjugate-Structure Algebraic-Code-Excited Linear Prediction.

In [8], the authors present results of the benchmark test that includes stress testing of queue calls, VoIP calls,

Table 2: Queue calls [8].

Activity test	Jitters	CPU usage	Simultaneous calls	CPU usage per 1 call
5 calls to queue	None	14%	10	1.40%
10 calls to queue	None	18%	20	0.90%
15 calls to queue	None	28%	30	0.93%
20 calls to queue	None	36%	40	0.90%
30 calls to queue	None	67%	60	1.11%
40 calls to queue	None	84%	80	1.05%

Table 3: MOS score [9].

Compression	Bit rate (kbps)	MOS score	Compression delay (ms)
G.711 PCM	64	4.1	0.75
G.726 ADPCM	32	3.85	1
G.728 LD-CELP	16	3.61	3 to 5
G.729 CS-ACELP	8	3.92	10
G.729 × 2 encodings	8	3.27	10
G.729 × 3 encodings	8	2.68	10
G.729a CS-ACELP	8	3.7	10
G.723.1 MP-MLQ	6.3	3.9	30
G.723.1 ACELP	5.3	3.65	30

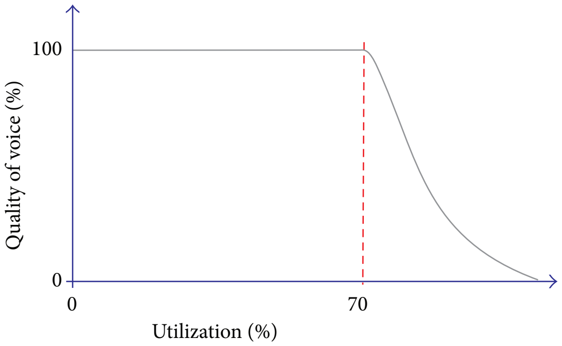


Figure 3: Voice quality versus processor load (utilization).

and extension to extension calls. Queuing calls are used by call centers that prefer to answer to the incoming calls automatically and place them in a queue instead of rejecting them. Queuing allows the acceptance of more calls into the system than existing extensions or agents capable of answering them. While on hold, the callers receive different announcements (position in the queue) followed by music.

Considering Tables 1 and 2, we conclude that CPU can process from 70 to 500 calls with 100% of utilization.

4.2. Quality of Service. The VoIP QoS is determined by the quality of voice, transit time of packets across the Internet, queuing delays at the routers, packet travel time from source to destination, jitter as deviations of the packet interarrivals, packet loss, call setup and tear downtime, and so forth.

The quality of voice is a subjective response of the listener. A common benchmark used to determine the quality of sound produced by specific codecs is the Mean Opinion Score (MOS). Listeners judge the quality of a voice sample that corresponds to a particular codec on a scale of 1 (bad) to 5 (excellent). The scores are averaged to provide the MOS for that sample. Table 3 shows the relationship between codecs and MOS scores presented by Cisco [9].

Cortés-Mendoza et al. (2015) propose using processor utilization in order to ensure QoS. Each codec provides a certain quality of voice only if processor utilization is low enough. Theoretically, processor utilization of 100% provides the best expected performance. However, in [8], the authors show that 20 calls via a VoIP provider produced no jitters; CPU usage in total was 19%. With increasing number of calls up to 90 and utilization up to 85%, CPU cannot be able to handle the stress anymore and jitters and broken audio symptoms will appear.

Considering different types of calls with different codecs, we use a threshold of 70% to ensure a high voice quality.

Figure 3 shows that the voice quality is reduced fast when the processor utilization exceed 70%.

4.3. VoIP Provider Costs. VoIP provider costs are primarily tied to their assets and the maintenance of these assets. For example, providers have an infrastructure that needs to be powered and cooled. It has storage arrays containing storage disks, and these arrays are connected to chassis which are all housed. So, major provider costs can be categorized as servers cost (computing, storage, software, and associated VoIP components), infrastructure cost (power distribution, cooling equipment, space for facilities, etc.), operational cost (energy, cooling, etc.), and network cost (links and transit equipment). A number of other costs exist.

To offer competitive prices to prospective customers VoIP providers should optimize the process. Inefficient resource management has a direct negative effect on performance and cost.

Virtualization technologies allow creating VoIP virtual servers, which can then be hosted in data centers and rented out (leased) on a subscription basis to any scale.

In a typical cloud scenario, a VoIP provider has the choice between different resources that are available on demand from cloud providers with certain service guarantees. These service levels are mainly distinguished by the amount of computing power guaranteed to receive within a requested time and a cost per unit of execution time the VoIP provider has to pay. This cost depends on the type of requested computing resources, for instance, VMs with different performance.

In order to evaluate the provider cost for cloud solution, we use a metric that is useful for systems with VM. It must allow the provider to measure the cost of the system in terms of number of demanded VMs and time of their using.

In this paper, two criteria are considered for the model: the billing hours for VMs to provide a service and their utilization to estimate quality of service.

In the first scenario, we consider single-objective optimization problem minimizing the total cost of VMs with given restrictions. In order to ensure good QoS, the utilization of the VMs is kept under the certain threshold (e.g., 70%).

In the second scenario, we consider the biobjective optimization approach that is not restricted to find a unique solution but a set of solutions known as a Pareto optimal set. In this case, we minimize two conflicting objectives: the cost of VMs and QoS degradation. A tradeoff between the two objectives depends on the VoIP provider's preference.

5. Model

We address the model for VoIP in distributed cloud environment with heterogeneity of resources with different number of servers, execution speed, amount of memory, bandwidth, and so forth.

Let us consider that VoIP cloud infrastructure consists of m heterogeneous super node clusters SNCs : $\mathrm{SNC}_1, \mathrm{SNC}_2, \ldots, \mathrm{SNC}_m$ with relative speeds $s_1, s_2, \ldots . s_m$. Each SNC_i, for all $i = 1, \ldots, m$, consists of m_i SNs. Each SN_k^i, for all $k = 1, \ldots, m_i$, runs $k_i(t)$ VM at time t. We assume that VMs of one SNC are identical and have the same processing capacity.

The SNC contains a set of routers and switches that transport traffic between the SNs and 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 architecture is local. The interconnection between SNCs is provided through public Internet.

We consider n independent calls or jobs $J_1, J_2, \ldots, J_n$ that must be scheduled on set of SNCs. The job J_j is described by a tuple $\{r_j, p_j, u_j\}$ that consists of its release date $r_j \geq 0$, duration p_j (lifespan), and contribution to the processor utilization u_j. The release time of a job is not available before the job is submitted, and its duration is unknown until the job has been completed. The utilization is a constant for a given job that depends on the used codec and is normalized for the slowest machine.

In order to evaluate the system performance, we use metrics that are useful for VoIP systems, where traditional measures such as makespan, throughput, and response time become irrelevant.

For this kind of systems, the metrics must allow the provider to measure the performance of the system in terms of financial attraction which helps him to assure benefits as well as user satisfaction for the service.

Two criteria are considered in the analysis: Minimization of the service provider cost and minimization of the quality of service degradation.

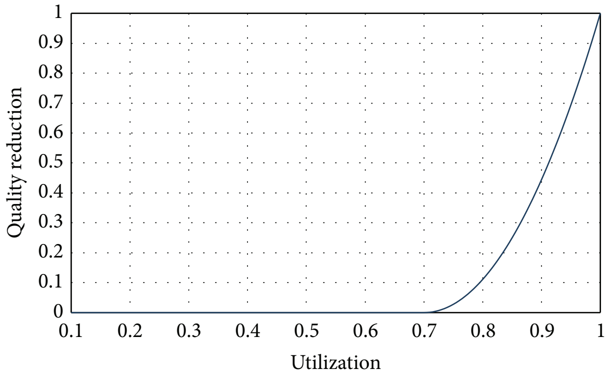


FIGURE 4: Voice quality reduction versus processor load (utilization).

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_i by $\overline{m}_i = \int_{t=0}^{C_{\max}} k_i(t) \cdot m_i dt$ and run in m SNC by $\overline{m} = \sum_{i=1}^{m} \overline{m}_i$. The VM is described by a tuple $\{\mathrm{vmu}_i(t)\}$, where $\mathrm{vmu}_i(t)$ is the utilization (load) of VM_i at time t. VM hosts Asterisk running process that handles calls.

As an optimization criterion, we introduce a quality reduction function based on the VMs utilization (Figure 4).

6. Methodology of Analysis

To derive bounds of the provider cost, we consider two scenarios. The maximum cost can be archived if provider guarantees the voice quality with quality reduction equal to 0 by setting the utilization threshold to 70%. Then, in the second scenario, we realize a biobjective analysis, where no threshold is used (100% of utilization is allowed), to study the compromise between the provider cost and voice quality reduction.

6.1. Degradation in Performance. To choose a good strategy, we perform an analysis based on the degradation methodology proposed in [19] and applied for scheduling in [20]. It shows how the metric generated by our algorithms gets closer to the best found solution.

The analysis is conducted as follows. First, we evaluate the degradation in performance (relative error) of each strategy relatively to the best performing strategy as follows:

$$(\gamma - 1) \cdot 100, \quad \text{with } \gamma = \frac{\text{strategy metric value}}{\text{best found metric value}}. \qquad (1)$$

Then, we average these values for all scenarios and rank the strategies. The best strategy with the lowest average performance degradation has rank 1. Note that we try to identify strategies, which perform reliably well in different scenarios; that is, we try to find a compromise that considers all of our test cases. For example, the rank of the strategy could not be the same for any of the scenarios individually.

6.2. Biobjective Analysis. In multiobjective optimization, one solution can represent the best solution concerning provider cost, while another solution could be the best one concerning the QoS. The goal is to choose the most adequate solution and obtain a set of compromise solutions which represents a good approximation to the Pareto front.

Two important characteristics of a good solution technique are convergence to the Pareto front and diversity to sample the front as fully as possible. A solution is Pareto optimal if no other solution improves it in terms of all objective functions. Any solution not belonging to the front can be considered of inferior quality to those that are included. The selection between the solutions included in the Pareto front depends on the system preference. If one objective is considered more important than the other one, then preference is given to those solutions that are near-optimal in the preferred objective, even if values of the secondary objective are not among the best obtained.

Often, results from multiobjective problems are compared via visual observation of the solution space. One of formal and statistical approaches uses a set coverage metric $\mathrm{SC}(A, B)$ that calculates the proportion of solutions in B, which are dominated by solutions in A:

$$\mathrm{SC}(A, B) = \frac{\{|b \in B| ; \exists a \in A; a \leq b\}}{|B|}. \qquad (2)$$

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

7. Call Allocation

In our model, CPU utilization is a key performance metric for VoIP quality of service measurement. It can be used to track QoS reductions, when it increases above the certain threshold, or improvement, when it is below, and it is useful for VoIP QoS problem studying.

The concept of VM utilization used in our study is simple. Assume that the VM is allocated on a single core processor of 2.0 GHz. VM utilization in this scenario is the percentage of time the processor spends doing VM work (as opposed to being idle). If the processor does 1 billion cycles worth of VM work in a second, it is 50% utilized for that second.

In general, monitoring CPU utilization, where VM is running, is straightforward: from a single percentage of CPU utilization to the more in-depth statistics. We can also gain a bit of insight into how the CPU is being used. To gain more detailed knowledge regarding VM utilization, we must examine all details of the VM parameters, software installed, and hardware of a system.

There are a lot of factors that contribute to the processor utilization. In our case, we reduce ourselves to consider Asterisk running processes and calls.

The call allocation problem is similar to a well-known one-dimensional on-line bin-packing problem, the classical NP-hard optimization problem with high theoretical relevance and practical importance. Bin-packing concerns placing items of arbitrary height into a one-dimensional space (bins with fixed capacity) efficiently.

Bin-packing remains one of the classical difficult problems. Scientists have analyzed and studied this computational puzzle for decades, yet none have obtained an algorithm which derives the optimal solution in reasonable amount of time. We consider an on-line variant of the problem in which items are received one by one.

Bins represent VMs, and the items height defines the call contribution to the processor utilization. Before info about the next call is revealed, the scheduler needs to decide whether the call is packed in the currently available VMs or a new VM must be rented. The scheduler only knows the contribution of the call to the processor utilization u_j due to the used codec. All decisions have to be made without knowledge of duration of the call, call arrival rate, and so forth.

The principal novelty of this problem variation lies in the temporal existence of the items. After a call lifespan is reached, the VMs can free space for processing more calls, so the state of the VMs is determined not only by the decision maker during call allocations. Unlike the standard formulation, bins are always open and dynamic and even completely packed. Items in bins can be terminated (call termination) and utilization can be changed at any moment.

As mentioned in Section 6, we consider two scenarios. In the first scenario, the bin size is equal to 0.7 which corresponds to 70% of VM utilization, so that the quality reduction is zero. In the second scenario, the bin size is equal to 1 which corresponds to 100% of VM utilization, so that the quality reduction can appear.

On both scenarios, we do not sort the input items due to the fact that we face an on-line bin-packing problem. Instead, we can sort bins based on their utilization.

We study twenty strategies (Table 4), Rand, RR, FFit, Bfit, WFit, MaxFTFit, MidFTFit, MinFTFit, RR_05, RR_10, RR_15, Wfit_05, Wfit_10, Wfit_15, BFit_05, BFit_10, BFit_15, FFit_05, FFit_10, and FFit_15, and evaluate their performance with the real workload considering six months of the MIXvoip company service.

We categorize all strategies in four groups by the type and amount of information used for allocation decision (1) knowledge-free (KF), with no information about applications and resources; (2) utilization-awareness (UA) with CPU utilization information; (3) time-awareness (TA) with VM rental time information; and (4) time-awareness with CPU utilization information (TA + UA).

In our previous work, Cortés-Mendoza et al. (2015) [3] study three well-known bin-packing strategies adapted for the described problem, First-Fit (FFit), Best-Fit (Bfit), and Worst-Fit (Wfit), and two commonly used allocation strategies, Round Robin (RR) and Random (Rand).

The significant contribution of this paper compared with the previous work is that we analyze twenty strategies, consider different scenarios solving monoobjective and biobjective problems, and provide a deeper study of our

Table 4: 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 able 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
	RR_05 RR_10 RR_15	Allocates job j to VM that finishes not less than in 5, 10, and 15 minutes using the RR strategy
TA + UA	BFit_05 BFit_10 BFit_15 FFit_05 FFit_10 FFit_15 WFit_05 WFit_10 WFit_15	Allocates job j to VM that finishes not less than in 5, 10, and 15 minutes using the WFit, BFit, and FFit strategies

```
Input: Voice node list (VNlist) and call.
Output: Allocation of call in one voice node.
(1)  Sort VNlist by utilization on decreasing order.
(2)  assigned ← false
(3)  node_index ← 1
(4)  Do
(5)      node_voice ← get(VNlist, node_index)
(6)      Add call to node_voice
(7)      if utilization of node_voice <= 0.7 then
(8)          assigned ← true
(9)      else
(10)         remove call from node_voice
(11)         node_index ← node_index + 1
(12)     endif
(13) While (size of VNlist >= node_index and
(14)     assigned = false)
(15) If assigned = false then
(16)     Create new_node_voice
(17)     Add call to new_node_voice
(18)     Insert new_node_voice into VNlist
(19) Endif
```

Algorithm 1: Best fit (BFit).

```
Input: Voice node list (VNlist), time and threshold.
Output: A voice node list for processing call.
Create new_VNlist
For each node_voices on VNlist
    If time_end(node_voice) <= time + threshold
        Add node_voice to new_VNlist
endfor
return new_VNlist
```

Algorithm 2: Admissible VMs list (AVML).

algorithms performance taking into account billing hours and quality reduction.

Algorithm 1 describes the BFit strategy, where voice nodes in the list are sorted in decreasing order of their utilization. We use the term the voice node instead of VM to have coherence with the call allocation terminology.

Line (1) may be changed depending on the strategy of allocation. For example, the list is sorted in increasing order in WFit strategy, and this line is not used in FFit strategy [21].

The main idea of the time-aware approach is to allocate calls to VM taking into account the finishing time of rented hours.

The goal is to allocate calls to VM to reduce number of billing hours. We try to avoid next hour renting due to continuation of the call over the rented hour.

For instance, MaxFTFit schedules the call to VM with farthest away finish time. MinFTFit schedules the calls to the voice node with nearest finish time. It tries to use already running voice nodes. The objective of MidFTFit is not to allocate calls to VMs that are not in beginning or end of the rental time.

We also introduce the time-aware versions of RR and WFit strategies (RR_05, RR_10, RR_15, WFit_05, and WFit_10), where we do not allocate calls to VMs in which rented time is finished in certain threshold. By these thresholds, we try to avoid next hour renting due to continuation of this call over the rented hour. It could reduce the number of billing hours. We study three thresholds: 5, 10, and 15 minutes before the end of renting hour.

For these strategies, the algorithm has a new procedure, named AVML (Admissible VMs List) (see Algorithm 2).

8. Experimental Setup

8.1. Simulation Toolkit. All experiments are performed using the CloudSim [22], a framework for modeling and simulation of cloud computing infrastructures and services. It is a standard trace based simulator that is used to study cloud resource management problems. We have extended CloudSim to include our algorithms for call allocation, supporting dynamic calls arrival, updating the system parameters before scheduling decisions, using the utilization of the resources, dynamically creating VMs, and providing statistical analysis using the java (JDK 7u51) programming language.

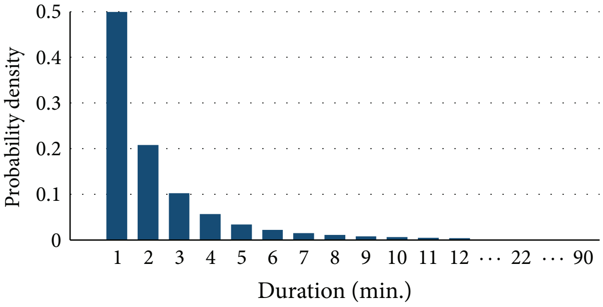


Figure 5: Call duration distribution.

Parameters are directly taken from traces of real VoIP service studied by Simionovici et al. (2015) [2]. We use SWF (Standard Workload Format) with four additional fields to process the calls.

8.2. Workload. The workload is a set of registered phone calls that have been registered by the VoIP system. It is recorded 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; and so forth.

Supported call statistics could include incoming/outgoing call attempts, whether successful or not, calls rejected or failed, number of calls whose connected time is less than the configured minimum call duration (MCD), number of calls losing more than the configured number of packets, number of calls encountering more than the configured amount of latency and jitter, calls disconnected, and so forth.

Total call distributions per hour and per day during six months (from 1 November 2014 to 17 April 2015) are presented in Cortés-Mendoza et al. (2015) [3].

They demonstrate typical behavior for business customers: two peak hours, 8–11 AM and 13–17 PM. Over a week, the traffic is high from Monday till Friday, while for weekends it decreases considerably.

Figure 5 shows the call duration distribution during the six months, which depends significantly on the clients (e.g., call centers, schools, and business companies). In our example, the duration of the majority of the calls is short (e.g., 1–5 minutes).

Dang et al. (2004) [23] showed that the call arrival process is fitted by a Poisson process and the call duration distribution by a generalized Pareto distribution with parameter values indicating finite variances. The authors tested a series of probability distributions and showed that the model agrees well with the data in high-density regions and also fits the low-density regions, known as tails of the distribution (Figure 5).

For the analysis, we use 24 workloads; each includes phone calls made during one week. Figure 6 shows mean number of calls per hour in a day during 24 weeks.

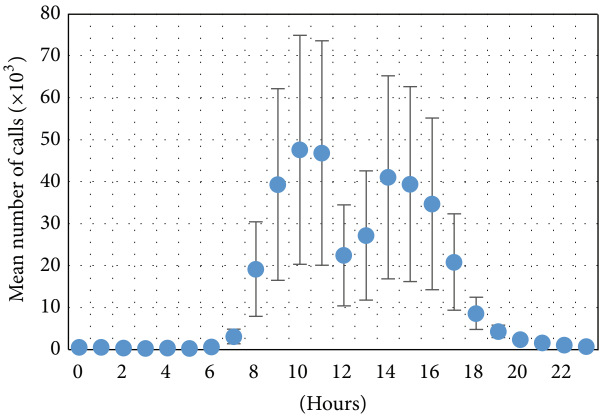


Figure 6: Mean number of calls per hour in a day during 24 weeks with the standard deviation.

During the weekends, the workload is low. It needs only one VM to process it. For this study, we removed the jobs of the weekends because they can be replaced with 24 billing hours per day. One VM is always running even with no calls.

8.3. Scenarios. In our scenarios, each VM runs an instance of Asterisk (voice node). The VMs are deployed by several super node voices (SNs) and all of them belong to one SNC. The VoIP providers rent the VMs by hours [24]; when the VM rental time is finished the VM can be turned off only if VM is not processing any calls; in other cases, this VM continues running for one more hour.

9. Scenario 1: Cost Analysis with Guaranteed Quality of Service

In the first scenario, we evaluate the provider cost generated by the twenty strategies: BFit, BFit_05, BFit_10, BFit_15, FFit, FFit_05, FFit_10, FFit_15, MaxFTFit, MidFTFit, MinFTFit, Rand, RR, RR_05, RR_10, RR_15, WFit, WFit_05, WFit_10, and WFit_15.

We use the utilization threshold as the constraint to guarantee the quality of service.

Figure 7 displays the number of billing hours during 24 weeks. We see that the workload is low during weeks 8 and 9, so that the difference of billing hours generated by strategies is about 50. In other weeks, the dispersion is higher up to 160 billing hours in week 5.

Table 5 shows the average number of billing hours during considered 24 weeks. BFit and FFit are shown to be the best strategies using 252.08 and 252.42 billing hours per week on average to deal with given workload. WFit and MinFTFit are worst ones with 351.08 and 363.96 billing hours, respectively.

RR_05, RR_10, and RR_15 strategies have a better performance than non-time-aware RR. Similarly, WFit_05, WFit_10, and WFit_15 are better than WFit. The difference between the best strategy, BFit, and worst one, MinFTFit, is about 111 billing hours per week on average.

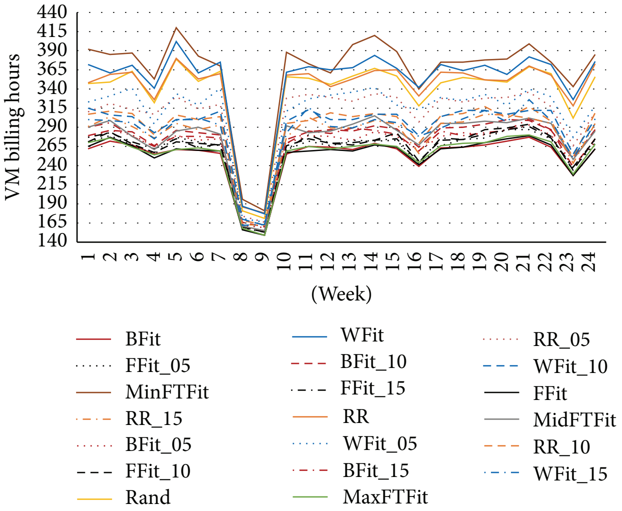


Figure 7: The number of billing hours during 24 weeks.

10. Scenario 2: Biobjective Analysis

10.1. Degradation. In multiobjective analysis, the problem can be simplified to a single objective problem through different methods of objective weighted aggregation. There are various ways to model preferences; for instance, they can be given explicitly to specify the importance of every criterion or a relative importance between criteria. This can be done by a definition of criteria weights or criteria ranked by their importance.

In this section, we perform a joint analysis of two metrics according to the mean degradation methodology described in Section 6.1.

First, we present the analysis of the billing hours for rented VMs and quality reduction separately. Then, we find the strategy that generates the best compromise between them.

In Table 6, we present the average degradation of billing hours, quality reduction, and their means. The last three columns of the table contain the ranking of each strategy regarding the provider cost, quality, and their means. Rank-BH is based on the billing hours' degradation. Rank-QR refers to the position in relation to the degradation of quality reduction. Rank is the position based on the averaging two rankings.

We see that the best strategy for the cost optimization is BFit which allocates calls based on best fit strategy, where we put the call into the fullest VM, which leaves the least utilization left over. However, it is the worst strategy for the voice quality. It tends to increase utilization and reduce quality.

The best strategy for the voice quality is WFit, where we put the call into the VM, which leaves most of utilization left over. It tends to underutilize VMs keeping the quality but increases VM number and renting cost.

A good compromise is MaxFTFit strategy that allocates the call to VM that finishes his hour far away.

Table 5: Average weekly billing hours.

Rank	Strategy	VM billing hours
1	BFit	252.08
2	FFit	252.42
3	MaxFTFit	254.71
4	FFit_05	259.46
5	FFit_15	261.75
6	FFit_10	261.79
7	BFit_05	266.13
8	BFit_15	269.21
9	BFit_10	273.25
10	MidFTFit	276.08
11	RR_15	279.79
12	WFit_15	283.79
13	RR_10	289.00
14	WFit_10	290.42
15	RR_05	306.88
16	WFit_05	311.29
17	Rand	336.29
18	RR	340.46
19	WFit	351.08
20	MinFTFit	363.96

10.2. Solution Space and Pareto Fronts. To solve the general biobjective problem, we want to obtain a set of compromise solutions that represent a good approximation to the Pareto front. This is not formally the Pareto front as an exhaustive search of all possible solutions is not carried out but rather serves as a practical approximation of a Pareto front.

Figure 8 shows the solution sets for the twenty strategies obtained based on 109 days of workload. This two-dimensional solution space represents a feasible set of solutions that satisfy the problem constraints. Note that we address the problem of minimizing cost and maximizing the quality. For better representation, we convert it to the minimization of two criteria: degradations of both the cost and quality reduction.

The solution space covers a range of values of cost degradation from 0 to 0.65, whereas values of quality reduction degradation are in the range from 0 to 0.26.

We see that the solution space is divided in three groups located in right lower side, left lower side, and in the middle. BFit, FFit, and MaxFTFit are located in the lower right side being among the best solutions in terms of the billing hours. They outperform other strategies, like RR, which are in current use for VoIP service. WFit is located in the left side being among the best solutions in terms of quality reduction. The three versions of time-aware WFit (WFit_05, WFit_10, and WFit_15) have a good behavior.

WFit is the best for quality reduction degradation (QRD = 0). The range of the cost degradations is from 0.16 to 0.56. WFit_05 increases the QRD to 0.017 but reduces the cost up to 0.05. For WFit_10, the QRD increases from 0.017 to 0.06, but the cost reduces to 0.023.

Table 6: Degradation and ranking.

Strategy	Billing hours (BH)	Quality reduction (QR)	Mean	Rank BH	Rank QR	Rank
BFit	0.263	21.099	10.681	1	20	4
BFit_05	6.911	17.930	12.420	7	16	6
BFit_10	8.405	17.342	12.874	9	13	5
BFit_15	8.091	17.240	12.665	8	12	3
FFit	0.535	21.031	10.783	2	19	4
FFit_05	3.483	18.758	11.120	4	18	5
FFit_10	4.638	18.069	11.354	5	17	5
FFit_15	4.812	17.819	11.315	6	14	3
MaxFTFit	2.096	17.835	9.965	3	15	1
MidFTFit	10.536	16.442	13.489	10	11	4
MinFTFit	39.147	12.800	25.973	20	10	7
Rand	31.263	1.094	16.178	17	4	4
RR	32.681	0.732	16.707	18	2	3
RR_05	22.542	2.144	12.343	15	5	3
RR_10	15.388	4.666	10.027	13	7	3
RR_15	11.772	6.980	9.376	11	9	3
WFit	35.435	0.000	17.718	19	1	3
WFit_05	23.904	1.085	12.494	16	3	2
WFit_10	16.606	3.301	9.954	14	6	3
WFit_15	13.292	5.428	9.360	12	8	3

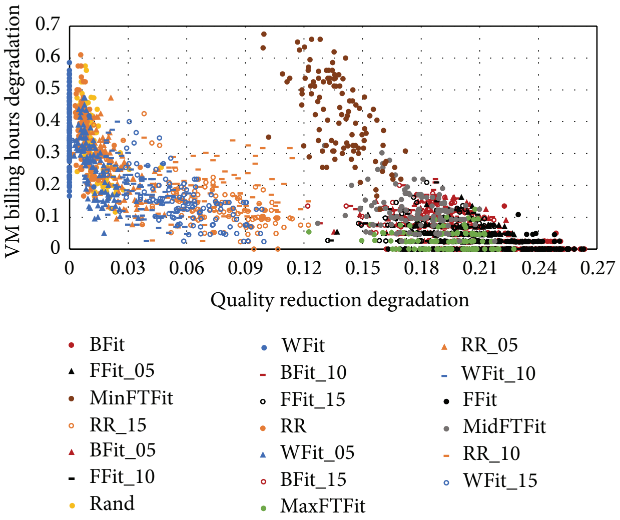


Figure 8: The solution space.

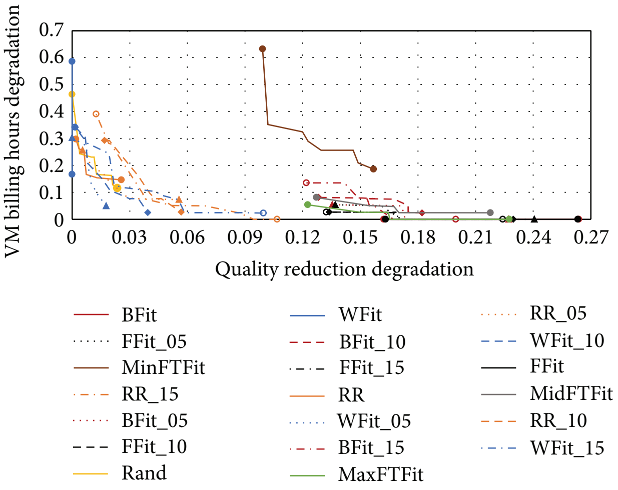


Figure 9: Pareto fronts.

Finally, WFit_15 has a wide range of solutions for QRD (from 0.019 to 0.099) but only 20% of its solutions are over the 20% of cost degradation.

WFit versions cover different sectors in the Pareto front, and they show the best compromise between both objectives for the twenty strategies.

The MaxFTFit solution space is in the same range for cost as Bfit and FFit. It overcomes the quality reduction of both strategies.

WFit_05, WFit_10, WFit_15, and MaxFTFit strategies cover better the solution space and Pareto front. They are good options for the VoIP providers. Figure 9 shows the twenty approximations of Pareto fronts generated by the studied strategies.

Using the set coverage metric, described in Section 6.2, two sets of nondominated solutions can be compared. The rows of Table 7 show the values SC(A, B) for the dominance of strategy A over strategy B. The columns indicate SC(B, A), that is, dominance of B over A. The last two columns show the average of SC(A, B) for row A over column B and ranking based on the average dominance. Similarly, the last two rows show average dominance of B over A and rank of the strategy in each column.

Table 7: Set coverage and ranking (days).

A											*B*											
	BFit	BFit_05	BFit_10	BFit_15	FFit	FFit_05	FFit_10	FFit_15	MaxFTFit	MidFTFit	MinFTFit	Rand	RR	RR_05	RR_10	RR_15	WFit	WFit_05	WFit_10	WFit_15	Mean	Rank
BFit	1	0	0	0	0.30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.065	18
BFit_05	0.07	1	0.06	0.02	0.08	0.17	0.05	0.01	0.01	0.01	0	0	0	0	0	0	0	0	0	0	0.074	17
BFit_10	0.03	0.17	1	0.07	0.02	0.11	0.10	0.05	0.02	0.02	0	0	0	0	0	0	0	0	0	0	0.079	16
BFit_15	0.03	0.23	0.28	1	0.05	0.10	0.11	0.10	0.03	0.01	0	0	0	0	0	0	0	0	0	0	0.096	14
FFit	0.21	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.061	19
FFit_05	0.33	0.06	0.04	0.02	0.30	1	0	0	0.01	0	0	0	0	0	0	0	0	0	0	0	0.088	15
FFit_10	0.18	0.17	0.09	0.06	0.18	0.38	1	0.06	0.04	0.03	0	0	0	0	0	0	0	0	0	0	0.109	11
FFit_15	0.21	0.29	0.19	0.12	0.21	0.33	0.34	1	0.03	0.06	0	0	0	0	0	0	0	0	0	0	0.139	8
MaxFTFit	0.43	0.42	0.28	0.24	0.46	0.60	0.47	0.38	1	0.09	0	0	0	0	0	0	0	0	0	0	0.218	2
MidFTFit	0.02	0.20	0.31	0.18	0	0.14	0.13	0.08	0.03	1	0	0	0	0	0	0	0	0	0	0	0.105	13
MinFTFit	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0.050	20
Rand	0	0	0	0	0	0	0	0	0	0.01	0.94	1	0.01	0.09	0.03	0	0	0.05	0.01	0.02	0.107	12
RR	0	0	0	0	0	0	0	0.01	0	0	0.90	0.52	1	0.10	0.02	0.03	0	0.07	0.00	0.02	0.133	10
RR_05	0	0.04	0.04	0.04	0	0	0	0.01	0.01	0.08	0.97	0.03	0.02	1	0.21	0.11	0.01	0.01	0.19	0.14	0.145	7
RR_10	0.01	0.13	0.20	0.17	0	0.06	0.06	0.07	0.04	0.28	0.99	0.02	0	0.02	1	0.31	0	0	0.01	0.21	0.178	6
RR_15	0.02	0.32	0.39	0.39	0.02	0.11	0.17	0.16	0.06	0.52	0.98	0	0	0.02	0.01	1	0	0	0	0.02	0.210	3
WFit	0	0	0	0	0	0	0	0	0	0	0.84	0.27	0.41	0.08	0.01	0.02	1	0.09	0.03	0.01	0.138	9
WFit_05	0	0.04	0.02	0.02	0	0.02	0	0.01	0	0.03	0.96	0.42	0.15	0.51	0.20	0.15	0.01	1	0.22	0.15	0.195	4
WFit_10	0	0.09	0.15	0.17	0.01	0.05	0.06	0.06	0.03	0.25	0.97	0.01	0.04	0.06	0.38	0.28	0	0.02	1	0.28	0.194	5
WFit_15	0.02	0.26	0.32	0.32	0.01	0.14	0.16	0.14	0.08	0.45	0.98	0	0.01	0.02	0.06	0.39	0	0	0.02	1	0.219	**1**
Mean	0.128	0.171	0.169	0.140	0.132	0.160	0.132	0.106	0.069	0.142	0.477	0.113	0.082	0.095	0.096	0.114	0.051	0.062	0.074	0.092		
Rank	12	19	18	15	14	17	13	9	3	16	20	10	5	7	8	11	**1**	2	4	6		

The ranking of the strategies is based on the coverage percentage. The higher ranking implies that the front is better.

Table 7 reports the SC results for each of the twenty Pareto fronts. According to the set coverage metric, the strategy that has the best compromise between the number of billing hours and quality reduction is Wfit_15, followed by MaxFTFit, RR_15, and WFit_05.

We see that MaxFTFit dominates the fronts of other strategies in the range of 0–60%, with 21.8% in average occupying the second rank. $\mathrm{SC}(A, \mathrm{MaxFTFit})$ shows that MaxFTFit is dominated by other fronts on 6.9% in average. Meanwhile, WFit_05 and MaxFTF with the second and third ranks are dominated by other strategies on 19.5% and 21.9% on average, respectively. They are dominated for other strategies on 6.2% and 6.9%.

However, we should not consider only Pareto fronts, when many solutions are outside the Pareto optimal solutions. This is the case of BFit_xx, FFit_xx, and RR_xx: although the Pareto fronts are of good quality, many of the generated solutions are quite far from them, and, hence, a single run of the algorithm may produce significantly worse results.

11. Conclusions and Future Work

In this paper, we formulate and study scheduling problems addressing VoIP service in cloud computing. We define models of the provider cost and quality of service and propose new on-line nonclairvoyant bin-packing algorithms for call allocation. Unlike the standard formulation of the problem, our bins are always open, even if they are completely packed. Items in bins can disappear after call termination, and utilization can be changed at any moment. The problem is dynamic, when no knowledge about call duration or its estimation is used.

Due to the fact that VM parameters are changing over time, traditional scheduling techniques based on number of calls do not adapt well to this dynamism. VoIP solutions do not take into account uncertainty in dynamic and unpredictable cloud environments.

Our approach is suitable for environment with presence of uncertainty. It takes allocation decisions depending on the actual cloud and VM characteristics at the moment of allocation such as number of available virtual machines and their utilization. It can cope with different workloads, type of calls (voice, video, conference, etc.), cloud properties, and cloud uncertainties, such as elasticity, performance changing, virtualization, loosely coupling application to the infrastructure, and parameters such as an effective processor speed and actual number of available virtual machines. We propose twenty VoIP scheduling strategies and evaluate their performance by comprehensive simulation analysis on real data considering six months of the MIXvoip company service.

We show that the proposed algorithms can be efficiently used in a VoIP cloud environment. The monoobjective and biobjective analyses provide a good compromise between saving money and voice quality.

However, further study is required to assess their actual performance and effectiveness in a real domain. This will be the subject of future work. Moreover, quality in communication systems, hypervisor-level scheduling, dynamic consolidation of calls and VMs, and distributed load balancing are other important issues to be addressed.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work is partially supported by CONACYT (Consejo Nacional de Ciencia y Tecnología, México) (Grant no. 178415). The work of Pascal Bouvry is partly funded by INTER/CNRS/11/03 Green@Cloud. The work of Alexander Yu. 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 (Government Regulation no. 218 from 09/04/2010).

References

[1] A. M. Alakeel, "A guide to dynamic load balancing in distributed computer systems," *International Journal of Computer Science and Network Security*, vol. 10, no. 6, pp. 153–160, 2012.

[2] 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 *Proceedings of the 4th IEEE International Workshop on Cloud Computing Systems, Networks, and Applications (CCSNA '15), in Conjunction with IEEE Global Communications Conference (GLOBECOM '15)*, San Diego, Calif, USA, December 2015.

[3] J. M. Cortés-Mendoza, A. Tchernykh, A.-M. Simionovici et al., "VoIP service model for multi-objective scheduling in cloud infrastructure," *International Journal of Metaheuristics*, vol. 4, no. 2, pp. 185–203, 2015.

[4] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[5] A. Tchernykh, U. Schwiegelsohn, V. Alexandrov, and E. Talbi, "Towards understanding uncertainty in cloud computing resource provisioning," *Procedia Computer Science*, vol. 51, pp. 1772–1781, 2015.

[6] MIXvoip S.a. Company, https://www.mixvoip.com/.

[7] P. Montoro and E. Casilari, "A comparative study of VoIP standards with asterisk," in *Proceedings of the 4th International Conference on Digital Telecommunications (ICDT '09)*, pp. 1–6, Colmar, France, July 2009.

[8] "3CX Phone System and ATOM N270 Processor Benchmarking," http://www.3cx.com/blog/voip-howto/atom-processor-n270-benchmarking/.

[9] Cisco Systems, Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation, http://www.cisco.com/c/en/us/support/docs/voice/h323/14069-codec-complexity.html.

[10] L. Madsen, R. Bryant, and J. V. Meggelen, *Asterisk: The Definitive Guide*, O'Reilly Media, 3rd edition, 2011.

[11] J. So, “Scheduling and capacity of VoIP services in wireless OFDMA systems,” in *VoIP Technologies*, S. Kashihara, Ed., chapter 11, pp. 237–252, InTech, Rijeka, Croatia, 2011.

[12] H. Lee, T. Kwon, D.-H. Cho, G. Lim, and Y. Chang, “Performance analysis of scheduling algorithms for VoIP services in IEEE 802.16e systems,” in *Proceedings of the IEEE 63rd Vehicular Technology Conference (VTC ’06)*, pp. 1231–1235, Melbourne, Australia, July 2006.

[13] M. Folke, S. Landström, U. Bodin, and S. Wänstedt, “Scheduling support for mixed VoIP and web traffic over HSDPA,” in *Proceedings of the IEEE 65th Vehicular Technology Conference (VTC-Spring ’07)*, pp. 814–818, IEEE, Dublin, Ireland, April 2007.

[14] N. Bayer, B. Xu, V. Rakocevic, and J. Habermann, “Application-aware scheduling for VoIP in wireless mesh networks,” *Computer Networks*, vol. 54, no. 2, pp. 257–277, 2010.

[15] S. Wu, L. Zhou, D. Fu, H. Jin, and X. Shi, “A real-time scheduling framework based on multi-core dynamic partitioning in virtualized environment,” in *Network and Parallel Computing: 11th IFIP WG 10.3 International Conference, NPC 2014, Ilan, Taiwan, September 18–20, 2014. Proceedings*, vol. 8707 of *Lecture Notes in Computer Science*, pp. 195–207, Springer, Berlin, Germany, 2014.

[16] A. Mazalek, Z. Vranova, and E. Stankova, “Analysis of the impact of IPSec on performance characteristics of VoIP networks and voice quality,” in *Proceedings of the 5th International Conference on Military Technologies (ICMT ’15)*, pp. 1–5, IEEE, Brno, Czech Republic, May 2015.

[17] L. R. Costa, L. S. Nunes, J. L. Bordim, and K. Nakano, “Asterisk PBX capacity evaluation,” in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium Workshop (IPDPSW ’15)*, pp. 519–524, IEEE, Hyderabad, India, May 2015.

[18] K. Cheng, Y. Bai, R. Wang, and Y. Ma, “Optimizing soft real-time scheduling performance for virtual machines with SRT-xen,” in *Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid ’15)*, pp. 169–178, Shenzhen, China, May 2015.

[19] D. Tsafrir, Y. Etsion, and D. G. Feitelson, “Backfilling using system-generated predictions rather than user runtime estimates,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 6, pp. 789–803, 2007.

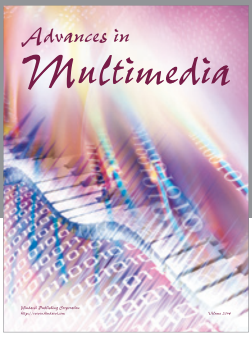
[20] A. Tchernykh, L. Lozano, U. Schwiegelshohn et al., “Online bi-objective scheduling for IaaS clouds ensuring quality of service,” *Journal of Grid Computing*, vol. 14, no. 1, pp. 5–22, 2016.

[21] A. Lezama Barquet, A. Tchernykh, and R. Yahyapour, “Performance evaluation of infrastructure as a service clouds with SLA constraints,” *Iberoamerican Journal of Research Computing and Systems*, vol. 17, no. 3, pp. 401–411, 2013.

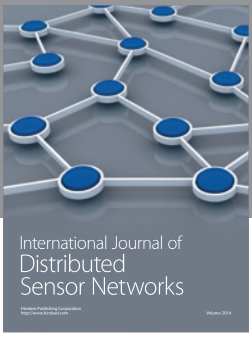
[22] “CloudSim: a framework for modeling and simulation of Cloud Computing infrastructures and services,” http://www.cloudbus.org/cloudsim/.

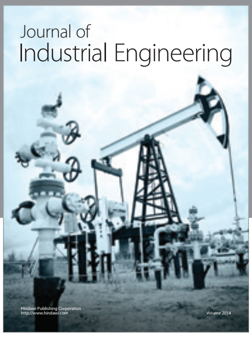
[23] T. D. Dang, B. Sonkoly, and S. Molnár, “Fractal analysis and modeling of VoIP traffic,” in *Proceedings of the 11th International Telecommunications Network Startegy and Planning Symposium*, pp. 123–130, Vienna, Austria, June 2004.

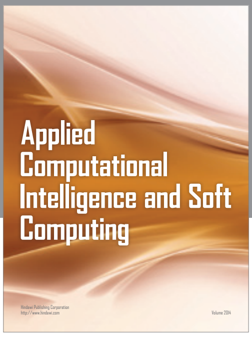
[24] J. M. Cortés-Mendoza, A. Tchernykh, A. Drozdov, P. Bouvry, A. Simionovici, and A. Avetisyan, “Distributed adaptive VoIP load balancing in hybrid clouds,” in *Proceedings of the 1st Russian Conference on Supercomputing (RuSCDays ’15)*, pp. 676–686, Moscow, Russia, September 2015.

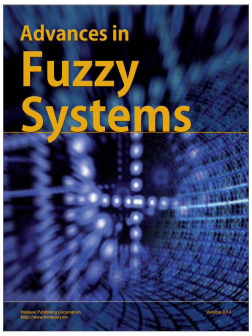

Advances in
Multimedia

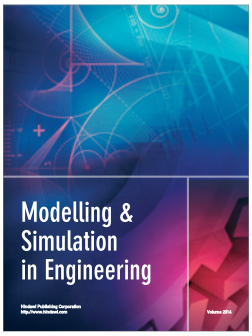
The Scientific
World Journal

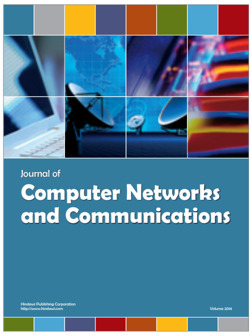

International Journal of
Distributed
Sensor Networks

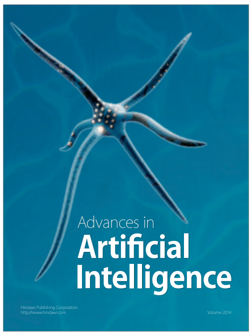

Journal of
Industrial Engineering

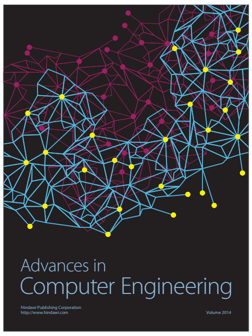

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

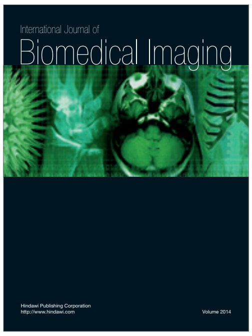

Modelling &
Simulation
in Engineering

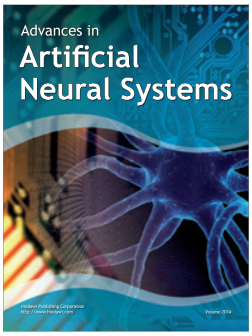

Journal of
Computer Networks
and Communications

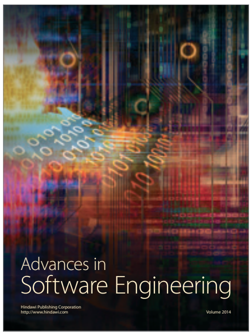

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

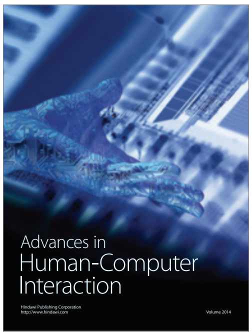
International Journal of
Computer Games
Technology

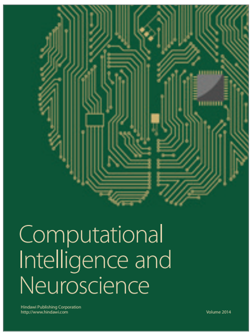

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering