VoIP Service Model for Multi-objective Scheduling in Cloud Infrastructure

Jorge M. Cortés-Mendoza[1], Andrei Tchernykh[1*], Ana-Maria Simionovici[2], Pascal Bouvry[2], Sergio Nesmachnow[3], Bernabe Dorronsoro[4], Loic Didelot[5].

[1] CICESE Research Center, Ensenada, Baja California, México.
chernykh@cicese.mx, jcortes@cicese.edu.mx
[2] University of Luxembourg. Luxembourg.
{Ana.Simionovici, Pascal.Bouvry}@uni.lu
[3] Universidad de la República, Uruguay. sergion@fing.edu.uy
[4] Universidad de Cádiz. bernabe.dorronsoro@uca.es
[5] MIXvoip S.A., Luxembourg. ldidelot@mixvoip.com

Abstract. Voice over IP (VoIP) is very fast growing technology for the delivery of voice communications and multimedia data over Internet with lower cost. Early technical solutions mirrored the architecture of the legacy telephone network. Now, they have adopted the concept of distributed cloud VoIP. These solutions typically allow dynamic interconnection between users on any domains. However, providers face challenges to use infrastructure in the best efficient and cost effective ways. Hence, efficient scheduling and load balancing algorithms are a fundamental part of this approach especially in presence of uncertainty of a very dynamic and unpredictable environment. In this paper, we formulate the problem of dynamic scheduling of VoIP services in distributed cloud environments and propose a model for bi-objective optimization. We consider it as the special case of the bin packing problem, and discuss solutions for provider cost optimization ensuring quality of service.

Keywords—Cloud computing; Load balancing, Voice over IP (VoIP), Provider cost minimization, Quality of service (QoS).

Reference to this paper should be made as follows: Cortes-Mendoza J. M., Tchernykh A., Simionovici A.M., Bouvry P., Nesmachnow S., Dorronsoro B., Didelot L., (2015). “VoIP Service Model for Multi-objective Scheduling in Cloud Infrastructure”.

Biographical notes: Jorge M. Cortés-Mendoza received his Bachelor’s degree in Computer Sciences from the Autonomous University of Puebla (Benemérita Univercidad Autónoma de Puebla, Mexico) in July 2008, and his Master Degree in Computer Science from CICESE Research Center in March 2011. Since 2013, he is working on distributed computing, cloud computing, load balancing and scheduling.

[*] corresponding author

Andrei Tchernykh received the Ph.D. in Computer Science from the Institute of Precision Mechanics and Computer Engineering (IPMCE, Moscow) of the Russian Academy of Sciences in 1986. During 1975-1995, he stayed in IPMCE, IHPCS, and SCC research institutes. He is `currently a full Professor in the Computer Science Department, CICESE Research Center, Ensenada, Mexico, and leads the Parallel Computing Laboratory. He is PI and co-PI of number of international research projects and grants. He served as the general co-chair, program co-chair and committee member of professional conferences. His main interests include scheduling, load balancing, adaptive resource allocation, scalable energy-aware algorithms, cloud computing, multi-objective optimization, heuristics and meta-heuristic, and computing with uncertainty.

Ana-Maria Simionovici is a PhD candidate at the University of Luxembourg. She is currently working on evolutionary computing, prediction and load balancing. She holds Master’s degree in Computational Optimization from University of Al. Ioan Cuza, Iasi, Romania (2012).

Pascal Bouvry earned his PhD in Computer Science with great distinction at the University of Grenoble (INPG), France in 1994. His research at the IMAG laboratory focussed on mapping and scheduling task graphs onto Distributed Memory Parallel Computers. Currently, he is a Professor at the Faculty of Sciences, Technology and Communication of the University of Luxembourg and heading the Computer Science and Communication research unit. Professor Bouvry is currently holding a full-time professor position at the University of Luxembourg in Computer Science. His current interests encompass optimisation, parallel/cloud computing, ad hoc networks and bioinformatics.

Sergio Nesmachnow (PhD in Computer Science from Universidad de la República, Uruguay, 2010) is a full-time Professor at Universidad de la República, Uruguay. He is a researcher in ANII and PEDECIBA, Uruguay. His current research interests include optimisation, parallel metaheuristics and scientific high-performance computing. He has published more than 100 papers in impact journals and conference proceedings, and participated in technical committees and as an invited speaker of international conferences. He currently serves as the Editor-in-Chief of International Journal of Metaheuristics, Inderscience Publishers.

Bernabé Dorronsoro received the degree in Engineering (2002) and the PhD in Computer Science (2007) from the University of Málaga (Spain), and he is currently working at the University of Cádiz (Spain). His main research interests include sustainable computing, Grid computing, ad hoc networks, the design of new efficient metaheuristics, and their application for solving complex real-world problems in the domains of logistics, telecommunications, bioinformatics, combinatorial, multiobjective and global optimisation. Among his main successful publications, he has several articles in impact journals and two authored books. Dr. Dorronsoro has been a member of the organising committees of several conferences and workshops, and he usually serves as reviewer for leading impact journals and conferences. Loic Didelot is the current Founder at Pindo S.a., CEO at Corpoinvest holding, co-owner at Forschung-Direkt Company. Since February 2008, he is also the founder and co-owner of MIXvoip S.A. He has competences in building highly

scalable and secure products based on Linux and open source solutions while considering the commercial alternatives. He has great know-how in web development (PHP, CSS, Javascript, AJAX), Linux and security, Asterisk. He is interested in database applications that need to scale out.

1 Introduction

The Quality of Service (QoS) guarantee that has to be delivered to the end users is one of the major challenges for cloud computing. For Voice over Internet Protocol (VoIP), it comprises requirements on all aspects of a call such as service response time, throughput, loss, interrupts, jitter, latency, resource utilization, and so on. Several ways exist to provide QoS: scheduling, traffic control, dynamic resource provisioning, etc. The development of effective dynamic VoIP scheduling solutions involves many important issues: load estimation and prediction, load levels comparison, performance indices, system stability, job resource requirements estimation, resource selection for job allocation, etc. (Alakeel, 2010, Tchernykh et al. 2015, Tchernykh et al. 2014, Schwiegelshohn et al. 2012). Virtualization in cloud computing adds other complexity dimension to the problem: distribution of the virtual machine (VM) and their migrations.

Many businesses who are adopting VoIP systems are always looking for a way to cut down costs. Provider cost is determined by hardware efficiency, resource management system deployed on the infrastructure, and the efficiency of applications running on the system (Beloglazov et al. 2012).One of the ways to reduce a cost is to avoid provisioning of more resources than required by users and QoS.

In this paper, we describe a model for VoIP load balancing focusing on both important aspects: QoS and provider cost optimization. There are two main beneficiaries of this optimization: technology providers running its software on the cloud (e.g. the VoIP provider) and end users.

The paper is structured as follow. The next section briefly discusses VoIP service considering underlined infrastructure, software and calls. Section III presents several aspects of the QoS and provider cost. Section IV provides the problem definition (jobs, cloud infrastructure and criteria). Section V describes a general approach for VoIP assignation and the algorithms. Section VI presenting the result of mono-objective and bi-objective analysis and Section VII concludes the paper by presenting the main contribution of the work.

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 network1. VoIP services significantly reduce calling rates.

A cloud based VoIP 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.

VoIP requires the service availability all the time for any number of users. To deal with increasing number of clients, providers can invest in a large infrastructure to avoid

[1] ISDN (Integrated Services Digital Network) is a set of standards for digital transmission over ordinary telephone copper wire.

loss of calls (hence, users). In this case, the infrastructure is usually underutilized. Moreover, servers should be replaced due to resource degradation.

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

VoIP calls require signaling, channel setup, voice signal digitization, encoding, etc. The voice nodes handle the calls with different features such as: voicemail, call forwarding, music on hold, conference calls, etc. depending on customers.

From a client perspective, in order to use VoIP services, an Internet connection and an IP hard-phone or soft-phone are needed. The clients connect to a voice server, which is the main part of the VoIP telephony system, typically located in a data center or in a public cloud (Figure 1). The voice nodes communicate with the database in the system, where all the users are registered with name and password. Every call is recorded with details such as: destination, duration, etc.

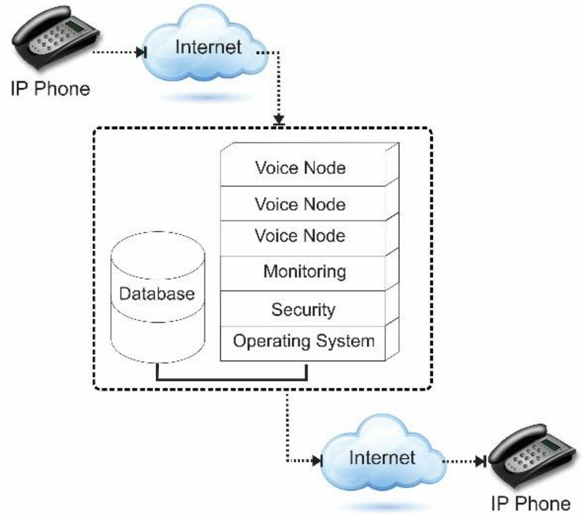

Fig. 1 . Cloud VoIP architecture.

General VoIP architecture includes elements of communication infrastructure that connect phones remotely through the Internet. The servers provide gateway, interconnection switches, session controllers, firewall, etc. They use software to emulate a telephone exchange.

A drawback of this architecture arises when the hardware reaches its maximum capacity. Traditional VoIP solutions are not scalable. To scale it is necessary to increase infrastructure or replace existing hardware. Overprovisioning and, hence, cost overrunning 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.

In a cloud-based VoIP solution, the voice nodes are operated as VMs that provide a variety of services. 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 manage telephones via clouds different characteristics need to be improved. The most important is the utilization of the infrastructure.

The advantage of this architecture consists in increased scalability and low cost. However, it has several unsolved problems, for instance, to optimize the overall system performance, the processor utilization has to be high, but it reduces quality of the call. Hence, load of the VoIP servers should be reduced to guarantee the QoS. On the other hand, the processor idle time increases the useless expenses of the cloud provider.

The most important cause of the load imbalance is the dynamic nature of the problem in both computational and communication costs (Tchernykh et al. 2015a). Load-balancing is one of the mechanisms to maximize VoIP system performance by minimizing the number of processing units without overloading them. It can improve the local load imbalance and guarantee QoS.

2.1 Infrastructure

A telephone system is consisted of multiple components: telephones, cables, physical or virtual machines that host and run call exchange software, signaling and communication modules, software that establishes the voice mails, etc.

MIXvoip (2015) developed the concept of the super-node (SN) and Super Nodes Cluster (SNC) to enrichment features for telephone exchanges (Figure 2).

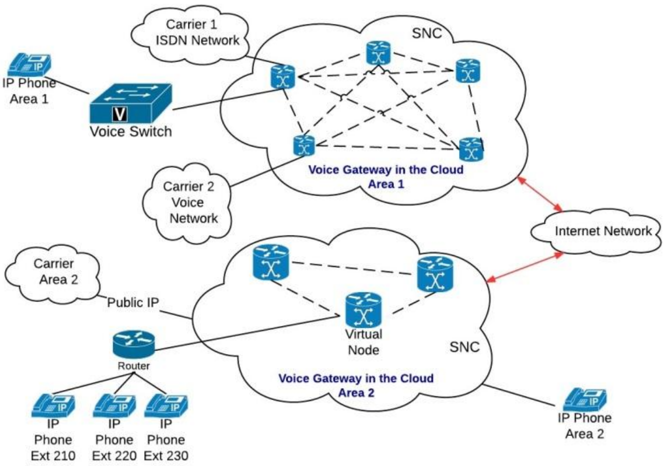

Fig. 2 SNC deployment

SNC is a set of SN deployed in a cloud, and interconnected logically at a local level. This allows minimizing the path between two local users, and increasing the quality of the voice. This deployment brings redundancy on a given geographical area, but ensures a high voice quality between the SNC nodes through the public Internet. SNCs are deployed in widely distributed geographical locations. As shown in Figure 2, when a user in Area 1 wishes to establish a call, it will be sent to the nearest SN in his area. If customers are small businesses, they do pass on average 50% of their calls locally or in neighboring countries. The deployment of this architecture allows providing services near ISDN quality in a public IP network.

The interconnection of the system with other operators is provided through the Internet or a physical wire connection between two devices in a data center, where the operator and infrastructure meet on short distances.

2.2 Software

The most known telephone software for processing calls and providing a powerful control over call activity is Asterisk (Madsen et al. 2011). It is a framework under free license for building multi-protocol, real-time communication solutions. It establishes and manages connections between two end devices. It is the most known Private Branch Exchange (PBX) to place calls via Internet, and to connect to traditional Public Switched Telephone Network (PSTN). Delivering information and transferring data are based on specific protocols, such as Session Initiation Protocol (SIP) and Real Time Protocol (RTP).

SIP is one of the most known protocols for signaling, establishing presence, locating users, setting, modifying and tearing down sessions between end-devices. It is used for controlling multimedia communication sessions such as voice and video calls over IP networks.

After the connection is established, 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 one or multiple 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 VoIP Quality of Service

3.1 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 when transcoding is not used. In the latter case, Asterisk is in charge of only routing the call. However, depending on the binary rate of the codec, the processor load is influenced as well. Table 1 shows processor utilization for call without transcoding (Montoro and Casilari, 2009).

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, etc. Some codec compression techniques require more processing power than others. An example of compression methods is presented in Table 2.

In (Georgiou, 2015) the authors presented results of the benchmark test that includes stress testing of Queue Calls, VoIP Provider Calls and Normal Extension to Extension calls.

Table 1 Processor utilization for 1 call 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%

Table 2 Codec Compression Method (“Understanding Codecs,” 2006).

Abbreviation	Method
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

Queuing Calls is used by Call Centers that prefer to answer to the incoming calls automatically and place them in a queue, instead of rejecting them. It 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.

Table 3 Queue Calls.

Normal Call Center Activity Test	Jitters	CPU Usage	Simultaneous Calls	CPU Usage per 1 Call
5 Calls to Queue	None	14%	10	1.4%
10 Calls to Queue	None	18%	20	0.9%
15 Calls to Queue	None	28%	30	0.93%
20 Calls to Queue	None	36%	40	0.9%
30 Calls to Queue	None	67%	60	1.11%
40 Calls to Queue	None	84%	80	1.05%

3.2 Quality of Service

QoS requirements for VoIP are very important. The service quality degradation is determined by the transit of the packets across the Internet, queuing delays at the routers, packet travel time from source to destination, jitter: deviations of the packet inter-arrivals, packet loss, call set-up time, and call tear-down time, etc.

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.

In general, QoS standards for VoIP traffic are set for voice. One of the possible generalizations of the voice quality is processor utilization. Each codec provides a certain quality of speech only if processor utilization is low enough in order to ensure QoS. Theoretically, processor utilization of 100% provides the best expected performance.

However, with increasing number of call, hence utilization, CPU cannot be able to handle the stress anymore and jitters and broken audio symptoms will appear (Figure 3).

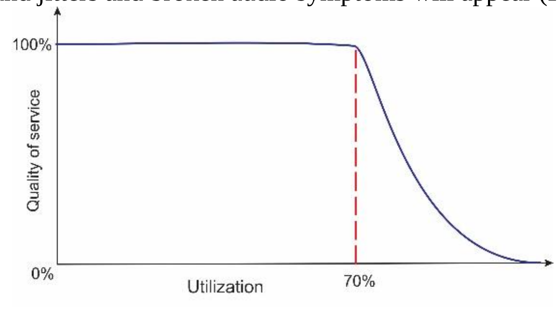

Fig. 3 Voice quality versus processor load (utilization).

3.3 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 (compute, storage, software, and associated VoIP components); Infrastructure cost (power distribution, cooling equipment, space for facilities, etc.); Operational cost (energy, cooling…); Network cost (links, 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 it is 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 to 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 increase quality of service.

In the first scenario, we consider single-objective optimization problem: to minimize the total cost of VMs. In order to ensure good QoS of the VoIP traffic, the utilization of the VMs should be kept under the certain threshold (e.g. 70%).

In the second scenario, we consider the bi-objective optimization approach that is not restricted to find a unique solution but a set of solution known as a Pareto optimal set. In this case, we minimize the cost of VMs and VM utilization. A tradeoff between the two objectives depends on the VoIP provider’s preference.

4 Model.

We address the model for VoIP in distributed cloud environment with high heterogeneity of the resources with different number of servers, execution speed, energy efficiency, amount of memory, bandwidth, etc.

Let us consider that VoIP cloud infrastructure consists of m heterogeneous SNCs: $SNC_1, SNC_2, ..., SNC_m$ with relative speeds $s_1, s_2, ..., s_m$. Each SNC_i, for all $i = 1...m$, consists of m_i SNs. Each $SN^i{}_k$, for all $k = 1...m_i$, runns k_i(t) VM at time t. We assume that VMs of one SNC are identical and have the same processing capacity.

We denote the number of billing hours in SNC_i by $\bar{m}_i = \int_{t=0}^{C_{max}} k_i(t) \cdot m_i \, dt$ and run in all SNC by $\bar{m} = \sum_{i=1}^{m} \bar{m}_i$. The VM is described by a tuple $\{vmu_i(t)\}$, where $vmu_i(t)$ is the utilization (load) of the VM at time t. VM hosts one or multiple Asterisk running processes that run and handle calls.

The SNC contains a set of routers and switches that transport traffic between the SNs and to the outside world. They are characterized by the amount of traffic flowing through it (Mbps). 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 $J_1, J_2, ..., J_n$ 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 (time) is unknown until the job has completed. The utilization is a constant for a given job that depends on the used codec and normalized for the slowest machine.

We define the provider cost model by considering a function that depends on the number of VMs and the running time $\bar{m}$.

In multi-objective 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 that 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 multi-objectives problems are compared via visual observation of the solution space. One of formal and statistical approaches uses a set coverage metric SC(A,B) that calculates the proportion of solutions in B, which are dominated by solutions in A:

$$SC(A,B) = \frac{\left|\{b \in B; | \exists a \in A : a \leq b\}\right|}{|B|} \tag{1}$$

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

5 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 regressions, when it increases above the certain threshold, or improvement, when it is below, and is a useful for VoIP QoS problem studying.

The concept of VM utilization used in our problem is simple. Assume 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 our self to consider Asterisk running processes and calls.

5.1 Dynamic packing with open bins

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

We consider on-line variant of the problem in which items are received one by one. Before info about the next item is revealed, the scheduler needs to decide whether the next item is packed in the currently open bin or a new bin is opened.

In other words, each decision has to be made without any knowledge on the duration of the call, only contribution to utilization is known due to the known used codec.

The principal novelty of this problem variation lies in the fact that the state of the bin is determined not only by actions of the decision maker during item allocations, but also by item completions after their lifespan. Unlike in standard formulation, bins are always open, even completely packed, and dynamic. Items in bins can be terminated (call termination) and utilization can be changed at any moments.

Bin-packing remains one of the classic 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.

In this paper, we consider that the bin size is equals to 1 that corresponds to 100% of VM utilization.

We use an adaptation of three known bin packing strategies First-Fit (FFit), Best-Fit (BFit), and Worst-Fit (WFit) to allocate calls to VMs (Table 4). In our case, we do not have the option to sort the input, due to the fact that we face with an online bin packing problem. Instead, we sort bins in decreasing order by their utilization in the BFit strategy.

After allocation phase, we have a set of VMs that host one or multiple Asterisk running processes and calls. Each VM is characterized by a relative execution speed, and current utilization (load).

Table 4 Allocation Strategies

	Description
Rand	Allocates job *j* to a suitable machine randomly selected using a uniform distribution.
RR	Allocates job *j* to a suitable machine using a Round Robin algorithm.
FFit	Allocates job *j* to the first machine available and capable to execute it.
BFit	Sorts VMs in decreasing order by their utilization, and allocates job *j* to the first VM.
WFit	Sorts VMs in decreasing order by their utilization, and allocates job *j* to the last VM

6 Experimental Validation

6.1 Simulation Toolkit

All experiments are performed using the CloudSim: a framework for modeling and simulation of cloud computing infrastructures and services (Calheiros et al. 2011). It is a standard trace based simulator that is used to study cloud resource management problems. We have extended CloudSim to include our algorithms using the java (JDK 7u51) programming language.

We extend the CloudSim by introducing the support of dynamic arrival of the jobs (calls), updating the system parameter before scheduling decisions (utilization of the resources), and implementing the broker policies for call allocation.

Parameters are directly taken from traces of real VoIP service considered in (Simionovici et al., 2015). We use SWF (Standard Workload Format) with four additional fields to process the calls.

6.2 Workload

The workload is a set of registered phone calls that have been handled by the 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, etc.

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, jitter; calls disconnected; etc.

Examples of the call distribution during a day and week are presented in Figure 4 and Figure 5. For business customers, it is typical that the peak hours are between 8-11 AM and 13-18 PM. Over a week, the traffic is very high from Monday till Friday, while for weekends it decreases considerably.

Figure 5 shows an example of call duration distribution during a working day, which depends significantly on the clients (e.g. call centers, schools, business companies, etc.). In our example, the duration of the majority of the calls is short (e.g. 1-5 minutes).

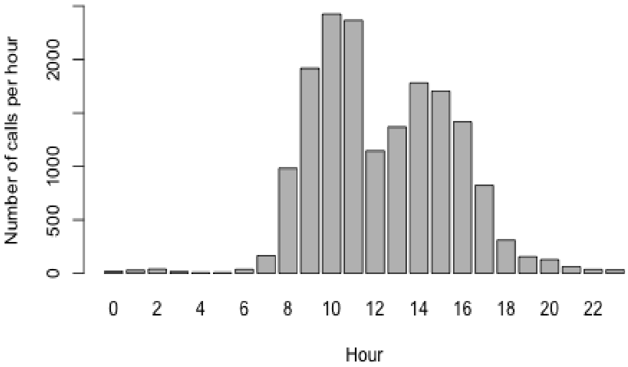

Fig. 4 Example of the call distribution during a week.

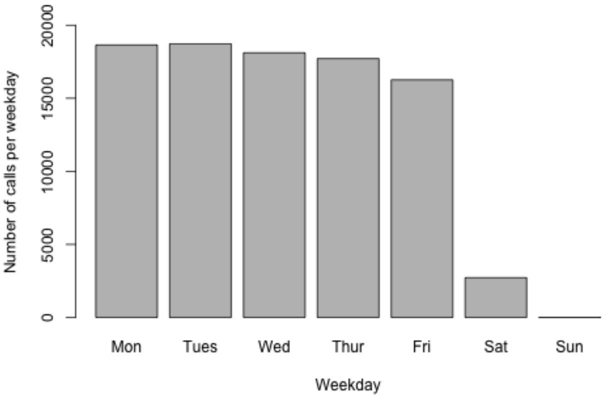

Fig. 5 Example of the call distribution during a week.

Dang et al. (2004) 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 silence and transmission durations are fitted by a generalized Pareto distribution as well. A series of probability distributions were tested with the data and the Generalized Pareto Distribution resulted as the best fit. 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 6).

For the analysis, we use 30 workloads; each includes phone calls made during one day.

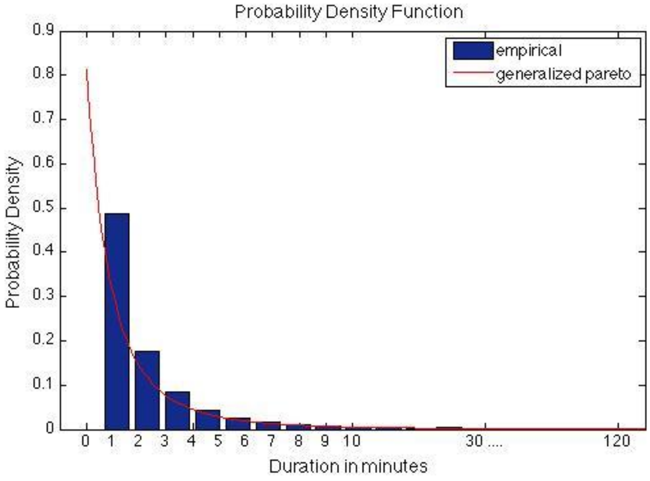

Fig. 6 Example of call duration distribution and Generalized Pareto Distribution (PDF).

6.3 *Experimental results*

First, we analyze the mono-objective problem, where a utilization threshold is used to guaranty the QoS. Then, we realize a bi-objective analysis, where no threshold is used (100% of utilization is allowed) to study the relation between total cost (the number of hours running VMs) and the utilization of the VMs.

Mono-objective analysis. We evaluate the performance of the five strategies with a 70% utilization threshold of VMs to ensure the QoS. Figure 7 shows the number of billing hours during a month. The strategies have similar behavior when workload is low (during weekends).

During the week, strategies produce significally different cost. Bfit and FFit use about 55 billing hours, while Rand, RR, and WFit use about 70 hours.

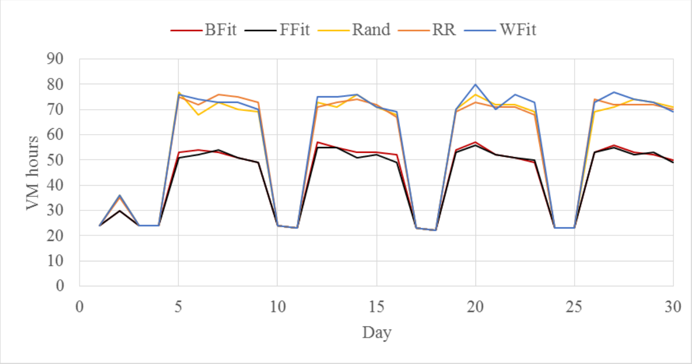

Fig. 7 The number of billing hours during 30 days

Figure 8 shows an example of the number of billing hours during a day. The workload is low during the first hours of the day, all strategies have the same behavior and they use just one VM to process the calls. The maximum number of VMs running during peak hours is 5 VMs.

Figure 9 shows the average number of billing hours during 30 days. FFit and BFit are the best strategies. They use 42.7, 43.2 billing hours on average.

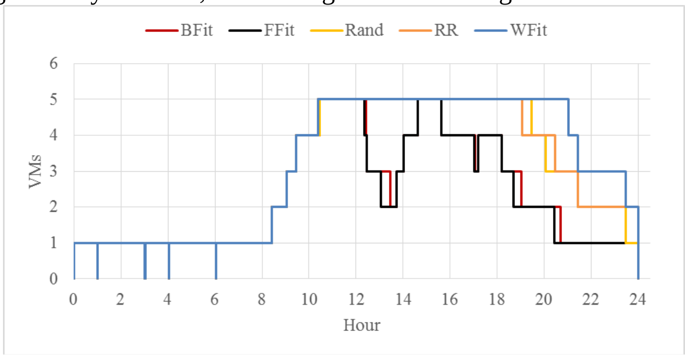

Fig. 8 Example of the number of billing hours during a day.

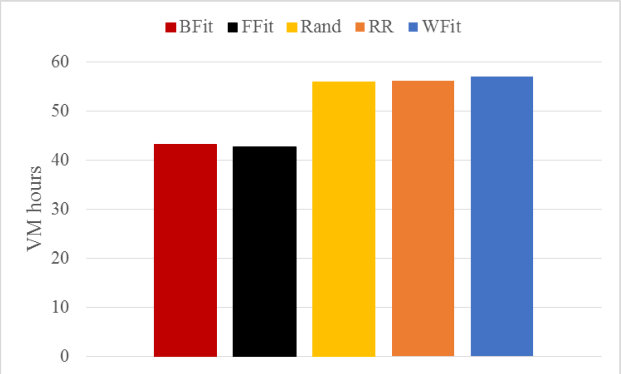

Fig. 9 Average billing hours per day.

The monthly difference between these strategies Rand, RR, and WFit is about 13 hours per day. Rand and RR need 55.9 and 56.1 billing hours. The worst strategy is WFit with 57.1 billing hours per day on average.

Bi-objective analysis. For the bi-objective 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 10 shows a set of solutions approximating the Pareto front for each of 5 strategies: Rand, RR, FFit, BFit, WFit and 30 workloads. This two-dimensional solution space represents a feasible set of solutions that satisfy the problem's constraints.

Note that we address the problem of minimizing cost and utilization. For better representation, we convert it to the minimization of two criteria: degradations of both the cost and utilization.

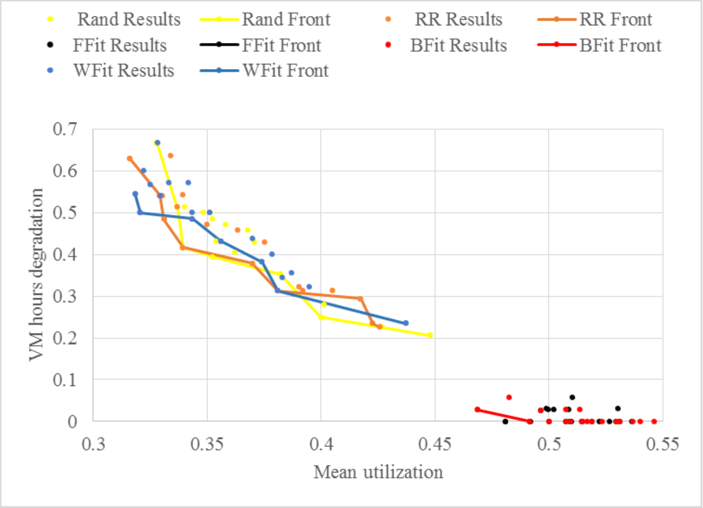

Fig. 10 The solution sets and Pareto fronts.

The solution space covers a range of values of cost degradation from 0 to 0.67, whereas values of utilization degradation are in the range from 0.31 to 0.55.

We see that BFit and FFit are located in the lower-right corner, being among the best solutions in terms of both objectives. They noticeably outperform Round Robin that is in current use for VoIP service.

However, we should not consider only Pareto fronts, when many of the solutions are outside the Pareto optimal solutions. This is the case of BFit: although the Pareto front is of good quality, many of the generated solutions are quite far from it, and, hence, a single run of the algorithm may produce significantly worse results. FFit solutions cover cost degradations from 0 to 0.058, whereas BFit solutions are in the range from 0 to 0.057.

Set coverage method is used to analyze the performance of the studied bi-objective scheduling strategies. Using this metric, two sets of non-dominated solutions can be compared.

Table 4 and Table 5 report the SC results for each of the five Pareto fronts. The rows of the table 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 B over A, and rank of the strategy in each column. We see that SC(FFit,B) dominates the front of the BFit strategy in 50%, on average. SC(A,FFit) shows that BFit is not dominated by the fronts of other strategies. Meanwhile, SC(RR,B) dominates the fronts of the other two strategies in the range 67% to 72%. SC(A,RR) shows that WFit and Rand dominate RR for 12% on average.

The ranking of strategies is based on the percentage of coverage. The higher ranking of rows implies that the front is better. The rank in columns shows that the smaller the average dominance, the better the strategy. According to the set coverage metric, the strategy that has the best compromise between minimized the number of billing hours and minimizing utilization is FFit, followed by RR on the second position.

Table 4. Set coverage and ranking FFit and Bfit.

A \ B	FFit	BFit	Mean	Rank
FFit	1.0	0.5	0.75	**1**
BFit	0.0	1.0	0.50	**2**
Mean	0.50	0.75		
Rank	**1**	**2**		

Table 5. Set coverage and ranking, Rand, RR, and WFit.

A \ B	Rand	RR	WFit	Mean	Rank
Rand	1.0	0.111	0.428	0.513	**3**
RR	0.666	1.0	0.714	0.793	**1**
WFit	0.444	0.111	1.0	0.518	**2**
Mean	0.703	0.407	0.714		
Rank	**2**	**1**	**3**		

7 Conclusions

In this paper, we formulate and discuss the model for job allocation problem addressing VoIP in cloud computing. We define models of the provider cost and quality of service, and propose new allocation bin packing algorithms for VoIP super node clusters. It is suitable for environment with presence of uncertainty, and take into account QoS and cost optimization.

It does not take into account call duration, its estimation, topology, and communication bandwidth. It takes allocation decisions depending on the actual cloud and VM characteristics at the moment of allocation such as number of available virtual machines, their utilization, etc.

Due to these parameters are changing over time, allocation adapts to these changes. This approach can cope with different workloads, cloud properties, and cloud uncertainties such as elasticity, performance changing, virtualization, loosely coupling application to the infrastructure, parameters such as an effective processor speed, number of available virtual machines, and actual bandwidth, among many others.

The proposed algorithm can be used for a VoIP cloud environment. However, further study is required to assess its actual efficiency and effectiveness in each domain. This will be the subject of future work. Moreover, dynamic consolidation and load balancing is another important issue to be addressed.

Acknowledgements

This work is partially supported by CONACYT (Consejo Nacional de Ciencia y Tecnología, México), grant no. 178415. The work of P. Bouvry and A. Simionovici is partly funded by INTER/CNRS/11/03 Green@Cloud.

References

Alakeel, A. M. (2010). A Guide to Dynamic Load Balancing in Distributed Computer Systems. International Journal of Computer Science and Network Security, 10(6), 153 – 160.

Beloglazov, A., Abawajy, J., and Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. Future Generation Computer Systems, 28(5), 755–768.

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., and Buyya, R. (2011). CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. Softw. Pract. Exper., 41(1), 23–50.

Dang, T. D., Sonkoly, B., and Molnar, S. (2004). Fractal analysis and modeling of VoIP traffic. In Telecommunications Network Strategy and Planning Symposium. NETWORKS 2004, 11th International (pp. 123–130).

Georgiou, L. (2015). 3CX Phone System and ATOM N270 Processor Benchmarking. http://www.3cx.com/blog/voip-howto/atom-processor-n270-benchmarking/

Madsen, L., Meggelen, J. V., and Bryant, R. (2011). Asterisk: The Definitive Guide. O'Reilly Media, Inc.

MIXvoip. (2015). http://mixvoip.com/

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

Simionovici, A.-M., Tantar A.T., Bouvry, P., Tchernykh, A., Jorge M. Cortes-Mendoza, and Didelot, L.(2015). VoIP Traffic Modelling using Gaussian Mixture Models, Gaussian Processes and Interactive Particle Algorithms. Presented at the The Fourth IEEE International Workshop on Cloud Computing Systems, Networks, and Applications 2015, San Diego, CA, USA.

Schwiegelshohn, U., Tchernykh, A. (2012). Online Scheduling for Cloud Computing and Different Service Levels, 26th Int. Parallel and Distributed Processing Symposium Los Alamitos, CA, pp. 1067–1074.

Tchernykh, A., Pecero, J., Barrondo, A., Schaeffer E. (2014). Adaptive Energy Efficient Scheduling in Peer-to-Peer Desktop Grids, Future Generation Computer Systems, 36:209–220.

Tchernykh A., Schwiegelsohn U., Alexandrov V., and Talbi E. (2015). Towards Understanding Uncertainty in Cloud Computing Resource Provisioning. SPU'2015 - Solving Problems with Uncertainties (3rd Workshop). In conjunction with The 15th International Conference on Computational Science (ICCS 2015), Reykjavík, Iceland, June 1- 3, 2015. Procedia Computer Science, Elsevier, Volume 51, Pages 1772–1781.

Understanding Codecs: Complexity, Hardware Support, MOS, and Negotiation. (2006). http://cisco.com/c/en/us/support/docs/voice/h323/14069-codec-complexity.html