PhD-FSTC-2016-46
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defence held on 25/10/2016 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN *INFORMATIQUE*

by

Ana-Maria Simionovici
Born on 14 September 1988 in city of Iasi, (Romania)

LOAD PREDICTION AND BALANCING FOR CLOUD-BASED VOICE-OVER-IP SOLUTIONS

Dissertation defense committee

Dr Pascal Bouvry, Dissertation Supervisor
Professor, Université du Luxembourg

Dr Steffen Rothkugel, Chairman
Professor, Université du Luxembourg

Dr Henri Luchian, Vice Chairman
Professor, Alexandru Ioan Cuza University of Iasi

Dr Franciszek Seredynski
Professor, Cardinal Stefan Wyszynski in Warsaw

Dr Andrei Tchernykh
Professor, Centro de Investigación Cientfica y de Educación Superior de Ensenada

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Prof. Pascal Bouvry for the continuous support during the Ph.D study and research, for the guidance and encouragement that added considerably to my experience. He provided me with direction, guidance and shared his expertise in many areas. It was an honor to be part of his team composed of people who were not only great colleagues but became my friends as well.

I would like to thank the other members of my committee, Prof. Dr. Henri Luchian, and Prof. Dr. Steffen Rothkugel, for guiding my research, insightful comments and support. My sincere gratitude also goes to Alexandru Tantar and Johnatan Pecero for providing assistance, exchange of knowledge and interesting collaborations.

I would like to express my appreciation to Loic Didelot and his team for the technical support during our collaboration. All you have been providing me with qualitative information and data that was the core of my thesis. I would also like to thank Prof. Albert Zomaya and Prof. Andrei Tchernykh for advising me with the research and to all the other collaborators for the fruitful cooperation.

My research would not have been possible without the financial assistance of FNR (Fonds National de la Recherche Luxembourg), Luxembourg Ministry of Economy and University of Luxembourg and I express my gratitude to these institutions.

I would especially like to thank to my family for the support and patience they have showed through the process. Words can not express my gratitude for my mother, Dorica, my father, Constantin and my sister, Oana Mihaela. I dedicate my work to them. I would like to also thank to my other family members who have always encouraged me and have send me their prayers. A special thanks goes to my friends. They have been supporting me through difficulties and helped me to go beyond the obstacles, to see the beautiful side of life.

Abstract

The evolution of technology in the past years together with the release of Voice Over Internet Protocol (VoIP) products and services, lead to an increased usage of these advanced communication technologies. As VoIP became very popular, it has been changing the face of business every day. Corporations, Small and Medium Businesses (SMB) have integrated VoIP in their daily businesses in order to lower operational costs at the infrastructure level. Dynamic optimization based on incoming load analysis and load prediction, pro-actively scaling the available resources in order to handle the incoming traffic, is one approach to prevent this type of problems. Anticipation of the computational load induced on processors by the incoming requests, is used to optimize load distribution and resource allocation.

The main domains investigated in the scope of the thesis are: load prediction of the incoming traffic, allocation of tasks to processor and provision of computational resources in order to ensure Quality of Service (QoS). Powerful prediction models that evolve, adapt and consistently deal with varying factors, used for shaping future traffic patterns and capacity planning, are designed. The developed prediction framework proposes a combination of different methodologies, namely, Interactive Particle System (IPS) which is based on interactive particle algorithms, Gaussian Mixture Model (GMM) that is model based learning of mixture of Gaussians, and supervised learning that defines distributions over functions Gaussian Process (GP). Insights of how particle algorithms are used for optimization, in conjunction with implicit learning models, are given.

Load balancing techniques are designed for a real VoIP system and the problem of job allocation for VoIP in cloud computing is addressed. VM-Aware Adaptive Rate of Change (VMA-AdRoC) is an extension of the RoC-LB algorithm, particularly useful for systems that handle VoIP traffic, sensitive to poor network configurations and poor resource distribution management. A site load distribution method was built using and Integer Linear Programming (ILP) approach, a model that integrates prediction and takes decisions based on the objective of minimizing the cost via minimizing the number of running machines that handle the placed calls. When fitted into the Domain Name System (DNS) framework, Global Server Load Balancer (GSLB) is provided and the selection of voice nodes to handle calls is based on site health conditions, site response time, geography-based site, routing cost and so further. Dynamic bin packing with open bins is a variation of the well known one-dimensional online bin-packing problem, considered to address congestion and overload issues at the level of a cloud-based

VoIP systems. Different bin packing strategies, First-Fit, Best-Fit, Worst-Fit, are developed and their performance is compared with two widely used server load balancing strategies, namely Round Robin and Random.

Contents

List of Figures

List of Tables

Acronyms

AIC Akaike Information Criterion 64

BPaaS Business-Process-as-a-Service 31

B2BUA Back-to-back user agent 54

CaaS Communication-as-a-Service 7

CDR Call-Detail-Record 21

CC Cloud Computing 15

CODEC COder/DEcoder 24

CPS Call per Second 54

CSC Cloud Service Customer 32

CSP Cloud Service Provider 32

DiffServ Differentiated Services 26

DNS Domain Name System 1

EM Expectation-Maximizatiom 64

FMI Future Market Insights 13

GMM Gaussian Mixture Model 1

GP Gaussian Process 1

GPML Gaussian Process Machine Learning 67

GSLB Global Server Load Balancer 1

HaaS Hardware-as-a-Service 31

IaaS Infrastructure-as-a-Service 15

ICT Information and Communication Technology 28

IEC International Electrotechnical Commission 28

IETF Internet Engineering Task Force 23

ILP Integer Linear Programming 1

IP Internet Protocol 12

IPS Interactive Particle System 1

IPS-AL Interactive Particle System - Average Likelihood 70

IPS-ML Interactive Particle System - Maximum Likelihood 72

ISP Internet Service Provider 93

ISDN Integrated Service Digital Network 21

ISO International Organizations for Standardization 28

ITU-T International Telecommunication Unit 24

LAN Local Area Network 23

LB Load Balancer 27

MaaS Metal-as-a-Service 31

MAPE Mean Absolute Percentage Error 69

MGCP Media Gateway Protocol 24

ML Machine Learning 40

MLE Maximum Likelihood Estimation 64

MOS Mean Opinion Score 25

MPLS Multiprotocol Label Switching 26

NN Neural Networks 42

NIST National Institute of Standards and Technology 16

OSI Open System Interconnect 22

QoS Quality of Service 1

PaaS Platform-as-a-Service 31

PBX Private Branch Exchange 20

pdf Probability Density Function 63

PSTN Private Switched Telephone Network 12

RoC-LB Rate of Change-Load Balancer 15

RR Round Robin 54

RTP Real Time Protocol 24

SaaS Software-as-a-Service 31

SDP Session Description Protocol 23

SIP Session Initiation Protocol 23

SMB Small and Medium Businesses 1

SN Super Node 34

SNC Super Nodes Cluster 5

SRV Service Record 55

STaaS Storage-as-a-Service 31

SVM Support Vector Machine 43

SWF Standard Workload Format 97

TCP Transmission Control Protocol 23

UDP User Datagram Protocol 23

UA User Agent 52

UAC User Agent Client 23

UAS User Agent Server 23

URI Uniform Resource Identifier 52

VMA-AdRoC VM-Aware Adaptive Rate of Change 15

VoIP Voice Over Internet Protocol 1

VSC Voice Service Consumer 32

VSP Voice Service Provider 28

VIP Virtual IP 54

VM Virtual Machine 27

Chapter 1

Introduction

1.1 Context

Voice Over Internet Protocol (VoIP) refers to any technology capable of creating voice communications and establishing multimedia sessions over Internet Protocol (IP)[1] networks. VoIP has become one of the most important trend in telecommunications due to lower costs, communication improvement, features and extended functionalities. In the past years, the use of this new way of communication has been increasing greatly with the high number of subscribed users and industry competitors. It has been integrated in many companies and public institutions, the homes of private customers and business markets.

VoIP has been widely adopted as a solution over the traditional circuit-based telephony, Private Switched Telephone Network (PSTN), as a cheaper alternative to analog telephony. In comparison with PSTN, VoIP is user-friendly, it has a simpler connectivity and requires less hardware equipment. However, VoIP infrastructure implementation is a non-trivial challenge when one considers reliability, stability, speech quality. It is a challenging task for service providers to adopt VoIP architecture,f to deliver the same quality of speech as the PSTN with a lower operational cost, lesser investment in equipment and maintenance. Instead of using analog telephone lines, VoIP converts voice signals into digital data packets that are transmitted over the Internet. VoIP enables the use of different services which are not provided by the traditional telephone systems, such as voicemail, video conferencing, call forwarding, music on hold.

Nowadays, the use of Internet became trivial, is used as a primary communication channel and for many individual VoIP customers the most attractive services are the ones provided for free (app-to-app calls, app-to-app texting, video calls). The global demand for VoIP services has been influenced by the availability and the speed of broadband connections, the increasing use of social media, the widespread use of smart phones and tablets.

A large number of providers (e.g. Skype, Google) enable the use of various hangout type applications for which an account to a domain and a connection to the Internet is sufficient. However, there are costs for international, domestic messages, calls for mobile and landline numbers. The market is filled with service providers that offer VoIP solutions with low call rates for local and long distance phone calls. In order to find the provider that suits best their needs, VoIP customers have to identify their precise telecommunication requirements. Customers are in charge of comparing prices, services, features, equipment, in order to chose a VoIP provider. For many businesses it is mandatory to be reachable via phone numbers, landlines being a must.

1.2 Motivation

The evolution of technology in the past years together with the release of new VoIP products and services lead to an increased usage of this type of advanced communication. VoIP became very popular and has been changing the face of business every day. By integrating VoIP in their daily businesses, for corporations of all sizes, costs were greatly reduced. The use of VoIP has changed the market trends and influenced highly the structure of the telecom market structure. One important advantage that results from adopting VoIP is the connectivity of everything: any time, any place and any thing connection. It enables for people the possibility of working remotely from any location, at any time, in collaboration with other entities located in different parts of the world while using video conferencing and VoIP technology. It is a convenient solution in respect to the traditional telephony, it is less expensive and increases productivity of employees. One main difference between the subscription to a local tradition carrier and a VoIP provider is programming one phone number to ring on different telephones located separately (home phone, office phone, cell phone) and the configuration of virtual extensions that handle users' outgoing calls.

According to Future Market Insights (FMI) [1], "In 2012, the corporate consumer segment registered 98.9 billion subscribers, accounting for US$ 43.27 Bn of the global VoIP services market in terms of revenue. FMI forecasts that the number of subscribers in the corporate consumer segment will increase to around 204.8 Mn by 2020, accounting for US$ 86.20 Bn in terms of revenue.". In order to reduce costs for communication services providers are deploying innovative virtualized solutions and are transitioning towards cloud infrastructure models. Cloud computing is adopted by VoIP service providers as a solution to enhance service innovation, to minimize expenses, to transition from the obsolete voice networks. Advanced virtual VoIP systems can be deployed in hosted environments with high quality phone services delivery.

[1] http://www.futuremarketinsights.com/reports/global-VoIP-services-market

This thesis focuses on the development of dynamic optimization based on incoming load analysis and prediction, in order to prevent the overload of the servers in a VoIP system. The information is gathered by inspecting the real system of VoIP provider company and by analyzing the results given by the predictive algorithms which is used to optimize load distribution and resource allocation. Traffic patterns regarding the behavior of users when placing calls are defined. Powerful predictive models are defined and they evolve, adapt and consistently deal with varying factors used for shaping future traffic patterns and capacity planning. Load balancing techniques for a real VoIP system together with the job allocation problem for VoIP in cloud computing, are designed. The benefits of implementing such techniques in the real-life environment are: improvement of the service, reduction of the associated carbon emissions, reduced idle CPU times and optimal exploitation of resources.

1.3 List of contributions

The aim of this thesis is to study the different aspects of VoIP platforms and to improve the VoIP services via a cloud-based solution, with the scope of lowering operation costs at infrastructure level. The main contributions of this PhD thesis are the following:

1. a comprehensive literature study regarding VoIP technologies and environments;

2. addressing congestion and overload issues at the level of a cloud-based VoIP system;

3. analysis of VoIP traffic and obtaining insights about the structure and trends of traffic;

4. state of the art, development and analysis of particle algorithms;

5. proposition of prediction models in charge of anticipating the number of calls in a real VoIP environment;

6. formulation of the load balancing problem addressing VoIP in cloud computing federation and proposal of VoIP load balancing models;

7. formulation of dynamic scheduling of VoIP services in distributed cloud environments and proposal of bi-objective optimization model.

1.4 Dissertation outline

In the current chapter a general introduction to the subject is given together with the context, the motivation, PhD contributions and the outline of the dissertation.

This dissertation is divided into six chapters:

Different characteristics of Voice Over Internet Protocol (VoIP) environments, protocols, QoS, are described in Section 2.1. The paradigm of Cloud Computing (CC) together with its deployment and service models is introduced in Section 2.2. An overview of VoIP solutions is also given in Section 2.2 together with commercial VoIP Communication-as-a-Service (CaaS) solutions and the proposed Infrastructure-as-a-Service (IaaS) model.

Chapter 3 introduces the problem of resource management for VoIP systems. The main domains investigated in the scope of the thesis are load prediction [2], of the incoming traffic, and balancing for cloud VoIP based systems. Well known prediction methods with state of the art in VoIP are detailed in Section 3.1.2. Interactive particle methods, sampling methods based on rare events and on the distribution of the sample from a multivariate normal distribution (Wishart distribution) are also introduced in Section 3.1.2. Load balancing is defined with its basic and advanced aspects with a state of the art in VoIP are presented in Section 3.2.1.

Prediction methods used to anticipate the incoming VoIP traffic during the next time frame is presented in Chapter 4. Interactive Particle Systems is a predictive algorithm introduced in Section 4.1.1 [3]. The design of predictive modeling in VoIP using Gaussian Mixture Model (GMM) and Gaussian Process (GP) is presented in Section 4.1 [4]. The environmental setup with data collection, scenarios used to test the quality of solutions given by each prediction methods are presented in Section 4.2.

Chapter 5 discusses the problem of congestion and overload issues at the level of VoIP cloud-based systems. Several approaches are tested on synthetic data, benchmarks designed out of the logs from a real VoIP system. VM-Aware Adaptive Rate of Change (VMA-AdRoC)[5] is a robust extension of the Rate of Change-Load Balancer (RoC-LB) algorithm, particularly useful for systems that handle VoIP traffic. A site load distribution was built using an Integer Linear Programming (ILP) approach, model that integrates prediction and takes decisions based on the objective of minimizing the number of running machines that handle the placed calls. Different load balancing strategies based on dynamic bin packing were developed [6]. The experimental setup together with practical implementation, results, are presented in Section 5.2.

Chapter 6 concludes the thesis and future perspectives are given.

Chapter 2

Background

2.1 Voice Over IP Technology

Voice Over Internet Protocol (VoIP) is the group of technologies capable of placing calls over the Internet, by transmitting voice data over an IP network. VoIP is being often referred to as Internet telephony, IP telephony, broadband phone service[1]. VoIP service providers focus intensely on the development of new VoIP services, infrastructure with their expansion in the corporate and individual customer sectors. This requires not just a transition from the use of circuit-switched networks to the use of packet-switched networks, but also the replacement of the obsolete traditional telecom networks.

This chapter defines different aspects of VoIP environments and highlights VoIP opportunities and challenges. The integration of a VoIP system in the technology infrastructure of an enterprise has high complexity and the implications for developers and service providers for adopting VoIP are described in this chapter.

2.1.1 Definition of Voice Over IP

VoIP technology is an expanding field fueled by two main objectives: cost reduction and revenue increasing. VoIP is often associated as being a way of carrying phone calls over an IP data network, the foundation for advanced unified communications and new multimedia services (for e.g. video conferencing, integrated contact centers, unified messaging). A large number of industrial experts, network and service providers, researchers try to define this phenomenon.

The National Institute of Standards and Technology (NIST) describes VoIP as a widely trend with major impact in telecommunications[7]. The transmission of voice data takes place over Private Switched Telephone Net-

[1] http://www.voip-info.org/wiki/view/What+is+VOIP

work (PSTN), with a different architecture in relation to traditional circuit-switched telephone networks, introducing security risks and challenges. In the same document it is described how security mechanisms deployed for IP networks must be furtherer developed for VoIP, security problems introduced by the digitized voice transmitted as packets. VoIP increases the complexity of existing network technologies by extending it with components specific for VoIP. For example, in order to build a VoIP system, specific components (software and hardware) with high performance requirements are added to the network. Call processors/call managers, gateways, routers, firewalls, and protocols are just few of the various components specific for VoIP networks whose performance sensitivity have a direct impact on the quality of voice transmitted over the Internet. The services that are usually offered by the traditional telephony systems should be taken in consideration and enabled (i.e. emergency numbers).

CISCO [2] makes a clear distinction between VoIP, IP telephony, IP communications and Unified Communications. CISCO defines VoIP as a method to carry phone calls over an IP network (Internet or internal network) with the benefit of reduced costs generally implied by the use of traditional telephone services. The interconnection of phones together with the billing plan, dialing plans and features (e.g. conferencing, forward, dial to call, hold) are services offered by IP telephony. Business applications in charge of the enhancement of communications features are part of the IP communications domain, and together with the unification, simplification of different forms of communications, use of different technologies, form Unified communications [8].

2.1.2 Towards VoIP

Telecommunications technology has been driven by the evolution of advanced applications and computing platforms. The different means of communications lead to various traffic types (for example voice, data communications, image and video traffic) with different requirements on the network platforms. Convergence occurs in devices (smartphones, smart televisions, smart computers), applications (information services, e-commerce), industries(entertainment, consumer electronics), networks (Internet, broadcast networks, WiFI) leading to new telecom industry trends [9]. It is a common practice for enterprises to share a certain number of external phone lines instead of allocating for each end user a dedicated landline.

Telecommunications technology has faced an evolution and modernization from analog and digital networks to VoIP. The oldest form of transmission of voice in telephony is analog. Since the telephone was invented in 1876, in analog networks human speech is converted into electrical wave

[2] `http://www.cisco.com/c/en/us/index.html`

forms and converted back to speech at the other end. In telephony, the analog wave form is a continuous variable defined by amplitude and frequency, transported between the phones. One of the highest drawback of the analog networks is the accumulated noise implied by the distance traveled by the signal through the network. The basic amplifiers in the analog networks add power to the impaired, noisy signal that arrives at the receiver with high error rates. With the limitations of this early technology and with the growth of the service demand, digital networks have been introduced since 1950. In digital networks, the signal is a series of discrete pulses and contains 0s or 1s that either represent changes in voltage in the case of electrical networks (1 for high voltage and 0 for low voltage) or changes in light levels in the case of optical networks (1 for the presence of light and 0 in the absence of light). The digital signal suffers impairments and loses power when traveling over the network. However, signal regenerators are used to examine if the weakened signal was a previous zero or one, and regenerates a new signal that is furtherer transmitted. With this mechanism, the noise is virtually eliminated and the error rate is reduced. One of the limitations digital networks have is that the circuit-switched connections imply the use of a dedicated channel for the duration of a call which consumes the same amount of bandwidth, 64Kbps of data, during a silent pause.

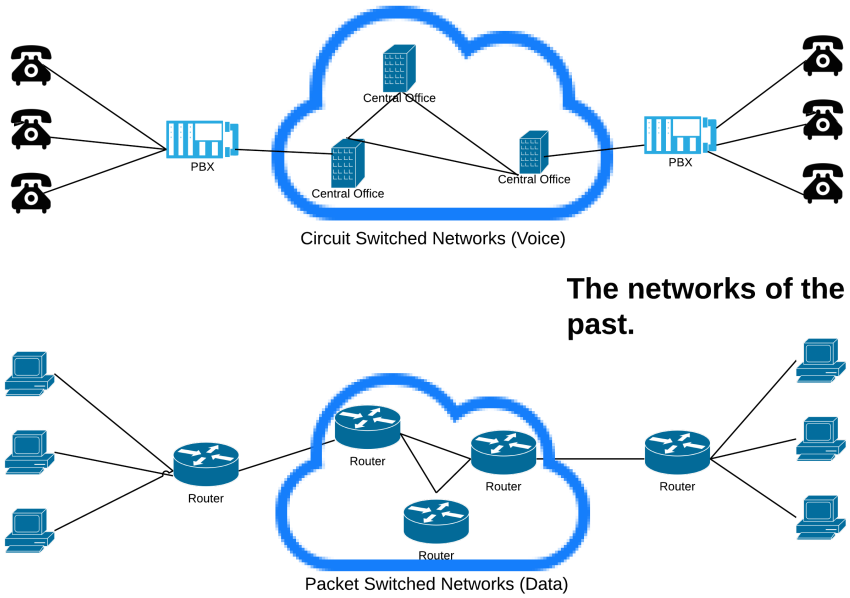


Figure 2.1: Separated Networks. Separated applications and services.

In the past there was a clear separation between the functionalities of circuit-switched networks and packet-switched networks that handled calls and data, respectively. In Figure 2.1 there is displayed an example of the

networks of the past, with voice networks using circuit switching and data networks using packet switching. In circuit switched networks there is a dedicated path between the origination and destination of a call, while in data networks there is no dedicated path. The cost for placing calls in circuit switched networks is based on the distance and duration. The use of packet-switched networks for IP-based calls has increased due to the efficiency and low cost of data networks. A channel in a packet-switched network is not dedicated to one user and the message is split into data packets, sent over the network and reassembled at the destination node [9] [10]. VoIP is driving the convergence of data, video, voice using IP-based networks. In Figure 2.2 there is an example of converged networks with separated or integrated applications [3].

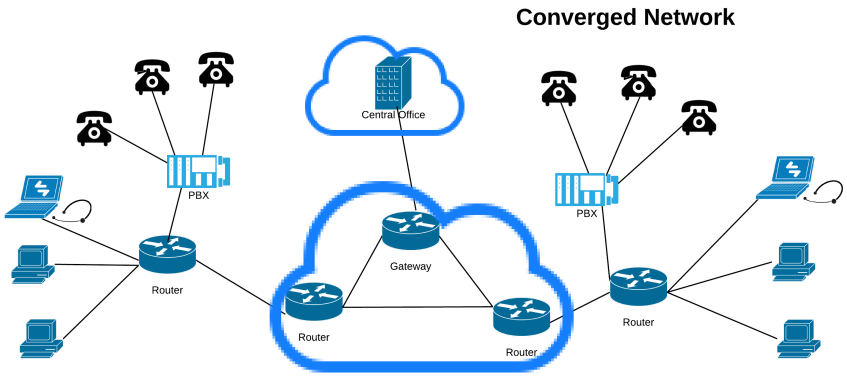


Figure 2.2: Converged Network. Separated or integrated applications.

The origination of VoIP phone calls mirrors the steps followed by the traditional digital telephony, namely signaling, channel setup, digitization of the analog voice signals, encoding and decoding methods of the speech. Audio signals from the phone are converted into digital data and transmitted using an Internet connection [11]. The first VoIP technical solutions mirrored the architecture of the traditional telephone networks, and has evolved to closed networks for private business with free IP calls and low prices for the access of the other communication networks while moving furtherer to the concept of federated VoIP. In Figure 2.3 there is an example of VoIP architecture where devices can either use Internet to place phone calls to other devices (primary link) or the public switched telephone network to place calls to the traditional telephones (secondary link) [4]. When the primary link is used, the voice is converted into digital format and is divided into packets transmitted over the network and converted back to analog at

[3] `http://www.slideshare.net/habib_786/voice-over-ip-voip-26670219`
[4] http://brightmags.com/how-does-voip-work/

the receiver. For the secondary link, when a call is placed from a VoIP client to a traditional telephone, the conversion of the data takes place before it reaches the PSTN.

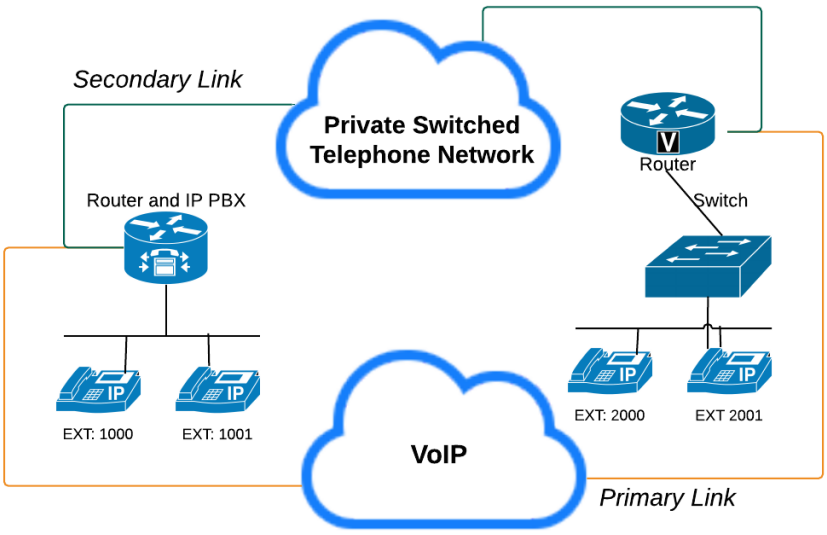


Figure 2.3: Voice Over IP Architecture example.

Within an enterprise it is possible to exchange calls on local lines and to share a certain number of external phone lines that connect to the PSTN. In order to enable this type intercommunication, a Private Branch Exchange (PBX) telephone system can be implemented. A PBX can handle the circuit switching locally, manage the connection and select outgoing lines automatically, without requiring an operator. When a PBX is used phone lines are shared between extensions and incoming calls are routed to the appropriate extensions or holding queues, based on the predefined dial plan. A PBX system is the device that connects and transfers the placed calls from an enterprise to the PSTN. Lately, the term PBX is used for complex in-house telephone switching systems. One of the most successful Private Branch Exchange (PBX) used nowadays is Asterisk [12].

Asterisk [5] is a framework for building multi-protocol, real-time communications solutions that provides a power control over call activity. It establishes and manages the connection between end devices, by sending the voice portion of the call and everything that is not voice, known as overhead. Figure 2.4 displays few of the most important clients that use Asterisk.

[5] `https://www.digium.com/`

Figure 2.4: Digium - Business Phone Systems based on Asterisk.

Voice over IP PBX also named IP-PBX was developed to connect individual extensions over the Internet. It is possible to combine a hosted PBX with a standard in-house PBX and as such, customers can have enabled connectivity when the Internet connection is down. Integrated Service Digital Network (ISDN) [6] is and out-of-band signaling protocol separates the channels for the overhead and the audio portion of the call, enabling the use of the traditional telephony carrier when the Internet connection is down.

An example of a telephone system solution for VoIP is given in Figure 2.5. The users connect via Internet to a server that runs either on a physical machine or in a data center or in the cloud. All the users must be registered to a Registrar Server in order to indicate their current IP address and the URLs used to receive calls. The security layer can include SSH, FTP and access control or sessions for password protected phone login. Voice nodes control call features such as voice mail, call transfer, conference function and these solutions are mainly deployed on LINUX distribution. After authorizing the user to place calls, two links are made. First, the user calls Asterisk (voice node) that connects to the database server in order to check the credentials of the user, e.g. if the call can be placed (e.g. credit on a pre-pay account). Second, Asterisk tries to reach the other end-device. After the call is finished, when the user hangs-up, call details are stored in the Call-Detail-Record (CDR) (e.g. id client, destination, prefix, duration call).

The situation presented above stands for outbound calls placed by the users of VoIP services. VoIP providers are in charge of call forwarding to different telecom operators. For incoming calls, the carrier connects to the VoIP provider which, in turn, looks up for the number and returns the number and location where the user can be reached.

Enterprises and companies adopt IP telephony as a solution where channels are carried over the Internet connection, voice and data are unified on a common infrastructure, and extra features are provided for free. It is a common practice to keep the subscriptions to trusty legacy phone system for dedicated lines, due the built-in security, emergency location services, reliability (hardwired landline phones during power outage are active).

[6] http://www.jet.net/isdn/isdnintro.html

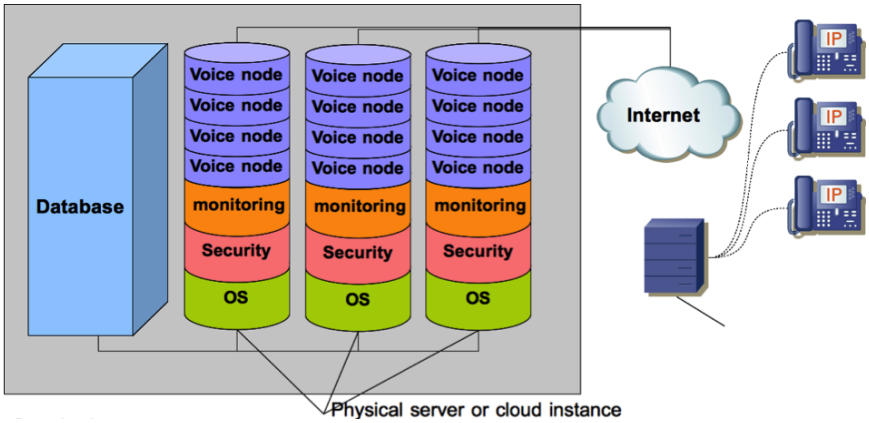


Figure 2.5: Distributed VoIP architecture.

2.1.3 Overview of Internet telephony communication protocols

VoIP systems have two main functionalities: establishing connection between end devices and transmitting the voice portion of the call. Signaling refers to the localization of the users, setting up or tearing down sessions between devices. The protocols have specific roles in the call-setup process and real-time sensitive data transmission[13]. In VoIP an end device can be either an user agent or application, clients are the entities that originate calls and servers that handle incoming calls. The end point in a VoIP connection can be either a softphone (desktop or smartphone applications) either a hardphone (IP phone). An IP phone is physical device that connects to the network and has the voice portion, referred to as media or payload, and everything that is not voice and is transmitted, e.g. caller ID, connection and disconnection for billing. A phone system is composed of the following parts:

- hardware components: telephones, cables from the telephone to the phone system and back, cables from the phone system to the Internet, the box where the PBX software runs, for example: Asterisk; the interface card that accepts the cable from the phone; optional ports to connect to the local provider or long-distance carrier;
- the telephone signaling: transmission and receipt of the phone calls; originating numbers; conference bridging;
- the features: the call plan, click-to-dial, call forwarding, call transfer, call waiting etc.;

In order to describe VoIP networking and different parts of the data communication process, the Open System Interconnect (OSI) can be used

as a reference model. The OSI model simplifies the connections between different types of networks regardless of the internal structure and technology [14]. OSI model is used to partition a communication system into seven abstraction layers and can be used to describe the VoIP networks in the following manner [15]:

1. *Physical layer*: copper twisted-pair, coax cables, connectors, fiber-optic cables, plugs, power sources etc.;
2. *Data link layer*: basic foundation for the transmission of the packets through Local Area Network (LAN);
3. *Network layer*: datagrams and IP addresses implemented at this layer must function in order for the VoIP applications to work;
4. *Transport layer*: User Datagram Protocol (UDP) [16] and Transmission Control Protocol (TCP) [17] are encapsulated within IP and are used for data transmission, end-to-end connection and flow control;
5. *Session layer*: Session Description Protocol (SDP) [18] is in charge of managing the structure for communication between applications, coordinating the connection of media and negotiating the media format of a VoIP call;
6. *Presentation layer*: it houses data representation and encryption;
7. *Application layer*: provides services for network management for applications such as: telephone calls, conference calls, voice mail etc.

Session Initiation Protocol (SIP) [19] is a signaling communications protocol developed by the Internet Engineering Task Force (IETF) standards, widely used for controlling multimedia communication sessions such as voice and video calls over IP networks. The main features of SIP are: call on hold, call forward, third party call, conference, click-to-dial, call setup delay, capability exchange, packet loss recovery, fault tolerance, hierarchical namespace of features or error codes, transparent proxying, supported/required options protocol elements. A SIP network consists of SIP Nodes, Registrar Server, Proxy, Location Server. SIP Nodes are the devices that interact with a call generated using SIP protocol, softphones or hardphones. User Agent Client (UAC) is the entity that creates SIP requests and User Agent Server (UAS) is the entity that generates a response to the SIP request (e.g. acceptance, rejection, redirection of the request). SIP Registrar Server is a database server that stores the initial location of the callee's phone. SIP Proxy manages the calls between the outside world and LAN, resolves the username into an address which is forwarded to the appropriate end point. SIP Location Server provides information regarding the location of the resources within the network, and it stores the location of devices that perform

registration. SIP methods define how a call is handled in the network, for example: INVITE, REGISTER, BYE. The majority of SIP implementations are built on top of the TCP. In Figure 2.6 there is a description of a SIP call flow.

H.323 [20] is a VoIP signaling protocol developed by the International Telecommunication Unit (ITU-T), that includes specifications for audio and video communications across packetized networks. An H.232 network consists of several endpoints, a gateway, optionally a gatekeeper, Multipoint Control Unit and Back End Service. The gateway connects the network with the outside world, that can be either a SIP network or PSTN. The gatekeeper is in charge of managing the bandwidth and the address resolution. When it is part of the network, the Back End Service maintains the information (e.g. services, configuration) about endpoints. Multipoint Control Unit is an optional component in a H.323 network, in charge of facilitating the communication between more than two endpoints. H.232 encompasses a suite of media control recommendations provided by the ITU-T, that have different roles in the call setup process.

Real Time Protocol (RTP) [21] is a protocol standardized by IETF used for transmitting and receiving media, inside VoIP calls, in real-time. After the connection is established, the media transportation is done via RTP. COder/DEcoders (CODECs) are used for converting the voice portion of a call in audio packets and the conversation is transmitted over RTP streams. The majority of the RTP implementations are built on top of the UDP. RTP is often used in conjunction with SIP.

Media Gateway Protocol (MGCP) [22] is a protocol developed by IETF with signaling and call control communication functionalities. It controls the media gateways on IP networks connected to PSTN. The media gateway is a network element that converts the audio signals carried over the telephone circuits and the data packets carried over packet networks. For example, a trunking gateway manages a large number of digital circuits because it bridges the telephone network with the VoIP network. MGCP manages calls and conferences, a master/slave protocol implemented between the endpoints (media gateway) and the servers (call agents).

2.1.4 VoIP performance and Quality of Service

In comparison with data networks, VoIP networks have stricter constraints and are more sensitive to delays. The quality of calls can be affected by several factors, also determined by the transit of the packets across the Internet, queuing delays at the routers, packet travel time from source to destination. Quality of Service (QoS) [7] comprises requirements on all aspects of a call, such as: service response time, throughput, loss, call set-up time,

[7] `http://www.cisco.com/c/en/us/td/docs/ios/solutions_docs/qos_solutions/QoSVoIP/QoSVoIP.pdf`

resource utilization. The quality of voice is subjectively perceived by the listener. A common benchmark used to determine the quality of sound is the Mean Opinion Score (MOS) [23] and the scores are listed in Table 2.1.

MOS value	**Quality of speech**	**Impairment**
5	Excellent	Perfect, no effort for understanding
4	Good	Fair, attention is required
3	Fair	Annoying, moderate effort is required
2	Poor	Very annoying, nearly impossible to communicate
1	Bad	Impossible to communicate

Table 2.1: MOS scores

Codec	**Bit Rate**	**IP Bit Rate**	**MOS score**
G.711 [24]	64 Kbps	87.2 Kbps	4.1
G.729 [25]	8 Kbps	31.2 Kbps	3.92
G.723.1 [26]	6.4 Kbps	21.9 Kbps	3.9
G.723.1 [26]	5.3 Kbps	20.8 Kbps	3.8
G.726 [27]	32 Kbps	55.2 Kbps	3.85
G.726 [27]	24 Kbps	47.2 Kbps	3.85
G.728 [28]	16 Kbps	31.5 Kbps	3.61
iLBC [29]	15 Kbps	38.4 Kbps	4.14

Table 2.2: Codec information and bandwidth calculation.

An important role in determining the quality of a VoIP is played by the CODEC, an acronym for COder/DECoder. Codecs are algorithms that convert the voice portion of the calls into packets transported over the network. The audio or video signal is compressed at the sender and decompressed at the receiver, in order to save the bandwidth utilization. VoIP phones typically support different codecs, and the use of a specific one is negotiated between the communicating devices. Codecs provide different quality of speech and MOS is used to asses the quality of it. In Table 2.2 the most common codecs with their bit rates [8], the actual IP bit rate used [9], and

[8]The number of bits per second that need to be transmitted in order to deliver a voice call, theoretical usage of bandwidth.

[9]Bandwidth expanded with UDP/IP headers. Nominal Ethernet Bandwidth. (one direction).

their MOS score are presented [10] [11].

Some codec compression techniques require more processing power than others. An example of compression methods is presented in Table 2.3 [30].

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

Table 2.3: Codec Compression Method

The performance and the QoS metrics of a VoIP network consist of latency, jitter, packet loss [31]. VoIP has a low tolerance for disruption, packet loss and it is challenging to provide the same quality of call setup and voice relay functionality as the traditional telephone network [12]. Latency is the time required to transmit voice, from source to destination, and its value should be kept as low as possible with the upper-bound for one way traffic of 150 ms. Jitter refers to non-uniform delays and often is caused by low bandwidth. These variations in delay are detrimental to the overall QoS due to the fact that when packets arrive they are processed out of sequence. The maximum allowable duration of jitter is 100 ms before deterioration occurs. There are different strategies of implementing a buffer that controls the jitter in order to determine when to release the voice data. Packet loss occurs when VoIP packets are not delivered at all, it can result from excess latency or jitter, when late arrived packets are discarded or when the surrounding packets have been released from the buffer. Packet loss in VoIP should be less than 1%, to avoid audible errors [13] .

The sensitivity of a VoIP network's QoS to all these parameters requires that the enterprise's hardware supports QoS for VoIP and can deliver the traffic at high speed with preference over the other data traffic. QoS guarantees must be provided for signaling and data traffic in VoIP. There are different mechanisms and technologies for the provision of QoS [32]. For example, Multiprotocol Label Switching (MPLS) [33] is a routing mechanism that guarantees a path through the network, from source to destination, and reserves bandwidth for different traffic classes. Differentiated Services (DiffServ) provisions QoS by guaranteeing per-hop behavior of network traffic[34]. In general, QoS standards for VoIP traffic are set for the

[10] `http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7934-bwidth-consume.html`

[11] `http://www.itu.int/net/itu-t/sigdb/speaudio/Gseries.htm`

[12] `http://www.voip-info.org/wiki/view/QoS`

[13] `https://www.sevone.com/content/guide-ensuring-perfect-voip-calls`

delivery quality of the voice. Each codec provides a certain quality of speech when the server is under normal conditions in order to ensure QoS and performance. For example, the packet loss rate is inverse proportional to the MOS.

Excellent indicators used to determine if the server is under normal conditions are CPU load, disk utilization, server response time, available memory. Effective load distribution methods can be used to distribute the load across the servers. Generally, overloaded servers lead to deteriorated performance. A server can be kept under normal load by setting thresholds of the indicators, followed by using a Load Balancer (LB) that applies a load-distribution method to distribute the load across the real servers.

In [35] the results of performance and stress testing of SIP servers, SIP clients and IP networks, are presented. While different scenarios are considered, it was observed that there are bursts in CPU load and that peaks occur even without processing any calls. They show that CPU load peaks and big RTP delays can be caused by processes that are not related to VoIP. Even though there is no straightforward relationship between call quality (max jitter) and simulated call load, RTP processing is sensitive to the execution of other tasks performed on Linux server. Maximum jitter depends on the number of calls observed and mean jitter increase with number of simultaneous calls.

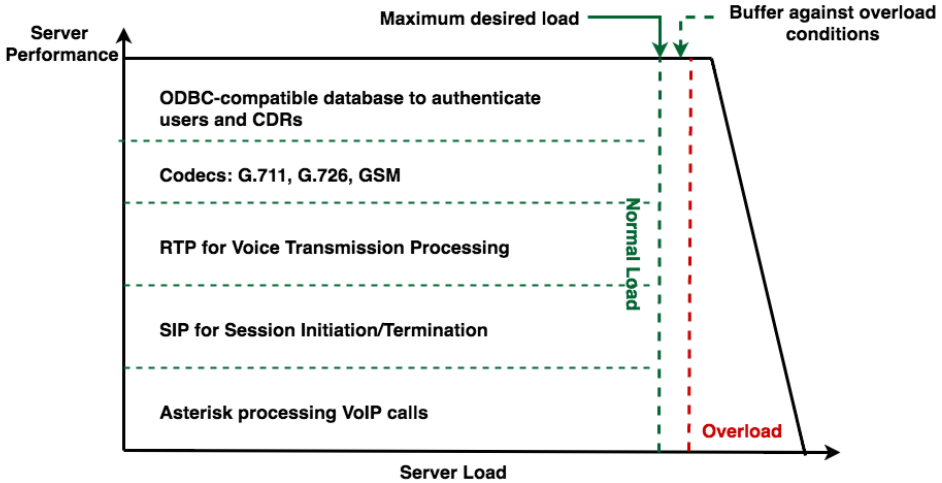


Figure 2.7: Optimizing server performance for VoIP.

In order to avoid delays caused by overloads of CPU and network stack or file system operations, to ensure QoS for the Virtual Machine (VM) where Asterisk is running, the utilization allocated for VoIP calls will be kept under a certain threshold (e.g. 70%). VoIP applications are CPU intensive when transcoding is performed and it is mandatory to have enough resources to successfully process each step required to deliver phone calls with high

QoS. Therefore, the processes involved in the operation need priority. The priority level of a process that runs frequently can be assured by restricting process CPU usage and using different techniques to distribute resources depending on the needs of the application. The Kernel of an Operational System cannot determine what CPU processes are important and it is in the responsibility of the Voice Service Provider (VSP) to define the priority. The internal scheduler can be configured using *nice command* to define a tasks priority, *cpulimit command* to pause processes to not exceed a certain limit or *cgroups command* to limit the amount of resources available to the process [14]. Availability is the most important requirement for VoIP applications since without enough resources, the incoming requests can not be handled. In the context of this thesis, processes that have priority to resources to ensure QoS are the ones that deliver qualitative phone calls: the signaling and termination process (SIP), the transmission of voice over the network (RTP), the conversion of voice signal into data packets (CODEC) (Figure 2.7).

2.2 Cloud Computing and VoIP in Cloud

Cloud Computing (CC) can be seen as an overlay where seamless virtualization is implemented while dealing with privacy constraints. It relies on sharing computing resources rather than on having local servers or personal devices to handle applications. Cloud computing is used to increase capacity or add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software. As a specific aspect, one may consider a computational demand and offer scenarios, where entities (individuals, enterprises, etc.) negotiate and pay for access to resources through virtualization solutions (administered by a different entity that acts as provider) [36].

2.2.1 Definition of Cloud Computing

CC is a hot topic in Information and Communication Technology (ICT), with increased popularity and a significant influence in the computing industry, that reshaped the ICT market. Multiple definitions are standardized and a wide number of researchers and industrial experts are involved in shaping this phenomenon. The International Organizations for Standardization (ISO) together with the International Electrotechnical Commission (IEC) defines Cloud Computing as [37] "a paradigm enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning and administration on-demand.". The National Institute of Standards and Technology (NIST) defines CC in the

[14] `http://blog.scoutapp.com/articles/2014/11/04/restricting-process-cpu-usage-using-nice-cpulim`

following manner [38]: "Cloud Computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". NIST also outlines the five essential characteristics of CC: on-demand self-service, broad-network access, resource pooling, rapid elasticity, measured service.

CC encompasses any subscription-based or pay-per-use service that, in real time over the Internet, extends IT's existing capabilities [39]. Demanding parties may however have diverging requirements or preferences, specified by contractual terms, e.g. stipulating data security, privacy or the service quality level [40] [41]. Moreover, dynamic and risk-aware pricing policies may apply where predictive models are used either in place or through intermediary brokers to assess the financial and computational impact of decisions taken at different time moments. Legal enforcements may also restrict access to resources or data flow, e.g. data crossing borders or transfers to different resource providers. As common examples, one can refer to Amazon Web Services [42] or Google Apps Cloud Services [43].

Five important characteristics of cloud computing are cost, performance, scalability, mobility, and virtualization, described in the following:

1. *Cost*: in cloud computing the resources do not belong to the users which do not have to buy or maintain them, the initial investment is not needed;

2. *Performance*: improving the processing power, maximizing storage capacity by consolidating of CPUs, memory and storage to deploy services;

3. *Scalability*: the user can increase or decrease resources (storage, CPUs, memory, etc.) at any time, the billing depending on the pricing model;

4. *Mobility*: the data can be accessed anytime anywhere, remotely via Internet (using a laptop, smartphone, etc.);

5. *Virtualization*: a single physical resource appear as many individually separated virtual resources. It allows the use of the server capacity effectively, reducing unused CPU cycles, and minimizing wasted energy.

The need for resource capabilities arises as the cost-saving benefit of dynamic scaling is brought by the cloud phenomenon, given that cloud computing uses virtual resources with a non eligible setup time. Prediction is necessary and statistical models that describe resource requirements in cloud computing already make the object of pay-as-you-go, e.g. providers of environments that scale transparently in order to maximize performance while minimizing the cost of resources being used [44]. By predicting the load,

allocating tasks to the processors, dynamically turning off reserve computational resources, the high power required by cloud computing system can be reduced drastically [45].

2.2.2 Cloud Computing Deployment and Service Models

Cloud computing considers four deployment models based on the location of the clouds: *Public*, *Private*, *Community*, and *Hybrid*. Public Clouds are managed by their providers. The infrastructure is shared between organizations and users grant access to resources through subscriptions. Private Clouds can be accessed only by the provider of the resources; the cloud is fully owned by a single company with total control over the applications running on the infrastructure. Hybrid Cloud combines private with public cloud and user applications run either on a private either on public infrastructure (applications with relative importance are scheduled in private clouds). Community Clouds allow sharing infrastructure between organizations with common concerns or similar polices [38].

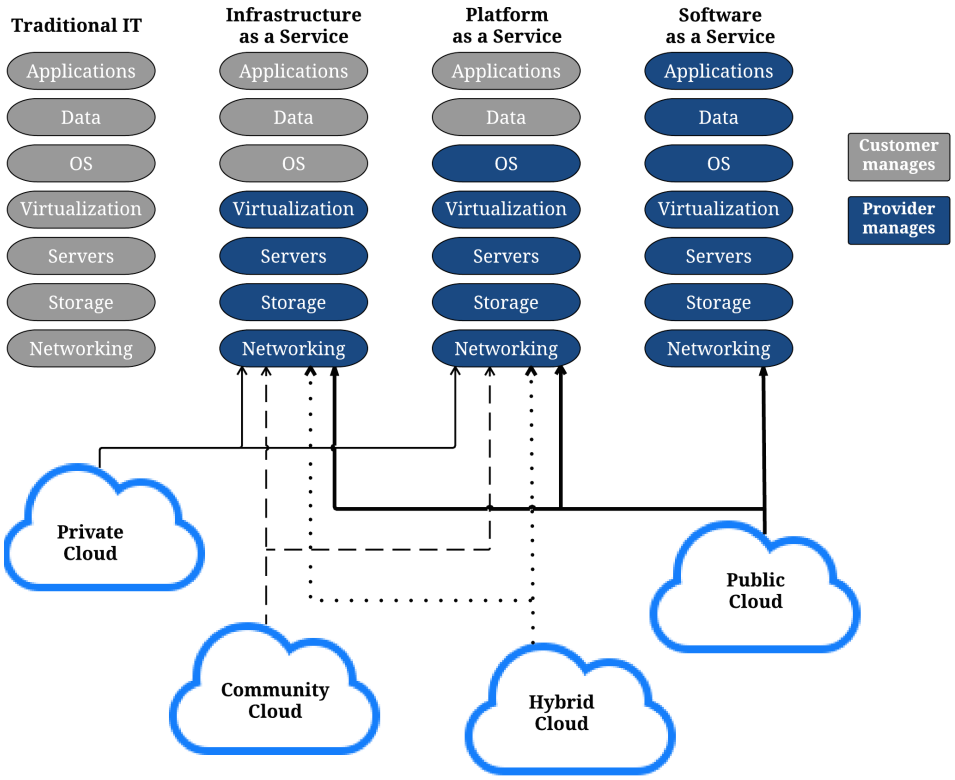


Figure 2.8: Cloud Computing Service and Deployment Models.

CC providers offer services through different models, often referred to as X-as-a-Service. The most common levels of Cloud Services are Software,

Platform and Infrastructure as a Service (Figure 2.8). *Infrastructure-as-a-Service (IaaS)* involves offering computational resources including processing, disk storage, network and other computational resources, case in which users have full control over software, storage and processing capacity. *Platform-as-a-Service (PaaS)* involves offering a development platform in the cloud and user deploy applications with possible configuration settings for their applications. *Software-as-a-Service (SaaS)* includes software offered by cloud providers and users rent software applications running on a demanded infrastructure, being able to change the application configuration settings.

With the evolution of the cloud market, new service models have emerged throughout the business landscape. *Business-Process-as-a-Service (BPaaS)* [15] involves delivering business process outsourcing to the cloud [46]. Big Data and Internet of Things (IoT) are two examples of technology trends for which the *Storage-as-a-Service (STaaS)* [16] is relevant; for example enterprises rent storage space on a cost-per-gigabyte-stored and cost-per-data-transfer basis. *Hardware-as-a-Service (HaaS)* is a service provision model with on-demand access to fully-configurable hardware [17] . *Metal-as-a-Service (MaaS)* involves a dynamic provisioning of hyperscale computing environments [18]. *Communication-as-a-Service (CaaS)* is a service leasing model with outsourced enterprise communications solutions [19] .

2.2.3 Voice over IP in Cloud

VoIP is the technology that enables the placement of phone calls using a broadband Internet connection, with reduced call rates. Traditional VoIP solutions cost less than PSTN services, due to the use of a single network that carries both data and voice traffic. As previously mentioned, telephone systems consists 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.

Traditional VoIP solutions must support the existing telephone system and should be prepared for a growth and expansion of the customer's needs. Over-provisioning and overcapacity are mechanisms adopted by VoIP providers in case the number the customers grows significantly, in order to be able to deliver services during peak hours. A drawback of this architecture arises when the hardware reaches its maximum capacity. A cloud based VoIP not only further reduces costs but adds new features and capabilities,

[15] http://www.gartner.com/it-glossary/business-process-as-a-service-bpaas/

[16] http://searchstorage.techtarget.com/definition/Storage-as-a-Service-SaaS

[17] http://searchitchannel.techtarget.com/definition/Hardware-as-a-Service-HaaS

[18] http://www.webopedia.com/TERM/M/metal-as-a-service_maas.html

[19] https://www.techopedia.com/definition/16031/communications-as-a-service-caas

provides easier implementations and integrates services dynamically scalable. Scalability and quick provisioning are example of advantages that CC brings, and in the context of the thesis, the focus is on Small and Medium Businessess (SMBs) that are Voice Service Providers (VSPs), organizations, that grow rapidly and need to deal with temporary infrastructural needs. The Cloud Service Provider (CSP) owns the equipment and is responsible for the running and the maintenance of it.

For the purpose of this work, a clear distinction is done between the VoIP Cloud Service Customers (CSCs) in the following manner. A Voice Service Consumer (VSC) adopts VoIP as a solution for his business in order to reduce the monthly bills implied by the use of the traditional business phone system. It implies either outsourcing fully to the services offered by a Voice Service Provider (VSP) either adopting CaaS as a solution. A Voice Service Provider (VSP) can use CaaS to resell solutions. The VSP which is a full service communications system integrator specialized in VoIP, can adopt IaaS as a solution (Figure 2.9).

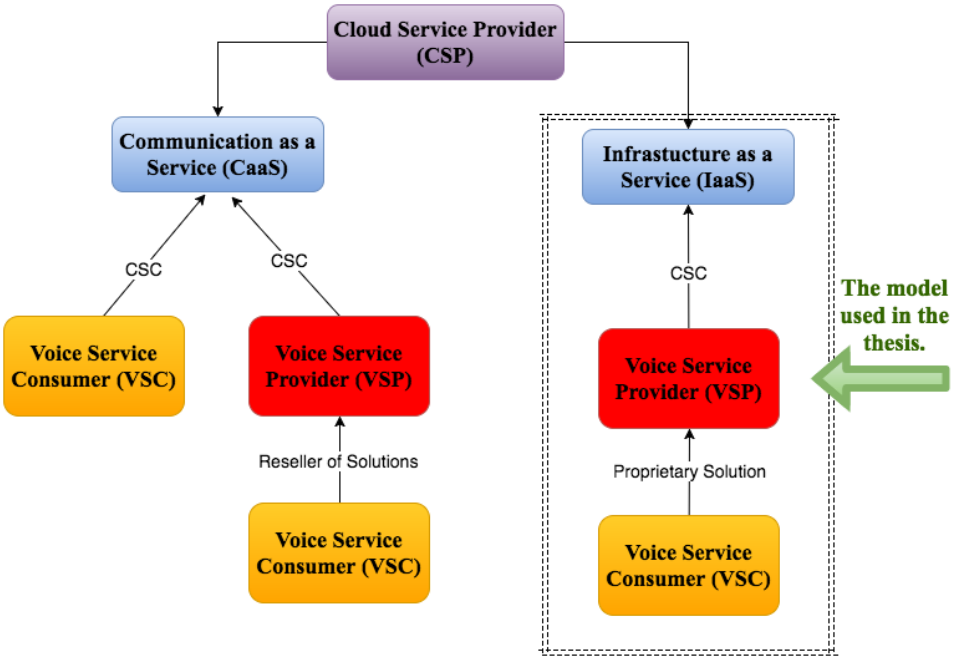


Figure 2.9: CSP, CSC, VSP, VSC: Flow Description.

2.2.3.1 VoIP CaaS

Communication-as-a-Service (CaaS) can be seen as a branch of Software-as-a-Service (SaaS) that facilitates business communications and includes solutions for Internet telephony with instant messaging, videoconferencing applications. CSP offer products and services over the Internet while guar-

anteeing QoS and managing the hardware and software. The market of CaaS is growing since it is reducing the overhead of businesses and it is optimizing the business processes. VSCs are responsible to find the best solution for their business, to consider the current infrastructure technology and growth of the business. It is recommended to determine the cost of choosing a low upfront subscription with monthly fee per user in comparison with the cost of developing a cloud phone system. Cloud based VoIP solutions bring scalability, flexibility and reliability. In Table 2.4 there are presented Communication-as-a-Service (CaaS) models with their characteristics, and reference to commercial providers.

CaaS Models	**Characteristics**	**VSP Examples**
Virtual PBX	- partial voice communication system; - supports inbound calls; - solution adopted by SMB that have remote teams; - challenge: the provision of a system that supports outbound calls.	Titan VoIP[20] Multitel [21] Mobex [22]
Hosted PBX	- full voice communication system; - supports both inbound and outbound calls; - VSP owns the IP-PBX server, that handles the signaling, calls and features; - VSC programs the features, network maintenance, staff training; - VSP handles the new feature installation, upgrades of the IP-PBX; - VSP ensures the possibility to forward calls ; - it involves purchasing IP phones, a router, a network switch dedicated to VoIP.	1Pipe Telecom [23] VOX Connect [24] Cebod Telecom [25]
On-premise PBX	- full voice communication system; - enables VoIP and the routing of calls through the traditional telephone system; - VSP programs the features, network maintenance, staff training; - it involves purchasing servers with interface cards for the connection with PSTN, IP phones.	VOX Connect New-Tel[26] Freedom Voice[27]

Table 2.4: Description of CaaS models and characteristics.

2.2.3.2 VoIP IaaS

IaaS provides the capability for CSCs to deploy and run operating systems and applications. VoIP requires service availability at all times for any

[20] `http://www.titanvoip.com/`
[21] `https://www.multitel.net/pbx/`
[22] `http://mobex.biz/phone-numbers-pay-go/`
[23] `https://www.1pipe.com/`
[24] `http://www.voxconnect.com/`
[25] `http://www.cebodtelecom.com/`
[26] `http://en.new-tel.pro/`
[27] `https://www.freedomvoice.com/`

number of users. To deal with the increasing number of clients providers invest in a large infrastructure in order to avoid loss of calls (hence, users). In this case, the infrastructure is usually underutilized and servers resource degradation occurs. Infrastructures are easily scalable in IaaS with resources distributed as a service and dynamic scaling. When significant spikes occur in VoIP, the VSP deal with the infrastructure demand in order to provide QoS for VSCs.

In cloud-based VoIP solutions voice nodes are operated as VMs that provide a variety of services. Distributed cloud based VoIP architectures assume that voice nodes are distributed geographically; hence, they are grouped in different locations (data centers). For an effective deployment and effective voice traffic management via clouds, characteristics of the resources are to be considered. The most important is the utilization of the infrastructure. The advantage of this architecture consists in increased scalability and low cost. With several unsolved problems such as the optimization of the overall system performance, processor utilization has to be high leading to a reduced quality of the call. Thus, the load of the VoIP servers should be kept under thresholds to guarantee QoS. On the other hand, the processor idle time increases the useless expenses of the cloud provider.

In the context of the thesis it was developed the concept of the supernode (Super Node (SN)) [28] and Super Nodes Cluster (SNC) to enrichment features for telephone exchanges (Figure 2.10). SNC is a set of SNs deployed in a cloud interconnected logically at a local level. This enables the minimization of the path between two local users and increases the quality of the voice. This deployment brings redundancy on a given location and ensures the delivery of high voice quality. SNCs are deployed in widely distributed geographical locations. As shown in Figure 2.10, 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 usually place phone calls locally or in neighboring countries. The deployment of this architecture allows providing services with the quality ensured by ISDN, via 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.

[28]Supernode2 - 2012/2014. Supernode 2 to build cloud-based solution for VoIP environments. Professor Dr. Pascal Bouvry, Development Project with MixVoIP, University of Luxembourg and Ministry of Economy of Luxembourg.

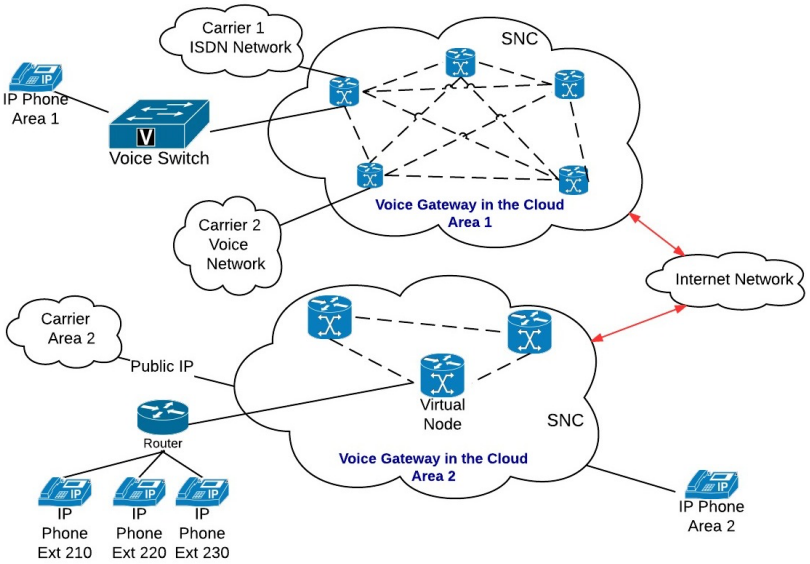


Figure 2.10: SNC deployment.

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

In the typical cloud scenario, a VoIP provider can choose between different resources 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, in the context of the thesis it is considered a metric that regards the hourly billing of VM. It allows providers to measure the cost of the system in terms of number of demanded VMs and time of their using.

In this thesis, it is formulated and discussed the model for job allocation problem addressing VoIP in cloud computing. Providers face challenges to use infrastructure in the best efficient and cost effective ways. Models for the provider cost, quality of service are defined and call allocation techniques are proposed. Prediction, efficient scheduling and load balancing algorithms

are the fundamental parts of this study, with respect to uncertainties of dynamic environments.

2.3 Summay

In this chapter it is described the Voice Over Internet Protocol (VoIP) technology together with evolution of such advanced applications and their corresponding platforms, from separated to converged networks and to VoIP architecture. There are defined the main functionalities of VoIP systems together with an overview of the specific Internet telephony communication protocols. Due the fact that VoIP networks have strict constraints and are sensitive delays, performance and QoS metrics are presented. Cloud Computing (CC) is presented from a general perspective with the service and deployment models and furthermore, in relation with VoIP. VoIP Communication-as-a-Service (CaaS) and IaaS are the CC service models analyzed in terms of different aspects and also current solutions provided by commercial market. It is also introduced the service model, IaaS, used to formulate the problem of dynamic scheduling of VoIP services in distributed cloud environments for optimization, presented in Chapter 5.

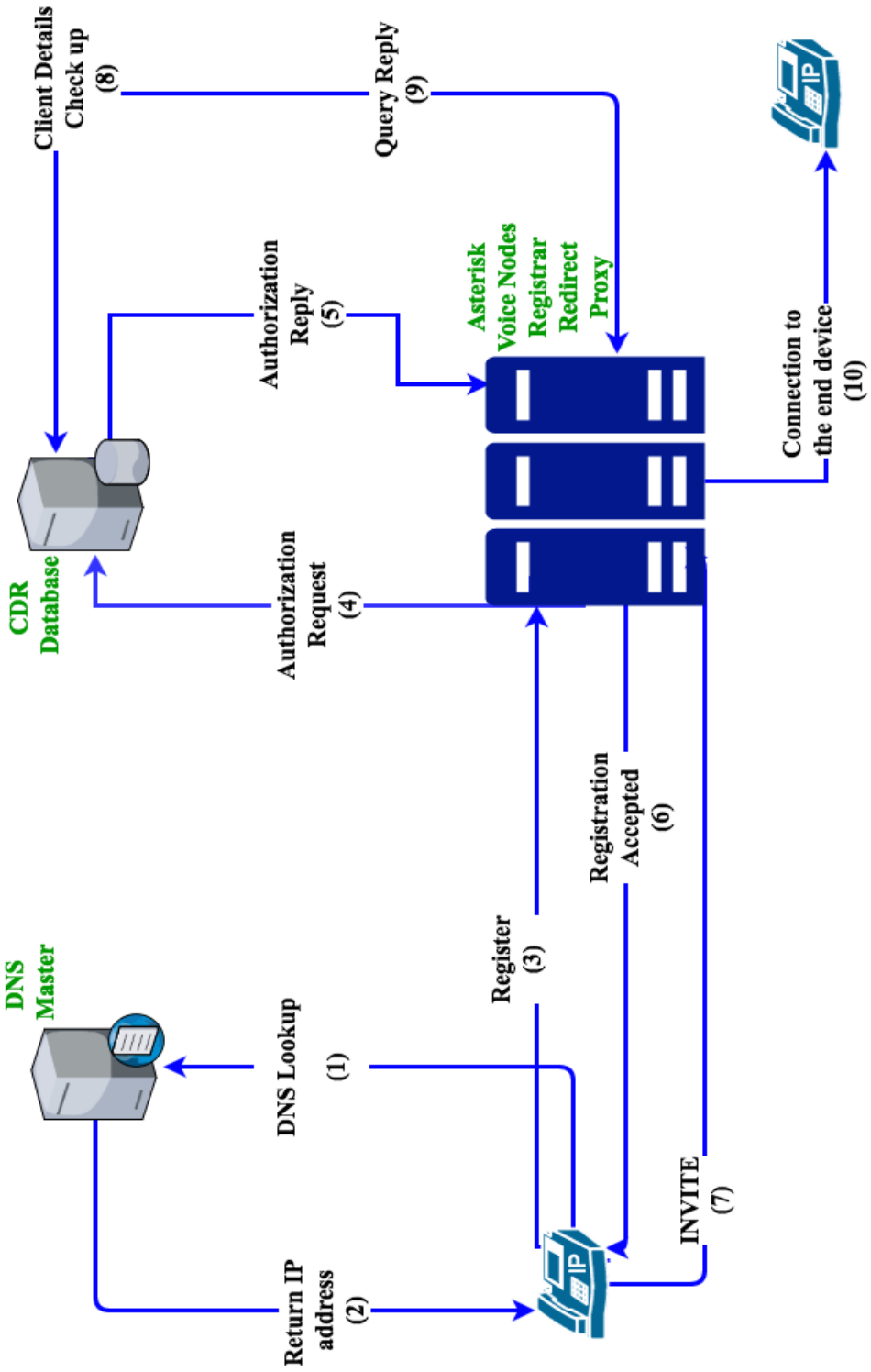


Figure 2.6: SIP call flow logic example.

Chapter 3

Resource Management for VoIP

An important problem one needs to deal with regarding VoIP systems is server overload. Dynamic optimization, based on incoming load analysis and prediction, is an innovative approach to prevent the overload of the servers in a VoIP system. One way to restraint such problems is to rely on prediction techniques designed for the incoming traffic, namely as to proactively scale the available resources. Anticipating the computational load induced on processors by the incoming requests is used to optimize load distribution and resource allocation.

The development of effective dynamic VoIP scheduling solutions involves important issues: load estimation and prediction, load levels comparison, performance indices, system stability, job resource requirements estimation, resource selection for job allocation. Adaptive solutions are required to lower operational costs at the infrastructure level. The main domains investigated in the scope of the thesis are: load prediction of the incoming traffic, allocation of tasks to processors and provision of computational resources in order to ensure Quality of Service (QoS).

3.1 Load Prediction

In the scope of this thesis there are addressed congestion and overload issues at the level of cloud-based VoIP systems. A direct aim consists in developing predictive algorithms, capable of anticipating the computational load induced on processors by the incoming requests.

The implementation of prediction in real-life environments leads to improved delivered services and a sensible reduction of the associated carbon emissions, e.g. as a result of an improved load management, reduced idle CPU times or optimally exploited resources. Outcomes and implications connect to improving voice quality by conducting a predictive analysis

process of the incoming requests in order to anticipate computational requirements and correspondingly scale resources. A direct impact on the infrastructure management costs, performance and idle time management or, most important, on energy consumption is gaining upon [47]. As an emergent effect, it is equally foreseen to attain non-negligible carbon emissions footprint reductions as aftereffect of optimized scaling and utilization of available resources.

3.1.1 VoIP Traffic Analysis

This study was built on collaboration between the University of Luxembourg and MIXvoip [1], a company that hosts and delivers commercial VoIP services, with an important market share in the Luxembourg area and, at this time, a significant number of subscribed clients [48]. Organizational details, functional patterns and historical records were provided by MIXvoip and are used as a reference for predicting the load of a processor in response to the incoming traffic. Based on the traffic analysis, in pursuance of preparing in advance the execution environments, relevant prediction models were build.

3.1.1.1 Data extraction and sampling

In Figure 3.1 it is given a description of the path leading from data collection to data analysis, prediction and system reaction. The arrival date of phone calls, their origin, destination, and duration are examples of data generally used to build prediction models. This information is usually stored in a Call-Detail-Record (CDR) database, which was used to define the job model for load prediction. An example of . As a first step, the information with detailed records of the placed calls is collected and analyzed, e.g. the account code and profile id. Statistics are the input for the prediction model that based on a given time frame estimates the number of calls to be placed during the next time frame.

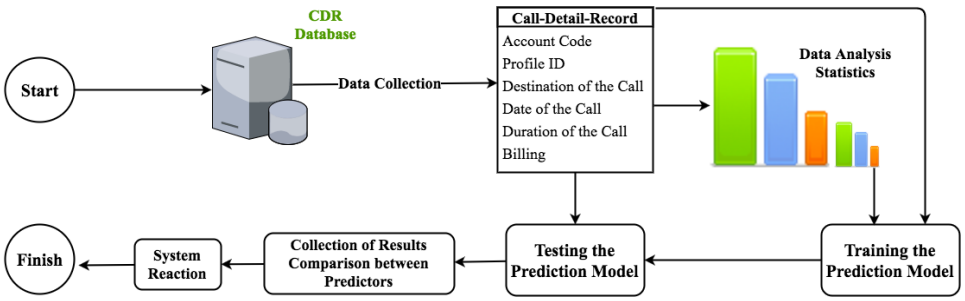


Figure 3.1: Data collection, data analysis, prediction, decision.

[1] https://www.mixvoip.com/

Field Name	Description	Example
id	Index of call in the CDR File	100100
account_code	User id of the caller	50
IP	IP of the phone where the call is placed	127.0.0.1
dst	Destination number of call	444444
prefix	Prefix of the country where the call is placed to	00352
dst_name	Destination country name of where the call is placed	Luxembourg
carrier	Telecommunications service provider; It can be established which prefix goes to which carrier, decision based on price, quality	BT - British Telephone
calldate	Arrival of the call in the system	2016-05-01 00:03:00
duration	Duration of the call in second	38
codec_id	Name of codec used by Asterisk	G.711

Table 3.1: CDRs Structure Example.
Job model for load prediction.

3.1.1.2 Capture of Trends and Call Distributions

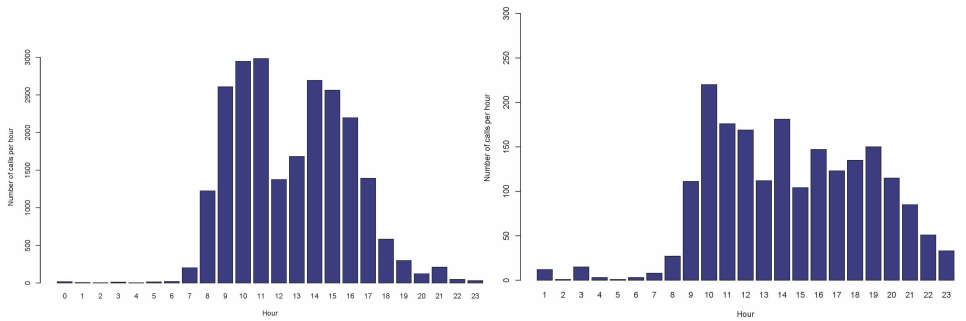
The clients of MIXvoip, during the Dynamic MixVoIP (DYMO) Project that was running at the moment ([2]), were SMB and their historical records were used to establish the profile of the users. As expected, most of the calls are done at peak hours, during weekdays except for the public holidays. The data collected from MIXvoip was analyzed and user profiles are outlined; it was observed there are patterns for calls placed during different hours of a day, peak hours or abnormal situations. During public holidays, the number of calls heads to very few per hour and, during a normal day, there are hours when servers are either overloaded (peak hours) or underloaded (with low traffic). The information was used to observe the evolution of servers load during working days and peak hours.

In Figure 3.2 it is shown how throughout a day, the highest traffic flow takes place during working hours, 8-11 AM and 13-18 PM(Figure 3.2 (a), (c), (d)). Considering the profile of the clients of MIXvoip (SMB), it can be seen that most of the calls take place during working days (Figure 3.3). During holidays the traffic drops significantly (Figure 3.2 (b), 3.3 (b)) and in this situations, computational resources should be turned off dynamically in favor of costs reduction. Ensuring availability of resources during peak, rush hours, without over-provisioning is also challenging.

3.1.2 Machine Learning and predictive algorithms

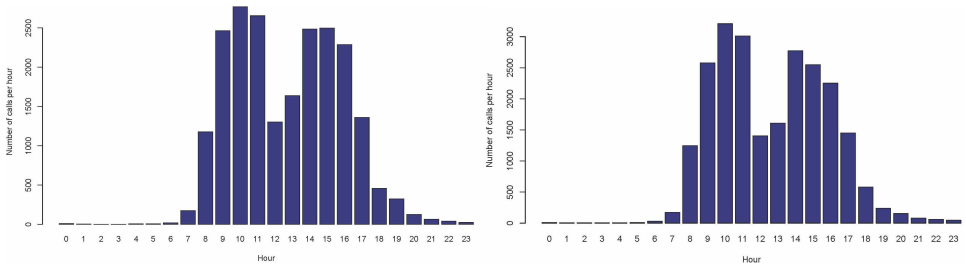
Machine Learning (ML) is a subfield of computer science that uses methods and algorithms to build analytical models, with exciting applications for real problems. ML has evolved from the computational learning theory and the field of artificial intelligence [49]. ML is used for a wide range of

[2]Funded by FNR and Luxembourg Ministry of Economy, `https://www.fnr.lu/projects/dynamic-mixvoip-3/`



(a) Monday-working day

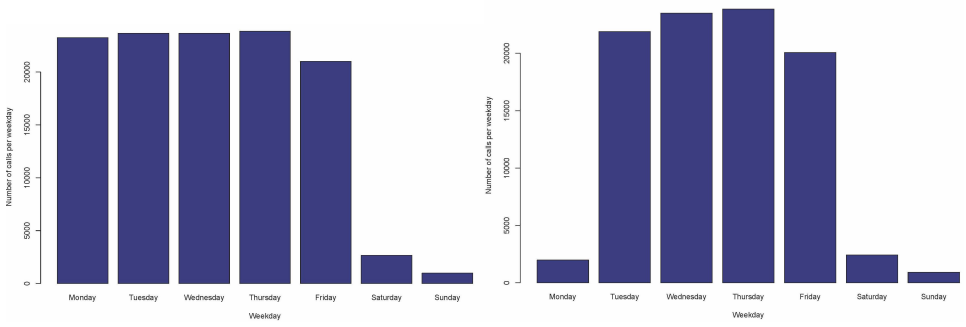
(b) Monday-holiday



(c) Tuesday-working day

(d) Wednesday-working day

Figure 3.2: Distribution of calls during working hours.



(a) Monday-working day

(b) Monday-holiday

Figure 3.3: Distribution of calls during weekdays.

applications and, to name a few: automated text categorization [50], facial expression recognition [51], network intrusion detection [52], automated traffic classification[53]. The learning process of an algorithm consists of its representation using a formal language, the evaluation function that scores the algorithms based on their performance, and the optimization method that defines the efficiency of the algorithm [54]. Data mining, predictive analysis drove a big wave of innovation and spread rapidly throughout computer science.

Preparing in advance the execution environment leads to optimal job execution while dynamically scaling resources in response to users' demand. Thus, as a first objective of this work, various ML techniques are developed to predict the traffic during a time frame, shaping future patterns and used for capacity planning. While extensive studies exist in each of the mentioned areas, only few sources consider such holistic approaches and analyses for VoIP environments. Furthermore, no conclusive agreement in the optimization domain exists on how to deal with highly dynamic time-dependent systems, causality and impact in a predictive framework, information coherence or descriptive power of the models when facing a fast changing environment where different scenarios are possible.

In the context of the thesis a simulator for rare events and particle likelihood estimation were implemented and used to define the environment sensitive models, resilient to errors or missing data, to adapt and to reflect context changes. Expertise from advanced optimization techniques were drawn upon, including dynamic, time-dependent factors, and interacting stochastic particle methods.

3.1.2.1 Well-known Prediction Methods

Gaussian Mixture Model (GMM) [55] is one of the most widely spread clustering and density estimation methods. For the general case, a mixture model is a parametric probability density function represented as a weighted sum of component densities, which optimally fit real unknown distribution of the data. A cluster, in this model, is mathematically represented by a Gaussian distribution [56]. When used as a prediction tool, GMMs give the probability that a new value is generated from one of the Gaussian components, that naturally group some data.

Gaussian Process (GP) [57] is a powerful tool in machine learning statistics and one of the most important approaches for Bayesian learning. It relies on effective methods for placing prior distributions over a space of functions, generalizing the multivariate Gaussian probability distribution. A GP is described by a mean function and a positive semi-definite covariance function. The use of Gaussian process models for prediction has become very attractive in a wide range of areas and problems, for example in the geostatistics field.

Neural Networks (NN) [58] are widely used for prediction problems, classification or control, in areas as diverse as finance, medicine, engineering, geology or physics. An artificial neural network is a computational model [59] inspired from the structure and function of biological neural networks. It is considered to be a strong nonlinear statistical data modeling tool where the complex relationships between inputs and outputs are modeled or patterns are found. Pattern classification, function approximation, object recognition, data decomposition are only a few examples of problems where artificial neural networks were applied.

Support Vector Machine (SVM) [60, 61] are frequently used in the machine learning community for classification problems and regression analysis. For example, SVMs are used for learning and recognizing patterns, for a given input. In [62] it is presented a framework for adaptive visual object tracking based on structured output prediction using SVM for the problem of tracking arbitrary objects.

Linear regression is commonly used for predictive analysis to describe the data and define the one-to-one or one-to-many relationship between dependent variable and independent variable/variables [63, 64]. Regression analysis is used for causal analysis, effect and trend forecasting. It is a technique applied by statisticians to estimate parameters from historical data of the linear model used to predict future behavior.

3.1.2.2 Rare Events

In [65] rare events are defined as events with a low but non-zero probability of occurrence, for example: $0 < P(X \in A) \leq 10^{-8}$. In the thesis it is used a special class of Sequential Monte Carlo approach, namely interacting particle methods. Interactive particle methods represent a general class of stochastic methods used to simulate target laws. In literature, there exist different techniques of using particle methods depending on their purpose exist [66, 67]. One important aspect of such methods is the distributed evaluation of the evolved particles, used for the computation of intermediate conditional probabilities. The conditional probabilities of the nested sequence of events, that lead to the rare event, are estimated accurately straightforward. The probability of the rare event is computed as the product of conditional probabilities.

The most traditional approaches used for rare event simulation are the *multi-level splitting* ones. In this case, there are *a priori* specified levels. The particles that pass the threshold associated to the level are used for re-sampling [68, 69]. One drawback of these approaches is the high variance of the results. In Figure 3.4 there is an example of fixed splitting method for particles. The three splitting level set consists of $\{L_0, L_1, L_2\}$, with possible independent paths for a particle x_0. Starting from level L_0, one path of the particle x_0 up-crosses level L_1. Three independent copies of the particle are

started from the entrance state at level L_1 and two of them down-cross 0 with one copy that up-crosses the level L_2 [70].

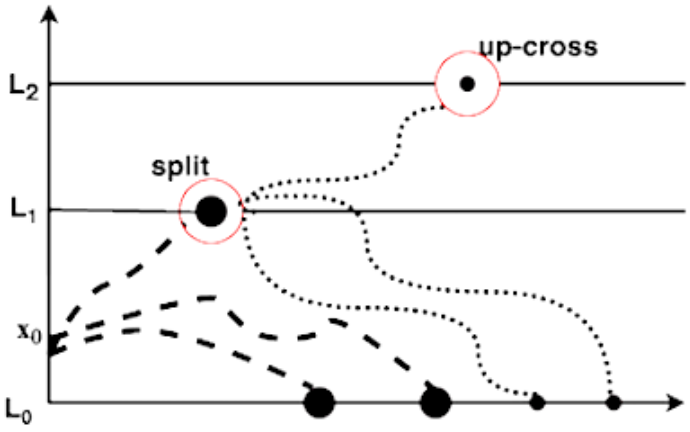


Figure 3.4: Fixed effort splitting.

The second method considered in the scope of the thesis is the *adaptive approach*. Particles are sorted in respect to their value, and a percentage ratio is established *apriori*. The value of the threshold, of the level, is determined by the value that delimits the given ratio of the highest ranked particles. As a next step, re-sampling of the particles that are below the determined threshold, takes place and at the end of each iteration all the particles are considered for the next iteration. The algorithm continues until the critical level is reached. The type of selection considered in Step 2 of Algorithm 4 is *accept-reject like selection*: if either the value of particle is not improved after the perturbation either it is improved with a value not higher than the delimiting threshold, it is replaced by one particle that satisfies both conditions.

3.1.2.3 Wishart Distribution

The Wishart distribution, noted $W(\Sigma, d, n)$, is used to model random covariance matrices and to describe probability density functions of random nonnegative-definite $d \times d$ matrices [71]. The parameter n refers to the degrees of freedom while Σ, a scale matrix, is a nonnegative-definite symmetric matrix of size $d \times d$.

A matrix $W \equiv \{X_i \sim \mathcal{N}_d(0, \Sigma)\ \}$, sampled from a Wishart distribution is the distribution of a sum of n rank-one matrices defined by independent normal $X_i \in R_d$ with $E(X) = 0$ and $Cov(X) = \Sigma$. When $X = \mu + A\mathcal{N}(0, I_d)$, Σ can be represented as $\Sigma = A \times A^T$, where the lower triangular matrix A is extracted using LU-decomposition [3] (Algorithm 1).

[3] `http://mathworld.wolfram.com/CholeskyDecomposition.html`

Algorithm 1 Wishart Distribution Sampling

Generate μ from $\mathcal{N}(0,1)$ e.g. Box Müller
Determine the (lower) triangular matrix A via a Cholesky decomposition of Γ as AA^T
Calculate $X \leftarrow \mu + A\mathcal{N}(0, I_d)$, d iid variable from $\mathcal{N}(0,1)$ e.g. Box Müller
Calculate $\Sigma \leftarrow \sum_{i=1}^{n} X_i X i^T$

The Box Müller transformation is used to generate pairs of independent, standard, normally distributed random numbers [4]. Box Müller transforms a two-dimensional continuous uniform distribution into a two-dimensional bivariate normal distribution. When u_1 and u_2 are uniformly, and independently distributed between 0 and 1, with mean $\mu = 0$ and variance $\sigma^2 = 1$, z_1 and z_2 are calculated as:

- $z_1 = \sqrt{-2 \ln u_1} \cos(2\pi u_2)$
- $z_2 = \sqrt{-2 \ln u_1} \sin(2\pi u_2)$

3.1.3 State of the art in VoIP

The focal points of the thesis are: an Interactive Particle System (IPS) [72] based algorithm, a Gaussian Mixture Model (GMM) and a Gaussian Process (GP). They are able to provide flexible modeling approaches (IPS), traffic shaping determined by clients (GMM), and scalable solutions with good prediction precision (GP). GPs approximate distributions that are obtained by Artificial Neural Networks, when for some specific cases the number of neurons tends to infinity. These methods are described as adaptive and dynamic simulation algorithms, that offer an increase in the numerical approximation performance and precision. IPS, GMM and GP are powerful methods that link-up, through visualization, the distribution of data with the models and enable advanced knowledge insertion. As seen in Section 3.1.1, the data is represented by underlying heterogeneous populations that can be explained by normal distributions. Moreover, the data collected from the CDRs is analyzed into a set of quantifiable real-valued properties, rather than categorical or ordinal.

In the past years a series of different studies targeted prediction modeling in VoIP. Most existing approaches are based on predicting the speech quality of VoIP. For example, the impact of packet loss and delay jitter on speech quality in VoIP has been studied by Lijing Ding in [73]. He proposes a formula used in Mean Opinion Score (MOS) prediction and network planning. A parametric network-planning model for quality prediction in VoIP networks, while conducting a research on the quality degradation characteristic

[4] `http://mathworld.wolfram.com/Box-MullerTransformation.html`

of VoIP, is presented by Alexander Raake in [74]. The work addresses the different technical characteristics of VoIP networks linked with the features perceived by the users. It is given a detailed description of VoIP quality and discussed how wide-band speech transmission capability improves telephone speech quality.

In [75], the authors present a solution for non-intrusively live-traffic monitoring and quality measuring. Their solution adapts to new network conditions and extends the E-Model, proposed by the International Telecommunication Unit (ITU-T), to a less time-consuming and expensive model. A model for objective, non-intrusive, prediction of voice quality for IP networks applied to voice quality monitoring and playout buffer control in VoIP networks, is presented in [76]. They develop perceptually accurate models for a non-intrusive prediction of the voice quality, models that avoid time consuming subjective tests, and conversational prediction voice non-intrusive models quality for different codecs.

The problem of traffic anomaly detection in IP networks has been studied in [77]. Estepa, R. presents an easy closed form for prediction of the mean bit-rate of one conversation generated by SID-capable speech codecs as a function of the codec and the number of frames per packet used. [78] explores the cumulative traffic over relatively long intervals, to detect anomalies in voice over IP traffic and to identify abnormal behavior, when different thresholds are exceeded.

In [79] the authors present a GMM based text-dependent system for speaker identification, with a minor impact on the packet loss rate. Similar discussion of automatic speaker recognition over VoIP can be found in [80]. The authors study codec parameters and compressed packet streams over VoIP considering Probabilistic Stochastic Histogram algorithm with Vector Quantization Probabilistic Stochastic Histogram (VQPSH) and Gaussian Mixture Model Probabilistic Stochastic Histogram (GMMPSH).

The call traffic on VoIP networks under heavy network conditions is modeled as a linear GP in [80]. The authors provide an accurate predictive representation of different traffic patterns. The performance of a VoIP system for speech recognition at the receiver level and a Gaussian algorithm for vector quantization, are presented in [81]. The authors use matching Mel-Frequency Cepstral Coefficients features to represent the raw speech signal. In [82], a model for call holding times that follows a Generalized Pareto Distribution along with a fractional Gaussian Noise Model for aggregated VoIP traffic, are described.

In the previous studies, GMM and GP focus on a variety of models used to shape VoIP calls characteristics. These models consider speaker recognition for VoIP transmissions, call duration and call holding times. Different predictors, namely Gaussian Mixture Model (GMM), Gaussian Process (GP) including an evolutionary algorithm, Interactive Particle System (IPS) are described and compared using real life data. In Chapter 4 it

is presented how prediction models fit to the field and their deviations from the real VoIP traffic behavior are analyzed.

3.2 Load Balancing

Load balancing is a job distribution decision-making process, used in production systems and computing. It is widely known as a technique applied for an efficient resource utilization, which can be implemented with hardware and software support. Jobs arrival rate, communication delay, the variability of the job parameters with other factors that influence systems performance. In order to deal with such complex factors, it is essential to design efficient and scalable load balancing algorithms. Few of the direct resulted benefits of applying such techniques are: implementing fail-over, ensuring scalability, avoiding bottlenecks, shifting from over-provisioning, reducing response time, reducing energy consumption etc [83].

In [84], the main aspects of server load balancing with basic and advanced aspects, networking fundamentals and design, are presented. The author introduces the four general main applications that load balancers have. A Server LB distributes the load between servers to design capacity scaling and fault tolerance. Global Server LBs are implemented for the redirection of users' requests to different data center sites, with a fast response. Firewall LBs deal with the distribution of the load across multiple firewalls so that one may scale the capacity and ensure fault tolerance. Transparent cache switching redirects the traffic to caches, for a fast response time and good performance of web servers (Figure 3.5). The author gives the necessary insides for the reader to understand and choose load balancing products in the most convenient shape, to meets their needs in terms of price, performance, reliability, scalability and so on.

Software products for load balancing follow algorithms that implement load distribution processes and that run on top of the load-balanced servers. The black-box products, that incorporate the hardware and software, are packaged with special operating system and software, and are referred to as *appliances*. *Switches products* add extensions to the traditional Layer 2/3 switch, by expanding functionalities.

In order to apply load balancing in VoIP, VSPs may categorize the types of traffic that flow through the network. The objectives of a load balancer are designed by the VSP, with regard to using different load-distribution methods. Load balancing can be performed *stateful* wherein the protocol is designed to recognize the session initiation and termination (e.g. for SIP traffic - session oriented protocol). In *stateless* load balancing the distribution methods are applied regardless of the individual sessions and the incoming traffic is considered (e.g. RTP).

Load distribution methods are used to allocate the load in the system,

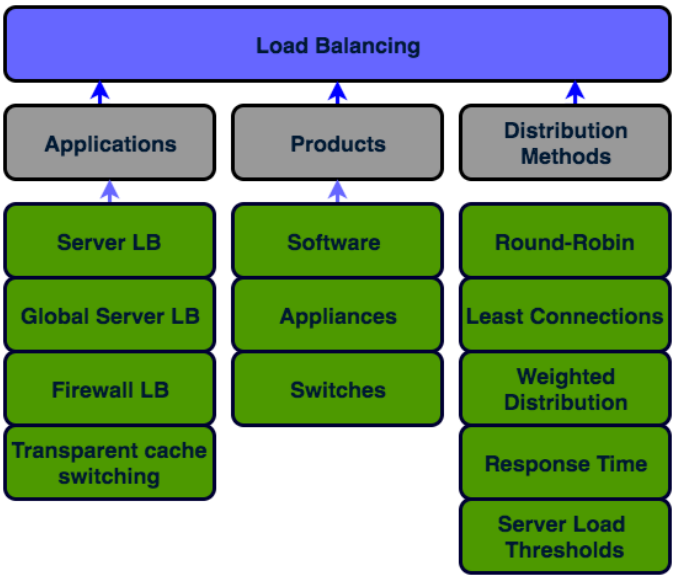


Figure 3.5: Load Balancing: Definition and Applications.

and few of the most common ones are: round robin, least connections, weighted distribution, response time, server probes and server load thresholds. *Round Robin* is a distribution method for which servers wait for their turn, that does not consider the number of active concurrent connections. It consumes a little amount of resources and is effective for systems that handle a high number of concurrent requests, roughly equivalent in terms of necessary processing resources and duration. The *least connections* method requires that the LB keeps track of the number of active concurrent connections on each node and that requests are forwarded to the least loaded node. When *weighted distribution* method is used, relative weights are assigned to the servers, followed by the use of different load-balancing methods. *Response time* is used in situations where the performance of applications is important. It can be used as a stand-alone method or it can be used to define threshold values for other methods. *Server load thresholds* are set for various indicators (CPU load, disk, available memory, network throughput), measured and used together with any load distribution applied method to distribute the load across servers.

Load balancing of services, computational jobs, virtual machines, virtual storages, database requests, and VoIP traffic on the network is a major concern for an efficient use of cloud computing. The development of an effective dynamic load balancing algorithm involves many important issues: load estimation, load levels comparison, performance indices, system stability, amount of information exchanged among nodes, job resource requirements estimation, job selection for transfer, remote nodes selection, etc [85]. The

impact of task scheduling and resource allocation in dynamic heterogeneous grid environments, given independent jobs has been studied in [86].

Few of the important aspects of the problem studied are: distribution of the nodes, storage replications, and virtual machine migrations. Some algorithms are efficient when nodes are located in proximity and communication delays are rather negligible, relevant information for cloud infrastructures (distribution of nodes). In situations where CSPs settle for a partial replication of the storage data, the information is distributed and saved in different nodes, leading to increased utilization, fault tolerance and data availability. A full replication of data increases storage and communication overheads. Migration of VMs is a common practice for redistributing workloads of heavily loaded nodes. The complexity of such operations consists of decisions related to the destination for relocation, what VMs are to be moved, profit and cost of the migration (VM migrations).

The deployment and the management of telephony tools via clouds is challenging, many factors having to be considered. In this thesis, infrastructure utilization is investigated. In order to optimize the overall system performance and the load of processors that handle the voice signal, IP (jobs) are balanced. When the processors are overloaded, the quality of the phone call is affected. One should consider the capacity of the network and the idle time of the resources, which increases expenses. Load-balancing improves VoIP performance by keeping processor idle time and interprocessor communication overhead as low as possible. To minimize the overall computation time, it is recommended to equally distribute the work to be computed. Load imbalance occurs during migration processes, the time arrival, variability on the utilization process, the interference from other users in time-sharing mode. Parameters such as: processor speed, number of available processors, and actual bandwidth, are changing over time and load balancing algorithms target an improvement of resources distribution and QoS.

3.2.1 Load balancing techniques

In this subsection a wide range of load balancing algorithms implemented for cloud computing and other computer environments [87], is overviewed. The main characteristics of the algorithms with their metrics are sketched in Table 3.2. A description of relevant algorithms is given in the following.

Autonomous Agent Based Load Balancing Algorithm (A2LB) [88] is a dynamic load balancing algorithm designed for cloud computing environments. The addressed issues are: maximum resource utilization, maximum throughput, minimum response time, dynamic resource scheduling with scalability and reliability. The load balancer mechanism is composed of the use of three agents: Load agent (LA), Channel agent (CA) and Migration agent (MA). LA is a static agent that controls the information policy and that

maintains detailed information of data-centers. The main objective of the algorithm is to calculate the load on every available VM. CA is a static agent that controls the transfer policy, the selection policy and the location policy. MA is an ant (special category of mobile agents) that moves to other data-centers and that communicates to inquire the status of the VMs.

Active Clustering (AC) [89] is an algorithm designed for large scale CC systems. Similar services are connected and instances are grouped via a local rewiring of the network. It is using self-aggregation and the performance is increased when the workload is delegated between the nodes aware of their similarity.

Biased Random Sampling (BSR) [89] is a self-organization algorithm that samples randomly the system domain, such that the load of the nodes are maintained close to the global mean measure. The representation of the server's load is represented by edges and the network of which the initial availability is measured. Incoming jobs are handled in the nodes that have not reached the utilization threshold, otherwise they are distributed to the neighbors randomly.

Compare and Balance (CB) [90] is used to reduce the migration time of VMs, by calculating the probability of a VM to migrate. It calculates the cost of the nodes part of the system, information used for comparison. When the cost of the current node exceeds the cost of a randomly chosen node, VMs are migrated with a certain probability in order to achieve equilibrium of the system.

Fuzzy-based Firefly Algorithm for Dynamic Load Balancing in Cloud Computing (FFA-DLB) [91] is a dynamic load balancing implemented for cloud computing environments. The proposed solution is a combination of the Firefly algorithm with fuzzy logic, an algorithm that separates the cloud based on the frequent node allocation and that balances the load across the variety of partitions. It has as goal the separation of hot-spots and the least loaded nodes, followed by a classification of nodes into groups (like lightly loaded, normal, and heavily loaded). The set of tasks arrive at the load balancer after the partition of the cloud. This algorithm considers a balancing factor based on the parameters of the VM and files are processed from the input. The fuzzy inference engine determines the assignation of tasks, with the condition that already assigned tasks are migrated when a high necessity occurs.

Genetic Algorithm (GA) based on load balancing [92] is a load balancing strategy used in cloud computing, that minimizes the make span of a given task set. It uses a binary representation for the chromosomes, a random single point crossover, and a mutation with probability of 0.05. It considers an estimated penalty (delay cost), the amount of money that cloud service provider needs to pay to the customer when the job finishing time exceeds the deadline advertised by the service provider.

Honeybee Foraging (HF) [89] is an algorithm that regulates the sys-

tem demand. Web services are allocated dynamically to the server and local server actions are taken as part of the global load balancing. Self-organization is a characteristic of this algorithm and the servers are grouped into virtual servers with their queue of jobs. The global colony profit is communicated using " advert board" and idle servers use the information to chose, advert and serve requests or to randomly serve queue requests.

Honeybee Foraging Behavior (HBB -LB) [93] is a dynamic load balancing algorithm developed for scheduling tasks, in cloud computing environments. The authors develop an algorithm in charge of balancing tasks on machines, in order to minimize the waiting time in the queue. The algorithm balances the priority of tasks (honey bees), that are removed from the overloaded nodes. The information regarding the number of priority tasks and the load of the VMs is updated, used by tasks to choose VMs based on their load. Whenever a high priority task has to be submitted to different VMs, it considers the VM with the minimum number of high priority tasks, as the particular task is executed at the earliest time.

Job - Idle -Queue (JIQ) [94] is a large-scale, dynamically scalable algorithm designed for cloud data centers. The discovery of lightly loaded servers is based on the job assignment. It consists of two LB systems that assign jobs on idle servers. The dispatchers assign jobs to the first element of the I-queue, structure that maintains the information regarding idle processors. When the I-queue is empty, jobs are allocated randomly by the dispatcher to the processor.

Load Balance Scheduling Based on Firefly (LBS-BF) [95] is a mechanism designed for cloud computing. The load balancer computes a load index of the shared resources, that are instantiated to effectively use the resources dynamically. The fireflies attraction is linked to the objective function and the monotonic decay of the attractiveness with distance. It generates the scheduling index and the distance calculation serves to find the closely associated nodes in the cloud network. The technique proposed uses three parameters: the attraction between nodes and requests, the scheduling index, and the distance between nodes. The parameters consider CPU rate, memory rate, processing time and the loads of the nodes.

Power Aware Load Balancing (PALB) [96] is an algorithm that maintains the state of all the nodes and allocation decisions are taken based on the number of computing nodes with their utilization percentage. It is a power aware algorithm split in three mechanisms: assignation of VMs to nodes, powering additional computing nodes and powering down the idle nodes.

Double Threshold Power Aware Load Balancing (DT - PALB) [97] is an updated version of the Power Aware Load Balancing algorithm. It applies VMs migration for minimizing the energy consumption in the system. When the utilization of a node is under 25% (lower threshold), the load balancer migrates workloads (VMs) to reduce its utilization to zero and power it off.

Self-Organized Agent Systems (SOAS) [98] is an algorithm designed for

distributed systems, that dynamically load balances the system. It extends the sand pile model where avalanches are used to allocate incoming tasks to computing resources. The system is reconfigured and the model fits the use of non-clairvoyant scheduling with Bags-of-Tasks.

Task Scheduling based on LB (TSLB) [99] is an algorithm that considers the request and the requirements of the users. Firstly, users are allocated to VMs. Secondly, VMs are assigned to host resources. During the execution, the a priory known demands can be changed. In this situation, VMs can be moved to nodes with available resources. VMs that reside on the same node can be moved, in order to free up resources.

Task-Based System Load Balancing using Particle Swarm Optimization (TBSLB-PSO) [100] is an algorithm that uses Central Task Scheduler (CTS), that transfers extra tasks from overloaded VMs to VMs with similar properties. The information regarding the cloud schedulers for VM features is centralized on a blackboard, and is used for the migration decision and to ensure QoS. The migration process considers: the amount of data, memory, bandwidth and numbers of CPU of the VMs. Idle Physical Machines (PM) are not selected as new PM hosts, in the interest of decreasing the energy consumption.

VM to physical Machine Mapping (VMMP) [101] is an algorithm that calculates the probability that a node is selected, probability direct proportional to the weight of the node. The utilization of the resources is summed and normalized, information that is used to assign a weight on each node. The algorithm calculates the amount of available resources and the higher the weight of the node, the higher probability of a node to be chosen as a destination for the VMs.

3.2.2 State of the art in VoIP

Session Initiation Protocol (SIP) Load Balancer (LB) is a load balancer designed for the SIP traffic that takes place between a pool of available SIP servers, that scales and optimizes SIP infrastructures. The main scope is to increase the overall throughput of the service or application and their reliability, by handling failover requests mid-call from unavailable nodes to available ones. The dynamic provision of SIP service requests and responses, between the available nodes, satisfy in real-time the demands of SIP services. The LB parses, alters and forwards the incoming SIP messages (e.g. INVITE), sent by User Agent (UA) to the same SIP Uniform Resource Identifier (URI), to an available node. The communication takes place between the SIP Servers and the SIP Load Balancer (LB), independently of the UA [102].

Mobicents SIP Load Balancer (LB) is a SIP-based proxy, proposed for VoIP infrastructures with multiple ingress proxy servers. It balances SIP services load to achieve scalability and to ensure availability. The proposed

Table 3.2: Load Balancing Algorithms

Algorithms	HBB-LB	DT-PALB	TBSLB-PSO	LBS-BF	A2LB	GA	FFA-DLB	JIQ	TSLB	HF	BRS	AC	CB	VMMP	PALB	SOAS	RoC-LB
Characteristics																	
Centralized	Y	Y	Y			Y	Y		Y					Y	Y		
Decentralized								Y		Y		Y	Y			Y	Y
Hierarchical					Y												
On line		Y			Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y
Off line	Y		Y										Y				
Non-clairvoyant								Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
QoS						Y											
Migration cost			Y										Y				Y
Static																	
Dynamic	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Replication																	
Migration	Task	VM	Task				Task		VM	Job	Job	Job	VM			Task	Task
Metrics																	
Utilization		Y	Y	Y		Y	Y		Y				Y	Y	Y	Y	Y
Performance	Y							Y	Y	Y	Y	Y					Y
Overhead								Y					Y				Y
Response Time	Y			Y	Y	Y		Y	Y								Y
Scalability										Y	Y	Y					Y
Throughput	Y									Y	Y	Y				Y	
Energy consumption		Y	Y												Y		
Cost						Y	Y										

solution pools SIP servers, and are used to balance the traffic between server nodes in order to satisfy service demand in real-time. The availability of the server nodes is checked through health monitoring and the distribution of the traffic between the servers is done using the Round Robin (RR) algorithm. Compared to the traditional SIP-based LBs, in the solution proposed by Mobicents, new nodes can be added automatically without the need to reconfigure or update the LB manually. It considers requests and system parameters, for example CPU consumption [103].

In [104] it is proposed an advanced version of SIP Server Load Balancer (LB). Brocade offers a solution that behaves as a load balancer, for proxy or registrar traffic, rather than SIP proxy server. It passes through and translates SIP traffic from SIP servers to SIP clients. Part of the implementation, the SIP server real IP are replaced with Virtual IP (VIP). The application provides persistence for UDP SIP traffic and in stateful mode, sessions are created so that subsequent transactions with the same persistence parameter are send to the same real server, enabling the load balancing of the Back-to-back user agent (B2BUA) SIP servers.

Loadbalancer.org is an international organization which offers Load Balancer (LB) solutions, to optimize specific application environments, developed for leading application vendors (e.g. Microsoft, VMware) [5]. Load Balancing Skype for Business distributes requests between a pool of servers that run Skype for Business, a commercial VoIP solution purchased by companies due the provision of unified voice communications, audio, video and web conferencing [105]. Independently of their location, users connect using the platform that allows seamless communications for and easier management, a wider choice and flexibility. The Load Balancer (LB) configures a series of VIPs, the connection points for internal and external clients.

Radware [6] is a company that addresses different needs of IT organizations, namely protection against Distributed Denial of Service (DDoS), delivery of applications and load balancing solutions. The VoIP load balancing solutions offered by Radware guarantee: network availability, security, scalability, and VoIP traffic management. It optimizes business operations, minimizes service delivery degradation and prevents downtime. The solution consists of local and global clustering together with health checks on SIP devices. The use of unstable and unreliable devices is minimized by routing away the SIP clients, to achieve optimal call completion [106].

OpenSIPS is an Open Source SIP proxy/server with multiple functionalities (SIP router/ switch, SIP Redirect, SIP Registrar, etc.) and modules (LB, B2BUA etc.), that handle a high number of Call per Second (CPS) with high routing flexibility and integration [7]. The LB module defines des-

[5] `http://loadbalancer.org/ca`
[6] `https://www.radware.com/`
[7] `https://www.opensips.org/`

tinations depending on the set of resources and services. It takes in consideration the capacity and the number of concurrent calls (maximum load) that can be handled. In OpenSIPS, calls are routed between destinations, for which it considers the number of ongoing calls, computed as load status. Together with the information regarding the preconfigured destinations and their maximum load accepted, LBs route calls to destinations with the largest available slot [107].

In [108] the authors present a SIP dedicated load balancing scheme designed for servers and clusters of servers, to handle heavy VoIP traffic while regarding QoS. The selection of a SIP server to handle the requests is based on Domain Name System (DNS) Service Record (SRV) records, workloads metrics, that are collected from proxy servers, the overall workload in each server and their capacity of processing transactions. SIP clients communicate with the LB to find the best proxy available. In the situation the LB is not responsive, clients communicate directly with the DNS to retrieve the list of available SIP servers (via SRV records) and choose one. The authors suggest in the article that in case of hardware failure situation, when the LB is not informed about the availability status of the servers, to use daemons that collect workload metrics and inform dynamically the LB.

3.3 Load prediction and balancing framework

The overall long-term perspectives fixed in the context of the thesis consider several axes of research that include incoming load analysis and prediction, load balancing and energy-efficient optimization, management of cloud-based environments. The study leads, in a first phase, to a clear specification of the design axes required for a dynamic, autonomous VoIP cloud-based environment. In a subsequent phase, having as basis the environment already in use at MIXvoip, a comprehensive exploration of the mentioned research axes is conducted [2]. The outcomes and implications connect to improving voice quality by conducting a predictive analysis process of the incoming requests, used to anticipate computational requirements and correspondingly scale resources. A direct impact on infrastructure management costs, performance and idle time management or, most important, on energy consumption is implied. As an emergent effect, non-negligible carbon emissions footprint reduction is attained, as a consequence of the optimized scaling and utilization of available resources. A brief outline of what cloud environments stand for is provided followed by a more detailed discussion of the explored research directions.

The conducted research offers comprehensive insights of how optimization paradigms, in conjunction with predictive analysis, resource allocation and load balancing algorithms, can be adapted to include the constraints of a real-word large scale dynamic environment. In addition to exploring

an unified algorithmic perspective, the complementary analysis of concepts emerging from dynamic optimization, stochastic modeling, learning or resource allocation, offers a better understanding of how models for time dependent factors should be defined and analyzed, e.g. evolving constraints, time varying optimization objectives or changing environments. Different scenarios, strategies and assessment of implications that stem from on-line decisions is studied. Error propagation, reaction time and resilience issues in the presence of stochastic events that disrupt the normal functional patterns of the system, is addressed.

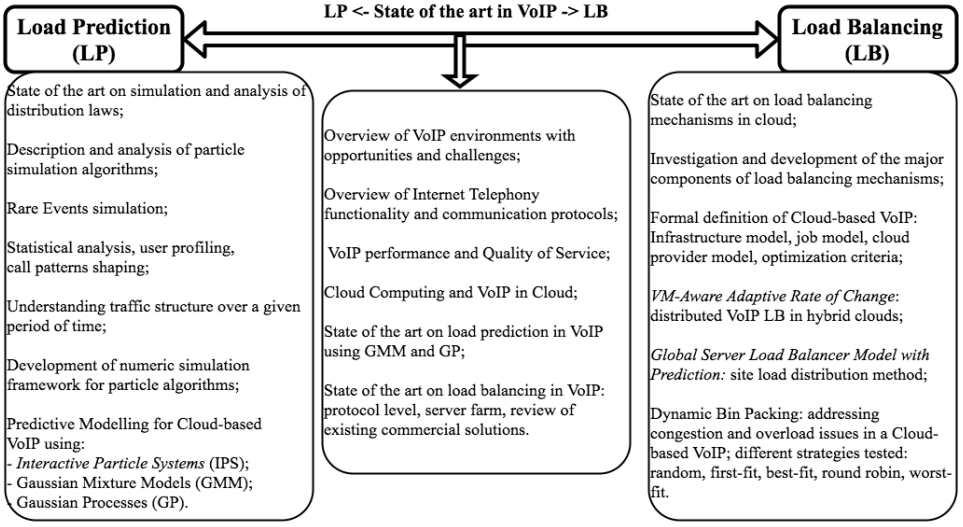


Figure 3.6: Load Prediction and Load Balancing Framework.

A focus for prediction-based performance optimization is given, addressing at the same time and with same level of importance, load balancing and resource allocation issues for cloud-based VoIP solutions. Important aspect concerns resource allocation and load balancing mechanisms, handled using integer programming techniques and exact heuristics. Aim of the study is the development of intelligent load balancing mechanisms that optimally spread the traffic on distributed processors and servers in the cloud. Several requirements are addressed, for instance increased scalability, high performance and high availability, disaster recovery.

Major components of load balancing mechanisms are investigated and developed. The load information is quantifiable by a load index set to zero when the processor is not loaded and increases, as the processor load gets heavier. As load measurements occur frequently, computing them efficiently leads to an exhaustive use of many parameters. An overhead of short update periods may negate the advantages of up-to-date indices, and long update periods may render load indices obsolete (Load index measure). The workload information is collected and maintained to apply load-balancing

decisions (Information exchange). The balancing act considers the cost of collecting global load information and maintaining the accurate state of the system. Load balancing operations are defined by three rules: the location rule, distribution rule, and selection rule.

3.4 Summary

In this chapter the main domains investigated in the scope of the thesis are presented. Congestion and overload issues in cloud-based VoIP environments are addressed using load prediction and load distribution techniques. The relevance of using load prediction is presented together with the analysis of the traffic from a real VoIP system. The data was collected from the CDRs database, provided during the collaboration between the University of Luxembourg and MIXvoip, a VSP in Luxembourg and close proximity countries. Trends and call distributions are captured and presented, to outline patterns of the incoming traffic which are used as benchmarks to check the quality given by the prediction techniques proposed in Chapter 4. General information regarding the most widely spread prediction methods used in ML is given. The concept of *rare events* together with their formal definition, use, are presented. A review of the state of the art in predictive analysis using IPS, GMM, GP for VoIP systems is provided. Load Balancing is introduced with its main, advanced concepts and aspects. Load distribution methods are described together with a wide range of algorithms. A state of the art in VoIP and commercial solutions is also presented. The close connection between the two main axes investigated in the thesis, namely load prediction and load balancing, is outlined. Scaling resources based on the information given by predicting the incoming traffic has a direct impact on the infrastructure management of VoIP cloud-based environments.

Chapter 4

Prediction Model

In this chapter it is presented a series of prediction models built for a real Voice Over Internet Protocol (VoIP) environment, namely methods that predict the traffic load that will take place successively. The prediction of the incoming voice traffic is used for outlining traffic patterns, for capacity planning, for improving the quality of VoIP services. During public holidays, Sundays and Saturdays, the number of calls is very low while during weekdays peaks occur. Users' behavior that adopt VoIP as a solution and that place VoIP calls, is outlined. This information is considered for decision making in allocating resources for VoIP applications to be run in the system. Due to the dynamic evolution of requests, decisions concerning the distribution of computational resources in a VoIP environment must be also taken dynamically. A dynamic system can adapt, scale and as the VoIP must be available 24 hours a day throughout the whole year, the availability requirement must address this business requirement.

The content of this chapter is based on three reviewed and accepted publications of which two were presented during the conferences: "EVOLVE 2013, A Bridge between Probability, Set Oriented Optimization, and Evolutionary Computation" [2], "2015 IEEE Globecom Conference" [4] and one published in the "Journal of Telecommunications and Information Technology, 2013" [3].

4.1 Design and Implementation of the Predictive Models

In this section there are presented three algorithms adapted for VoIP, based on interactive particle algorithms (Interactive Particle System (IPS)) [3], model-based learning of mixture of Gaussians (Gaussian Mixture Model (GMM)) and supervised learning that defines distributions over functions (Gaussian Process (GP)) [4]. The considered methods are used to predict the incoming traffic in servers, the number of calls, based on previous ob-

servations.

The customers' requests are modeled using a multi-Gaussian probability distribution. The models considered for prediction learn from the data which follows the Gaussian distribution, and predict the load of a processor in response to the incoming voice traffic. Algorithm 2 is used for visualizing the distribution of the placed calls, and an example of sampling result is given in Figure 4.1.

Algorithm 2 Extraction of samples pseudo-code

Store values from database in file, read file, set values X and Y for each day
$size_1 = \frac{max(X)}{nrSeg}$, $size_2 = \frac{max(Y)}{nrSeg}$
Count the days belonging to each interval where
$(i-1) \times size_1 \leq X \leq i \times size_1$ and
$(j-1) \times size_2 \leq Y \leq j \times size_2$, $i, j \leq nrSeg$
return samples

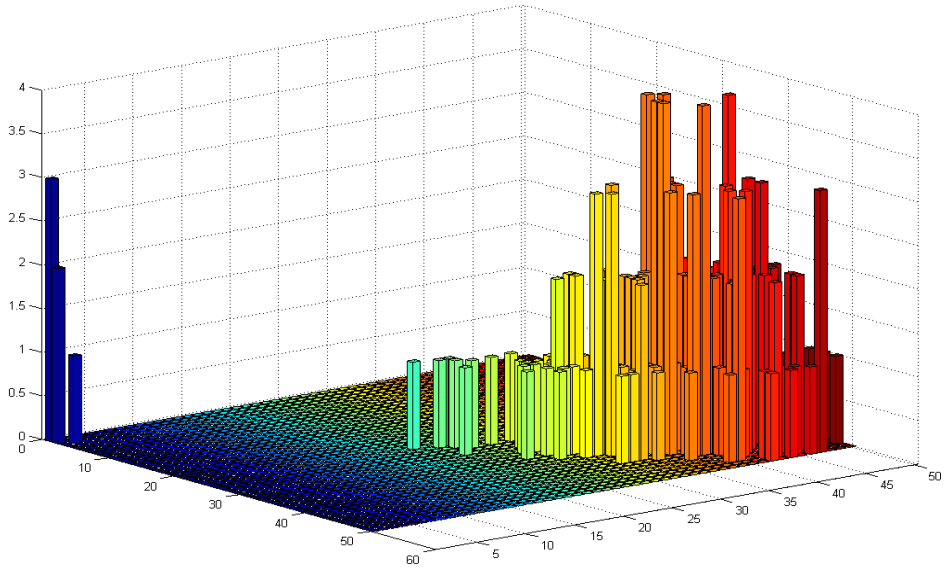


Figure 4.1: Sample of calls placed in 2012 from 10:00-10:59 AM and 11:00-11:59 AM.

Each day is defined by two parameters: the number of calls placed during sequential time frames, $D(X, Y)$. A number of segments is chosen following Algorithm 2 and the maximum number of calls for X, Y is calculated. The size of each interval is computed by dividing the maximum number of calls for X, Y to the number of segments $(size_1, size_2)$. The information is used to classify the days as belonging to an interval for X, respectively Y. The

distribution of days that belong to each pair (X, Y) is plotted in Figure 4.1, where the presence of Gaussian subpopulations within the overall population is displayed.

4.1.1 Predictive Modeling using Interactive Particle Systems

The model presented in this subsection is based on interacting particle algorithms, commonly used for parameters' estimation given a Gaussian model [109]. Pierre Del Moral and Arnaud Doucet define interactive particle methods as an extension of Monte Carlo methods, that allows sampling from complex high dimensional probability distributions. The algorithm estimates normalizing constants and approximates the target probability distributions, by a large cloud of random samples termed particles. Each particle evolves randomly in the space and, based on its potential, will survive or not. Many applications took benefit of this intuitive genetic mutation-selection type mechanism, in the areas of nonlinear filtering, Bayesian statistics, rare event simulations or genetic algorithms [110]. The sampling algorithm applies mutation transitions and includes an acceptance - rejection selection type transition phase. This approach is closely related to evolutionary-life algorithms, while providing theoretical error bounds and performance analysis results [72].

IPS starts with N particles, denoted by ξ_0^i, $1 \leq i \leq N$, that evolve according to the transition $\xi_0^i \rightarrow \xi_1^i$, given a fix set A. When $\xi_1^i \in A$, it is added to the new population of N individuals $((\hat{\xi}_1^i)_{1 \leq i \leq N})$, or is replaced by an individual randomly from A when otherwise. The sequence of genetic type populations is defined by $\xi_n := (\xi_n^i)_{1 \leq i \leq N} \xrightarrow{selection} \hat{\xi}_n := (\hat{\xi}_n^i)_{1 \leq i \leq N} \xrightarrow{mutation} \xi_{n+1}$. $\hat{\xi}_{n-1}^i \rightarrow \xi_n^i$, seen as a parent. An overview of convergence results, including variance and mean error estimates, fluctuations and concentration properties, is given in [72]. For parameters' estimation, a population of particles is generated as presented in the following. Each particle encodes a mean vector μ of size d, a matrix and $W \equiv \{X_i \sim \mathcal{N}_d(0, \Sigma)\ \}$ sampled from a Wishart distribution [21]. Wishart distribution, $W(\Sigma, d, n)$, is used to model random covariance matrices and to describe the probability density function of random nonnegative-definite $d \times d$ matrices. The parameter n refers to the degrees of freedom and Γ in Algorithm 3 denotes a scale matrix. The initial population of particles is generated firstly and a perturbation step is repeated for a given number of times, with a specified constant value, a. The particles evolve during the transition step.

Algorithm 3 IPS - Initial population pseudo-code

Step 1 - Generation of Particles
Fix some population size, N
Draw $X \sim \mathcal{N}_d(\mu, \Sigma)$,
for $i = 1 \rightarrow N$ **do**
Generate μ from $\mathcal{N}(0,1)$ e.g. Box Müller
Determine the (lower) triangular matrix A via a Cholesky decomposition of Γ as AA^T
Calculate $X \leftarrow \mu + A\mathcal{N}(0, I_d)$, d iid variable from $\mathcal{N}(0,1)$ e.g. Box Müller
Calculate $\Sigma \leftarrow \sum_{i=1}^{n} X_i X i^T$
Likelihood $L = (2\pi)^{-N \times d/2} \times |\Sigma|^{-N/2} \times exp^{\sum_{i=1}^{N} \frac{-(x-\mu)^T \times \Sigma^{-1} \times (x-\mu)}{2}}$
end for
return initial population

Algorithm 4 IPS - Perturbation of Particles pseudo-code

Step 1 - Perturbation, mutation of the encoded parameters
for $k = 1 \rightarrow steps$ **do**
Perturb the encoded vector $W \equiv \{X_i \sim \mathcal{N}_d(0, \Sigma)\}$
Draw samples $\{Y_i \sim \mathcal{N}_d(0, \Sigma)\}$, $1 \leq i \leq n$
Construct a new vector $\{X_i \leftarrow \sqrt{a} \times X_i + \sqrt{1-a} \times Y_i\}$, $1 \leq i \leq n$
Perturb μ, $\mu \leftarrow \mu + val$, val generated from $\mathcal{N}(0,1)$
Calculate new Sigma, Σ_i^{new}
Calculate L_i^{new}, likelihood with new Σ_i^{new} and μ_i^{new}
if $L_i^{new} > L_i$ **then**
$P_i(\mu_i, \Sigma_i) \leftarrow P_i(\mu_i^{new}, \Sigma_i^{new})$
end if
end for

Step 2 - Selection of the particles for the next generation
Order particles based on likelihood;
Select $L^{threshold}$ of the particle at the position given by percentage of particles to survive.
for $i = 1 \rightarrow N$ **do**
if $L_i \geq L^{threshold}$ **then**
add P_i to the list of surviving particles after perturbation
else
$P_i(\mu_i, \Sigma_i) \leftarrow P_j(\mu_j, \Sigma_j)$, P_j chosen randomly from the list of the particles that have survived the perturbation.
end if
end for
return perturbed population

The perturbation step consists of the modification of each particles' parameters, subject to distribution invariance constraints. New values for val are generated from $\mathcal{N}_d(0, \Sigma)$ and $\mu \leftarrow \mu + val$ is calculated. We draw new samples $\{Y_i \sim \mathcal{N}_d(0, \Sigma)\}$, $1 \leq i \leq n$. The encoded vector is recalculated using the vector Y and the value a, step described in the algorithm. After recomputing the Σ parameter, the likelihood of the perturbed particle is determined.

The perturbed particles are considered during the selection step, if their likelihood is improved (Algorithm 4 - Step1). The perturbed particles are sorted using their likelihood values and those with a better likelihood than $L^{threshold}$ survive. Otherwise, they are replaced with a particle chosen randomly from the set of the surviving particles during that iteration. In this case, the old one is discarded during the acceptance/rejection selection phase. The step is repeated and values of the parameters that improve the likelihood are recorded (Algorithm 4- Step2). After the last perturbation applied, the final population is obtained. Two prediction methods are

Algorithm 5 IPS - Prediction Methods pseudo-code

Extract final population
if $IPS - ML$ **then**
Extract from final population P_d, where $L_d = max(L)$
Extract μ_d, Σ_d that describe best the data

$$\mu_d = \begin{bmatrix} \mu_d^{\alpha} \\ \mu_d^{\beta} \end{bmatrix}$$

$$\Sigma_d = \begin{bmatrix} \Sigma_d^{\alpha_1} & \Sigma_d^{\alpha_2} \\ \Sigma_d^{\beta_1} & \Sigma_d^{\beta_2} \end{bmatrix}$$

Calculate prediction : $Z^{\beta}|Z^{\alpha} = z \sim \mathcal{N}(\mu^c, \Sigma^c)$
$\mu^c = \mu_d^{\beta} + \Sigma_d^{\beta_1} \times (\Sigma_d^{\alpha_1})^{-1} \times (z - \mu_d^{\alpha})$
$\Sigma^c = \Sigma_d^{\beta_2} - \Sigma_d^{\beta_1} \times (\Sigma_d^{\alpha_1})^{-1} \times \Sigma_d^{\alpha_2}$
end if

if $IPS - AL$ **then**
Prediction based on the likelihood of each particle, weighted likelihood and the training test:
Let $Ltotal = \sum_{i=1}^{N} L$
for $i = 1 \rightarrow size(samples)$ **do**

$$Z^{\beta} = \frac{\sum_{i=1}^{N} Z^{\alpha} \times L_i}{Ltotal}$$

end for
end if
return prediction

presented in Algorithm 5. First, the particle with the maximum likelihood is chosen, its parameters are obtained and using the first component (Z^{α}, input for testing), the second component (Z^{β}) is extracted. For the second IPS prediction method, the likelihood of all the particles is considered. The component Z^{β} is calculated via a weighted sum of the likelihoods, multiplied with the first component and divided by the sum of likelihoods. The result is the estimation of the traffic occurring during the second time frame, for every working day of the chosen period.

The intuitive genetic mutation-selection type mechanism is being used in a diverse range of domains, e.g. rare event simulations, genetic algorithms. The proposed interactive particle algorithm can be seen as a derived evolutionary algorithm, where mutations and selections are applied without considering crossover methods.

4.1.2 Predictive Modeling using Gaussian Mixture Models

In this section, one of the most widely used clustering method, a *model-based* learning of *mixture of Gaussians*, is presented. The clusters are mathematically represented by continuous parametric distributions, Gaussian distributions, referred to as *components*. The Gaussian Mixture Model (GMM) is a probabilistic density model, that considers that the input data is generated from a mixture of a finite number of Gaussian distributions[111]. A GMM is described by a weighted sum of M component Gaussian densities, parameterized by mean vectors, covariance matrices and mixture weights of all the component densities.

In Figure 4.2, two solid curves corresponding to individual Gaussian density components are shown together with the estimated probability that an observation is drawn from that component distribution [1].

Formally, the GMM is a distribution with the general Probability Density Function (pdf):

$$p(x) = \sum_{i=1}^{M} w_i N(x; \mu_i, \Sigma_i),$$

$$N(x; \mu_i, \Sigma_i) = \frac{1}{2\pi^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} exp\left\{\frac{-1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\right\},$$

where $N(x; \mu_i, \Sigma_i)$ is the pdf[112] of the normal distribution, μ_i is the mean vector, $\mathbf{\Sigma}_i$ is the covariance matrix, x is a D- dimensional continuous-valued data vector, and the mixture weights w_i have the property that $\sum_{i=1}^{M} w_i = 1$, $\forall i, w_i \geq 0$.

The parameters of the model are represented by the equation:

$$\lambda = \{w_i, \mu_i, \Sigma_i\} \text{ with } i = 1.....M$$

[1] `http://courses.ee.sun.ac.za/Pattern_Recognition_813/lectures/lecture06/node1.html`

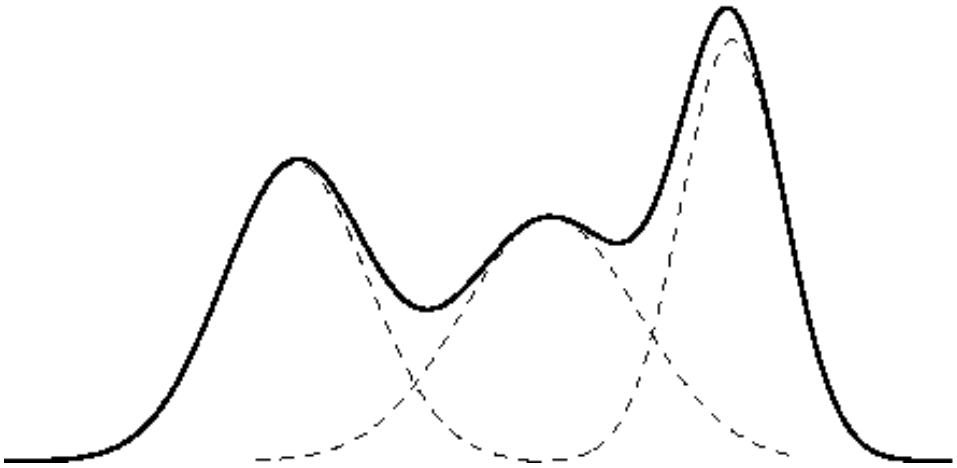


Figure 4.2: Gaussian Mixture Model with 3 components.
Source: [2]

After the training step, the parameters of the GMM are estimated using the Maximum Likelihood Estimation (MLE) method. The parameters expected to maximize the likelihood of the GMM are:

$$p(X|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda),$$

where T stands for the sequence of training vectors $X = \{x_1, ..., x_T\}$, with λ parameters of the best GMM.

At each iteration, the parameters are updated using an Expectation-Maximizatiom (EM) approach (Algorithm 6). EM is a maximization likelihood based algorithm, used for fitting mixture models to the set of the training data, and it requires the number of components to be incorporated into the model. The algorithm starts with components generated randomly, and for each data point from the input set, the probability of being generated by each component is calculated (*Expectation step*). The parameters are updated in order to maximize the likelihood of the data (*Maximization step*) [3]. The *Expectation and Maximization steps* are repeated until convergence.

The Akaike Information Criterion (AIC) score is used to measure the goodness of the models and to identify the best model out of a series of models, with an increasing number of analyzed time intervals. The parameters given to the algorithm are: the number of components and the maximum number of iterations.The AIC score is used to select efficiently the number of components in a GMM.

After the GMM is trained with the input data, the model is used for prediction by assigning each sample to the class of the Gaussian it belongs to, with the highest probability.

[3] `http://scikit-learn.org/stable/modules/mixture.html`

Algorithm 6 Update for parameters of GMM - EM [113]

Estimation of a new model $\bar{\lambda}$ such that $p(X|\bar{\lambda}) \geq p(X|\lambda)$
for $i = 1 \rightarrow iterations$ **do**

$\bar{w_i} = \frac{1}{T}\sum_{t=1}^{T} Pr(i|x_t, \lambda)$, mixture weights;

$\bar{\mu_i} = \frac{\sum_{t=1}^{T} Pr(i|x_t,\lambda)x_t}{\sum_{t=1}^{T} Pr(i|x_t,\lambda)}$, means;

$\bar{\sigma_i} = \frac{\sum_{t=1}^{T} Pr(i|x_t,\lambda)x_t^2}{\sum_{t=1}^{T} Pr(i|x_t,\lambda)} - \bar{\mu_i}^2$, variances (diagonal covariance).

end for

4.1.3 Predictive Modeling using Gaussian Processes

In this section it is presented one of the most attractive models for probabilistic classification and supervised learning applications, that defines distributions over functions, namely Gaussian Process (GP)[114] [115]. GP are applied to analyze complex data sets in statistics and machine learning. GPs are flexible parametric models, that use the information from the training data to learn the hyperparameters using the marginal likelihood [116]. GPs are used to fit complicated models and have applications in robotics, process engineering (real-valued regression), recognition (classification), user rating and disease screening (ordinal regression) [117].

The training dataset is used in supervised learning for learning input-output mappings. The input vector has many variables and the targeted output is, in the regression case or the classification case, continuous or discrete. Training the data is inductive, and it is used in the context of the thesis to make predictions for new input values. The data used together with samples drawn from prior distributions leads to the *posterior* distribution over functions.

Definition: *A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution* [4].

GPs are defined by a covariance function $k(x, x')$ and a mean function $m(x)$, a generalization of the Gaussian distribution defined by a mean vector and a covariance matrix. The covariance function gives the model of the data with its characteristics. Learning in a GP focuses on the problem of finding the suitable properties of the covariance function, controlling the properties of the specific GP.

A function $f \sim \mathcal{G}P(m, k)$ is distributed with the mean function m and a covariance function k. The resulting covariance matrix must be positive

[4] `http://www.inference.phy.cam.ac.uk/hmw26/papers/gp_intro.pdf`

definite and in order to ensure that, the covariance function must be positive definite as well. The value of the stochastic function f at the location of input x is the associated, random variable, $f(x)$ to a given argument x.

Given a process x_i, indexed by i, the vector of means and the covariance matrix can be evaluated to draw samples from the function f:

$f \sim \mathcal{G}P(m, k)$, $m(x)$ mean function, $k(x, x')$ covariance function;
$\mu_i = m(x_i)$, $\Sigma_{ij} = k(x_i, x_j)$, for n locations and $i, j = 1, ..., n$;

are used to generate a random vector from the distribution with coordinates: $\mathbf{f} \sim \mathcal{N}(\mu, \Sigma)$; the variable x has a Gaussian normal distribution with μ and Σ, mean vector and covariance matrix, respectively.

The covariance function of a Gaussian Process is a function that shapes the similarity between two samples (the covariance between pairs of random variables), $k(\mathbf{x}, \mathbf{x}') = E[f(\mathbf{x})f^*(\mathbf{x}')]$ (expectation), with parameters that have to be estimated. The kernel (covariance) function $k(x, x')$ is evaluated at x and x'.

GPs can be used to define distributions over functions, that does not depend on the training data but specifies properties of the functions. *Posterior* Gaussian Process (GP) is often used to make predictions for unseen test cases. The conditional distribution of the set of function values, that correspond to test input sets, given the known function values of the training cases is [116]:

- $\mathbf{f}_* \mid \mathbf{f} \sim \mathcal{N}(\mu_* + \Sigma_*^T \Sigma^{-1}(\mathbf{f} - \mu), \Sigma_{**} - \Sigma_*^T \Sigma^{-1} \Sigma_*$, where
- $\mathbf{f}$ known function values of the training case
- $\mathbf{f}_*$ set of function values corresponding to test set input X_*
- $\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}(\begin{bmatrix} \mu \\ \mu_* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_* \\ \Sigma_*^T & \Sigma_{**} \end{bmatrix})$, joint distribution where
- $\mu = m(x_i)$ training means, $i = 1, ..., n$, and $\mu_\mathbf{i}$ test means
- Σ training set; Σ_* training-testing set; Σ_{**} test set covariances

The set of covariance functions one can opt for is wide, many having adjustable parameters that can be inferred or learned from the data, using marginal likelihood or cross-validation methods[5]. The commonly used covariance functions are: constant, linear, polynomial, squared exponential, exponential, neural network, rational quadratic, Matèrn. In the scope of this study, a Matèrn kernel function was opted for, among others, allowing to inferre smoothness from the data. The Matèrn class is defined by:

$$k_{Matern}(r) = \frac{2^{\ell-\nu}}{\Gamma(\nu)}(\frac{\sqrt{2\nu r}}{\ell})^\nu K_\nu(\frac{\sqrt{2\nu r}}{\ell}),$$

with $r = |\mathbf{x} - \mathbf{x}'|$, positive parameters ν, ℓ, a modified Bessel function K_v.

gp Function Description		
Computation of marginal likelihood	Specification	Parameters
gp(hyp, @infExact, meanfunc, covfunc, likfunc, x, y) hyp - hyperparameters structure x - training input y - training output		
meanfunc = {@meanSum, {@meanLinear, @meanConst}}	mean function	@mean linear - linear @meanConst - constant @meanSum - addition
covfunc = {@covMaterniso, 3} hyp.cov = log([ell; sf])	covariance function Matern form	ell - characteristic length-scale sf - standard deviation of the signal sf
likfunc = @likGauss	likelihood function	Gaussian
@infExact	exact inference method	exact

Table 4.1: Gaussian Processes for Machine Learning Setup.

In order to implement the GP inference for prediction, the Gaussian Process Machine Learning (GPML) package was used [118]. The function **gp** does posterior inference, learns hyperparameters, computes the marginal likelihood and makes predictions. The function is called with a structure of hyperparameters, inference method, mean, covariance and likelihood functions together with the inputs and outputs of the training data. The negative log probability of the training data is returned when the function is called without test inputs. When test inputs are given as an argument, the prediction is computed and the mean and variance at the test location is returned4.1.

4.2 Experimental Results

The data corresponds to the traffic, that took place during a period higher than two years, and was used as a case study to test the different predictors. The methods have been trained on three data sets, each having two time intervals. The training and test sets used for the prediction are composed of different periods of the day, in order to validate the behavior and the quality of the solutions given by the predictors, in various scenarios. The predictors presented in this chapter are trained and validated using the experimental setup described in the following.

4.2.1 Setup and Data Collection

Different time frames and scenarios with static and dynamic setups for the predictors are considered. The output given by the algorithms is the predicted number of calls during a future time frame. The predictors presented in this chapter are trained and validated using the experimental setup presented in Table 4.2. The training sets have two time frames each, such that the dimension of the space is $d = 2$.

[5] `http://scikit-learn.org/stable/modules/gaussian_process.html`

Experimental Setup		
Number of time frames	*Training set*	*Hours: 10 am/11 am 12 pm/ 13 pm ; 20 pm/ 21 pm*
	Test set	*Hours: 10 am/11 am 12 am/ 13 pm ; 20 pm/ 21 pm*
Number of days	*Training sets*	594
	Test set	249
Interactive Particle System	*Number of particles*	1000
	Dimension of space	2
	Number of iterations (perturbation step)	100
Gaussian Mixture Model	*Maximum number of Gaussians for training*	10
	Number of iterations for training	100
Gaussian Processes	*Covariance function (Matern form)*	@covMaterniso
	Likelihood function (Gaussian)	@likGauss
	Mode	exact
	Number of evaluations to perform	300

Table 4.2: Input data set and parameters for IPS, GMM and GP.

1. **Static scenario**

   In this scenario the data is split into training and validation set. The training set is a matrix with two columns that represent data in two time intervals. The test set is the data shifted to the time interval that is not used for training (Figure 4.3).

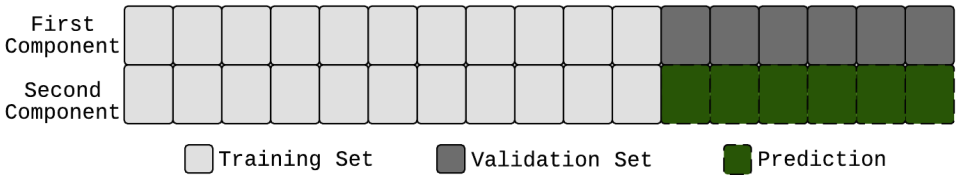


Figure 4.3: Data extraction - Training and validation set, values to predict.

2. **Static scenario with data shuffle**

   A dynamic predictor capable of describing the time-changing structure of the data is required for the evolving trend inside the data. In order to (i) assess the validity of this assumption and (ii) construct a rough lower bound predictor, it is considered a scenario where the full data set is shuffled before being split into training and validation data. Such a setup provides an *a priori* knowledge on the number and evolution

of calls (during the training phase). While such an approach is not applicable in practice, it allows to construct a predictor which better captures the overall structure of the data. This predictor is used as a comparison basis for other predictors, when discussing both the static and the dynamic setups. In this scenario the data is shuffled (Figure 4.4) before splitting it into the training and validation sets (Figure 4.3). The predictors presented in the previous subsection are trained and validated in the same manner (Table 4.2).

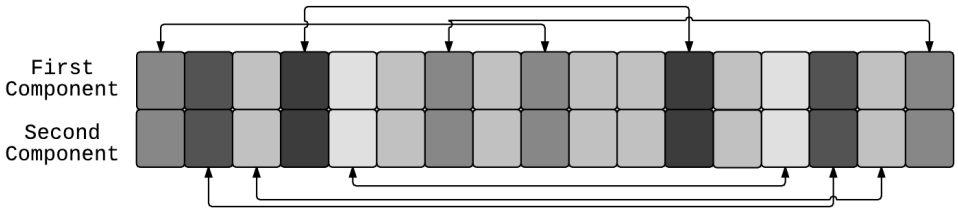


Figure 4.4: Data shuffle before splitting into training and validation set.

3. **Dynamic scenario**

   For the dynamic setup of the predictors, the training set is split and ordered into a collection of observations, each one being a record of the traffic that took place during a time interval. For a number of observations Q and a window size q, the predictors are trained for $m = Q - q + 1$ steps and m values are predicted(Figure 4.5). At each iteration, only one value is predicted and the calculated error is computed, and used for Mean Absolute Percentage Error (MAPE) evaluation.

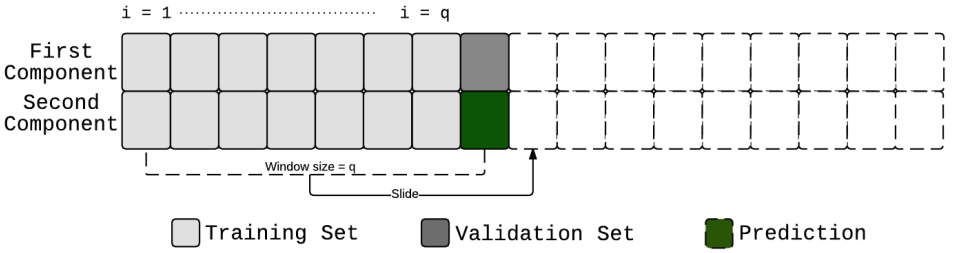


Figure 4.5: Siliding window approach example.

Often used in statistics to express accuracy and compare fitted time series, the Mean Absolute Percentage Error (MAPE) is calculated as: $M = \frac{1}{N}\sum_{i=1}^{N}|\frac{A(i)-R(i)}{A(i)}|$, with the actual set A, and the result $R(i)$. MAPE has no restriction in terms of value on the upper level and it is equal to zero when the fit is perfect.

An ANOVA test [119] is used to validate the results, by applying unpaired multiple comparisons between the predictors. The average of the errors given by the predictors, after 30 iterations, is calculated, independent samples given as input for the statistical test.

4.2.2 Comparison between predictors

The predictors have been trained and tested according to the experimental setup presented in the previous section. Under different scenarios and for different time frames, the errors given by the predictions are calculated and used as input for the ANOVA test, to apply unpaired multiple comparisons between the predictors.

1. Static scenario

A number of $N = 1000$ particles is generated using Interactive Particle System (IPS), and each encodes a vector μ and a matrix Σ. Using the input samples, the likelihood of each particle is calculated (initial population). The perturbation step is applied for a number of times and only a percentage (20%) of particles will survive at each iteration (mutation - selection step). The final population of the algorithm is extracted and used to calculate the prediction, by either considering it (weighted likelihood) or by selecting the particle with the maximum likelihood.

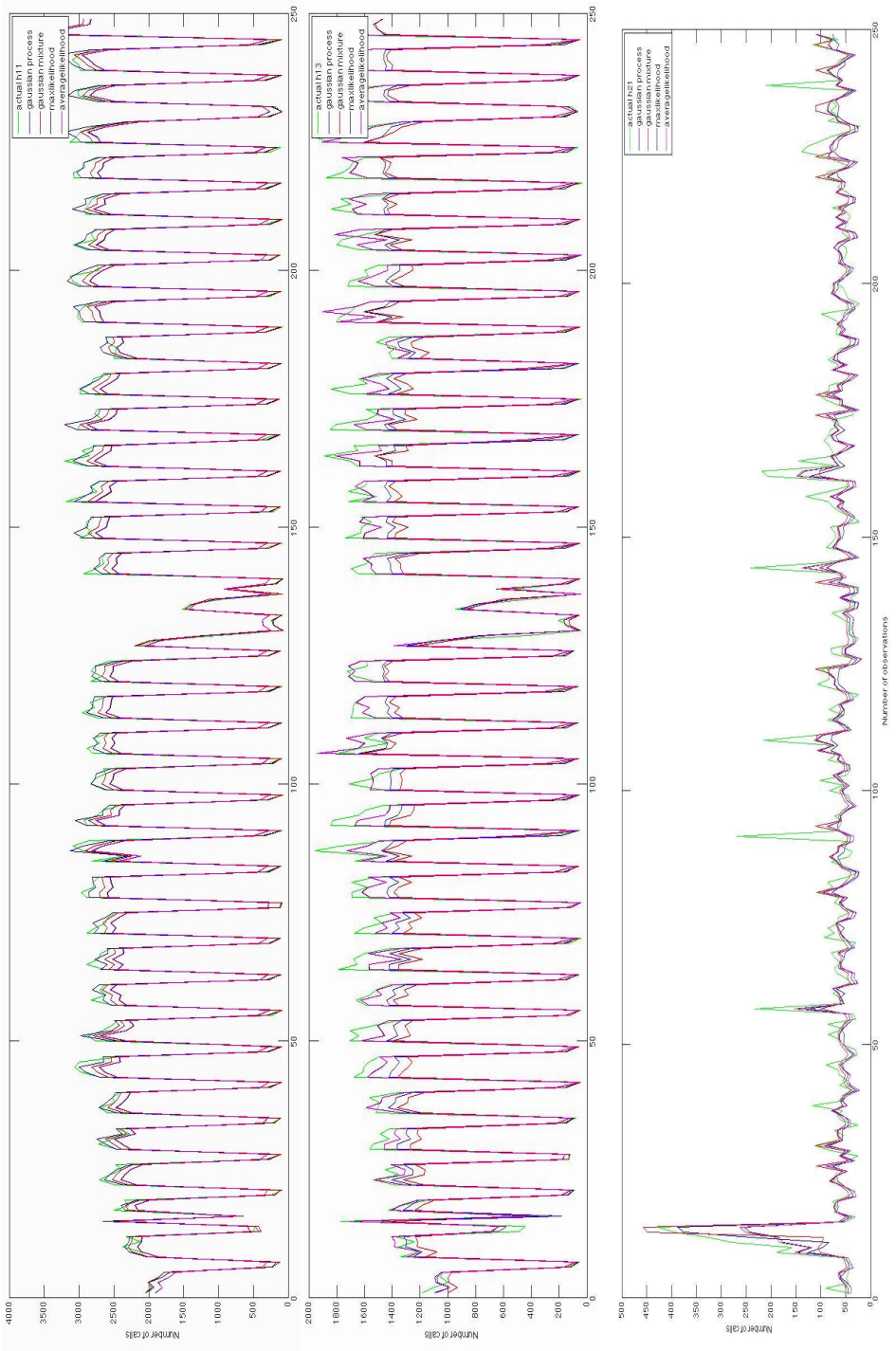
The Gaussian Mixture Model (GMM) is trained with the same input samples. The maximum number of Gaussians used for training is 10. AIC score is used to measure the model that fits best the data. For example, if the smallest value for the AIC is given by the GMM with 2 components, the specific mixture will be used for prediction. The number of iterations applied for training is 100, and at each iteration the parameters are updated using an expectation-maximization-approach.

We train and use the Gaussian Process (GP) method to predict the traffic over a future time frame. The algorithm runs with the parameters: training set, test set, number of iterations (300) and estimation of the parameters (exact).

In Figure 4.6 the results of the prediction together with the actual values, are plotted. The standard deviation of each classifier is shown in Table 4.3. We identify GP, as best performing for the first data set (10am - 11am) while GMM performs best on the second data set (12pm - 13 pm). For the last set of data considered (20pm - 21pm) Interactive Particle System - Average Likelihood (IPS-AL) performs best.

2. Static scenario with data shuffle

In this scenario the data is shuffled (Figure 4.4) before splitting it into training and validation set (Figure 4.3). The predictors presented in the



(a) Data set 10am - 11am (b) Data set 12pm - 13pm (c) Data set 20pm - 21pm

Figure 4.6: Results of prediction for each classifier in the static scenario.

Classifiers	H10 am/ H11 am		H12 pm/ H13 pm		H20 pm/ H21 pm	
	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation
Interactive Particle System Maximum Likelihood	0.2283	0.1227	0.1623	0.0112	0.2441	0.0058
Interactive Particle System Average Likelihood	0.2335	0.1279	0.1644	0.0101	**0.2432**	**0.0053**
Gaussian Mixture Model	0.0831	0.0030	**0.1527**	**0.0142**	0.3332	0.0274
Gaussian Processes	**0.0709**	**4.2345e-17**	0.1544	8.4690e-17	0.2520	1.1292e-16

Table 4.3: Mean Absolute Percentage Deviation and Standard Deviation for each classifiers, in average, after 30 runs.

previous subsection are trained and validated in the same manner (Table 4.2). In Figure 4.7 the results of the prediction together with the actual values, are plotted. In Table 4.4, MAPE and the Standard Deviation are presented for this scenario.

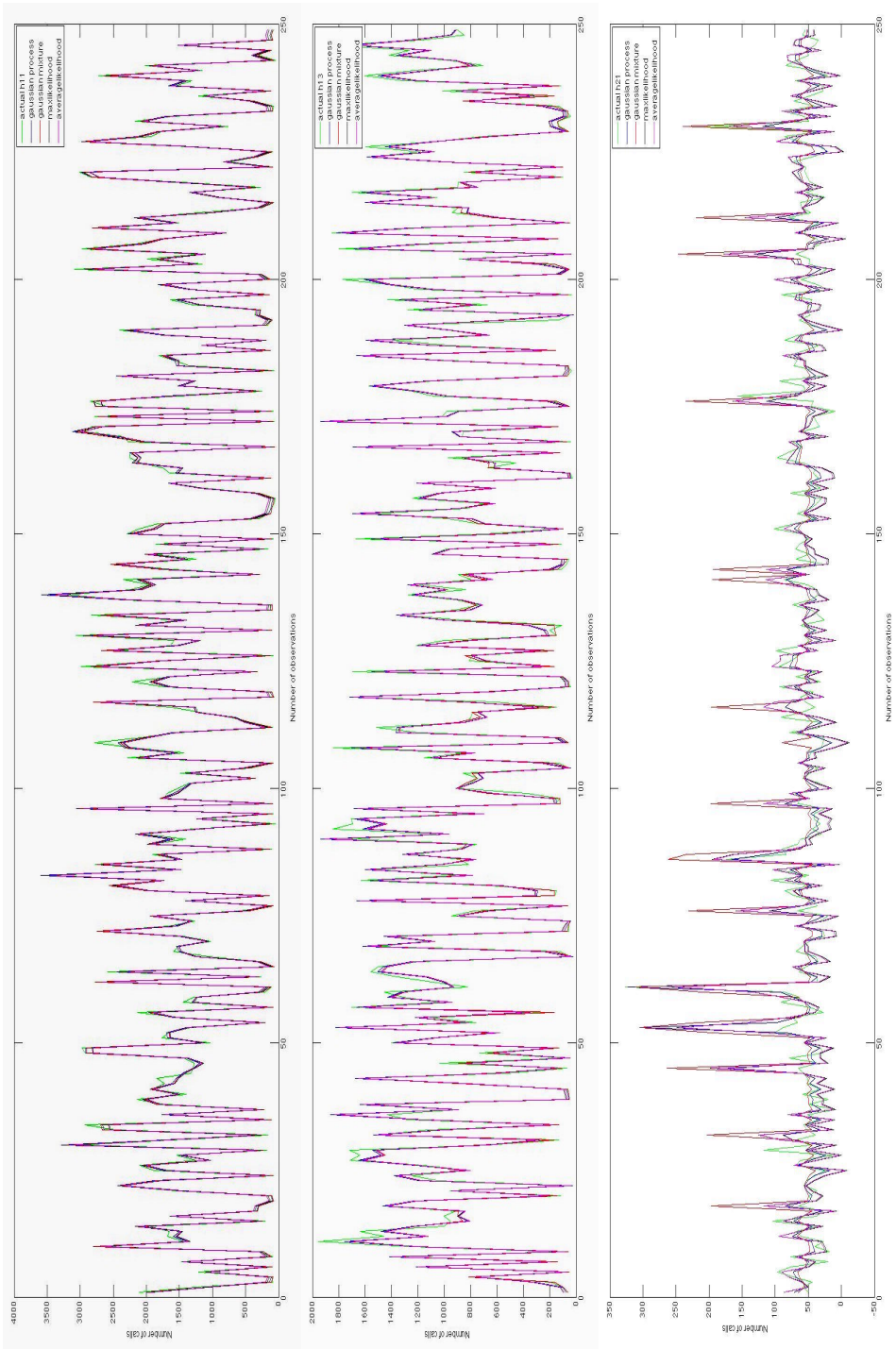
In this scenario, GMM performs best for two different data sets (10am - 11am and 12pm - 13pm) while the other three predictors (GP, IPS-AL and Interactive Particle System - Maximum Likelihood (IPS-ML)) give better results on the 20pm - 21pm data set.

Classifiers	H10 am/ H11 am		H12 pm/ H13 pm		H20 pm/ H21 pm	
	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation
Interactive Particle System Maximum Likelihood	0.5815	0.3034	0.2183	0.0551	0.3641	0.0381
Interactive Particle System Average Likelihood	0.5979	0.3233	0.2227	0.0539	0.3630	0.0303
Gaussian Mixture Model	**0.1047**	**0.0016**	**0.1423**	**0.0108**	0.4928	0.0219
Gaussian Processes	0.1259	0	0.1921	5.6460e-17	**0.3114**	**0**

Table 4.4: Mean Absolute Percentage Deviation and Standard Deviation for each classifiers, in average, after 30 runs.

3. Dynamic setup scenario

Dynamic setup for the predictors is presented in the following: for a number of observations Q and a window size q, the predictors are trained



(a) Data set 10am - 11am (b) Data set 12pm - 13pm (c) Data set 20pm - 21pm

Figure 4.7: Results of prediction for each classifier in the static with data shuffle scenario.

for $m = Q - q + 1$ steps and m values are predicted one by one, and MAPE is computed.

IPS-ML and IPS-AL are trained and tested, as in the static setup scenario, for a number of iterations $m = 100$. First, the N particles are generated and their likelihoods are calculated. The perturbation and acceptance phases are applied until the final population is resulted. Depending on the chosen method (IPS-ML or IPS-AL), one value is predicted. For a number of $m - 1$ times, these steps are repeated. The results are stored and compared with the real values.

GMM is trained with the first set of observations, and the parameters of the model with the best AIC score are estimated. In order to predict the first value, the parameters are updated using an expectation-maximization-approach. These steps are repeated for $m - 1$ times in order to complete the prediction and calculate the errors.

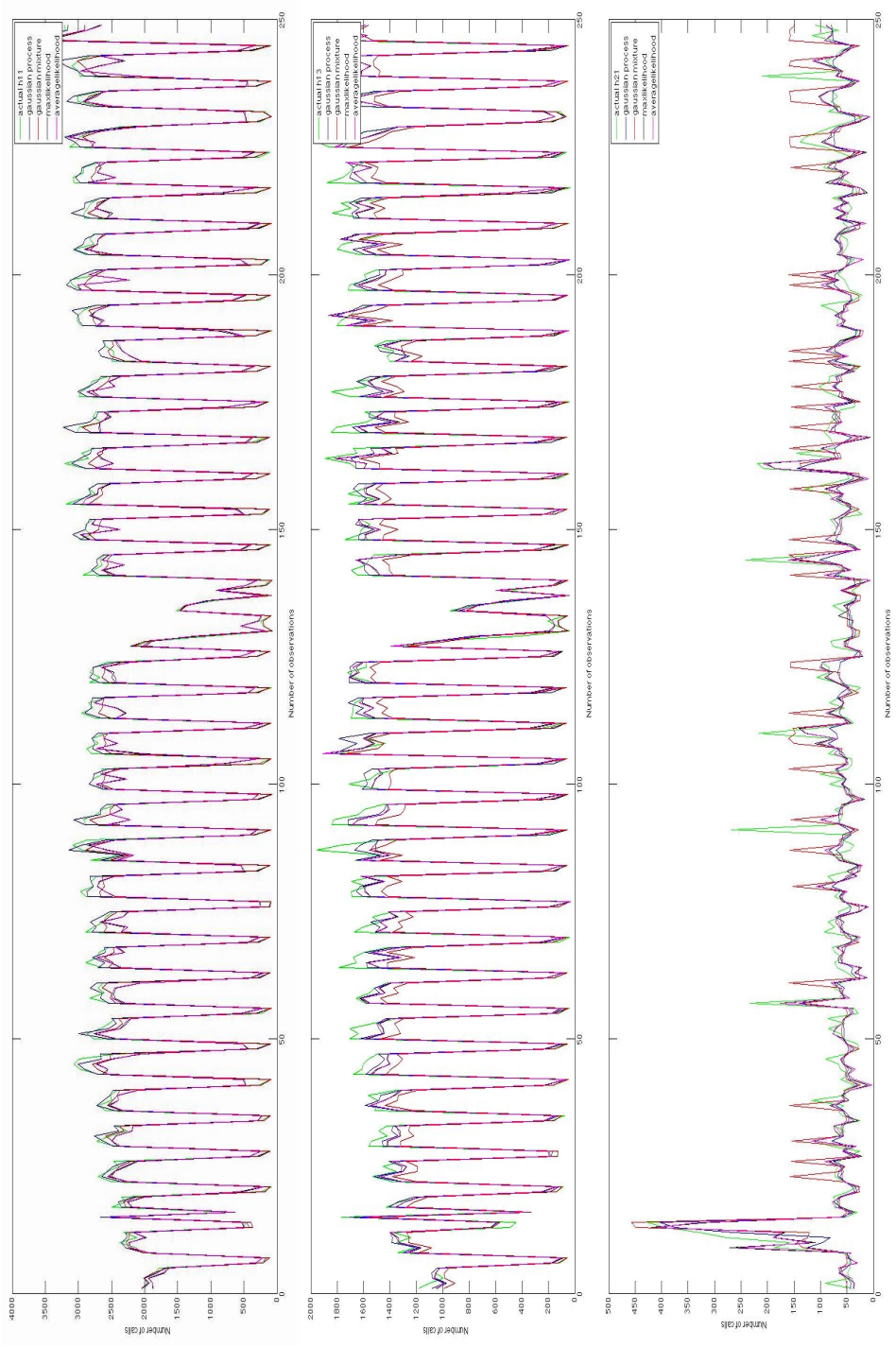
The GP is trained 100 times with the first set of observations. The first predicted value and the parameters of the GP are calculated. After building the first GP, its parameters together with the second set of observations are considered to build the second GP. For $m - 1$ steps, the GPs are trained 10 times, and the predicted values are computed with the errors given by the prediction.

In Figure 4.8 the results of the prediction together with the actual values, are plotted. In Table 4.5 the errors of the predictors in the dynamic scenario, MAPE and Standard deviation are presented. Compared with the previous scenarios, two predictors (GMM and GP) are improved.

	H10 am/ H11 am		H12 pm/ H13 pm		H20 pm/ H21 pm	
Classifiers	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation	MAPE Value	Standard Deviation
Interactive Particle System Maximum Likelihood	0.5105	0.0250	0.2586	0.0193	0.2857	0.0122
Interactive Particle System Average Likelihood	0.5024	0.0242	0.2502	0.0193	0.2847	0.0106
Gaussian Mixture Model	**0.0857**	**0.0083**	**0.1487**	**0.0123**	**0.3389**	**0.0412**
Gaussian Processes	**0.0684**	**2.8230e-17**	**0.1238**	**4.2345e-17**	**0.2448**	**5.6460e-17**

Table 4.5: Mean Absolute Percentage Deviation and Standard Deviation for each classifiers, in average, after 30 runs.

In Table 4.3, Table 4.4 and Table 4.5, the average of the errors of the predictors, after 30 iterations, is presented. The errors given by each predictor are the independent samples given as input for the statistical test. In the static case, GMM and GP performs better than IPS-AL and IPS-ML



(a) Data set 10am - 11am (b) Data set 12pm - 13pm (c) Data set 20pm - 21pm

Figure 4.8: Results of prediction for each classifier in the dynamic scenario.

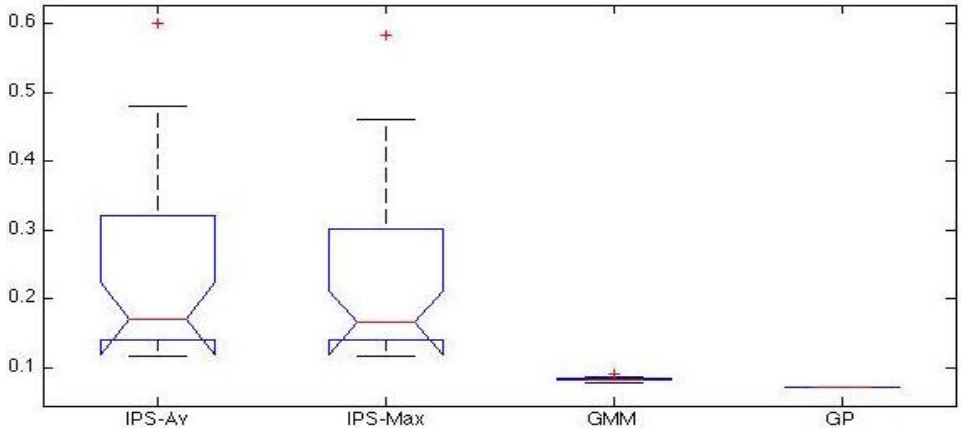
for the data sets: 10am - 11am and 12pm - 13pm (Figure 4.9 (a), (b)). In this scenario, for the 20pm - 21pm (Figure 4.9 (c)) data set, GMM gives the lowest quality solution in terms of the prediction.

In the static scenario with shuffled data GMM and GP perform better than IPS-AL and IPS-ML for the data set: 10am - 11am (Figure 4.10 (a)). For the data set 12pm - 13pm (Figure 4.10 (b)) GMM works better than all the other predictors along with GP. For the 20pm - 21pm data set GP gives the best solution followed by IPS-ML and IPS-AL (that are not significantly different), and GMM that performs worst (Figure 4.10 (c)).

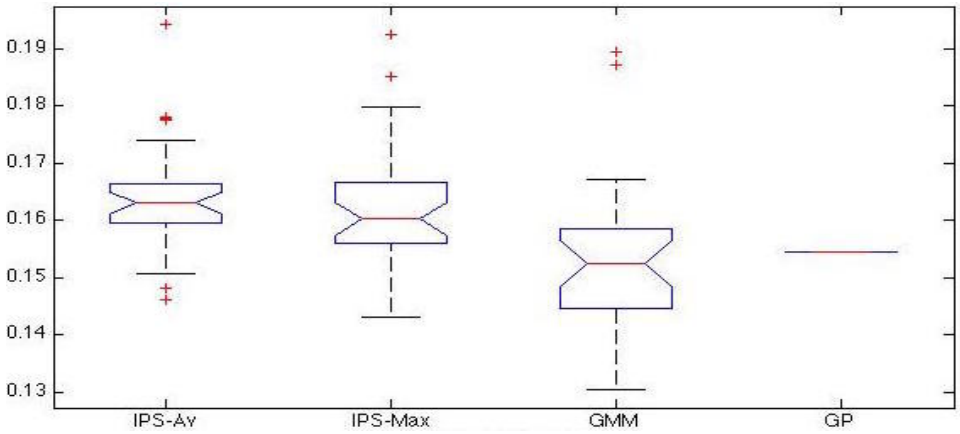
For the dynamic scenario, the results of the test show that GP works best followed by GMM for the data sets: 10am - 11am (Figure 4.11 (a)) and 12pm - 13pm (Figure 4.11 (b)), and the results given by IPS-AL are not signficantly different from IPS-ML. For the last scenario and the 20pm -21pm (Figure 4.11 (c)) data set the predictor that gives best solution in terms of quality is GP followed by IPS-AL and IPS-ML that are not significantly different and GMM that performs worst.

4.3 Summary

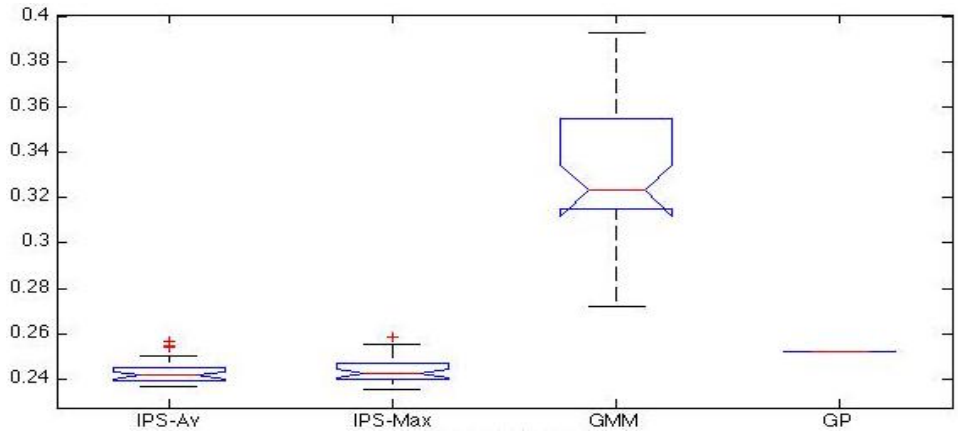
The prediction framework proposed in this chapter combines different methodologies used as an input for the load balancing model. The predictors considered in the context of the thesis are Gaussian Mixture Model (GMM), Gaussian Process (GP) and Interactive Particle System (IPS), trained and tested under different scenarios. Insights are provided on how particle algorithms are used for optimization, in conjunction with implicit learning models, strategies and scenarios. For the Interactive Particle System (IPS) algorithm, two different methods for predicting the load of the servers are explored and the results are compared. The first one takes in consideration the particle with the maximum likelihood and the second one is a weighted likelihood based prediction. An overview of previous work is provided, as Gaussian Mixture Model (GMM) and Gaussian Process (GP) are used for VoIP calls characteristics shaping, namely call duration and call holding times. The IPS methods are compared with the quality of solutions given by GMM and GP, trained and tested with the same data for comparison. None of the previous work considers GMM or GP for modeling the amount of incoming calls placed during a time frame in a VoIP system.



(a) Data set 10am - 11am

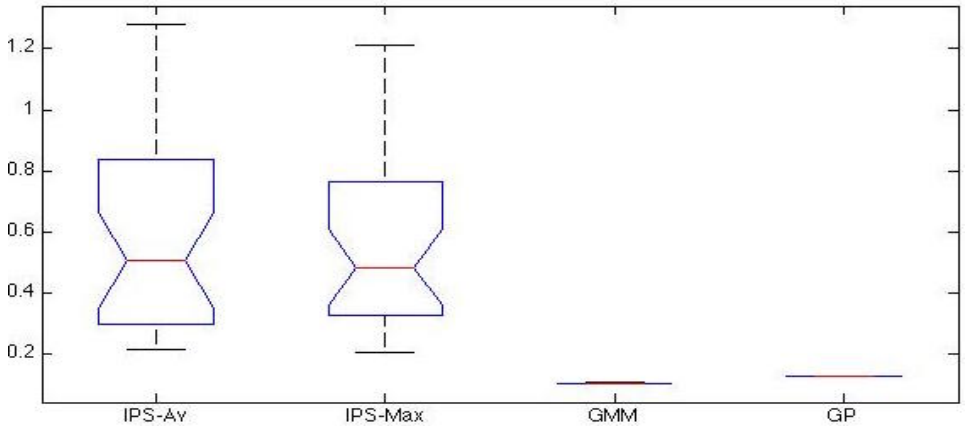


(b) Data set 12pm - 13pm

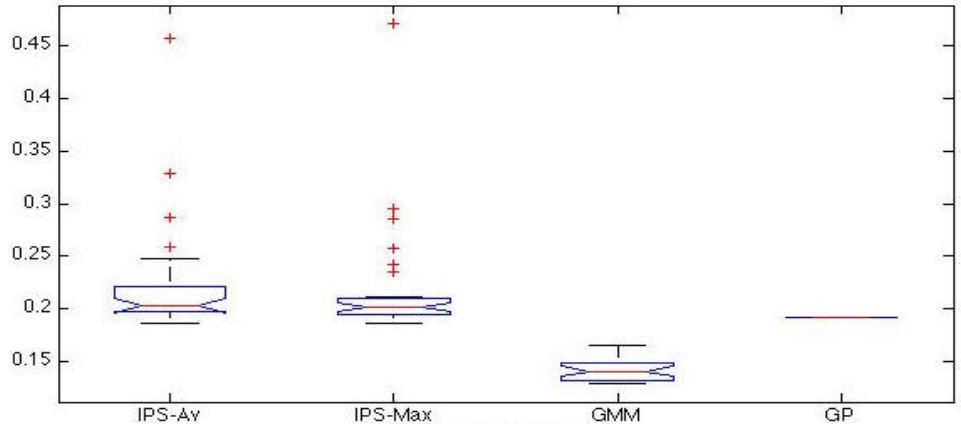


(c) Data set 20pm - 21pm

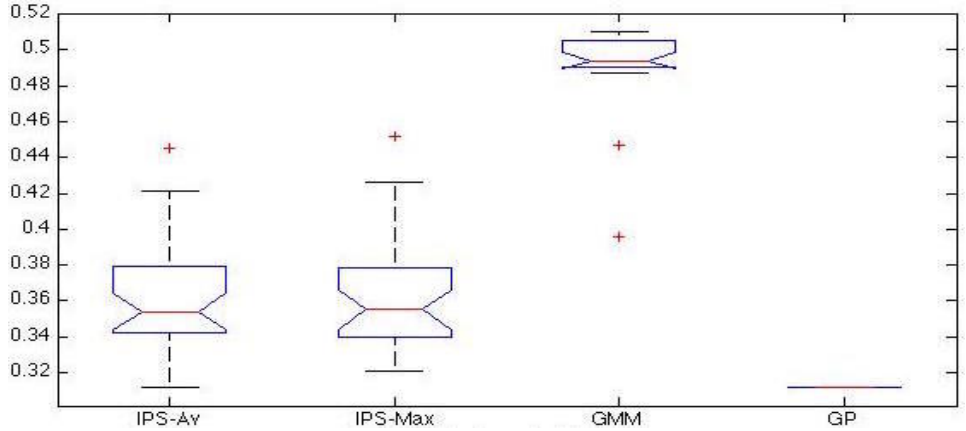
Figure 4.9: ANOVA test for the results given by the predictors in Static setup scenario



(a) Data set 10am - 11am

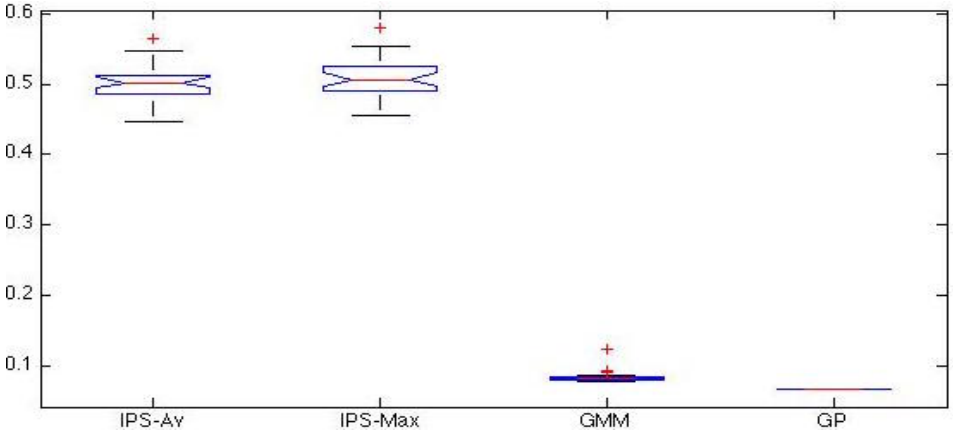


(b) Data set 12pm - 13pm

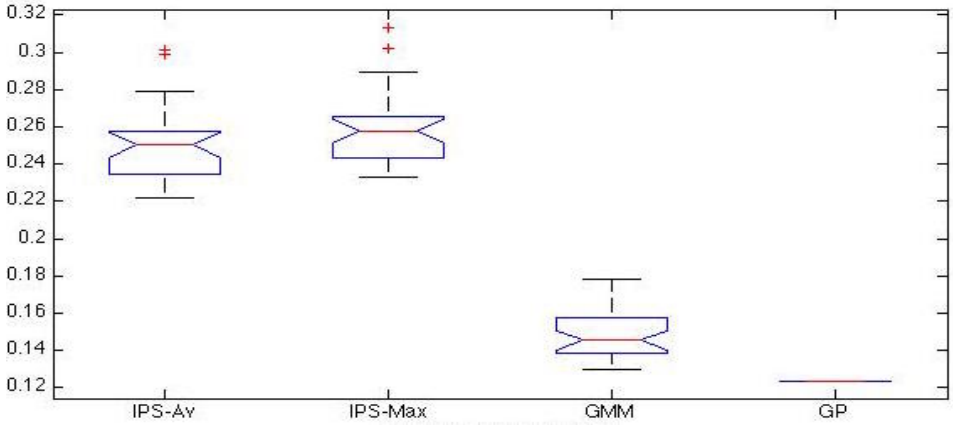


(c) Data set 20pm - 21pm

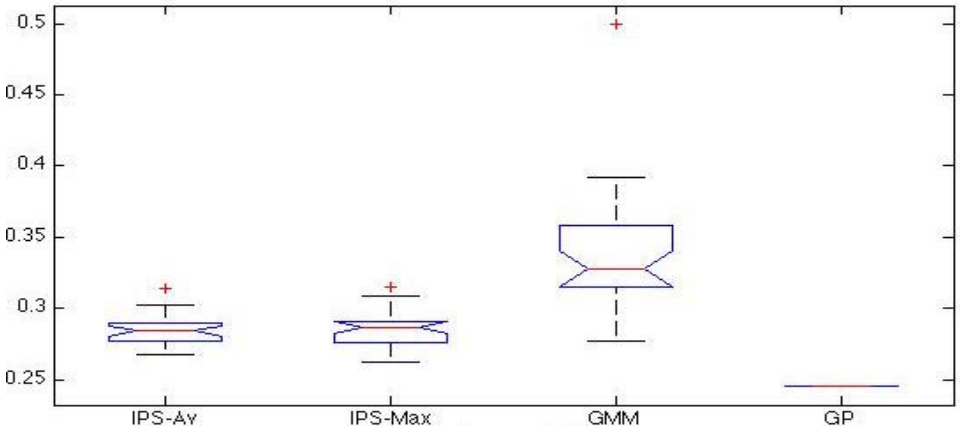
Figure 4.10: ANOVA test for the results given by the predictors in Static setup scenario with shuffled data.



(a) Data set 10am - 11am



(b) Data set 12pm - 13pm



(c) Data set 20pm - 21pm

Figure 4.11: ANOVA test for the results given by the predictors in Dynamic setup scenario.

Chapter 5

Load Balancing

Existing VoIP implementations with the technology in itself demand a sustained computational and bandwidth support. In this chapter, congestion and overload issues, at the level of cloud-based VoIP systems, are investigated. Novel solutions, that effectively combine resource allocation and load balancing methods, are presented. The proposed approaches are tested on synthetic data, benchmarks designed out of MIXvoip logs for their cloud-based environment currently in use. The information is gathered by first inspecting the system, and the data provided by the predictive algorithm is used to apply load distribution mechanisms and resource allocation. The most important cause of the load imbalance is the dynamic nature of the problem, with both computational and communication costs[87, 120].

Having the ability to cope with all the stochastic or time-dependent deterministic factors, that shape VoIP systems, represents a significant step forward from monolithic, classical solutions. This type of approach typically leads to not only an improvement of the service offered, but also to a sensible reduction of the associated carbon emissions, e.g. as a result of an improved load management, reduced idle CPU times or optimally exploited resource. The content of this chapter is based on two peer reviewed publications, one of which was presented during the conference: "Russian Supercomputing Days 2015" [5] and the other published in the "International Journal of Metaheuristics 2015" [6].

5.1 Design and Implementation of VoIP Load Balancing in Cloud Computing

In this section, a clear specification of the design axes required for a dynamic, autonomous VoIP cloud-based environment is given. Virtualization technologies enable the creation of VoIP virtual servers, which are then hosted in data centers and rented out (leased) on a subscription basis to any scale. Voice Service Provider (VSP) costs are in direct relation with the

type of infrastructure and its maintenance. The metric considered in the context of this study, for VSPs to evaluate the cost for the cloud solutions, is the number of VMs together with the period of their usage. In the following, there are described the VoIP cloud infrastructure model, the job model for scheduling and the optimization criteria considered for load balancing paradigms.

5.1.1 Formal Definition

Depending on their needs, VSPs that adopt as a solution Infrastructure-as-a-Service (IaaS), have a wide choice for demand and costs depend on the type of requested computing resources. Examples of existing costs are shown in Table 5.1, and constitues of the assets bought to run the infrastructure (CApital EXpense - CAPEX) and the expenses necessary to provide the service (OPerational EXpense - OPEX) [121].

Provider Cost	**Resource Type Example**
CAPEX	Machines, Servers, Storage, Interconnectivity, Room Equipment, Building Estimation
OPEX	Manpower, Energy (power and cooling), Hardware Support, Maintenance, Facility Costs, Software licensing (Platform and Application)

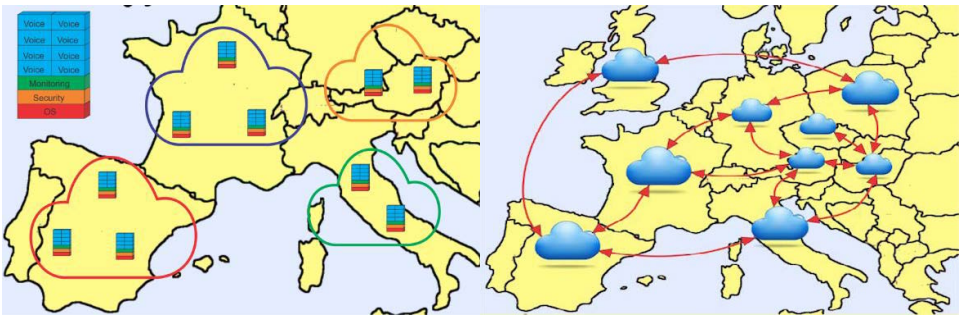
Table 5.1: VoIP provider cost categories example.

In general, the notion of machine covers a variety of concepts, namely physical servers, virtual machines, data centers. In the context of the thesis, it is considered that tasks are allocated on VMs, which are processors dependent on the involved layers of the VoIP cloud scheduling problem, furtherer allocated on servers. In the following, it is addressed the model for VoIP in distributed cloud environments, with high heterogeneity of resources, different number of servers, execution speeds, energy efficiency, amount of memory, bandwidth. For the optimization problem approached in this chapter, two objectives are considered: the minimization of the total number of billing hours for VMs and the VM utilization.

5.1.1.1 Infrastructure model

In Figure 5.1 it is displayed an example of a cloud based VoIP architecture, where voice nodes are grouped in geographically distributed data centers. The geographic location has a great impact on resource management, when considering the availability of auxiliary resources (electrical energy), regulations (legal restrictions for data processing). Different cloud providers, that

run data centers, may cooperate or not to build a common platform (cloud federation vs multi-cloud system). The design of a multi-level distributed VoIP load balancer is build to improve the local load imbalance in data centers together with forward-looking techniques to scale on federation of data centers (Figure 5.1).



(a) (b)

Figure 5.1: SNC in multi-cloud (a) and cloud federation (b).

Super Nodes Clusters (SNCs) are heterogeneous clusters of nodes, composed of multiple homogeneous machines deployed in cloud, interconnected logically at a local level (Super Node (SN)). SNs run one or multiple Asterisk processes, that handle calls with initiation/termination sessions (SIP), voice transmission processing (RTP), connectivity to the database to authenticate users and record the call transactions (CDR). The connectivity between SNs is provided through Internet or physical wire connections between two devices in a data center, where the operator and infrastructure meet on short distances.

Let us consider that the VoIP infrastructure consist of heterogeneous Super Nodes Clusters (SNCs). Each SNC incorporates multiple, identical, homogeneous SNs that have the same processing capacity. Each SNC_i has a relative speed s_i and consists of m_i set of SNs, with $i = 1 \dots m$. A supernode SN_k^i is part of the i-th cluster SNC_i and runs $k_i(t)$ VM at time t, with $k = 1 \dots m_i$. Each VM hosts one or multiple Asterisk processes launched to handle VoIP calls, and is described by its utilization (load) at time t, $vmu_i(t)$. This information is used to compute the number of billing hours for each SNC. The number of billing hours in SNC_i is denoted by $\overline{m_i} = \int_{t=0}^{C_{max}} k_i(t) m_i dt$. The total number of billing hours over the SNCs is the total number of billing hours of each SNC_i computed by $\sum_{i=1}^{m} \overline{m_i}$. The number of billing hours for VMs that provide VoIP services is one of the criteria considered for the cost minimization problem. The infrastructure model is sketched in Figure 5.2.

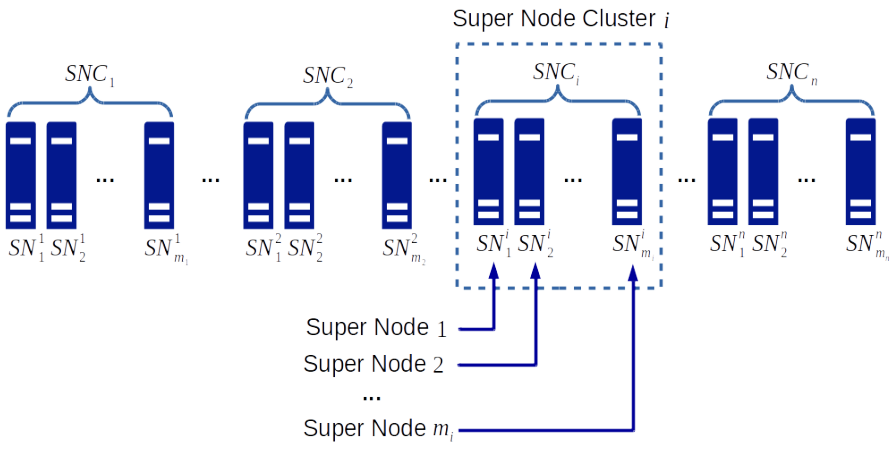


Figure 5.2: SNC infrastructure model.

5.1.1.2 Job model

VoIP calls have a different impact on processor utilization, that depends on the operations performed by Asterisk, for the establishment of sessions. For the placement of a phone call, firstly the session initiation is done followed by the transmission of voice packets. Thus, there is a spike in terms of utilization when a call is established. If transcoding operations are performed, the utilization is higher than when transcoding is not used. In the latter case, Asterisk [1] is in charge of only routing the call. Depending on the binary rate of the codec settled between the involved parties (UAC and UAS), the processor load is influenced as well. Many factors influence the performance of the systems that run Asterisk to process calls, including the type of channels used for the placement and receival of the calls [2]. The models presented in this chapter may be qualified as over-simplified in comparison with real industrial-scale cloud infrastructure providers, for example Amazon Web Services (AWS) [3] and Cloud Computing Platform [4].

In Table 5.2 there is an example of processor utilization for different number of calls, without transcoding [122]. Through experimental tests, the authors of [122] show that the impact on the processor load is driven not only by the codec but also both signaling and transport operations.

In [123], the authors present results of benchmark tests that include stress testing of Queue Calls, VoIP Provider Calls and Normal Extension

[1] http://blogs.digium.com/2011/10/03/top-10-tricks-you-didnt-know-asterisk-could-do/
[2] https://forum.openwrt.org/viewtopic.php?id=20312
[3] https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services//
[4] https://www.google.com/about/datacenters/inside/locations/index.html

Protocol	Codec	10 Calls	20 Calls
SIP/RTP	G.711	2.36%	4.64%
	G.726	2.13%	4.46%
	GSM	2.58%	4.55%
	LPC10	1.92%	3.61%

Table 5.2: Processor utilization
without transcoding example.

to Extension Calls. Queuing Calls are used by Call Centers to answer to the incoming calls automatically, place them in a queue instead of rejecting them. It allows the acceptance of more calls into the system than the existing extensions can support or human agents are capable of picking up. While on hold, the callers receive different announcements (position in the queue) followed by music (Table 5.3). Table 5.2 and Table 5.3 are used to calculate processor utilization per VoIP call, information given as an input for the synthetic benchmarks presented in Section 5.2.

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

Table 5.3: Processor utilization for Queue Calls example.

In Table 5.4 it is given an illustration of a CDR database, with its structure, enclosed fields and the job model for load balancing. The information is recorded during the placement of a VoIP call and is defined by the VSP depending on the billing plan, knowledge to be extracted. VoIP workload can be categorized as: outgoing/incoming internal/paying calls, friend call, redirected call through, fix/mobile call etc. In the scope of the thesis, the fields were extracted, analyzed and the few that were used as an input for the prediction tool and the load balancing algorithms (marked with green) consist of: the arrival date of the call, the duration in seconds. The type of codec is considered, for the reason that the load of a processor is influenced by the type of operations that take place (with/without transcoding).

Field Name	Description	Example
id	Index of call in the CDR File	100100
account_code	User id of the caller	50
IP	IP of the phone where the call is placed	127.0.0.1
dst	Destination number of call	444444
prefix	Prefix of the country where the call is placed to	00352
dst_name	Destination country name of where the call is placed	Luxembourg
carrier	Telecommunications service provider; It can be established which prefix goes to which carrier, decision based on price, quality	BT - British Telephone
calldate	Arrival of the call in the system	2016-05-01 00:03:00
duration	Duration of the call in second	38
codec_id	Name of codec used by Asterisk	G.711

Table 5.4: CDRs Structure Example. Job model for load balancing.

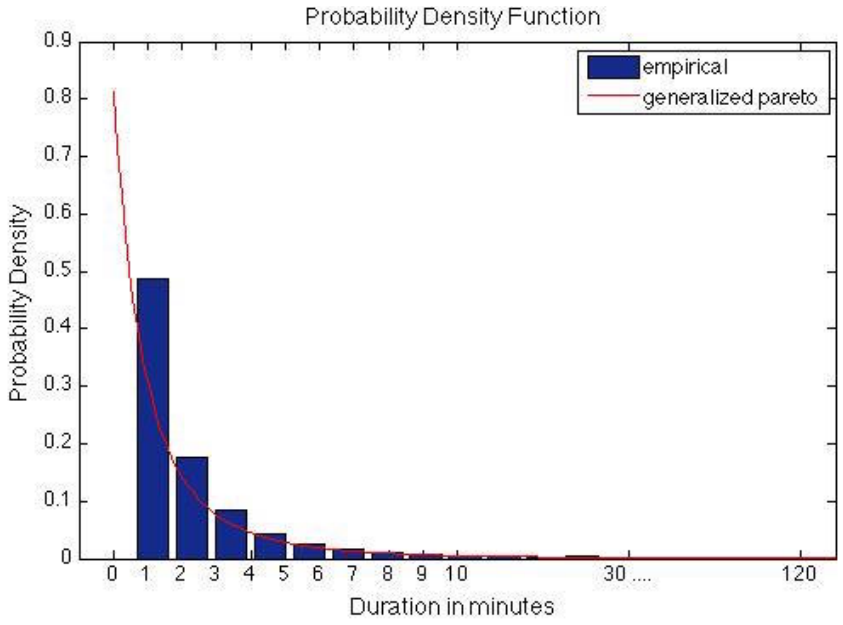


Figure 5.3: Example of call duration distribution and Generalized Pareto Distribution.

VoIP call arrival is modeled in [82] using Poisson process and the generalized Pareto distribution is used to model the call holding times. The silence and transmission durations are fitted by a generalized Pareto distribution as well. The authors analyze call level and packet level traffic, and test a series of probability distributions to fit the data. The generalized Pareto Distribution fits best the call holding times. The model agrees well with the

data in high-density regions and also fits the low-density regions, known as tails of the distribution. In Figure 5.3 there is an example of a generalized Pareto Distribution that fits the call duration in the tail distribution. It is visible that a very high percentage of calls have a duration under 5 minutes while very few are short calls.

Let us consider n independent jobs to be scheduled on SNCs, part of a federation of clouds. The job J_i is described by a tuple $< r_i, p_i, u_i >$ that consists of: its release time $r_i \geq 0$, the duration of the job and the contribution to the processor utilization. The release time of a job, r_i, is not available before the job is submitted; the duration of the job, p_i, is unknown until the job has completed its execution; and contribution, u_i, is a constant for a given job. Utilization is a constant for a given job , which depends on the used codec and is normalized for the slowest machine. With virtualization techniques and resource sharing mechanisms, the status of resources is constantly changing. Each job is allocated to one cloud and migration may be applied when necessary.

5.1.1.3 Optimization criteria

CSPs offer cloud instances competitive from a cost point of view. VSPs that adopt IaaS can determine the total price of a cloud service based on the cost for renting resources (storage space, server CPU time) and the rental time of the infrastructure (price influenced by length of commitment). For this reason it is important to consider the type of resources to rent, the time-stamp of the session and the rented time interval.

In Figure 5.4 it is shown an example of VoIP load balancing that considers utilization of VMs. VSP requires three machines to deliver VoIP services during a time frame, and the load is balanced between the VMs. VM_2 starts its execution when VM_1 is overloaded with the incoming traffic. In Figure 5.4 (a) there is shown a scenario with a VSP that rents three machines, which are underutilized, and the traffic is not balanced in the system. Load balancing methods are one approach to move forward from adopting over-capacity as a solution. In Figure 5.4 (b) it is presented an example of VoIP load balancing, where the load is distributed between VM_1 and VM_2. When VM_1 rental time is almost completed, a consolidation technique is applied to reduce VM_1 utilization (number of call running on VM_1). By adopting this approach, providers reduce the number of VMs rented for processing VoIP calls.

In Figure 5.5 it is presented an example of VoIP load balancing with rental of VMs that provide services with QoS. In this scenario, the VSP requires 3 VMs to handle the VoIP traffic that takes place throughout a day (Figure 5.5 (a)). In this case, the utilization of VM_1 is above the utilization threshold and QoS is not considered as an independent optimization criteria. By maintaining the utilization of VMs under the upper bound, while

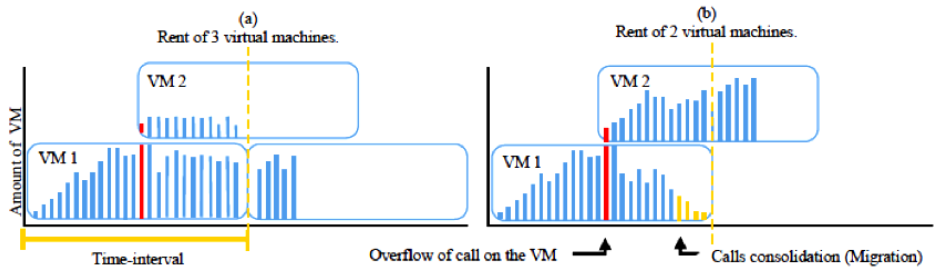


Figure 5.4: VoIP load balancing.

applying allocation strategies, QoS is ensured. The number of VMs required to process VoIP calls is reduced. The infrastructure model, in the context of the thesis, considers that the deployment takes place in geographically distributed locations, countries, by renting infrastructure from public or/and private clouds providers.

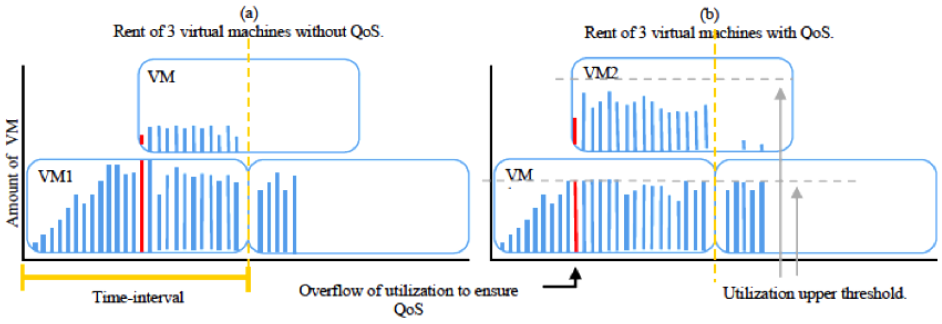


Figure 5.5: VoIP with Quality of Service.

For the optimization model, two criteria are considered, namely the *billing hours* of VMs to provide a service and their *utilization* for ensuring QoS. There is a wide variety of cloud pricing models for virtual servers, provided by CSPs, from which VSPs choose regarding their objectives and needs. In this study, the metric proposed considers both the number of rented instances and their usage period. The metric is used to compute the quality of the solutions given by different load balancing policies.

Generally, QoS standards for VoIP consider network metrics, key performance indicators. The most common measurements for network performance used for the delivery of VoIP services are: delays for packet delivery (latency) and its variations (jitter), packet loss, often caused by poor network configurations. In order to improve how well the network is accommodating VoIP traffic, we look into the number of concurrent session to configure by CPU. High CPU utilization on a server can disrupt VoIP services and when overloaded, sessions cannot be initiated, leading to users complaints and

dissatisfaction.

In the first scenario a single-objective optimization problem is considered: the minimization of the total cost of rented VMs. In the second scenario a bi-objective optimization approach is considered: the minimization of the total cost of rented VMs and the VM utilization. A set of solution is found, known as a Pareto optimal set, and depending on the VSPs preference, there will be a trade-off between the objectives.

5.1.2 Adaptive VoIP Load Balancing Model for Hybrid Clouds

In [124], work supported by NASA and Irvine Research Unit [5], the authors present a distributed algorithm for load balancing, based on the local rate of change, *Dynamic RoC-LB*. In their view, load balancing can be used as an efficient mechanism for improving the throughput and for speeding up the execution of the tasks. With preference for high processor utilization, the processing elements are kept busy by not considering only the number of running tasks but also utilization. The authors consider dynamic strategies in their study, applied in order to minimize the average completion time of the applications that run in parallel in multiprocessor systems, and to improve the utilization of the processing elements. Their approach is a load balancing strategy with the following characteristics: dynamic (run-time decisions), distributed (locally and asynchronously), on demand (nodes require tasks to avoid starvation), preemptive (tasks may be interrupted, migrated and restarted on the new host node) and implicit (no user assistance required).

The *difference in load* is an adaptive parameter calculated independently for each processing element, used to compute the number of sampling intervals required to reach an idle state, and to predict the amount of tasks to be finished in subsequent intervals. When its value is below zero, the element risks starvation and it requests tasks. Migration decisions depend the on current load, the load changes in the time interval (Rate of Change), and the load balancing parameters. Older tasks have priority for migration, with a higher probability to amortize the cost of migration. The load of each processing element is computed, and depending on its status and thresholds, defined actions are triggered. The difference in load is a multiple of the time slice duration, while a fine sampling of the time slice implies a higher overhead.

The algorithm considers three thresholds: *upper bound* U_b, *lower bound* L_b, and *critical bound* C_b. Each node holds two tables: a *sink* (with nodes that initiate load requests) and a *source* table (nodes that receive load requests). When the load of a node is larger than U_b, it becomes a *source* and immediate actions are taken to avoid spikes in the system. If the pre-

[5] `https://2015.spaceappschallenge.org/location/irvine-ca/`

dicted load is less than L_b, the node is a *sink*. When the current load is between these two bounds, the node is in the neutral state. However, if the load or the predicted load falls below C_b, the node immediately initiates requests for incoming load. A summary of the algorithm is presented above; for an in-depth description of the algorithm, please refer to the work in [124].

VMA-AdRoC is a robust extension of the *RoC-LB* algorithm [5]. It is particularly useful for systems that handle VoIP traffic, sensitive to poor network configurations and poor resource distribution management. In *VMA-AdRoC*, the accuracy of each balancing decision depends on the actual cloud characteristics at the moment of balancing. Cloud parameters are changing over time and balancing parameters should be adapted to these changes. This dynamic approach can cope with different workloads and cloud properties.

In the initial study, the algorithm is designed with adaptive parameters, used by the each processing element to calculate the number of sampling intervals. The length of the sampling interval is a multiple of the considered time slice duration, with particular value for each SN, for example the number of load requests received. In *VMA-AdRoC*, the sampling interval is an adaptive parameter calculated as: $s_i = [t - s_i, t]$, for each SN at time t. Finer sampling trigger load balancing actions, with a negative impact by generating a larger communication overhead.

In *RoC-LB*, the difference in load is a value used to predict the number of tasks to be finished in subsequent intervals; each processing element assumes that its own difference in load will remain the same forever. This constraint is relaxed in *VMA-AdRoC* in the sense that prediction is used to compute future utilization not by only considering the difference in load, but also the adaptive parameter, s_i. The utilization change is expressed as:

$$\Delta_i = \frac{u_i(t) - u_i(t-s_i)}{s_i},\ u_i(t)\ \textit{utilization of}\ SN_i\ \textit{at time}\ t.$$

SN_i is using Δ_i to predict its own future utilization. Δ_i is the utilization change that takes place during the sample interval s_i and represents the utilization change, the consumption speed. It can be also used to estimate the number of sampling intervals to reach an idle state. $rd_i(t)$ is the response delay at time t, an adaptive parameter defined as the time frame between the initiation of a load request and the reception of load. This is another main difference in respect to *RoC-LB*, which is predicting the number of sampling intervals to reach an idle, a fix value for each SN.

VMA-AdRoC is a two-parameter problem, meaning that the traffic is balanced between SNs by considering both utilization , $u_i(t)$, and answer time of a request, $rd_i(t)$. To define where a load is requested from or send to, each SN keeps two lists. The sink list records SNs that previously requested jobs, and source list enrolls SNs that previously offered jobs. When the time

to reach idle state is less than $rd_i(t)$, SN initiates a migration request. Let us note that s_i and $rd_i(t)$ have independent values for each SN.

In *VMA-AdRoC*, unlike *RoC-LB*, when utilization is above U_b the SNs send jobs to the *sinks*. SNs that initiate requests are part of the *sink* list. A sink selects a SN from its *source* list for a load request, and sends a requesting message. The source can accept the request or broadcast the request to other SNs, from its own source list. SNs do not send concurrent load requests; it waits an answer for previous requests until sending launching another one. The result of this message is the load coming from other SNs or the request comes back as unfulfilled.

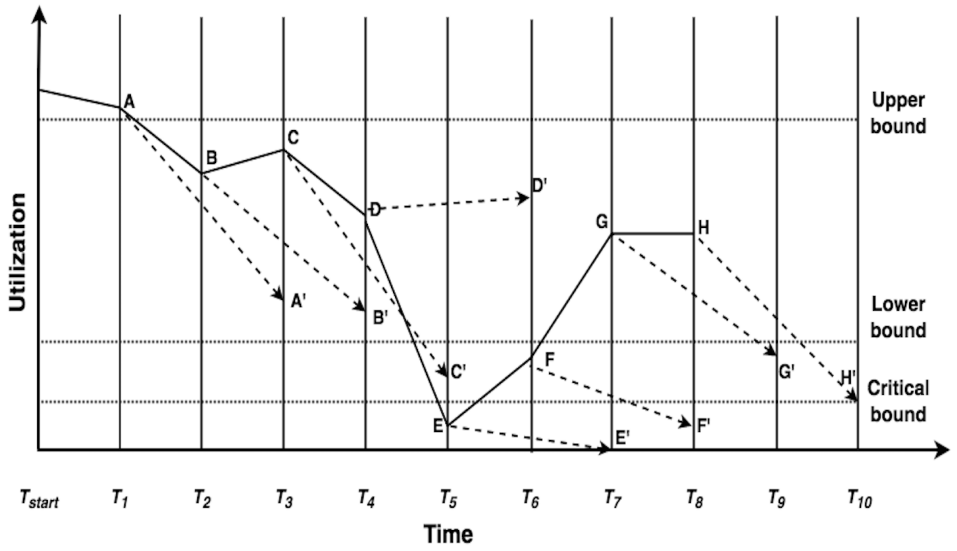


Figure 5.6: Dynamic load balancing scenario.

Figure 5.6 shows possible load balancing scenarios. Solid line show real utilization, dashed lines are predicted utilization. At time T_5, E initiates a request for load, regardless of the predicted future utilization, based upon the estimation Δ_i value; or it initializes migration of the load to other VM. Estimation F' on T_8 is under C_b, but cannot initiates a new request since it is not yet T_7, T_8; a new request is generated at T_8 if the prediction is accurate. Another main difference between VMA-AdRoC and RoC-LB, is that in the case when the utilization is above U_b, SN sends immediately jobs to the *sinks*. The modification of this action, in our algorithm, considers the sensitivity of VoIP towards overload of resources.

5.1.3 Global Server Load Balancing Model with Prediction

Different mechanisms for allocating the incoming calls for an optimal spread of the traffic on the distributed processor and servers in the cloud were

investigated. A load balancing model was built using an Integer Linear Programming (ILP) approach [6]. The model integrates the prediction of the incoming traffic and has as objectives the minimization of the cost by minimizing the number of running machines that handle the placed calls. ILP models have been adopted for diverse optimization problems [125]. The development of the allocation model for simultaneous calls was done using linear programming platform IBM CPLEX [7] and a high performance cluster (HPC) [8] was used to run the experiments.

The SIP flow is depicted in Figure 2.6. As a first step, every IP phone does a DNS lookup that results in receiving an IP address that binds to one of the voice nodes that handles incoming traffic. In Figure5.7 there is an example of Global Server Load Balancer (GSLB) that was investigated in the context of the Supernode2 Project from the Ministry of Economy Luxembourg. High availability and fast response time are the major factors considered to investigate the use of GSLB. A LB can be fitted into the DNS framework to provide GSLB and the best voice node to handle a call can be selected by checking the site health conditions, site response time, site load conditions, geography-based site, routing cost and so further.

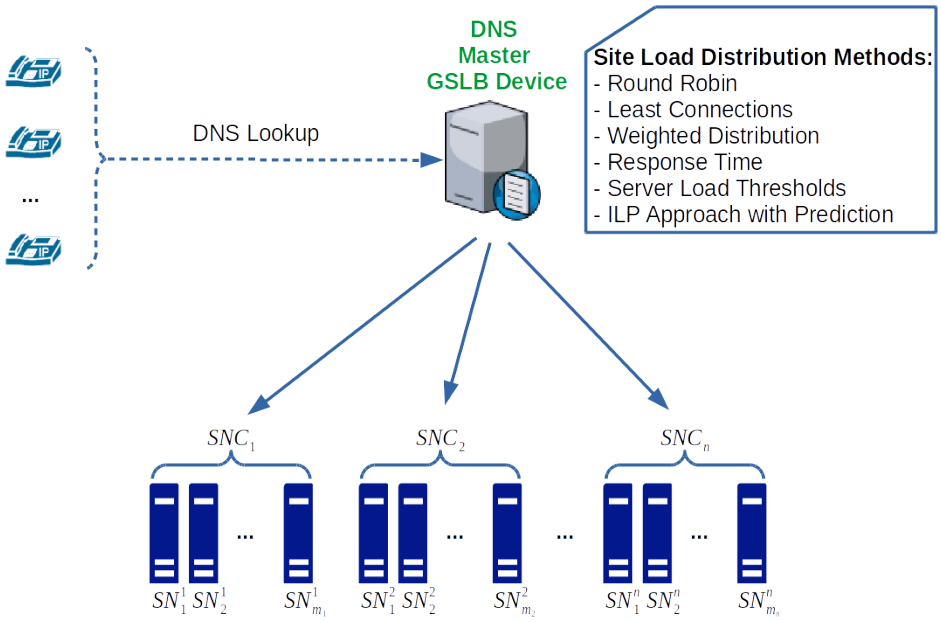


Figure 5.7: GSLB Distribution Methods.

[6] Work performed during the affiliation with The University of Sydney, Centre for Distributed and High Performance Computing, Professor Dr Albert Y. Zomaya., November - December 2013.

[7] `https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`

[8] `https://hpc.uni.lu/`

Using GSLB the request of a user can be redirected to one of the locations that provides the best response time while operating the voice applications server farms at multiple data centers. The most common used method for GSLB is DNS and as previously mentioned, different ways exist for the selection of the voice node that handles incoming traffic. In order to ensure availability, which is the most important QoS aspect in VoIP systems, site load conditions can be measured. The voice nodes (SNs) may have different capacity and current load conditions, the GSLB load balancer can use different load distribution methods (Figure 5.7). Formally, the proposed model consists of the following constants, sets, variables and constraints.

Constants, Sets, Variables:
Let define:

- SNC a set of super node clusters, set of size m;
- SNC_j a super node cluster and a set of m_i of SNs and $j \in \{1, 2, \ldots, m\}$;
- SN_k^j a super node with identity k from the set of SNs in SNC_j and k in $1 \ldots m_i$;
- q as the number of predicted calls;
- x_l as call to be assigned with $l \in \overline{1, q}$;
- $limit$ as a vector of size m that defines the maximum number of calls supported by each SNC;

Constraints:

- $\sum_l x_{l,j,k} = 1$ meaning that call x_l can only be assigned to one SN_k part of SNC_j
- $SNC_j - SN_k^j \geq 0$ meaning that for each running $SNC_j = 1$, node k may be running calls, $SN_k^j = 1$
- $SN_{jk} - x_{ljk} = 0$ meaning that a call x_l may be assigned to the node SN_k part of SNC_j
- $\sum_l x_{l,j,k} \leq limit(j)/|SNC_j|$ meaning that the number of calls assigned to a super node will not exceed the limit.

ILP Model Pseudocode

Minimize $\sum_j SNC$
Subject To:

- $\sum_l x_{l,j,k} = 1$
- $SNC_j - SN_k^j \geq 0$
- $SN_k^j - x_{l,j,k} = 0$
- $\sum_l x_{l,j,k} \leq limit(j)/|SNC_j|$

Bounds:

- $0 \leq x_{l,j,k} \leq 1$
- $0 \leq SNC_j \leq 1$
- $0 \leq SN_k^j \leq 1$

return assignation

Using the ILP distribution method, the capacity of each SNC, the current load and the incoming traffic, namely the prediction, are considered by the GSLB. One challenge in this scenario is the load DNS caching of the DNS response. For example, after serving a client, the DNS server caches the response and uses it to answer subsequent clients until the cache record expires [9]. The ILP approach minimizes the number of SNC used for assignation while maximizing the number of queries per SN. However, this type of approach has its limitations due the local DNS caching problems that occur, easier to manipulate when the VSP is the Internet Service Provider (ISP) of the end users. Moreover, the number of concurrent connection capacity is a metric that does not consider utilization of resources and as such, developing dynamic bin packing for this type of VoIP infrastructure in cloud is considered in the following.

5.1.4 Multi-objective Scheduling in Cloud Infrastructure for VoIP platforms

Optimization methods are widely used in engineering design, decision-making problems, scientific experiments [126, 127]. There are two main types of modeling optimization problems, to compute one or multiple optimal solutions. When one objective function is used to find the optimal solution, one deals with a single-objective optimization problem. In the case there are two or more objective functions involved, multi-objective optimality is used to find one or multiple optimal solutions [128].

Using the decision-making process, regarding the rent of IaaS with optimal choices concerning provider cost in respect to QoS, solutions are found which are optimal in respect to both objectives. In the case of conflicting

[9] Client as standalone end user or company.

objectives, each one corresponds to a different optimal solution. Thus, the optimal solutions that arise due the trade-off between conflicting objectives, are important. Some solutions assign more importance to cost rather than QoS and otherwise, depending on the preference. When one objective is considered to be more important in comparison with the second one, the preference is given to those solutions, that are near-optimal in the preferred objective, even if the values given by the secondary objective are not among the best obtained.

The feasible objective space contains solutions that are optimal, named *Pareto-optimal*. These solutions form a curve, referred to as *Pareto-optimal* front. A solution A is said to dominate a solution B, if A is not worse than B in all objectives and A is strictly better than B in at least one objective. When checking the dominance between two solutions, A and B, as result three possibilities exist: solution A dominates solution B, or solution B dominates solution A, or none of the solutions dominate each other (A and B are non-dominated in respect to each other). The dominance relation is *transitive*, *not reflexive*, *not symmetrical*, *not asymmetrical*.

Solution A dominates a solution B, $A \preceq B$, when:

- B is dominated by A
- A is non-dominated by B or
- A is non-inferior to B

There are two types of dominance, namely *strong dominance* and *weak Pareto-Optimality*. The formulation above refers to *weak dominance*. *Strong dominance* is defined as: a solution A strongly dominates a solution B, if solution A is strictly better than solution B in all objectives. A *weakly non-dominated set* refers to the set of solutions that are not strongly dominated by any other member of the secondary set of solutions. Multi-objective optimization problems are handled using classical methods and evolutionary methods. Examples of classical approaches are the: *weighted sum methods*, the *weighted metric methods*, *goal programming methods*. The drawback of these methods is, through the conversion of the multi-objective optimization problem into a single-objective problem, the limitation towards finding solutions in the nonconvex region of a Pareto-optimal set. Multi-objective evolutionary algorithms are inspired from natural genetics and natural selection, artificially used to construct genetic algorithms and evolution strategies. The trade-off fronts, produced by multi-objective evolutionary algorithms, can be assessed using performance metrics to evaluate the closeness to the Pareto-Optimal front. For example, coverage of two sets is a metric that calculates the proportion of solutions B, weakly dominated by solutions in A:

$$SC(A,B) = \frac{\{b \in B; | \exists a \in A : a \preceq b\}}{|B|}$$

When $SC(A, B) = 1$, all members of B are weakly dominated by A, whereas $SC(A, B) = 0$ means that no member of B is dominated by A. Both $SC(A, B)$ and $SC(B, A)$ (the dominance operator is not symmetric) have to be computed to understand how many solutions of A are covered by B and vice versa.

5.1.4.1 Dynamic bin packing with open bins

In the scope of this study, a variation of the well-known one-dimensional on-line bin-packing problem is considered to address congestion and overload issues at the level of a cloud-based VoIP system. In [129] it is proposed a communication-aware model of cloud computing applications. Dynamic scheduling of VoIP services in distributed cloud environments and the allocation strategies are described in the following. The novelty of the proposed variation is contoured by the state of the bins, determined not only by actions of the decision maker during item allocations, but also by item completions after their lifespan. Bi-objective approximation is used to test the performance given by the on-line variant of the optimization problem, results presented in the Simulation Results Section of the current chapter.

Classical bin packing problems belong to the combinatorial optimization field, with the objective of packing items into uniform-sized bins, while minimizing the total number of bins and considering the capacity of each bin which can not be exceeded. Algorithms are often distinguished based on the nature of the input, namely *online* and *offline*. Offline algorithms process in advance inputs to guarantee good performance while online algorithms produce their output on the fly when the input is received [130, 131]. The classical bin packing problem is an NP-hard problem, meaning that the optimal solution may not be found in polynomial time [132, 133]. In a simplistic manner, *online bin packing* refer to the placement of items throughout time, items of which parameters are known at the arrival of the item in the system (item size and arrival time). Migration of items in online bin packing is not recommended as the overhead cost of migration might be high with negative impact on VoIP QoS.

Online dynamic bin packing is a problem introduced by Coffman et. all in [134], with the objective of minimizing the maximum number of bins used over a time period, without exceeding their maximum capacity. Items are packed in bins and their size, arrival are known at their arrival and their departure time, known when the item departs from the assigned bin. Online dynamic bin-packing can be furtherer categorized into *one-dimensional* or *multi-dimensional* problems, depending on the number of dimensions of the items that describe the input. A full description of online dynamic bin packing problems can be found at [135].

Let us define a variation of the one-dimensional online bin-packing problem, where items of arbitrary height are packed into a one-dimensional space

(bins with fixed capacity) efficiently. The items are received in a sequential manner and the scheduler decides whether the next arriving item is packed in the currently open bin or a new bin is opened. The placement decisions are made without having knowledge about the duration of the call, but only the contribution to utilization (due to the settled codec). However, the state of the bin is shaped by considering the allocation of the items and also the item completion after their lifespan. In the scope of the study, it is considered that the bins are always open, even completely packed, and dynamic. Items in bins can be terminated (call termination) and utilization is dynamic, changing at any moment. The size of a bin takes values between (0, 1], corresponding to VM utilization. The distribution of VoIP calls is performed place between the SNs, the bins in the context of the thesis. Each SN is a bin, assumed to be allocated on a single core processor of 2.0 GHz, where Asterisk is running to handle the incoming VoIP traffic.

Three well known bin packing strategies First-Fit (FFit), Best-Fit (BFit), and Worst-Fit (WFit) are adapted to the current model, to allocate VoIP calls to the SNs. An aforementioned aspect of the online bin packing problem is the unsorted input. However, the bins (SNs) are sorted in a decreasing order by their utilization. The performance of the bin packing strategies is compared to two other widely used allocation strategies, namely round robin and random (Table 5.5).

	Description
Rand	Allocates job j to a suitable machine randomly selected using an 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 SNs in a decreasing order by their utilization, and allocates job j to the first SN.
WFit	Sorts SNs in a decreasing order by their utilization, and allocates job j to the last SN.

Table 5.5: Allocation Strategies.

5.2 Experimental Results

5.2.1 Setup

The experiments are performed using CloudSim environment [10] which is a framework for modeling and simulation of cloud computing infrastructures

[10] CloudSim Implementation - CICESE, `http://www.cicese.edu.mx/`

and services [136]. It is a standard trace based simulator that is used to study cloud resource management problems. The CloudSim[11] platform was extended to include the new algorithms using Java (JDK 7u51) programming language [12].

The extension of the CloudSim consists of the development of new features that supports the dynamic arrival of the jobs (calls), updating the system parameters 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 [4]. Standard Workload Format (SWF) [13] is used with additional fields to process the calls.

The workload is a set of recorded phone calls successfully handled by the real environment from MIXvoip. The structure of the CDR database is exemplified in Table 5.4 and each each recorded call is described by the fields presented. CDRs can be used to analyze the performance of the VoIP system by applying statistical methods to compute incoming/outgoing call attempts, the number of successful and dropped calls, the number of rejected calls, the calls with a length shorter than the configured minimum call duration (MCD), the amount of calls with a packet loss rate higher than the configured limit or high jitter and latency, etc.

5.2.2 Simulation Results

In this subsection the problem of VoIP load balancing in cloud computing is modeled through mono-objective and bi-objective optimization. The two criteria examined are the minimization of the total number of VMs used for call processing and the minimization of resources utilization. In order to test the assumptions presented, the setup is presented together with the results of the synthetic benchmarks, and the mono and bi-criteria optimization problems are analyzed and approximated. In order to study the relation between total cost (the number of hours running VMs) and the utilization of the VMs, the threshold constraint is not considered in the initial experiments for the bi-objective analysis. The solution consists of a Pareto front that corresponds to the different design strategies in the associated variable space.

Mono-objective and bi-objective analysis are used to compare the quality of solutions given by the allocation strategies. The call allocation strategies are used for the distribution of the incoming traffic to one of the SN in the system. The phone calls (jobs j) are allocated to the suitable SNs in the following manner: randomly by using a uniform distribution ($Rand$), applying the Round Robin load distribution method (RR), and using the three known bin packing strategies ($FFit$ - allocation to the first SN avail-

[11] `http://www.cloudbus.org/cloudsim/`

[12] `http://www.oracle.com/technetwork/java/javase/documentation/index.html`

[13] `http://www.cs.huji.ac.il/labs/parallel/workload/swf.html`

able; *BFit* - allocation to the most loaded SN; *WFit* - allocation to the least loaded SN).

Mono-Objective Approximation. The aforementioned strategies are implemented in CloudSim and the workload is distributed accordingly. The utilization threshold is used to reserve resources for the processes (session initiation/termination, voice processing etc.) that handle VoIP calls in order to ensure availability, the most relevant QoS metric. The performance of the strategies with 70% utilization threshold for SNs is evaluated. In Figure 5.8 it is shown the daily number of billing hours required by each strategy to handle the incoming traffic. The allocation strategies have a similar behavior when the incoming traffic is low, namely during weekends (e.g. Day 10 - Day 11 inclusive). During weekdays, there is a visible difference in the quality of solutions given by the distribution strategies. The number of billing hours computed by *BFit* and *FFit* is 55 in comparison with *Rand, RR, WFit* that use 70 hours of billing hours for the allocation of the traffic.

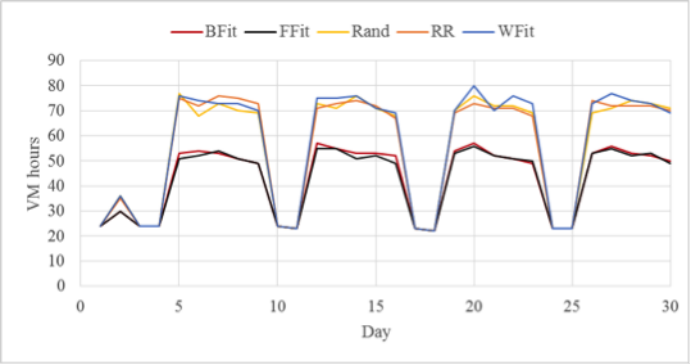


Figure 5.8: Number of billing hours during 30 days.

Throughout a day, the allocation strategies have a similar behavior during time frames with low incoming traffic. In Figure 5.9 the number of SNs necessary to process the VoIP calls placed during a given day is exemplified. As expected, the highest number of SNs required to handle calls is during peak hours with a drop for the lunch break.

In Figure 5.10 the average number of billing hours during 30 days is shown. *FFit* and *BFit* give the best solution with the lowest number of billing hours. They use 42.7 and 43.2 billing hours respectively in average. *Rand* and *RR* use 55.9 and 56.1 billing hours while the highest number of billing hours is required for *WFit* with 57.1 billing hours per day in average. The approximated monthly difference between *FFit*, *BFit* and *Rand*, *RR*, *WFit* is 13 hours per day.

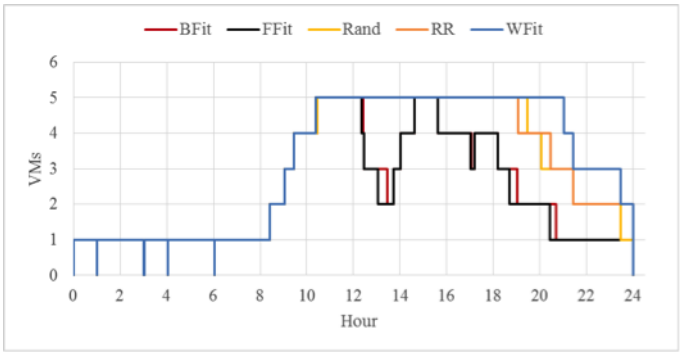


Figure 5.9: Number of billing hours throughout a day.

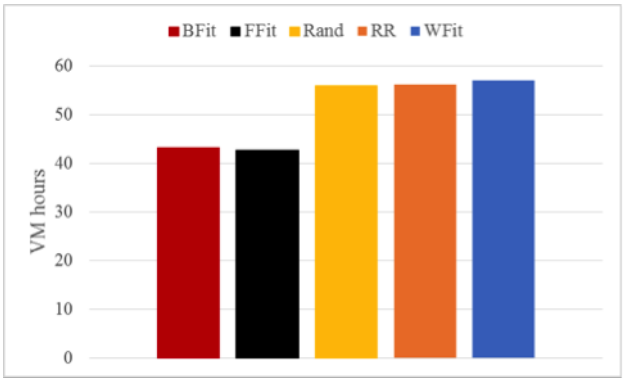


Figure 5.10: Average billing hours per day.

Bi-Objective Approximation. Minimization of the total cost of rented VMs and minimization of their utilization are two conflicting objectives, each objective corresponding to a different optimal solution. When a VSP that uses IaaS as a solution is willing to sacrifice cost (pay more), the VSP can use as call allocation strategy the one that returns as solution a higher number of VMs with lesser utilization. The extent of sacrifice in cost (higher cost) is related to the gain in reduced utilization. In the above-mentioned decision-making problem of renting the proper number of VMs to handle the incoming voice traffic, a set of compromise solutions that represent an approximation of the Pareto front are obtained and the trade-off solutions are illustrated and analyzed. For an easier representation, the problem of minimizing the two criteria is renamed to: *degradation of cost* and *degradation of utilization*.

In Figure 5.11 the set of optimal solutions can be visualized. It is shown a set of solutions approximating the Pareto front, for each of the five call allocation strategies: *Rand, RR, FFit, BFit, WFit*. 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. FFit solutions cover cost degradations from 0 to 0.058, whereas BFit solutions are in the

range from 0 to 0.057. The set coverage method is used to analyze the performance of the scheduling strategies. The results are shown in Table 5.6 and Table 5.7.

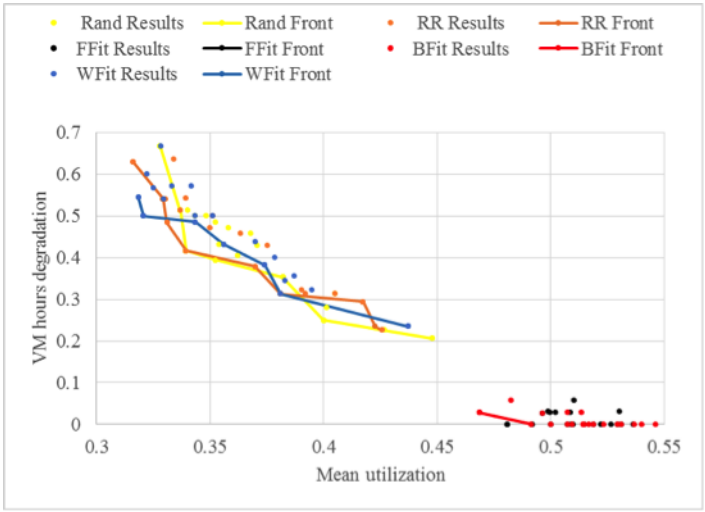


Figure 5.11: Pareto front.

$SC(A, B)$ calculates 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.

SC (A, B)	FFit	BFit	Mean	Rank
FFit	1.0	0.50	0.75	1
BFit	0.0	1.0	0.50	2
Mean	0.50	0.75		
Rank	1	2		

Table 5.6: Set coverage and ranking FFit and Bfit.

$SC(FFit, B)$ 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.

SC (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	2
WFit	0.444	0.111	1.0	0.518	1
Mean	0.703	0.407	0.714		
Rank	2	1	3		

Table 5.7: Set coverage and ranking, Rand, RR, and WFit.

5.3 Summary

In this chapter the model for job allocation problem addressing VoIP in cloud computing is formulated. Models for VoIP load balancing are defined. The last advances of load balancing in distributed computer environments, to understand the main characteristics and requirements of load balancing algorithms, are overviewed. It is also shown that none of these works directly addresses the problem space of the considered problem, but do provide a valuable basis for the current study.

In cloud computing, load balancing bounds can be dynamically adjusted to cope with the dynamic workload situation. To this end, the past workload must be analyzed for a certain time interval to determine appropriate lower and upper bounds. The time interval should be set according to workload characteristics, communication delays, and cloud configurations. There are defined models for the provider cost, quality of service, and are proposed new allocation bin packing algorithms for VoIP super node clusters. It is suitable for an environment with presence of uncertainty, and take into account QoS and cost optimization. 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 is used for a VoIP cloud environment and it is simulated using real VoIP traces and corresponding VoIP cloud configurations.

Chapter 6

Conclusions and Perspectives

This chapter is a summary of the overall work, issues that were addressed in the thesis and moreover, a personal perspective towards the implications of the research project with respect to the chosen axes. This thesis was a real challenge with a fine line that always had to shift between the research objectives and the industry solutions constraints. The real challenge was to consider all the different aspects, theoretical and practical, not only focusing on a solution for the initial project proposal but also to adapt to the ICT changes that took place in the last years. Large collaboration efforts were undertaken as well in the context of the Supernode2 project from the Ministry of Economy.

6.1 Conclusions

DYMO (Dynamic MixVoIP) project targeted the implementation of an algorithm capable to improve the VoIP quality of the services provided by a VSP via a cloud-based solution and to lower the operational costs at infrastructure level. Since 2012, there was a collaboration with MIXvoip, a Luxembourgian company that offers services locally and in the surroundings. In the incipient phase of the project the adopted solution though developed in cloud, it was not designed for such environments. The core of the project was to find an alternative solution to overcapacity and replace it with one that can take decisions based on the evolution of requests and to adapt the system dynamically in order to lower the operational costs at infrastructure level. Therefore, the first phase of the project targeted the prediction of the load of the traffic in the system.

Firstly, the incoming traffic in the system was analyzed and used to built a prediction tool for the traffic load. The requests of the customers are modeled using a multi-Gaussian probability distribution. Several algorithms capable of predicting the load of a processor in response to incoming voice traffic are provided, with focus on particle simulation algorithms. A

detailed bibliography on particle algorithms with theoretical and numerical advances in the field is provided. A first analysis of the proposed algorithms was conduced on synthetic problems by following a rare event simulation perspective, law of a stochastic process where particles cross a specific critical threshold. A numeric simulation for particle algorithms was developed, that includes different benchmarks and execution analysis test cases, with a large number of particles and simulation levels, different selection strategies, adaptive selections of levels. The prediction tool was extended by introducing new predictors for the VoIP traffic, namely model-based learning of mixture of Gaussians (Gaussian Mixture Model (GMM)) and supervised learning that defines distributions over functions (Gaussian Process (GP)).

The models learn from the data which follows the Gaussian distribution. Considering the errors, the resulted comparison between the predictors, how fast the algorithm converge, the GP gave best quality solutions. However, there are other reasons to use machine learning for prediction of VoIP traffic. IPS was chosen for the evolutionary side and the full control over the representation of the particles. GP is represented by the covariance function and the mean function, and it is interesting to observe the continuous change of the solution representation given by the variation of the parameters. GMMs is one of the most mature methods for clustering, deployed for engineering applications and used for density estimations of data represented by underlying heterogeneous populations can be explained by a normal distribution.

One important aspect of the thesis was to gain knowledge on existing VoIP implementations and technology. It is important to understand how VoIP works, what type of modules one system consists of, what are the challenges, in order to mathematically model the load balancing mechanisms for cloud based VoIP. A better perspective together with a bigger picture of the behavior of VoIP systems were achieved during trainings at the company and personal time investment.

The development of a test environment, a simulator of a real VoIP telephone system was an important aspect of the thesis. The work was continued using open source tools to stress the servers and check the impact of the traffic on the resources in the system. The opportunity of using high quality VoIP equipment from KMUTT Lab, Bangkok, to run specific tests on the simulator environment for the further validation of the performance of the proposed solutions, was investigated. A large number of papers that describe VoIP performance measurements and practices, while considering the current commercial solutions in use, are reviewed and presented. As direct conclusions related to the technology, it is recommended to consider the current IT infrastructure in order to adopt VoIP as a solution and that VSP have a good understanding of the behavior of VoIP traffic users, model the future requests in order to prepare for scalability. VoIP is very sensitive to network changes and from my point of view, the most important QoS aspect for a VSP is to ensure availability. There are many tools and there

is much information on how to develop the system, monitor it, scale up or down the resources using cloud as a solution. In VoIP, the quality of the solutions provided can be quantified as the number of satisfied clients.

Load balancing is one axes of research addressed in the thesis and it implies mapping jobs, virtual machines in this context, to physical computational resources. Usually, the simplest way of handling this problem is by using approaches like sender-initiator, where the machine that is heavily loaded expels some of its load to some other machine. In turn the machine that receives the request can accept or not the demand given that the extra charge may imply overshooting local critical load levels. Depending on the nature of the environment, two types of load balancing approaches can be identified: static and dynamic. In the static case, once an assignment solution is applied, no changes can be made, i.e. moving tasks to a different machine than the already assigned ones. In the dynamic case, two subclasses can be distinguished: preemptive and non-preemptive assignment. In the first case, tasks can have their assignment changed at execution time while in the non-preemptive case, once a job is started on a machine, that job must be completed on the same machine. Another important aspect that was considered is the nature of the environment and of the jobs that need to be assigned (deterministic or stochastic). For virtual machines, as the execution time depends on the end-user, the total required execution time is generally not known in advance, where from the stochastic nature of the problem.

Load balancing formulations for cloud environments suppose different perspectives according to the goal followed and the available data. Classical approaches include (i) scheduling problems, seen as job-shop scheduling, multiprocessor scheduling, grid task scheduling, with objectives that consider, among others, makespan minimization (deterministic jobs) generally using evolutionary algorithms or (ii) flow problems (multicast flows) minimizing flow time, i.e. total completion time of all tasks, communication costs needed in transferring the jobs between resources, communication or response time (the quantity of information to be exchanged, communication costs and delays). Note that other objectives co-exist as flow assignation and packet loss. For the current case it is adopted a load balancing perspective that takes into account the usage of resources and energy consumption.

Furthermore, the thesis contributes to extending ones understanding of dynamic environments and their modeling, how concepts relate to time, anticipation and short or long term impact can be defined in order to capture coherent traits of a real-life large scale system. In addition, as a second major contribution of the study, highly efficient load-balancing algorithms are developed, able to deal with constraints and performance measures not addressed to such an extent and in such an integrative manner before in the literature. The obtained results further offer the basis for identifying algorithms that can be used at different levels inside a cloud environment, e.g.

as part of a decentralized solution where the evolution of the entire system is driven by an emergent behavior of how individual low-level actions control the efficient allocation of resources, energy-efficiency or direct towards specific performance indicators and objectives to attain.

6.2 Perspectives

In this section research questions that were encountered and interesting literature that could have been followed up, are presented. It has been shown that the prediction models learn from the data which follows the Gaussian distribution. For the reasons already listed, Gaussian Mixture Model (GMM) (model-based learning of mixture of Gaussian) was used for the extension of the prediction tool developed to anticipate the incoming VoIP traffic. One might find an interesting approach the extension of the GMM to a model that does not necessarily a priori limits the number of components, namely *Infinite Gaussian Mixture Model*[137]. The difficultly of using an infinite Gaussian mixture consists of approximating the proper number of mixture components that fits best the model. Predicting the number of clusters adds complexity to the computation of the model parameters. However it is considered as being an interesting technique to focus on, as future work.

In the thesis the GMMs are fitted onto the data using the EM algorithm to estimate the parameters of the components. The algorithm is described and it is used for the maximization of the likelihood of the observed data[138]. One of the major limitations of this algorithm is that the data likelihood is not convex in the model parameters and is impacted in the algorithm that gets stuck in the local optima when focusing on irrelevant correlations in the data. One may consider as an interesting problem, the study of diverse feature design to initialize EM to find a global optima. Furtherer tuning may be performed at the level of the GMM aiming for features simplification and improvement of performance [1].

Another aspect that drew attention during the development of the prediction tools was the distribution of the errors. The prediction is applied on multiple scenarios, using different techniques. The errors are calculated and statistical tests are computed to test the initial assumptions and to be able to conclude over the solution quality given by each. One might find interesting to calculate the distribution of the errors more than the value itself, since the total MAPE value might be influenced by many small error values or at some point the algorithms become instable (e.g. GMM with EM) and gives unexpected out of range errors.

[1]Discussion opened by Prof. Dr Henri Luchian, Alexandru Ioan Cuza University of Iasi, September 2016.

Interactive Particle System (IPS) is a predictor with particles that encode the mean vector and covariance matrix to be perturbed, estimated and improved in order for the model to fit the data. This approach was developed considering the distribution of the data. It would be interesting the test a different technique were the particles encode a different type of distribution, e.g. half-normal distribution [139], generalized inverse distribution [140], Laplace distribution [141]. The beauty of the IPS is, as previously mentioned, the full control over the representation of particles. Thus, it is possible to develop an IPS with particles that encode the number of components for a GMM that fits best the data. It is also possible to encode inside a particle a wide number of Gaussian Process (GP) covariance functions and use it to compute the most appropriate for a data set.

There are many products designed for dynamic cloud computing. One might find interesting to adopt and customize *Elastic IP Addresses* for VoIP and to balance the traffic between geographically distributed areas, since the solution provided is the assignation of an Elastic IP to a VM that operates within a specific availability zone [2].

The optimal spread of the traffic on the distributed processors and servers in the cloud was investigated and site load distribution methods are presented. A model built using an ILP approach was used as an input for the dynamic bin packing with open bins problem. The objective of the method is to minimize the number the number of servers used for call assignation while considering prediction. It would be interesting to implement it at the DNS level, compare the performance of the technique with the other well-known and widely used load distribution methods.

There are many studies on dynamically load-balancing tasks and one that drew attention in particular is the self-organized criticality model called sandpile developed for tasks that arrive in the system in the form of Bag-of-Tasks[142] [3]. It is a decentralized agent system that handles the system in a critical state at the edge of chaos. When new tasks are assigned to resources it either has no impact or generate avalanches that reconfigure the state of the system. Different types of avalanches occur that are handled in order to maximize the performance of the system regardless of the task features. It is an interesting opportunity to design the sandpiler for cloud-based VoIP systems and even further, to create the avalanches by adding the prediction on top of the scheduler with direct objective of minimizing the total cost of the infrastructure.

The axes studied in the scope of the thesis can either be grouped to solve a similar problem either decentralized and applied for smaller sets of problems. The variety of possibilities and opportunities shows the endless

[2] `http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html`

[3] For example, sandpile for SIP servers.

applications of load prediction and load balancing in ICT, the complexity of VoIP systems developed in Cloud Computing (CC), the diversity of such infrastructures and multitude of solutions that can be adopted.

Publications

1. A.-M. Simionovici, A.-A. Tantar, and P. Bouvry, "Dynamic MixVoIP", EVOLVE, A Bridge between Probability, Set Oriented Optimization, and Evolutionary Computation, July 2013.

2. A.-M. Simionovici, A.-A. Tantar, P. Bouvry, and L. Didelot, "Predictive modeling in a voip system," Journal of Telecommunications and Information Technology, 2013.

3. A. Simionovici, A. Tantar, P. Bouvry, A. Tchernykh, J. M. Cortes Mendoza, and L. Didelot, "VoIP traffic modeling using Gaussian Mixture models, Gaussian Processes and Interactive Particle algorithms", in 2015 IEEE Globecom , San Diego, CA, USA, December 6-10.

4. J. M. Cortes-Mendoza, A. Tchernykh, A. Drozdov, P. Bouvry, A.-M. Simionovici, D. Kliazovich, and A. Avetisyan, "Distributed adaptive voip load balancing in hybrid clouds," in Russian Supercomputing Days, Moscow Russia, September 2015.

5. J. M. Cortes-Mendoza, A. Tchernykh, A.-M. Simionovici, P. Bouvry, S. Nesmachnow, B. Dorronsoro, and L. Didelot, "VoIP service model for multi-objective scheduling in cloud infrastructure," Int. Journal of Metaheuristics, vol. 4, no. 2, pp. 185 - 203, Jan. 2015.

Bibliography

[1] J. Postel, "Internet Protocol," USC/Information Sciences Institute, Tech. Rep. RFC 760, January 1980. [Online]. Available: http://www.ietf.org/rfc/rfc760.txt

[2] A.-M. Simionovici, A.-A. Tantar, and P. Bouvry, "Dynamic MixVoIP," July 2013.

[3] A.-M. Simionovici, A.-A. Tantar, P. Bouvry, and L. Didelot, "Predictive modeling in a voip system," *Journal of Telecommunications and Information Technology*, 2013. [Online]. Available: http://www.itl.waw.pl/czasopisma/JTIT/2013/4/32.pdf

[4] A. Simionovici, A. Tantar, P. Bouvry, A. Tchernykh, J. M. Cortés-Mendoza, and L. Didelot, "Voip traffic modelling using gaussian mixture models, gaussian processes and interactive particle algorithms," in *2015 IEEE Globecom Workshops, San Diego, CA, USA, December 6-10, 2015*, 2015, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/GLOCOMW.2015.7414113

[5] J. M. Cortés-Mendoza, A. Tchernykh, A. Drozdov, P. Bouvry, A.-M. Simionovici, D. Kliazovich, and A. Avetisyan, "Distributed adaptive voip load balancing in hybrid clouds," 2015. [Online]. Available: http://ceur-ws.org/Vol-1482/676.pdf

[6] J. M. Cortés-Mendoza, A. Tchernykh, A.-M. Simionovici, P. Bouvry, S. Nesmachnow, B. Dorronsoro, and L. Didelot, "Voip service model for multi-objective scheduling in cloud infrastructure," *Int. J. Metaheuristics*, vol. 4, no. 2, pp. 185–203, Jan. 2015. [Online]. Available: http://dx.doi.org/10.1504/IJMHEUR.2015.074251

[7] D. R. Kuhn, T. J. Walsh, and S. Fries, *NIST Special Publication 800-58: Security Considerations For Voice Over IP Systems*. National Institute of Standards and Technology, January 2005. [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-58/SP800-58-final.pdf

[8] "[online] CISCO, Unified Communications System- What is VoIP (Voice-over-IP)?" http://www.cisco.com/c/en/us/products/unified-communications/networking_solutions_products_genericcontent0900aecd804f00ce.html.

[9] L. Goleniewski, *Telecommunications Essentials: The Complete Global Source for Communications Fundamentals, Data Networking and the Internet, and Next-Generation Networks.* Addison-Wesley Professional, December 2001, vol. 1.

[10] P. Sherburne and C. Fitzgerald, "You don't know jack about voip," *Queue*, vol. 2, no. 6, pp. 30–38, Sep. 2004. [Online]. Available: http://doi.acm.org/10.1145/1028893.1028895

[11] S. R. Ahuja and R. Ensor, "Voip: What is it good for?" *Queue*, vol. 2, no. 6, pp. 48–55, Sep. 2004. [Online]. Available: http://doi.acm.org/10.1145/1028893.1028897

[12] L. Madsen, J. V. Meggelen, and R. Bryant, *Asterisk: The Definitive Guide.* 1005 Gravenstein Highway North, Sebastopol, United States of America: O'Reilly Media, April 2011, vol. 3rd.

[13] T. J. Walsh and D. R. Kuhn, "Challenges in securing voice over ip," *IEEE Security Privacy*, vol. 3, no. 3, pp. 44–49, May 2005.

[14] "Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model," ISO/IEC JTC 1, Information technology, in collaboration with ITU-T., International Organization for Standardization, Tech. Rep. ISO/IEC 7498-1, June 1996. [Online]. Available: http://www.ecma-international.org/activities/Communications/TG11/s020269e.pdf

[15] T. Wallingford, *Switching to VoIP.* O'Reilly Media, June 2009. [Online]. Available: http://www.cosmocom.gr/wp-content/uploads/2013/05/Switching-to-VoIP.pdf

[16] J. Postel, "User Datagram Protocol," USC/Information Sciences Institute, Tech. Rep. RFC 768, August 1980. [Online]. Available: http://www.ietf.org/rfc/rfc768.txt

[17] ——, "Transmission Control Protocol," USC/Information Sciences Institute, Tech. Rep. RFC 761, January 1980. [Online]. Available: http://www.ietf.org/rfc/rfc761.txt

[18] M. Handley, V. Jacobson, and C. Perkins, "SDP: Session Description Protocol," The Internet Society, Tech. Rep. RFC 2327, July 2006. [Online]. Available: https://tools.ietf.org/html/rfc4566

[19] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," Internet Official Protocol Standards, The Internet Society, Tech. Rep. RFC 3261, June 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt

[20] "ITU-T Recommendations H.323: Packet-based multimedia communications systems," International Telecommunication Union, Tech. Rep., December 2009. [Online]. Available: http://www.itu.int/rec/T-REC-H.323/en

[21] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," Internet Official Protocol Standards, The Internet Society, Tech. Rep. RFC 3550, July 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3550.txt

[22] F. Andreasen and B. Foster, "Media Gateway Control Protocol (MGCP) Version 1.0," The Internet Society, Tech. Rep. RFC 3435, January 2003. [Online]. Available: https://tools.ietf.org/html/rfc3435

[23] "Methods for objective and subjective assessment of speech quality: Mean opinion score interpretation and reporting," ITU-T International Telecommunication Union, Tech. Rep., May 2013. [Online]. Available: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-P.800.2-201305-I!!PDF-E&type=items

[24] "ITU-T Recommandation G.711," ITU-T International Telecommunication Union, Tech. Rep., November 1988. [Online]. Available: http://www.itu.int/rec/T-REC-G.711-198811-I/en

[25] "ITU-T Recommandation G.729," ITU-T International Telecommunication Union, Tech. Rep., June 2012. [Online]. Available: http://www.itu.int/rec/T-REC-G.729-201206-I/en

[26] "ITU-T Recommandation G.723.1," ITU-T International Telecommunication Union, Tech. Rep., May 2006. [Online]. Available: http://www.itu.int/rec/T-REC-G.723.1-200605-I/en

[27] "ITU-T Recommandation G.726," ITU-T International Telecommunication Union, Tech. Rep., December 1990. [Online]. Available: http://www.itu.int/rec/T-REC-G.726-199012-I/en

[28] "ITU-T Recommandation G.728," ITU-T International Telecommunication Union, Tech. Rep., June 2012. [Online]. Available: http://www.itu.int/rec/T-REC-G.728-201206-I/en

[29] S. Andersen, A. Duric, H. Astrom, R. Hagen, W. Kleijn, and J. Linden, "Internet Low Bit Rate Codec (iLBC)," The Internet Society, Tech. Rep., December 2004. [Online]. Available: https://tools.ietf.org/html/rfc3951

[30] U. C. S. CISCO, "Understanding codecs: Complexity, hardware support, mos, and negotiation." [Online]. Available: http://www.cisco.com/c/en/us/support/docs/voice/h323/14069-codec-complexity.html

[31] T. Christiansen, I. Giotis, and S. Mathur, "Performance evaluation of voip in different settings." [Online]. Available: https://courses.cs.washington.edu/courses/cse561/04au/projects/papers/Mathur-Giotis-Christiansen.pdf

[32] G. Prabhakar, R. Rastogi, and M. Thottan, "Oss architecture and requirements for voip networks," *Bell Labs Technical Journal*, vol. 10, no. 1, pp. 31–45, Spring 2005.

[33] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," The Internet Society, Tech. Rep., January 2001. [Online]. Available: https://www.ietf.org/rfc/rfc3031.txt

[34] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," The Internet Society, Tech. Rep., December 1998. [Online]. Available: https://tools.ietf.org/html/rfc2474

[35] S. software company, "Performance and stress testing of sip servers, clients and ip networks." [Online]. Available: http://startrinity.com/VoIP/TestingSipPbxSoftswitchServer.aspx

[36] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the clouds: A berkeley view of cloud computing," *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, vol. 28, no. 13, 2009. [Online]. Available: https://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf

[37] "[online] ISO/IEC JTC 1/SC 38 Cloud Computing and Distributed Platforms," http://www.iso.org/iso/iso_technical_committee.html?commid=601355.

[38] "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, Tech. Rep. Special Publication 800-145, September 2011. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[39] D. F. Parkhill, *The Challenge of the Computer Utility.* Addison-Wesley, 1966.

[40] V. Stantchev and C. Schröpfer, "Negotiating and enforcing qos and slas in grid and cloud computing," in *Advances in grid and pervasive computing.* Springer, 2009, pp. 25–35. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.474.6758&rep=rep1&type=pdf

[41] M. Guzek, "Holistic, autonomic, and energy-aware resource allocation in cloud computing," Ph.D. dissertation, University of Luxembourg, Luxembourg, 2014. [Online]. Available: http://hdl.handle.net/10993/19946

[42] "[online] Amazon Elastic Compute Cloud (Amazon EC2)," https://aws.amazon.com/ec2/.

[43] "[online] Google Cloud Platform," https://cloud.google.com/.

[44] A. Ganapathi, Y. Chen, A. Fox, R. Katz, and D. Patterson, "Statistics-driven workload modeling for the cloud," in *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on*, March 2010, pp. 87–92.

[45] S. Kim, J. I. Koh, Y. Kim, and C. Kim, "A science cloud resource provisioning model using statistical analysis of job history," in *Dependable, Autonomic and Secure Computing (DASC), 2011 IEEE Ninth International Conference on*, Dec 2011, pp. 792–793.

[46] T. Lynn, N. OCarroll, J. Mooney, M. Helfert, D. Corcoran, G. Hunt, L. Van Der Werff, J. Morrison, and P. Healy, "Towards a framework for defining and categorising business process-as-a-service (bpaas)," in *Proceedings of the 21st International Product Development Management Conference*, 2014. [Online]. Available: https://www.researchgate.net/publication/263505922_Towards_a_Framework_for_Defining_and_Categorising_Business_Process-As-A-Service_BPaaS

[47] K. Nagothu, B. Kelley, J. Prevost, and M. Jamshidi, "On prediction to dynamically assign heterogeneous microprocessors to the minimum joint power state to achieve ultra low power cloud computing," in *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on.* IEEE, 2010, pp. 1269–1273. [Online]. Available: http://engineering.utsa.edu/~bkelley/Pdf/Asilomar%20Final%20version%20paper.pdf

[48] "[online] MIXvoip Smart Business Telephony," https://www.mixvoip.com/.

[49] J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, "An overview of machine learning," in *Machine learning*. Springer, 1983, pp. 3–23.

[50] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002. [Online]. Available: http://doi.acm.org/10.1145/505282.505283

[51] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: machine learning and application to spontaneous behavior," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 568–573.

[52] C. Sinclair, L. Pierce, and S. Matzner, "An application of machine learning to network intrusion detection," in *Computer Security Applications Conference, 1999. (ACSAC '99) Proceedings. 15th Annual*, 1999, pp. 371–377.

[53] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05)l*, Nov 2005, pp. 250–257.

[54] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[55] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*. CRC Press, September 1987.

[56] C. Ash, *Continuous Random Variables*. Wiley-IEEE Press, 1993, pp. 97–170. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5265265

[57] W. Davenport and W. Root, *The Gaussian Process*. Wiley-IEEE Press, 1987.

[58] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[59] I. Aleksander and H. Morton, *An introduction to neural computing*. Chapman and Hall London, 1990, vol. 240.

[60] N. Cristianini and J. Shawe-Taylor, "An introduction to support vector machines," 2000.

[61] T. Joachims, *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[62] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 263–270.

[63] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied linear statistical models*. Irwin Chicago, 1996, vol. 4.

[64] H. L. Seal, "Studies in the history of probability and statistics. xv: The historical development of the gauss linear model," *Biometrika*, vol. 54, no. 1/2, pp. 1–24, 1967. [Online]. Available: http://www.jstor.org/stable/2333849

[65] F. Cérou and A. Guyader, "Adaptive multilevel splitting for rare event analysis," *Stochastic Analysis and Applications*, vol. 25, no. 2, pp. 417–443, 2007. [Online]. Available: https://hal.inria.fr/inria-00070307/document

[66] P. Del Moral, *Feynman-Kac Formulae*. Springer, 2004. [Online]. Available: http://www.springer.com/us/book/9780387202686

[67] A. M. Johansen, P. Del Moral, and A. Doucet, "Sequential monte carlo samplers for rare events," in *University of Cambridge, Department OF Engineering*. Citeseer, 2005. [Online]. Available: http://web.maths.unsw.edu.au/~peterdel-moral/JDD06c.pdf

[68] P. Del Moral, A. Doucet, and A. Jasra, "Sequential monte carlo samplers," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 3, pp. 411–436, 2006. [Online]. Available: http://www.stats.ox.ac.uk/~doucet/delmoral_doucet_jasra_sequentialmontecarlosamplersJRSSB.pdf

[69] P. Del Moral and P. Lezaud, "Branching and interacting particle interpretations of rare event probabilities," in *Stochastic hybrid systems*. Springer, 2006, pp. 277–323. [Online]. Available: http://link.springer.com/chapter/10.1007%2F11587392_9

[70] Z. I. Botev and D. P. Kroese, "Efficient monte carlo simulation via the generalized splitting method," *Statistics and Computing*, vol. 22, no. 1, pp. 1–16, 2012. [Online]. Available: http://link.springer.com/content/pdf/10.1007%2Fs11222-010-9201-4.pdf

[71] S. W. Nydick, "The wishart and inverse wishart distributions," 2012. [Online]. Available: www.math.wustl.edu/~sawyer/hmhandouts/Wishart.pdf

[72] P. Del Moral and A. Doucet, "Particle methods: An introduction with applications," INRIA, Research Report RR-6991, 2009. [Online]. Available: https://hal.inria.fr/inria-00403917

[73] L. Ding and R. A. Goubran, "Speech quality prediction in voip using the extended e-model," in *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, vol. 7, Dec 2003, pp. 3974–3978 vol.7.

[74] A. Raake, *Speech quality of VoIP: assessment and prediction.* John Wiley & Sons, 2007.

[75] M. Al-Akhras, H. Zedan, R. John, and I. ALMomani, "Non-intrusive speech quality prediction in voip networks using a neural network approach," *Neurocomputing*, vol. 72, no. 10, pp. 2595–2608, 2009.

[76] L. Sun and E. C. Ifeachor, "Voice quality prediction models and their application in voip networks," *Multimedia, IEEE Transactions on*, vol. 8, no. 4, pp. 809–820, 2006.

[77] R. Estepa, A. Estepa, and J. Vozmediano, "Accurate prediction of voip traffic mean bit rate," *Electronics Letters*, vol. 41, no. 17, pp. 985–987, 2005. [Online]. Available: http://www.researchgate.net/publication/3388230_Accurate_prediction_of_VoIP_traffic_mean_bit_rate

[78] M. Mandjes, I. Saniee, and A. Stolyar, "Load characterization and anomaly detection for voice over ip traffic." *IEEE transactions on neural networks/a publication of the IEEE Neural Networks Council*, vol. 16, no. 5, pp. 1019–1026, 2005. [Online]. Available: http://ect.bell-labs.com/who/iis/publications/papers/01510706.pdf

[79] Q. Dan, Y. Honggang, T. Hui, and W. Bingxi, "Two schemes for automatic speaker recognition over voip," in *Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on*, vol. 2. IEEE, 2008, pp. 695–699.

[80] D. Yessad and A. Amrouche, "Fusion strategies for distributed speaker recognition using residual signal based g729 resynthesized speech," in *Information Fusion (FUSION), 2013 16th International Conference on.* IEEE, 2013, pp. 432–437.

[81] O. Nhway, "An investigation into the effect of security on reliability and voice recognition system in a voip network," in *Advanced Communication Technology (ICACT), 2011 13th International Conference on.* IEEE, 2011, pp. 1293–1297.

[82] T. D. Dang, B. Sonkoly, and S. Molnár, "Fractal analysis and modeling of voip traffic," in *Telecommunications Network Strategy and Planning Symposium. NETWORKS 2004, 11th International.* IEEE, 2004, pp. 123–130. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.8103&rep=rep1&type=pdf

[83] N. J. Kansal and I. Chana, "Cloud load balancing techniques: A step towards green computing," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 238–246, 2012.

[84] C. Kopparapu, *Load balancing servers, firewalls, and caches.* John Wiley & Sons, 2002.

[85] A. M. Alakeel, "A guide to dynamic load balancing in distributed computer systems," *International Journal of Computer Science and Information Security*, vol. 10, no. 6, pp. 153–160, 2010.

[86] K. Joanna and S. U. Khan, "Multi-level hierarchic genetic-based scheduling of independent jobs in dynamic heterogeneous grid environment," pp. 1–19, 2012.

[87] A. Tchernykh, J. M. Cortés-Mendoza, J. E. Pecero, P. Bouvry, and D. Kliazovich, "Adaptive energy efficient distributed voip load balancing in federated cloud infrastructure," in *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on.* IEEE, 2014, pp. 27–32.

[88] A. Singh, D. Juneja, and M. Malhotra, "Autonomous agent based load balancing algorithm in cloud computing," *Procedia Computer Science*, vol. 45, pp. 832–841, 2015.

[89] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing," in *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on.* IEEE, 2010, pp. 551–556.

[90] Y. Zhao and W. Huang, "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud," in *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on.* IEEE, 2009, pp. 170–175.

[91] S. N. C. S. K. R., "A fuzzy-based firefly algorithm for dynamic load balancing in cloud computing environment," *Journal of Emerging Technologies in Web Intelligence*, pp. 435–440, 2014.

[92] K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal, and S. Dam, "A genetic algorithm (ga) based load balancing strategy for cloud computing," *Procedia Technology*, vol. 10, pp. 340–347, 2013.

[93] P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Applied Soft Computing*, vol. 13, no. 5, pp. 2292–2303, 2013.

[94] Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. Greenberg, "Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services," *Performance Evaluation*, vol. 68, no. 11, pp. 1056–1071, 2011.

[95] A. P. Florence and V. Shanthi, "A load balancing model using firefly algorithm in cloud computing," *Journal of Computer Science*, vol. 10, no. 7, p. 1156, 2014.

[96] J. M. Galloway, K. L. Smith, and S. S. Vrbsky, "Power aware load balancing for cloud computing," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2011, pp. 19–21.

[97] J. Adhikari and S. Patil, "Double threshold energy aware load balancing in cloud computing," in *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*. IEEE, 2013, pp. 1–6.

[98] J. J. Laredo, B. Dorronsoro, J. Pecero, P. Bouvry, J. J. Durillo, and C. Fernandes, "Designing a self-organized approach for scheduling bag-of-tasks," in *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2012 Seventh International Conference on*. IEEE, 2012, pp. 315–320.

[99] Y. Fang, F. Wang, and J. Ge, "A task scheduling algorithm based on load balancing in cloud computing," in *International Conference on Web Information Systems and Mining*. Springer, 2010, pp. 271–277.

[100] F. Ramezani, J. Lu, and F. K. Hussain, "Task-based system load balancing in cloud computing using particle swarm optimization," *International Journal of Parallel Programming*, vol. 42, no. 5, pp. 739–754, 2014.

[101] J. Ni, Y. Huang, Z. Luan, J. Zhang, and D. Qian, "Virtual machine mapping policy based on load balancing in private cloud environment," in *Cloud and Service Computing (CSC), 2011 International Conference on*. IEEE, 2011, pp. 292–295.

[102] "[online]redhat: SIP Load Balancing Basics," http://access.redhat.com/documentation/en-US/JBoss_Communications_Platform/5.1/html/SIP_Load_Balancer_User_Guide/chap-Load_Balancing.html#sslb-SIP_Load_Balancing_Basics.

[103] "[online] Mobicents: Introduction," http://www.mobicents.org/incubator/sip-balancer/intro.html.

[104] "[online] Brocade ServerIron ADX Advance Server Load Balancing Guide: SIP Server Load Balancing," http://www.brocade.com/en/backend-content/pdf-page.html?/content/dam/common/documents/content-types/configuration-guide/serveriron-12502-advancedslbguide.pdf.

[105] "[online] loadbalancer.org : Load Balancing Skype for Business," http://loadbalancer.org/ca/applications/microsoft-apps.

[106] "[online] radware : VoIP Load Balancing," https://www.radware.com/Resources/voip_load_balancing.aspx.

[107] "[online] OpenSIPS : Load Balancing in OpenSIPS," https://www.opensips.org/Documentation/Tutorials-LoadBalancing.

[108] G. Kambourakis, D. Geneiatakis, T. Dagiuklas, C. Lambrinoudakis, and S. Gritzalis, "Towards effective sip load balancing," in *Proceedings of the 3rd Annual VoIP Security Workshop*, 2006. [Online]. Available: http://www.cs.columbia.edu/~dgen/papers/conferences/conference-04.pdf

[109] P. Del Moral, A.-A. Tantar, and E. Tantar, "On the foundations and the applications of evolutionary computing," in *EVOLVE-A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation*. Springer, 2013, pp. 3–89. [Online]. Available: http://link.springer.com/content/pdf/10.1007%2F978-3-642-32726-1.pdf

[110] M. Breaban and H. Luchian, "A unifying criterion for unsupervised clustering and feature selection," *Pattern Recognition*, vol. 44, no. 4, pp. 854–865, 2011.

[111] B. Scherrer, "Gaussian mixture model classifiers," 2007. [Online]. Available: http://www.medialab.bme.hu/medialabAdmin/uploads/VITMM225/GMMScherrer07.pdf

[112] M. I. Ribeiro, "Gaussian probability density functions: Properties and error characterization," *Institute for Systems and Robotics, Lisboa, Portugal*, 2004. [Online]. Available: http://users.isr.ist.utl.pt/~mir/pub/probability.pdf

[113] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977. [Online]. Available: http://www.stat.missouri.edu/~dsun/9720/EM_JRSSB.pdf

[114] C. B. Do, "Gaussian processes," December 2007. [Online]. Available: https://see.stanford.edu/materials/aimlcs229/cs229-gp.pdf

[115] M. Ebden, "Gaussian processes: A quick introduction," August 2008. [Online]. Available: http://www.robots.ox.ac.uk/~mebden/reports/GPtutorial.pdf

[116] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 63–71. [Online]. Available: http://link.springer.com/content/pdf/10.1007%2F978-3-540-28650-9_4.pdf

[117] I. Murray, "Introduction to gaussian processes," 2008. [Online]. Available: http://www.inf.ed.ac.uk/teaching/courses/mlpr/2014/slides/16_gp.pdf

[118] C. E. Rasmussen and C. Williams, "Documentation for gpml matlab code version 3.6." [Online]. Available: http://www.gaussianprocess.org/gpml/code/matlab/doc

[119] R. V. Hogg and J. Ledolter, *Engineering statistics*. Macmillan Pub Co, 1987.

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

[121] J. Emeras, S. Varrette, and P. Bouvry, "Amazon elastic compute cloud (ec2) vs. in-house hpc platform: a cost analysis," in *Proc. of the 9th IEEE Intl. Conf. on Cloud Computing (CLOUD 2016)*. IEEE Computer Society, 2016.

[122] P. Montoro and E. Casilari, "A comparative study of voip standards with asterisk," in *Digital Telecommunications, 2009. ICDT'09. Fourth International Conference on*. IEEE, 2009, pp. 1–6.

[123] L. Georgiou, "3cx phone system and atom n270 processor benchmarking." [Online]. Available: http://www.3cx.com/blog/voip-howto/atom-processor-n270-benchmarking/

[124] L. M. Campos and I. D. Scherson, "Rate of change load balancing in distributed and parallel systems," *Parallel Computing*, vol. 26, no. 9, pp. 1213–1230, 2000.

[125] A. Stathakis, "Satellite payload reconfiguration optimisation," Ph.D. dissertation, University of Luxembourg, Luxembourg, 2014. [Online]. Available: http://hdl.handle.net/10993/18760

[126] B. Dorronsoro, G. Danoy, P. Bouvry, and A. J. Nebro, "Multi-objective cooperative coevolutionary evolutionary algorithms for continuous and combinatorial optimization," in *Intelligent Decision Systems in Large-Scale Distributed Environments*. Springer, 2011, pp. 49–74.

[127] E. Kieffer, A. Stathakis, G. Danoy, P. Bouvry, E. G. Talbi, and G. Morelli, "Multi-objective evolutionary approach for the satellite payload power optimization problem," in *Computational Intelligence in Multi-Criteria Decision-Making (MCDM), 2014 IEEE Symposium on*, Dec 2014, pp. 202–209.

[128] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[129] D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry, S. U. Khan, and A. Y. Zomaya, "Ca-dag: Modeling communication-aware applications for scheduling in cloud computing," *Journal of Grid Computing*, vol. 14, no. 1, pp. 23–39, 2016.

[130] E. G. Coffman Jr, J. Csirik, G. Galambos, S. Martello, and D. Vigo, "Bin packing approximation algorithms: survey and classification," in *Handbook of Combinatorial Optimization*. Springer, 2013, pp. 455–531.

[131] E. G. Coffman Jr, M. R. Garey, and D. S. Johnson, "Approximation algorithms for bin packing: a survey," in *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996, pp. 46–93.

[132] J. Csirik and G. J. Woeginger, "On-line packing and covering problems," in *Online Algorithms*. Springer, 1998, pp. 147–177.

[133] E. G. Coffman, G. Galambos, S. Martello, and D. Vigo, "Bin packing approximation algorithms: Combinatorial analysis," in *Handbook of combinatorial optimization*. Springer, 1999, pp. 151–207.

[134] E. G. Coffman, Jr, M. R. Garey, and D. S. Johnson, "Dynamic bin packing," *SIAM Journal on Computing*, vol. 12, no. 2, pp. 227–258, 1983.

[135] M. Burcea, "Online dynamic bin packing," Ph.D. dissertation, University of Liverpool, 2014.

[136] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[137] C. E. Rasmussen, "The infinite gaussian mixture model." in *NIPS*, vol. 12, 1999, pp. 554–560.

[138] J. V. Graça, K. Ganchev, and B. Taskar, "Expectation maximization and posterior constraints," 2007.

[139] F. Leone, L. Nelson, and R. Nottingham, "The folded normal distribution," *Technometrics*, vol. 3, no. 4, pp. 543–550, 1961.

[140] C. Robert, "Generalized inverse normal distributions," *Statistics & Probability Letters*, vol. 11, no. 1, pp. 37–41, 1991.

[141] S. Kotz, T. Kozubowski, and K. Podgorski, *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance.* Springer Science & Business Media, 2012.

[142] J. L. J. Laredo, P. Bouvry, F. Guinand, B. Dorronsoro, and C. Fernandes, "The sandpile scheduler," *Cluster Computing*, vol. 17, no. 2, pp. 191–204, 2014.