Projeto de Graduação

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

December 14, 2023

Optimization of computational resource allocation for CORE network applications in mobile telecommunications scenarios

Gustavo de Farias Padilla

PUC RIO

www.ele.puc-rio.br

Projeto de Graduação

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Optimization of computational resource allocation for CORE network applications in mobile telecommunications scenarios

Student: Gustavo de Farias Padilla

Advisor: Marco Grivet

Work presented as partial requirement to the conclusion of the Bacharel em Engenharia Elétrica em Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil.

Acknowledgments

First of all, I would like to thank my advisor Prof. Marco Grivet supporting me with this topic. This study would not be possible without his encouragement, patience, guidance, motivation and assistance.

I would like to express my love to my family, for their encouragement in both good and difficult times. To my parents, thank you for your advice, patience and support in my decisions throughout the years. To my mother, Lindaci, thank you for teaching me the importance of working hard and that the achievements that we can reach in life. To my father, Pedro, for teaching me the power of the choices. To my wife, Luiza, for showing me that we sometimes need a break in order not to get completely insane.

Abstract

The use of mobile phone networks has been increasing at a very fast pace, where users seek a better experience in the use of video streaming applications, Internet of Things (IoT) networks, and real-time demands, requiring high transmission rates, low latency, and connectivity to a large number of users. To meet such requirements, the use of an increasing amount of computational resources in CORE servers is necessary.

A significant challenge in this environment is the allocation of resources for activities related to providing services to end-users of a converged 5G CORE network. When resources are allocated with greater efficiency and utilization, it is possible for operators to have more resources that can be invested in more remote locations, improving the user experience in all the most remote corners of the country.

Keywords: Mobile Networks, CORE Network, Mobile Phone Service, 4G Network, 5G Network, NFVI, Resource Allocation, CPU, Computacional Resources, Optimization

Otimização de alocação de recursos computacionais para aplicações de redes CORE em cenários de telecomunicações móveis

Resumo

O uso das redes de telefonia móvel têm aumentado em uma velocidade muito grande, onde usuários buscam uma melhor experiência na utilização de aplicações de streaming de vídeo, redes de IoT (Internet of Things) e demandas em tempo real, demandando altas taxas de transmissão, baixa latência e conectividade a um elevado número de usuários. Para cumprir tais requerimentos é necessária a utilização de um quantitativo crescente de recursos computacionais nos servidores de CORE.

Um grande desafio nesse ambiente é a alocação de recursos para a realização de atividades relacionadas à proveninecia de serviços para os usuários finais de uma rede CORE 5G convergente. Quando os recursos são alocados com uma maior eficiencia e aproveitamento, é possível que as operadoras possam ter mais recursos que possam ser investidos em locais mais afastados, melhorando a experiência de usuários em todos os cantos mais remotos do país.

Palavras-chave: Redes Móveis, Redes CORE, Telefonia Móvel, Rede 4G, Rede 5G, NFVI, Alocação de Recursos, CPU, Recursos Computacionais, Otimização

Summary

List of Figures

List of Tables

List of Abreviations

1G – First Generation
AMPS – Advanced Mobile Phone System
2G – Second Generation
GSM – Global System for Mobile
EDGE – Enhanced Data Rates for GSM Evolution
3G – Third Generation
UMTS – Universal Mobile Telecommunications System
W-CDMA – Wideband Code Division Multiple Access
HPSA – High Speed Packet Access
4G – Fourth Generation
LTE – Long Term Evolved
5G – Fifth Generation
NR – New Radio
NSA – Non-Standalone
SA – Standalone
NFV – Network Function Virtualization
NF – Network Function
RAM – Random Access Memory
UE – User Equipment
SMSC – Short Message Service Center
P-GW – PDN Gateway
NAT – Network Address Translator
VoIP – Voice over IP
IP – Internet Protocol
DEE – *Departamento de Engenharia Elétrica*
CETUC – *Centro de Estudos em Telecomunicações*
MIMO – Multiple-Input Multiple-Output

1 Introduction

In this chapter the research background and the motivations of this thesis are presented. Then, the main contributions are shown. Also, the structure of this study are provided. The last section shows the notations used throughout the thesis are introduced.

a Motivation

The first cellular network standard used for 1G is the AMPS. This innovation was revolutionary at that time, even if it looks extremely limited. This generation of cellular network was initially implemented in the United States and some limited developed countries but later was expanded to other countries, including Brazil.

This was a great improvement for professionals who don't have a fixed workplace, such as salesmen who could expedite order and improve communication with teams and customers on demand. A Key point is that this technology was extremely expensive and was primarily used for local and quick calls.

In the beginning of the 1990's emerged the 2G technology as , it would only become popular in Brazil in the end of that decade. This was really where Brazil started to enter the Mobile Network Development. 2G had, in counterpart to 1G, digital technology, text messages and data services of up to 100 Kbps in the beginning of the life cycle of this generation. The possibility of sending and receiving quick messages was very positive to the wide spread of the technology and the data services were able to provide some simple information such as email and weather forecast.

After some improvements over time, at the end of the second generation of mobile network, by using the EDGE (widely known as 2.5G) the data transfer speed could reach 384 Kbps. This was well enough for the simple and rudimentary requirements that 2G had in the beginning of it's life cycle, but was not enough for the ever growing demand of users.

3G began to have properties that are more similar to the current mobile internet. The third generation further enhanced the functionality of sending emails and text messages, they also added features that are now daily used, such as internet access, video calls, VoIP, and mobile television. The technology is still relevant in certain parts of Brazil, still representing a relevant quantity of cell phones, but with newer specifications. When 3G became available in the country, it employed the UMTS. Later, social networks became commonly utilized, this led to new opportunities in various areas like communication, marketing, and advertising. 3G added additional uses to technology associated previously only with cellphones, including the use of laptops.

This innovation allowed for the continued execution of activities in other locations besides the office. A notable example is events, such as business fairs, which can utilize technological assistance regardless of the presence of a fixed bandwidth or wireless connection.

If 3G marked the beginning of the era of smartphones, 4G led to the transformation of mobile devices into computers. 4G is currently the most common generation of cell phones in Brazil, despite the coverage of 3G still a little bigger [2], this has led to the ability to make quality video calls and watch high-resolution video content. In remote work mode, meetings are hosted in virtual spaces, and online games have become part of the connected real world.

5G was first implemented in 65 that began with countries like China, the U.S. and Brazil. In general, the new technology will facilitate faster, more resilient internet connections with smaller waits for connection. When compared to 4G that takes 6 hours to download a 60-minute HD movie, 5G can do it in just 6 seconds. Additionally, 5G can increase the number of connected people in each region, which can currently be achieved with 1 million connections per square kilometer.

According to Cisco [3], 5G networks emerge amid a high demand for connectivity, as it is projected that more than two-thirds of the world's population will be connected to the Internet, and the number of devices connected to the network will be three times greater than the world's population. 5G can have different types of usage in order to overcome such challenges. It can prioritize high data transfer, focus on low latency or high reliability. To address all these possibilities, every network needs to be built according to the necessary specifications of the services to be implemented for their specific customer, region or demand. This includes the CORE network equipments.

Despite the marvels that technology is able to achieve, computational, network, storage, and memory resources are scarce. Simply increasing the number of installed networks in response to the growing number of users would significantly raise the number of equipment, and consequently, the cost to acquire and maintain them would be even higher.

Resource allocation is a process in which the resources are allocated to consumers by cloud providers based on their flexible requirements [4]. In the context of CORE networks, resource allocation usually refers to computational and processing resources such as CPU, clock, RAM, cache memory, buffer, etc. Additionally, an element is needed to manage how services use the existing resources in the network, avoiding underutilization and exhaustion of equipment. Resource management is a challenging task as it requires expertise, overburdening the network operator and service provider. Also, it is necessary to be an agile process to avoid additional delays that could harm maximum delay requirements and result in financial losses for the company. Therefore, resource allocation is essential for this type of project as well.

An enabling paradigm of recent technology that allows flexible management of network infrastructure resources is NFV. It enables, along with virtualized network functions, the provision of resources in a more suitable manner within the network. By utilizing it, virtualized network functions such as NAT, SMSC, P-GW, and many others are abstracted from physical hardware into software [1], and a single piece of hardware can become capable of running various different virtualized functions simultaneously. Hardware costs can become much lower by using off-the-shelf servers without the need of dedicated specific models, enabling the implementation of a single physical network infrastructure with multiple virtual infrastructures running inside of it. In this way, it can be done locally at various network nodes [5].

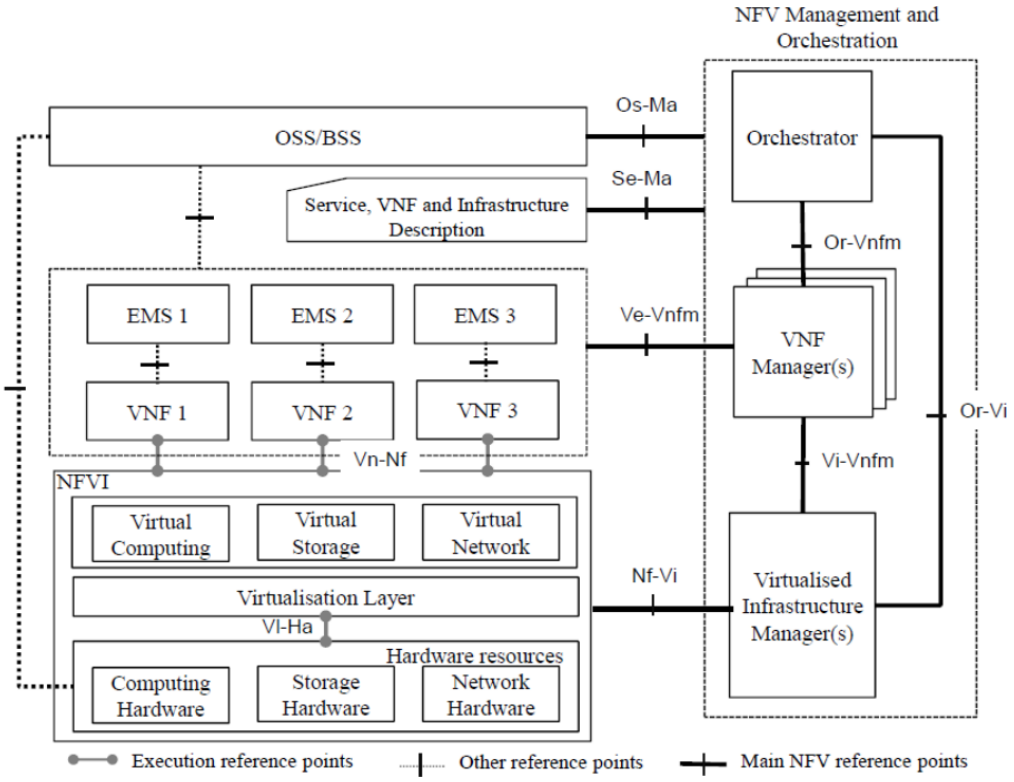

Figure 1: NFV architectural framework (ETSI, 2014b) [1]

With NFV applied to real-life scenarios, the network can have many positive outcomes, achieved through resource allocation within CORE networks, allowing the network design team and maintenance team to have many benefits compared to traditional network models [1] such as:

- Independence: software is no longer integrated with hardware. As a result, their evolution can be independent of each other.
- Flexibility: the decoupling of software from hardware helps to reassign and share the same infrastructure resources, which allows performing different functions at different times. As a result, the deployment of network functions and their connections becomes faster and more flexible.
- Scalability: decoupling software from hardware provides more flexibility to dynamically scale the actual performance of virtualized network functions with finer granularity.
- Reduced energy consumption: with the ability to scale resources up and down, TSPs are able to reduce the OPEX needed to run network devices.

- Cost: Using NFV systems could significantly reduce the cost of using networking solutions.

Therefore, to study an improvement of the resource management, this graduation project explores the MATLAB tool to propose a model capable of optimizing network resource allocation compared to unoptimized allocation models.

b Contributions

According to Cisco [3]:

- The total number of global mobile subscribers will grow from 5.1 billion (66% of the population) in 2018 to 5.7 billion (71% of the population) by 2023. Also, more than 70% of the global population will have mobile connectivity by 2023.
- 5G devices and connections will represent over 10% of global mobile devices and connections by 2023. By 2023, global mobile devices will grow from 8.8 billion in 2018 to 13.1 billion, and within this context, 1.4 billion of them will be compatible with 5G.
- The fastest-growing category of mobile devices is Machine to Machine (M2M), followed by smartphones. The mobile M2M category is projected to grow at a Compound Annual Growth Rate (CAGR) of 30% from 2018 to 2023. Smartphones will grow at a CAGR of 7% over the same period.

When networks with a large amount of users are taken into consideration, there is a need to allocate more resources in a single area to serve a large number of cells [6], for example, in football match where a large number of people are video calling when the house team scores a goal. Often, with the current resources available in the network, this is not feasible. Thus, it is necessary to preview and optimize the usage of the resources available.

Therefore, to address this challenging problem, this project proposes the creation of a strategic algorithm that improves application, specially CORE NFs, resource allocation in NFV environments in order to improve the resource usage aiming to decrease power consumption, implementation costs, maintenance costs and many other aspects of engineering deployment of new elements of CORE network.

c Thesis Outline

This thesis is organized as follows:

- Section 1 gives some technical background on this thesis;
- Section 2 shows the overall proposed system model and describes the insight and design of the NFVI architecture;
- Section 3 presents the construction of the theoretical achievable rates and the practical results;
- Section 4 displays the conclusions and discusses possible extensions for the presented study;

d Notation

Regarding the notation, bold upper and lower case letters such as $\mathbf{A}$ and $\mathbf{a}$ denote matrices and vectors, respectively. $\mathbf{I}_n$ is a $n \times n$ identity matrix. Additionally, diag$(\mathbf{A})$ is a diagonal matrix only containing the diagonal elements of $\mathbf{A}$. The inverse of sine function is denoted by $\sin^{-1}(\cdot)$. Moreover, vec$(\mathbf{A})$ is the vectorized form of $\mathbf{A}$ obtained by stacking its columns, while the inverse of this operation is unvec$(\mathbf{a})$, depending on the context.

2 System Model

In this chapter, it is presented the constraints and parameters that govern the system. It is shown mathematically how to achieve the results and also the adaptations necessary to run the commands on the matlab environment.

a Cost function

The function that we will try to minimize is the cost of the project: The included costs in this calculus are the following:

- C_M = Maintenance Cost
- C_I = Implementation Cost

In the first year (C_{A1}) the maintenance cost will be added together with the Implementation Cost. The total cost can be measured as an annual cost (C_{AN}) for the following years.

$$\mathrm{C}_{A1} = vCPU * (\mathrm{C}_I + \mathrm{C}_M) \tag{1}$$

$$\mathrm{C}_{AN} = vCPU * (\mathrm{C}_M) \forall (\mathrm{N} \in \mathbb{Z}) \wedge (\mathrm{N} > 1) \tag{2}$$

The annual cost for the first year will be the product between the quantity of total vCPU by the result of the sum of implementation cost and the maintenance cost over the period of one year. The annual cost of the following years will only be the product of the total vCPU quantity and the maintenance cost of each year period. As in this case we are isolating one specific resource (logical CPU), the cost calculation will be entirely based on it.

In order to fulfill the requirements for telecommunications usage scenario, it is necessary to have a product that have a very low failure rate, together with many other requirements that surely are not necessary for the general consumer. For this reason, CPU chip manufactures and telecommunication companies have been using and producing chips and parts with a very high standard. This standard is Usually called Telecommunications-Grade or Carrier-Grade equipment [7].

Based on public data provided by telecommunications companies [8] and CPU Chip manufactures [9], we can find the following information about a certain model of a carrier grade CPU chip:

- 1520,50 USD ≈ R$7568,63 (Exchange Rate from 2023/09/08 where 1USD = 4,98 BRL [10])
- pCPU = 16
- vCPU = pCPU * 2 = 32
- TDP = 150W

Having said that, the price per vCPU will be:

$$\frac{\mathrm{R\$7568,63}}{32} \approx \mathrm{R\$236,52} \tag{3}$$

It is also possible to calculate the average Power withdraw for each individual vCPU based on the provided TDP:

$$\frac{\mathrm{TDP}}{\mathrm{vCPU}} = \frac{150\mathrm{W}}{32} \approx 4,7\mathrm{W} \tag{4}$$

Considering that the year have an average 8766h, the total amount of Energy spent by a vCPU is approximately:

$$\frac{150\mathrm{W}}{32} * 8766h = 41090,625Wh = 4,1090625 * 10^{-2}MWh \tag{5}$$

The average energy cost for the industry in Brazil is R$ 487,14/MWh [11], so the approximate average cost per year for each vCPU will be:

$$4,1090625 * 10^{-2}\mathrm{MWh} * \frac{\mathrm{R\$487,14}}{1\mathrm{MWh}} \approx \mathrm{R\$20,02} \tag{6}$$

With all the provided information, it is possible to calculate the approximate Cost Function for our specific scenario:

- C_{A1} = C_I+C_M
- C_{AN} = C_M

$$\mathrm{C}_{A1}(BRL) \approx (236,52 + 20,02) * vCPU \approx 256,54 * vCPU \tag{7}$$

$$\mathrm{C}_{AN}(BRL) \approx 20,02 * vCPU \tag{8}$$

We will consider the vCPU resource that are not in use, because the vCPU resource in use will be utilized to bring profit to the network provider, the objective that we must decrease in order for the optimization take place. The goal is to minimize the cost, minimizing the quantity of idle vCPUs in the telecommunications provider servers.

b Resource Allocation Optimization

In order to be able to model the system, it was isolated solely the resource of vCPU in the proposed model. In real-life scenarios there will be many other constraints related to other resource types such as Memory, Storage, IOPS, NIC requirements and others.

The theoretical calculation for the resource allocation optimization comes from the statement that if a blade's resource is currently not being used, it is not necessary to include it in the scope of the project.

In order to calculate the availability of the blade's resources, we must present the necessities and constraints that the problem involves. We will describe it in a mathematical problem and it will be used in order to model the simulation software.

Definitions:

- K = Parameter that indicates the number of VMs.
- B = Parameter that indicates the number of blades.
- M = Parameter that indicates the number of vCPU per each blade.
- i = Index that represents a VM. Each VM have an "i" index.
- j = Index that represents a VM. Each VM have an "j" index.
- b = Index that represents a blade. Each blade have an "b" index.
- z[b] = Vector that indicates the idle vCPU in each "b" blade.
- c[i] = Vector that indicates the required vCPU for each "i" VM.
- X[i,b]= Binary Decision Matrix that shows VM allocation per blade.
- X[i,b]= Binary Decision Matrix that shows VM allocation per blade.
- Q[i,j]= Binary Parameter Matrix that shows Anti-Affinity rules.

Further developing some of the presented concepts:

- "X[i,b] = 1" means that VM "i" is allocated to the blade "b".
- "X[i,b] = 0" means that VM "i" is not allocated to the blade "b".
- X[i,b] $\in \{0,1\} \forall (1 \leq \mathrm{i} \leq \mathrm{K}) \wedge (1 \leq \mathrm{b} \leq \mathrm{B})$
- "Q[i,j] = 1" means that VM "i" cannot be allocated to the same blade as VM "j".
- "Q[i,j] = 0" means that VM "i" can be allocated to the same blade as VM "j".
- Q[i,j] $\in \{0,1\} \forall (1 \leq \mathrm{i} \leq \mathrm{K}) \wedge (1 \leq \mathrm{j} \leq \mathrm{K})$
- Q[i,j] = 1 $\forall (\mathrm{i} = \mathrm{j})$

Constraints:

- 1 - Allocation Constrain

  Each VM must be installed within a single blade.

$$\sum_{\mathrm{b}=1}^{\mathrm{B}} \mathrm{X}[\mathrm{i},\mathrm{b}] = 1 \; \forall \; (1 \leq \mathrm{i} \leq \mathrm{K}) \tag{9}$$

  Because all the vCPUs of each VM must be inside the same blade, we can consider that the maximum number of "B" $\mathrm{B}_{\mathrm{MAX}} = K \; ; \; B \leq K$

- 2 - Capacity Constrain

  Each blade have a limited amount of vCPUs.

$$\sum_{i=1}^{K} c[i] * X[i,b] \leq M \; \forall \; (1 \leq b \leq B) \tag{10}$$

- 3 - Incompatibility Constrain

  Incompatible VMs cannot be in the same blade.

$$X[i,b] + X[j,b] + Q[i,j] \leq 2 \; \forall \; (1 \leq b \leq B) \wedge (1 \leq i \leq K) \wedge (1 \leq j \leq K) \tag{11}$$

  This will guarantee that in case 2 VMs are in the same blade, they will not have incompatibility restriction.

- 4 - Typology Constrain

  Each blade can only have 2 VMs of each type.

$$\sum_{r=1}^{R} X[i_r,b] \leq 2 \; \forall \; (1 \leq b \leq B) \tag{12}$$

  This will guarantee that it is not possible to have more than 2 VMs of the same type in each individual blade.

Considering the proposed constraints and defined parameters, we can proposed the following allocation function:

$$\sum_{b=1}^{B} \sum_{i=1}^{K} c[i] * X[i,b] \tag{13}$$

In order to calculate the available vCPU quantity in each host we can follow the definition:

$$z[b] = M - \sum_{i=1}^{K} c[i] * X[i,b] \tag{14}$$

And the total amount of available vCPU resource will be:

$$\|z\| = \sum_{b=1}^{B} (M - \sum_{i=1}^{K} c[i] * X[i,b]) \tag{15}$$

In order to confirm that the resources from the last blade are idle, we must confirm that there is no VM allocated to the last blade. To achieve that we will use the auxiliary rectangular function:

$$\prod \frac{b - (\frac{B-1}{2})}{B-1} \tag{16}$$

Applying this auxiliary function to the defined equation of ||z||:

$$\|z\| = \sum_{b=1}^{B} (\prod \frac{b - (\frac{B-1}{2})}{B-1} * (M - \sum_{i=1}^{K} c[i] * X[i,b])) \tag{17}$$

If the presented equation is true, we can conclude that there is no resource allocated into the last blade ($z[B] = M$). Based on this, we can consider that $\sum_{i=1}^{K} c[i] * X[i,B] = 0$, therefore:

$$\sum_{i=1}^{K} X[i,B] = 0 \tag{18}$$

In this case, it is possible to reach z[B] = M. In order to continue the optimization algorithm, we must decrease one blade and try again the test (B = B - 1). This procedure should be repeated until it is not possible to have a scenario where z[B] = M. When this scenario is no longer possible, the current value of B is the optimized value for this application.

c Software Adaptation

In order to Matlab easily solve the proposed problem, we must linearize the Matrices. X[B,K] for instance will use the following relatioship between the Matrix and it's linearized vector $\underline{y}$:

$$\mathrm{pos_y(i,b)} = \mathrm{(i-1)B + b}\ \forall\ \mathrm{(1 \leq b \leq B)} \wedge \mathrm{(1 \leq i \leq K)} \tag{19}$$

This way, the linearized vector $\underline{y}$ will have the following format:

$$\underline{y} = [\mathrm{x_{11}, x_{12}, ..., x_{1B}, x_{21}, x_{22}, ..., x_{2B}, ..., ..., ..., x_{(K-1)B}, x_{K1}, x_{K2}, ..., x_{KB}}]^{\mathrm{T}} \tag{20}$$

The Constraint Number 1 can be described as an equality, such equality can be described the following format : $\tilde{A}\underline{y} = 1$ where $\dim(\tilde{A}) = K \times N$

$$A = \left.\begin{bmatrix} \overbrace{11\cdots1}^{B} & 00\cdots0 & \cdots & 00\cdots0 \\ 00\cdots0 & \overbrace{11\cdots1}^{B} & \cdots & 00\cdots0 \\ \vdots & \vdots & \ddots & \vdots \\ 00\cdots0 & 00\cdots0 & \cdots & \overbrace{11\cdots1}^{B} \end{bmatrix}\right\} \mathrm{K} \Rightarrow \tag{21}$$

$$\Rightarrow \text{ defining } \underline{u}_B = \begin{bmatrix} \overbrace{11\cdots1}^{\mathrm{B}} \\ \end{bmatrix},\ \text{so } A = I_K \otimes \underline{u}_B \tag{22}$$

The Constraints Number 2, 3 and 4 can all be described as a different inequality.

The constraint number 2 can be written in the format $\tilde{B}\underline{y} \leq \underline{\mathrm{M}}$ where $\dim(\tilde{B}) = B \times N$ and can be expressed by:

$$\tilde{B} = [c_1 I_B, c_2 I_B, ..., c_K I_B] = I_B \otimes \underbrace{[c_1, c_2, ..., c_K]}_{\underline{c}} = I_B \otimes \underline{c} \tag{23}$$

Meanwhile, for the constraint number 3, we must consider that the pairs of incompatible VMs can be written in the format $\{[u_1, v_1], [u_2, v_2], ..., [u_L, v_L]\}$. Based on this, we can express the constraint by:

$$\mathrm{y_{pos(u_n,k)} + y_{pos(v_n,k)} \leq 1}\ \forall\ \mathrm{(1 \leq k \leq K)} \wedge \mathrm{(1 \leq n \leq L)} \tag{24}$$

This restriction can be written in the for of:

$$\tilde{C}\underline{y} \leq 1 \wedge \dim(\tilde{C}) = K \times N \tag{25}$$

Lastly, for the Constraint number 4, we can follow the same logic as of the previous, where:

$$\tilde{D}_r\underline{y} \leq 2 \wedge (\dim(\tilde{D}) = K \times N) \wedge (1 \leq \mathrm{r} \leq \mathrm{R}) \tag{26}$$

The objective function can be written in the format of:

$$z = \underline{f}^T \cdot \underline{y}\ \forall\ f_i = \begin{cases} 1 & \text{if } \ i = k \cdot B\ \forall\ (1 \leq k \leq K) \\ 0 & \text{else} \end{cases} \tag{27}$$

The Matrix E should be defined as the following:

$$E = \begin{bmatrix} \tilde{B} \\ \tilde{C} \\ \tilde{D}_1 \\ \vdots \\ \tilde{D}_R \end{bmatrix} \tag{28}$$

and the vector $\underline{e}$ should be:

$$\underline{e} = \begin{bmatrix} M \\ \underline{1} \\ \underline{2} \\ \vdots \\ \underline{2} \end{bmatrix} \tag{29}$$

With all the considerations presented, the optimization problem can easily be solved by matlab if we express it like the following:

$$\min z = \underline{f}^T \cdot \underline{x} \text{ s.t. } (\tilde{A} \cdot \underline{x} = 1) \wedge (E \cdot \underline{x} \leq \underline{e}) \, \forall \, \underline{x} \in \{0,1\}^N \tag{30}$$

d Summary

In this chapter we have presented a review on some of the technical definitions and system constraints that should be considered in order for the system to be understood and simulated. It was also presented the changes that were required in order for the matlab software better understand the problem and is able to easily solve it.

3 Achievable Rate Analysis

In this chapter, it will be presented the theoretical achievable rate for both best scenario and worst scenario possible, given some specific constraints. The algorithm that was built will also provide some very clarifying results that will be analyzed and compared to the theoretical scenarios. Then, the numerical results are demonstrating the potential of the proposed system.

a Theoretical Achievable Rate

In order to understand the theoretical worst scenario we will consider all the constraints and install each VM in a single Blade. This is more clear when it is analyzed based on the equation 9. Based on this constraint, we can see that in the worst scenario possible there should be one blade per each VM.

$$\mathrm{B_{MAX}} = \mathrm{K} \tag{31}$$

Using a similar analogy, it is possible to measure the best scenario possible, where we isolate all the constraints and simply install all the VMs using every resource available.

$$\mathrm{B_{MIN}} = \lceil \frac{\|\mathrm{c}\|}{\mathrm{M}} \rceil \tag{32}$$

Based on the concepts presented, our optimized number of blades B should be:

$$\lceil \frac{\|\mathrm{c}\|}{\mathrm{M}} \rceil \leq \mathrm{B} \leq \mathrm{K} \tag{33}$$

Mathematically, there will be some scenarios that it will be possible to reach the minimum value of B and also some scenarios that will not be possible for it to be lower than K. This shows that the algorithm objective is clearly to reach the lowest number as possible, as far away as possible from the result of Equation 31 and as close as possible to the result of Equation 32.

b Numerical Results

Based on some examples of 5G Core applications CPU requirements and constraints, it is possible to execute the proposed optimization script and obtain the amount of resources that were optimized. This can also be used to calculate the cost that were spared using this algorithm using the Equations 7 and 8.

Based on this, the "Optimization" numerical value should be defined as the following:

$$\mathrm{Optimization}(\%) = \frac{\mathrm{B_{MAX}} - \mathrm{B}}{\mathrm{B_{MAX}} - \mathrm{B_{MIN}}} * 100 \tag{34}$$

The following results were obtained by using the proposed algorithm and the value of M CPUs per blade equal to 40:

Simulation Name	$\mathrm{B_{MIN}}$	$\mathrm{B_{MAX}}$	B	Optimization (%)
20231105	53	160	55	98
20231018	10	31	10	100
20231206_1	19	44	20	96
20231206_2	27	54	33	78
20231206_3	5	24	5	100
20231206_4	26	97	26	100

Table 1: Numerical Results for different simulations.

By comparing the numerical results obtained and the theoretical initial scenario, it is possible to obtain the quantity of Blades that were improved in the optimization algorithm.

$$\mathrm{Optimization} = \frac{\mathrm{B_{MAX}} - \mathrm{B}}{\mathrm{B_{MAX}} - \mathrm{B_{MIN}}} \tag{35}$$

Simulation Name	B_{MAX}	B	Optimization	C_{A1} (BRL)	C_{AN} (BRL)
20231105	160	55	105	1.077.468	84084
20231018	31	10	21	215.493,60	16.816,80
20231206_1	44	20	24	246.278,40	19.219,20
20231206_2	54	33	21	215.493,60	16.816,80
20231206_3	24	5	19	194.970,40	15.215,20
20231206_4	97	26	71	728.573,60	56.856,80

Table 2: Cost Improvement in the simulated scenarios.

By using the equations 7 , 8 and considering the M value of 40 vCPU per each blade used in the simulations, it is possible to show how the improvements will impact in the theme of financial resources to the company:

It is possible, in some cases, to see an improvement of over 1 million BRL. This is a very powerful measurement that, by using a strong algorithm that can generate positive results, will have many achievements in a company's financial situation. This can impact in:

- Financial Return to Investors.
- More Investments on the company, generating an even better service for the consumers.
- Financial Return to Employees.
- Improvements on the Energy Distribution.
- Lower A/C Requirements for the server rooms.
- Lower maintenance costs, for it there will be less servers to fail.

c Summary

This chapter has shown that by considering the proposed algorithm in the Chapter 2 , when matched with the ideal case scenario in the VM allocation result it is possible to achieve great success. It is possible to observe improvement of up to 100% when compared to the worst and best case scenario. Simulation results show that the proposed algorithm outperforms the simple allocation algorithm and can still match the requirements for network reliability.

4 Conclusion and Future Work

In this project it is proposed a novel algorithm that can solve a resource allocation problem, optimizing it based on the given requirements.

a Conclusions

Different from conventional systems, the proposed algorithm can consider many characteristics at the same time in order to reach the optimized resource allocation. The result vector can be seen as an allocation vector for all the VMs inside all the blades and this can be useful for many NFV related tasks for this VMs. The algorithm can be reviewed to include many other constraints that may be useful for many other specific scenarios.

The primary goal of this project concept is to improve the Allocation of computational resources for CORE Network application but this could be expanded to any other application that requires or could use a NFV environment.

Another Key point is that if the Client of this connection (the owner of the VM services) can choose, before-hand the location of it's VMs based on it's requirements, the given resources and the output result of the algorithm, it will be possible to further improve the processing power on the server side.

Based on the algorithm simulation, simulation results show that the proposed system is superior to conventional allocation algorithm. Moreover, numerical results of adding more VMs of different sizes and consequently, a greater number of maximum blades on the system, show an even greater improvement, displaying the system scalability.

b Future Work

Several future research topic are suggested:

- Add more constraints in the system model in order to better suit the real world applications.
- Improve the system model in order to ease the possibility of adding new constraints.
- Further study the algorithm in order to improve it's efficiency when the system have a great number of VMs.
- Adapt the algorithm to dynamic environments where is necessary to have near-real-time resource allocation adjustments.
- Conduct a study on the scalability of the algorithm in large-scale network environments.

A Appendix

In this chapter, it will be presented the relevant information used in order to obtain the results for the simulation and the project.

a matlab code

```
clear
Start = tic;
DEBUG = 1;
caso  = {'none','iter'};
%-------------------------------------------------------------------------
%   Dados do projeto
%-------------------------------------------------------------------------
% DATA = readmatrix('valores_simulacao_20231018.csv');
DATA = readmatrix('valores_simulacao_20231206_4_forMILP4.csv');
nU = DATA(1,1);                 % no. de áusurios
VM = DATA(2,1:nU);              % demanda de CPU's pelos áusurios

nT = DATA(3,1);                 % no. de casos de tipologia
if nT>0
offset = 3;
Tipologia = cell(nT,2);
    for k=1:nT
        Line = 2*k-1;
        SIZE = DATA(Line+offset,1);
        Tipologia{k,1} = SIZE;
        Tipologia{k,2} = DATA(Line+offset+1,1:SIZE);
    end
end

offset = 3+2*nT;
nQ = DATA(offset+1,1);                % no. de casos de incompatibilidades
Q  = DATA(offset+2:offset+nQ+1,1:2);  % matriz de incompatibilidades

M  = 40;                              % capacidade da blade em CPU's
if max(VM) > M
    fprintf('\n\çãtViolao da regra de ouro. Bye...\n\n');
    return;
end

fprintf('\n\útNmero de VMs            : %4d\n',nU);
fprintf('\tCapacidade da blade      : %4d cpus\n',M);
fprintf('\tNo. de imcompatibilidades: %4d\n',nQ);
fprintf('\tNo. de tipologias        : %4d\n\n',nT);

x0 = reshape(eye(nU),[nU*nU,1]);   %çãsoluo inicial ífactvel

B = nU;
while(B>=2)
    fprintf('************** CASO B = %d *****************\n',B);

    tIter = tic;
    N  = nU*B;                      % ãdimenso do problema
    nA = (nQ+nT+1)*B;               % no. inicial de restr. de desigualdade

    fprintf('\tNo. de ávariveis             : %8d\n',N);
    fprintf('\tNo. de rest. de igualdade    : %8d\n',N);
    fprintf('\tNo. de rest. de desigualdade : %8d\n',nA);
%-------------------------------------------------------------------------
%   EQUALITY CONSTRAINTS    Aeq.x = beq     dim(Aeq) = nU x N
```

```
%
% çã  restrio de çãalocao
%
    Aeq = kron(eye(nU),ones(1,B));
    beq = ones(nU,1);

    r = Aeq*x0-beq;
    if min(r)~=0 && max(r)~=0
        fprintf('ERRO 1\n');
        return;
    end
%-------------------------------------------------------------------------------
%   INEQUALITY CONSTRAINTS    A.x <= b     dim(A) = (nQ+nT+1).B x N
%
    nA = (nQ+nT+1)*B;
    A  = zeros(nA,N);
    b  = zeros(nA,1);
    line = B+1;
%
% çã  restrio de capacidade
%
    A(1:B,:) = kron(VM,eye(B));
    b(1:B)   = M*ones(B,1);
%
% çã  restrio de incompatibildade
%
    if nQ>0
        for w=1:nQ
            for r=1:B
                for s=1:2
                    A(line,(Q(w,s)-1)*B+r) = 1;
                end
                b(line) = 1;
                line = line+1;
            end
        end
    end
%
% çã  restrio de Tipologia
%
    if nT>0
        for s=1:nT
            nS = Tipologia{s,1};
            Lista = Tipologia{s,2};
            for r=1:B
                for w=1:nS
                    A(line,(Lista(w)-1)*B+r) = 1;
                end
                b(line) = 2;
                line = line+1;
            end
        end
    end
%
% çã  funo objetivo
%
    temp = zeros(B,1);
    temp(B) = 1;
    c  = kron(ones(nU,1),temp);
%
%   bounds
```

```
%
    lb = zeros(N,1);
    ub = ones(N,1);
%
% çã  optimizao (0,1)
%
    intcon  = 1:N;
    options = optimoptions('intlinprog','Display',caso{DEBUG+1});
    [x,fval,exitflag] = intlinprog(c,intcon,A,b,Aeq,beq,lb,ub,x0,options);

    fprintf('\çãtDurao: %.3f seg.\n\n',toc(tIter));
%
%   check se continua ou ãno
%
    if fval>0
        sol  = x0;
        Bopt = B;
        break;
    else
        w  = reshape(x,[B,nU])';
        idx = sum(w)>0;
        B   = sum(idx);
        ww  = w(:,idx);
        x0 = reshape(ww',[nU*B,1]);
    end
end
%
% ã  Impresso da çãsoluo final
%
SOL = reshape(x0,[Bopt,nU])';
fprintf('\n\çãtSoluo Ótima envolve %d blades\n\n',Bopt);
for s=1:Bopt
    idx  = find(SOL(:,s)==1);
    nCPU = sum(VM(idx));
    fprintf('\tBlade %2d - %2d CPUs em uso çã  Utilizao de %6.2f%%\n',...
        s,nCPU,100*nCPU/M);
    for r=1:length(idx)
        fprintf('\t\tVM-%2d (%2d CPUs)\n ',idx(r),VM(idx(r)));
    end
    fprintf('\n');
end

fprintf('\çãtDurao: %.3f seg.\n',toc(Start));
```

b example csv 20231018

```
31;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
16;16;8;12;12;4;4;8;8;4;4;4;8;8;8;4;4;4;36;36;36;36;36;12;12;4;4;12;12;2;2
0;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
19;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1;2;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1;3;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
2;3;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
4;5;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
6;7;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
8;9;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
10;11;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
10;12;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
11;12;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
13;14;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
13;15;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
14;15;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
16;17;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
16;18;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
17;18;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
24;25;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
26;27;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
28;29;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
30;31;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

c example csv 20231206-1

```
44;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
8;8;4;4;4;8;8;8;4;4;4;4;36;36;36;36;36;36;36;36;36;36;36;36;36;36;36;12;12;4;4;12;12;12;12;2;2;16;16;12;10;10;2;2
3;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
15;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
13;14;15;16;17;18;19;20;21;22;23;24;25;26;27;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
4;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
32;33;34;35;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
3;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
38;39;40;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
10;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
1;2;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
3;4;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
3;5;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
4;5;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
6;7;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
6;8;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
7;8;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
28;29;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
30;31;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
36;37;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

d example csv 20231206-3

24;;;;;;;;;;;;;;;;;;;;;;;;
6;6;2;2;4;4;4;4;20;20;4;4;6;6;4;4;40;40;4;4;4;4;2;2 0;;;;;;;;;;;;;;;;;;;;;;;;
12;;;;;;;;;;;;;;;;;;;;;;;;
1;2;;;;;;;;;;;;;;;;;;;;;;;;
3;4;;;;;;;;;;;;;;;;;;;;;;;;
5;6;;;;;;;;;;;;;;;;;;;;;;;;
7;8;;;;;;;;;;;;;;;;;;;;;;;;
9;10;;;;;;;;;;;;;;;;;;;;;;;;
11;12;;;;;;;;;;;;;;;;;;;;;;;;
13;14;;;;;;;;;;;;;;;;;;;;;;;;
15;16;;;;;;;;;;;;;;;;;;;;;;;;
17;18;;;;;;;;;;;;;;;;;;;;;;;;
19;20;;;;;;;;;;;;;;;;;;;;;;;;
21;22;;;;;;;;;;;;;;;;;;;;;;;;
23;24;;;;;;;;;;;;;;;;;;;;;;;;

References

[1] A. M. Alwakeel, A. K. Alnaim, and E. B. Fernandez, "A pattern for network function virtualization infrastructure (nfvi)," in *Proceedings of the 26th Conference on Pattern Languages of Programs*, 2019, pp. 1–9.

[2] G. Carvalho and S. Vasconcelos, "Análise da concentração do mercado brasileiro de telefonia móvel," *Revista de Defesa da Concorrência*, vol. 8, no. 1, pp. 47–71, 2020.

[3] Cisco, "Cisco annual internet report (2018–2023) white paper. [online]," *Cisco Systems: San Jose, CA, USA*, 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[4] M. F. Manzoor, A. Abid, M. S. Farooq, N. A. Nawaz, and U. Farooq, "Resource allocation techniques in cloud computing: A review and future directions," *Elektronika ir Elektrotechnika*, vol. 26, no. 6, pp. 40–51, 2020.

[5] C. Mouradian, T. Saha, J. Sahoo, M. Abu-Lebdeh, R. Glitho, M. Morrow, and P. Polakos, "Network functions virtualization architecture for gateways for virtualized wireless sensor and actuator networks," *IEEE Network*, vol. 30, no. 3, pp. 72–80, 2016.

[6] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2522–2545, 2016.

[7] A. S. Saito, "Implementação de suporte à alta disponibilidade em ambientes nfv usando osm," 2018.

[8] L. Huawei Technologies Co. E9000 blade server ch121 v5 compute node user guide product specifications (accessed: 2023/09/03). [Online]. Available: https://support.huawei.com/hedex/hdx.do?docid=EDOC1000053358&id=EN-US_TOPIC_0000001586802985

[9] I. Corporation. Intel xeon gold 6226r processor (accessed: 2023/09/03). [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/199347/intel-xeon-gold-6226r-processor-22m-cache-2-90-ghz.html

[10] B. C. do Brasil. Banco central do brasil currency conversion (accessed: 2023/09/08). [Online]. Available: https://www.bcb.gov.br/en/currencyconversion

[11] J. M. D. d. A. BARROS *et al.*, "Estudo de viabilidade de mercados serem atendidos como clientes do grupo a." 2021.