
T.C.

ALTINBAS UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

# ENERGY OPTIMIZATION IN CLOUD COMPUTING USING ANT COLONY OPTIMIZATION OF FOG NODES

Ali Talib ALHUMIRI

Master of Science

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Istanbul, 2021

# ENERGY OPTIMIZATION IN CLOUD COMPUTING USING ANT COLONY OPTIMIZATION OF FOG NODES

by

**Ali Talib ALHUMIRI**

Electrical and Computer Engineering

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled “ENERGY OPTIMIZATION IN CLOUD COMPUTING USING ANT COLONY OPTIMIZATION OF FOG NODES” prepared and presented by “Ali Talib ALHUMIRI” was accepted as a Master of Science Thesis in Electrical and Computer Engineering.

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Supervisor

Thesis Defense Jury Members:

| | | |
|---|---|---|
| Asst. Prof. Dr. Abdullahi Abdu IBRAHIM | School of Engineering and Architecture, Altinbas University | _____ |
| Asst. Prof. Dr. Muhammad ILYAS | School of Engineering and Architecture, Altinbas University | _____ |
| Asst. Prof. Dr. Sewale Musadaq Taha | Faculty of communication, Beykent University | _____ |

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate studies:

____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ali Talib ALHUMIR

# DEDICATION

I would like to thank God Almighty for the strength of mind, health, strength, guidance, knowledge and skills to complete this study.

This thesis is sincerely dedicated to my parents. There are no words to describe what it means to me There is nothing I can repay for what you did to me. I will continue to do my best to fulfill your expectations.

Finally, I dedicated this to my wife, my brother, my sisters, my relatives, my friends and to the martyrs of Iraq who fell for the freedom of this country and they were the ones who encouraged me during this study.

# ACKNOWLEDGEMENTS

I would like to thank my supervisor Asst. prof. Dr. Abdullahi Abdu IBRAHIM for all support during my study. It's great pleasure to express my deepest gratitude to my friends who have shared with me best moments during my study for the Master degree.

# ABSTRACT

## ENERGY OPTIMIZATION IN CLOUD COMPUTING USING ANT COLONY OPTIMIZATION OF FOG NODES

ALHUMIRI, Ali Talib,

M.Sc., Electrical and Computer Engineering ,Altınbaş University,

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: 08/2021

Pages: 48

Smart grids are extended electrical networks that consist of multiple energy sources and a quantity of devices connected to each other to provide the best reliability in power generation and energy management. Or by improving the contact within the network parts and their control, and the goal in this is to improve the aspects of the smart grid system, and suggest a hidden or opaque logical controller for renewable energy and its fossil sources in a network and its system that operates on the Internet and supervises the state of the network, its malfunctions and how to solve the controller Foggy, which provides an extra floor of smart grid support.

**Keywords:** Cloud, SLA, VMware, Architecture Computing, Service.

# TABLE OF CONTENTS

**Pages**

# LIST OF FIGURES

**Pages**

# LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| SLA | : | Service-Level Agreement |
| IoT | : | Internet of Things |
| PUE | : | Power Usage Ectiveness |
| SOA | : | Service Oriented Architecture |
| VMWare | : | Virtual Machines Ware |
| VPN | : | Virtual Private Network |
| IAAS | : | Infrastructure as a Service |
| SAAS | : | Software as a Service |
| PAAS | : | Platform as a Service |
| CSN | : | Cloud Service Consumer |
| ISN | : | Cloud Service Integrator |
| PSN | : | Cloud Service Provider |
| QOS | : | Quality of Service |
| WSLA | : | Web Service Level Agreement |
| VRP | : | Vehicle Routing Problem |
| FAP | : | Fleet Assignment Problem |
| WFP | : | Work Force Problem |
| TSP | : | Traveling Salesman Problem |

# 1. INTRODUCTION

## 1.1 BACKGROUND

It is known throughout the world that telecommunications networks are constantly evolving. That is why more and more new technologies and applications are being developed that aim to facilitate and streamline the activities of both companies and private users. This constant evolution implies the appearance of new possibilities within the world of communications. One of them is the possibility of decentralizing certain functions through application migration. In this context, what is known as Cloud Computing arises. This type of system consists of a cloud of servers that provide certain services to the users who access it. Every computer network carries a large number of complexities that must be taken seriously and given the corresponding relevance. One of these adversities is the consumption of the system, and how to optimize it in order to reduce costs and increase the speed of communications. [2]:

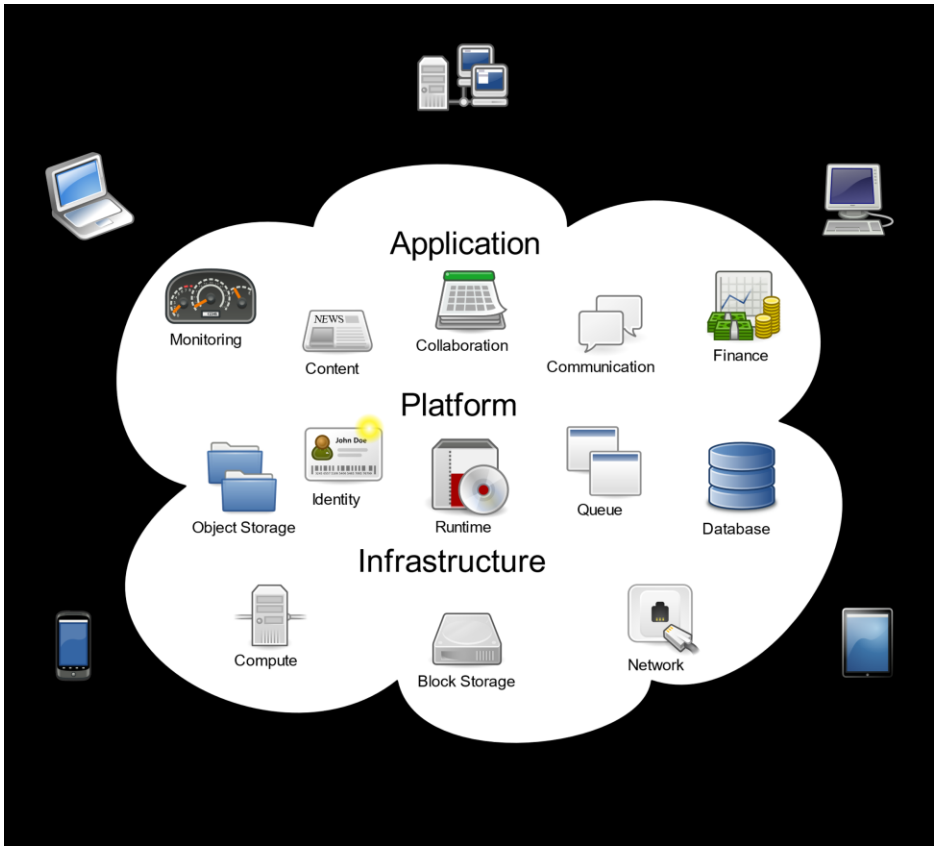


**Figure 1.1:** Basic Cloud Computing System.

## 1.2 MOTINATION

Recently, the process of giant companies has been transferred and internationalized, and the rapid development in the use of information technologies and data processing, led to the growth of the computing needs of companies and giant institutions at a tremendous speed and outperformed computing and its power in order to meet the need of computing systems and this development sparked interest in architecture computing and make it completely dependent on the simultaneous execution of operations on multiple computing equipment.

## 1.3 PROBLEM STATEMENT

The main aim of any cloud computing is to solve the problem of energy optimization and resource allocation and to develop a series of algorithms that provide cost savings in a Cloud Computing system based on SLA. However, this main objective can be broken down into a series of more specific objectives. These goals are listed below:

• Study of energy savings in a Cloud Computing environment.

• Define a series of energy saving techniques or mechanisms.

• Put these techniques to the test by developing efficient algorithms. It should be noted that it is necessary to achieve a series of milestones or goals that are the means to achieve the objectives. These milestones are the following:

1. Establish a real and concrete environment on which to work and carry out the necessary tests related to the project.

2. Create a certain level of service through the relationships agreed with the SLA protocol.

3. Design and develop various algorithms that reduce the cost of the system without affecting the quality of service requested by the user, using energy-saving mechanisms common in cloud-type computing environments.

4. Analyze the performance of the algorithms developed in the project through comparisons, tables and graphs of results

## 1.4 CONTRIBUTION

This project proposes three algorithms that provide certain cost savings by minimizing the effective use time of the cloud nodes. For this, we have worked with a mechanism for negotiating service quality parameters called Service Level Agreement Thanks to this protocol, certain levels of quality of service have been established (user profiles, guaranteed traffic, ...). The proposed algorithms must be implemented in elements of the cloud called Schedulers. In order to carry out this project, a series of variables that directly influence the total consumption of the system have been taken into account, such as the number of CPUs (or cloud nodes) that is agreed in the SLA negotiation, or the times task or temporary limit values (deadline times). Other factors outside the SLA negotiation have also been taken into consideration, such as the total virtual time of the system. In order to determine the efficiency of all the algorithms proposed and implemented in this project, a series of simulations have been carried out in the project. Thanks to these simulations, it has been possible to study the behavior and benefits of the management and resource allocation mechanisms proposed in this document.

## 1.5 THESIS ORGANIZATION

The thesis is structured as follows: Section 2 is where we review some of the previous work and implementation on smart grids. Section 3 is where we give a background about all the components. The fourth part is the space in which the detail model and its implementation are explained, the fifth part is the space in which it explains the simulation of our model and recording the details on it, the sixth section is present in our future.

# 2. LITRITAURE ART REVIEW

In order to complete the theoretical part of this report, it has been deemed necessary to make a brief comment on the state of the art in relation to how energy saving is addressed in Cloud Computing systems. The techniques discussed in this section will serve as inspiration when designing the algorithms proposed in this project, and which are detailed in chapter 3. After an exhaustive study of the savings in Cloud Computing and the way to address it by the different researchers and analysts of the sector, have drawn a series of conclusions. The main studies and proposals for cost savings in Cloud Computing systems focus on two aspects: economic and energy (but only at the hardware level). There are also authors.

Who base their research on reducing computational cost by reducing runtimes and hourly tasks (see [18])? There are many articles and reviews that address cost savings from an economic point of view. For example, V. Yarmolenko and R. Sakellariou in [9] and [16] de Nan new variables of economic rewards and compensation using the SLA protocol. A similar analysis is made by L. Wu et al. in [28], where an algorithm based on an estimated economic budgeting is designed. For their part, A. Andrzejak et al. in [21] they also make use of SLA technology to relate different types of traffic with the charging systems and thus reduce the economic cost of the system. The other major line of work to reduce the cost of the system is focused on reducing the energy cost of hardware with cooling techniques, such as the Google proposal in its whitepaper [26]. Another widely used tool for cost savings is the calculation of effectiveness indices such as PUE (Power Usage E ectiveness).

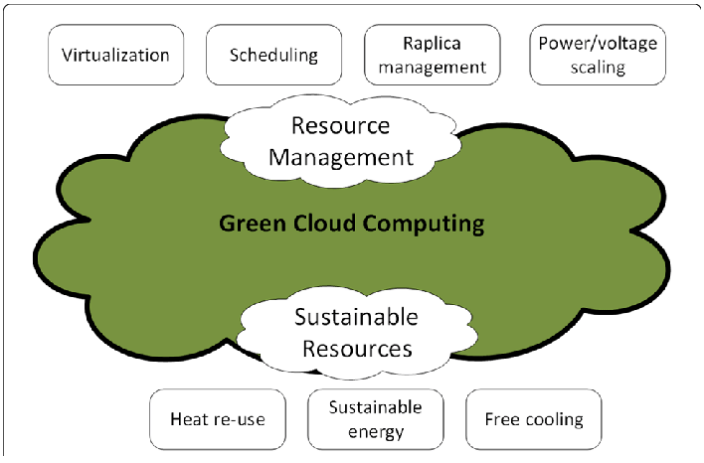


**Figure 1.2:** sustainable cloud computing

Along these lines, X. Wang et al. in [27] they focus their research on developing an energy efficiency algorithm based on this variable and the use of a very widespread Google framework called Map Reduce. Taking into account the nature of Cloud Computing systems, there are several techniques that help reduce the cost of the system, in addition to those already mentioned. As already mentioned in the previous section, virtualization is a key concept when studying and analyzing any aspect related to Cloud Computing. In that sense, and within all that is cost reduction, it is almost obligatory to talk about the migration of machines. In this type of environment, migration is a fundamental aspect, since it allows to relieve the load of certain resources and increase the flexibility of the system. Therefore, algorithms designed to reduce energy costs often take into account the state of servers or resources. If a server is saturated or heavily loaded, it is considered quarantined or isolated, and must be migrated to another virtual machine. This migration is known as load balancing, and it is one of the techniques that have been used in this project. If, on the contrary, it is desired to reduce the energy cost by avoiding the expansion of resources and using only a certain resource or MV, a technique called consolidation is usually used, and it is also one of the energy reduction mechanisms used in this project. M. Mishra et al. work in this direction in [29]. The third mechanism that has been used in this project to try to reduce the energy cost in Cloud Computing systems is what has been called Time Shifting (or Time Overlapping), inspired by the work of V. Aggarwal et al. in [30]. The Time Shifting technique or temporary displacement is used mainly in real time services. There is an international consortium called The Green Grid whose objective is to increase energy efficiency in Data Centers. It works and proposes standards in relation to everything that is energy saving in information technologies. It covers both power and energy issues and pollution and pollution. For more information, consult the web at [31].

# 3. MATERIALS AND METHODS

In this chapter we review some of the essential components that we used to implement our model; we give a brief introduction to the component.

## 3.1 CLOUD COMPUTING

Using a server(s) from the internet to handle requests at any time makes for a cloud-based system. It is possible to access your information or service on any mobile device or computer, regardless of where you are located. A number of web hosting services have been servicing their customers from the beginning. This is a smarter way to minimize costs while ensuring a consistent and protected Internet access, as well as immunizing governments and law enforcement agencies from hacker attacks. In cloud computing, it is possible to increase the amount of network-based products and services that you can use. Having a pay-per-per-use scheme provides greater efficiency in delivery and flexibility for suppliers, too. Furthermore, the buyer saves on salary (premises, specialized material, etc.). The paradigm of cloud computing will serve as a transition to transitioning to a standardized service and responding to business needs in an extensible and scalable manner. In the event of unforeseen demands or revenue peaks, paying for only the consumption rendered is permitted or financed. The appeal of cloud computing is due in part to its dynamic technological infrastructure which includes automation, rapid scale, and flexibility, as well as a high capacity to provide these advantages as well as avoiding the fraudulent and pirate use of software. Many people consider cloud computing to adopt an architecture consisting of customer service-oriented applications (Service Oriented Architecture, SOA).

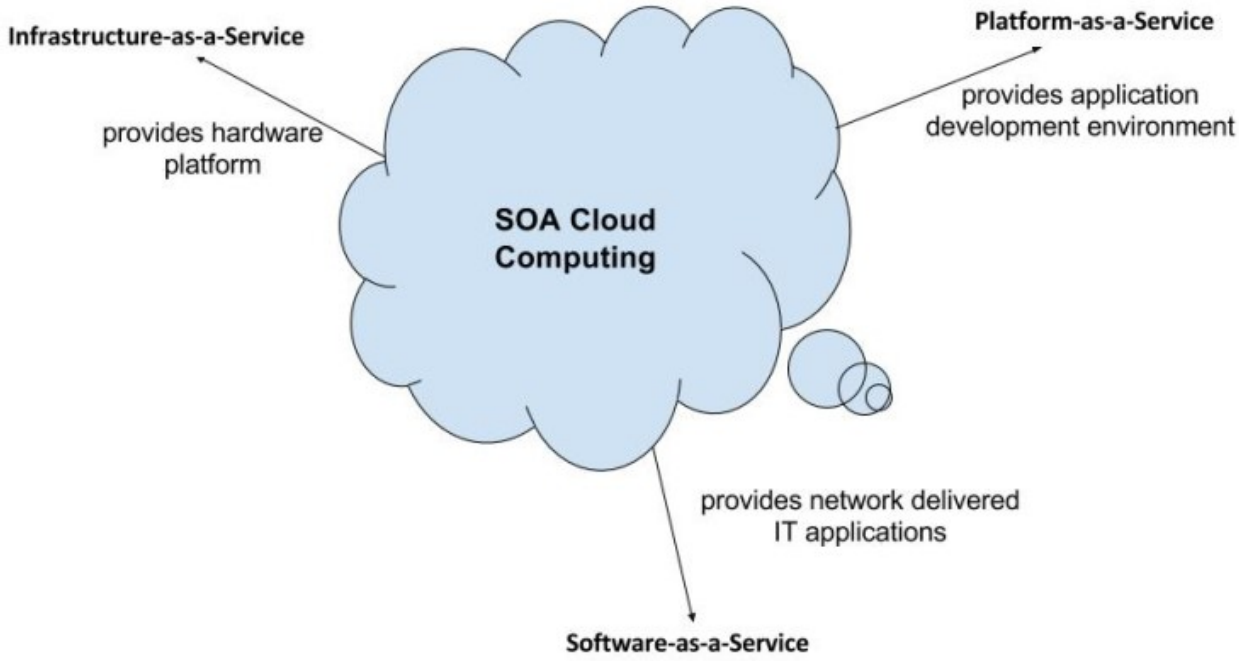


**Figure 3.1:** Software as a Service Cloud Computing

A service-oriented architecture allows for the creation of scalable and flexible information systems by using multiple software layers and the incorporation of a service delivery process. At the moment, cloud computing's current scale, there are different definitions of cloud, all of which are right. Sometimes, a product's value is improved by marketing. Such are the reasons there are many definitions for this term. For example, McKinsey & Company reports that the cloud service model is entirely hardware-based, in that buyers incur no up-front costs because their systems will benefit from an abstracted mix of computing, networking, and storage capabilities. When it is treated as a model in which information is held on the server but retrieved from the internet using a client's browser, Cloud Computing is a design paradigm in which information is still stored on the internet and cached on the user's machine (desktops, entertainment centers, laptops, etc.). People have vastly different responses when you pose the same question to different professions and members of the IT and communications world. In fact, if you do this one simple calculation, you can work out all the answers! Thus, based on the previous definitions and after careful examination, the claim can be substantiated that: Cloud computing is a computing resources are provided on demand, allowing anyone to use it on the Internet from any device at any time.

### 3.1.1 IMPORTANT CONCEPTS RELATED TO CLOUD COMPUTING

A number of words relating to cloud computing are included in this section, which are obligatory and will encourage text monitoring from now on. Virtualizing Virtualization is a crucial technology for the emergence of cloud computing. It's not mandatory, but it's really helpful. Virtualizations involves the development of an operating system, a server, a storage unit, or a network resource virtual version. It involves software such that a physical resource (hardware) can concurrently operate multiple isolated virtual machines with their respective operating systems. Like independent real machines, such virtual machines behave. For the start of these virtual machines, the VMM or so-called hypervisor are used. The Hypervisor is a software that enables several operating systems to isolate and simultaneously share a single hardware. It interacts with the host system to interoperate between operating systems and the visual representation, whether hardware or operating system. Hypervisors include VMWare, Xen, KVM, Microsoft Hyper-V Server, VirtualBox and others.

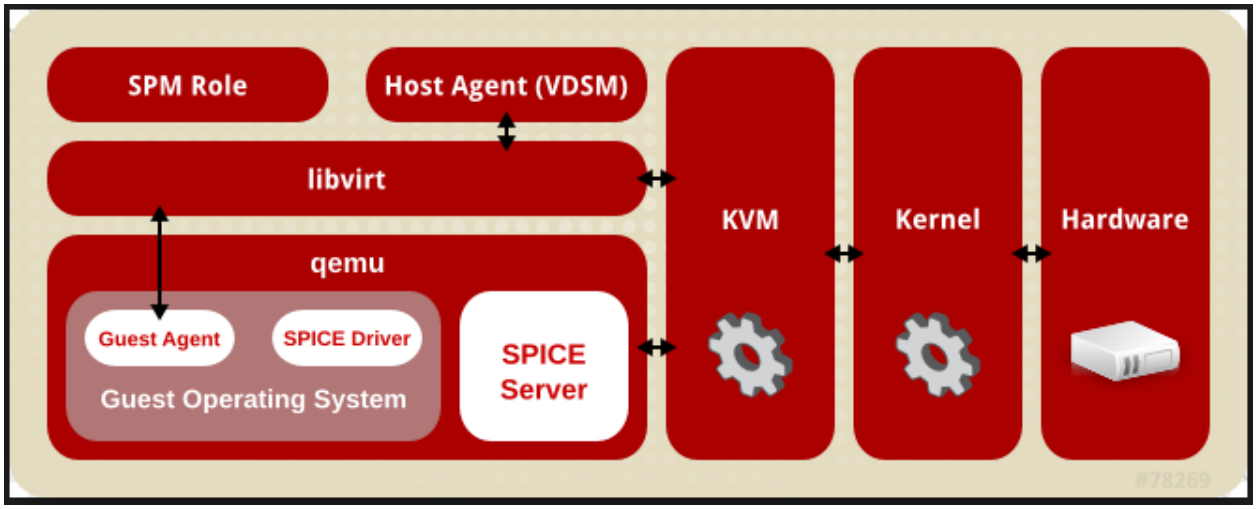


**Figure 3.2:** Data Centers Red Hat Virtualization

Cloud Computing can make the most of resources by being able to manage virtualized environments on demand. This is a characteristic that differentiates it from the traditional computing model, since Cloud environments are scalable and capable of adjusting to fluctuations in demand. As noted in [1], virtualization provides, for example, the homogenization of resources for clients, independence of the physical platform used and the partitioning of physical

resources so as not to fall into restrictions related to the number of machines, while introducing an additional layer of security and isolation. In fact, many of the main advantages provided by the use of cloud-type computing systems are a direct consequence of the virtualization of available resources. Multitenancy This concept is one of the core elements of cloud applications. A multi-tenant application architecture is one where all application users, regardless of the organization to which they belong, share the same code, database, and infrastructure instance. Using this approach, an optimization in the use of resources is achieved and the scalability and performance of the system are improved, since different consumers can share the same physical infrastructure. One of the expected characteristics of virtual runtimes is that they provide the impression of being dedicated environments, equivalent to real environments. When a user makes a request to the cloud, they receive a virtualized item, isolated from the rest, and ready for their exclusive use. Giving access to shared non-isolated elements would allow users to interact with material belonging to others in a trivial way, generating risks related to security.

### 3.1.2 METHODS OF IMPLEMENTATION

There are different types of clouds depending on the needs of companies, the service model offered and how they are deployed in them. Depending on where the applications are installed and which customers can use them, it can be differentiated between public, private or hybrid clouds, each with its advantages and disadvantages. A fourth variant of cloud can also be found called the community cloud.

• Public clouds. The services they offer are located on servers external to the user, and they can access the applications for free or for a fee over the Internet (or in certain situations, through VPNs1). It is understood by three parts, and can be mixed with many functions of different users on servers, storage systems and other cloud infrastructure. End users do not know what other client jobs may be running on the same server, network, disks as their own among the inherent advantages of the implementation, the following can be mentioned: The short period of time for service availability. Provides cloud service provider outsourcing of all core functions of an organization in the case in which the user is part of a company. It makes it possible to take advantage of the infrastructure of the service providers, additionally allowing the ability to expand high and flexible in the adjustment of service dimensions. We prefer to use standard software packages. It has fewer initial bills than the rest of the apps. In addition to this, the

payment of variable public debt bills. According to the principle of payment for use the company information is contained in the public cloud along with the rest of the clients of the provider which, in addition to not being able to locate the information physically, imposes on the provider a number of very challenging data protection and security requirements. The PET therefore shows the regional distribution of the tracer concentration for example, in oncology, 2- (18F) - fluoro-2-deoxy-Dglucose (18FDG), composed of glucose and Fluor-18, is used, since its concentration in tumor cells is a reflection of the increased glycide metabolism in order to maintain a high rate of growth and proliferation [9]. Single Photon Emission Tomography (SPECT) ′. It also requires the administration of a radiotracer, and it is also used as a complement to other techniques, since it allows a more functional analysis. It is very useful in the monitoring of tumors and in the identification of the degree of malignancy. Lumbar puncture. It consists of taking a sample of cerebrospinal fluid to look for tumor cells or signs of infection. Biopsy. It is an essential, basic and necessary test to carry out the definitive diagnosis. It consists of performing a histopathological examination, in which an excised sample of the tumor is analyzed under a microscope. In this way, it is possible to know what type of tumor it is, what type of cells it is composed of, and what its aggressiveness is. All these characteristics have a series of drawbacks associated with them. The main disadvantages of public clouds are thus: The infrastructure is shared with more organizations. Little transparency for the client, since the rest of the services that share resources, storage, etc. are unknown. Reliance on third party security. Depending on the context, this dependency can be an advantage, since if a third party takes care of security, it reduces the work in that aspect of the company.

• Private clouds. They have characteristics similar to the public cloud but on a private network. It serves an organization with its own infrastructure and provides services to that organization or its clients. Businesses get the benefits of IaaS but without some of its disadvantages such as data privacy or speed of access to infrastructure. Since the systems are normally found in the facilities of the customer and typically do not provide third-party services, these platforms are a good choice for businesses that need high data security and service levels. It should be noted that the cloud solution in private clouds can on certain occasions be handled by the provider or a trusted third party. We may quote the following as inherent characteristics of this type of implementation: Short commissioning time and high resource distribution flexibility. In contrast to the public cloud, the contract approach needs an economic investment. It has local systems

and databases linked. It provides the opportunity to benefit from current personnel and investments in already executed information systems. It means more specificity in the solution obtained, since it is structured to adapt to the needs of the contractor. It enables full control of their facilities, processes and corporate knowledge. It allows monitoring and monitoring of the security specifications and safeguards the data stored. Many of these features have a number of disadvantages. Private clouds have the following disadvantages: High cost of content. Dependence on the facilities contracted. Slow return on investment because of its character of internal operation.

• Clouds hybrid. The public and private cloud models are combined. This helps an organization to monitor its key applications by using cloud computing to make sense. It can also be described as a private cloud that can be supplied with resources from other clouds, public, private and community-based. Most importantly, nutrient clouds are still specific entities linked by standardized or proprietary technologies. This technology enables portability of data and application such as load re-equilibrium between clouds. One use of this form of company is the opportunity to build up the ability of a private system to meet peak demand at particular times, by purchasing infrastructure from a public cloud provider. An organization using this form of implementation may take advantage of the advantages of each type of cloud, with a number of additional characteristics as shown below: It provides more versatility in cloud service provision. YOU retain more leverage over company and data services. As detailed above, rapid commissioning is done with a hybrid cloud approach. It involves more difficulty in integrating the cloud solution since it is a solution consisting of two distinct ways of incorporating cloud services. It enables the best features of the two types of cloud deployment to be integrated with respect to data control and management of the company's basic functions. It enables the supplier to choose a scalable and versatile infrastructure that makes the solution highly agile. It enables the internal management of the entity's cloud resources.

• Community clouds. Community clouds. These are clouds used by various organizations with common functions and services which enable collaboration between stakeholders. If a group cloud is analyzed, its strengths and disadvantages are in theory between the privacy and the public. The number of resources available in a shared cloud is generally greater than in the private cloud, with the obvious elasticity benefits that this entails. However, the amount of resources available in a public cloud solution is smaller than those available, reducing the

elasticity of the cloud. On the other hand, the number in users of such a cloud is smaller than the public cloud, which offers better protection and privacy benefits3. The Healthcare Community Cloud for easier access to essential health-care information and applications and Government Community Cloud for easier access between public authorities and public administrations to interoperability services. Like other clouds, group clouds have many benefits, for example: Internal policy enforcement. Reduction of costs by infrastructure and services sharing. Quick investment return. Quick return on investment. However, this cloud has strong technology disadvantages, as the contracted infrastructure is obviously dependent on all the problems that this entails (mainly security). Cloud computing is a very general term that is used in various ways, as noted in the previous section. However, the three major service models (in decreasing order) seem to be agreed: Software as a service, software as a service, platform as a service or platform as a service, and infrastructure as a service or infrastructure (IaaS). Each of these models specifies how the cloud provider provides its service. It is important to remember that the services of the underlying layers may be used in each modality or layer.

• Service as a software, SaaS. It is a software delivery model in which an application is delivered as an Internet service. It enables the final application to become completely scalable both in terms of number of users and storage requirements within an on-demand infrastructure. Instead of downloading and managing the software, consumers are easily accessible via the network and free from complicated device management. In addition to supplying the services required for the use of such information, the organization hosting the program is responsible for preserving customer information. This eliminates software and hardware costs as well as maintenance and operational costs. The service provider controls security considerations. Only editing preferences and restricted administrative rights are available to the subscriber. The offer of this service model is currently relatively broad and diverse, but it stands out: Salesforce, SAP Business by Design, Google Apps, Windows Live and Mobile Me.

• Service platform, Pass. This model involves providing a series of computer platforms to design, test, deploy, host and maintain the operating systems and applications of a customer or subscriber. Though sometimes referred to as SaaS evolution, it is more of a model that provides everything needed to sustain the entire life cycle of applications and web services completely accessible on the internet. The subscriber never controls the virtual machines or attacks the device, but uses the software APIs of each cloud and its languages of programming instead. The

service subscriber can therefore be verified as having partial control over the applications and surroundings with guration since the environment would rely on the infrastructure implemented by the service provider. Security between the service provider and the subscriber is shared. This service model therefore makes it easy to deploy customer applications without expense and complexity in the purchase and management of hardware and related software layers. Today, Force.com, Google App Engine and Microsoft Azure are the key providers in this business model.

• Service infrastructure, IaaS. It is a model in which a web-based service is operated by the provider as a simple computing infrastructure (servers, disk spaces, software and networking equipment), in which environment can be built to build, run or test apps. In this model, many customers coexist with the same infrastructure and you pay for what you use. The main aim of this model is to prevent subscribers' purchases of services, as the provider provides these resources as virtual objects accessible through a service interface. In general, the subscriber retains the decision-making control of the operating system and the environment. Security management is also mainly the responsibility of the subscriber. Instead of purchasing or directly equipping resources such as servers, data center space and network equipment, clients opt for this form of cloud family opt for outsourcing for savings in investments in YAU systems. The factors of this outsourcing are based on the amount of services used by the customer, which is based on the pay-per-use model [2]. Amazon Web Services (Amazon EC2, Amazon S3 or Amazon Elastic Map Reduce, among others), Gogrid and Mosso are some of IaaS's most relevant providers (from Rackspace) [3].

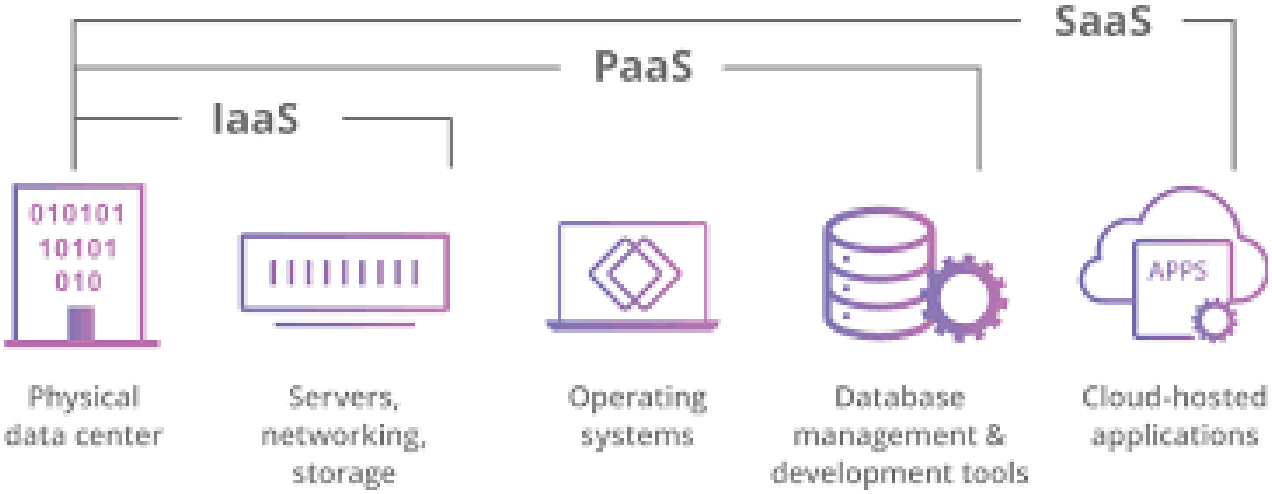


**Figure 3.3:** SaaS Model

### 3.1.3 ADVANTAGES AND DISADVANTAGES

In order to analyze the Cloud Computing paradigm more deeply, it is necessary to highlight the advantages and disadvantages that Cloud Computing offers compared to other more traditional computing systems. The main advantages coincide with those mentioned in each of the types of clouds, but can be summarized as follows:

• Low cost. It does not require implementing, updating or repairing technological infrastructures (data centers, local servers, networks, etc.).

Everything is delivered by the Cloud Computing provider, including software maintenance (in the case of SaaS). The economic savings are produced, especially in the initial investment and in the infrastructure. In the case in which the consumer is a company, since it is not necessary to worry about the maintenance of the servers, purchase of hardware, updates, etc. it is possible to maintain a greater dedication to the main focus of the business.

• Savings in space. By not investing in infrastructure to access through the Internet, it is not necessary to have a place to put the servers.

• Scalability. When working with the On-Demand concept (on demand), an increase or decrease in resources assigned to the account is requested at any time, according to the user's real needs. Therefore, companies can transparently scale their services by adapting to the needs of their clients quickly and on demand.

• Security. Each Cloud Computing plan has high reliability logical and physical firewalls, delivered by the provider so that the systems cannot be compromised. In the previous section, it has been commented that responsibilities regarding security depend on the service model (SaaS, PaaS or IaaS). However, there is still a great discussion today about whether or not the cloud is more secure than traditional models.

• Speed. The high-speed access will make the use of Cloud Computing a pleasant experience for its users.

• Accessibility. When accessed through the Internet, you can work and request cloud resources from anywhere.

• Stability. The uptime is an important aspect for the Web sites and applications of the companies that contract the Cloud services. With the resource distribution system, this uptime increases signi cantly, because if one machine has a fault, the others will continue to serve the applications and files. By using resource management optimization mechanisms (such as virtualization and load balancing) it is possible to make the most of available resources and better cope with adverse situations, such as demand spikes. Next, a series of disadvantages or disadvantages of this computer system are shown4:

• Privacy & Security. One of the biggest fears that Cloud Computing raises is the loss of privacy of the data that circulates through it. By having to store them on third-party servers, along with data from other clients, the mistrust arises that someone can see your information and use it for unethical purposes.

• Breach of data protection regulations. Most of the companies that offer Cloud Computing services do not have data centers in our country, some even have them located outside Europe. This is a problem when the data cannot be stored outside our borders for reasons related to current laws.

• Increased latency. It is not an obstacle in many applications, but there are cases in which this feature can impair the correct operation of the system.

• Loss of control over stored data. As they are not hosted on their own systems, the control over the data is less.

• Intellectual property or other conflicts. The customer information, not being within the company's infrastructure, it is possible that problems arise about who this information belongs to.

• With denasality in the transmission of data through Internet connections. New, more secure and efficient encryption systems are needed.

• Network and provider dependency. If the service fails or the internet connection cannot work. In addition, if the provider went bankrupt, the entire company infrastructure and all customer data would be lost.

• Beware of automatic scalability. Cloud computing is susceptible to DDoS (Distributed Denial of Service) attacks. This attack consists of someone with bad intentions attacking an application or any website with the intention of collapsing it, launching requests from multiple terminals concurrently and in bulk. In the end, the service drops as it is not able to satisfy all the requested requests.

• Captive client problem. It is difficult to import the information and applications of a customer, from one provider to another,

Since each one has different characteristics and offers different APIs. From the hardware point of view, Cloud Computing incorporates three new aspects with respect to traditional computing systems [3]:

• The illusion of unlimited availability of on-demand computing resources. This eliminates the need for Cloud Computing users to plan their access to the particular service in advance.

• The elimination of a previous commitment on the part of the Cloud users, allowing companies to start small and increase the number of hardware resources only when required.

• The ability to pay for the use of computing resources in the short term, as needed (for example, hourly processes or daily storage) and release those resources when necessary. In this way, cost savings are achieved by releasing the physical machines when the user leaves them and they become useless. With all that has been said about cloud-type computing systems, it is not daring to affirm that the use of Cloud Computing has no limits. It is currently being used primarily to:

• Internet services (hosting of sites and applications).

• Business management software (companies host their applications in their Cloud Computing account and employees connect to them directly for their daily performance).

• Commercial software. [11].

## 3.2 SERVICE LEVEL AGREEMENT (SLA)

Since the trend of consumers is towards the adoption of a service-oriented architecture, the quality and reliability of the services become fundamental aspects. However, consumer demands vary significantly, depending on a multitude of factors. Since it is impossible to meet all consumer expectations by the service provider, there must be a balance between both parties. This balance is achieved through a negotiation process. At the end of the negotiation process, the supplier and the consumer commit to an agreement. It is important to note that both parties do not always agree, so the negotiation process may involve several messages between them, until both the consumer and the service provider reach an agreement in which the needs and demands of both are satisfied. , clearly defining the obligations of each of the parties. Another aspect that this type of agreement usually covers are compensation in the event of breach of contract. The commercial use of web services, and in general in service-oriented environments, is usually regulated by what is known as service level agreements (SLAs). According to H. Ludwig et al. An SLA contract includes the obligations of each of the parties, the methods to measure their degree of compliance, their billing model and the penalties in case of violation of the terms of the contract [4]. In short, an SLA is a legally binding contract between a provider and a user, which defines the terms of the agreement between both parties at the service level, specifying obligations, responsibilities, guarantees and compensation. While this contract can be defined in any way as long as clients are able to understand and accept it,


**Figure 3.4:** SLA Service Level Agreements SLAs

It is commonly offered simply as a text document that is accessible to the parties. Currently, cloud providers offer a very basic view of SLAs for their clients. As an obligation, compensation is established, usually in the form of a billing discount, to users in the event of contract violation. Other candidate aspects to be part of a service agreement are CPU or memory requirements, geographic restrictions, etc. Using an SLA, the cloud itself is in charge of carrying out operations aimed at fulfilling contracts. In addition, the cloud can also perform SLA health checks, determine if their terms have been violated, take corrective actions, notify the user of the incident, and execute appropriate compensation procedures. It is clear that having an agreement between the parties involved in a service-oriented environment is highly beneficial for both the consumer and the service provider. For consumers, the SLA has the following advantages: • Determining if a service provider is meeting your expectations. • Offer real statistics to compare with the SLA. • Indicate the days and hours in which the SLA was not met. • Document when the SLA was not met to secure the offsets specified in the SLA contract. For Service Providers, the SLA report has the following advantages: • Show your customers that you really care about uptime [16].

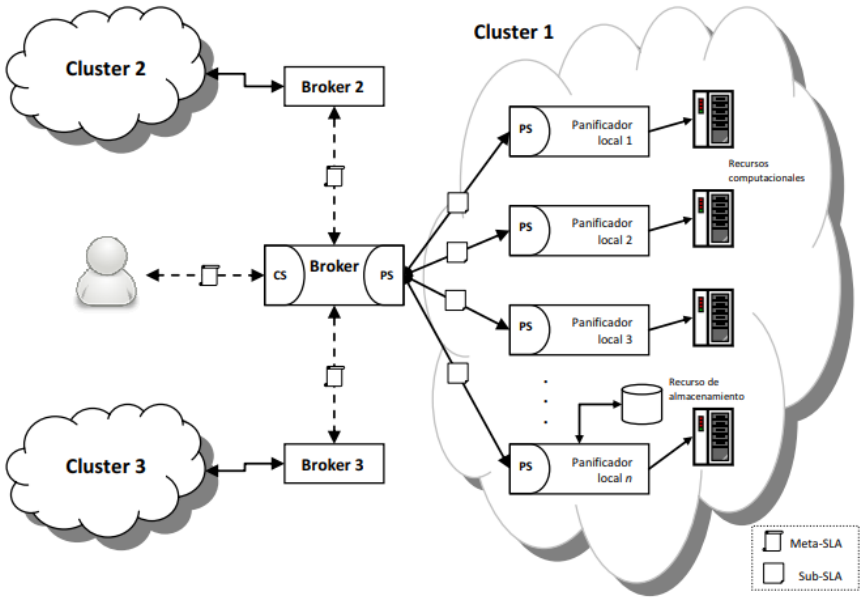


**Figure 3.5:** Machine Learning Scheme

### 3.2.1 SLA ARCHITECTURE

In Cloud Computing systems based on SLA it is common to distinguish between one or more parties that provide service and one or more parties that consume those services. As already mentioned, the SLA agreement is negotiated between a customer and a supplier, and contains information on the level of service agreed between the two parties. In this project, a Cloud Computing type system has been considered with the elements (or parts) associated with the SLA agreements, following an approach similar to that described in [5] and [9]. Said models distinguish, in general, four roles and three possible elements that are part of the negotiations of SLA agreements in Cloud Computing systems. In gure 1 you can see a possible architecture that represents the environment in which this project is working, with the elements and the roles they assume in each case. The three elements that participate in the negotiation are the following:

• User: must negotiate the SLA contract and agree with the broker regarding the terms of the SLA. It can be considered as the end consumer of the cloud resource.

• Broker (or super scheduler): is in charge of negotiating and validating the SLA contract with the end user. These SLAs can be mapped into one or more SLAs which will later be negotiated and validated by local resources, and more specifically, by their planners.

• Local planner: is responsible for planning the work associated with the SLA that has been previously negotiated and validated (the restrictions associated with said SLA can be stored locally on the resource itself, in some type of registry).

As previously mentioned, a single SLA contract negotiated and agreed between the end user and the broker, can be translated into multiple SLAs between the broker and several different resources to serve the user's request. In fact, in this project, generally a request (with its respective SLA agreement) ends up leading to several communications between the broker and the local planners. In this way, a service request from a user to the cloud causes several tasks to be executed on several different resources within the cloud itself. In such a case, the user can decide to set the constraints of a given workflow as an integer, and the broker translate it into several specie c SLAs for individual tasks, which will be addressed to various local planners. To indicate the possible differences between these two types of SLA agreements, the terms meta-SLA and sub-SLA are used in [9]. It should be noted that in this project the SLA agreements

with which we work are those that appear in the gore as sub-SLA, that is, they do not correspond to the contracts that occur between the broker and the end user, or between the different brokers. It is interesting to note that the main objective is to illustrate, in a general way, the different interactions that can occur as a result of the use of SLAs to agree on the expected level of service when a certain task is requested. There are other possible architectures, or variants of the one presented here, but it is beyond the scope of this project to deepen this aspect. For example, a possible variant would be to establish a closer relationship between a broker and a specific local planner. In that case, the broker can exercise greater control over the local planner, so that no SLA agreement between the two would be necessary. There are also dynamic architectures in which, over time, certain links are established between a user and a specific broker.

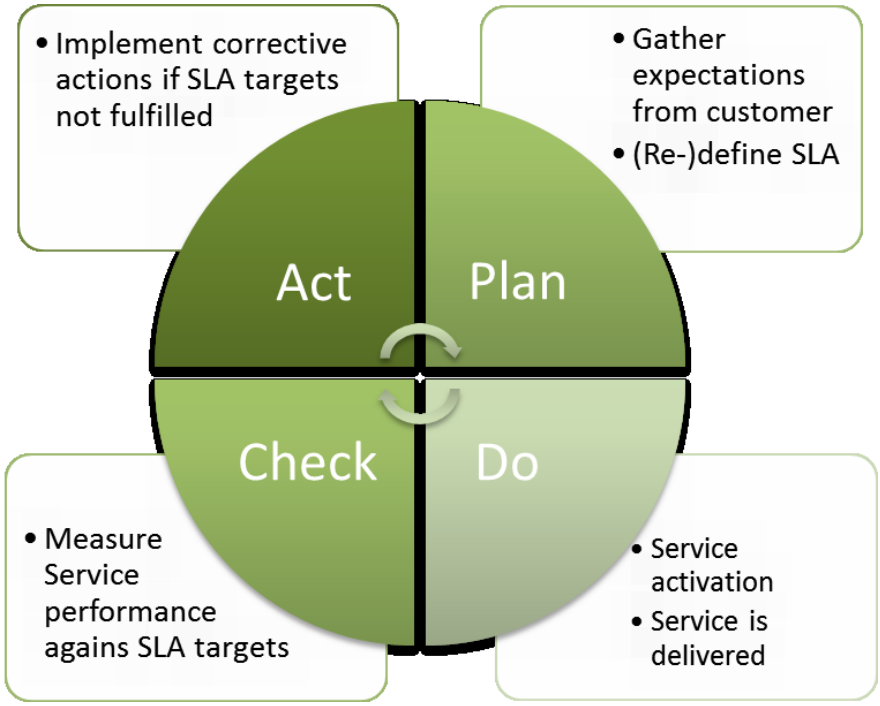


**Figure 3.6:** Service Level Management itil

These links are based on the nitty that has been forged between the two. Consequently, the more satisfied a user has been with a certain broker in the past, the more an affinity there will be between them, and the more likely that user will establish connections with that broker to access the cloud services. Each of these elements of the system exposed previously can adopt a certain role within the negotiation of the SLA. The different roles that the parties to the negotiation can take are the following:

• Cloud Service Consumer (CSN): it is the part of the negotiation that issues a request for cloud service and thus triggers the creation and execution of the requested service. At a conceptual level, a CSN can be a single user or a set of them.

• Cloud Service Integrator (ISN): it is the part that offers composite services from the cloud for the Cloud Service Consumers. For this purpose, the ISN must identify, request, negotiate and encompass services from various Cloud Service Providers. In general, both consumer and provider can assume the role of ISN.

• Cloud Service Provider (PSN): it is the part of the negotiation that is responsible for providing a service (either atomic or composite) from the cloud to either the Cloud Service Integrator, or the directly to the Cloud Service Consumer.

• Cloud Service Agent (ASN): this role can only be assumed by the broker, and can be considered as a third party of trust that combines the set of requirements of the ISN (which in this case takes the role of consumer) with a list or set of possible services from various PSNs, based on their functionality and quality of service (QoS).

Therefore, it could be said that the ASN is the mediator in the pairing process between the ISN and the PSNs. Given a certain level of service, the ASN will propose those services that respond to that level of service. This project has been based on this task of the broker. Once the different elements that are part of the system architecture have been exposed, the importance of resource planning and management is evident, especially in high-performance environments such as the cloud. The existence of efficient planning algorithms could be of crucial importance when estimating the capacity of the system, and increase the probability of acceptance of a given request. Many of these algorithms are based on priority queues or rewards and financial savings. However, this project has focused on developing resource planning and management algorithms based on cost savings. Service level objects and focuses primarily on the description and monitoring of quality agreements. WS-Agreement [12] is a recommendation proposed by the Object Grid Forum working group, which provides a structure for describing agreement offers and a protocol for obtaining service level agreements. WSAgreement has been widely accepted, and that is why for this project, it has been decided to take into account the specifications proposed by this web services protocol. Its design has been heavily in fenced by WSLA. WSLA

(Web Service Level Agreement) was proposed by Ludwig et al. [11], and allows customers and suppliers to describe service level objectives and focuses primarily on the description and monitoring of quality agreements. The conceptual map to describe agreement offers in WS-Agreement, taken as a reference for a basic agreement model. In this recommendation, a settlement offer has two clearly differentiated parts: the context and the terms. In turn, the terms can be distinguished between those that describe the services that are the object of the agreement (terms of service) and those that describe the obligations regarding said services (terms of the guarantee). Figure 3 shows an example of two settlement offers. The agreement that could finally be established after following the corresponding negotiation process. In WS-Agreement, agreements follow a similar scheme to agreement offers5. In this case, both the agreement and the agreement offers are bilateral, that is, they include obligations from both parties. It must be said that it is also common in many proposals that settlement offers by customers only require obligations from suppliers, while settlement offers by suppliers only guarantee obligations to customers. These are called unilateral. It should be added that, in contexts other than WS-Agreement, agreement offers proposed by customers are also known as demands, while agreement offers proposed by providers are also known simply as offers. Once all the clauses of the contract have been established, the resource planner comes into play, using the efficient optimization algorithms presented in this project. There are numerous models to describe agreements in general and description of obligations in particular. Mainly, specific languages based on XML have emerged that have had a great impact on the community, among which WSLA (Web Service Level Agreement) stands out, which was proposed by H. Ludwig et al. in [10] and [11]. This language allows customers and suppliers to describe [19].

### 3.3 MACHINE LEARNING

Machine Learning is a branch of Artificial Intelligence that is responsible for the design and development of algorithms that allow a computer to improve behavior automatically through experience. That is why it is called Machine Learning or Machine Learning, since a machine is capable of learning on its own through empirical data. More specifically, an ML algorithm is able to extract common characteristics and patterns from a data set (examples or training data), and apply them to new data sets (test data). This discipline is used in different fields, such as medical diagnostics, market analysis, robotics, banking, speech and written language

recognition, image pattern detection, DNA sequence classification, or marketing. There are different types of algorithms classified based on their output. The most significant distinction would be between supervised learning and unsupervised learning. There are more types of algorithms, such as semi-supervised learning, reinforcement, but for the scope of this project they are not relevant. In this type of learning, a function is deduced from a set of training data that is already correctly labeled, hence it is called supervised. In these algorithms a set of training data is provided in the form of a vector. The algorithm, after analyzing the input vectors, infers a function known as a classifier or regression function depending on whether the output is discrete or continuous. The inferred function must be able to predict the class for any input vector. First the training phase is carried out. In this phase, the relevant attributes for the classifier are extracted from a corpus. These attributes will have been previously classified

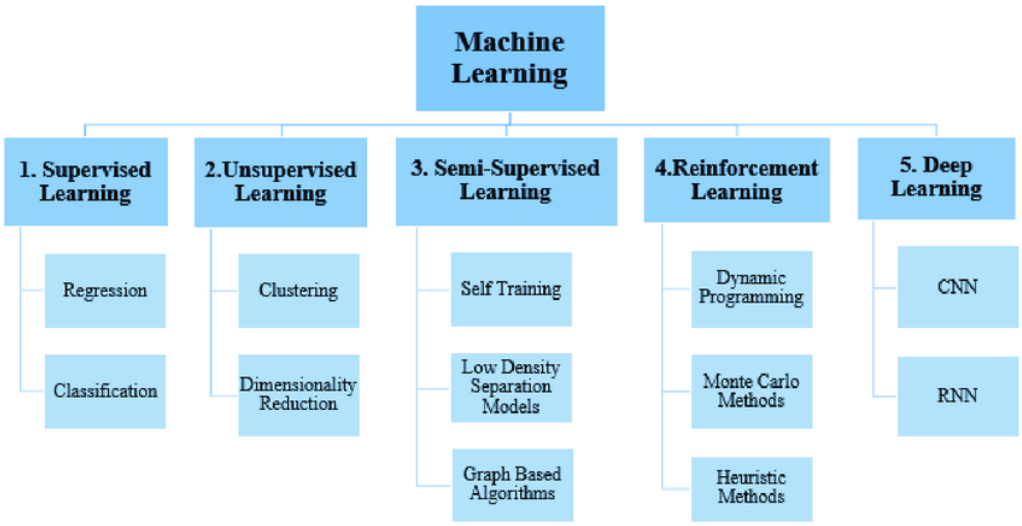


**Figure 3.7:** SLA Architecture

So, their class is already known a priori. From the extraction of attributes comes a set of vectors that will form the input to the algorithm, which from the data and its classes will generate a model that generalizes the characteristics extracted from the corpus. In the next phase, the prediction phase, the same process of extracting attributes will be carried out to generate a vector, but we will no longer know the class of these examples. The algorithm, as it will have already been able to learn a model, will be able to apply it to the data and thus predict the classes of these examples. In this type of learning we try to determine the structure of the data, that is, we try to group large quantities of data based on the characteristics they share. It receives that

name because nothing a priori is known. It is a way of finding hierarchy and order in an unstructured collection of data.

These groups can be seen as a number of elements which in some respects are identical but different from the elements of other groups. Clustering is one of the key strategies of unattended learning. It is typically expressed by the view, as shown in the following example, of which elements form groups on an axis (x, y). Decision-based learning is based on the concept of division and law. These algorithms create decision-making bodies, which are graphic models which have been recurrently constructed by selecting a node attribute and generating a branch of any possible value of that node. For each value, another attribute and branches with its values are re-established until the final nodes corresponding to the classes are reached. Decision trees do well with large data sizes, since not all data have to be loaded into the memory at once. The measurement time scales well with the number of columns increasing linearly. The benefits of decision-making trees are to be understandable and interpreted easily, the generation of rules is easy, the complexity of the problem is reduced and the time for training is not long. Some of the downsides are that if a high-level error was made, subsequent nodes would be poorly formed. The most difficult thing in building a decision tree is to decide on which attribute a node is based, so if there were many attributes, the algorithm would have many training data choices, and a model would build up to generate new ones. Examples. Examples. This problem is called over fitting and sometimes causes several features in training data to fail. However, when combined with Ensemble approaches, decision trees can function well. Instead of studying a single model, these methods learn many and combine estimates of each model. Bayesian networks are a probabilistic model, represented by a directed acyclic graph, a graph that does not have cycles, i.e. for each vertex there is no direct path that starts at the same vertex and ends there. In these graphs, the nodes represent the attributes and the arcs depend on the attributes. Each node has an associated probability function, which is used to predict each instance's probability. The values of the parent nodes shall be used in each node, and the corresponding likelihood for each value of the attribute represented by the node shall be represented. Given this model, it would be possible to identify a new instance by evaluating the likelihood of new cases based on the variables of known instances in Bayesian. For Bayesian Networks a learning algorithm must be created, defining two components: a function to evaluate a given data-based network and a method to search the space of possible networks. In the fields of bioinformatics,

document classification, information recollection, image processing, policy structures, games or marketing, Bayesian networks are used to model awareness.

### 3.3.1 PROBLEM OF OPTIMIZATION

Many realistic optimization problems pursue the best configuration of a number of variables in order to achieve certain goals. These issues can be subdivided into two categories: those which encode solutions with real variables and those encode solutions with discrete variables. Among the latter are the problems class called Combinatory Optimization (OC) (An object is sought in OC problems from a finite (or possibly infinite) set of objects. This object is usually a whole, a subset, a permutation, or a graph Definition: Definition: The following are: • a set of variables X = {{x1, ..., xn}; • domains of variable D1..., Dn; • limitations between variables; • objective f function to reduce, where f: D1 x · x Dn -> The following is defined: R+ R + The set of all possible assignments shall be S = {s= (x1, v1), (xn, vn)} | vi {s} S is the search space (or solutions), and the candidate solution is each member of this set. In order to address a problem of combinatorial optimization, a solution s* = S must be found with the minimum function value, i.e. f(s*) – f(s) – f(s) for all s* – S*.

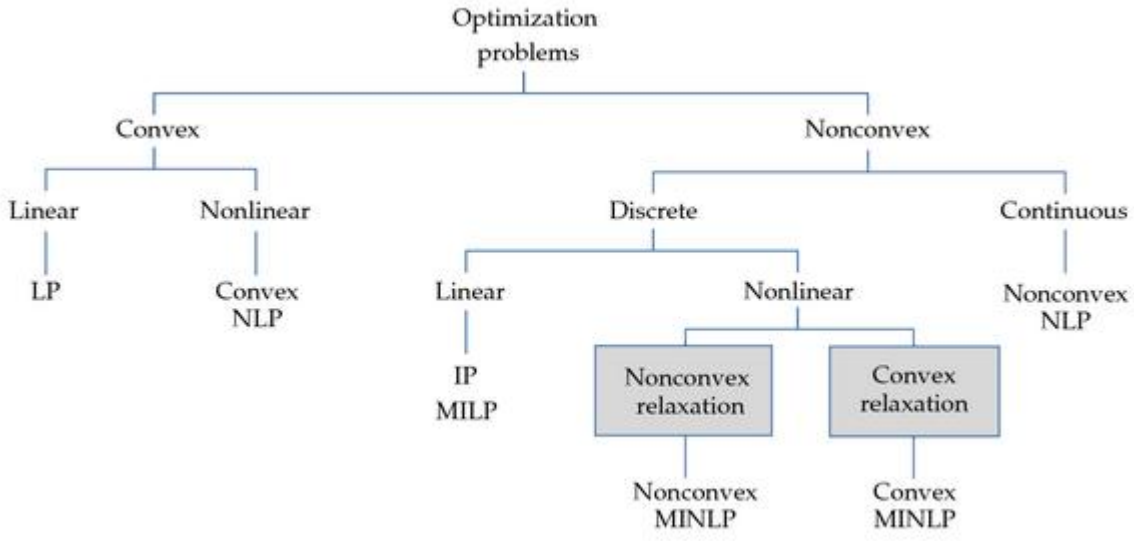


**Figure 3.8:** Problem Optimization

The responsibility of the Optimizer component is to solve the combinatorial optimization problems received by the AMBÚ SaaS platform. The general idea is that the I / O component receives the problem, and the Router component assigns it to an Optimizer component, which solves them, and the I / O system delivers the solution to the customer. The Optimizer component can solve a certain type of problem by several different techniques and can use the computing resources available to you. The resolution techniques can be linear programming and metaheuristics. The computing resources are the NVIDIA graphics cards and the hardware threads of each Optimizer. An optimizer can be associated with a computer, but it is not necessary. If we associate an Optimizer to a quad core computer with hyper-threading, we will say that this Optimizer has 8 threads. If we associate an Optimizer with a server with four Xeon processors and two Tesla graphics cards, we will say that the Optimizer has 16 threads and two graphics cards. That same server can be associated with an Optimizer with 6 threads and a graphics card, and another Optimizer with 8 threads and a graphics card, leaving two for the operating system and integration with the rest of the platform. With this approach, different configurations can be created to make the calculation network more efficient. The architecture of an Optimizer. An optimizer component has a series of executable, which are the ones that really solve the problems. An Optimizer can have some executable and an optimizer can have other different executable. This structure offers a lot of configuration versatility in the AMBÚ service network.

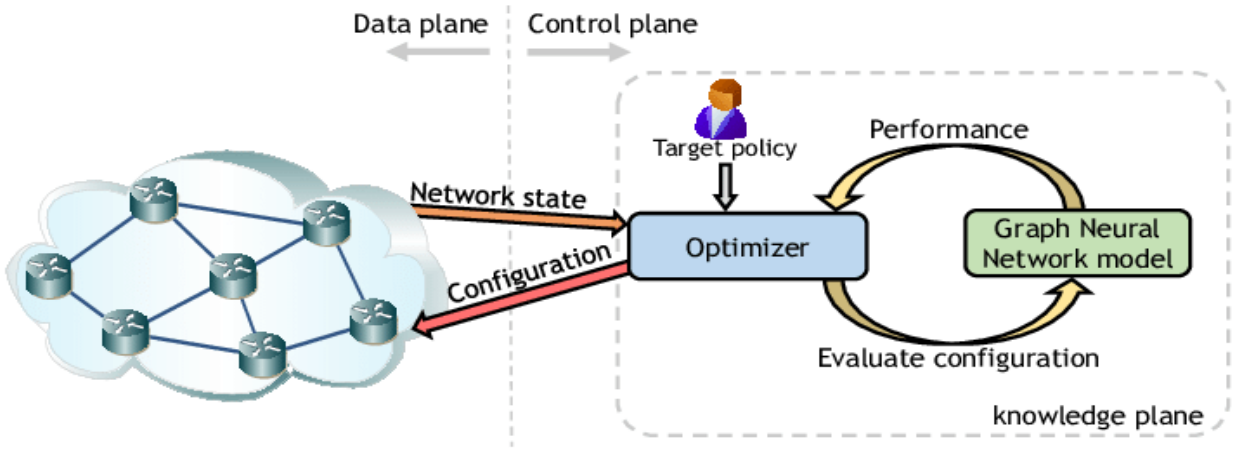


**Figure 3.9:** Network Optimization and gcn

For example, you can configure an optimizer with executable that use graphics cards and install it on a server with graphics cards, but a general-purpose Optimizer can coexist on that server, or all executable that solve problems can be associated with an Optimizer by linear programming, and install it on the server that has licenses for a specific linear programming solver. There is an executable for each resolution technique, for example, to resolve the FAP, the following executable may exist:

• Linear programming

• Ant colonies for multi-core

• Genetic algorithms for multi-core

• Genetic algorithms for NVIDIA cards

The router assigns problems to an Optimizer, and tells it the technique to use to solve each one. Depending on the resolution technique assigned to each problem, the Optimizer selects the executable that solves that type of problem with that technique. An Optimizer can launch several executable in parallel, and detects when the resolution of a problem ends, to launch the next one. The communication between the Optimizer and the executable is done through four XML files, which are indicated as parameters in the invocation of the executable. The four files are the following:

• Configuration: the value of execution parameters is indicated, such as the number of threads that can be used and the maximum time that can be used to solve the problem.

• Input: this file contains the data of the problem to be solved. For example, if the problem is the FAP, this file contains the description of the routes and the types of vehicle.

• Output: this file is created by the executable, and contains the solution to the problem. For example, if the problem is FAP, the output file contains trip times, vehicle type assignments to trips, and vehicle rotations.

• Log: it is a trace file that indicates the sequence of actions that the executable has been developing in solving the problem. It is useful to check the correct execution of the algorithms, and especially to detect errors. The optimizers are registered in the router when they start their

execution, indicating the computing resources available. From then on, the router can assign problems to you for resolution.

### 3.3.2 PROBLEM OF RESOURCE ALLOCATION

One type of ACO problem is optimization problems in resource allocation. The most studied resource allocation problem is the traveling salesman problem (TSP), which consists of creating the route of a traveling salesman, who leaves his home, seen once by all his clients, located in different locations, and finally comes home. Depending on how you order the visits to the clients, the traveler will travel more or less distance. The objective of the traveler is to travel the minimum distance to make all the visits. A widely applied resource allocation problem is the vehicle routing problem (VRP). In this problem, you have a fleet of vehicles that must make a series of deliveries to customers. Vehicles have a maximum load capacity. Depending on the deliveries assigned to each vehicle and the route each one takes, all deliveries may be made with a different number of vehicles and with a different total distance traveled. The objective of the fleet manager is to make all deliveries with a minimum cost (or minimum total distance traveled) the fleet assignment problem (FAP) of a passenger transport company, consists of calculating the time of departure of the journeys, and define the rotations (sequence of journeys ordered in time) carried out by each type of vehicle. Depending on the schedule of the journeys, the assignment of each journey to a type of vehicle, and the sequence of journeys made by each vehicle, different transport operations can be obtained. The goal of this problem is to maximize profit or minimize costs. A very important type of resource allocation problem is the work force problem (WFP), which consists of defining the shifts of a staff to cover a workload. Depending on the start of the shifts, the number of people assigned to each shift, and the rest days of each person, different personnel quadrants will be obtained. The objectives are to minimize staff costs and maximize the quality of worker shifts. Some, but not all, of the resource allocation problems have been listed in order to show their nature: there are many ways to allocate resources to activities, some are better than others, and the goal is to find the best one. This search for the best solution is not an easy task, and requires the use of appropriate techniques. Due to the practical importance of these combinatorial optimization problems, many types of algorithms have been developed to solve them. These algorithms can be classified as exact or approximate algorithms Exact algorithms guarantee that they find an optimal solution for a finite instance of an OC

problem in a delimited time Many OC problems are NP-hard, which means that there is no algorithm that solves them in polynomial time in these cases, exact methods may take exponentially growing time, leading to computation times that are too long for practical applications, and approximate methods are necessary. A widely used methodology when solving a combinatorial optimization problem is to use an exact technique to assess the size of the problem that can be solved, and then propose an approximate technique if the exact technique cannot solve the problem instances to be solved.

# 4. PROPOSED METHOD

It is clear that in order to design an algorithm that can reduce the cost of the system, it is necessary to go through a series of steps first. Throughout each of these steps or phases, a series of increasingly complex methods have been implemented to arrive at the final design of the algorithms proposed in this document. This chapter details the evolution of the algorithms that have been used in the project. Before detailing each of the methods implemented in this project, it is necessary to remember that each and every one of the methods has two phases: estimation and assignment. In the first phase, an estimation of the planning of the system resources is carried out. The procedures described in this chapter correspond only to this estimation phase. Once the estimate has been made, an allocation process occurs, in which the resource resulting in the estimate is assigned or not, depending on whether or not it meets any of the two basic requirements implemented in this project. The first of these requirements is related to the availability of the service in question. Specifically, if the assignment of a specific resource to a particular request causes the service to exceed its availability, the request will be discarded. The second restriction that the allocation of a resource to an incoming request must meet is that the maximum delay requested by the user must not be exceeded. Therefore, if the fact that a certain server provides service to a certain request implies a breach of any of these two requirements, the request is automatically discarded.

## 4.1 PLANNING SCHEME

In the first place, the scheme presented in chapter 2 has been proposed by means of some methods that have been called elementary methods. These methods are the simplest as they simply allocate resources without considering any elaborate optimization criteria. It should be noted that these methods have been developed in a chronological and evolutionary way, that is, in order to design an elementary method, the previous elementary method has been designed first. The methods from which the rest of the methods have been designed and developed to arrive at the hybrid resource management and allocation mechanisms proposed in this document are discussed below. This first method is the simplest of all. Basically it is a direct allocation of system resources in a sequential way. Once the requests have been ordered according to their priority value, each request is assigned as many servers as CPUs have been requested by the user. In this method, no time criteria or restrictions are taken into account. In other words, the

availability of the service in question, or the load of a specific service or server, is not taken into account. Furthermore, the assignment is carried out independently of the temporary values of the request (start time, end time and task time). Another important characteristic of this method is that the task is only carried out by one server, and the rest of the assigned servers are reserved. Therefore, the first server that has been assigned to the request will be tD seconds in full mode and the rest of the time in idle mode, while the rest of the assigned servers, being reserved to offer service to said request, will be in idle mode the tTOT seconds. When calculating the cost of each request using this method, it is necessary to calculate the total time in which the servers involved are in each mode. From the previous explanation, the total times in full mode and idle mode can be calculated for each request, so following the nature of the method itself, it is obtained that:

$$T_{FULL} = t_D$$

$$T_{IDLE} = t_S + t_{LAX} + t_{TOT} \cdot N_{CPU} \text{ ó } T_{IDLE} = t_D + (t_S + t_{LAX}) \cdot N_{CPU}$$

A limitation of the algorithm corresponding to this method is that if it finds a node where the request cannot be inserted, it does not go to the next node, but discards the request directly. As expected, since in this method resources are allocated sequentially, no server is released in the entire process, with the cost savings that this would entail. The elementary method II is an evolution of the elementary method I. operation is similar to the first, but in this method there is a distribution of the task time between several CPUs. That is, this algorithm assigns as many CPUs as the user has requested, also sequentially, without attending to restrictions or optimization criteria or user requirements. Therefore, the only difference between this method and the previous one is that in this case, the task time is distributed among all the CPUs assigned to the request in question. As a consequence, the server will be in full mode only one nth of the total task time (tD / NCP U, where NCP U is the number of nodes assigned to the request), and it will be in idle mode the rest of the time. It is important to note that the rest of the time that had been allocated to homework time is now part of the lax time. Therefore, the new laxity time is h NCP U −1 NCP U · tD + tLAXi. This fact will be of special relevance in the basic method I and the two hybrid methods proposed in this project. Because of this, and for convenience, the procedure

for distributing the task time among the assigned CPUs has been used in all successive methods. In this case, the times in idle and full mode for each processed request. This is the last elementary method that has been designed in this project. The elementary method II has been taken as a starting point, and a temporal optimization criterion has been added. It is the only elementary method that has a non-sequential selection procedure. For each request processed, the algorithm performs a search for the server with a lower accumulated workload of the service corresponding to the request, and assigns it to the same. This process is repeated for each requested CPU in each case. Therefore, the algorithm performs as many searches as there are nodes requested by the user. As already mentioned, this method is based on the previous elementary method, so it also performs a task time distribution for each assigned server. Therefore, it can be said that the server will be in full mode and in idle mode for exactly the same time as the elementary method II. The only difference is in the selection of the assigned node. By performing a search for the server with a lower accumulated workload, and therefore with a shorter time in the active state, the number of accepted requests is increased, thus improving the system's performance. Due to the characteristics of this method, the resulting times in idle and full mode can be defined in the same way as in the previous method, that is,

$$T_{FULL} = \frac{t_D}{N_{CPU}} \cdot N_{CPU} = t_D$$

$$T_{IDLE} = \left[ t_S + t_{LAX} + t_D \cdot \frac{N_{CPU}-1}{N_{CPU}} \right] \cdot N_{CPU}$$

*ó*

$$T_{IDLE} = \left[ t_{TOT} - \frac{t_D}{N_{CPU}} \right] \cdot N_{CPU} = t_{TOT} \cdot N_{CPU} - t_D$$

**Figure 4.1:** Department of Transportation

## 4.2 ACO FRAMEWORK

ACO was born as a resolution technique for the traveling salesman problem (Traveling Salesman Problem, TSP), in which a merchant must visit a series of clients located in different cities, making the shortest route, so that leaving from their town of residence, visit all clients once, and return to your location. The TSP is one of the most studied combinatorial optimization problems in recent decades The VRP and the TSP are closely related. Once the VRP customers are assigned to the vehicles, the VRP is reduced to several TSPs. The VRP is more difficult to solve than TSP, because the VRP consists of two problems: the allocation problem and the TSP apply the Ant System (a variant of ACO) to solve the VRP in its most basic form (limited capacity of vehicles, a central warehouse, and identical vehicles). They feature a hybrid Ant System algorithm along with the 2- opt heuristic to enhance individual vehicle routes, once the ants have built the routes. Improve on the previous system by using a parametric function to guide the ants on their journey, and improve the results of metaheuristics by applying the 2-opt heuristic as local search. They also use candidate lists to focus their search on promising solutions. Candidate lists were used for the first time in an Ant System algorithm. presented a better implementation for CVRP by applying another ACO algorithm, they use the ‘savings’ measure associated with combining two clients on the same route versus placing them on separate routes. They also add a new step within the Ant System algorithm: a step called increasing the attractiveness of the candidate list, which orders all possible target nodes according to their attractiveness. Once this is done, another new iteration begins. They also use local search using the 2-opt heuristic. In most of the ACO applications used to solve CVRP, the local search used consists of the application of the 2-opt heuristic.) Indicate that it would be reasonable to use the Lin and Kernighan heuristics in problems where the number of clients per route is high. The idea is to improve the quality of the solutions generated by the ants before the pheromone deposition, which will lead to a better convergence of the whole method. The best exact algorithm for TSP is 'branch and cut' with this method, the computing time required to obtain the optimal solution grows as a function of the number of cities in the problem. There are constructive heuristics that can find reasonably good solutions for TSP instances in a reasonably fast time, such as Savings Heuristics or Farthest Insertion. By means of algorithms using hybrid stochastic local search, solutions close to the optimum can be found. Among these methods, ILK-H stands out as one of those that offer the best results. Methods based on population of solutions, such as mimetic or

ACO algorithms offer good performance for TSP resolution. We refer the reader to as a guide to methods for solving the TSP, where this paragraph is a very brief summary obtained from there. The challenge in solving the TSP is in solving problems with 15,000 nodes. This amount is far from our claims in solving real VRP problems, where a vehicle, depending on the type of material to be transported and the distances between customers, can serve a maximum of several dozen customers or less on each route that leaves the warehouse. Consider, for example, the distribution of furniture or animal feed on intercity routes. Our objective is to open a first avenue of investigation in the application of exact methods in the local search process within ACO algorithms for the vehicle route problem. It is clear that these local search processes will improve the convergence of the method in number of iterations. But the computation time of each iteration will increase, therefore, it is a question of verifying to what extent linear programming can be used, and checking its applicability in route problems. The objective of the study is not to develop the best VRP resolution technique; it is about making a first approach to the inclusion of the ILP as a local search. We are going to use the more traditional TSP model, we are going to solve the TSP through General Branch & Bound, and we will use the GLPK library which is distributed under the GNU license. Clearly these three aspects can be improved in a second phase of the project, if the feasibility of the proposed technique is observed.

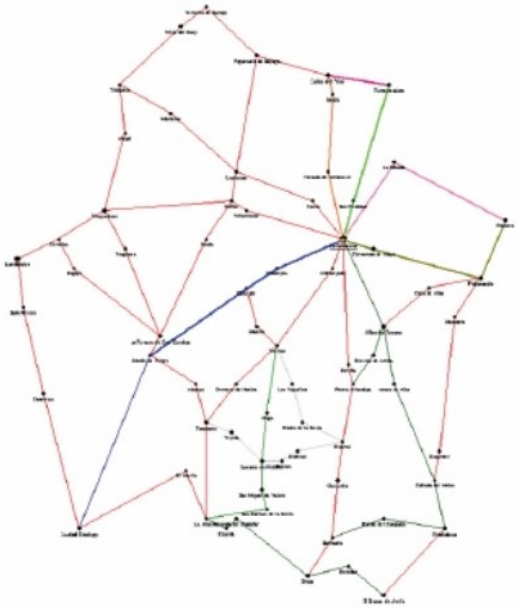

**Figure 4.2:** Definition of Vertical

## 4.3 SIMULATION AND RESULTS

This first basic method uses a time-overlapping offset. The objective is to take advantage of the times when the server is in idle mode, attending to a request, to assign it to another request. The main advantage of this method is precisely the time displacement that occurs, since thanks to this overlap between requests it is possible to reuse part of the time allocated to a request to assign it to another request without either of the two being affected. It would be affected the moment the overlap begins to span the task time of the other request. The algorithm corresponding to this method ensures that this case never happens. The procedure is similar to that of the elementary method III, but in this case the server that offers the best use of time in idle mode is assigned to the request.

This use corresponds to the amount of time that an incoming request is able to use of the time in idle mode of the previous request:

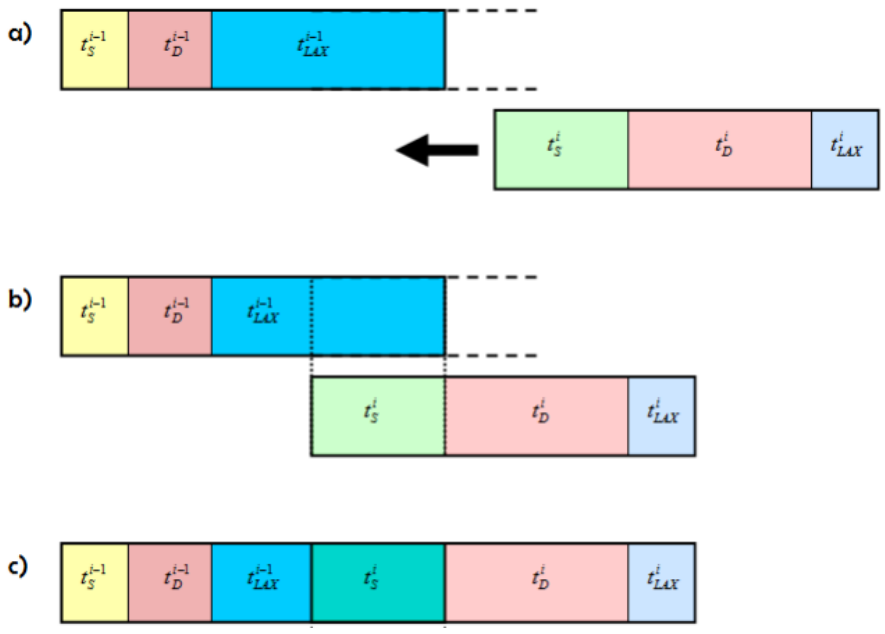


**Figure 4.3:** vertical results

# 5. CONCLUSION

Once all the features of the algorithms exposed in this project have been analyzed and all the objectives that had been set at the beginning of it have been addressed, a series of conclusions can be drawn: Based on the results presented in this document, it could be confirmed that the algorithm Hybrid I presents very low costs but at the cost of a reduction in the rest of the benefits analyzed in this project (ratio of requests accepted / lost, robustness to system load peaks, guarantees in non-equiprobable environments, ...). The hybrid algorithm II is the one that represents the highest energy cost for the system, but offers a certain level of quality of service (analyzed benefits). Finally, the hybrid algorithm III is, among all the proposed algorithms, the one that has proven to be the best in terms of reducing the total cost of the system (even in adverse environments). This is so due to the efficient use of the energy saving mechanisms implemented in said algorithm. Three hybrid algorithms have been implemented based on a series of elementary algorithms in order to optimize energy savings in Cloud Computing. For this purpose, these elementary algorithms have been developed and improved to achieve basic algorithms that constitute the basis of the final hybrid algorithms proposed in this document. Thanks to this evolution when developing algorithms, it has been possible to verify the importance of the implementation of optimization algorithms in planning and managing resources in these types of environments, and not only at the cost level, but also at the level of quality of service, performance and other relevant aspects in IT systems. Although it is true that in this type of environment load balancing is a very widespread mechanism when it comes to reducing energy costs, in the particular case presented in this project it has proven to be a tool to guarantee a certain level of service quality, rather than to reduce cost (although it also helps to reduce total system cost). In fact, what has proven to be the most effective when it comes to minimizing the total cost of the system is to use the load balancing technique together with some type of consolidation system, since in this way it is possible to take better advantage of the benefits in terms of energy cost savings, it refers to both mechanisms. One of the strengths when reducing the cost of the system is to minimize the number of machines used. In this sense, and after deeply analyzing the results obtained, it seems clear that a consolidation mechanism in an environment such as the one described in this project is very beneficial in terms of energy cost savings, since it is possible to reduce the number of servers necessary to provide service to a whole set of requests. For environments like the one exposed in this project, a mechanism for

reducing the active time of the servers implies a very interesting saving in the total cost of the system, since reducing the time that the servers are active reduces the energy cost of the system. Therefore, for future work based on the environment implemented in this project (or similar), it is highly recommended that the use of this mechanism be enforced whenever possible and as much as possible. For example, in this case, a tuning mechanism has been shown to significantly reduce server power-up time, and therefore, it supposes an energetic saving very to take into account. As expected, of the four types of nest tar c implemented in this project, type III tar c (High Load) is the one that most influences when it comes to reducing energy costs, due to its very nature. However, after the tests carried out, it has been shown that with cost reduction techniques, it is possible to alleviate the burden of this type of traffic. It may be interesting to take advantage of the results shown in this document so that suppliers take into account the benefits of each of the situations analyzed in this project, and develop resource management and planning strategies. It is therefore interesting to find a balance between the amount of incoming traffic (and consequently, the size of the block of requests with which the algorithm must work), the number of resources available in the system and the availability of the service. With this, it is possible to reduce the energy cost of the system substantially. Taking into account that this project is based on a simulation developed entirely by me, it is obvious that there are factors that have not been taken into account in the realization of said simulation, with what this implies. The entire decision-making process that has been carried out in this project implies that, although the system has been designed at all times with the intention of getting as close as possible to a real system, the results obtained may not be exactly theorists. Therefore, it is necessary to highlight the fact that both the results obtained and the conclusions that have been drawn from them are associated only with the initial data of this project.

# REFERENCES


[1] Uchechukwu, K. Li, Y. Shen, “Energy Consumption in Cloud Computing Data Centers”, International Journal of Cloud Computing and Services Science, Vol.3, No.3, pp. 31-48, 2014

[2] J. Koomey, “Estimating Total Power Consumption by Server in the U.S and the World”, Lawrence Berkeley National Laboratory, Stanford University, pp. 1-31, 2007.

[3] J. Toress, “Green Computing, The next wave in computing”, Jordi Torres, In Ed. UPC Technical University of Catalonia, Barcelona, 2010.

[4] P. Kogge, “The Tops in Flops”, IEEE Spectrum, pp. 49-54, 2011.

[5] M. Al-Fares, A. Loukissas, A. Vahdat, “A scalable, commodity data center network architecture,” ACM SIGCOMM Computer Communication Review, Vol. 38, No. 4, pp. 63–74, 2008.

[6] J. Ren, Y. Zhang, N. Zhang, D. Zhang, and X. Shen, “Dynamic channel access to improve energy efficiency in cognitive radio sensor networks,” IEEE Trans. Wireless Commun., Vol. 15, No. 5, pp. 3143–3156, 2016.

[7] Uchechukwu, K. Li, Y. Shen, “Improving Cloud Computing Energy Efficiency”, IEEE Asia Pacific Cloud Computing Congress, 2012.

[8] Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, S. Lu, “Dcell a scalable and fault-tolerant network structure for data centers”, ACM Sigcomm Computer Communication Review, Vol. 38, no. 4, pp. 75– 86, 2008.

[9] Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, “Bcube. a high performance, server-centric network architecture for modular data centers”, ACM Sigcomm Computer Communication Review, vol. 39, no. 4, pp. 63–74, 2009.

[10] H. Wu, G. Lu, D. Li, C. Guo, Y. Zhang, “Mdcube. a high performance network structure for modular data center interconnection”, In the Proceedings of the 2009 ACM 5th International Conference on Emerging Networking Experiments and Technologies, Rome, Italy, pp. 25–36, 2009.

[11] S. K. Abda, S. A. R. Al-Haddadb, F. Hashimb, A. B. H. J. Abdullahc, and S. Yussof, ‘‘An effective approach for managing power consumption in cloud computing infrastructure’’, J. Comput. Sci., vol. 21, pp. 349–360, 2017.

[12] L. Guo, G. Hu, Y. Dong, Y. L. Luo, Y. Zhu, “A Game Based Consolidation Method of Virtual Machines in Cloud Data Centers with Energy and Load Constraints”, IEEE Access, pp. 4664-4676, 2018.

[13] J. Xu, H. Wu, L. Chen, C. Shen, “Online Geographical Load Balancing for Mobile Edge Computing with Energy Harvesting”, IEEE Transaction, pp. 1-30, 2017.

[14] S. Mazumdar, M. Pranzo, “Power efficient server consolidation for cloud data center”, Elsevier, pp. 4-16, 2017.

[15] S. K. Abd, S.A. R Al-Haddad, F. Hashim, A.B.H.J.A. azizol, S.Yusso, “An effective approach for managing power consumption in cloud computing infrastructure”, Journal of Computational Science, pp. 1-33, 2016.

[16] T. Kaur, I. Chana, “Energy Efficiency Techniques in Cloud Computing: A Survey and Taxonomy”, ACM Computing Surveys, Vol. 48, No. 2, pp. 22.1-22.46, 2015.

[17] D. Kliazovich, P. Bouvry, F. Granelli, N. L. S. da Fonseca, “Energy consumption optimization in cloud data centers” John Wiley & Sons, Luxembourg, pp. 193-215, 2015.

[18] Y. Ding, X. Qin, L. Liu, and T. Wang, ‘‘Energy efficient scheduling of virtual machines in cloud with deadline constraint’’, Future Generat. Comput. Syst., Vol. 50, pp. 62–74, 2015.

[19] Hammadi, L. Mhamdi, “A survey on architectures and International Journal of Computer Sciences and Engineering Vol.7(2), Feb 2019, E-ISSN: 2347-2693 © 2019, IJCSE All Rights Reserved 256 energy efficiency in data center networks”, Elsevier, p.p. 1- 21, 2014.

[20] M. Wu, R. S. Chang, and H. Y. Chan, ‘‘A green energyefficient scheduling algorithm using the DVFS technique for cloud datacenters,’’ Future Generat. Comput. Syst., Vol. 37, pp. 141–147, 2014.

[21] X. Z. Member, L. T. Yang, H. Chen, J. Wang, S. Yin, X. Liu, “Real-Time Tasks Oriented Energy-Aware Scheduling in Virtualized Clouds’’, IEEE Transaction, pp. 1-14, 2014.

[22] Ghribi, M. Hadji, D. Zeghlache, “Energy efficient vm scheduling for cloud data centers”, IEEE/ACM 13th International Symposium on Cluster, Cloud, and Grid Computing, p.p 671-678, 2013.

[23] W. Chawarut, L. Woraphon, ‘‘Energy-aware and real-time service management in cloud computing,’’ In the Proceedings of the 2013 IEEE 10th International Conference on Electrical Engineering or Electronics, Computer, Telecommunications and Information Technology, Krabi, Thailand, pp. 1–5, 2013.

[24] E. Volk, A. Tenschert, M. Gienger, “Improving energy efficiency in data centers and federated cloud environments. comparison of CoolEmAll and Eco2Clouds approaches and metrics”, In the Proceedings of the IEEE 3rd International Conference on Cloud and Green Computing, pp.443-450, 2013.

[25] Boru, D. Kliazovich, F. Granelli, P. Bouvry, A.Y. Zomaya, “Energy-Efficient Data Replication in Cloud Computing Datacenters”, IEEE Globecom workshop on Cloud Computing Systems, Networks and Applications, pp. 446-451, 2013.

[26] T. V. do, C. Rotter, “Comparision of scheduling schemes for on demand Iaas request”, Elsevier, Volume 85, Issue 6, 2012.

[27] Beloglazov, J. Abawajy, R. Buyya, ‘‘Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing,’’ Future Generat. Comput. Syst., Vol. 28, No. 5, pp. 755–768, 2012.

[28] S. Chaisiri, B.S. Lee, D. Niyato, “Optimization of Resource Provisioning Cost in Cloud Computing”, IEEE transactions on services computing, Vol. 5, No. 2, p.p. 164-177, 2012.

[29] S. Mazumdar, M. Pranzo, ‘‘Power efficient server consolidation for cloud data center’’, Future Generat. Comput. Syst., Vol. 70, pp. 4–16, 2017.

[30] W. Zhu, Y. Zhuang, L. Zhang, ‘‘A three-dimensional virtual resource scheduling method for energy saving in cloud computing’’, Future Generat. Comput. Syst., Vol. 69, pp. 66–74, 2017.

[31] H. Duan, C. Chen, G. Min, Y. Wu, ‘‘Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems’’, Future Generat. Comput. Syst., Vol. 74, pp. 142–150, 2017.

[32] L. Zhou, ‘‘On data-driven delay estimation for media cloud’’, IEEE Trans. Multimedia, Vol. 18, No. 5, pp. 905–915, 2016.

[33] R. Li, Q. Zheng, X. Li, and J. Wu, ‘‘A novel multi-objective optimization scheme for rebalancing virtual machine placement’’, In the Proceedings of the IEEE 9th International Conference on Cloud Comput., San Francisco, CA, USA, pp. 710–717, 2016.

[34] M. Nir, A. Matrawy, “Economic and Energy Considerations for Resource Augmentation in Mobile Cloud Computing”, IEEE Transactions on Cloud Computing, p.p 1-14, 2015.

[35] J. M. H. Elmirghani, T. Klein, K. Hinton, L. Nonde, A. Q. Lawey, T. E. H. El Gorashi, M. O. I. Musa, and X. Dong, “GreenTouch GreenMeter Core Network Energy-Efficiency

Improvement Measures and Optimization", Optical Society of America, Vol. 10, p.p. A250-A269, 2018.

[36] P. Ehsan, P. Massoud, "Minimizing data center cooling and server power costs", In the Proceedings of the ACM/IEEE 4th International Conference on on Low Power Electronic and Design (ISLPED), pp. 145-150, 2009.

[37] V. Liu, D. Halperin, A. Krishnamurthy, T. Anderson, "F10 A fault-tolerant engineered network", In Presented as part of the USENIX 10th Symposium on Networked Systems Design and Implementation (NSDI 13), pp. 399–412, 2013.

[38] J. Moore, J. Chase, P. Ranganathan, R. Sharma, "Making Scheduling 'Cool' Temperature-Aware Workload Placement in Data Centers" In the Proceedings of the USENIX Annual Technical Conference, Anaheim, CA, USA, pp. 10–15, 2005.

[39] M. Armbrust, "Above the clouds: A Berkeley view of cloud computing", Technical Report UCB/EECS-2009-28, 2009.

[40] BONE project, "WP 21 Tropical Project Green Optical Networks: Report on year 1 and unpdate plan for activities", No. FP7-ICT-2007- 1216863 BONE project, Dec. 2009.

[41] J. Koomey, "Estimating Total Power Consumption by Server in the U.S and the World", February 2007, http://enterprise.amd.com/Downloads/svrpwrusecompletefinal.pdf

[42] Jordi Toress, "Green Computing: The next wave in computing", In Ed. UPC Technical University of Catalonia, February 2010.

[43] Peter Kogge," The Tops in Flops", pp. 49-54, IEEE Spectrum, February 2011.

[44] W.C. Feng, X. Feng and C. Rong, "Green Supercomputing Comes of Age," IT Professional, vol.10, no.1, pp.17-23, Jan.-Feb. 2008.

[45] X. Li, Y. Li, T. Liu, J. Qiu, and F. Wang, "The method and tool of cost analysis for cloud computing," in the IEEE International Conference on Cloud Computing (CLOUD 2009), Bangalore, India, 2009, pp. 93-100.

[46] L. Shang, L.S. Peh, and N. K. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," In the 9th Figure 4: Power-Saving Features 1 2 3 4 5 6 0 500 1000 1500 2000 2500 3000 3500 4000 4500 Scenarios Total Power (KW) server chiller crah pdu International Symposium on High-Performance Computer Architecture (HPCA 2003), Anaheim, California, USA, 2003, pp. 91-102.

[47] B. Rajkumar, B. Anton and A. Jemal, “Energy Efficient Management of Data Center Resources for Cloud Computing: A Vision Architectural Elements and Open Challenges”, In Proc. International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2010), Las Vegas, USA, July 12-15, 2010.

[48] Bohra and V. Chaudhary, “VMeter: Power modelling for virtualized clouds," International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE, vol., no., pp.1-8, 19-23 April 2010.

[49] F. Chen, J. Schneider, Y. Yang, J. Grundy and Q. He, "An energy consumption model and analysis tool for Cloud computing environments," Green and Sustainable Software (GREENS), 2012 First International Workshop on, vol., no., pp.45-50, 3-3 June 2012.

[50] B. Yamini and D.V. Selvi, "Cloud virtualization: A potential way to reduce global warming," Recent Advances in Space Technology Services and Climate Change (RSTSCC), 2010, vol., no., pp.55-57, 13-15 Nov. 2010.

[51] Z. Zhang and S. Fu, "Characterizing Power and Energy Usage in Cloud Computing Systems," Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on, vol., no., pp.146-153, Nov. 29 2011-Dec. 1 2011.

[52] A.C. Orgerie, L. Lefevre, and J.P. Gelas, "Demystifying energy consumption in Grids and Clouds," Green Computing Conference, 2010 International, vol., no., pp.335-342, 15-18 Aug. 2010.

[53] Sarji, C. Ghali, A. Chehab, A. Kayssi, "CloudESE: Energy efficiency model for cloud computing environments," Energy Aware Computing (ICEAC), 2011 International Conference on, vol., no., pp.1-6, Nov. 30 2011-Dec. 2 2011.

[54] X. Fan, W.D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in the 34th International Symposium on Computer Architecture (ISCA 2007), San Diego, California, USA, 2007, pp. 13-23.

[55] J. Moore, J. Chase, P. Ranganathan, R. Sharma, “Making Scheduling "Cool": Temperature-Aware Workload Placement in Data Centers” In Proc. of the 2005 USENIX Annual Technical Conference, Anaheim, CA, USA, April 10–15, 2005.

[56] P. Ehsan and P. Massoud, “Minimizing data center cooling and server power costs”, In Proc. Of the 4th ACM/IEEE International Symposium on Low Power Electronic and Design (ISLPED), 2009, Pages 145-150

[57] Meisner, B. T. Gold, T. F. Wenisch, “PowerNap: Eliminating Server Idle Power”, In Proc. of the 14th international conference on Architectural support for programming languages and operating systems (ASPLOS), USA, 2009

[58] S. Pelley, D. Meisner, T. F. Wenisch, and J. W. VanGilder, “Understanding and absracting total data center power,” in WEED: Workshop on Energy Efficienct Design, 2009.