



6th of June 2013



Reducing the traffic bottleneck in cloud-based data management

**Innovative
Communications
Technologies &**



Entrepreneurship
10th Semester

Student:

Alexandros Fragkopoulos

Supervisors:

Albena Mihovska
Sofoklis Kyriazakos

Abstract

In the era of networking and computing power one can find many solutions as to how to get connected to the rest of the world and provide one's services and products. Nowadays, cloud computing is one of the leading approaches on how one can own computing and storage resources without actually built them. But things are not always that simple. Considerations that businesses have also their own (maybe of lower performance and capacity capabilities) infrastructures and don't want to waste them, or that the cloud is considered yet insecure or even that they don't trust the availability time of the cloud infrastructures, are some reasons for a business to think carefully before making a migration to the cloud. This project comes with a mixed solution. It makes a combination of cloud resources together with the utilization of private on-premises resources. The idea is based on the concept that a company needs the additional resources (provided it has any of its own) when the network becomes overloaded. This is the reason why a combined, hybrid scheme would make more sense. This solution though has been already provided by cloud bursting which is a kind of a hybrid cloud. Another problem that is often faced in cloud computing is the creation of various kinds of bottlenecks. This is not only loss in performance but also loss of money for the customers. Given this kind of problems and having them act as incentives the project's target moves one step further and proposes a scheme which includes a randomized routing algorithm that will make a different use of the cloud provider's data centers in the greater area around the company and an SDN on-premises network that will be responsible for the selection (if many) of the applications that have to migrate to the cloud for a limited amount of time (during peak hours of the day for the application) to offload the internal network. The aim of this is to distribute traffic across different data center and decentralize it. This project's use case is an e-commerce website's migration. The company makes use of its infrastructure for most of the time, has complete control over its data and applications and is using a pay-per-use charging model for the usage of cloud services. Since the project introduces a randomized routing algorithm it is important to mention that this algorithm tries to address the bottleneck (i.e. storage I/O bottleneck) that is created by having many clients served from one data center (probably because it is located near an industrial zone) whereas, other data centers stay under-utilized. This algorithm will choose randomly a data center that is within the limitations the cloud provider's data centers and can accommodate the enterprise's data and instances. Then, the data will be replicated just before the load has to be redirected to the cloud. By using, an on-premises, SDN network there is also a benefit for the cloud provider that one can inform the local SDN controller about the available data centers that can be used in this list for random discovery. Thus, data centers that are overloaded or experiencing malfunctions can be excluded from the process. Last but not least, this proposed framework enables cloud providers to set an additional fee (in BW price) when the chosen data center is close to the enterprise.

Acknowledgement

I would like to express my gratitude to my supervisors Albena Mihovska and Sofoklis Kyriazakos who guided me throughout the learning process and helped me learn a lot. I consider it an honor to have worked with them for my master thesis. I also owe my deepest gratitude to my parents who have given me the opportunity to follow my dreams and not only supported it me financially but also encourage me at every step of my life. Last but not least, I am deeply grateful to my friend Ioanna Kazantzidi for supporting me the past two years when most needed.

Table of Contents

Abstract	1
Acknowledgement.....	2
List of Tables.....	5
List of Figures.....	5
List of Abbreviations.....	5
Chapter 1 Introduction	7
1.1 Motivation	8
1.2 Problem Formulation	8
1.3 Usage Scenario Description	9
1.4 Contributions of this research project	10
Chapter 2 Cloud Computing	11
2.1 Definition of Cloud Computing	11
2.2 Migration to the Cloud	13
2.3 Challenges for the Cloud.....	15
2.4 Service Level Agreements (SLAs).....	16
2.5 Data centers and their architectures.....	16
2.5.1 Network-Attached Storage (NAS).....	19
2.5.2 Storage Area Network (SAN)	19
2.6 Cloud Bottlenecks	20
2.7 Intercloud communication	22
2.8 Content Delivery Networks (CDNs)	23
2.9 Domain Name System (DNS)	23
Chapter 3 How this research contribution compares to other research work in the area	25
3.1 Network as a Service (NaaS).....	25
3.2 Software Defined Networking (SDN)	27
Chapter 4 Proposed Design & Implementation	30
4.1 Theoretical Framework and Design of the Algorithm	30
4.1.1 How SDN and HTTP redirection enables part of the process	30
4.1.2 An ecommerce website as the main example for the migration process	32
4.1.3 Theoretical background of the distributed discovery of the data centers	32

4.1.4	Differentiation from current approaches and rationale	34
4.2	On-premises infrastructure	35
4.3	Algorithm Design	35
4.4	Implementation	38
4.4.1	Program that simulates the code in the SDN controller	38
4.4.2	Calculation and comparison of the proposed idea's cost to other solutions.....	41
Chapter 5	Conclusions & Future work.....	46
5.1	Conclusions	46
5.2	Future work.....	47
Appendix	48
Bibliography	52

List of Tables

Table 1	Numbers that will be used for estimation of the costs	42
Table 2	Cost when migrating entirely to the Cloud	43
Table 3	Calculated costs for the projects proposal	44
Table 4	Calculation of the CDN solution	44
Table 5	Presentation of all the solutions and their costs	45

List of Figures

Figure 1	Complete architecture of the usage scenario	9
Figure 2	Cloud Definition depicted in an image	13
Figure 3	Data Center design	18
Figure 4	Design of SAN storage being part of the data center	20
Figure 5	Difference in Server performance to I/O performance [7]	21
Figure 6	Bottlenecks due to response time [7]	22
Figure 7	The SDN architecture	28
Figure 8	OpenFlow's proposal on combining MPLS with SDN & OpenFlow	29
Figure 9	Randomized discovery of a DC with the proposed routing algorithm	33
Figure 10	Standard procedure when everything function well on-premises	36
Figure 11	Migration process to the cloud infrastructures	37
Figure 12	Execution of the algorithm for the discovery process for migration	39
Figure 13	Another execution of the algorithm where the radius is smaller	40

List of Abbreviations

API	Application Programming Interface
BW	Bandwidth
CAPEX	Capital Expenditure
CDN	Content Delivery Network
CNAME	Canonical Name
DB	Database

DC	Data Center
DCIN	Data Center Interconnect Network
DDNS	Dynamic DNS
DNS	Domain Name Server
EoR	End of Row
FPGA	Field-Programmable Gate Array
GPS	Global Positioning System
HD	Hard Disk
HTTP	HyperText Transfer Protocol
I/O	Input/Output
IaaS	Infrastructure as a Service
IBM	International Business Machines
INPE	In-Network Processing Elements
IOPS	Input/Output Operations per Second
IP	Internet Protocol
IT	Information Technology
ITU-T	International Telecommunication Union for Telecommunications
MIB	Management Information Base
NaaS	Network as a Service
NAS	Network-Attached Storage
NIST	National Institute of Standards and Technology
OPEX	Operating Expense
PaaS	Platform as a Service
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random-Access Memory
SaaS	Software as a Service
SAN	Storage Area Network
SATA	Serial ATA
SDN	Software Defined Network
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SSD	Solid-State Drive
ToR	Top of the Rack
TTL	Time To Live
URL	Uniform Resource Locator
VLAN	Virtual LAN
VM	Virtual Machine
WAN	Wide Area Network

Chapter 1 Introduction

Throughout the years that computers have been around, many trends have passed that served the needs every time of the current problems in computational power. Many phases have appeared in this process to make this faster and better. At first, people started by using mainframes that had very basic computational skills [1]. Then, the personal computer came into play and opened the door to empowering the user and giving one the processing capabilities that one needed. A little later the first steps of networking were done and clusters became popular and a group of people could access a cluster and retrieve information from there [2]. Still everything was in the same location and cluster computing was limited to the group of users on this location. When internet made its appearance, servers were accommodated in remote offices and people could access them through the global network anytime. As the needs of computational power and storage capabilities became more demanding the grid gave the solution for retrieving information in a web like design and a person could retrieve all those information from various nodes. Nowadays, the focus has been turned into new technology that is described by remote area housing of the facilities, scalability of the services that are being accommodated, easy access from anywhere in the world and self-service, transparent way into locating the pooled resources and measuring service by pay per use. This new technology is called the cloud or cloud computing and it is one of the topics underutilization and research for the current era. Cloud computing enables individuals, organizations, companies and many more to store data in a virtualized manner make use of processing resources from powerful data centers and get connected to applications and services easy and fast from anywhere [3]. Consumers also have the ability to monitor and control many things through the internet and even make different business deals with the cloud providers that could benefit them and their enterprises. Although, everything looks promising, things can get challenging when it comes the moment for a consumer (e.g. an enterprise) to decide whether one should migrate to the cloud or not, the answer is not always that simple. Many factors can make the migration beneficial for the consumers but there is always more than meets the eye. SLA's (service level agreement) (i.e. contracts for services) are something that is always tricky and making contracts requires a great deal of thinking about all the parameters that might get affected or changed by the transition. Basically, there are three main types of cloud deployments: the private cloud, the public cloud and the hybrid cloud. All three types are used for different purpose by different user groups [4]. In this project the main concern will be around a form of hybrid clouds and how can an enterprise fully utilize an internal small data center and expand when needed to the cloud-based environment. Different issues arise on these two different states and here there will be an effort to address them. The work that will be presented has some similarities with the CDNs and it has to be clarified that this project doesn't attempt to compete with already established solutions but to offer an alternative that can be beneficial for some use case scenarios.

The following chapters provide an insight on the problems that are created from cloud's usage (e.g. bottlenecks) and based on that, the project's proposal is made. This project aims to deal with under-utilized data centers. This can be accomplished by making a randomized routing

algorithm. Through this algorithm data centers are meant to receive traffic not entirely based on their relative location to enterprises but also traffic for random locations that is forwarded there. This offloads data centers that are surrounded by businesses. This proposal also provides financial incentives for the cloud provider. By adding different fees to enterprises based on their distance from the data centers new financial opportunities open.

1.1 Motivation

This project proposes a randomized routing algorithm to distribute traffic across different data centers. The idea is inspired from the need to decentralize cloud operation and avoid unnecessary overloading of certain data centers whereas others stay under-utilized. Furthermore, the goal of this project is also to keep the cost low for the small and medium enterprises by using their existing infrastructure and making as little alterations as possible and also combining the small on-premises data centers with new technologies such as NaaS and SDN that are presented in detail in Chapter 3. Such technologies enable additional control over the internal network and give even more incentives on how the proposed idea could be combined with them to make the most out of this. The current cloud solutions - on many different levels - are indeed well-established but they also have disadvantages or challenges which act as a further incentive to research this matter and try to address some issues that arise.

1.2 Problem Formulation

Cloud solutions are a relatively new concept. There has been made an effort to identify some problems that might affect small and medium enterprises that might not have the ability to financially support a complete migration to the cloud [5] or they just wish to remain local for some applications [6]. This project's usage scenario has to do with the process of moving the functionality from the internal small enterprise data center to the cloud in order to offload the enterprise network and redirect some high-bandwidth consuming applications to the cloud. This migration will take place in a randomized manner and for a specific time interval. This will happen in order to balance more efficiently the load between data centers, in order to offload data centers that are located very close to areas that are full of enterprises and business buildings. Although data centers tend to have a really high I/O performance, there are cases when bottlenecks are created (e.g. server bottlenecks because of low processing power [7]) and the data center's performance is decreased. In order to avoid this situation, a more random traffic architecture will avoid unnecessary testing of data centers' limitations when not needed.

1.3 Usage Scenario Description

A small/medium enterprise has already a small data center that facilitates in the same building as the offices. Due to the demanding needs there is a need to expand and acquire more resources. The additional resources will enable the business to develop their applications even further and add new features and make their application more competitive and thus more profitable. But the problem is that the enterprise is small and they don't have the appropriate budget to migrate fully to the cloud, neither do they wish to underutilize their on-premises data center. The application that creates the biggest load is an e-commerce website and the load is during some hours of the day. The only solution that will benefit both the usage of their own infrastructures but also the additional needs of the increasing clients is the solution that is proposed by the project. In Figure 1 one can see the design of the architecture that the enterprise makes use of.

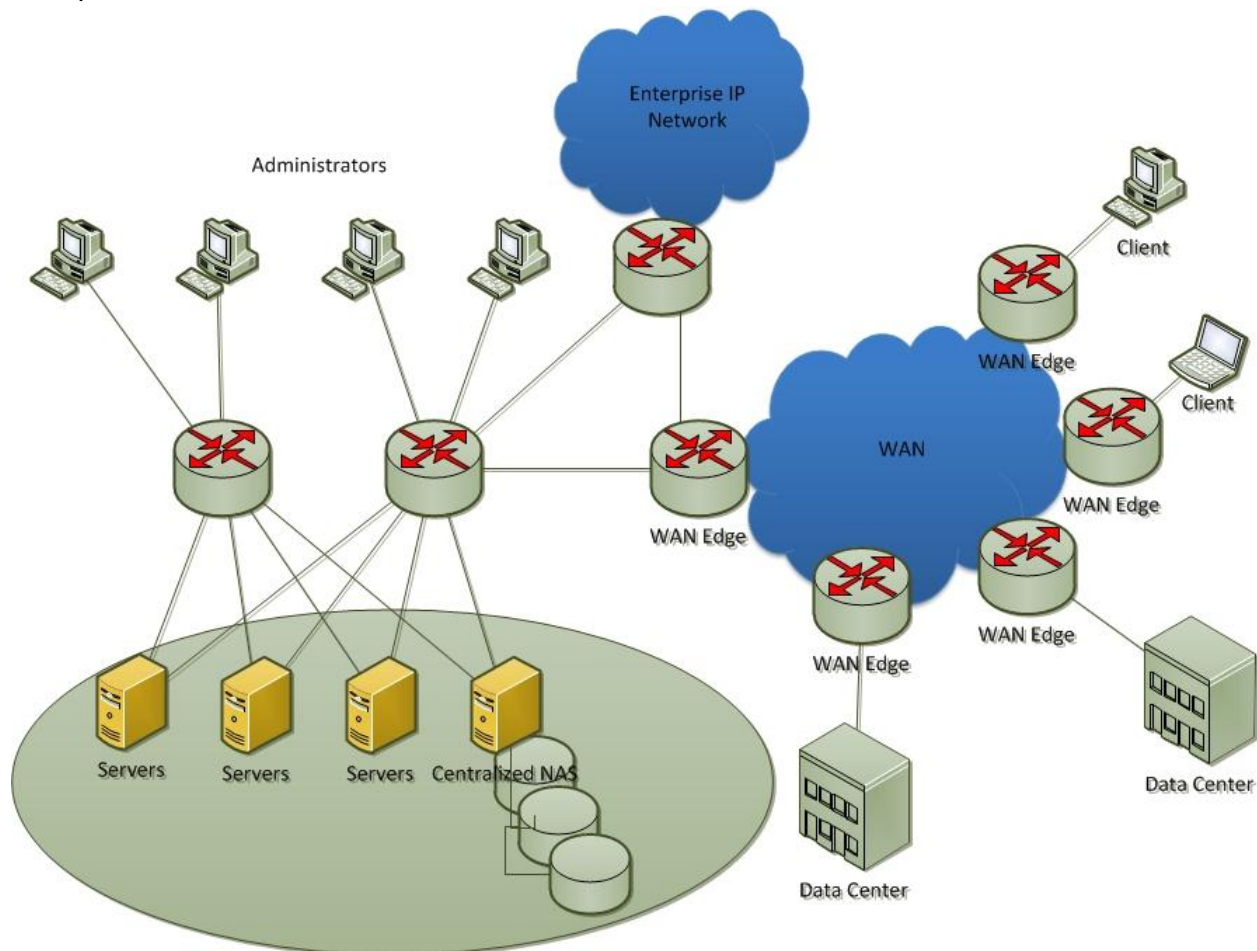


Figure 1 Complete architecture of the usage scenario

The process will be a migration of the data and instances that will take place on a daily basis. But there is an advantage also for the cloud provider. This project tries to address the routing of the replication of the data to the cloud with a randomized routing algorithm. Through this process the enterprise will benefit from the cloud infrastructures as long as it is needed and the

cloud provider will benefit from the randomized routing scheme for the replication of the data to the cloud and as a result of the decentralization of the tenants.

1.4 Contributions of this research project

We tried to address different types of problems both financially and technologically. The result is small but important contributions that are stated below:

1. Lowering the financial expenses for the cloud services by having a dynamic and time-limited migration to the cloud.
2. Distributing the load across various data centers, in a relatively close geographical distance from the enterprise and in a semi-randomized way (control of availability of a data center makes it a semi-random process) by making use of the Fibonacci Spiral.
3. Keeping the on-premises infrastructure as intact as possible.
4. Presenting financial opportunities (charging higher BW (bandwidth) price for closer data centers) to the cloud provider by making use of the randomized routing algorithm.
5. Making use of new technologies, such as SDN to deploy the hybrid cloud model. Thus, it enables even more compatibility with future applications.
6. Making it even more appealing to enterprises as they can increase their capacity by adding a number of low-performance SATA drives without worrying about I/Os. For performance issues that arise during peak hours there is always the redirection to the cloud.

The second and the last argument above indicate also the differences that this research work has, compared to the existing hybrid cloud solution of cloud bursting [8].

Chapter 2 Cloud Computing

The focus on the project is to provide a solution suitable for the decentralized cloud operation. This Chapter analyzes the essence of clouds and describes some mechanisms that are used to increase the performance and stability of this new type of technology. The main goal is to analyze most of the things that matter around DCs and Clouds but also to have a look at problems that can occur.

2.1 Definition of Cloud Computing

Cloud Computing is a term that has been created some decades ago [9] but hasn't been widely used until recent years. This term explains a different approach - different than grid or cluster computing - in having shared resources (computational, storage and services) in a virtualized manner accessible from anywhere, anytime [9].

According to NIST [10], cloud computing consists of some basic elements. Firstly it has to comply with some basic characteristics. These characteristics determine if the type of computing is based on a cloud. So, the cloud computing has to be:

- on-demand and self-service
 - A tenant has to have the ability to use/provision computing capabilities or storage resources anytime without the need of the cloud provider's intervention
- accessible from the broad network
 - The cloud has to be accessible from any type of device and by anywhere
- scalable
 - Any type of service on the cloud has to be elastic and easily scalable. Whether the customer wants more or less resources to provision for a certain application, one has to be able to do that
- has to have a pool of resources
 - The tenants have to be provided with location independent resources. Since the cloud computing is all about virtualized resources, there has to be a clear separation of the physical side (e.g. which data center serves the tenant). Tenants must be able to assign resources as needed.
- measured service
 - The services supplied by the cloud provider are all measured and tenants are able to know what exactly they consume and most of the time this is a way to put a pricing scheme on the services or resources a tenant uses.

The characteristics mentioned above define the cloud. Through cloud computing each tenant is able to use a range of service models. The service models that exist and are the most basic are the following:

- Software as a Service (SaaS)
 - With this type of service the consumer can reach the applications and interact with them. On this higher level of interaction there is limited customization since the tenant is only able to alter some characteristics of the application, without having the ability to do anything else. This kind of service can be accessed by any type of client device.
- Platform as a Service (PaaS)
 - The second type of service moves a little deeper into control and gives the tenant new degrees of freedom. On this level the tenant can create one's own programs and applications by using the tools that the cloud provider supplies. Still the level of abilities is limited on the applications and the tenant can't manage or alter the cloud infrastructure.
- Infrastructure as a Service (IaaS)
 - On the deepest level a tenant can reach for a service, there are various abilities. Storage, network and computing resources can be provisioned by the tenant and the tenant can run different operating systems and customize the environment as he wishes. The cloud infrastructure is once again unable to be utilized. In the next chapter there will be a description on a type of service that is proposed to allow tenants to modify the existing cloud infrastructure network for their applications, in a more efficient way according to their needs and it is called Network as a Service (NaaS). But this type of service is not included in the three fundamental ones.

A last categorization that has to be made for cloud computing is the different types of deployment models when it comes to what type of cloud a tenant or a group of tenants will make use of. The following types of clouds are the ones used:

- Public Cloud
 - The public cloud type of deployment model is open to the public and it can be managed by a company or a governmental actor. The public cloud exists on the cloud providers' infrastructures.
- Private Cloud
 - The private cloud is a cloud that can exist inside the cloud provider's infrastructures and cannot. It gives access to a certain number of people from a company and it is controlled by the company or some combination of a third party actors and the company.
- Community Cloud
 - the community cloud is a type of cloud that gives access to a group of people from different companies. These companies have some shared interest and this is why they use this type of cloud. It can be managed by this companies or a third party actor.
- Hybrid Cloud
 - This cloud is a combination of some of the types mentioned above. The different types remain independent but are connected for reasons of data and application portability [11].

Figure 2 below summarizes all the definition of cloud computing along with its characteristics and the service/deployment models.

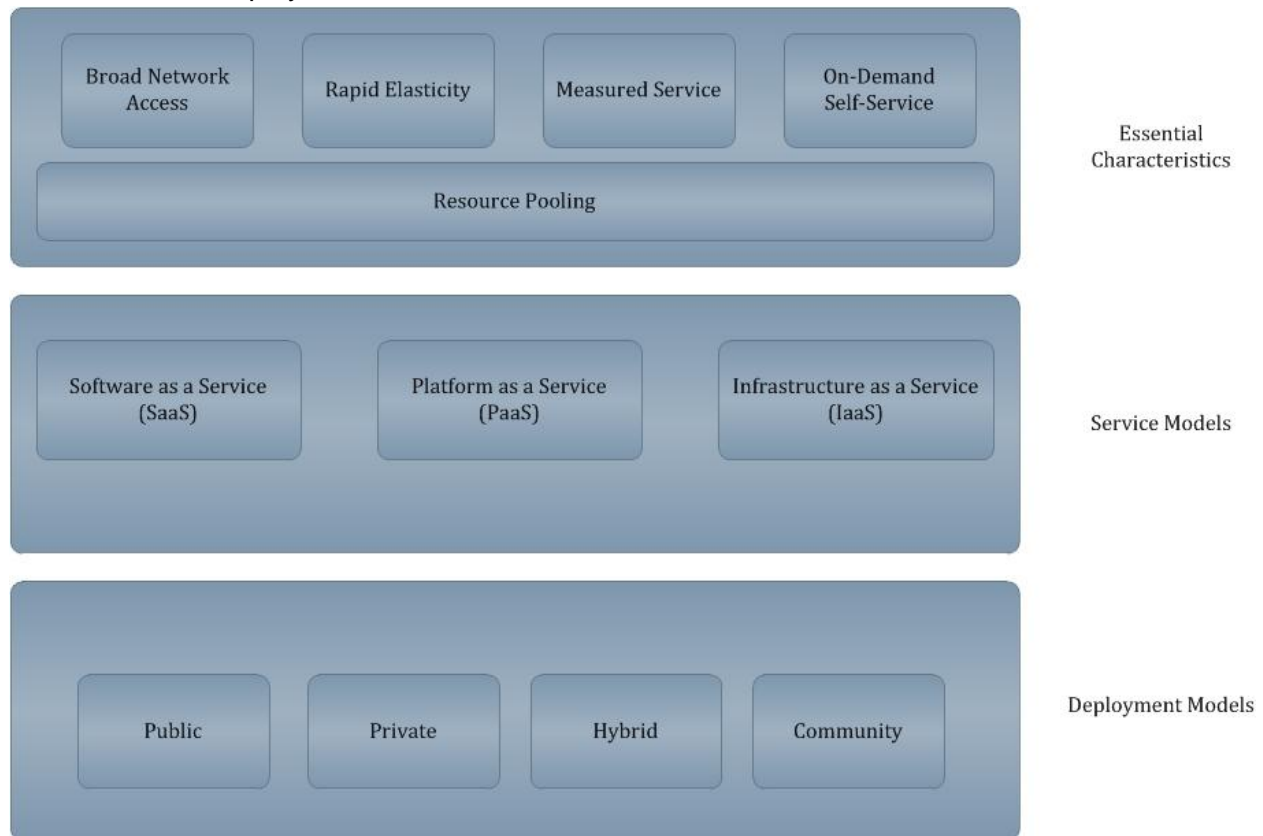


Figure 2 Cloud Definition depicted in an image¹

In this section there was a detailed explanation about cloud computing and all of its characteristics. Furthermore, there was a presentation of the different service models but also the different types of deployment models.

2.2 Migration to the Cloud

There is a variety of reasons on why a customer (meaning an enterprise) should migrate to a cloud. Most of the small/medium enterprises run their applications on their private data center. But the decision on whether they should migrate or not to cloud solutions is not always so clear. According to [9] the reasons for this migration can be of business, economic and technological nature. All parameters have to be taken into consideration for a worthy and successful migration.

¹ http://www.cloudcontrols.org/wp-content/uploads/2011/06/NIST_Visual_Model_of_Cloud_Computing_Definition.jpg

According to the same source [9], there are chances that an application will be able to be migrated as it is but there are also other use cases that show a migration of some specific parts of an application. This selective migration of features is based on one of the five axes: application, code, usage, architecture and design. The migration can be described by the simplified formula:

$$P \rightarrow P'_C + P'_I \rightarrow P'_{OFC} + P'_I$$

The application is denoted by P before the migration process. The P'_C shows the part of the application that is after the migration and P'_{OFC} is the part that has been optimized for the Cloud. If the application migration is successful on all five axes then the P'_I part will be zero. Otherwise the application will have the moved part into the cloud and the local part both running together. This happens in the case of the hybrid deployment cloud model.

There also other factors to take into consideration such as economic. The transition into a cloud-based application environment reduces both the Operating Expenses (Opex) and the capital expenses (Capex). The scalability and elasticity of the cloud gives also another advantage which is the unstable load-wise environment. All of the former reasons give benefits for one to migrate to the cloud.

Some last but not least parameters that are important for the decision on the migration are the partial licensing of applications and the tariffs that accompany the scalable resources. Although, according to [9] these tariffs can always vary. So, in order to finally take a decision whether the migration is the answer for an enterprise or not, there should be a modeled technique to aid for this. There is a questionnaire model proposed in [9] that suggests the following formula in order to assess the questions and categorize them and in the end check whether the number is above or under the threshold.

$$C_l \leq \sum_{i=1}^M B_i \left(\sum_{j=1}^N A_{ij} X_{ij} \right) \leq C_h$$

The questionnaire follows this formula for defining the importance of different questions based on the matter of migration. So, B_i is a relative weight that is being given to each group of questions N . M represents each class of questions. A_{ij} is a specific constant to every question and X_{ij} is a number which ranges from zero to one to show the relevance and usefulness of the question to the questionnaire. C_l and C_h both correspond to the lower and higher thresholds respectively. These values give some space for tolerance between cases of migration to avoid minor cases.

2.3 Challenges for the Cloud

The cloud architecture and the function of the cloud, give rise to some questions and some challenges. According to Accenture [12] the major challenges are four. Starting from the most important - the first challenge concerns the safety of the data that are stored in remote DC's. Enterprises are concerned about the safety of the DC's firewalls and if DC's can effectively mitigate or segmentize attacks, since the virtualized shared pool of resources can in fact be physically close. Since this is a potential scenario, tenants worry about the degree of segmentation of an attack and if a DC will be able to stop the attack from one tenant and not being spread to the rest. Another legal issue that comes up is the legitimacy of the data across different countries. Supposing that the data are stored in DC's outside of the country of the data's origin, there could be some legal issues on some cases.

As a second challenge Accenture sees the complexity of managing the contractual relationships between tenants and contractors (cloud providers).

The third challenge is the lock-in risk created by the cloud providers because of different technologies. In this challenge, tenants have to think about the risk of having from using the services of one provider concerning the ease for changing providers.

The fourth and final challenge according to Accenture has to do with the management of the cloud. This challenge is based on the control that an IT department of an enterprise can have over multiple applications. Since the applications can change anytime through other personnel (e.g. business users) of the enterprise, IT personnel has to try and keep the applications under control.

On the Handbook of Cloud Computing [4] the authors express more technical concerns and follow the same type of challenges as Accenture on the first challenge. Concerning the challenge on control of the cloud platforms they propose another challenge which is the lack of complete knowledge of the enterprise applications from the cloud providers. Thus, the control raises some concerns on the IT departments. Another important factor as a challenge is the case of lower performance due to the distance from a user until the DC. This challenge will be part of this project's analysis. There will be an effort to be addressed. Some other challenges as the reliability of the DC and the uptime of the applications all day-round tend to be a smaller issue [13].

Another challenge that slowly shows up is the need for more big data based cloud infrastructures. Big data is called large amounts (e.g. in the scale of petabytes to exabytes) of data (e.g. data that are received by a satellite or data that are received by the solar activity) that are being collected and then stored to get processed and create some big data analytics about a certain task that has been measured [14]. Big data are difficult to be distributed and/or processed because of their volume and the speed that things have to be done. Any action would consume large amounts of storage or processing capabilities from a data center. Currently, it is not very clear whether cloud providers can meet the demands of a big number of big data applications on their facilities [15].

There are still many open issues in order for cloud computing architecture to reach up to a level where every aspect of all these features runs smoothly and more trust is built between cloud providers and consumers.

2.4 Service Level Agreements (SLAs)

A very crucial point when a business wishes to make use of the services of the cloud, is to make the correct contract and pay attention to the detail as this is the only document that might allow the consumer to ask for guarantee of the quality of the services or/and compensation in case that something is not expected as it was planned. So since the relationship between the consumer and the cloud provider will have a contractual nature some things have to be carefully chosen. These things are the following:

- Uptime
 - How much time will the services be available without failing to function at the data center's side servers?
- Performance and response time
 - Which are the minimum parameters that establish the quality of the services supplied by the cloud provider?
- Infrastructure and security
 - Insurance that data are always secure by cryptographic ways either when being stored or when being transmitted.
- Error correction time [16]
 - The time it takes for the errors to be resolved.

Also an SLA should be reconsidered and maybe altered when one of the services change. Other things that have to be described and settled are the accessibility of the tenant's data and the location of them. The latter could create issues if the data are geographically stored somewhere where local legislation prohibits the possession of them. SLAs in general are very important for both the client and the provider and if done carefully each side enjoys the benefits of a good business relationship.

2.5 Data centers and their architectures

Data centers are the main physical location where the cloud is based. The DC infrastructures are responsible for storing data, computing and creating those complex networks where each tenant possesses a virtualized space and makes use of computing capabilities. Since the DC's are one of the most important parts of the cloud, there should be an overview on how they function and how do they provide their tenants with high amounts of resources anytime and from anywhere.

Data centers are a very large topic by itself. The scope of this project is not about providing with details about data centers rather only some basic features have been incorporated in this research work that would also give a small overview of how data centers function and how this project's goal is connected to them.

Cloud data centers typically have many parts that are involved in the process and not only the buildings themselves. These are the following:

- Data center
- Internet Access
- Wide area backbone
- Management
- Tenant's premises

The complete design has all of the parts above. Thus, data centers are able to provide the network, storage and computing resources they have. Inside the data center building there is also a network architecture that takes care of the information that are exchanged within the data center. The structure of this network according to CISCO [17] consists of three main layers:

- the core layer
 - This layer is the gateway of the building and has high redundancy and bandwidth. It is directly connected to the aggregation layer and handles the traffic of the packets between different aggregation entities.
- the aggregation layer
 - The aggregation layer has the job to interconnect the access layer and handling the traffic for all these sessions that run on the physical servers. Important to state that the aggregation as well as the core layer do not have any server but only network racks.
- the access layer
 - The access layer is the layer connected directly to the physical servers and it operates in modes of L2 or L3 [17]. Cisco's recommendation is grouping two switches in the access layer together for server redundancy or for managing, producing and backing up.

In Figure 3 one can see a simplified version of the data center design containing the three layers and their interconnection.

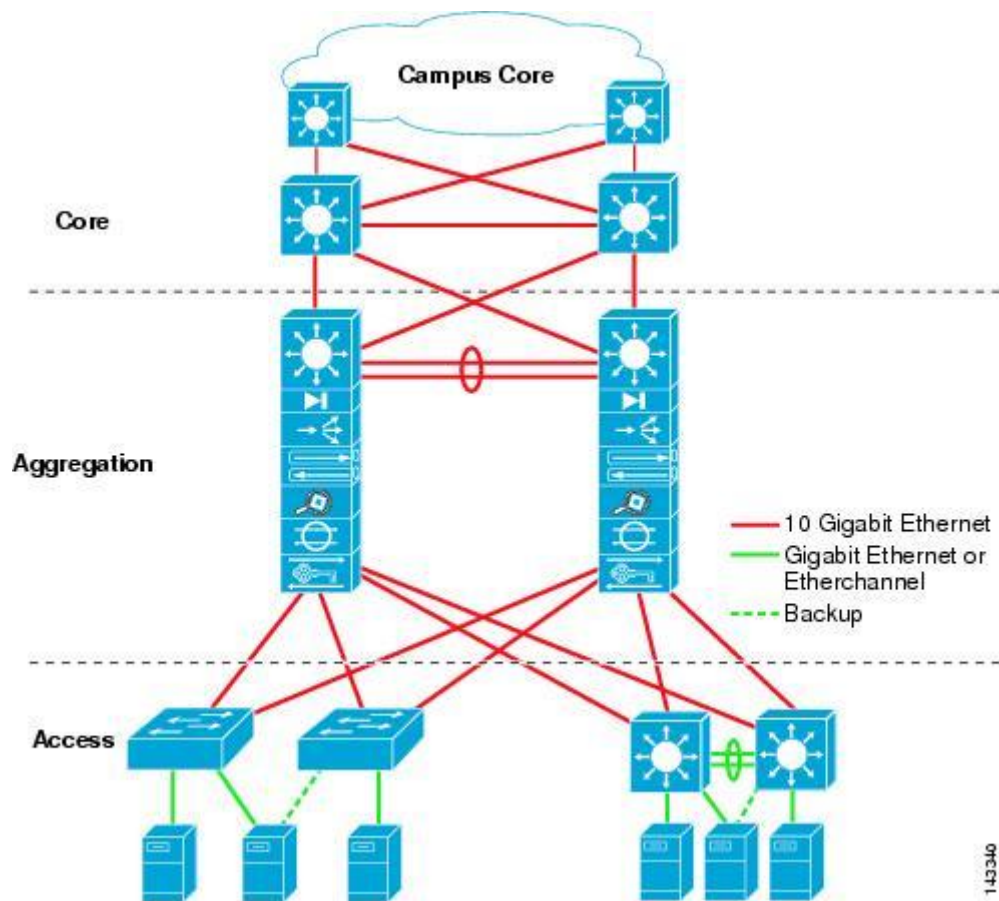


Figure 3 Data Center design²

Another important characteristic of data centers is the physical design of the network. Two ways are mainly used for connecting the data centers' servers, switches and SAN's (Storage Area Networks): Top of Rack (TOR) and End of Row (EoR).

Top of Rack design, means that in each rack all the servers are being connected with two switches on top, middle or bottom with copper. But the copper is not expanded outside of the rack. The connection outside of the rack is made with fiber and it connects the racks with the aggregation switches and the SAN. One clear advantage this design has is the limited usage of copper. This enables higher speeds since copper is only used for very small distances and the rest is fiber. Although this design doesn't have messy cable connections and is quite fast, it has also some disadvantages like the fact that there are many switches in the access layer (two per rack) and this creates a great need of ports in the aggregation switches and this can create a constraint concerning the scalability.

The next commonly used physical design is End of Row (EoR). End of Row has a different structure and copper is more widely used in this design. Each rack has multiple servers and they are connected from there until the end of the row with the switches, with copper. Since each row of racks has one place to connect all of them onto a switch, that means that the switches are less in number compared to the Top of Rack. Since this is the case, the aggregation layer doesn't face the same problem of the need of too many ports and the access

² <http://www.cisco.com/en/US/i/100001-200000/140001-150000/143001-144000/143340.jpg>

layer has a lower maintenance cost since the switches are generally less. Disadvantages here considered being mostly the extended use of copper and the way this affects multiple attributes, such as: increased cable management, lack of ability to use less power and unable to adapt higher speed between servers and switches. Also, this design is more limited for future expansions and will likely face more challenges. The last disadvantage is that if one wants to make upgrades or alterations, it should be done on the entire row (since the changes take place per row in this design) thus the architecture is less elastic [18].

Another matter that applies to the architecture of data centers is the way data is stored and improved ways to combine performance and storage capacity. In small data centers NAS (Network-Attached Storage) systems dominate, since this solution is cheaper to install and easier to manipulate. Big data centers have a second alternative, the SAN type of storage architecture. SAN is a more appropriate architecture for a data center and with an increased cost [19].

2.5.1 Network-Attached Storage (NAS)

As said before a NAS filer is easier to be maintained and easier to be understood. This storage system gives lower opportunities in expansion than SAN but instead provides with a simpler and more user friendly solution to small data centers that need capacity over performance. Although this problem can be avoided by introducing expensive storage disks (such as SSD's) but this will increase massively the cost of investment for a NAS filer. And as described in [20], *"...NAS systems provide a richer, typed, variable size (file), hierarchical interface..."*. NAS is not a server-like storage system and this means that no server type actions can be done with this system. It is only good for storage capabilities and runs over Ethernet [21], [20].

2.5.2 Storage Area Network (SAN)

A SAN is a more performance oriented system which introduces higher amount of complexity and make it more difficult for a small enterprise to maintain it. Thus, SAN's are often found in big data centers where performance and speed is crucial. Where throughput is crucial (thing that is extremely necessary in a data center) and not an option, SAN performs a lot better because of the way it treats the data and the underlying protocols that are used. According again to [20] *"...SAN systems provide a simple, untyped, fixed-sized (block), memory-like interface..."*. SAN can act as a server and it uses fiber channel adapters for the communication in the rest of the network (i.e. servers). Since it uses this type of connection (fiber channel) SAN provides higher I/O in contradiction to NAS. Last but not least is that SAN is more expensive than NAS and in small/medium enterprises where budget can be limited enough, SAN systems are not easy to acquire [19], [20], [21], [22].

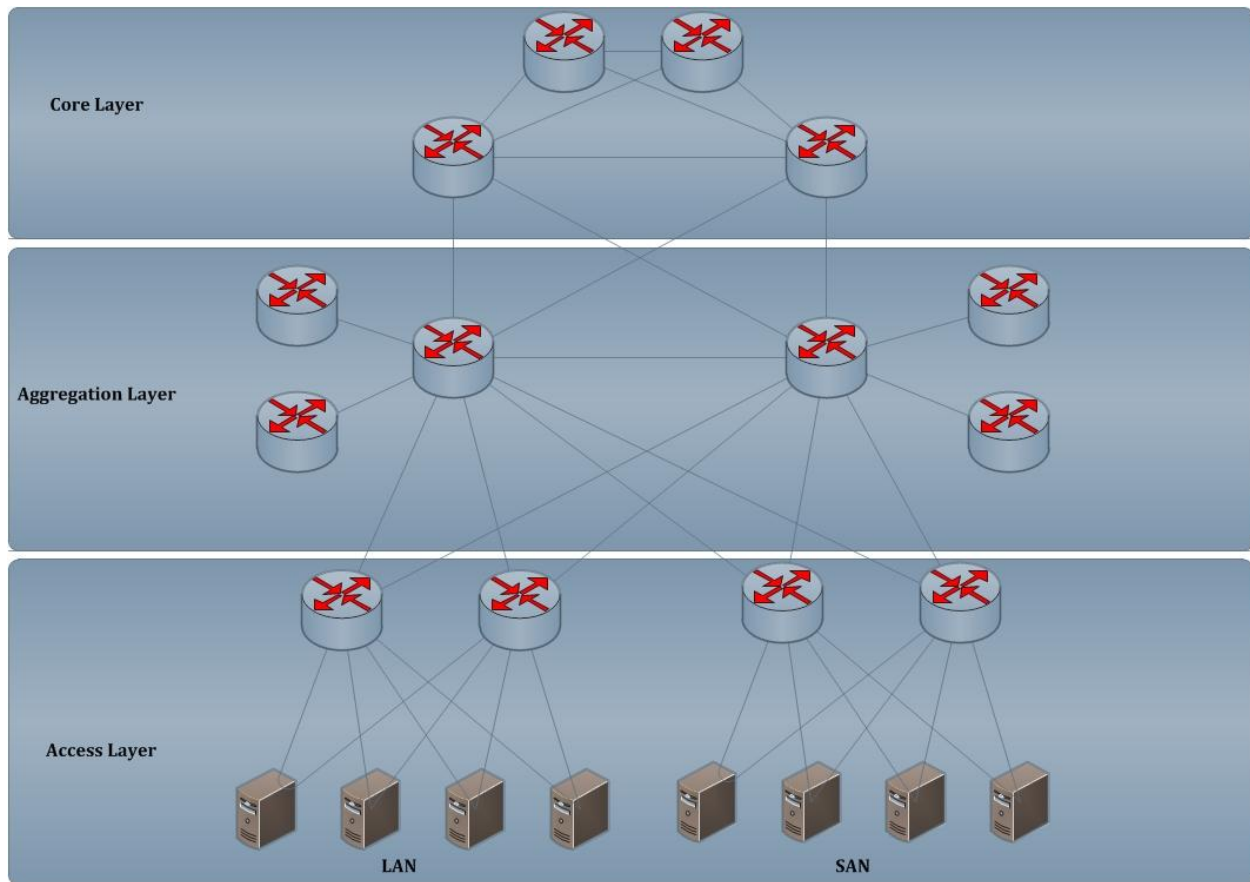


Figure 4 Design of SAN storage being part of the data center³

The design of how a SAN is connected in a data center to the rest of the topology is depicted in Figure 4 above.

2.6 Cloud Bottlenecks

Even if clouds present an image that bandwidth is high enough, response times are really low and clouds are capable of doing anything, there is an existing disadvantage to clouds that causes bottlenecks for their complete successful deployment.

Bottlenecks in data centers can be found in the following locations:

- Servers
- Network infrastructure
- Applications
- Storage Infrastructure

The purpose of a data center is to deliver high I/O and show no signs of delays. But due to the aforementioned entities and networks, sometimes bottlenecks are unavoidable. Although

³ http://www.spirent.com/~media/Images/SolutionPages/Images_STC/EpriseDataCenterNW/DC_LAN-SAN_Topology.jpg?w=400&h=372&as=1

servers do create bottlenecks as stated in [7], nowadays there are more important bottlenecks since processing capabilities of servers are quite high [23]. The only problem that could appear is the bottleneck or the lower performance due to the fact that one can use resources from different servers (e.g. VM's across different servers) or different storage locations – within a data center – and in a virtualized scheme, the connections among the different physical entities don't have to intervene and reduce the performance. Thus, there is an increased need for higher bandwidth links in order for the physical links to be faster and produce a seamless experience to the end user. However, bottlenecks arise also in the storage infrastructure (e.g. DAS, NAS, SAN) are far more important since big data have come to stay and this new trend needs high amounts of storage space and what's more difficult in this case, is the process of finding and delivering the content to the servers with a low response time and allow them to process it in order to serve the applications' needs. According to [7], servers processing capabilities have increased far more than the I/O capabilities of the storage infrastructure (see Figure 5). This creates an unwanted gap that harms and degrades the performance of the application. Due to this bottleneck users can potentially experience instability while they use the application or even downtime of the application. In order to reduce this bottleneck, data centers have used flash memory – attached to the server – to have the data of common usage available in lower time. This has reduced the I/O problem of finding and delivering data from SANs or DASs but still work needs to be done [23].

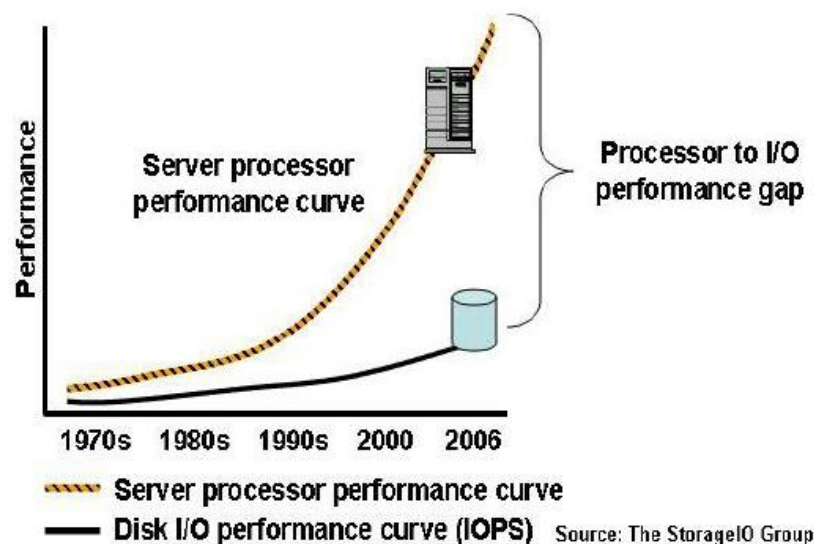
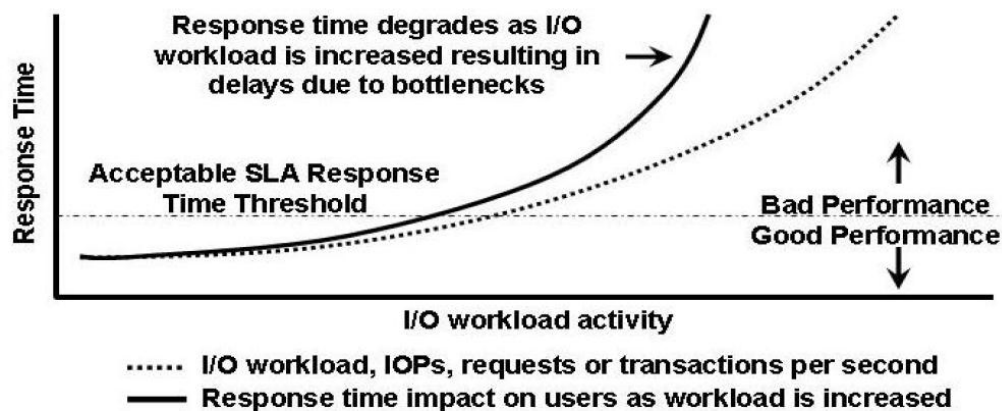


Figure 5 Difference in Server performance to I/O performance [7]

Except bottlenecks that can be created from the I/O differences in server and storage locations or unstable low bandwidth links, additional bottlenecks can exist due to the fact that the user's network (e.g. LAN, WAN), is overloaded and packets are dropped or there are delays due to the over-utilization of the existing network infrastructure. This bottleneck is something that is out of the responsibility of the cloud providers but it still can create reasons for bad user experience and reduced productivity.

One last reason for bottlenecks' creation is the seasonal or peak workload. During this times, data center reduce their response times as more tenants require more data and transactions and the system gets overloaded (see Figure 6).



Source: The StorageIO Group

Figure 6 Bottlenecks due to response time [7]

The outcome of this bottleneck is the same as in all of the above and that is that the customer will experience delays and the user experience will be reduced. All the aforementioned bottlenecks create some unwanted results for cloud providers and they try to reduce them since the QoS, QoE (Quality of Experience) and SLA agreements that they sign with the customers do force them to be stable and avoid dropped performance due to bottlenecks that are created mostly by – in reality – the use of the data center infrastructures from other tenants.

2.7 Intercloud communication

Cloud providers can have multiple data centers which have to be interconnected to provide a seamless experience to the consumers and re-direct VM instances or create extended VLANs. For all these to be done, the so called data center interconnect networks (DCINs) try to address this issue. This network is a layer 2 network and all it does is that it exchanges information across multiple data centers and enable a lot of mechanisms such as server migration or avoidance of loss of data due to disaster. Having this server duplication can be proven extremely helpful in the case of a disaster but, generally, this idea means that there has to be duplication in almost every little aspect of the server configuration in different DCs. Other mechanisms that interconnection of multiple data centers features are highly-available clusters, application mobility across different data center locations and balancing of the workload [4]. Since Layer 2 networks have some challenges when expanding outside of a data center, here are some challenges for L2 network expansions on more than one data centers.

- There has to be an effective mechanism that prevents from an End-to-End Loop across two intranets on different data centers when duplication takes place. There must be a monitoring mechanism that avoids endless loop of duplication and protects from topology changes that happen to one data center, to happen also to the other.
- The L2 expanded network has to be safe, thus cryptography in the data that are being exchanged has to be done and federal or regulatory needs have to be met.

- The core network has to be able to see this extension of the L2 network. This need is mostly needed for hybrid or private cloud deployment models.
- The load across WAN links has to be balanced therefore a mechanism is needed to dynamically alter workloads between different VMs.

2.8 Content Delivery Networks (CDNs)

Although this project's work is not about CDNs, there will be a small reference to them as they are used as a method for comparison to this project's proposal. CDNs or Content Delivery Networks is a relatively new type of network that came to play in 1998 and is used to deliver web services and content across the world in a more efficient and faster way than the traditional framework of a centralized server. CDNs are reducing the load of the actual server behind them by bringing the content to their infrastructure and provide it to clients without passing all the requests to the hidden server. CDNs actually help to reduce the latency as the load is distributed, thus each CDN server is faster and closer to the client's location. By having reduced latency, reduced load to the original server and multiple CDN servers to act as mirrors of the original one, it was possible to get higher throughputs and better accessibility to the website than before. Companies realized that and started investing on that as it is consisted only from Opex [24].

2.9 Domain Name System (DNS)

Another important side of the internet in general but also of this project has to do with DNS. Domain Name System was created to solve issues of creating a distributed database of where is what in the internet world. DNS uses three basic components to do this: domain name space and resource records, name servers and resolvers.

Domain Name Space is a tree structured space that holds data for e.g. hosts. These data are the resource records. This means that all the information that one queries the system to be provided with, coming from this tree structured database. Since this information has to be held somewhere, this job is being handled by the name servers. The name servers are servers distributed over the entire network and each one of them is the authority for a certain area of the domains that is called a zone. In this zone the server has the resource records that needed but also has a connection to other name servers where it redirects queries for domain spaces that it doesn't have knowledge about. The last component but also very crucial to the system is the resolver. A resolver is used to make sure that the query will be answered by a name server, regardless if the query has to be answered from the local name server or another one. It is a program and it can access at the least one name server and retrieve information on where to get the query answered if the data are not located there [25].

A type of Resource Record that has to be mentioned is the CNAME record or Canonical Name record. As said above DNS resolvers try to find the connection between a domain name and an IP to direct the client to the correct location of the website. CNAME records enable administrators to redirect users from different subdomains to the actual domain – which is called the A record –. Another usage of CNAME is to redirect a user to another domain which is located in another server.

Chapter 3 How this research contribution compares to other research work in the area

During the initial phase of this project many proposals and articles have been read around cloud optimization and reduction of the bottlenecks but two technologies found promising and more interesting than the rest. They set the foundations of this work and even inspired to go a step further. Thus, it is important to present NaaS and SDN which are connected. With SDN being the leader on this new way of seeing the networks, NaaS complements and improves certain aspects based on specific applications.

3.1 Network as a Service (NaaS)

All the services that are being provided through the cloud providers to the consumers do not give each tenant the ability to change or control one's network inside the DC [26]. This means that tenants have to rely on overlay networks for some sort of control. NaaS's proposal [26] gives an insight on how this ability to the tenants could become reality.

The authors propose a model that is based on intelligence in the switches throughout the data center network. All levels such as: access, aggregation and core have to have software & FPGA-based programmable switches. In this way the control over the data center network will be done.

The NaaS model [26] enables many improvements throughout the data center network as the network is not a black box anymore and alterations can be done to improve the overall view. The application-specific approach that they propose brings improvements to many areas such as decrease of the traffic. This outcome is achieved by introducing the routers and switches to the tenants and allowing them to control which applications have priority and when and how the packets will be moving around the network. Furthermore, they introduce caching mechanisms for some applications that use a certain number of information many times in a small time interval. The idea behind this mechanism is that since the network is not based on application layer, anymore, to be controlled, it is easier for the tenants to get some applications and tasks done in the network nodes as the packets are being transferred between two end hosts. They also support that due to content-centric networking, packets are being sent many times in order to reach all recipients and with NaaS this could be avoided by making copies of the packets at the last point before the node the two routes don't have the same direction anymore. This would reduce the usage of bandwidth and increase the utilization of the network. Generally, the idea of in-network processing would benefit in three major operations for packets:

- packets that are being forwarded: typically, a prioritization procedure
- packets that are being duplicated: a procedure that is being found on content-centric/content-based networking

- Packets that are being modified: a procedure that aids in e.g. stream processing and processes packets that come in random sequence and creates new packets or modifies the existing ones.

In-network processing can be beneficial for many different applications and in the end it could increase the throughput of the data center.

The authors show three important requirements in order for NaaS to be able to be implemented. These are the following ones:

- The DC should be able to adapt and integrate the NaaS technology with current hardware solutions and not expensive new incompatible ones.
- There should be high-level of programming in order for the tenants to be isolated from the basis of the DC network. Having all the way to the basis free network control would create a major security flaw, thus, the level of programming should be high and understandable for the tenants to use it.
- NaaS model requires for its success absolute isolation between tenants and in contradiction to the existing software-based solutions for routers, there should be different network resources provided to each tenant.

The authors propose an architecture that would give NaaS the flexibility it needs to function. In its networking device there should be the option of having parallel processing (i.e. more processor cores). This networking device would be called the NaaS box and will either be inside the switch hardware or connected through a fast link with it. NaaS boxes will have in-network processing elements (INPE's). These INPE's will have the same application logic on a number of the total number of NaaS boxes. NaaS boxes will connect the VM's instances of one tenant across different physical servers. Since NaaS boxes will have multiple cores, there can be multiple processing for various INPE's.

Concerning the programming model, there should be up to certain level of freedom for software developers (i.e. tenants) to be able and program their personal physical network. The model should also be malicious code proof, since the same hardware will be used for multiple tenants at the same time. The NaaS model also has an alternative for applications that are old or unaware of the model. It just maps them as traditional data and they simply do not take part in the rest of the NaaS framework.

Finally, the authors show through a simple flow-level simulation that even in this way the data center network can have significantly decreased completion times for NaaS aware applications. This will of course benefit even the non-NaaS aware applications as they get more bandwidth and more available resources to complete their tasks. The setup they have used for the simulation is an 8192-server fat tree topology that makes use of 32-port 1 Gbps switches and all switches contain NaaS boxes.

Overall, in [26] they argue that NaaS can work in reality although they observe that further research has to be done to improve scalability programmability and performance isolation [26]. The ideas presented in this paper [26] have given inspiration for this project and up to a level there are similar ideas presented here as in this paper.

3.2 Software Defined Networking (SDN)

As this project circulates around matters of an improved version of the current data center implementations, the state of the art work has to do with higher performance and I/Os as far as concerning inside the facilities but also outside of them. A general look of a new technology that co-exists with IP networks and improves the way packets are being exchanged is called SDN. Software Defined Networking decouples the data and control planes. This means that the network becomes more intelligent and it is decoupled from the applications. This separation enables building flexible and highly adaptable networks with the focus to each application's needs. SDN gives control to the developers over the network infrastructure and enables them to program it according to their needs [27]. SDN as a technology has more than one advantages. The following lines describe these advantages of SDN:

- SDN enhances a better management over the network and a better visibility into the resources that one uses, such as: storage, computational and the network as a resource itself [28].
- Since the data and the control plane are separated, this gives the option to minimize the operational cost of the DC infrastructure since consumers can program the way their network resources interact without the need of intervention by the infrastructure personnel [28].
- Network resources will be monitored and configured by multiple consumers simultaneously [27].
- The control over the entire network from the application owner gives the user a seamless feeling of functionality since the network's information is centralized and can be easily configured and monitored according to its preferences.
- Since the control on network devices is centralized, SDN enables and builds a safer and more secure environment.
- Similarly to NaaS, the network becomes active and plays a role in the partial processing of the packets and the traffic over the network.
- The programming from a global point to the whole network gives the option to the developers to create routing protocols that will be specified for each application usage.
- Since SDN management level has this holistic view of a network it can be programmed to choose an alternate path (backup path) to a destination if something fails. The important difference to the current non-SDN approach is that the configuration is centralized and takes place on a management server and not in every switch of the network.
- SDN can also coexist with current technologies (IP) and allow for improved network management but at the same time still remain compatible with the most known applications.

Most known way to implement an SDN is with the aid of OpenFlow. SDN-enabled switches implement the first forms of SDN with OpenFlow [29]. OpenFlow is a way to manage the information that have to be given to each individual networking element and create an abstraction of the networking environment to the programmer, so one can utilize the most out of the network without the need to go into details every time there is a need to program the

behavior of a new application. In Figure 7 one can see the existence of three layers for the total implementation of the SDN environment.

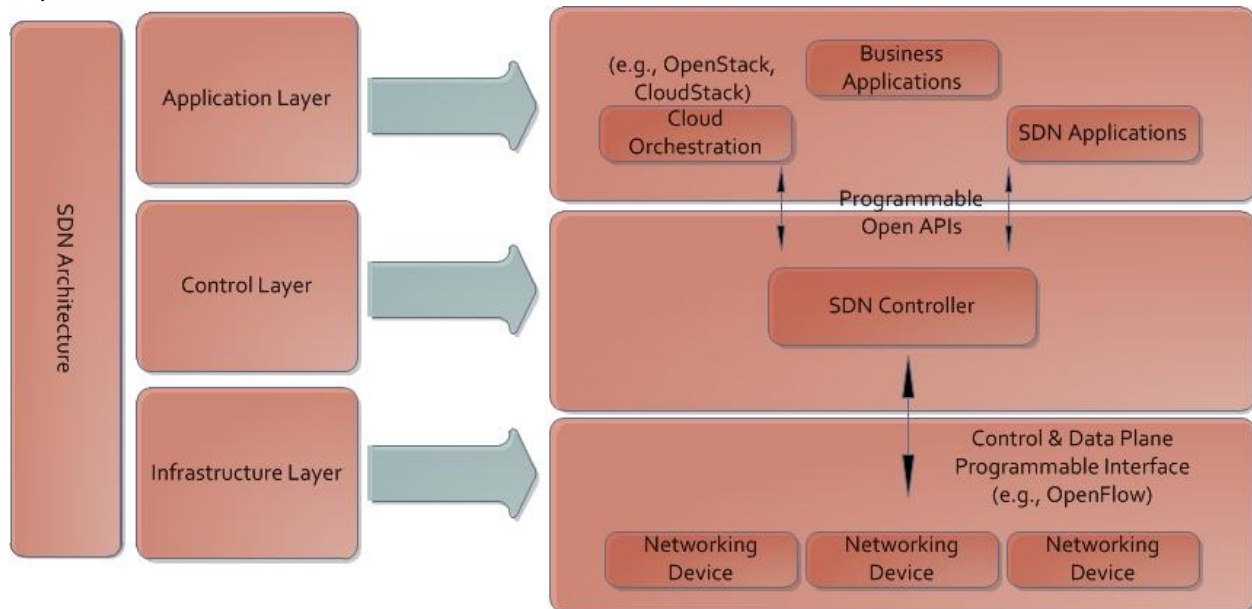


Figure 7 The SDN architecture⁴

As said above all the control that takes place in the networking devices is done by the SDN controller, an entity that receives information from all over the network. The applications that make use of the abstracted complexity of the SDN controller are on top of it. The communication between the SDN controller and the applications will be done through open APIs. A proposal from OpenFlow was made to combine the MPLS network with the SDN and OpenFlow. This gives new opportunities and could simplify the network and relieve it from the numerous protocols that run all together to keep all the different services running smoothly. In the diagram below (see Figure 8) one can see the architecture from the proposal of OpenFlow.

⁴ <http://h17007.www1.hp.com/images/solutions/technology/openflow/sdn-architecture.png>

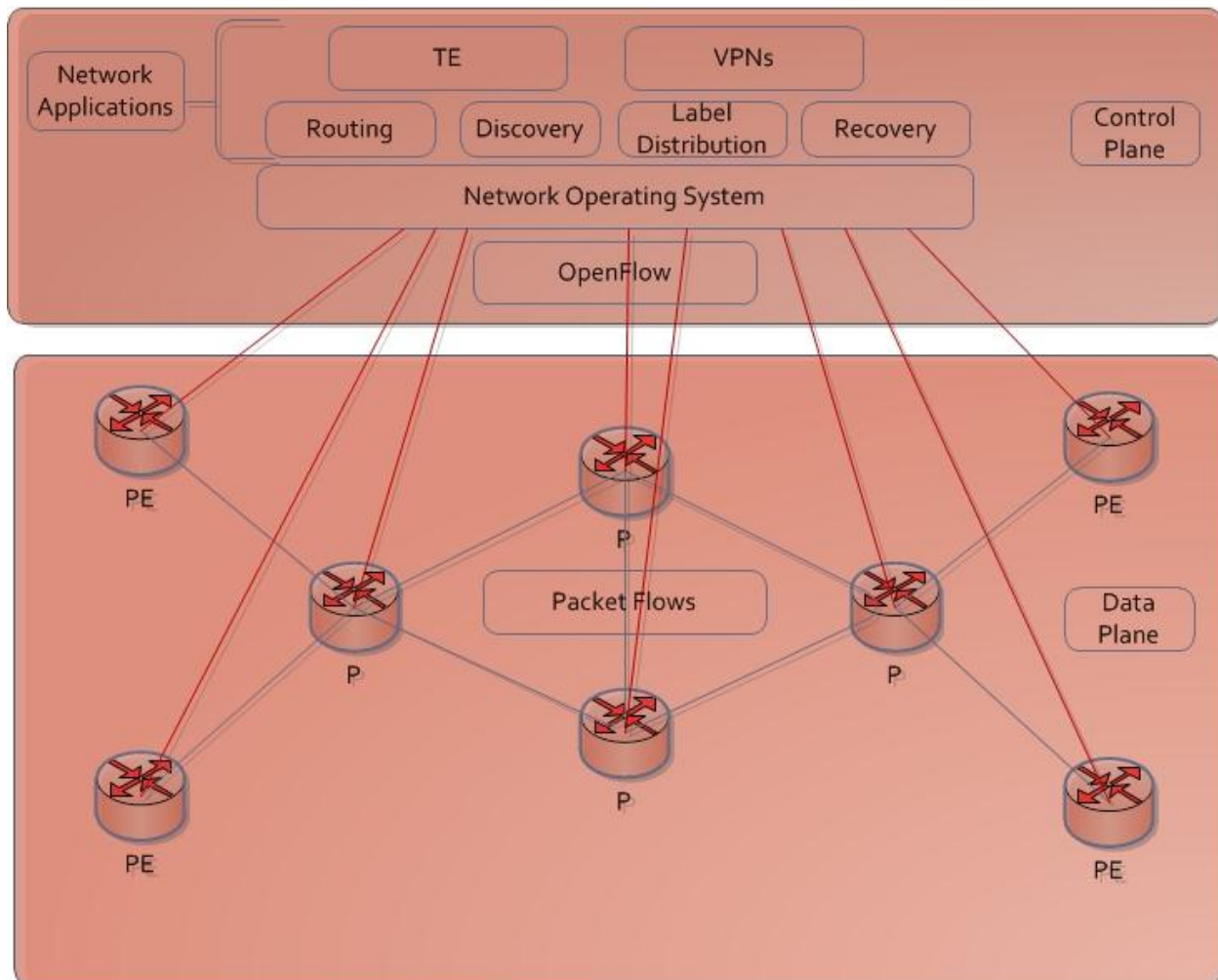


Figure 8 OpenFlow's proposal on combining MPLS with SDN & OpenFlow⁵

Software Defined Network is definitely not solving all the issues that exist in current networks but enhances the control over the network [30] and simplifies the procedure of manipulating the network as a new dimension to solve issues and not working from end to end as with older technologies. Thus, it is a motivation and an inspiration to the determination of this project's solution, as a state of the art approach to the radical change of control to the network.

⁵ http://www.openflow.org/wk/images/9/9d/SDN_based_MPLS.png

Chapter 4 Proposed Design & Implementation

The first two chapters provided the scientific background that will be needed for the presentation of the proposed algorithm together with the complete architecture and framework. Furthermore, the third chapter gave an insight to the starting point that has been set for the current proposal. In chapter four there will be an analysis of the theoretical framework and the design of this project's algorithm as a proposal to address some challenges concerning some specific use cases of small and medium-sized enterprises.

4.1 Theoretical Framework and Design of the Algorithm

As said before, enterprises can make use of the services provided by the cloud providers and enjoy the privileges of the cloud. Sometimes though, the privileges are not obvious, especially in the case of some enterprises [31]. Under these circumstances, in this project a proposal is made to use complementary the cloud provider with the small private data center that the enterprise already has. This additional cloud aid will take place only when the internal enterprise network is close to its limitations. QoS metrics will be monitored across the network to derive this information. The enterprise will then shift some of its applications (applications that would significantly free up a lot of network resources) to the cloud and ensure the correct utilization of the internal network but also the limited charge by using the pay on-demand policy when using the computational power of the cloud provider. This model is a form of a hybrid cloud. It has though differences since here it is proposed to shift some usability - according to the network's status - from the enterprise's intranet to the cloud and work via WAN or MAN. The hybrid cloud is more proactive in the way that it distributes applications between the private and the public cloud but the project's proposal is more reactive and thus more dynamic [4]. The process of migration to the cloud (the transition to a DC) will be happening in a randomized and distributed manner. This will help to avoid connecting a lot of tenants to the same data center - just because the DNS groups them to the closest locally DC or point of presence. The design of the scenario can be seen in Figure 1.

4.1.1 How SDN and HTTP redirection enables part of the process

But we shall start from the beginning of the concept. At the state of the art work, presented in the third chapter, some ideas were proposed on how the network could be monitored and utilized in a more efficient and application-specific way. In order for enterprises to create such a design in their own network there should be an entity, such as a NaaS box, that monitors the packets that are moved along the way and process them accordingly. Generally, this will be an SDN case and there will be an SDN controller that is responsible for all the on-

premises switches and will monitor the processes while they are running and exchanging data over the internal network. In this project the SDN controller will monitor an ecommerce website.

When the SDN controller monitors an application and the network reaches to a state when there are many clients connected (many concurrent client requests) to it and it is a peak time and everybody is using high I/O applications, the SDN controller will decide to move an application to the cloud. So, this entity will play the role of the network orchestrator and will control the availability of the resources according to the information that receives from the switches. This information can have the form of MIBs with the use of the SNMP standard [32]. Solutions like that are used today in the data centers to monitor the health of their network [33]. The task of making this forwarding – of the client requests – to the cloud will be aided by a program that makes DNS and HTTP redirections and load balancing. This program (such as DNS Made Easy [34]) can be controlled by the SDN controller's application layer via an API form and will enable the http requests of the customers to be forwarded to the cloud once the site is accommodated there during its peak time. In order to avoid duplicating content on the multiple locations of servers we are going to use an HTTP redirection. Since the current discussion is about a web server and an ecommerce website, this means that the content will make use of the HTTP protocol. This means that the HTTP redirection will be successful and the clients will be redirected to the DNS of the cloud infrastructures.

The HTTP though has two status requests for this task, 301 and 302. Status 301 means that the redirection to the new reference point will be permanent. This can be configured so that the client uses the new reference instead of the path of following the old one and then being redirected to the new. The 302 status is the chosen one for this use case. 302 status means that the redirection is temporary and provides the option to return to the initial domain name after a certain TTL [35]. After the TTL (Time To Live) passed the HTTP redirection program will change again the DNS entry and the customers will use the website from the on-premises servers. In order for the customers to not experience any difference in which URL they have to use, there will be a second URL for the website that functions from the cloud but it will be masked from the initial one. So, the customers will not experience anything different in the URL typing process [36].

The application that will run inside the SDN controller will be already programmed to have a list of the applications that have to migrate from the internal network to the cloud. In this use case the ecommerce website will have to migrate but generally there could be any application at its place or even multiple applications that migrate gradually until the SDN controller senses that the internal network is utilized at its full but not overloaded.

The idea is in a way similar to the concept of cloud bursting [8] however, the novelty in this research work is the use of an SDN network for on-premises application selection for migration. Further, in order to replicate that data to the cloud one has to make use of the project's proposed randomized routing algorithm to one of the available DCs that one has acquired resources.

The SDN controller will be updated, remotely, from the cloud provider as to which are the points of presence of the cloud provider (geographical location) or which are the available data centers. This list has to be open to changes from the cloud provider to mitigate the issue of uploading the latest update of the data to an overloaded data center or a data center that the cloud provider doesn't own resources but used to be.

4.1.2 An ecommerce website as the main example for the migration process

The ecommerce website has to migrate with its VMs and the data that are stored for those. Thus, the VM instances that serve the website's needs should not contain other applications, hence will be no complications to other instances or applications of the internal network. The problem of overloading appears during the peak time of the day. Thus, the application has to be made ready to move to the cloud without the need to change something. In other words, the website has to be programmed to be easily scalable and to function well in a cloud environment. In this way the VM instance will be ready to move when the internal network shows inability to server more clients. An ecommerce website has various servers that run continuously to keep the site up and running. The most important of these servers are:

1. Web Server
2. Application Server
3. File Server
4. Mail Server

A transaction system is used by a third party thus it is not mentioned as a server here. The data in an e-commerce solution are consisted of various parameters that have to be recorded and mostly are file servers, such as:

1. Database for Products
2. Database for Customers
3. Database for Transactions
4. Database with pictures of the Products

There are other issues as well that may arise during the making of the website or concerning the needs of it but these are the most crucial ones.

There are some things that yet haven't been presented such as the procedure of distributing different enterprises to different data centers - from the same cloud provider - to avoid bottlenecks from the side of data centers and even avoid centralizing all the migrations of many applications from different enterprises to one data center. Therefore, in this project another proposal is made to give an alternative to the DNS technique that is used by the cloud providers to allocate resources - on different data centers - based on the location of their customers and the location of their customers' clients [9], [37].

4.1.3 Theoretical background of the distributed discovery of the data centers

The way that is chosen to distribute enterprises across different data centers, during the migration process, in a randomized manner has its roots on the Fibonacci Spiral [38]. The Fibonacci spiral has some interesting characteristics that make it an option for the decentralized approach that enterprises will be distributed among different data centers (not necessarily the closest located to the companies' buildings). The idea behind the Fibonacci spiral came as a result of trying to see network traffic distribution in a different manner than the ones today. After all, the Fibonacci spiral can be found quite often in nature [39]. The main use of this spiral is to re-direct applications or websites to be served by an off-premises data center and not - necessarily- the one that is the closest to the enterprise building. The mechanism works as follows:

- the Fibonacci spiral is created geographically around the location of the enterprise
- a random number will decide the rotation of the spiral
- another random number will determine a point that will be pinpointed on the spiral line
- the closest data center to the random point will serve the enterprise

As written above the Fibonacci spiral will start with any angle and could potentially have any point chosen on it. This creates two degrees of freedom and makes it possible to pick potentially any point inside the spiral's radius. A third degree of freedom will be the ability to increase the size of this spiral so it covers greater distances around the enterprise's specific geographical point. Afterwards, the Fibonacci spiral will have done its task and can be regenerated when there is a need for a selection of another data center if the cloud provider shows – via the SDN application - that the selected data center has high latency and exceeds the maximum tolerance of the application or the website. Having the Fibonacci Spiral makes the opening around an area faster and greater due to the logarithmic nature of the spiral itself. That means that even with a small random number the chances for the random point to be far from the enterprise are enough, hence, the data center will not be the closest one. The following Figure 9 depicts in two different cases the above explanation.

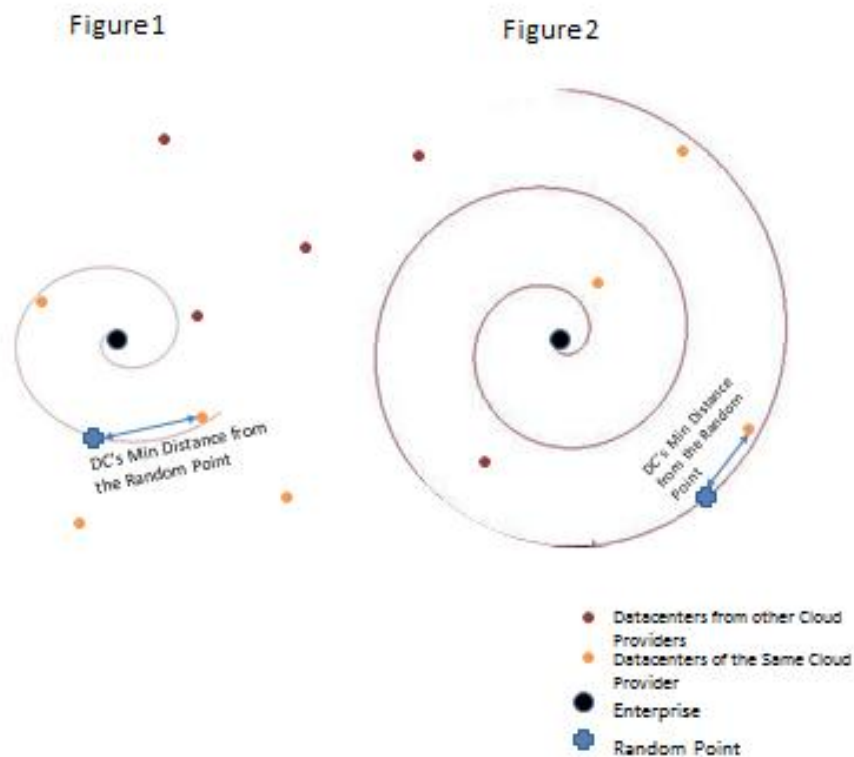


Figure 9 Randomized discovery of a DC with the proposed routing algorithm

In Figure 9 above one can see that the spiral changes size according to the area that a DC has to be discovered. These two different cases project also the difference in the initial angle of the

spiral and the random point that is given on this line. The closest DC from this “random point” is selected.

4.1.4 Differentiation from current approaches and rationale

Although this is a novel idea, there was the question as to why one should choose this novel proposal instead of using the DNS approach that is well-established and functional? The answer to that question lies in the very essence of the idea. The idea to have a randomized approach in routing is definitely not new but it has its advantages. According to [40] random processes in routing are used to increase performance and reduce the congestion in a network. Examples of random routing can be also found in wireless ad-hoc networks [41]. Basically, if one has to provision resources for making the migration to a data center this means that the same load will always be given to a specific DC, whereas if the process is random then these resources can be available and ready to be given to the first one that requests them. The Fibonacci spiral also has the aforementioned logarithmic scale in its opening. So it will widen in short distance around the enterprise. This will make it easier to choose (randomly) a data center that is far and possibly underutilized (if the enterprise is in an industrial area where many enterprises are collocated and the closest DC is over-utilized). This could create other issues and possibly other bottlenecks but this project is about an algorithm that functions with SDN networks and has the ability to be provided with a global view of the network. Under these circumstances – of a viewable network – resources can be utilized randomly and in a distributed manner and if something fails there will always be an alternative that can be found by the global network orchestrator and given as an option for the migration.

Another argument could be that there are not that many data centers to choose from (when it comes to a specific cloud provider). From a market point of view the data center and cloud industry has an annual growth and this will continue to go in that direction [42]. Since data centers evolve and can handle more difficult tasks, more demanding applications and many more clients and businesses, cloud providers will acquire more space to be available closer to more locations and with higher availability, we assume that this direction will lead to the building of many more. This gives new opportunities to have more data centers and more locations to choose from. This becomes even more interesting as Europe will probably facilitate more data centers in the future. According to a data center risk index study for 2013 [43], Nordic countries are going higher in the ranking, year by year, because of various reasons: political stability, ease of doing business, easier cooling abilities and many more. As stated in [44] Facebook and Google already have chosen Nordic countries to accommodate their large data centers and many more will start to add in this list.

Having more data centers provides with the right incentive to start higher pricing policies for those who want to be served by data centers that are closer to them (so, reducing the radius of the Fibonacci spiral) and have lower pricing policies for those that don't matter to have some distance. The pricing scheme could include this additional price in the data that are transferred from and to the data centers themselves. Although the differences to performance and latency are not expected to be great, real-time applications may benefit from this.

Another argument is that by using this randomized routing algorithm the network is being decentralized especially around dense urban areas. So the bottleneck of having a great load on a line to a certain direction is being mitigated; at least for the process of uploading or downloading data to and from the cloud for the migration process.

Later in this chapter the complete algorithm will be presented with all the procedure before and after the use of the Fibonacci spiral as well as the entities that will take care of the whole task.

4.2 On-premises infrastructure

Going one step further, in this project's use case, the architecture of the on-premises infrastructure is the following. It is a small enterprise data center that uses a NAS filer as a storage media system. The filer has a lot of SATA disks for low infrastructure cost. But the ability of SATA disks to keep performance high when there is too high demand from I/O requests is impossible. There are solutions (e.g. Fiber Channel HD's [45]) that can increase speed, hence, performance, but they are expensive [46] and the main goal in this project is to address performance issues by keeping storage disks as inexpensive as possible and by making as little alterations to the existing enterprise data center as possible. Since the data will also be locally stored there should be a device to replicate and backup data and databases to the cloud side and locally and keep them both up to date [47], [48] close to the migration process – from either side. By having all the data and databases to a remote storage device (i.e. cloud storage resource), the process of migrating to the cloud whenever the internal enterprise network can't serve more clients, becomes easier and faster. In the whole topology the SDN-enabled network controllers will monitor the rest of the switches (which will also be SDN enabled) of the internal network. This will be the way to monitor the packets from certain applications and the utilization of the complete network, as well as more parameters available from SDN for other tasks.

4.3 Algorithm Design

The whole framework above will follow a general pattern that is described by the following algorithm.

1. Serve the application/website users (employees and clients) from the on-premises enterprise data center

2. Keep data in the NAS filer and create a replication and then snapshots of the selected high I/O applications or the chosen ecommerce website on the cloud– Figure 10. The replication of data will take place every day a little before the migration from either side.

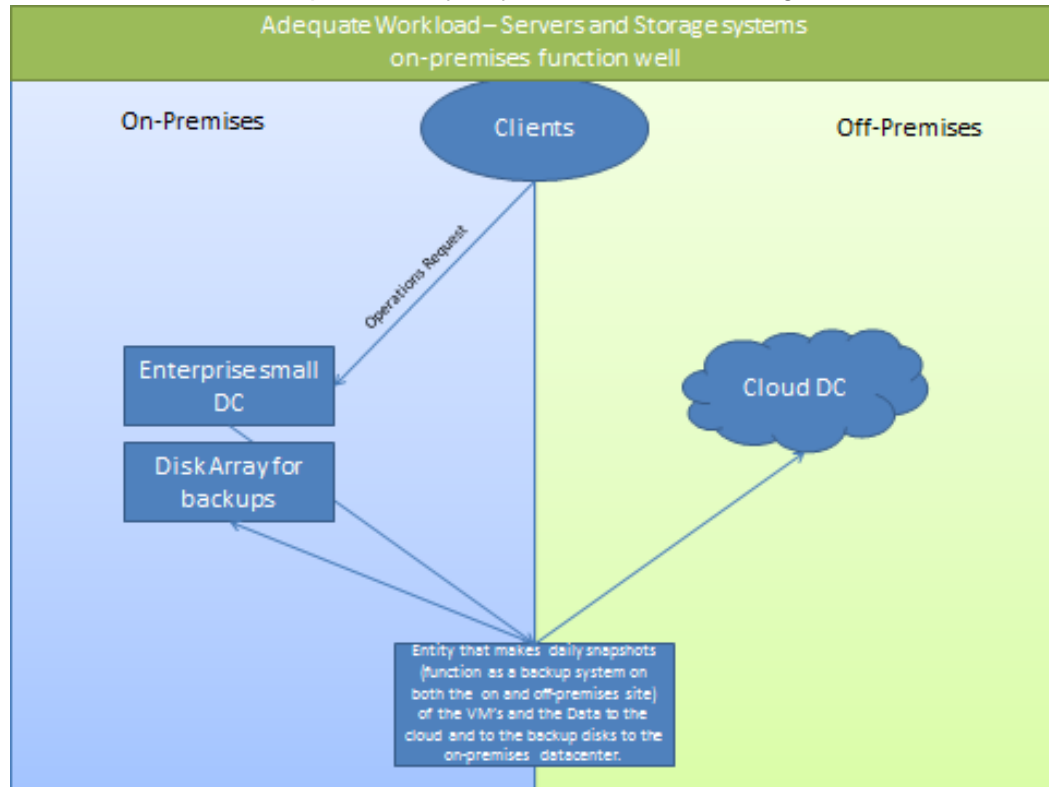


Figure 10 Standard procedure when everything function well on-premises

3. If the time reaches to the interval of the peak hours of the day
 - a. Check which pricing policy is the tenant/enterprise under (other characteristics than pricing mechanisms could also play a role in the decision below).
 - i. If they are on the high pricing policy, narrow the opening of the spiral
 - ii. If they are on the low pricing policy, widen the opening of the spiral
 - b. Request to receive the GPS coordinates and IPs (according to the number of the servers that have to be allocated) for available resources from all the DCs that the cloud provider has allocated resources to.
 - c. Run the proposed routing algorithm for the WAN network to find a data center
 - d. Check whether the chosen DC complies with the requirements (1.DC belong to the selected cloud provider's DC's, 2. Is it inside the range that the Fibonacci spiral expands, 3. It is the closest DC to the random point on the spiral line)
 - i. if yes, then send the latest updates for the VM instance and a data snapshot
 - ii. if not then re-run the proposed routing algorithm (go to step 3a) and exclude this DC for this tenant's (enterprise) migration case

(At this point the live migration is taking place and the users requests will be sent to the cloud infrastructures so the website can handle the additional workload – see Figure 11)

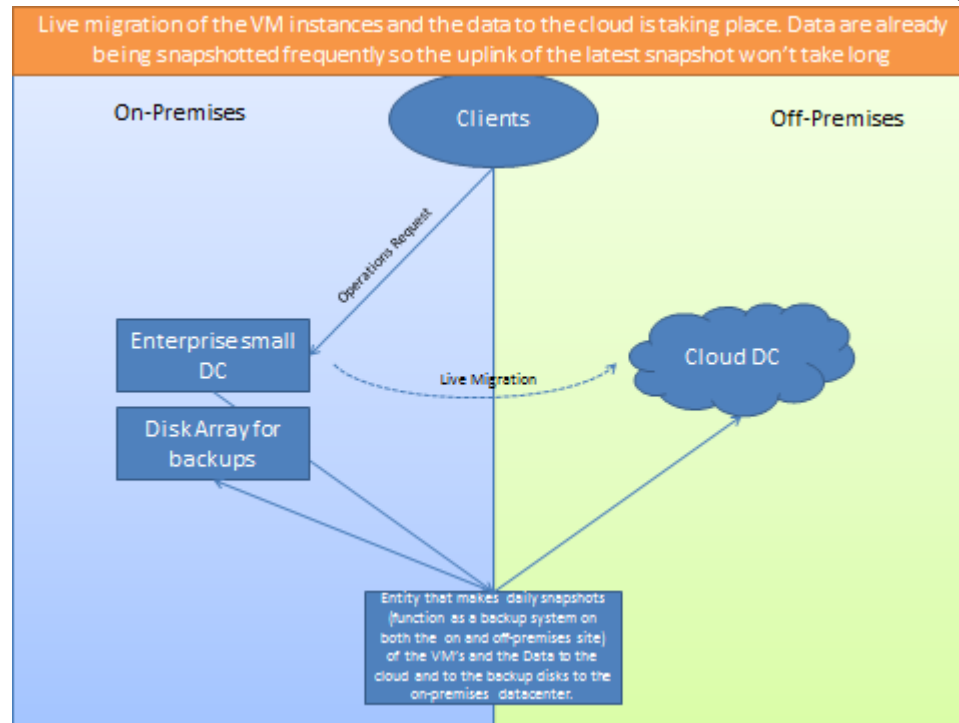


Figure 11 Migration process to the cloud infrastructures

4. Check again the on-premises QoS metrics with information retrieved from the SDN controller
 - a. If everything's ok then proceed to step 5
 - b. If not then re-run the algorithm and set more application off-premises from the selected ones to the same DC as the rest of the resources that have been allocated.
5. In the current case of an ecommerce website redirect all the client requests to the data center during the hours the website functions from off-premises by using HTTP redirection.
6. If the peak hours have passed, then migrate back the updated databases and snapshots of the instances and return the website applications to run on-premises again (this is to be done in order to avoid unnecessary charging from the usage of some cloud services)

4.4 Implementation

During the last sections there was a complete analysis on how the proposed algorithm will work. In this subchapter there will be a presentation of a program that was written to simulate how this algorithm will partly function. During this implementation phase some parts of the algorithm will be tested due to lack of time and devices to make the complete simulation. Later in the current subchapter there will also be a cost comparison of this framework in relation to other existing solutions in the market.

4.4.1 Program that simulates the code in the SDN controller

The algorithm has been implemented in Matlab and it approximates the functionality of the randomized routing algorithm with the Fibonacci Spiral and a simple data center selection process. This program was made to test how the process could work if it was implemented as an application on top of a real SDN-enabled controller. The program accepts as inputs the following:

1. The initial angle (given by a randomized movement of a mouse, for the purpose of testing)
2. The point on the spiral (given by a randomized movement of a mouse, for the purpose of testing)
3. The number of squares (more squares equals to wider spiral around the basis)
4. Geographical coordinates of the available data centers from the cloud providers
5. Geographical coordinates of the enterprise

These inputs are enough for the SDN controller to produce a random point on a Fibonacci spiral around the enterprise and then find the closest data center to that point and pass the information in the network and redirect the packets from the selected applications to the correct IP (these will be the instances that are accommodated in the data center's). In Figure 12 one can see the projected map and the location of the enterprise as well as the random point on the spiral line marked with a red 'x' mark (an arrow is pointing to the 'x' mark).

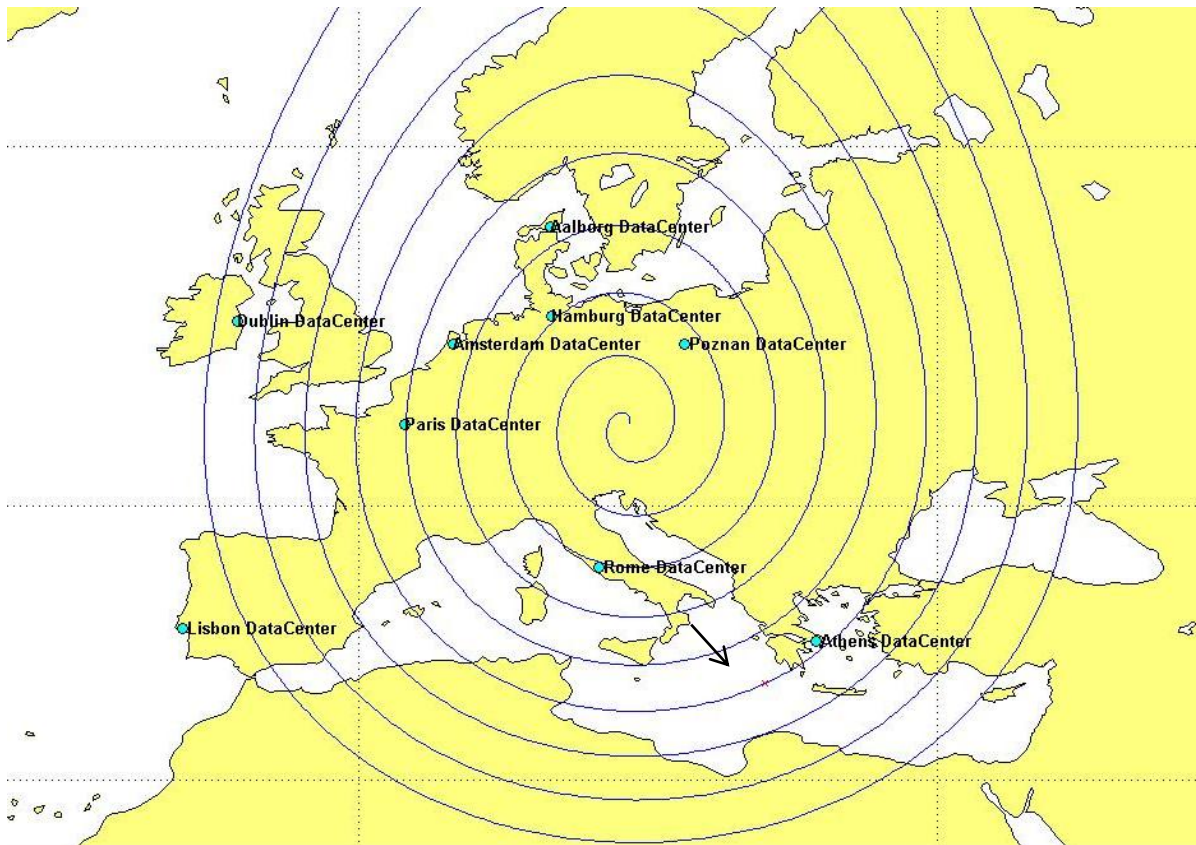


Figure 12 Execution of the algorithm for the discovery process for migration

A run of the algorithm decided the DC where the data will be replicated and the servers will start running, will be the one in Athens. The structure of the output of the program is presented like this:

```
RecommendedLink =
  Geometry: 'Point'
    Lat: 37.9791
    Lon: 23.7166
    Name: ' Athens DataCenter'
    InstanceIP: '31.217.191.255'
```

Although the enterprise is located in Czech Republic, the data center that is chosen is further than the closest one. Another run of the algorithm can give different results, as every time the process is random and the outcome can be any data center that accommodates servers owned by the cloud provider that the enterprise has a contract with. One more example can be seen in Figure 13 where the outcome is different. This time the radius of the Fibonacci spiral is much smaller and the chances of having closer the chosen data center to the location of the enterprise are bigger.

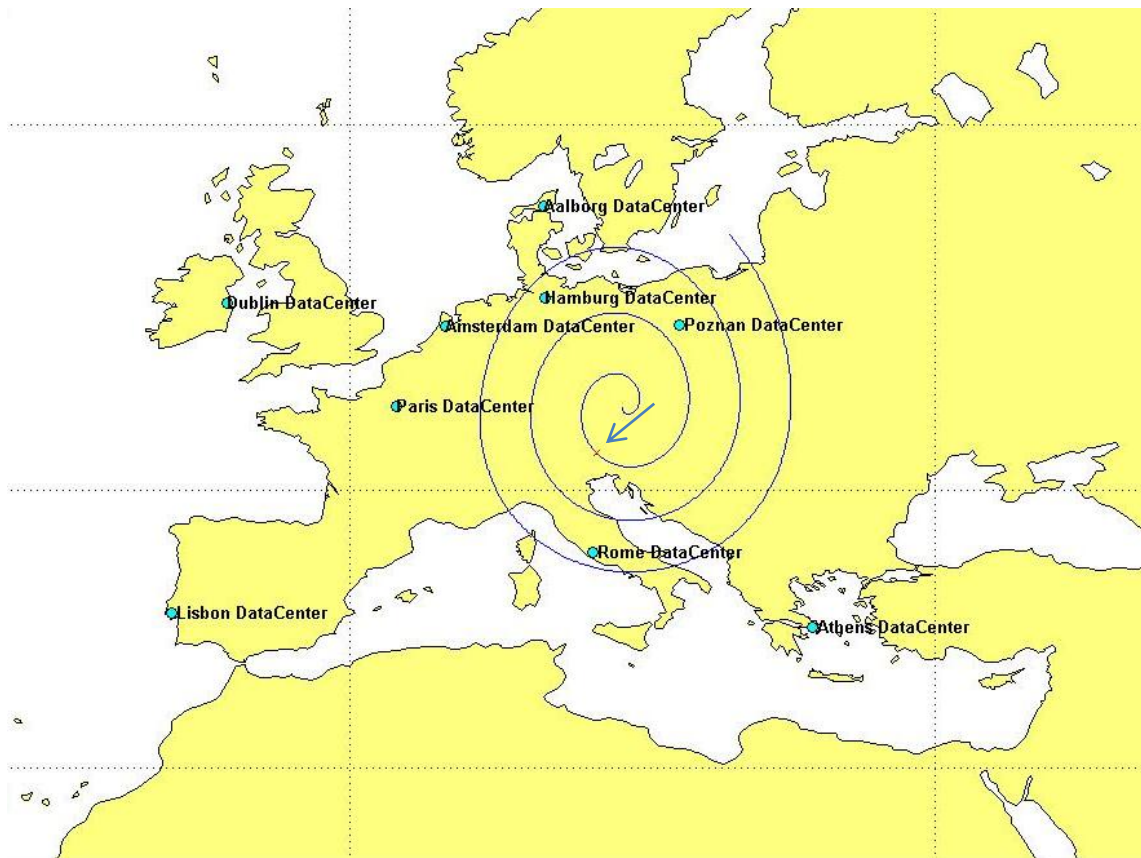


Figure 13 Another execution of the algorithm where the radius is smaller

The structure of the output will be the same, only the chosen DC and its data will be altered:

RecommendedLink =

Geometry: 'Point'

Lat: 41.8956

Lon: 12.4823

Name: ' Rome DataCenter'

InstanceIP: '91.218.224.5'

Out of this output one can retrieve the IP address/es of the instance/s (each IP for every server) and pass them to the redirection process – HTTP status 302. In this way latest updated data will migrate to the cloud and then the site will start functioning from this location for the very busy hours.

It should be mentioned again that the 4th input will be mostly controlled by the cloud provider. According to the ongoing work of ITU-T in FG Cloud Technical Report Part 3 (02/2012) [49] there has been a proposal that there will be a resource orchestrator that will have a global view of the data centers and information about resources and availability of them. From this entity the on-premises SDN controller will receive the information needed for the continuation of the migration process. The outputs of this program are nothing more than the geographical location of the data center as well as the IP (or IPs) that it is assigned to the instance/instances of the customer to the cloud resources. Of course this is a subprogram of a main program that it is initiated when it is the time interval during the peak hours of the day.

The code as well as small explanations along the lines is included in appendix 1 where one can refer to for more details on how the implemented part of the proposed idea, works.

4.4.2 Calculation and comparison of the proposed idea's cost to other solutions

In order to evaluate the project's proposed solution a cost analysis was performed concerning: the proposed solution, a CDN (see Table 4) and a complete cloud solution (see Table 2). This analysis will provide information about the different solutions and which is more appropriate for different circumstances.

The aim of this project is how to move between the on- and off-premises an e-commerce website and how clients' requests are being served either by the infrastructure of the company or the infrastructures of the cloud provider.

As stated before an e-commerce website consists of different types of servers which all should be updated every time before the redirection of the clients to the other location. These servers handle different tasks for the complete functionality of the website. Each type of server that one will need for an e-commerce website is mentioned below:

- a. Web server
- b. Application server
- c. File server
- d. Mail server
- e. Load balancer (between multiple servers of the same kind)

The Load Balancer, however, is not a server but it is crucial part – software-wise – when one has more than one servers, of the same kind, and needs to balance the load among them for a particular task (e.g. more file servers). Concerning the transaction system, there will be a connection with a third party to serve the needs of payments to our website, thus we do not need another type of server only an API connection, although all the transactions will be stored in one of the databases in the website's file server. The needs for a medium-sized ecommerce website are listed below in order to have a starting point for the calculations that will take place afterwards.

Table 1 Numbers that will be used for estimation of the costs

	Typical numbers for a medium-sized ecommerce website
Data Transfer	
BW in	10Mbps
BW out	10Mbps
Instances	
Instances/Server (File server, Mail server, Website Server) - If needed more than one	1
Size of each Instance	200GB (average)
Virtual Cores/instance	4 cores
RAM/instance	8GB
Storage	
Size of volume that data from DBs will be stored (medium-sized ecommerce website)	500GB
Ecommerce Website	
Number of Clients' Http Requests (medium-sized ecommerce website)	100 concurrent

The following assumptions concerning the beginning and duration of peak time have been made. If one assumes that the peak time of an e-commerce website is during 12:00 to 19:00 then one has seven hours to increase the resources so the client requests are served seamlessly and they won't experience delays or even downtime due to the limited, on-premises, data center. Another assumption is having 30 days per month to make the calculation. So, the peak hours per month will be 210 hours and the rest will be 510 hours and these hours the website can be accommodated and served by the servers, on-premises. This means that 29.17% of the month the website will be fully off-premises and to the cloud site. Although, only during this 29.17% the website will be accessed from off-premises, the storage resources on the cloud will be constantly used as the mirroring and snapshotting of the current state of data will be daily updated both on- and off-premises. This means that certain functions and services of the cloud infrastructures will be constantly used whereas others only during the peak time.

In these three different solutions there was an effort to have the same basis so the cost comparison has meaning and one can choose between different solutions with different pros and cons. It is not the project's goal to analyze the CDN option but one can refer to the second chapter for some basic information around CDNs. In the following lines three different tables are seen and each contains information about the total cost of the three different solutions including the project's proposed idea. The tool that is used can be found in [50] and retrieves information by major cloud providers. There is not complete accuracy to the calculations. These are given mostly to show a comparison of the different solutions and a comparative advantage of the one over the other. Thus, deployments' calculations were done with the same tool (in the proposed idea and the complete cloud migration) and the parameters were kept the same wherever there wasn't a need for changing.

Initially the calculation of the cost was performed for the solution of the completely cloud-based solution which is an IaaS service. Table 2 below presents the infrastructure that was calculated for acquiring in the data center. The numbers are an approximation for a medium-sized ecommerce website.

Table 2 Cost when migrating entirely to the Cloud

Deployment Summary: Cloud Complete Migration	
3 x Servers	3 x All servers - n1-highmem-2-d (n1-highmem-2-d) on Google Europe
1 x Storage	1 x storage (500.0GB) - Cloud Storage Standard on Google Europe
1 x Databases	1 x All DB's - D16 (Cloud SQL D16) on Google Europe
3 x Data Links	26000.0GB from All servers (Server) to storage (Storage)
	26000.0GB from storage (Storage) to All servers (Server)
	26000.0GB from All servers (Server) to All DB's (Database)
	26000.0GB from All DBs (Database) to All servers (Server)
	26000.0GB from Users (Remote Node) to All servers (Server)
	26000.0GB from All servers (Server) to Users (Remote Node)
Monthly Cost*	3915 USD
	*With Three Year Contract

It has to be clarified that the term “all servers”, refers to the web, application and mail server. There is no Load Balancer due to the lack of multiple servers of the same type. The file server has its own option – “all databases”. The “data links” refer to the acquired bandwidth for all the purposes in the website.

The second calculation was made for this project's proposed idea. This means that the cloud resources are not used entirely and continuously. Some of the resources remain active except the peak hours every day. This reduces the end price. On Table 3 one can see this observation.

Table 3 Calculated costs for the projects proposal

Deployment Summary: Proposed Scheme and Cloud	
3 x Servers	3 x All servers - n1-highmem-2-d (n1-highmem-2-d) on Google Europe (servers function for 210 hours/month)
1 x Storage	1 x storage (500.0GB) - Cloud Storage Standard on Google Europe
1 x Databases	1 x All DB's - D16 (Cloud SQL D16) on Google Europe
3 x Data Links	26000.0GB from All servers (Server) to storage (Storage)
	26000.0GB from storage (Storage) to All servers (Server)
	26000.0GB from All servers (Server) to All DB's (Database)
	26000.0GB from All DB's (Database) to All servers (Server)
	26000.0GB from Users (Remote Node) to All servers (Server)
	26000.0GB from All servers (Server) to Users (Remote Node)
Monthly Cost*	3076 USD
	*With Three Year Contract

Although BW needs can be reduced (since the website is used from the cloud provider's infrastructures only some hours per day) it will not. This is because there is a need to show that even with no other reduction than the time using the computing resources; the price can drop at about 24% of the complete cloud migration. The only disadvantage is that with this project's solution there is a need for Capex. Capex in the proposed solution is the sunk cost of a SDN controller (can handle up to 100 switches [51], which is high enough for a medium sized enterprise), a device that creates on- and off-premises replicates of the data and the instances and some extra Opex which is the cost of the IT personnel to set them up and a DNS annual subscription to a DNS provider for the HTTP redirecting of the website.

The last but not least calculation is about the CDN solution. In this solution there is only a bandwidth and a storage cost. In Table 4 one can see the amount of traffic and storage as well as the price according to [52] pricing schemes.

Table 4 Calculation of the CDN solution

Services	
Bandwidth	26 TB
Storage	500 GB
Monthly Cost	3495 USD

Important to note here that in the CDN case, there is no three year contract but a pay-as-you-go policy. In Table 5 one can see all the costs (Capex and Opex) from all the solutions that were presented.

Table 5 Presentation of all the solutions and their costs

Different Deployment Solutions	Cost per month for each deployment model
Complete Cloud migration Solution	3915 USD
CDN Solution	3495 USD
Project's Proposed Solution	3076 USD (excluded the additional sunk Capex cost (an SDN controller, SDN-enabled switches, device for replication production))

Even though this project's proposal has some device that needed to be installed first (i.e. the sunk cost), it has the lowest charge per month from the other solutions.

Chapter 5 Conclusions & Future work

This Chapter presents a review of the work done and the conclusions drawn. An approach on how to deal with a certain type of bottleneck that is created in the data centers has also been recommended.

5.1 Conclusions

This research work is a novel combination of solutions and a novel randomized routing algorithm for distributing traffic across different data centers. An e-commerce website was in the usage scenario and it was shown how to migrate it to the cloud for a specific amount of time and try to keep it as clear from duplications as possible. An HTTP redirection mechanism was used to avoid having different data on and off-premises and a restriction was made to the functionality from both sides simultaneously. The proposed hybrid solution has been compared to other existing solutions that have big shares in the market, such as the cloud and CDNs. In Chapter 4 one can find the tables that depicting this cost comparison and the financial gain out of using the proposed hybrid model. The outcome of this research does not try to show that this mechanism can replace the well-established CDNs but to give an alternative to small-sized and medium-sized businesses that wish to use their small data center in conjunction with the cloud and make the cloud an extension to their resources when needed. The goal was also to propose a different approach for the cloud providers and data center engineers, to try a new alternative in decentralizing the tenants of the cloud and use a randomized routing scheme. The project presents simple ideas that – except SDN that still is in development – already exist out there and the combination of different mechanisms makes it unique as an approach. Furthermore, the presence of the SDN controllers in the architecture opens up new ideas that can go this work one step further on how to work around problems such as the network bottlenecks and even business opportunities. A first effort to mitigate the bottleneck in a data center I/O capability was made by creating this randomized routing algorithm that is used via the application layer of an SDN controller and retrieves information from both the on-premises network but also from the cloud provider.

The global orchestrator can also put an additional fee (in BW) to those who will use a data center that is close to their location to replicate their data before migration and the ones that are served from a data center that is further and might have a relatively lower performance but the price will also be lower. This creates benefits for both the contractor (e.g. the cloud provider) and the tenant (the enterprise). Additional benefit for the enterprise can be that the storage media can be increased for storage issues without worrying about I/Os since there will always be the cloud to support them for situations when overloading is unavoidable and performance is critical.

5.2 Future work

To go one step further, the proposed and analyzed framework should be implemented. Due to time constraints, mainly, as well, as the lack of proper evaluation tools and an operating business environment this work could not include this step. The SDN technology still needs a while to evolve in order to have this idea put to the test in a real business environment. Although, there are SDN controllers out there commercially available but prices are still high and they will become more competitive to existing already established solutions. So the future work mostly is based on making this project actually work and to put the aforementioned ideas and theories to practice and see whether the outcome will be equally good as argued.

Except this project's research goal there are other large projects that also try to mitigate the effect of bottlenecks to performance. One such research is conducted by IBM. IBM researches the topic of Nano photonics to a scale that can be used in processors. According to IBM's research [53], processors in the near future can use Nano photonics to communicate with each other or with the memory. This increases substantially the speed between the circuits as they are not only electrical anymore rather than mixed with optical parts. A crucial bottleneck in circuits is the low performance due to the time it takes for a signal to be transmitted electrically even between components that have small distance. IBM combined processors with converters and now they can alter electrical signals to optical and vice versa, so as they can be integrated to complex and powerful systems – such as supercomputers and data center infrastructures – so as to improve them even more. Technologies such as the aforementioned one push performance to the limit and in the coming decades probably the bottlenecks will move to different locations in the whole architecture from the current ones.

Appendix

Appendix 1

Appendix 1 includes the Matlab code for the program that searches – in a randomized manner – for a data center around the point of interest (that will be the enterprise that wishes to migrate applications or their website to the cloud. This program is consisted out of three different ones (Fibonacci Spiral, randompoints and association) that cooperate and produce the results that can be found in Chapter 4. It has to be noted that the first program that creates the Fibonacci Spiral is made by [52] and has been modified – under the GNU LGPL License in this project in order to deliver the purpose that it was needed.

```
function [x4,y4]= FibonacciSpiral ( n, thetha,lonx,laty )

    phi = ( 1.0 + sqrt ( 5.0 ) ) / 2.0;

% SCALE controls how many steps we take between the actual points.

    scale = 150.0;

% Allocate storage for the intermediate data.

    n2 = scale * ( n - 1 ) + 1;
    x2 = zeros ( n2, 1 );
    y2 = zeros ( n2, 1 );

    x4 = zeros ( n2, 1 );
    y4 = zeros ( n2, 1 );

% Set the angle and radius of the first point.

    a = 0.0;
    r = 0.0;

% Set the increments.

    da = 2.0 * pi * ( phi - 1.0 ) / phi;
    dr = 1.0;

    da = da / scale;
    dr = dr / scale;

% This is done to make the points move according to the initial random
% angle
    for i = 1 : n2
```

```

a = mod ( a + da, 2 * pi );
r = r + dr;

x2(i) = r * cos ( a );
y2(i) = r * sin ( a );

x4(i)=lonx+x2(i)*cos(thera)-y2(i)*sin(thera);
y4(i)=laty+x2(i)*sin(thera)+y2(i)*cos(thera);

end

return

end



---



function [y,x,dif] = randompoints (n,lonx,laty)

dis=0;

while dis==0 % this is done to wait for a random cursor movement in order to
              % provide a random angle and point
coord=get(0,'PointerLocation');
pause (2)
coord2=get(0,'PointerLocation');
dif=abs(coord-coord2);
dis=all(dif);
end

dif(1)=dif(1)*n/2; % to scale the moving of the point from the cursor
movements
[x y]= FibonacciSpiral (n,dif(2),lonx,laty);
hold all

plot ( x, y, 'r-' )

plot (x(dif(1)),y(dif(1)),'o') %Selection of the random point on the spiral
axis equal

print(gcf, '-dpng', 'fibonaccispiral.png', '-r 92')

hold off

```

```

function [RecommendedLink] = mapassociation (n,lonx,laty)
%In this function there is the association of the Fibonacci spiral, the
%random points and the geographical world map. After some calculations and
%the outcome is the chosen data center for the data migration. It has to be
%noted that during the calculations there can be more criteria, that
%eventually exclude some DC options. But in the current version there are
%no minimum requirements, except that the DCs all have space for a specific
%cloud provider.

[y x p]= randompoints (n,lonx,laty);

latparis = 48.87084; lonparis = 2.41306; % Paris coords
latsant = -33.36907; lonsant = -70.82851; % Santiago
latnyc = 40.69746; lonnyc = -73.93008; % New York City
latdallas = 32.7828; londallas= -96.8039; % Dallas
lataalborg = 57.09; lonaalborg = 9.97; % Aalborg
latdublin= 53.34419 ; londublin=-6.26751 ; % Dublin
latamsterdam=52.373444 ; lonamsterdam=4.892229 ; % Amsterdam
lathamburg= 53.55686600 ; lonhamburg =9.99462200 ; % Hamburg
latlisbon= 38.707126 ; lonlisbon = -9.135499; % Lisbon
latpoznan= 52.416334 ; lonpoznan = 16.925393 ; % Poznan
latrome=41.895623 ; lonrome =12.482269 ; % Rome
latathens=37.979137 ; lonathens= 23.716580; % Athens

latfib = y; lonfib = x;

[Cities(1:12).Geometry] = deal('Point');
[Fib.Geometry] = deal('MultiPoint');

Fib.Lat = latfib; Fib.Lon = lonfib;

Cities(1).Lat = latparis; Cities(1).Lon = lonparis;
Cities(2).Lat = latsant; Cities(2).Lon = lonsant;
Cities(3).Lat = latnyc; Cities(3).Lon = lonnyc;
Cities(4).Lat = latdallas; Cities(4).Lon = londallas;
Cities(5).Lat = lataalborg; Cities(5).Lon = lonaalborg;
Cities(6).Lat = latdublin; Cities(6).Lon = londublin;
Cities(7).Lat = latamsterdam; Cities(7).Lon = lonamsterdam;
Cities(8).Lat = lathamburg; Cities(8).Lon = lonhamburg;
Cities(9).Lat = latlisbon; Cities(9).Lon = lonlisbon;
Cities(10).Lat = latpoznan; Cities(10).Lon = lonpoznan;
Cities(11).Lat = latrome; Cities(11).Lon = lonrome;
Cities(12).Lat = latathens; Cities(12).Lon = lonathens;

Fib.Geometry='comp';
Cities(1).Name = 'Paris DataCenter';
Cities(2).Name = 'Santiago DataCenter';
Cities(3).Name = 'New York DataCenter';
Cities(4).Name = 'Dallas Datacenter';
Cities(5).Name = 'Aalborg DataCenter';
Cities(6).Name = 'Dublin DataCenter';

```

```

Cities(7).Name = 'Amsterdam DataCenter';
Cities(8).Name = 'Hamburg DataCenter';
Cities(9).Name = 'Lisbon DataCenter';
Cities(10).Name = 'Poznan DataCenter';
Cities(11).Name = 'Rome DataCenter';
Cities(12).Name = 'Athens DataCenter';

Cities(1).InstanceIP = '37.59.232.43';
Cities(2).InstanceIP = '200.91.47.255';
Cities(3).InstanceIP = '32.106.39.255';
Cities(4).InstanceIP = '62.68.93.255';
Cities(5).InstanceIP = '193.181.13.255';
Cities(6).InstanceIP = '62.221.7.167';
Cities(7).InstanceIP = '77.109.81.255';
Cities(8).InstanceIP = '62.134.128.31';
Cities(9).InstanceIP = '176.31.8.247';
Cities(10).InstanceIP = '46.105.190.223';
Cities(11).InstanceIP = '91.218.224.5';
Cities(12).InstanceIP = '31.217.191.255';

dist = zeros ( length(Cities), 1 );

hold all

axesm('mercator','grid','on','MapLatLimit',[-75 75]); tightmap;
geoshow('landareas.shp')
lat=y;
lon=x;
lat1=y(p(1));
lon1=x(p(1));
geoshow(Cities,'Marker','o','MarkerFaceColor','c','MarkerEdgeColor','k');
geoshow(lat,lon)
geoshow(lat1,lon1,'DisplayType','Point','Marker','x')

textm([Cities(:).Lat],[Cities(:).Lon],...
      {Cities(:).Name}, 'FontWeight','bold');

    for i=1:length(Cities)
        dist(i)=distance(lat1,lon1,Cities(i).Lat,Cities(i).Lon);
    end
    d=min (dist);
    [l]=find(dist==min(d));

    RecommendedLink=Cities(l);

hold off
return

```

Bibliography

1. **Beach, Thomas E.** Types of Computers. *University of New Mexico - Los Alamos*. [Online] January 24, 2004. [Cited: May 30, 2013.] <http://www.unm.edu/~tbeach/terms/types.html>.
2. **Baker, Mark.** Cluster Computing White Paper. *Cornell University Library*. [Online] December 28, 2000. [Cited: May 30, 2013.] <http://arxiv.org/ftp/cs/papers/0004/0004014.pdf>.
3. **Jazz Wang, Yao--Tsung Wang.** Cluster, Grid and Cloud Computing. *DRBL*. [Online]
4. **Escalante, Borko Furht · Armando.** *Handbook of Cloud Computing*. s.l. : Springer Science+Business Media, 2010. 978-1-4419-6523-3.
5. *To Move or Not to Move: The Economics of Cloud Computing.* **Tak, Byung Chul and Uргаonkar, Bhuvan and Sivasubramaniam, Anand.** Portland, OR : USENIX Association, 2011.
6. **LeClaire, Jennifer.** So, Is the Cloud Secure or Not? IT Managers Wary. *News Factor*. [Online] April 2, 2013. [Cited: May 30, 2013.] http://www.newsfactor.com/news/So--Is-the-Cloud-Secure-or-Is-It-Not-story.xhtml?story_id=001000001VHE.
7. **Schulz, Greg.** Data Center I/O Performance Issues and Impacts. *StorageIO blog*. [Online] August 7, 2006. [Cited: March 17, 2013.] http://storageio.com/Reports/StorageIO_WP_080706_Cover.pdf.
8. **Chandrasekhar, Bharath.** What is Cloudbursting. *Trend Cloud Security Blog*. [Online] March 15, 2011. [Cited: May 23, 2013.] <http://cloud.trendmicro.com/what-is-cloudbursting/>.
9. **Rajkumar Buyya, James Broberg, Andrzej M. Goscinski.** *Cloud Computing: Principles and Paradigms*. s.l. : Wiley, 2011. 0470887990.
10. National Institute of Standards and Technology. *National Institute of Standards and Technology*. [Online] May 27, 2013. <http://www.nist.gov/index.html>.
11. **Peter Mell, Timothy Grance.** The NIST Definition of Cloud Computing. *National Institute of Standards and Technology*. [Online] September 2011. [Cited: March 20, 2013.] <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>. National Institute of Standards and Technology Special Publication 800-145.
12. **Leslie Willcocks, Will Venters, Edgar A. Whitley.** Meeting the challenges of cloud computing. *Accenture*. [Online] Accenture, May 2011. [Cited: April 10, 2013.] <http://www.accenture.com/us-en/outlook/Pages/outlook-online-2011-challenges-cloud-computing.aspx>.
13. **TOONK, ANDREE.** HOW OPENDNS ACHIEVES HIGH AVAILABILITY WITH ANYCAST ROUTING. *Umbrella Security Labs*. [Online] January 10, 2013. [Cited: February 28, 2013.] <http://labs.umbrella.com/2013/01/10/high-availability-with-anycast-routing/>.
14. **Bridgwater, Adrian.** Just how big is "big data" anyway? *CWDN*. [Online] June 7, 2012. [Cited: March 3, 2013.] <http://www.computerweekly.com/blogs/cwdn/2012/06/just-how-big-is-big-data-anyway.html>.
15. **Scarpatti, Jessica.** Big data analysis in the cloud: Storage, network and server challenges. *Search Cloud Provider*. [Online] July 2012. [Cited: April 20, 2013.] <http://searchcloudprovider.techtarget.com/feature/Big-data-analysis-in-the-cloud-Storage-network-and-server-challenges>.

16. **Trappler, Thomas.** If It's in the Cloud, Get It on Paper: Cloud Computing Contract Issues. *EDUCAUSE*. [Online] June 24, 2010. [Cited: May 10, 2013.]
<http://www.educause.edu/ero/article/if-its-cloud-get-it-paper-cloud-computing-contract-issues>.
17. Data Center Multi-Tier Model Design. *Cisco*. [Online] [Cited: April 20, 2013.]
http://www.cisco.com/en/US/docs/solutions/Enterprise/Data_Center/DC_Infra2_5/DCInfra_2.html.
18. **Hedlund, Brad.** Top of Rack vs End of Row Data Center Designs. *Brad Hedlund*. [Online] April 5, 2009. [Cited: March 10, 2013.] <http://bradhedlund.com/2009/04/05/top-of-rack-vs-end-of-row-data-center-designs/>.
19. **Onlamp.** Using SANs and Nas. *Onlamp*. [Online] March 14, 2002. [Cited: March 22, 2013.]
<http://www.onlamp.com/2002/03/14/sansnas.html>.
20. **Garth A. Gibson, Rodney Van Meter.** Network Attached Storage Architecture. *COMMUNICATIONS OF THE ACM*. November, 2000, Vol. 43, No.11.
21. **Jon Tate, Fabiano Lucchese, Richard Moore.** Introduction to Storage Area Networks. *ibm.com/redbooks*. [Online] July 2006. [Cited: April 17, 2013.]
<http://itcertivnetworking.riverinainstitute.wikispaces.net/file/view/Red+Book+-+What+is+a+SAN.pdf>.
22. **School of Medicine, University of Miami.** Storage Area Network (SAN) Technology. *School of Medicine, University of Miami*. [Online] [Cited: March 8, 2013.]
<http://it.med.miami.edu/x443.xml>.
23. **Richardson, Jeff.** Avoiding “Whack-A-Mole” in the Data Center. *datacenterjournal*. [Online] September 10, 2012. [Cited: March 10, 2013.] <http://www.datacenterjournal.com/it/avoiding-whack-a-mole-in-the-data-center/>.
24. *Content delivery networks: status and trends.* **Vakali, A. and Pallis, G.** 6, s.l. : Internet Computing, IEEE , 2003, Vol. 7. DOI 10.1109/MIC.2003.1250586 .
25. **Mockapetris, P.** RFC 1034. *IETF*. [Online] November 1987. [Cited: May 15, 2013.]
<http://tools.ietf.org/html/rfc1034>.
26. *NaaS: Network-as-a-Service in the Cloud.* **Paolo Costa, Matteo Migliavacca, Peter Pietzuch, Alexander L. Wolf.** 2012.
27. **Foundation, Open Networking.** Software-Defined Networking: The New Norm for Networks. *Open Networking Foundation*. [Online] April 13, 2012. [Cited: May 2, 2013.]
<https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>.
28. **Greenfield, David.** Six Benefits of Software Defined Networking. *WANSPEAK*. [Online] October 1, 2012. [Cited: May 10, 2013.] <http://blog.silver-peak.com/six-benefits-of-sdn>.
29. **NOX.** NOX. *NOX*. [Online] [Cited: April 28, 2013.] <http://www.noxrepo.org/>.
30. **Shenker, Scott.** An attempt to motivate and clarify Software-Defined Networking (SDN). Berkley : Ericsson Research and Technology Day, May 3, 2011.
31. **Martin, James A.** Should You Move Your Small Business to the Cloud? *PCWorld*. [Online] January 29, 2010. [Cited: March 18, 2013.]
http://www.pcworld.com/article/188173/should_you_move_your_business_to_the_cloud.html?page=2.

32. **M. Rose, K. McCloghrie.** RFC 1065 - Structure and Identification of Management Information for TCP/IP-based Internets. *IETF*. [Online] May 1990. [Cited: April 21, 2013.] <http://www.ietf.org/rfc/rfc1155.txt>.
33. **WTI.** What Can SNMP MIB Support Do for Your Data Center? *WTI*. [Online] August 2010. [Cited: May 12, 2013.] <http://newsblog.wti.com/index.php/2010/08/what-can-snmp-mib-support-do-for-your-data-center/>.
34. **DNS Made Easy.** *DNS Made Easy*. [Online] [Cited: May 20, 2013.] <http://www.dnsmadeeasy.com/>.
35. **R. Fielding, et al.** Hypertext Transfer Protocol -- HTTP/1.1, RFC 2616. *IETF*. [Online] 1999 June. [Cited: May 10, 2013.] <http://tools.ietf.org/html/rfc2616#section-10.3.2>.
36. **Easy, DNS Made.** HTTP Redirection Record. *DNS Made Easy*. [Online] [Cited: May 21, 2013.] <http://help.dnsmadeeasy.com/record-entry/http-redirection/>.
37. **Bolt-DNS.** *CloudAccess*. [Online] [Cited: March 15, 2013.] <http://www.cloudaccess.net/products/bolt-dns.html>.
38. **Fibonacci Spiral.** *Oracle ThinkQuest*. [Online] 1999. [Cited: February 15, 2013.] <http://library.thinkquest.org/27890/theSeries6.html>.
39. **Reich, Lee.** Nature follows a number pattern called Fibonacci. <http://phys.org/>. [Online] February 20, 2013. [Cited: February 24, 2013.] <http://phys.org/news/2013-02-nature-pattern-fibonacci.html#nRlv>.
40. **Chau, Sid C-K.** Randomized Algorithms: Applications & Principles, Part 2: Random Routing and Load Balancing . *University of Cambridge*. [Online] [Cited: May 20, 2013.] http://www.cl.cam.ac.uk/teaching/1011/CompSysMod/RandBits_Lec2V2.pdf.
41. *Doppler measurements rendering random routing.* **Dong, Liang.** s.l. : Signals, Systems and Computers, 2008 42nd Asilomar , 2008. Digital Object Identifier: 10.1109/ACSSC.2008.5074539 .
42. **Columbus, Louis.** Gartner Predicts Infrastructure Services Will Accelerate Cloud Computing Growth. *Forbes*. [Online] February 19, 2013. [Cited: May 30, 2013.] <http://www.forbes.com/sites/louiscolumbus/2013/02/19/gartner-predicts-infrastructure-services-will-accelerate-cloud-computing-growth/>.
43. **Cushman & Wakefield LLP, hurleypalmerflatt and Source8.** *Data Centre Risk Index 2013*. 2013.
44. **Jones, Penny.** Nordics continue rise as low-risk data center location. *Data Center dynamics*. [Online] May 2013. [Cited: May 23, 2013.] <http://www.datacenterdynamics.com/focus/archive/2013/05/nordics-continue-rise-low-risk-data-center-location>.
45. **Fibre Channel Hard Drives.** *The hard drive guide*. [Online] [Cited: April 4, 2013.] <http://www.jbmacat.com/fibre-channel.html>.
46. **Mike Karp, Network World.** Comparing SAS, SATA and Fibre Channel. *Network World*. [Online] September 9, 2004. [Cited: March 25, 2013.] <http://www.networkworld.com/newsletters/2004/0906stor2.html>.
47. **QNAP.** NAS data backup. *QNAP*. [Online] <http://www.qnap.com/en/?sn=999>.
48. **Avere.** Avere FlashMirror. *Avere*. [Online] 2012. [Cited: April 2, 2013.] http://www.averesystems.com/document/FlashMirror_Data_Sheet.pdf.

49. **TR, ITU-T FG Cloud.** *Focus Group on Cloud Computing, Technical Report, Part 3 (02/2012): Requirements and framework architecture of cloud infrastructure.* s.l. : ITU-T FG Cloud TR, 2012.
50. **Rightscale.** Plan for Cloud. *Plan for Cloud.* [Online] [Cited: May 21, 2013.] <https://planforcloud.rightscale.com>.
51. **Associates, Ashton Metzler &.** Ten Things to Look for in an SDN Controller. *NEC.* [Online] [Cited: April 30, 2013.] <http://www.necam.com/>.
52. Softlayer. *Softlayer.* [Online] [Cited: May 7, 2013.] <http://www.softlayer.com/cloudlayer/cdn>.
53. **S.Assefa, et. al.** A 90nm CMOS integrated Nano-Photonics technology for 25Gbps WDM optical communications applications. *Electron Devices Meeting (IEDM), 2012 IEEE International* . 2012, 33.8.1-33.8.3.