2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)

Model for Cloud Migration Cost

Tudor Antohi M.Sc.
Montréal, Canada
tudor_antohi@hotmail.com

***Abstract*—Actual trend today is towards a macro economy based on the OPEX costing model. This trend is the solution proposed for dealing with a large population sharing a limited number of resources on this planet. Cloud Computing is one of the consequences of this tendency. It promises access to a large number of services in an on-demand manner. This raises the problem of finding the best strategy to migrate on premise resources in the cloud. Despite the popularity of the subject, none of the existing papers treats directly the migration period when cost increases and solutions have to be found to limit the expenses and diminish the risk of service interruption. In this paper we propose a methodology of modeling a migration in the cloud that aims achieving lower costs or minimal time, depending on which constraint is needed first. Through a realistic experiment we show the migration of a web system using the proposed methodology. We also see a lot of possible developments and generalizations of the method in similar IT projects of transitioning from CAPEX to OPEX.**

Keywords: Cloud, Cloud Migration Cost, Infrastructure element, Technical components

I. INTRODUCTION

The advancement of new technologies requires more computing resources. Enterprises may migrate their existing infrastructures in the public clouds if their financial analysis forecast an optimal cost. Migration has to take into account other factors and cost is not the only constraint.

Many systems are critical for business and have risk constraints. One solution is to use redundant infrastructure elements for the migration period until all tests are finished and the system on premises can be deprecated. Migration period can become a cost issue and requires a methodology to diminish both risk and cost.

The migration strategy was rarely analyzed from both perspectives: cost and risk. It should have been a more popular subject given the importance. There is a large amount of work related to calculation of the cost of a system running in cloud compared to a system running on premises. However, little work concentrates on the migration period and this is the gap filled by this research.

The methodology presented at the end in Fig. 5 has been developed to optimize the cost of migrating of a system in Cloud. The first step was to model the architecture underlining the features determining cost. Given the fact that we pay the resources per use and bandwidth consumption, we determined the sequence of migration as the main factor of the migration cost. The second step was to understand which sequence of migration of the infrastructure elements composing a system is the most cost effective and offers minimal risk.

The original contribution of this paper is a non-commercial migration methodology which can be easily programmed and used combined with the original mathematical cost model developed to sustain it. The final scope was the creation of a migration algorithm which can be cost optimized. Result analyses are showing the importance of finding the best migration sequence for saving on cost.

The model was proved using a running example based on a multi-tier content management system exposed to web. The system has around 200GB of data documents and manages peaks of five times the average. The system is critical and high availability and minimal migration risks are important constraints. The methodology provided by this paper proves useful for choosing the right strategy and also to show a highly generalizable original method in migration theory applied not only for cloud migrations.

The remainder is structured as follows: Sections 2 reviews related work within the subject; Section 3 describes challenges and constraints; Sections 4, 5, 6, 7 and 8 describe the methodology; Section 9 provides the case study of the migration of a web system and discusses the significance of the results; Section 3 reviews related work within the subject; while Section 10 concludes the paper and describes future work.

II. RELATED WORK

The course INF767 presented by University of Sherbrooke is at the origin of the methodology presented in this paper. The definition of the problem, the hypothesis, the research methodology and the organization of this paper come from ideas presented in this course.

It is clear that the problem can be generalized and the paper limits to a particular case. There was an important part of research done to find if the problem already had a solution and cloud migration cost is a particular case. Transition systems as described in [2] and [2] are the closest idea found to the paper but [13] is not the complete solution either. The idea to model the migration as a transition system and do the model checking for system redundancy and high availability seems however sound and was used.

A large amount of academic papers researches the cost challenging problems of cloud migration and a list is found

978-1-7281-1661-7/19/$31.00 ©2019 IEEE
DOI 10.1109/CSCloud/EdgeCom.2019.00014

in [10],[11]. They all concentrate on comparing the fixed cost before and after migration and never during the migration period. A comprehensive research on all the papers has been done (almost 50 papers are mentioned and none of them was what we were looking for).

The first step is to model an initial architecture which can easily derive the cost elements for a cloud migration and going through different EAM tools it seemed that the closest one for the goal of the paper was Iteraplan as described in [6]. The base elements for an IaaS solution are technical components and infrastructure elements (machines, databases, application instances, network elements etc.). Technical components are software technologies like RedHat, Hadoop, Oracle, Apache or Jboss but also proprietary software running on the mentioned infrastructure elements and using other vendor software.

Caesar Wu et al [3] do both a practical and theoretical approach for calculating the cost of a datacenter in the cloud. Chapter 16 is the most relevant for the paper. The chargeback technology is explained with examples based on distributed resource management. This reference presents the raw parameters involved in cloud cost calculation with prices which are not relevant as of 2019 so they are not mentioned. The cloud cost parameters for infrastructure are shown in Table I.

TABLE I. EXAMPLE OF COST FACTORS IN OPEX

Resource	Metrics
vCPU	GHz
vCPU type	Categorical variable
Memory	GB
Disk Reads	Number of Disk reads per Second
Disk Writes	Number of Disk Writes per Second
IOPS	Number of IOs per Second
Network Writes	Number of Network Writes per Second
Network Reads	Number of Network Reads per Second
Storage	GB
Storage Type	Categorical variable

As a conclusion, the cost in cloud is based on an OPEX model and influenced by three factors. The first is the duration of activity of the IT block in the cloud. The second is the data stored or interchanged via network and IO systems. The last is the cost of licensed components.

According to Gartner [10], the most important cloud vendors in 2018 are still Amazon and Azure. Both offer, pricing sites like [8] and [9] able to simulate the cost of the system architecture in the cloud but they do not calculate cost for different migration scenarios. Combining the idea of transition systems and these documents was the essence of the method.

Analysis established that [3] and [6],[8],[9] can be used for mapping the technical components and the infrastructure elements in cloud blocks.

Reference [14] is relevant because it focuses exactly on the migration period using a technique similar of function points. The paper presents four types of cloud migration points related to changes in connectivity, code, infrastructure and databases and the total cost is estimated given the complexity degree of each. This is the first and only tentative found during research of modelling the cost of migration period.

TABLE II. CLOUD MIGRATION POINTS

CMP	Type	Complexity Level			Weight
		Low	*Average*	*High*	
CMPconn	LAN to LAN	1	3	4	3
	LAN to WAN	1	6	9	
	WAN to LAN	1	6	9	
CMPcode	PDT	3	6	10	5
	HIT	4	7	12	
	DMT	5	8	13	
	TMT	4	6	9	
CMPict	Aplication installation and configuration and test	1	2	7	2
	Infrastructure installation and configuration and test	1	3	9	
CMPdb	Query Modification	1	3	8	1
	Data Population	3	4	10	

Proposed formula to calculate the migration complexity is presented in (1).

$$\mathrm{CMP} = \sum_{i=1}^{4} \mathrm{CMPi} \times \mathrm{Wi} = \mathrm{CMPconn} \times \mathrm{Wconn} + \mathrm{CMPcode} \times \mathrm{Wcode} + \mathrm{CMPic} \times \mathrm{Wic} + \mathrm{CMPdb} \times \mathrm{Wdb} \quad (1)$$

CMPi stands for a cloud migration point and Wi stands for weight, a measure given in this article by the complexity level of each cloud migration type as presented in Table 2.

The research in [14] approaches the type of migration as a big bang. Everything is supposed to migrate in one step. A transition towards cloud, in order to keep high availability constraints, changes the architecture in intermediate states so the formula should also reflect the changes. The aim of this paper is to delve into these details. Also the case study does not cover code and database since refactoring code is not calculated. CMPconn and CMPic are however very important for determining the cost formula and not on

complexity. Every transition will change the cost of connectivity because it involves a change in network topology. Installation and test as demonstrated later in the paper have different impacts on the total cost.

Research has been done on Gartner Group site but there is no document pointing to the cost of transitioning a system into the cloud. Companies like Cloudamize are selling evaluation software and agents but they do not go through transition periods.

The final goal is to find the optimal transition path to the best cost and evaluate how much the migration sequence determines it. Books like [5] and [7] were the best to describe the search algorithms and the best paths to apply. The paper used finally a simple algorithm because as per the statistics book, it needed to prove there is a correlation between the migration sequence and the cost and delve into the details of the impact of the order of migration per infrastructure element.

III. Challenges, Constraints and Hypothesis

The main challenges of this research are:

- Not enough literature to analyze the cost during the migration period of systems.
- Two models had to be identified, a static architectural model of elements influencing the cost and a dynamical architectural model which defines the intermediate states. Every element migrated is installed and passes a testing period. The transition is through a sequence of intermediate states and each one has an impact on the migration cost that needs to be calculated.
- Mapping the cost model of the cloud on a system on premises requires precise measurements.
- Variables which influence the cost had to be detected and the values optimized.
- The system has to be high-available with zero-downtime during migration.

As mentioned, there are also established constraints in this paper in order to reduce the complexity of our research.

The high-availability challenge involves the creation of redundant cloud architectural blocks during short period of tests. Once the tests pass, the infrastructure elements of the datacenter on premises are decommissioned. This type of migration is popular because it reduces risk, but it also increases for a short period of time the cost of the system.

It is possible to migrate one to many infrastructure elements in the same time in the cloud and to decommission one to many elements in the same time once the tests are passed. This paper analyzes the case of a migration of one element at a time which is not always realistic, but the model can be improved to accept more elements migrated in the same time. See Conclusions chapter for more details. There is also an assumption of no delay in the transition of elements in the cloud, if element IE(i) finishes testing, element IE(i+1) follows without delay the installation period.

The cloud analysis is done for an IaaS system but improvable for PaaS and SaaS systems. Also, the case study limits analyses to Amazon AWS and Microsoft Azure found as main vendors of cloud IaaS in 2018. The research confirms that all cloud vendors are using similar cloud costing models.

The cost of the elements on premises and the human resource expenses are not considered. The cost of licensed components is considered the same in cloud and on-premises. The bandwidth is considered equally distributed between similar functional infrastructure elements in cloud and on-premises.

The case studies the migration without making any changes in application and database code. The constants are the network and IO bandwidth, the storage and the infrastructure element type paid by usage. The infrastructure must be installed and tested in the cloud before being declared fit for production. This is another difference compared to reference [8].

The analyses consider the sequence of the migration of infrastructure elements as the independent variable and the dependent variable is the total cost of the migration period.

The hypothesis is that the sequence of migration strongly influences the migration cost.

If the results show that the migration sequence influences strongly the cost than the methodology and the model are worth the research.

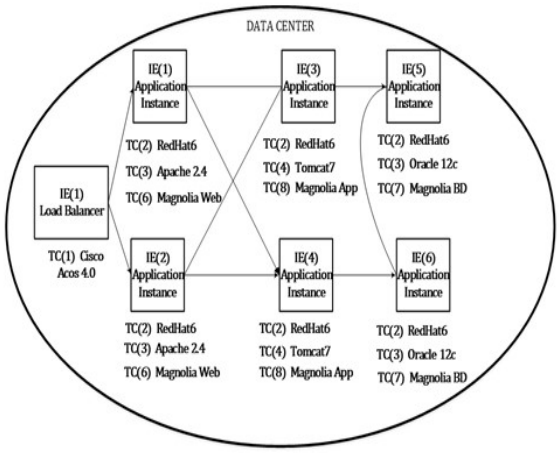

Fig 1 Multitier architecture used as running example. It has a 3 tier architecture with a load balancer in front of high availabity web

IV. Methodology

The proposed methodology has three main steps:

The first step is to model and map the system in an infrastructural cloud cost model. The infrastructure elements are decomposed in cost related attributes. During this step, values are assigned for installation and testing periods for the technical components. Also, there is a mapping part of every infrastructure element on premises in the best fit of a similar cloud architectural block. This step finds the target final

price. The second step is to build the cost model of the usage and lifecycle). Table IV defines the infrastructure

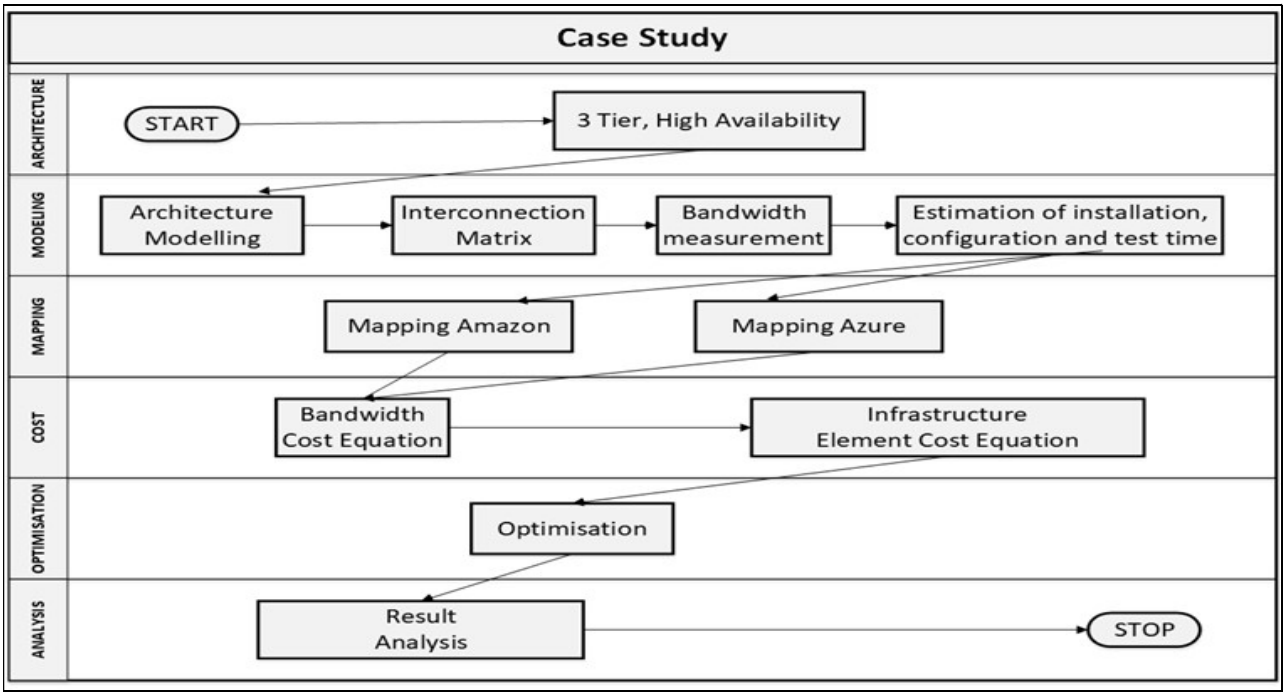

Fig 2 Methodology

transition and establish a set of formula which calculate the total migration cost given the constraints. Final step is to optimize the cost finding the best migration sequence.

The methodology is applied on the running example presented in Figure 1 and involves two extra steps. Finding the running example at the beginning and analysis at the end. The running example is composed of servers represented as infrastructure elements running technical components also known as software. It is composed of a multitier tier web architecture with a load balancer as Internet interface, a web proxy, an application server and a relational database tier. As mentioned for the transitions, all elements are shortly doubled for redundancy. They need time for installation and testing before they are fit for going live and enable the next transition step.

V. INFRASTRUCTURE COST MODEL

The model must decompose the IT architecture into blocks influencing the cost in the cloud. Each informational system is composed of infrastructure elements on which technical components are running. The infrastructural elements can be servers or application instances, databases, networks or storage related types. All infrastructure elements are interconnected via networks. Servers have internal configurations in terms of number of processor cores, storage used and network bandwidth. They are hosting the operating systems, applications and databases. In cloud, the infrastructure elements are paid per usage (outbound network elements presented in Figure 1.

The technical components are presented in Table III as they run on infrastructure elements. They influence the cost because during the time taken for installation and test, the infrastructure elements and data transfers are paid.

TABLE III. TECHNICAL COMPONENETS

#	Technical Components	Installation TC_I(i) (days)	Test TC_T(i) (days)	Total TC_Total(i) (days)
1	A10 Acos4.0	1	1	2
2	Redhat 6	0.75	0.25	1
3	Apache 2.4	0,2	0,2	0,4
4	Tomcat 7	0,4	0,4	0,8
5	Oracle 12c	0,5	0,5	1
6	Magnolia Web	0,5	1	1,5
7	Magnolia Database	0	0,2	0,2
8	Magnolia Application	0,5	5	5,5

An information system S is composed of N infrastructure elements IEi, with i varying from 1 to N. It runs M technical components TCk, with k varying from 1 to M. Every system works with a subset of infrastructure elements and technical components belonging to data center. Technical components are assigned to infrastructure elements. Table 5 presents the

matrix of connections between infrastructure elements and technical components. In Python an informational system S defined in our running example can be represented as a dictionary of tuples as an (2).

S = { IE(1) :(TC(1)), IE(2) :(TC(1)), IE(3) :(TC(2), TC(3),TC(6)),IE(4) :(TC(2),TC(4),TC(8)), IE(5) :(TC(2),TC(4),TC(8)), IE(6) :(TC(2),TC(3),TC(7)), IE(7) :(TC(2), TC(3), TC(7))}

(2)

TABLE IV. INFRASTRUCTURE ELEMENTS

Resource	Infrastructure Elements	Number of vCPUs (other characteristics)	Cost per day in the cloud for the mapped infrastructure element
1	Load Balancer	NA	CMP1(1)
2	Web1	2	CMP1(2)
3	Web2	2	CMP1(3)
4	Application Server 1	4	CMP1(4)
5	Application Server 2	4	CMP1(5)
6	Database Server 1	8	CMP1(6)
7	Database Server 2	8	CMP1(7)

The infrastructure elements are interconnected via networks and storage elements. Every technical component can be represented as in Table V as matrix IETC. A value of zero IETC(i,j) means TC(j) is not running on IE(i), a value of one means TC(j) runs on IE(i). IE(i) sends Xij gigabytes of network data to IE(j). Matrix IEX of Table VI is used to describe the bandwidth interconnections between infrastructure elements:

TABLE V. IETC = MATTRIX OF TECHNICAL COMPONENTS MAPPED TO INFRASTRUCTURE ELEMENTS

Infrastructure elements	Technical Components			
	TC(1)	*TC(2)*	*TC(i)*	*TC(M)*
IE(1)	0|1	0|1	0|1	0|1
IE(2)	0|1	0|1	0|1	0|1
IE(j)	0|1	0|1	0|1	0|1
IE1(N)	0|1	0|1	0|1	0|1

TABLE VI. IEX = NETWORK TOPOLOGY MATRIX

Infrastructure elements	Infrastructure elements (gigabytes exchanged per day between IE(i) to IE(j)			
	IE(1)	*IE(2)*	*IE(i)*	*IE(N)*
IE(1)	0	X12	X1j	X1n
IE(2)	X21	0	X2i	X2n
IE(j)	Xj1	Xj2	Xji	Xjn
IE1(N)	Xn1	Xn2	Xni	0

At high level, the formula which calculates the cost for a system once migrated in cloud is (3):

$$\text{Cost} = \text{CostIE} + \text{CostBW} = \sum_{i=1}^{2} \text{CMPi x Wi} \tag{3}$$

CMP1 represented as the first cloud migration point and is the number of hours spent by infrastructure element IE(i) in the cloud and W1 is the price usage per hour. This includes the licensed technical components and the storage used.

CMP2 represented as the second cloud migration point is the number of outbound gigabytes exchanged by infrastructure elements IE(i) during a specific number of hours billed and W2 is the price per gigabyte per hour.

Every infrastructure element IE(i) is running technical components needing time to be installed and tested. This time is not taken into consideration once the migration finished.

During migration in the cloud, the total number of hours CMP1it(i) is the sum of the installation and testing time for infrastructure element I from the creation until it goes live after test, like in (4) and (5) according to Table IV and Table V.

$$\text{CMP1it(i)} = \text{CMP1i(i)} + \text{CMP1t(i)} \tag{4}$$

$$\text{CMP1it(i)} = \left(\sum_{j=1}^{M} \text{IETC(i,j)} * \text{TC\_Total(j)} \right) \tag{5}$$

During migration CMP2(i) is the sum of the gigabytes exchanged during testing time for infrastructure element I from the creation until it goes live. It is assumed that installation time does not generate paid bandwidth since packets do not cross the cloud to datacenters, like in (6).

$$\text{CMP2(i)} = \left(\sum_{j=1}^{M} \text{IEX(i,j)} * \text{TC\_T(j)} \right) \tag{6}$$

VI. MIGRATION COST MODEL

The mapping of infrastructure elements namely host or servers in case of IaaS, in cloud related blocks is done using the variables from Table I.

The basic operation of mapping would consist in measuring the variables of Table I and finding the blocks of the cloud vendor which should minimally increase or at least be equal in power with the original measures.

The total cost during migration time is the sum of the cost of the infrastructure elements paid by usage in the cloud and the cost of the bandwidth is in (3).

The architecture changes with every element migrated going through intermediate statuses as represented in Figure 3. Every transition step incurs a different cost. The cost of

the bandwidth varies like in Table VII. Assuming the first I elements migrated in the cloud the bandwidth paid is for the outbound bandwidth between cloud and data center between the first I migrated infrastructure elements and the remaining N-I. In the matrix bellow the dollar sign $ was added to identify the paid outbound bandwidth and the letters **mig** to identify the infrastructure elements migrated.

Table VIII explains how the migration steps are adding cost. Every step takes into account an already existing number of elements migrated. During the installation of infrastructure element IE(i) there is no bandwidth cost change. During the testing of IE(i) the bandwidth cost changes according to network topology change.

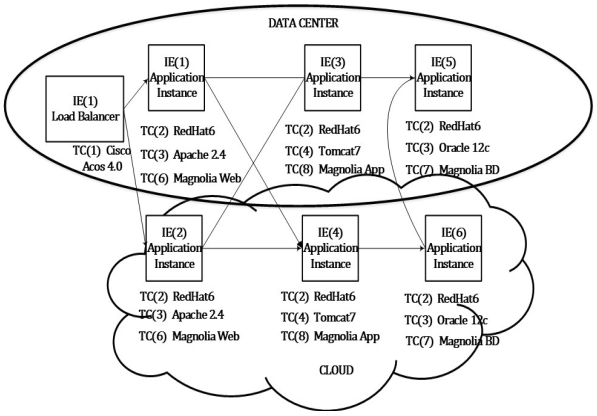

Fig 3 Multitier architecture used as running example during migration. Several infrastructure elements migrated in the cloud changing the network topology and the price.

TABLE VII. NETWORK TOPOLOGY MATRIX AFTER MIGRATING IEMIG(1)

Infrastructure elements	**Infrastructure elements (gigabytes exchanged per day between IE(i) to IE(j) dollar sign is used to identify paid traffic)**				
	IEmig(1)	***IEmig(2)***	***IEmig(i)***	***IE(i+1)***	***IE(n)***
IEmig(1)	0	X12	X1i	X1n	X$1n
IEmig(2)	X21	0	X2i	X2n	X$2n
IEmig(i)	Xi1	Xi2	0	Xjn	X$in
IE(N)	Xn1	Xn2	Xni	X$n,i+1	0

TABLE VIII. MIGRATION STEPS

Step	**Description**
IE(1)	First element IE(1), a host, is mapped in the cloud and started. The architecture changes in an intermediate state.
	The technical components running on IE(1) are installed and tested. During this period, the price of the cloud block equivalent to IE(1) is paid per usage. Only starting the test period, the network outbound price is paid for the network activity of IE(1).
IE(2)	IE(2) migrates. The architecture changes in a new intermediate state.
	The technical components of IE(2) are installed and tested. During this period the price for using the mapped blocks for IE(1) and IE(2) is paid. The network bandwidth price for IE(1) and IE(2) is also paid but slightly different. The price for network bandwidth of IE(1) is paid for the period of installation and test of IE(2). The price of network bandwidth of IE(2) is paid only for the period of testing IE(2)
-------	----------
IE(j)	IE(j) migrates. The architecture changes in a new intermediate state.
	The technical components of IE(j) are installed and tested. During this period the price for using the mapped blocks for IE(1)…IE(J) is paid. The network bandwidth price for IE(1), IE(2) … IE(j) is paid as such: • The price for network bandwidth of IE(1), IE(2) … IE(j-1) for the period of installation and test of IE(j). • The price of network bandwidth of IE(j) is paid only for the period of testing IE(j).
-------	----------
IE(N)	Last IE(N) migrates. The architecture changes in the final state.
	The technical components of IE(N) are installed and tested. During this period, the price for using the mapped blocks for IE(1)…IE(N) is paid. The network bandwidth price for IE(1), IE(2) … IE(N-1) is paid as such: • The price for network bandwidth of IE(1), IE(2) … IE(N-1) for the period of installation and test of IE(N). • The price of network bandwidth of IE(N) is paid only for the period of testing **IE(N)**
	End of migration. We can decommission the existing system on premises

The following two theorems are calculating the members of (3). They are proved by mathematical induction.

Theorem 1 (Cost IE). *Given CMPit1(j) the total number of hours for installation and test of infrastructure element IE(j) and W1(j) the price paid by hour of usage of same IE(j) than the total cost of migrating the infrastructure elements from element 1 to element I is:*

$$\text{CostIE(i)} = \sum_{j=1}^{i}(\text{CMP1it(j)} * (\sum_{k=j}^{i} \text{W1(k)})) \tag{7}$$

Proof. The proof is by mathematical induction and it takes two steps. In the first step, it is demonstrated that formula is true for i=1. The second step assumes that if formula is true for i=N then it demonstrates it is also true for i=N+1.

For I=1 there is only IE(1) migrated in the cloud, therefore cost of infrastructure elements paid in cloud is in (9), the time spend for installation and test of IE(1) times the cost per hour :

$$\text{CostIE(1)} = \text{CMP1it(1)} * \text{W1(1)} \tag{8}$$

Assuming (10) is true for i = n it will be proved it is true for i=n+1:

$$\text{CostIE(n)} = \sum_{j=1}^{N}(\text{CMP1it(j)} * (\sum_{k=j}^{N} \text{W1(k)}))$$

(9)

During the installation and test of infrastructure element IE(n+1) the elements 1..n continue working paid hours during the time to install and test IE(n+1) represented by CMP1it(n+1). Therefore, it confirms the formula in (11):

$$\text{CostIE}(n+1) = \text{CMP1it}(n+1) * \text{W1}(n+1) + \text{CMP1it}(n+1) * \left(\sum_{j=1}^{N} \text{W1}(k)\right) + \sum_{j=1}^{N} \left(CMP1it(j) * \left(\sum_{k=j}^{N} W1(k)\right)\right) = \sum_{i=1}^{N+1} \left(\text{CMP1it}(j) * \left(\sum_{k=i}^{N+1} \text{W1}(k)\right)\right) \quad (10)$$

Theorem 2 (Cost BW). *Given CMP2(j) the total number of network gigabytes used for installation and test of infrastructure element IE(j, CMP1i(j) the installation time of IE(j), CMP1t(j) the test time of IE(j) and W2 the price paid by hour of network usage of same IE(j) than the total bandwidth cost of migrating infrastructure elements from 1 to I is:*

$$\text{CostBW}(i) = \left(\sum_{j=1}^{i} (\text{CMP2}(j) * \text{CMP1}_t(j)) + \sum_{j=2}^{i} (\text{CMP2}(j-1) * (\text{CMP1it}(j))\right) * \text{W2} \quad (11)$$

Where CMP2(j) is:

$$\text{CMP2}(j) = \left(\sum_{k=1}^{j} \sum_{l=j+1}^{N} \text{Xkl}\right) \quad (12)$$

Proof.

For I =1 there is only IE(1) migrated in the cloud. The cost of bandwidth after migrating IE(1) is relevant only for the test period when ports are opened and tests are performed during CMP1t(1) hours as in (13).

$$\text{CostBW}(1) = (\text{CMP2}(1) * \text{CMP1t}(1)) * \text{W2} \quad (13)$$

Where the number of network gigabytes CMP2(1) exchanged in cloud is between migrated IE(1) and the rest of IE(2) to IE(n) left on-premises is in (14)

$$\text{CMP2}(1) = \text{X12} + \text{X13} + \cdots + \text{X1N} = \sum_{l=2}^{N} \text{X1l} \quad (14)$$

For I=2 there are IE(1) and IE(2) migrated in cloud. The cost of bandwidth after migrating IE(2) is relevant only for the test period when ports are opened and tests are performed during CMP1t(2) hours. During installation and test of CMPit(2) hours, element IE(1) has normal bandwidth consumption. The cost of bandwidth at step 2 is in (15):

$$\text{CostBW}(2) = (\text{CMP2}(1) * \text{CMP1t}(1)) * \text{W2} (\text{CMP2}(2) * \text{CMP1t}(2)) * \text{W2} + (\text{CMP2}(1) * \text{CMP1it}(2)) * \text{W2} = \sum_{j=1}^{2} (\text{CMP2}(j) * \text{CMP1}_t(j)) \quad (15)$$

Next step is to consider the formula true for infrastructure element i as in (16), before migrating element IE(i+1). The network bandwidth paid is assumed to be:

$$\text{CostBW}(i) = \left(\sum_{j=1}^{I} (\text{CMP2}(j) * \text{CMP1}_t(j)) + \sum_{j=2}^{I} (\text{CMP2}(j-1) * (\text{CMP1it}(j))\right) * \text{W2} \quad (16)$$

When IE(i+1) migrates, the cost of installation and testing of IE(i+1) has to be added to CostBW(i) before it goes live. This cost is the sum between the bandwidth cost of the I migrated elements during installation and testing of IE(i+1) plus the cost of the bandwidth used by IE(i+1) while testing (installation does not count):

$$\text{CostBW}(i+1) = = \text{CostBW}(i) + \text{W2} * \text{CMP1t}(i+1) * \text{CMP2}(i+1) + \text{W2} * \text{CMP2}(i) * \text{CMP1it}(i+1) = \left(\sum_{j=1}^{i+1} (\text{CMP2}(j) * \text{CMP1}_t(j)) + \sum_{j=2}^{i+1} (\text{CMP2}(j-1) * (\text{CMP1it}(j))\right) * \text{W2} \quad (17)$$

Formula (17) demonstrates theorem 2.

VII. Optimization

The optimization algorithm used finds the best migration sequence (IE(i1), IE(i2), IE(i3),…,IE(in)) which minimizes the migration cost. The solution applied in the case study is a bubble sorting algorithm which goes through these steps:

- Initialization of the matrix for bandwidth consumption between infrastructure elements
- Initialization of a matrix containing the technical components with values for days of installation and test for each
- Initialization of a matrix having the list of infrastructure elements and the prices paid per hourly usage in cloud (considering the OPEX cost)
- All N! permutations of the migration sequences of the infrastructure elements are generated in a matrix containing all possible migration sequences.
- Costs are calculated for each migration sequence and are maintained in a list of cost per migration sequence.
- The list of costs per migration sequence is sorted and the sequence which generates the minimal cost is saved.

The graph-search algorithm generates a tree-search. O(m2) is the complexity of the algorithm given the bubble

sort, where m is the permutation of N therefore N!. According to [5] there are four ways to define an algorithm. Completeness, and in this case the algorithm guarantees success. Optimality which is also guaranteed given the fact that all possible combinations are tested. Space complexity has an impact on memory used and is not a problem since the optimization persists only the last result

Time complexity is the sum of the number of vertices and edges. The graph generated by the problem has N! vertices and N^2 edges. The branching factor (maximum number of successors of each node) which is in our case N-1, and the depth and the maximum length which is in all cases N. The algorithm uses a blind uninformed search strategy since but there is further research to be done to progress into an informed search strategy with better chances to reduce the time complexity.

VIII. CASE STUDY

The case study uses the methodology described in Figure 2 for the running example presented. The first 3 sub-steps belong to the system model and the next 3 sub-steps belong to migration model.

The modelling for cost has been described in Sections VI and VII. Tables IV and VI have been filled with cost related information. Measurements have been done using tools like Grafana and Tcpdump. Estimation for the time spent to setup technical components has been done with cloud architects.

Each infrastructure element was mapped in Amazon and Azure to find the cost per hour.

TABLE IX. BEST AND WORST MIGRATION SEQUENCE

	Infrastructure Elements (producing the min and moax costs)	Cost Amazon (min and max costs)	Cost Azure (min and max costs)
1	(4,2,1,3,5,6,7)	730,68	
2	(6,7,2,3,1,4,5)	1353,69	
3	(1,4,5,2,3,6,7)		3612,78
4	(7,6,2,3,4,5,1)		27151,29

The critical path is calculated using an algorithm presented in Section VII. Table IX is showing the results for best and worst migration sequence. Migration costs can be two to eight times greater than minimum costs, so blind migration can have negative cost impacts. This confirms the hypothesis that the migration sequence is determinant for the cost of migration. The following paragraphs detail statistical analyzes for Azure presented in Table X using Excel with the Microsoft Analysis Module as well as the Real Statistics Analysis Pack add-on for inferential statistics.

TABLE X. DESCRIPTIVE STATISTICS

	Statistics	Value
1	Moyenne	1050,47
2	Error-type	2,20
3	Median	1061,39
4	Mode	972,81
5	Écart-type	156,39
6	Variance de l'échantillon	24457,37
7	Kurtosis	-0,90
8	Dissimetry	-0,14
9	Plage	623,01
10	Minimum	730,68
11	Maximum	1353,69
12	Sum	5294346,72
13	Number of samples	5040

The mean and median are used for central trend analysis. The two values are close, the average is 1050.47 and the median is 1061.39. They confirm that outliers are almost non-existent. The mode is the most frequent value (972.81), sixteen variables are associated with this category. It is more likely to be around this cost zone if the correct method is not used. Descriptive statistics also contain measures of variability that can analyze the distribution of costs. The maximum value minus the minimum value is (423.01). The standard deviation is a measure of degree of distance from the average. Its value of 156.39 confirms that the values are close to the average and the standard error around 1.5%.

Large values are mostly to the right of the mean (kurtosis has a negative value), but the dissymmetry is small.

The chances of falling on best cost values are small if the right methods are not applied. The probability is 8 (number of appearances of smaller cost) out of 5040 (number of samples or migration sequences) about 0.15% as shown in Figure 5.

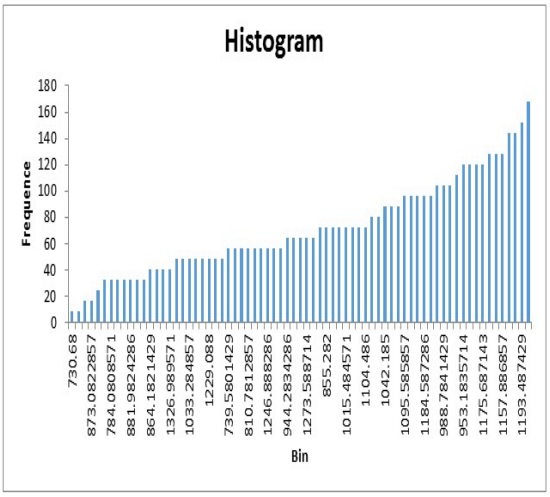

Fig 5 Histogram (number of migration sequences with same cost)

It is possible to analyze the impact of the migration position of each infrastructure element on the cost of migration, as in Table XI.

TABLE XI. ELEMENT POSITION AND COST IMPACT

Sequence (list of elements)	E1	E2	E3	E4	E5	E6	E7	COST
(4,2,1,3,5,6,7)	2	1	3	0	4	5	6	730,68
(4,2,3,1,5,6,7)	3	1	2	0	4	5	6	731,90

TABLE XII. CORRELATION BY POSITION

	Pos E1	Pos E2	E3	E4	E5	E6	E7
Correlation	0.03	0.14	0.14	0.41	0.41	(0.58)	(0.58)

The elements with the greatest impact on cost are IE6 and IE7, heavier in network traffic and more expensive priced. The more IE6 and IE7 migrate at the end, the cost impact is better on migration cost. The general correlation of all elements in a sequence gives a value of 0.99 indicating a probable huge impact of migration sequence on cost.

IX. CONCLUSION

Migration modeling can be done with automated means. Cost simulation is possible, but the solutions on the market mainly consider only the cost once the system is migrated in the cloud. The paper addresses the migration cost. The problem resolved by this research is a methodology generalizable to similar migration projects not only datacenters or IT. This is one other claim of this paper.

The case study is limited to an IaaS running example, but it is possible to apply it to PaaS cloud migrations. Mapping elements in PaaS replaces also some technical components, not just infrastructure elements.

Mathematical improvements of optimization have been mentioned. The number of infrastructure elements increases the complexity of these analyses and the sorting algorithm makes a difference.

This paper introduces a constraint for one infrastructure element at a time. It is possible to change the constraints and develop the formula for cases of multiple migrations of infrastructure elements in the same time. The total migration period will be shorter if it is possible to migrate several infrastructure elements in parallel.

Every previously mentioned constraint can become a direction of research of this interesting domain which can be extended beyond a simple migration in the cloud towards a general migration theory between systems. Or maybe an evolutionary theory where intermediate architectural states are compared to intermediate layers in an analogy with neural nets.

All these are subject of future research.

ACKNOWLEDGMENT

The idea of this research started during a discussion with my teacher from University of Sherbrooke, Lynn Legault. Her way of structuring my ideas helped me formulate this interesting problem and course ING 767 was one of the best materials to help me through.

Part of the architectural mode presented started with a work done between 2014 and 2016 which finished what I consider a very good architectural model as of 2019. I say thanks to my former colleagues Benoit Desjardins, Lucie Lévéille, Claude Lamy, Isaak Mekueko and Avigael Levy and many others who worked a bit or more to produce the model which was a bit revamped to be used inside this paper.

REFERENCES

[1] University of Sherbrooke, INF 767 Notes de Cours, 2016-2017.

[2] Roberto Gorrieri, Cristian Versari, Introduction to Concurrency Theory: Transition Systems and CCS, Springer, 2016.

[3] Caesar Wu, Cloud Data Centers and Cost Modeling: A complete Guide to Planning and Building a Cloud Data Center, 3rd ed., Morgan Kaufman, 2014, chp.16-18.

[4] T A Henzinger, A Pnueli, Zohar Manna, Timed Transition Systems, Ithaca, N.Y. : Cornell University, Dept. of Computer Science, [1992], 1992.

[5] Peter Norvig, Artificial Intelligence a Modern Approach, 3rd ed, 2010, chp.1-3.

[6] Inge Hanschke, Strategic IT Management: A Toolkit for Enterprise Architecture Management, Springer, München 2010, ISBN 3-642-05033-6

[7] Allen Downey, Think Stats : Probability and Statistics for programmers, Green Tea Press 2011.

[8] Amazon, "Amazon pricing calculator," 2019.

[9] Azure, "Azure pricing calculator," 2019.

[10] Meinardi, M., "Gartner Group, «Magic Quadrant for Cloud Infrastructure as a Service, Worldwide", Gartner Group 2018.

Article in a journal:

[11] Pooyan Jamshidi, "Cloud Migration Research: A Systematic Review", IEEE Transactions on Cloud Computing vol. 1, Dec. 2013.

[12] Mahdi Fahmideh, Ghassan Beydoun, Graham Low, Farhad Daneshgar, "Cloud Migration Process, A Survey, Evaluation Framework, and Open Challenges", Journal of Systems and Software , June 2016.

Article in a conference proceedings:

[13] Van T. K. Tran, K. L, "Minimization of Timed Transition Systems", International Conference on Concurrency Theory, CONCUR 1992, pp.340-354,1992.

[14] Van T. K. Tran, K. L, "Size Estimation of Cloud Migration Projects with Cloud Migration Points (CMP)", Conference: Proceedings of the 5th International Symposium on Empirical Software Engineering and Measurement, ESEM 2011, Banff, AB, Canada, September 22-23, 2011.