New Approach to Partitioning Confidential Resources in Hybrid Clouds

Kaouther Samet[1], Samir Moalla[1] and Mahdi Khemakhem[2]
[1]*Department of Computer Science, Faculty of Sciences, University Tunis El Manar, Tunis, Tunisia*
[2]*Department of Telecommunications, National School of Electronics and Telecommunications, University of Sfax, Sfax, Tunisia*

Keywords: Partitioning Resources, Hybrid Clouds, Confidentiality.

Abstract: Today, companies use more cloud environments such as hybrid clouds. Indeed, hybrid clouds give the opportunity to better manage resources mostly when companies have no space to store more resources in their private clouds. The best solution here is to allocate the required space in public cloud at a low cost. But how can resources be partitioned in hybrid clouds while assuring confidentiality of resources moved to public cloud. Many works have been done in this context. They suppose that confidentiality is assured by using encryption methods. But with this solution the cloud provider can access the resources stored on the cloud, which weakens the confidentiality of these. This work proposes an approach to the Confidential Resources Partitioning Problem in Hybrid Clouds (CRPHC) which aims at ensuring the confidentiality of resources by grouping as much as possible the most confidential resources in private cloud and resources with low degrees of confidentiality in public cloud while minimizing the size of resources to host in public cloud and consequently reducing the storage cost. This solution allows the possibility of using non-performing encryption methods which have a reduced treatement cost compared to efficient methods. Experimentally, our solution will be evaluated and compared to optimal solution given by CPLEX.

1 INTRODUCTION

Cloud computing has become a major concept referring to the use of memory, computing capabilities computers and servers around the world, all of them linked by a network such as the Internet. Today, companies use more cloud environments for deployment and execution of their applications. Usually, the most used type of clouds in the cloud environment is the hybrid cloud. The infrastructure of this type of cloud is composed of two or several public and private clouds. It is obvious for a company that applications must be deployed in the private cloud as resources can be provided by their cloud.

However, when the physical limit of the private cloud is reached, the company may need to use other resources (data, services or applications) from a public cloud. This occurs when applications and platforms of companies need to be enlarged and request additional resources that the private cloud is not able to provide. In this case, obtaining new resources from public cloud can solve this problem. Consequently, resources will be partitioned between private and public clouds. Among the obstacles, mentioned by authors of (Stoica and Zaharia, 2009), in cloud environment is confidentiality and the study of secure data in this environment is fairly new and has become increasingly important (Nepal and Calvo, 2014). Indeed, they consider that it is the most important obstacle in this environment. So, how can confidentiality of resources be ensured in the hybrid cloud? To ensure confidentiality in the clouds, encryption methods have been used. But to have better results, it is necessary to use performing encryption methods which are very expensive in terms of execution time and complexity (Chokhani, 2013). However these works fail to raise the problem that the public cloud providers theoretically have access to the received resources.

In this context, we propose an approach to solve the Confidential Resources Partitioning Problem in Hybrid Cloud (CRPHC) which aims at ensuring the confidentiality of resources by grouping as much as possible the most confidential resources in private cloud and resources with low degrees of confidentiality in public cloud while minimizing the size of resources to host in public cloud and cosequently reducing the cost of storage.This solution allows to use a non-performing encryption methods which have a reduced treatment cost compared to efficient methods. Experimentally, our solution will be evaluated

Samet K., Moalla S. and Khemakhem M..
New Approach to Partitioning Confidential Resources in Hybrid Clouds.
DOI: 10.5220/0005444505060513
In *Proceedings of the 5th International Conference on Cloud Computing and Services Science* (CLOSER-2015), pages 506-513
ISBN: 978-989-758-104-5
Copyright © 2015 SCITEPRESS (Science and Technology Publications, Lda.)

and compared to optimal solution given by the commercial software IBM-ILOG-CPLEX 12.5 applied to an integer linear programming formulation of the CRPHC.

The rest of the paper is organized as follows: in section 2, we present an overview of works studying the partitioning problem. In section 3, we list in details the integer linear programming formulation of the CRPHC, while in section 4 we clarify our approach to partitioning confidential resources in hybrid clouds. In section 5, we evaluate and compare our solution to optimal solution given by CPLEX. Finally, we end up giving our conclusion and future works in section 6.

2 STATE OF ARTS

The problem of partitioning resources in cloud environments has been seen from different viewpoints, while considering different types of criteria such as confidentiality, access frequency of query execution, communication, etc.

In the following, we present some studies for confidentiality management in clouds.

2.1 Confidentiality Management Assured by the Public Cloud Provider

We present below approaches Schism and Birch (Zhang; and Madden, 2010) and (Ramakrishnan and Livny, 1996) treating the problem of partitioning data in databases. They try to produce the best quality clustering with the available memory and time constraints.

Authors of (Tata and Moalla, 2012) propose a new algorithm that approximates the optimal placement of services based on communication and hosting costs induced by the shifting of components towards the public cloud. This research is interested in deciding which services will be deployed to the public clouds based on communications between services within the public cloud, and communications between services of the private cloud and services of the public cloud.

In (Wang and C.Jiang, 2012) and (Wang and Guo, 2013), authors propose a model for the multi-objective data placement and use a particle swarm optimization algorithm to optimize the time and cost in cloud computing.

In works already mentioned, the authors are not interested in the confidentiality of resources moved to the public clouds. In fact, they suppose that confidentiality is guaranteed by the public cloud provider using encryption methods applied on all the resources moved to the public cloud regardless of their degree of confidentiality (Mehrotra and Thuraisingham, 2012) and (Kantarcioglu and Thuraisingham, 2011). But, in this case, the cloud provider can consult and access to confidential resources in public cloud.

2.2 Confidentiality Management in Hybrid Clouds

In (Pilli and Joshi, 2013), authors present a solution approach to the data partitioning problem. They create different partitions and estimate the execution cost of the query workload for each of these partitions and check whether any monetary and confidentiality risk constraints were violated. Authors assume that all predicates have the same level of confidentiality.

Authors of (Mehrotra and Thuraisingham, 2012), (Kantarcioglu and Thuraisingham, 2011), (Marwaha and Bedi, 2013) and (Lamba and Kumar, 2014) propose approaches to ensure confidentiality in clouds based on encryption. Indeed, they suppose that resources confidentiality is assured by encryption methods. But in (Nepal and Calvo, 2014) authors consider that this solution is computationally inefficient and locates a large workload on the data owner when considering factors such as updating encryption keys. Likewise, according to (Chokhani, 2013) encryption methods have additional complexity in cloud environments which makes this operation very expensive and complex.

To solve this problem, we propose an approach to the Confidential Resources Partitioning Problem in Hybrid Clouds (CRPHC) which aims at ensuring the confidentiality of resources by keeping the most confidential resources in private cloud and moving resources with lower degrees of confidentiality into public cloud; while minimizing the size of resources to host in the public cloud.

3 INTEGER LINEAR PROGRAMMING FORMULATION FOR THE CRPHC

In this section we present an integer linear programming formulation for the Confidential Resources Partitioning Problem in Hybrid Cloud (CRPHC). Our aim is to:

- Ensure confidentiality by storing resources with low confidentiality in public cloud,
- Minimize resources storage cost in the public cloud while respecting a minimal size of resources to host in public cloud.

3.1 Problem Statement

Generally, the CRPHC can be defined on an undirected graph $G(X,A)$ where $X = \{1,2,\ldots,n\}$ is the set of vertices and $A = \{[i,j], i,j \in X, i \neq j\}$ is the set of edges representing the existence of communication between two vertices i and j.

A vertex presents data, service or application. Each vertex is characterized by a confidentiality degree d_i and size s_i. Each edge is characterized by a communication frequency f_{ij} if the two vertices i and j are accessed by the same query and need some communication.

Initially, we consider that all vertices are hosted in the private cloud and the public one is empty as illustrated in figure 1. Because the incapacity of the private cloud to host all vertices, the decision maker must specify which vertices to move to the public cloud. After the partitioning vertices (resources) process, we will obtain a private cloud which contains the resources with high confidentiality and a public cloud which contains resources with low confidentiality as illustrated in figure 2.

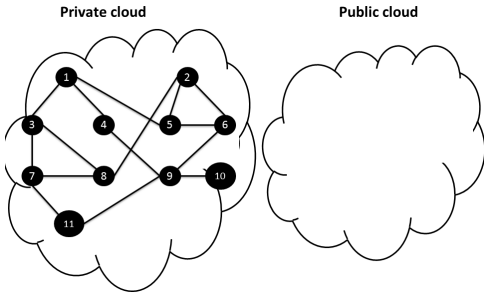

Figure 1: Hybrid cloud before partitioning process.

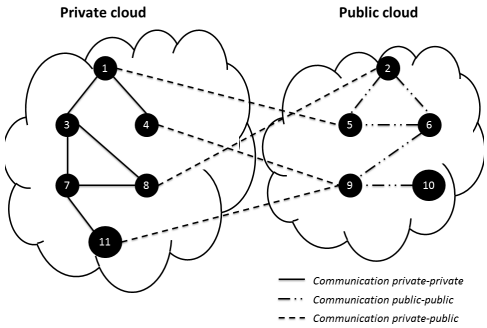

Figure 2: Hybrid cloud after partitioning process.

Usually, except confidentiality minimization, the partitioning process must also take into account the minimization of the total size of vertices affected to the public cloud and the minimization of the total communication frequency outside the private cloud (i.e. the *public-public* and *private-public* communication). In this work, we not interested to the minimization of the communication cost and thereafter we assume that $f_{ij} = 0, \forall i,j \in X$.

3.2 Mathematical Model

To formulate the mathematical model for CRPHC, we consider the following data and variables:

- n: number of vertices in the graph,
- X: set of vertices formed the graph,
- d_i: degree of confidentiality of each vertex $i \in X$. The affectation of values of degree of confidentiality is performed using the opinion of an expert based on transaction historic.
- s_i: size of each vertex $i \in X$,
- MS: the Minimal Size of ressources to host in public cloud. Indeed, the use of the public cloud is motivated by the insufficiency of the private cloud to host all vertices.
- $x_i \in \{0,1\}$: a binary decision variables. $\forall i \in X$, $x_i = 1$ if the vertex i is affected to the public cloud and $x_i = 0$ if it's affected to the private one.

Initially, the CRPHC can be formulated by a 0-1 linear program:

$$\text{Min Z} = \max_{i \in X} \{d_i x_i\} + \sum_{i \in X} s_i x_i \quad (1)$$

subject to

$$\sum_{i \in X} s_i x_i \geq MS \quad (2)$$

$$x_i \in \{0,1\}, \forall i \in X \quad (3)$$

Equation (1) represents the objective function of the CRPHC. It consists to minimize: (i) the maximum confidentiality degree between the vertices affected to the public cloud and (ii) the total size of resources to host in the public cloud. Inequality (2) represents the size constraint of the public cloud. Equation (3) represents the constraints of the decision variables.

We note that the objective function is composed by two inhomogeneous terms in term of their metrics. Indeed, the sizes and the confidentiality degrees are not belonging in the same values intervals. Henceforth, to eliminate this inconvenience, we use the normalized data $\bar{s}_i$ and $\bar{d}_i$ instead of s_i and d_i, $\forall i \in X$ where:

$$\bar{s}_i = \frac{s_i}{\max_{k \in X} s_k} : \bar{s}_i \in [0,1] \, \forall i \in X$$

$$\bar{d}_i = \frac{d_i}{\max_{k \in X} d_k} : \bar{d}_i \in [0,1] \forall i \in X$$

We also note that the proposed objective function is not linear. So, to linearize the model, in order to simplify its resolution by any mathematical models solver, we consider a new integer decision variable $\lambda \in \mathbb{N}$ to be the maximal confidentiality degree between verticies affected to the public cloud.

$$\lambda = \max_{i \in X} \{\bar{d}_i x_i\} : \lambda \in [0,1]$$

In the improvement model, λ must be minimized and each confidentiality degree of the vertices affected to the public cloud can not exceed λ. The modified model can be formulated as follows:

$$\text{Min } Z = \lambda + \sum_{i \in X} \bar{s}_i x_i : Z \in [0,2] \quad (4)$$

subject to

$$\sum_{i \in X} \bar{s}_i x_i \geq MS \quad (5)$$

$$\bar{d}_i x_i \leq \lambda, \forall i \in X \quad (6)$$

$$x_i \in \{0,1\}, \forall i \in X \quad (7)$$

$$\lambda \in \mathbb{N} \quad (8)$$

4 THEORETICAL BASIS OF CRPHC'S APPROACH

In this section, we describe our proposed approach to solve the CRPHC. To classify the resources, we are looking for grouping resources which have the closest degrees of confidentiality and to minimize size of resources which will be hosted in the public cloud. Resources classification must take into account to the already mentioned criteria such as:

- Minimizing degrees of confidentiality of resources (vertices) moved to the public cloud C_{pu}.
- The size of resources affected to C_{pu} must exceed slightly a fixed values MS. This constraint allows minimizing the storage cost of resources moved to the public cloud C_{pu}.

4.1 Principle

Our approach aims at partitionning n vertices to two clusters (Private Cloud C_{pr} and Public Cloud C_{pu}) with respecting certain number of criteria (already mentionned in the previous paragraph).

Step 1: The initial set of vertices will be patitioned into two clusters: The first cluster C_{pr} will contain confidential resources and the second cluster C_{pu} will contain non-confidential resources.

Step 2: The constraint of the MS of resources to host in public cloud MS must be verified. So two cases are possibles:

Case 1: the Total Size of resources affected to C_{pu} (TS_{pu}) is greater than MS, C_{pu} will be partitioned into two clusters CL_{pr} (Private CLoud) and CL_{pu} (Public CLoud). Then we have: $C_{pu} = CL_{pu}$ and $C_{pr} = C_{pr} \cup CL_{pr}$

Case 2: the Total Size of resources affected to C_{pu} (TS_{pu}) is smaller than MS, C_{pr} will be partitioned into two clusters CL_{pr} and CL_{pu}. Then we have: $C_{pr} = CL_{pr}$ and $C_{pu} = C_{pu} \cup CL_{pu}$.

Step 3: Repeat **Step 2** until MS is reached or exceeded.

Figure 3 illustrate the already described steps of the proposed approach.

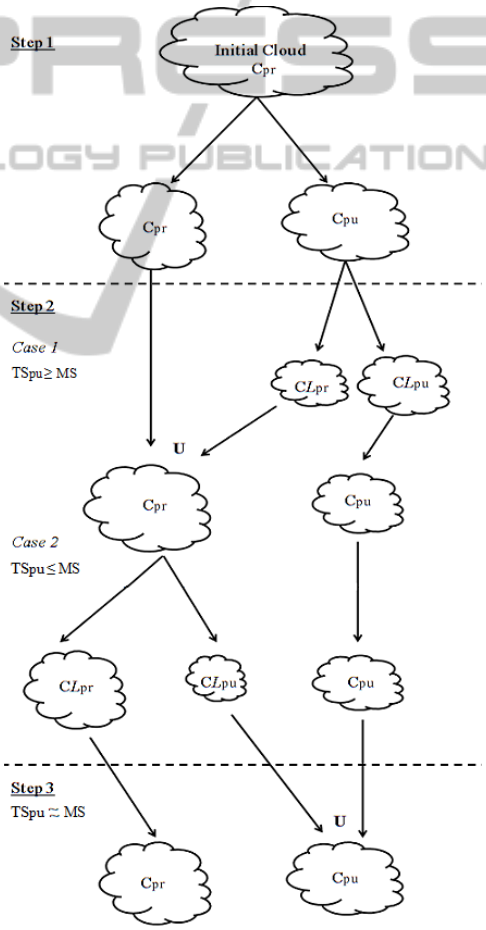

Figure 3: Illustration of approach process.

4.2 The CRPHC Algorithm

In this part, we present our solution to classifying resources into two clusters: private and public clusters.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

So we implement **CRPHC** Algorithm to classify resources into two clusters.

First, we suppose that all vertices (resources) are hosted in the private cloud, then each vertex is affected in the right cloud taking into account the previous criteria already mentioned. However to apply **CRPHC** Algorithm, a metric must be defined to classify a set of vertices.

In our case, the distance must take into account the degree of confidentiality d_i and the size s_i of each vertex $i \in X$. As a result, the private cloud C_{pr} will contain vertices which have the highest degree of confidentiality while minimizing the costs of storage in the public cloud. So, the public cloud C_{pu} will contain resources having lower confidentiality degree which will minimize the costs of storage in the public cloud.

4.2.1 Definition 1: (Distance)

We consider that each vertex is characterized by coordinates (s_i,d_i). So, the distance δ_{ij} between two vertices i and j will be defined as following:

$$\delta_{ij} = \sqrt{(\overline{s}_i - \overline{s}_j)^2 + (\overline{d}_i - \overline{d}_j)^2}$$

4.2.2 Definition 2: (Centroid)

A centroid is characterized by coordinates $(s_{\omega_k}, d_{\omega_k})$. These coordinates are calculated as following:

$$s_{\omega_k} = \frac{\sum_{i \in C_k} \overline{s}_i}{|C_k|},\ \forall k \in \{1,2\}$$

$$d_{\omega_k} = \frac{\sum_{i \in C_k} \overline{d}_i}{|C_k|},\ \forall k \in \{1,2\}$$

4.2.3 Algorithm

Our solution is based on a main algorithm (**CRPHC** Algorithm) which uses each time **Affect** Algorithm to classify different vertices of the graph into two clusters. The input of the CRPHC Algorithm is the graph to be partitioned. As is already mentioned, each vertex of the graph is characterized by a degree of confidentiality d_i and a size s_i. Then, the output is two clusters: private cloud C_{pr} and public cloud C_{pu}. For **Affect** Algorithm, the output is also a private cloud CL_{pr} and a public cloud CL_{pu}. Algorithm 1 describes the proposed approach to solve the CRPHC.

Initially, we assume that all vertices are hosted in the private cloud and the public cloud is empty. The **Affect** Algorithm will be applied to all the graph to give firstly two clusters: private cloud CL_{pr} and public cloud CL_{pu}. If the Total Size TS_{pu} of resources hosted in public cloud is greater than MS (see line 6), **Affect** Algorithm will be applied to public cloud CL_{pu}. This

Algorithm 1: CRPHC Algorithm.

```
   input  : G(X,A)/|X| = n.
   output: C_pr and C_pu where C_pr ∪ C_pu = X and
           C_pr ∩ C_pu = ∅.
 1 C_pr ← X;
 2 C_pu ← ∅;
 3 Affect(X)          /* Apply Affect (see
   Algorithm 2) */;
 4 C_pr ← {CL_pr};
 5 C_pu ← {CL_pu};
 6 if TS_pu ≥ MS      /* (TS: Total Size of
   resources affected to public cloud)
   */ then
 7     repeat
 8         X ← CL_pu;
 9         Affect(X);
10         C_pr ← C_pr ∪ CL_pr;
11         C_pu ← CL_pu;
12     until TS_pu > MS;
13 else
14     repeat
15         X ← CL_pr;
16         Affect(X);
17         C_pu ← C_pu ∪ CL_pu;
18         C_pr ← CL_pr;
19     until TS_pu < MS;
20 end
```

allows to decrease the Total Size of resources hosted in C_{pu}, indeed, vertices of CL_{pu} will be moved from the public cloud C_{pu} to the private cloud C_{pr} (see lines 7-12). So the **Affect** Algorithm will be applied to public cloud CL_{pu} until the MS is reached.

Likewise, if TS_{pu} is lower than MS (see lines 13-19), the **Affect** Algorithm will be applied to CL_{pr}. Indeed, the vertices of CL_{pr} will be moved from private cloud C_{pr} to public cloud C_{pu} until the MS is reached or exceeded.

Affect Algorithm consists in a first step to choose two vertices ω_{pr} and ω_{pu} from the graph (see lines 1-2). The choice of these vertices is performed using two functions $max()$ and $min()$.The function $max()$ choose the vertex with the maximal degree of confidentiality in the graph and the function $min()$ choose the vertex with the minimal degree of confidentiality in the graph. In this case, **Affect** Algorithm regroup vertices with higher confidentiality in one cluster (private cloud) and vertices with lower confidentiality in another cluster (public cloud).

To affect vertices to the right cluster (see lines 7-11), the idea is to compute the distance $\delta_{(i,\omega_k)}$, with $k = \{pr, pu\}$, between each vertex in the graph and each centroid. If the vertex is closest to ω_{pr}, it will be hosted to the private cloud CL_{pr} and if the vertex

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Algorithm 2: Affect Algorithm.

input : A set of vertices X to be partitioned.
output: Tow clusters CL_{pr} and CL_{pu} for the vertices hosted (respectively) in private and public clouds.

1 $\omega_1 \leftarrow max(X)$;
2 $\omega_2 \leftarrow min(X)$;
3 **repeat**
4 $CL_{pr} \leftarrow X$;
5 $CL_{pu} \leftarrow \emptyset$;
6 **for** *each* $i \in X$ **do**
7 **if** $\delta_{i\omega_{pr}} \leq \delta_{i\omega_{pu}}$ **then**
8 $CL_{pr} \leftarrow CL_{pr} \cup \{i\}$;
9 **else**
10 $CL_{pu} \leftarrow CL_{pu} \cup \{i\}$;
11 **end**
12 **end**
13 $old\_\omega_{pr} \leftarrow \omega_{pr}$;
14 $old\_\omega_{pu} \leftarrow \omega_{pu}$;
15 $new\_\omega_{pr} \leftarrow centroid(CL_{pr})$;
16 $new\_\omega_{pu} \leftarrow centroid(CL_{pu})$;
17 **until** $old\_\omega_{pr} == new\_\omega_{pr}$ *AND* $old\_\omega_{pu} == new\_\omega_{pu}$;

is closest to $\omega_{k_{pu}}$, it will be hosted to the public cloud CL_{pu}. Then, we compute centroids of the new clusters (see lines 15-16) and we repeat this process (see lines 4-16) until we have the stability of the two clusters i.e. we reach the same centroids in two successive iterations (see line 17).

5 EXPERIMENTAL EVALUATION

To apply and assess our approach, we need instances for CRPHC algorithm. Then we need to vary the following parameters:

- n: number of vertices of the graph,
- d: degrees of confidentiality of vertices of the graph,
- TS: Total Size of vertices of the graph,
- MS: Minimal Size of resources to host in public cloud.

But, it is not possible to find real cases or benchmarks based on previous parameters and able to varying them. This is why we need to create several graphs according to our need. For this reason we have developed a generator of graphs. Each graph is composed by six elements: number of vertices, vector of size of each vertex, a vector of confidentiality degree of each vertex, MS, matrix of execution frequency between two vertices and minimal degree of confidentiality in public cloud. In our work we just interest to the four first elements. The generated graphs are available on http://goo.gl/uvO8B8.

In this section, we will evaluate our solution with an optimal solution CPLEX. CPLEX is a computing tool of optimization (Aitha, 2014) which gives optimal solutions applied to a integer programming formulation.

So, to assess our solution we applied CPLEX to the same graphs that we used to test **CRPHC** Algorithm. Then we compare the results obtained with our solution and those obtained by CPLEX.

To better analyze and interpret results, we calculate the *Gap* between results given by **CRPHC** and those given by CPLEX. This *Gap* is given by:

$$Gap = (CRPHC\_OF - CPLEX\_OF)/cplex\_OF$$

Then, we recognize that a solution is called:

- Optimal: if the Gap associated is 0%,
- Excellent: if the Gap associated it does not exceed 15%,
- Incorrect: if the Gap associated exceeds 50%.

5.1 Varying Number of Vertices

These tests consist in varying the number of vertices of graphs (500, 1000, 1500, 2000, 2500 and 3000) and fixing the $MS = 20\%$, Total Size of the graph TS = 10000 and degree of confidentiality $d_i \in [0\% - 100\%]$.

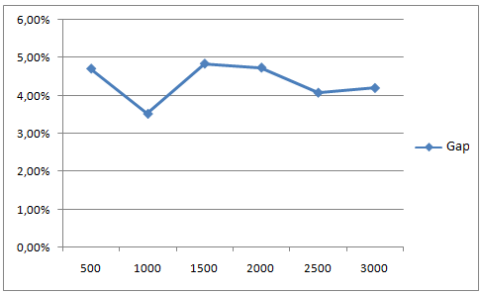

Figure 4: Gap for varying number of vertices.

Figure 4 shows that the Gap values are betwwen 3,4% et 5% so they are so close.

We remark that the Gap between our solution and optimal solution does not exceed 5%. So, in this case, our solution can be considered excellent.

5.2 Varying Degree of Confidentiality

For these tests, we fixed the number of vertices $n = 2000, MS = 20\%$ and Total Size of the graph $TS =$

10000. Then we varied the range of degree of confidentiality. To do this, we have chosen three ranges:

- $[0\% - 20\%]$ this range represents resources with low confidentiality,
- $[80\% - 100\%]$ this range represents resources with high confidentiality,
- $[0\% - 100\%]$ this range represents resources with low and high confidentiality.

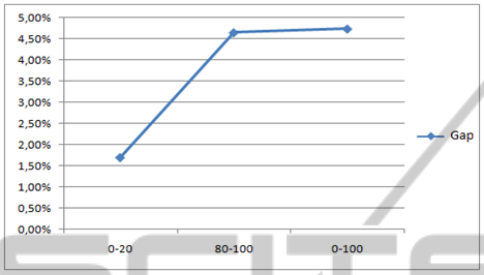

Figure 5: Gap for varying degree of confidentiality.

Figure 5 shows that the Gap for $[0\% - 20\%]$ is 1,6%. This value is low compared to values of other range of confidentiality. So, for resources with low confidentiality, the CRPHC_OF value is close to CPLEX_OF. Then the graph is almost stable in the order of 4,7% between $[0\% - 100\%]$ and $[80\% - 100\%]$.

The Gap between our solution and optimal solution does not exceed $4,7\%$. So, we can consider that our solution is excellent in this case.

5.3 Varying Total Size

For these tests, we fixed number of vertices $n = 2000, MS = 20\%$ and the degree of confidentiality $d_i \in [0\%$ - $100\%]$. Then we varied the Total Size of the graph (10000, 20000 and 30000).

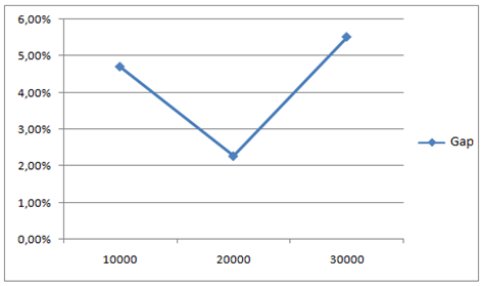

Figure 6: Gap for varying size.

Figure 6 shows that the best Gap value is given for $TS = 20000$.

Then we have the Gap between our solution and optimal solution does not exceed 6%. So, we can consider that our solution is excellent in this case.

5.4 Varying MS

For these tests, we fixed number of vertices $n = 2000$, degree of confidentiality $d_i \in [0\%$ - $100\%]$ and Total Size of the graph $TS = 10000$. Then we varied MS $(20\%, 30\%$ *and* $60\%)$ in the public cloud.

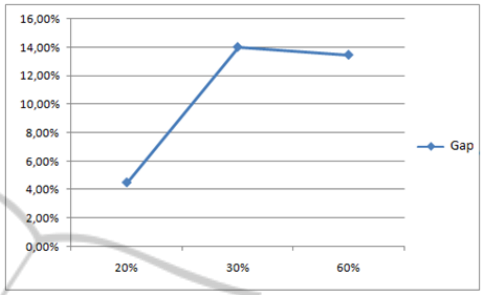

Figure 7: Gap for varying MS.

Figure 7 shows that the Gap for $MS = 20\%$ is 4,3%. This value is the best result given by our solution compared to $MS = 30\%$ and $MS = 60\%$.

Then we remark that the graph is almost stable in the order of 14% between $MS = 30\%$ and $MS = 60\%$.

The Gap between our solution and optimal solution does not exceed 14%. So, in this case, our solution can be considered excellent.

6 CONCLUSION AND FUTURE WORK

In this paper, we tackled a new approach for partitioning confidential resources between private and public components in hybrid cloud. Our objective is to ensure confidentiality by moving confidential resources to private cloud and resources with low confidentiality to public cloud. And also, minimizing the size of resources to host in public cloud. Then we have compared the results given by our proposed solution with optimum results given by CPLEX, and we have found that our results are acceptable.

In this work, we have supposed that hybrid cloud is composed by one private cloud and one public cloud. So to enlarge our work, we hope to propose an approach to partitioning confidential resources in hybrid clouds based on multitude of criteria which managing the allocation decision of each resource to one of the classes: private clouds and public clouds. Thus we place ourselves in a melting problem of sources of information (confidentiality, capacity, degree of dependence between resource, etc.). Then we focus on the notion of dynamicity such as confidentiality and sizes of resources that will be partitioned in hybrid clouds.

REFERENCES

Aitha, P. (2014). Cplex tutorial handout. *http://fr.scribd.com/doc/63956075/CPLEX-Tutorial-Handout*.

Chokhani, R. C. M. I. S. (2013). Cryptographic key management issues and challenges in cloud services. *National Institute of standards and Technology*.

Kantarcioglu, V. K. M. and Thuraisingham, B. (2011). Secure data processing in a hybrid cloud. *Pacific Asia conference on Intelligence and Security Informatics*.

Lamba, S. and Kumar, A. (2014). An approach for ensuring security in cloud environment. *International Journal of Computer Applications*.

Marwaha, M. and Bedi, R. (2013). Applying encryption algorithm for data security and privacy in cloud computing. *IJCSI International Journal of Computer Science Issues*.

Mehrotra, V. K. K. O. B. H. M. K. S. and Thuraisingham, B. (2012). Risk- aware data processing in hybrid clouds. *IEEE 5th International Conference on cloud Computing*.

Nepal, D. T. S. C. S. and Calvo, R. (2014). Secure data sharing in the cloud. *Security; Privacy and Trust in cloud Systems*.

Pilli, P. R. P. M. R. S. E. and Joshi, R. (2013). Improved technique for data confidentiality in cloud environment. *Networks and Communications*.

Ramakrishnan, T. Z. R. and Livny, M. (1996). Birch : An efficient data clustering method for very large database. *SIGMOD*.

Stoica, M. A. A. F. R. G. A. J. R. A. K. G. L. A. P. A. R. I. and Zaharia, M. (2009). Above the clouds : A berkeley view of cloud computing. In *a*.

Tata, F. B. N. T. S. and Moalla, S. (2012). Approximate placement of service based applications in hybrid clouds. *21st International conference IEEE WETICE*.

Wang, L. G. Z. H. S. Z. N. Z. J. and C.Jiang (2012). Multi-objective optimization for data placement strategy in cloud computing. *Springer*.

Wang, X. and Guo, W. (2013). A data placement strategy based on genetic algorithm in cloud computing platform. *10th Web Information System and Application Conference (WISA)*.

Zhang;, C. C. E. J. Y. and Madden, S. (2010). Schism : a workload-driven approach to database replication and partitioning. *VLDB; 36th International Conference on Very Large Data Bases*.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS