Hybrid heuristic algorithm for load balancing in the cloud

Hamza Rahhali[*] and Mostafa Hanoune[**],

[*,**] Hassan II University - Casablanca, Faculty of Sciences Ben M'sik
Laboratory of Information Technology and Modelling Casablanca Morocco

Summary
This article tends to the current advance in the area of the load balancing in the cloud where the primary difficulty is to minimize the energy consumed by the virtual machines, optimize resources and improve the performance. We suggest an effective methodology based on the introduction of a detailed modeling which take into account the QoS parameters in terms of performance, dependability, data security and cost, in addition to the integration of two highly hybrid heuristics. The computational experiment is done and we can enhance the objective function of a set of instances compared to several heuristics from the literature.

Key words:
Hybrid heuristic, load balancing, Cloud computing, Virtual machine.

1. Introduction

During the last decade, work in the field of cloud computing (CC) has exploded dramatically, due to technological advancements in computing and networking ([14], [2], [8]). Cloud computing is one of the most recent advancements in the information age, bringing together a much larger number of machines than the quantity of individuals with the heavy burden of making them work. The number of machines involved, the smooth operation of the services offered and the budgetary contemplations of the service providers make managing this new paradigm very complex in many areas. In view of [18], CC is characterized as a model for empowering ubiquitous, helpful, on-demand organize access to a shared pool of configurable computing resources, for example, systems, servers, stockpiling, applications, and administrations, which can be quickly provisioned and released with minimal 250 management exertion or service provider interaction. From now on, the growing number of Cloud Computing service proposition additionally leads to various research studies in all the issues that this new worldview raises, especially in QoS (Quality of Service). QoS brings together all the parameters of a cloud service provider infrastructure and strongly links it to users. The Cloud Computing QoS includes all the parameters that a service provider needs to focus on, clearing crosswise over different domains, from the pure performance of the system to the ecological effect of all perspectives cost, performance and operational safety. It is also important to remember the significance of the financial aspect for a service provider, and the analysis of all these parameters of QoS must be able to allow him to find the best.

An interesting topic related to cloud computing consists in considering the load balancing. In fact, the load balancing is a techniques used to dispersing a larger processing load to littler processing nodes for improving the overall performance of system. In other words, it is helped to distribute the dynamic workload over various nodes to guarantee that no single node is overloaded. In cloud computing environment load balancing is required disperse the dynamic nearby workload equally between all the nodes ([6], [25], [3], [5]). From one hand, load balancing helped systems and assets by giving a Maximum throughput with minimum response time. From the second hand, it helps in reasonable distribution of processing asset to accomplish a high user satisfaction and appropriate Resource use, moreover it contributes in implementing fail over, adaptability, and staying away from bottlenecks.

Cloud computing is construct directly in light of the idea of virtualization. This concept gives an abstraction layer between the physical equipment utilized and the diverse operating systems and applications that you need to use. Indeed, the use of virtual machines makes it conceivable to run multiple systems in parallel on the same physical machine, which is essential for setting up the operation of services within a cloud. In this context, different criteria for optimizing the allocation of virtual machines in physical machines have been proposed, for instance: the energy consumption that takes environmental issues into account, response time that allows for pure performance measurement, robustness that reassures service providers and users about the likelihood of being affected by a failure of the system, and the dynamism that ensures a certain reserve of performance in case of peak traffic. In this regard, the literature coats several researches. Sotomayor [23] proposed lease and virtual machine to give computational asset needs in teaching and researching. They displayed First-Come-First-Serve (FCFS) and backfilling scheduling algorithms for schedule user leases, and utilizing a greedy algorithm to map all identical VMs, which has a place with same user lease, onto same physical

Manuscript received April 5, 2018
Manuscript revised April 20, 2018

machine. Inconvenience of the greedy VM allocation algorithm is that two distinct leases could not be mapped to same physical machines. In [12], the authors proposed two power-aware VM allocation algorithms that speak to a few mixes of First Fit Decrease (FFD) and most brief length time heuristics. In spite of the fact that the VM allocation algorithms can decrease total energy consumption for computing physical machines, these VM allocation algorithms in [12], nonetheless, don't prompt ideal arrangement. Mathematical programming approach has been connected in customary scheduling problems on non-virtualized systems for a long time keeping in mind the end goal to discover an ideal schedule for performance ([21], [7]), or minimize energy utilization of heterogeneous computer clusters [1]. These works did not utilize virtualization and were not appropriate for virtual machine scheduling. As of late, there are some fascinating works utilizing mathematical programming for scheduling problems on virtualized systems such as [24]. Speitkamp and Bichler [24] proposed mathematical programming model for server solidification, in which each service was executed in a VM, to lessen number of utilized physical machine.

This work is devoted to the application of hybrid heuristics in order to balance the load in the Cloud. Our commitment to all past works, lies firstly in the introduction of a detailed modeling based on QoS parameters in terms of performance, dependability, data security and cost. Secondly, two new heuristics

are presented, the first is based on the genetic method and the simulated annealing while the second consists of the genetic method and the Tabu method.

The rest of this paper is organized in the accompanying way. As a matter of first importance, we present the mathematical model, as well as the resolution methods used. Next, we present our proposed method that is intended to solve the problem going through the definition of heuristics and the type of hybridization. Besides, we evaluate the new method comparing to other heuristics from the literature. A detailed comparative study is exhibited in order to provide perfect conclusions. At last, some finishing remarks are talked about.

2. Metric and Method of Resolution

2.1 Mathematical model

The term Quality of Service (QoS) refers to the ability to provide a service (including communication support) that meets the requirements for response time and bandwidth. QoS indicates the levels of performance, reliability, and availability given by an application and by the stage or foundation that hosts it. QoS is key for cloud users, who anticipate that providers will convey the publicized quality attributes, and for cloud suppliers, who need to locate the correct tradeoffs between QoS levels and operational costs. However, finding ideal tradeoff is a difficult choice problem, frequently exacerbated by the presence of service level agreements (SLAs) determining QoS targets and practical penalties associated to SLA violations.

Our objective function will be based on the four QoS metrics, namely: the energy consumption E, the response time RespT, the robustness Rob represented by the average number of virtual machines allocated on each of the physical machines, as well as the dynamism Dyn which represents the average of free CPU capacity on each of the lit physical machines can be used if there is a very rapid increase in the number of requests to be processed. Bearing in mind that energy, response time and robustness are metrics to be minimized, unlike the dynamism metric, which must be maximized. This is our objective function:

Fobj = E+RespT+Rob-Dyn (1)

The used parameters are written as follows:

- $E=\sum_{i=1}^{n_k} P(Fi)^{h_i}\times T_{rep}^{h,k}$ (with $P(Fi)$ the power formula) (2)
- $T_{rep}^{h,k}=\max(\frac{NbInstr^{v}}{\omega_{f,cpu}^{v,h}})$ (with $NbInstr^{v}$ the number of instructions to be executed by the virtual machine v) (3)
- $Rob=\sum_{i=0}^{n_k}\frac{n_V^{h_i}}{n_H}$ (4)
- $Dyn=\frac{\sum_{i=0}^{n_h}\varepsilon_{f,cpu}^{h_i,k}-\omega_{f,cpu}^{h_i,k}}{n_H}$ (5)

2.2 Resolution Methods

A Metaheuristic is an optimization algorithm aimed at solving difficult problems for which no more efficient classical method is known. Metaheuristics are often algorithms using probabilistic sampling. They try to find the global optimum of a difficult optimization problem, without being trapped by the local optima. There are many different Metaheuristics, ranging from simple local search to complex global search algorithms. These methods, however, use a high level of abstraction, allowing them to be adapted to a wide range of different problems. There is a very large number of well-known Metaheuristics in the literature, We find evolutionary algorithms, among which: there are evolution strategies, genetic algorithms, differential evolution algorithms, and distribution estimation algorithms. There are also other Metaheuristics such as the simulated annealing, the ant colony, the particle swarm optimization, and the Tabu search, etc. In our research, we opt for three Metaheuristics. First, the simulated annealing method since it can deal with arbitrary systems and cost functions, it statistically guarantees finding an optimal solution, and it generally gives a good solution. Secondly, the Tabu search because it reduces in the number of circuit simulations required to find a feasible solution. Finally, we used the genetic algorithm for several reasons namely the capacity of simultaneous optimization of the multiple phases or properties of a material in a single run, the facility of the incremental re-optimization of the whole system as more data is made available for either additional phases or material properties not included in previous runs, and the successful global optimization in the presence of multiple local minima in the parameter space.

2.2.1 Tabu search

The Tabu search [9] proposed by Glover in 1986, it is a Metaheuristic that aides a neighborhood heuristic search method to investigate the solution space past nearby optimality. One of the major parts of Tabu Search is its utilization of versatile memory, which makes more adaptable search behavior. Tabu search can be seen as starting similarly as ordinary nearby or neighborhood search, continuing iteratively from one point (solution) toward another until a picked end criterion is satisfied. Every solution x has a related neighborhood $N(x) \subset X$, and every solution $x' \in N(x)$ is come to from x by an activity called a move. Here is the algorithm:

Algorithm 1 Tabu Search Algorithm

```
Choose an initial solution i in S (the set solutions)
X ← i and k ← 0
repeat
   apply k ← k + 1 and generate a subset solutions in N (i, k)
   choose the best solution i' among the set of neighboring solutions N (i, k)
   i ← i'
   if f(i) ≤ f(X) then
      X ← i
   end if
   update the T list of Tabu movements
until Stop criteria is met
return X
```

2.2.2 Simulated Annealing

Annealing is alluded to as tempering certain combinations of metal, glass, or crystal by warming above its melting point, holding its temperature, and then cooling it gradually until the point that it sets into a flawless crystalline structure. This physical/chemical process produces high-quality materials. The simulation of this process is known as simulated annealing (SA) ([4], [15]). The defect-free crystal state corresponds to the worldwide minimum energy arrangement. There is an similarity of SA with an optimization methodology. In general, the simulated annealing algorithms function as takes after. At each time step, the algorithm arbitrarily choose a solution near than the current one, measures its quality, and afterward chooses to move to it or to remain with the present solution in light of both of two probabilities between which it picks based on the way that the new solution is preferable or more terrible over the present one. Amid the pursuit, the temperature is continuously diminished from an underlying positive incentive to zero and influences the two probabilities: at each progression, the probability of moving to a superior new solution is either kept to 1 or is changed towards a positive esteem; rather, the probability of moving to a worse new solution is dynamically changed towards zero. Here is the detailed algorithm:

Algorithm 2 Simulated Annealing Algorithm

```
Initialize a solution X
n ← 0
repeat
   i ← X
   choose j at random from the neighbors of i
   if f(j) ≤ f(i) then
      X ← j
   else
      if (Random <exp( f(i)-f(j)/T) then
         X ← j
      end if
   end if
   n ← n+1
until Stop criteria is met
return X
```

2.2.3 Genetic Algorithm

Genetic algorithm (GA) is a Metaheuristic propelled by the procedure of natural determination that has a place with the larger class of evolutionary algorithms (EA). Genetic algorithms are normally used to produce high-quality solutions to optimization and search problems by relying on bio-inspired operators for example, mutation, crossover and selection. The process of natural selection begins with the choice of fittest people from a populace. They deliver offspring which acquire the attributes of the guardians and will be added to the new generation. In the event that parents have better fitness, their offspring will be superior than parents and have a large chance at surviving. This procedure continuous iterating and toward the end, a generation with the fittest individuals will be found. This idea can be connected for a search problem. We consider a set of solutions for a problem and choose the set of best ones out of them. Five stages are considered in a genetic algorithm:

- Initial population
- Fitness function
- Selection
- Crossover
- Mutation

Here is the algorithm:

Algorithm 3 Basic Genetic Algorithm

```
Initialize population
repeat
  repeat
    crossover
    mutation
    phenotype mapping
    fitness computation
  until population complete
  selection of parental population
until termination condition
```

3. Heuristic Hybridization

The literature proposes three types of approaches to solve these kinds of problems. In the first place, exact methods give the optimal solution in the search space and they have an exponential running time in worst case. We cannot tackle NP-Hard problems using exact methods. In the second place, the heuristic methods wich are solution techniques that can quickly provide a solution in a sensible quality. We will talk in detail about this type of resolution. The third class of approaches of resolution is an exceptional type of heuristics, yielding a worst-case upper bound of the ratio between the cost of an approximate solution and the cost of optimal solution. these produced an error ensure.

3.1 Heuristics

The principal target of a heuristic is to provide a solution in a sensible time frame that is sufficient for taking care of the current problem. This solution may not be the best of the considerable number of solutions for this issue, or it might just estimate the correct solution. In any case, it is as yet an important way because finding it does not require a restrictively prolonged stretch of time.

Heuristics may give results independent from anyone else, or they may be utilized as a part of conjunction with enhancement calculations to enhance their effectiveness (e.g., they might be utilized to produce great seed esteems). Results about NP-hardness in software engineering make heuristics the main suitable choice for an assortment of complex optimization problems that should be routinely tackled in true applications.

Heuristics underlie the entire field of Artificial Intelligence and the computer simulation of reasoning, as they might be utilized as a part of circumstances where there are no known algorithms.

3.2 Hybridization Type

In this area we will center around two explanations behind hybridization in Evolutionary Algorithms: Enhancement in execution of the EAs and quality of the got solutions.

Much work has been devoted to the hybridization of different recombination operators in Genetic Algorithms. Some great cases in this line are the fuzzy logic controller proposed in [10] to control the support of various crossover operators or the experiments by Hong [11] where the interest of a few hybrid crossover operators balanced in view of the advance brought into the population by utilizing every one of them. Different algorithms propose island GAs where every island develops a population by methods of recombination operators with various qualities, attempting to accomplish a decent exchange off between exploration and exploitation methods by controlling transitory procedures [26]. At last, a few examinations propose the utilization of two distinct populations: one for the problem itself and another for the set of operators that will be utilized [16].

However, hybridization isn't limited to occurring inside the same developmental worldview. A few investigations propose Hybrid Evolutionary Algorithms where at least two distinct algorithms team up through the inquiry procedure. This is the situation for the GA-EDA algorithm [19] where a GA and an EDA are joined and connected to the determination of both discrete and continuous problems. Shi et al. [22] proposed a hybrid EA-PSO algorithm where the two subsystems are done in parallel, and a couple of people are traded each age. Tseng and Liang [27] proposed a hybrid approach that joins Ant Colony Optimization (ACO), a Genetic Algorithm and a

Local Search for the Quadratic Assignment Problem (QAP). In their tests, elective phases of ACO and GAs are executed. The pheromone esteems vital for the ACO are also updated while the algorithm is in the GA stage to guarantee the correct behavior of the ACO. The Local Search enhances solutions delivered by both algorithms.

At last, a third method of hybridization in Evolutionary Algorithms manages the encoding of solutions. Studies completed by [13] on the difficulty of various optimization problems have estimated one of the parts of problem complexity by relating the difference between fitness function values and the Euclidean distance in the solution space. Test comes about demonstrate that an interpretation of the fitness landscape can make a problem easier or harder to solve. However, few works have mulled over this issue. In [20] the authors propose a Genetic Algorithm to solve diverse optimization functions where the people can be encoded with a Cartesian or a pseudo-polar coding and be joined by utilizing both of them. In [17], five Genetic Algorithms are consolidated to explain a few occurrences of the TSP (Travelling Salesman Problem). Four of these GAs utilized an integer path representation, while other utilized a real ranking encoding.

4. Proposed Algorithm

A hybrid method is a search method consisting of at least two distinct search methods. It consists in exploiting the respective advantages of several methods by combining their algorithms according to a synergetic approach

A hybrid method may be bad or good depending on the choice and roles of its components. To define an effective hybrid method, one must know how to characterize the advantages and the limits of each method.

The methodology to proceed to an optimal solution depends on heuristic search in a given population and the direction of search is chosen by the diverse operators utilized.

4.1 GA-Tabu

It is a blend of basic GA and Tabu search algorithm. The object is to circulate the optimization task into two sections, the GA initially plays out the search and then the refinement is finished by the Tabu search algorithm. Both the algorithm keep running in parallel, after n iteration of Tabu search, the local optimal solution is infused into the current generation. The Tabu search method finds the local minima which supplements the GA to catch global minima. Here is the full algorithm:

Algorithm 4 GA-Tabu Search Algorithm

```
Initialize population
repeat
  repeat
    crossover
    Tabu search
    phenotype mapping
    fitness computation
  until population complete
  selection of parental population
until termination condition
```

4.2 GA-Simulated Annealing

Inside our Hybrid algorithm, Simulated annealing (SA) is utilized to optimize every single of the top N individuals in the developed population. However, it is not important to utilize the SA method to optimize the top N individuals at each generation, generally the Hybrid algorithm will have a very long run time cost. The same hybridization technique as GA-Tabu is used. Here is the full algorithm:

Algorithm 5 GA-Simulated Annealing Algorithm

```
Initialize population
repeat
  repeat
    crossover
    simulated annealing
    phenotype mapping
    fitness computation
  until population complete
  selection of parental population
until termination condition
```

5. Computational Results

We have compared the performance of the hybrid GA-Tabu and hybrid GA-SA with some of the recent algorithms in the literature that are shown in the next Table in terms of execution time and the consumed energy.

The properties of the physical machines are shown below:

- Number of physical machines: 110
- Maximum capacities of CPU (in MIPS) and Memory (in MB): 2000 MIPS / 2500Mo
- CPU frequency range
- Heterogeneity of powers delivered by physical machines according to CPU frequencies

The properties of virtual machines are presented below:

- Number of virtual machines: 400
- Capacities (CPU in MIPS and Memory in MB) of

virtual machines: 200/400/600/800 (MIPS and MB)

- The maximum percentage of reconfiguration of the CPU capacity accepted by the virtual machines: 20% of the CPU capacity of the virtual machine concerned
- Number of instructions to be executed by virtual machines: between 10 and 110 the CPU capacity of the virtual machine machines according to CPU frequencies
- Maximum migration time: 1 second

Finally, the configuration of the genetic algorithm for allocating the virtual machines at each moment of reallocation is as follows:

- Size of the random population of departure: 1500
- Size of the working population: 120
- Number of mutations: 100
- Number of crossings: 90
- Fixed number of generations: 600

A simple neighborhood is used for our problem, it is the set of solutions that can be built by swapping two virtual machines in a given solution.

Table 1: The performance of the hybrid GA-Tabu and hybrid GA-SA

Method	Running time	Consumed energy
SA	18 ms	4633565.5
Tabu	52 ms	4898627.0
GA	1566	4395289.5
GA-Tabu	1291	4296607.5
GA-SA	1116	4545965.0

6. Conclusion

Our main goal in this paper was to consider the load balancing in the cloud computing. For this purpose, a very effective way was suggested in identifying the global optimum solution to difficult function optimization problems consists on make the Genetic Algorithm (GA) cooperate with another local search algorithm. We have realized this idea by building two highly hybrid algorithms, the first is based on joining the GA with the Tabu search (TS), while the second is to combine the GA with the Simulated Annealing (SA). The table shows that our new hybrid algorithms provide better results in terms of running time (specially for GA-SA) and better results in terms of minimization of energy (specially for GA-TABU)

This new approach consists first of all in introducing a modeling based on Qos parameters such as the energy consumption, robustness, dynamism and the response time, and secondly the integration and comparison of two hybrid heuristics with well-known heuristics. in order to reach a close optimum in terms of running time and consumed energy.

Further work will focus on the parallelization of the hybrid heuristics used un this paper, by exploiting the accessibility of numerous processors in order to enhance the running time.

Acknowledgments

The authors wish to thank the anonymous referee for valuable comments and corrections.

References

[1] I. Al Azzoni. Power-Aware Linear Programming Based Scheduling for Heterogeneous Computer Clusters. Future Generation Computer Systems, pp. 745–754, May 2011.

[2] L. Badger, T. Grance, R. Patt-Corner, J. Voas. Draft cloud computing synopsis and recommendations. NIST special publication, 2011.

[3] Bala, Anju and Chana, Inderveer. A survey of various workflow scheduling algorithms in cloud environment. 2nd National Conference on Information and Communication Technology (NCICT), 2011.

[4] V. Cerny. Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. J Optim Theory Appl, pp. 41–-51, 1985.

[5] 2011} Chaczko, Zenon and Mahadevan, Venkatesh and Aslanzadeh, Shahrzad and Mcdermid, Christopher. Availability and load balancing in cloud computing. International Conference on Computer and Software Modeling, 2011.

[6] Chaudhari, Anand and Kapadia, Anushka. Load Balancing Algorithm for Azure Virtualization with Specialized VM. algorithms, pp. 2, 2013.

[7] J. R. Correa and M. R. Wagner. LP-based online scheduling: from single to parallel machines. Math. Program. pp. 109-–136, 2009.

[8] G. Galante, L. C. E. d. Bona. A survey on cloud computing elasticity. in: Proceedings of the IEEE Fifth International Conference on Utility and Cloud Computing, pp. 263-–270, 2012.

[9] Fred Glover. Future Paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research. pp. 533-–549. 1986.

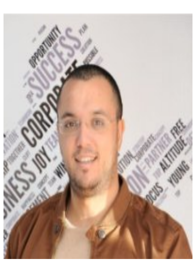
[10] F. Herrera and M. Lozano. Adaptation of genetic algorithm parameters based on fuzzy logic controllers. Genetic Algorithms and Soft Computing, pp. 95–-125, 1996.

[11] T. Hong, H. Wang, W. Lin, and W. Lee. Evolution of appropriate crossover and mutation operators in a genetic process. Applied Intelligence, pp. 7–-17, January 2002.

[12] N. Q. Hung, N. Thoai, and N. T. Son. Performance constraint and power-aware allocation for user requests in virtual computing lab. Journal of Science and Technology (Vietnam), Special on Int. Conf. on Advanced Computing and Applications (ACOMP2011), pp. 383-–392, 2011

[13] T. Jones and S. Forrest. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. Proceedings of the Sixth International Conference on Genetic Algorithms, pp.184-–192,1995

[14] W. Kim, Cloud computing: Today and tomorrow. Journal of object technology, pp. 65–-72, 2009.
[15] S. Kirkpatrick , Jr. CD. Gelatt, MP. Vecchi. Optimization by simulated annealing. Science, pp. 671–-680, 1983
[16] S.Koh, S.Leow,and K.Loke. An adaptive genetic algorithm for permutation based optimization problems.In Proceedings of the AIAI2005--Second IFIP Conference on Artificial Intelligence Applications and Innovations, 2005.
[17] A.LaTorre,V.Robles, and S. Muelas. Using multiple offspring sampling to guide genetic algorithms to solve permutation problems. Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, pp. 1119–1120, 2008
[18] P. M. Mell, T. Grance. the nist definition of cloud computing, Tech. rep., National Institute of Standards \& Technology, pp. 800--145, 2011.
[19] V. Robles, J. Pena, P. Larraaga, and V. Herves. A new hybrid cooperative search evolutionary algorithm. Towards a New Evolutionary Computation. Advances in the Estimation of Distribution Algorithms. Series: Studies in Fuzziness and Soft Computing, pp. 187-–219, 2004.
[20] T. Schnier and X. Yao. Using multiple representations in evolutionary algorithms. In Proceedings of the 2000 Congress on Evolutionary Computation, pp. 479–-486,2000.
[21] A. Schulz. Scheduling unrelated machines by randomized rounding. SIAM Journal on Discrete Mathematics, pp. 450, 2002
[22] X. Shi, Y. Liang, H. Lee, C. Lu, and L. Wang. An improved ga and a novel pso-ga-based hybrid algorithm. Information Processing Letters, pp. 255–-261, 2005.
[23] B. Sotomayor, K. Keahey, and I. Foster. Combining batch execution and leasing using virtual machines. HPDC 08 Proceedings of the 17th international symposium on High performance distributed computing, pp. 876-–896, 2008
[24] B.Speitkamp and M. Bichler. A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centers. IEEE Transactions on Services Computing, pp. 266-–278, 2010.
[25] Nayandeep Sran, Navdeep Kaur. Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing, 2013
[26] E.Takashima, Y.Murata, N. Shibata, and M. Ito. Self adaptive island ga. In Proceedings of the 2003 Congress on Evolutionary Computation 03, pp. 1072-1079. 2003.
[27] L. Tseng and S. Liang. A hybrid metaheuristic for the quadratic assignment problem. Computational Optimization and Applications, pp. 85-–113, 2006.



Hamza Rahhali received a M.Sc. in networks and telecommunications from Aix Marseille University, Marseille, France, in 2011. His research interests include load balancing algorithms and Scheduling algorithms in cloud computing

Mostafa Hanoune He is currently a Professor in the Department of Computer Science and Maths, University Hassan II of Casablanca, Morocco. His research interests are in the areas of computer sciences, data mining, Information Technology and Cloud computing.