# A Comparative Study of Levels Scheduling Algorithms in the Cloud Computing

Manel Femmam[1], Okba Kazar[1], Sohaib Baarir[2], and Mohamed El-kabir Fareh[3]

***Abstract*—**Cloud computing is a commercial model for distributed computing that provides users with resources or computing services with different specifications. These users have many requests and dynamic targets which lead to different levels of service. Depending on the precision of these applications or the business service orientation, three scheduling levels were identified (service level scheduling, task level scheduling, virtual machine level scheduling). The availability and the load on the resources increase with time, so scheduling in the Cloud is a complicated problem especially the need to ensure different (QoS) quality requirements. This article presents a bibliographic research about works dealing with the various scheduling levels in the Cloud Computing; a comparison between different levels scheduling algorithms related to this work has been done.

***Keywords*—** Cloud Computing, Scheduling, Scheduling tool in the cloud, Quality of Service (QoS).

## I. Introduction

THE Cloud Computing is a computer science paradigm, emerging from distributed computing in which applications; data and infrastructure are proposed as a service that can be consumed anywhere and anytime, in a flexible and transparent way. This simplification allowed developing the concept of resources on demand with the "pay-as-you-go" economical model.

The scheduling problem is one of the major challenges of the cloud. Three scheduling levels were identified: service level, task level and virtual machine level (VM) [1]. Therefore, it is essential for the cloud to integrate an efficient scheduler to optimize different criteria depending on suppliers and users.

The scheduler finds a virtual resource allocation among the various suppliers (a deployment plan) that optimizes QoS (Quality of Service) parameters that respects the placement constraints but also that manages other strong constraints (restrictions to perform certain tasks on certain types of resources). These (QoS) parameters can be defined and offered by various service contracts, SLAs (Service Level Agreements). This scheduling mechanism can be modeled in two ways. In the first way, a direct link is established between the client and the infrastructure services provider, this type of cloud model is twofold. In the second way, we find a broker that acts as a mediator between the client and the provider called three levels model [1].

The rest of this paper is organized as follows: section 2 defines the scheduling problem and its different levels, then section 3 presents the works that treats scheduling problem in each level. In section 4, a comparison was made between these different algorithms with different QoS considered parameters. Section 5 and 6 provide a synthesis and a conclusion with future work respectively.

## II. Scheduling in the Cloud

There are different entities in the Cloud scheduling model. In the service level scheduling, two types of entities are considered: client and cloud provider. For the task level scheduling, we consider a client service set and a set of instance type of VMs. In the third VMs scheduling level, assignment process actors are the set of VMs and the set of data center that make up the cloud. Therefore, the aims of all scheduling levels are to allocate sets of (clients services requests, client applications, VMs) to the set (cloud providers, VMs instance types and data center.

## III. Scheduling Levels

The scheduling process is applied to the different levels according to Cloud's nature of service; if it is market-oriented (scheduling in the service level and task level), if non-market-oriented (VM-level scheduling) [11].

### A. Service-level scheduling

This scheduling level is static and concerns a part of the resource management layer. The main criteria to optimize at this level are the profile.

### B. Task-level scheduling

This type of scheduling is dynamic and adapted to cloud changes. Its objective is to optimize the assignment according to the QoS constraints of each task and of each client, all with minimizing the total execution price and the time cost. The scheduler to the task level is dedicated to a single data center (Cloud), and it is unable to manage the resource of another cloud from another provider. The matching process between the task and the VM is made through a broker.

### C. VM-level scheduling

It is the lowest scheduling layer, it is used to provide with the set of VMs requested by the task in task level scheduling. The purpose of this level is to find the best scheduling of virtual machines on hosts that make up data centers (Cloud).

[1]LINFI Laboratory, Computer Science Department, Biskra University, Algeria
[2]LRDE, Le Kremlin-Bicêtre, France
[3]Computer Science Department, Biskra University, Biskra, Algeria

## IV. Works on the Scheduling Algorithms

In this section, we present a state of art dedicated mainly to scheduling algorithms proposed recently and used in cloud computing.

### *A. Service-level scheduling*

In **[2]**, the work presents a new architecture for brokerage in the Cloud and management of virtual machines in a multi-cloud environment to optimize the applications placement in this environment. The algorithm is based on Integer Programming Formulation.

The client can guide its VMs allocation by specifying a minimum of performance and maximum budget, as well as constraints, load balancing and hardware configuration of each VM.

Providers offer several configurations of virtual machines. Cloud broker has two main actions: the first action is to find the optimal placement of virtual resources of a virtual infrastructure through a set of cloud providers; the second one is the management and control of these virtual resources through the manager of virtual infrastructures. The latter is responsible for deployment of each VM in the selected cloud that is characterized by VMs models, as well as life cycle management of these ones. Then, a plan generated by the scheduler, which is a broker component of Cloud, this plan contains a list of VMs models. Each model includes targeted cloud provider to deploy VMs and specific attributes for the selected provider, such as the identifier of the VM image in this cloud.

In **[3]**, the authors proposed a multi-objective genetic algorithm MO-GA for jobs scheduling on cloud virtual machines. The algorithm minimizes energy consumption and maximizes provider's services profit under the delay constraint. The first MO-GA step is the generation of the initial population with two methods (random and greedy).

After the calculation of fitness, the solutions have the best rank (best fitness) are stored in the Pareto archive that contains non-dominating solutions generated through generations. The selection operator is the tournament, based on elitism strategy and crowding. Reproduction operators are crossover and mutation. Finally selecting the best solution in Pareto archive is done according to the current user preference between the two objectives.

Modular broker architecture was presented in **[4]**, based on different scheduling strategies using different optimization criteria (cost optimization and performance optimization), constraints (performance budget and types of instance) and environmental conditions (static / dynamic). The Cloud scheduler is a main component of cloud broker that is responsible for making scheduling decisions based on a dynamic pricing system, of users' demands and different performances of instances types. The scheduler can be configured to work with the different scheduling policies (cost optimization policy and performance optimization policy) based on different optimization criteria such as the cost of service, service performance, etc...Depending on these policies, the scheduler performs optimal deployment of different component services.

The work in **[5]** proposes a market-oriented package for hierarchical scheduling in a distributed cloud. This approach processes hierarchically level service and task level scheduling, which every workflow instance is mapped on the cloud service; the objective of this approach is to minimize the execution time, cost and time of processing. For service level scheduling, a random algorithm was established, whereas for the task level one, there are three meta-heuristics (genetic algorithm, optimization through ants' colony and optimization through particle swarm). A local scheduler is responsible for the latter level, which its function is to optimize the task-VM assignment in the data center of the Cloud. The first step is to get QoS constraints for each task, while the second step is to optimize the task-VM assignment for a data center portion to reduce the cost of overall performance of the latter. An initial list task to VM with a reasonable size should be used as an input to the task level scheduling algorithm; each virtual machine has its own local list of tasks. To start running the algorithm, a local scheduler shall acquire the current list of VMs and can be automatically combined with an embedded tasks graph that is made-up of all tasks. The purpose of this step is to generate an optimal schedule or close to the optimal one by a specific meta-heuristic algorithm. It can reduce the overall execution time of the system and the cost of data center, while meeting the QoS constraints of workflow and non workflow tasks.

**NB:** This work treats two levels at the same time: service-level scheduling and task-level scheduling.

### *B. Task-level scheduling*

The aim of the work **[6]** is to optimize the allocation of requests of VMs in a brokerage infrastructure in the three levels cloud: the client, the cloud provider and the cloud broker. The role of this latter is to find the best configuration of resources proposed by the supplier to meet demands of VMs. The aim of this optimization is minimizing the response time and the cost of VMs instances while satisfying the client and to maximize the broker's profit.

This model defines two types of services; the infrastructure service connected to provider and a business service connected to the broker. The client exposes his demands with their QoS constraints into two types. The first type is stated by the client. The second one is derived and used by the algorithm. To solve this problem, a new approach for the cloud brokering using a multi-objective genetic algorithm (MOGA) has been proposed to provide a set of Pareto optimal allocations on the best instances with minimal cost and response time.

In **[7]**, the author presents in the Pareto approach, the NSGA-II algorithm. This latter is a population-based meta-heuristics adapted to solve a multi-objectives scheduling problem of workflow in the IaaS infrastructure.

Multiple QoS metrics as (the makespan, the cost, reliability, and availability) as well as constraints on these metrics (time, budget and others), and strong constraints (a task Ta has to be executed on services like MVb or MVc) are taken into consideration to resolve this problem. The aim of the NSGA-II algorithm is to provide the user with a set of compromise solutions, instead of a single solution according to the requirement of QoS.

This work **[8]** develops an approach through aggregating objectives for solving the multipurpose scheduling problem of workflows in the IaaS virtualized infrastructure. In this approach, a genetic algorithm (GA) is proposed, which takes into account multiple service qualities (QoS) metrics: The completion time or makespan, the cost, reliability and availability can be used to determine to what extent the set of allocated virtual machines are used. The algorithm begins its running by a random initial generation. Each task of the solution has a rank that allows the indication of its execution order in the case of existing of two or several tasks that can be launched on the same time and on the same VM.

The second step is the scheduling process that is performed on each individual (solution) in order to satisfy the precedence relation between tasks. Each individual will be assessed using a fitness function. The genetic operators: selection, crossover, mutation and replacement will be applied to individuals to ensure the development of solutions. The stopping criteria of this algorithm is the number of iterations or until the overall value of the fitness shall no more develop.

A genetic algorithm based on the viral infection (GA*) is an improved genetic developed algorithm, taking into account multiple QoS metrics such as makespan, the cost, reliability, and availability more constraints on these metrics (the duration, the budget) and strong constraints that force to perform certain tasks on certain VMs. The same steps of the genetic algorithm shall be adopted in GA* except the penalty integrated to the fitness function to satisfy the QoS constraints, and a viral infection operator to process severe constraints specified in the SLA **[9]**.

### *C. VM-level scheduling*

The large energy consumption of high-performance computers forces researchers to explore ways for reducing this energy consumption. In this work, authors present a new scheduling algorithm based on the technical DVFS (Dynamic Voltage Frequency Scaling) to reduce the energy consumption of a single virtualized cluster. The aim of this schedule is the allocation of virtual machines in a cluster for a more efficient computing where virtual machines are dynamically provided to run Cluster jobs. The main idea is to reduce the frequency of the Cluster clock as low as possible to be compatible with the demands of virtual machines **[10]**.

In **[11]**, the authors developed a two-tier control system to optimize the management of data center resources, local controllers at the application level determine the amount of resources needed for the application to ensure its performance. A global controller at data center level determines the placement of VMs and resources allocation. It allocates all virtual machines at the same time, while trying to find the optimal allocation in accordance with the objectives and constraints.

A VMs placement policy was proposed by using an improved genetic algorithm with a fuzzy multi-objective evaluation to simultaneously minimize the total waste of resources, energy consumption and the cost of heat dissipation to combine different objectives. In this algorithm, the best solution is the one that belongs more to each fuzzy set provided with an aim.

**TABLE I** presents a comparison of different works stated above.

## V. SYNTHESIS

Several heuristics and meta-heuristics are studied to treat the scheduling problem; the latter is seen as a combinatorial optimization problem, where it is impossible to find the optimal global solution using algorithms or simple rules.

Depending on the level of scheduling, parameters are taken into consideration, such as: response time, cost, profit, availability, power consumption, etc to increase client satisfaction and in the same time to improve the use of different resources (**Table II** presents the metrics considered in the algorithms presented above). Most of these studies focused on aggregation of objectives, and on optimizing two metrics of QoS, often execution time and cost, and do not take into account other metrics as well as precedence constraints between tasks.

## VI. CONCLUSION

This work presents an overview about the different works dealt with scheduling levels (Service level, task level, virtual machine level) in the Cloud Computing. At each scheduling level, we presented different scheduling algorithms of existing tasks (workflow or independent tasks) with their service quality parameters that must meet users' requirements as well as it should improve the use of resources so that it can satisfy the user and the resource provider.

On our future works, we intends to direct towards the suggestion of formal approach for the conception of scheduling algorithms in the cloud computing.

We will work with Petri networks, specifically the Evolutionary Petri Net (EPN) which is based on a wide class of Petri nets called Resizable Petri nets, and two genetic operators (crossover and mutation) that provide a basis for the development of a robust evolutionary algorithm for optimization. To optimize some metric of service quality and specify, simulate, verify other performance measures (execution time, cost, number of satisfied customers, average waiting time, number of customers in the queue, wait ... .etc.).

**TABLE I:** DIFFERENT LEVELS SCHEDULING ALGORITHMS IN CLOUD COMPUTING

| N° | level | Reference | Algorithm/technique | Scheduling parameters | Advantages / results | Limits |
|---|---|---|---|---|---|---|
| 1 | Service-level scheduling | Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers **[2]** | Based on integer programming formulations | - Cost<br>- Throughput<br>- Execution time | Reduced cost | The approach is static: Static demand of virtual resources. |
| 2 | Service-level scheduling | Job scheduling model for cloud computing based on multi-Objective genetic algorithm **[3]** | Multi-Objective Genetic Algorithm (MO-GA) | - Energy consumption<br>- profit | Optimizes energy consumption and profit of provider | Does not take into account the precedence constraint. |
| 3 | Service-level scheduling | Scheduling strategies for optimal service deployment across multiple clouds**[4]** | - Cost optimization policy<br>- Performance optimization policy | - Cost<br>- Performance | The price is dynamic; several types of instance take into consideration. | Each criterion has an effect on the other; there is no relationship between the price charged and the performance of provider. |
| 4 | Service-level scheduling / Task-level scheduling | A market-oriented hierarchical scheduling strategy in cloud workflow systems **[5]** | Random algorithm<br>three meta-heuristic (GA, ACO, PSO) | - CPU time<br>- Makespan<br>- Cost | A minimum execution time, an effective hierarchical ordering strategy. | There is no relationship between the price charged and the performance of provider. |
| 5 | Task-level scheduling | A pareto-based genetic algorithm for optimized assignment of VM requests on a cloud brokering environment **[6]** | Multi-Objective Genetic Algorithm for Cloud Brokering (MOGA-CB) | - Response time<br>- Cost of VM instances | Relationship between the invoice price and performance | Only two QoS criteria taken into consideration. |
| 6 | Task-level scheduling | Optimal multi-constraints allocation of workflows in resources of cloud computing environment **[7]** | NSGA-II (non dominated sorting genetic algorithm) | - Makespan<br>- Cost<br>- Reliability<br>- Availability | Takes into account constraints of QoS, strong constraints and precedence constraints among tasks. | An important cost calculation, difficulty to choose the final solution. |
| 7 | Task-level scheduling | A genetic algorithm approach to a cloud workflow scheduling problem with multi-QoS requirements**[8]** | Genetic algorithm (GA) | - Makespan<br>- Cost<br>- Reliability<br>- Availability | Several metrics of QoS optimized, takes into account precedence constraints among tasks | - Does not take into account the constraints on metrics and strong constraints.<br>- Targets aggregation |
| 8 | Task-level scheduling | A genetic algorithm for multi-objective optimization in workflow scheduling with hard constraints **[9]** | Genetic algorithm based on viral infection (GA*) | - Makespan<br>- Cost<br>- Reliability<br>- Availability<br>- Budget<br>- Deadlines | Takes into account constraints of QoS, strong constraints and precedence constraints among tasks. | Provide a single solution for users (Targets aggregation). |
| 9 | VM-level scheduling | Power-aware scheduling of virtual machines in DVFS-enabled clusters **[10]** | Based on DVFS (dynamic voltage frequency scaling) | - Energy consumption | Effective to reduce energy consumption in a DVFS-enabled cluster | -Treats a single cluster.<br>-The hardware configuration hypotheses. |
| 10 | VM-level scheduling | Multi-objective virtual machine placement in virtualized data center environments**[11]** | multi-objectives fuzzy genetic algorithm | - Total resource wastage<br>- Power consumption<br>- Thermal dissipation cost | High performance, a combination of conflicting objectives. | Lack of diversity solution (one solution supplied), the placement of VMs is not dynamic. |

**TABLE II:** METRICS USED IN DIFFERENT ALGORITHMS

| Reference | Algorithm/ Technique | Makespan | Cost | Reliability | Availability | Energy | Constraint of QoS | Precedence Constraint | Meta-Heuristic | Pareto |
|---|---|---|---|---|---|---|---|---|---|---|
| [2] | Based on integer programming formulations | YES | YES | NO | NO | NO | YES | NO | NO | NO |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| [3] | MO-GA | NO | NO | NO | NO | YES | YES | NO | YES | YES |
| [4] | -Cost optimization policy<br>-Performance optimization policy | NO | YES | NO | NO | NO | YES | NO | NO | NO |
| [5] | -Random algorithm<br>-Three Meta-heuristic | YES | YES | NO | NO | NO | YES | YES | YES | NO |
| [6] | MOGA-CB | YES | YES | NO | NO | NO | NO | NO | YES | YES |
| [7] | NSGA-II | YES | YES | YES | YES | NO | YES | YES | YES | YES |
| [8] | GA | YES | YES | YES | YES | NO | NO | YES | YES | NO |
| [9] | GA* | YES | YES | YES | YES | NO | YES | YES | YES | NO |
| [10] | DVFS | NO | NO | NO | NO | YES | NO | NO | NO | NO |
| [11] | A improved Genetic algorithm with fuzzy multi-objective | NO | NO | NO | NO | YES | NO | NO | YES | YES |

## VII. References


[1] Y. Kessaci. “Multi- Criteria Scheduling in Cloud,” Doctorate thesis, Doctoral school of sciences for engineer, University Lille 1 Nord-de-France, Lille fundamental computing laboratory (UMR CNRS 8022) Inria Lille research center North of Europe, November 28th, 2013.

[2] J. Tordsson, R. S. Montero, R. Moreno-Vozmediano, and I. M. Llorente, “Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers,” Future Generation Computer Systems, vol. 28, no. 2, pp. 358 – 367, 2012. http://dx.doi.org/10.1016/j.future.2011.07.003

[3] Liu, J., Luo, X.G., Zhang, X.M., Zhang, F. and Li, B.N. “Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm,” IJCSI International Journal of Computer Science Issues, 10(2013)134-139.

[4] J. Luis Lucas-Simarro, R. Moreno-Vozmediano, R.S. Montero and I.M. Llorente. “Scheduling strategies for optimal service deployment across multiple clouds,” Future Generation Computer Systems, Vol. 29, Issue 6, pages 1431-1441, 2013. (Cited on pages 26 and 31).

[5] Z. Wu, X. Liu, Z. Ni, D. Yuan and Y. Yang. “A market-oriented hierarchical scheduling strategy in cloud workflow systems,” The Journal of Supercomputing, vol. 63, no. 1, pages 256–293, 2013. (Cited on pages 1, 2, 27 and 31).

[6] Y. Kessaci, N. Melab and E-G.Talbi, “A Pareto-based Genetic Algorithm for Optimized assignment of VM Requests on a Cloud Brokering Environment,” International IEEE Congress on Evolutionary Computation, June 20-23 Cancun, Mexico 2013. http://dx.doi.org/10.1109/cec.2013.6557869

[7] Yassa, S. “Optimal multi-constraints allocation of workflows in resources of cloud computing environment,” Doctorate thesis, University of Cergy-Pontoise Doctoral school of Sciences and engineering, July 10th, 2014 (Cited on pages 87-90).

[8] Yassa, S., Chelouah, R., Kadima, H., Granado, B. “A Genetic Algorithm Approach to a Cloud Workflow Scheduling Problem with Multi-QoS Requirements,” 26th European Conference on Operational Research, Rome 1-4 July, 2013.

[9] Yassa, S, Sublime, J., Chelouah, R., Kadima, H, Jo, G-S., Granado, B. “A Genetic Algorithm for Multi-Objective Optimization in Workflow Scheduling with Hard Constraints,” International Journal of Meta-heuristics, 2(2013)415 – 433.

[10] G. Von Laszewski, Lizhe Wang, Andrew J. Younge and Xi He. “Power-Aware Scheduling of Virtual Machines in DVFS-enabled Clusters,” In Cluster Computing and Workshops (CLUSTER), IEEE International Conference on, pages 1–10, New Orleans, LA, 2009. (Cited on pages 2, 29 and 31).

[11] J. Xu and J.A.B Fortes. “Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments,” In IEEE/ACM Int Conference Green Computing and Communications (GreenCom) & Int Conference on Cyber, Physical and Social Computing (CPSCom), pages 179–188, 2010. (Cited on pages 2, 16, 30 and 31). http://dx.doi.org/10.1109/greencom-cpscom.2010.137