Data-Intensive Scientific Workflow Scheduling Based on Genetic Algorithm in Cloud Computing

Siham.Kouidri[1], Chaimaa. Kouidri[2]
[1] Department of Computer Science, Faculty of Technolgy, University of Saida, Algeria
[4] Dept. of Computer Science, Faculty of Exact and Applied Sciences, University of Mustapha Istambouli, Mascara, Algeria
[1]skoudri2008@gmail.com;
[2] ckouidri2014@gmail.com

Abstract— Cloud Computing is increasingly recognized as a new way to use on-demand, computing, storage and network services in a transparent and efficient way. Cloud Computing environment consists of large customers requesting for cloud resources. Nowadays, task scheduling problem and data placement are the current research topic in cloud computing. In this work, a new technique for task scheduling and data placement are proposed based on genetic algorithm to fulfill a final goal such as minimizing total workflow response time. the scheduling of scientific workflows is considered to be an NP-complete problem, i.e. a problem not solvable within polynomial time with current resources The performance of this proposed algorithm has been evaluated using CloudSim toolkit, Simulation results show the effectiveness of the proposed algorithm in comparison with well-known algorithms such as genetic algorithm with Random data placement.

Key Words — Cloud Computing, Workflow Scientific, Scheduling, Virtual Machine , NP-Complet Problem, Data Placement, Genetic Algorithm.

1 Introduction

Nowadays, many scientists and researchers are moving towards Cloud computing for achieving high performance. This paradigm brings a new operational model where resources are managed by specialized data centers and rented only under demand and for the period of time they need to be used, is becoming very attractive for companies and institutions.

Scientific workflows are a popular way of modeling applications to be executed in parallel or distributed systems like Clouds. Once the data-intensive scientific workflow is composed, one of the most challenging research topics is how to schedule the different tasks onto the available resources. Traditionally, in parallel and distributed systems, workflow scheduling has been targeted to optimize the performance, measured in terms of the makespan or time of completing all tasks. This problem has been shown to be NP-complete which is difficult to obtain exact optimal solution and is suitable for using intelligent optimization algorithms to approximate the optimal solution. [1].

The advent of Cloud computing as a new model of service provisioning in distributed systems encourages researchers to investigate its benefits and drawbacks on executing scientific applications such as workflows. One of the most challenging problems in Clouds is workflow scheduling [2].
Scheduling is a process that maps and manages execution of inter-dependent tasks on distributed resources. Main motive of the scheduling is to allocate the suitable resources to the workflow tasks so that the execution of the workflow tasks completed within the deadline given by the customer. Suitable scheduling approach can have significant impact on the performance of the cloud computing.
The aim of this article is to present the scheduling and data placement problem as an optimization problem in the context of cloud-type platforms. We are going to propose a solution to minimize the runtime of the data-intensive scientific workflow. This solution can be in the form of an optimization algorithm, whether meta-heuristic (genetic algorithm), we propose to integrating task scheduling and data placement into one frame work for the sole goal of minimizing the total scientific workflow execution time of the cloud computing.

The rest of this paper is structured as follows: Section 2 explains some related works. Section 3 introduces our proposed approach. Section4 evaluates the performance of simulation experiments using CloudSim. Conclusion and future works are presented in section 4.

2 Related Works

Several works have been proposed to solve the scheduling problem in cloud computing.
Authors in [7] presents GHPSO algorithm to achieve the scheduling goals, this paper greatly improves the solution quality, so it can be used as an effective way to solve the cost minimization problem in cloud computing.
In [8] authors propose scheduling based on particle swarm optimization algorithm in cloud computing
The FCFS algorithm is considered as an easy method in scheduling algorithms, where processes are ordered by arrival time and submitted to the virtual machine [9].
In this paper [10], authors propose a scheduling algorithm integrated with task grouping, priority-aware and SJF (shortest-job-first) to reduce the waiting time and makespan, as well as to maximize resource utilization.
The authors of [11] propose the Max-min algorithm, the Max-min improvement is based on execution time instead of completion time as a basis for selection.
[13] propose the Round Robin (RR) algorithm focused on equity. RR uses the ring as a queue to store jobs. Each task in a queue has the same execution time and it will be executed in turn. If a job cannot be completed during its turn, it will be stored in the queue while waiting for the next turn. In addition to this, you need to know more about it.
A workflow tasks scheduling algorithm based on genetic algorithm in cloud computing is proposed in [14], In this algorithm, each task is assigned priority by an top-down leveling method for reducing the execution cost of workflow tasks scheduling, all workflow tasks

are divided into the different levels, which can promote the parallel execution of workflow tasks.

Authors in [15] present three scheduling algorithms are discussed such as LCFP, Min-Min, Max-Min and Genetic Algorithm. The idea of GA comes from natural selection which consists of population generation, selection, and mutation. This approach reduces the makespan of algorithm by using enhanced Max Min for initializing the population in GA.

[16] proposed a meta-heuristic based scheduling, which minimizes execution time and execution cost as well. An improved genetic algorithm is developed by merging two existing scheduling algorithms for scheduling tasks taking into consideration their computational complexity and computing capacity of processing elements.

In [17] A heuristic method to schedule bag-of-tasks (tasks with short execution time and no dependencies) in a cloud is presented. The number of virtual machines to execute all the tasks within the budget is minimum.

Tab 1: A comparative study of tasks scheduling algorithm.

References	Algorithm used	Advantages	Disadvantages
[7]	particle swarm optimization	reduce the task average running time, and raises the rate availability of resources.	lay stress on resource load balance of dynamic task scheduling in cloud computing, and consider other resource in cloud computing, for example, how internet quality of service and data distribution influence task scheduling
[8]	Hybrid particle swarm optimization	minimize costs within a given execution time.	multi-target workflow scheduling in cloud computing
[9]	FCFS	minimum execution time or minimum and prioritized cost	allocating resources to customers will be one of the main concerns of resource providers
[10]	Priority-aware and SJF	reduce the waiting time and makespan, as well as to maximize resource utilization.	to gain more scientific and realistic results, he will collect real data sets and adopt the real cloud-based environment. It is also important to evaluate the proposed solution in a physical real run-time environment to observe its true efficacy for cloud-based systems.
[11]	Max-min	override the appropriate task allocation card on resources	reduces overall time and balances the load between resources
[13]	Round Robin	improve resource usage and response time	The drawback of RR is that the largest job takes enough time for completion.
[14]	Genetic Algorithm	minimize cost, minimize execution time.	They did not take into account data transfer for data-intensive applications
[15]	LCFP, Min-Min, Max-Min and Genetic Algorithm.	minimize execution time.	They did not take into account data transfer for data-intensive applications
[16]	Genetic Algorithm	minimize execution time.	They did not take into account data transfer for data-intensive applications

3 Problem Description

Scientific workflows are a set of computer tasks organized to perform a composite (complex) mission in different fields such as climate modeling, genome sequencing, seismic analysis and oil exploration. Scientific workflows include hundreds of interconnected computational tasks according to different dependency models. Workflow tasks typically require large input data files and / or perform an extraordinary number of instructions. These factors provide a high level of productivity for their activities. As a result, the optimal scheduling process becomes a complicated problem. In the distributed execution paradigm, the tasks of a scientific application are assigned to different data centers for execution. When a task requires data processing, displacement and availability becomes a challenge and therefore a very long response time. One of the most difficult issues in workflow planning is optimizing the movement of data and therefore the cost of running the workflow. Meta-heuristic scheduling schemes give the best result compared to heuristic algorithms. The genetic algorithm is one of the best meta-heuristic algorithms. A genetic algorithm (GA) is a research algorithm based on the principle of evolution and natural genetics. It combines the exploitation of past results with the exploration of new areas of the research space [18].

4 Description of Proposed Work

Scientific applications are generally represented by workflows. Data-intensive workflows describe a number of tasks and the data dependencies between the tasks of an application (fig1). It is advantageous to use the cloud to run complex science applications because of its large data and virtual machine resource pools. One of the most difficult problems in workflows scheduling are optimizing the cost of running the workflow and the cost of loading data. Figure 2 gives the steps of GA. A genetic algorithm (GA) is an algorithm based on the principle of evolution and natural genetics.

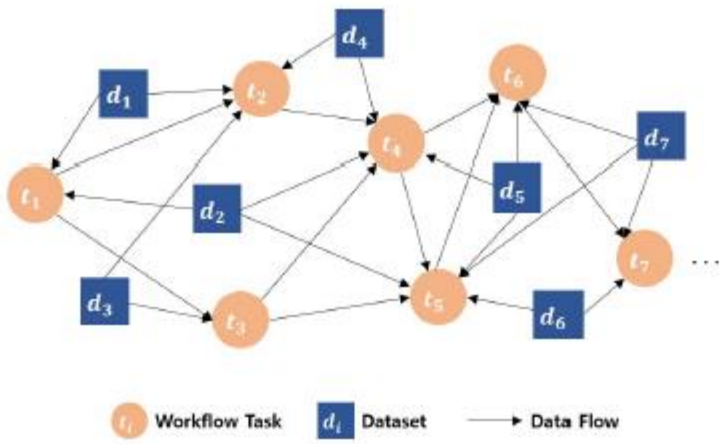

Fig1: Example of scientific workflow

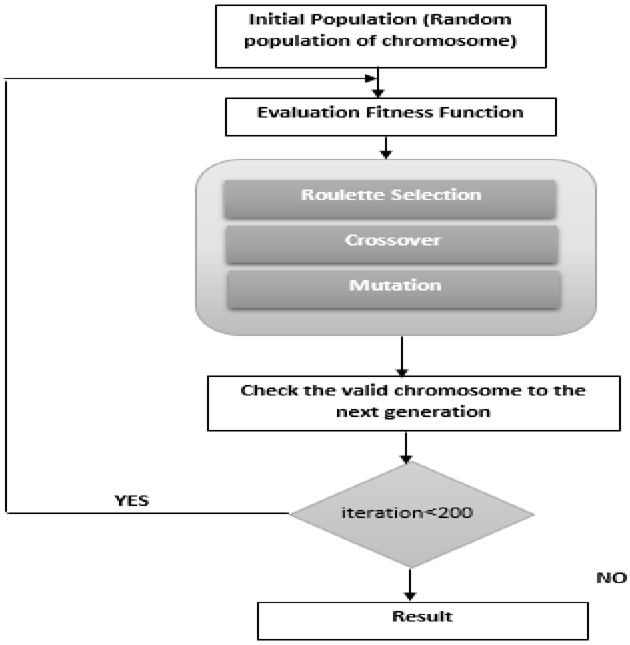

Fig.2 Steps in Genetic Algorithm

4.1 Chromosome Representation

The chromosome represents a complete sequencing of the workflow where each gene represents either a task and the virtual machine required to perform it or the location of a piece of data on it (VM). Each chromosome is a chain of genes encoding a specific solution. Figure 3 describes the modeling of a chromosome for a scientific application consisting of five tasks and six data that will be allocated on three virtual machines

Workflow elements	ID
Task0	0
Task1	1
Task2	2
Task3	3
Task4	4
Data1	5
Data2	6
Data3	7
Data4	8
Data5	9
Data6	10

Task and data assignment vector

0	1	2	3	4	5	6	7	8	9	10
Vm3	Vm1	Vm1	Vm2	Vm2	Vm3	Vm1	Vm3	Vm2	Vm3	Vm3

Fig.3 Chromosome description

4.2 Initial Population

We have merged Random solutions to generate the initial population. The success of each contribution to research is based on the problem modelling approach and on how to adapt the meta-heuristic to the scheduling problem. which encode scheduling solutions to an optimization problem, evolves toward better solutions.

4.3 Fitness Function

The objective of the fitness function is to evaluate each chromosome in the population. In case of minimization problem, the best fit chromosome will have the lowest numeric value for the objective function.

We provide details on the objective function in the following:

Min Fit=∑(T_execi+T_dataAccessi+T_releasei)

Fit : the fitness function it represent the completion time of scientific workflow.

1- T_execi: estimated execution time is measured based on the CPU processing capacity of the target virtual machine and the size of a task. It represents the processing time of a task on the VMj

$$T\_exec_i = \frac{Length\_T_i}{VM\_capacity_j * PE\_Number\_VM_j}$$

Where:

$Length_{T_i}$: is expressed as the number of instructions to be executed.

$VM_{capacity_j}$: is the speed of a processor, is expressed in MIPS (Million Instructions Per Second);

$PE_{NumberVM_j}$: represents the number of processors of virtual machine VMj.

2- T_dataAccessi: represents the estimate data access time to the data i.e. the processing time of local and remote data based on the following two formulas

$$TDataAccess_i = \sum_{k=1}^{n} \frac{LocalDatasetSize_k}{DiskTransfertCapacityVM_j} + TRemoteDataAccess_j$$

$$TremoteDataAccess_i = (\sum_{p=1}^{L} \frac{RemoteDatasetSize_p}{BPvm_j} + \frac{RemoteDatasetSize_p}{DiskTransertCapacityVM_j})$$

The principle of this estimate is to measure the processing time for data stored locally, and the time of missing data movement for the task and their treatment on vmj resource.

3- $T\_release_i$: We define the resource release time as the waiting time to release the CPU resources and the data set necessary for the execution of the task as shown in the following formula:

$$T - release = T - release - processorVM_j + T - release - Dataset$$

4.4 Selection

In the selection phase the Roulette wheel method is used. The roulette wheel selection operator minimizes the fitness function. The objective of the selection operation is to make duplicate copies of the good solution and eliminate bad solutions in a population, while maintaining the population size.

4.5 Crossover:

Selecting individuals from the parental generation, new individuals are obtained. There are so many crossover operators which can be used to get the better results. In our case we have chosen the one-point crossover.

4.6 Mutation:

There are several mutation operators based on the permutation-based representation of the schedule like Move, Swap, Move & Swap and Rebalancing. We chose simple Swap. The need for mutation operation is to keep diversity in the population.

5 Results and Discussion

The proposed Genetic Algorithm for Data-Intensive Scientific Workflow **GA-DISW** is developed using Cloudsim tools. it is an open source,scalable and low simulation overhead simulator [19]

In order to study the behavior of our proposal and analyze its results obtained by simulation, we compare them with other strategies. Several series of simulations have been launched. Simulation parameters are presented in Table 2.

Tab 2: GA Parameters.

Parameter	**Value**
Number of Cloudlets	[200,1000]
Number of VM	[10,80]
Number of Iterations	200
Population Size	100
Crossover Type	One-Point Crossover
Crossover Probability	0.5
Mutation Probability	0.5
Mutation Type	Simple swap
Termination condition	Number of iteration

We run each workflow instance through 4 simulation strategies:
Space shared: In space shared mode allocation of resources are done till the completion of task (i.e. no preemption of resources)[20];
Time Shared: in time shared mode preemption is allowed for resources continuously till the task completion[20].
GA: This strategy based on GA to submit the cloudlet in VM [14]. This approach does not take into consideration the movement of data.
GA-DISW: This simulation shows the performance of our approach.

1. **Response time :**

Tab 3: Response Time(ms) vs Number of Cloudlet.

N° Cloudlet (Tasks)	Algorithms			
	Space Shared	Time Shared	GA	**GA-DISW**
200	30500	12305	10526	**80365**
400	40035	18000	17325	**15230**
800	64500	31008	29652	**25303**
1000	80085	40000	35023	**33540**

In this first series of experiments, we measured the response time. For this simulation, we executed the simulation with the four approaches space shared, time shared, GA and GADISW. Simulation results were performed with the parameters described in the Table 2.

We notice through the table 3 that using GA-DISW, the response time of cloudlets decreases considerably in relation to space shared, time shared and GA

2. **Number of displacement :**

In this second experiment, we measured the number of displacements of each data for the same workflow. For this simulation, we executed the simulation with the four approaches (space shared, time shared, GA and GA-DISW).

The following Tab. 4 shows the resulting. We note with different data values, using the investment strategy reduces the number of moving data between VM compared to GA with random assignment

Tab 4: N°r of Data vs Number of displacement.

N°Data	**Algorithm**	
	GA-Random Data Placement	**GA-DISW**
D1	30	25
D2	52	15
D3	10	09
D5	25	07
D6	18	12

6 Conclusion and Future Work

In this work, we realize a process of tasks scheduling based on genetic algorithm for data-intensive scientific workflow. Our goal is to reduce data movement and therefore improved response time of workflow.

Our strategy is tested in the CloudSim simulator and compared with Space Shared, Time Shared politics and GA with random data placement. The strategies can provide a better response time and a minimizing data displacement.

In future, we will increase the capabilities of the proposed model by enabling replication of data sets for workflow scheduling.

7 References

1. Juan J. Durillo, Hamid Mohammadi Fard, Radu Prodan Institute of Computer Science, University of Innsbruck Innsbruck, Austria, document text " MOHEFT: A Multi-Objective List-based Method for Workflow Scheduling" 2012.

2. Saeid Abrishami a, ∗, Mahmoud Naghibzadeha, Dick H.J. Epemab -Article pdf-" Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds" 23 Mai 2012.

3. Dr. Maitreyee Dutta, Jailalita , Document text «A Review on Scheduling Algorithms for Workflow Application in Cloud Computing.

4. Jia Yu and Rajkumar Buyya "A Taxonomy of Workflow Management Systems for Grid Computing".

5. Joseph C. JACOB et al. "Montage: A Grid Portal and Software Toolkit for Science-Grade Astronomical Image Mosaicking". In: IJCSE 4.2 (2009), p. 73-87.

6. Sid Ahmed MAKHLOUF, « Gestion des ressources dans les systèmes à grande échelle Application aux environnements en Cloud ». thesis, juin 2019.

7. Optimization of FCFS Based Resource Provisioning Algorithm for Cloud Computing 2013.

8. Enhanced Max-min Task Scheduling Algorithm in Cloud Computing 2013.

9. Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing 2013.
10. Enhanced Particle Swarm Optimization For Task Scheduling In Cloud Computing Environments 2015.
11. Task Scheduling Using Best-Level-Job-First on Private Cloud Computing 2016.

12. An Improved SJF Scheduling Algorithm in Cloud Computing Environment 2016.

13. Dr. Amit Agarwal, Saloni Jain "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment" International Journal of Computer Trends and Technology (IJCTT), volume 9, number 7, Mar 2014.
14. Yang Cui, Zhang Xiaoqing "Workflow Tasks Scheduling Optimization Based on Genetic Algorithm in Clouds" 2018 the 3rd IEEE International Conference on Cloud Computing and Big Data Analysis.2018.

15. Shekhar Singh, Mala Kalra " Task Scheduling Optimization Of Independent Tasks In Cloud Computing Using Enhanced Genetic Algorithm". International Journal of Application or Innovation in Engineering & Management (IJAIEM). Volume 3, Issue 7, July 2014.
16. S. Kaur, and A. Verma. "An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment." International Journal of Information Technology and Computer Science (IJITCS) 4, no. 10, 2012.
17. Shaminder Kaur, Amandeep Verma .« An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment". Information Technology and Computer Science, 2012, 10, 74-79
18. Zomaya, A.Y., Ward, C. and Macey, B., Genetic scheduling for parallel processor systems: comparative studies and performance issues. Parallel and Distributed Systems, IEEE Transactions on, Vol. 10, Issur 8, pp.795-812, 1999.
19. Calheiros, R., Ranjan, R., Beloglazov, A., De Rose, C. & Buyya, R. (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience journal, 41(1), 23-50.
20. Ram Pratap Taskeen Zaidi: "Comparative Study of Task Scheduling Algorithms through Cloudsim", 7th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), August 29-31, 2018,