Energy-aware workflow scheduling in a Cloud Computing environment using Non-dominated sorting genetic algorithm (NSGA-II)

Tounes Sellami 1, @ , Khaled Sellami 1,@
1 : Département d'Informatique, université de Bejaia

1 Introduction

In this work, we are interested in the workflow scheduling in a Cloud Computing environment. Workflow technologies are integrated through resource provisioning technologies in Cloud so as to determine the proper quantity of resources required for any execution of workflows . This is done to reduce the financial cost from user perspectives and to fully exploit the resource utilization from cloud providers. In order to effectively schedule workflow tasks on cloud environments, a Workflow Managements Systems (WMS) uses an efficient scheduling method for allocating the tasks in a workflow to appropriate cloud resources with the aim of satisfying user requirements, well known as Quality of Services (QoS) constraints.

As this problem is a NP-complete and depends on the problem size, abundant scheduling algorithms have been proposed to resolve challenging problems quicker than metaheuristic one [1]. In addition to the usual QoS constraints such as time and cost to solve this problem, energy consumption has turn into a key concern in the field of Cloud computing.

We have to implement workflow scheduling using Dynamic Provisioning Based on Demand (DPBD) algorithm. Minimizing makespan and energy consumption of the cloud service are the dual objectives, whereas completing the tasks in a sequential manner and the priority of the tasks are the design constraints. The Pareto optimal solutions of the multi-objective optimization problem are obtained using the non-dominated sorting genetic algorithm (NSGA-II).

2 Problem description

1) A collection of heterogeneous virtual machine $VM=\{VM_j,\ j=1,2,...,m\}$ which are connected through a communication link $Lp_j=\{l_{j,1},\ l_{j,2},\ ...,\ l_{j,k}\}$, $1 \leq k \leq m$; where $l_{j,i} \in R+$ is the effective bandwidth - rated by Mega bits per second (Mbps) - in the connection among processors p_j and p_i, with $l_{j,j} = 1$. Figure 1 shows a sample architecture of our considered multiprocessor architecture with three VM fully interconnected.

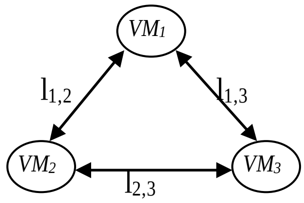

Fig. 1 Three heterogenous VM fully connected

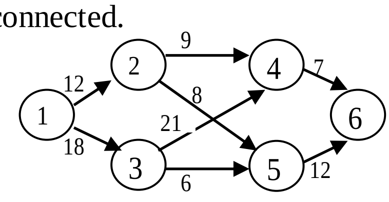

Fig. 2 Example of workflow application

For each VM, multiple levels of operational voltages (voltage scaling level-VSL) and frequencies are considered with Dynamic Voltage and Frequency Scaling enabled (DVFS-enabled) technique.

2) A collection of n dependent application tasks $V=\{v_i,\ i=1,2,...,n\}$ that have precedence constraints, represented in form of Directed Acyclic Graph (DAG), Figure 2 shows an example of workflow application with six tasks.

3 Dynamic Provisioning Based on Demand (DPBD) algorithm

DPBD algorithm is an online algorithm, which supplies the required number of resources based on the present requirement. It terminates some existing VMs, if there is a decrease in demand and also supplies additional VMs, if the demand is increased. The energy consumption of the cloud service is considerably reduced by terminating the idle VMs. Likewise, the makespan is controlled by supplying the additional VMs when the demand is increased. Thus the proposed algorithm helps to effectively minimize makespan and energy consumption of the cloud service. The step-by-step procedure of the proposed algorithm is shown in Algorithm 1.

Algorithm 1: Dynamic Provisioning Based on Demand (DPBD) algorithm

Let, P_{max}-maximum power consumption in kW; T - maximum makespan in hour; c - power consumption

of the resource per hour (kW/hr);

1 . **Start algorithm 1**

2. find VM_r (initial number of virtual machines required)
3. **while** maximum makespan is not reached **do**
4. **while** the allotted power consumption is not met **do**
5. **if** set of workflows *W* is not empty **then**
6. assign the priority for each task in *W*
7. select the idle VMs randomly
8. assign the tasks to the idle VMs based on their priority
9. get VM_f number of virtual machines completed their tasks
10. calculate VM_j - number of virtual machines to be terminated
11. update the number of required virtual machines VM_r **end if end While end while end algorithm**

4 Non-Dominated Sorting Genetic Algorithm (NSGA II)

The Non-dominated Sorting Genetic Algorithm (NSGA-II), recommended by Deb et al. [2], is considered in this paper to find the Pareto optimal solutions of the multi-objective optimization problem. In the step-by step procedure of NSGA-II the initial population is randomly generated and the designs are sorted into each front based on the principle of non-domination. Both objectives are maintained using a fitness assignment System, which favours non-dominated solutions. The front, which is ranked as one in the present population, is absolutely a non-dominant one. Similarly, the solutions existing in the second front are only dominated by the solutions existing in the first front and the assigning ranks to the fronts happen so on.

References

[1] Khaled Sellami, PF Tiako, Lynda Sellami, RabahKassa. “Energy Efficient Workflow Scheduling of Cloud Services Using Chaotic Particle Swarm Optimization”. 2020 IEEE Green Technologies Conference (GreenTech), 74-79

[2] Deb, K.: 'Multi-objective optimization using evolutionary algorithms', John Wiley & Sons, 2001