SAFC: Scheduling and Allocation Framework for Containers in a Cloud Environment

Tarek Menouer and Christophe Cérin

Université Paris 13, LIPN UMR CNRS 7030, 99 av. J.B Clément, 93430 Villetaneuse, France
{tarek.menouer,christophe.cerin}@lipn.univ-paris13.fr

1 Context of our work

In this paper, we introduce our research and the framework called Scheduling and Allocation Framework for Containers (SAFC). SAFC is proposed for cloud environments. It is also relying on an economic model to deal with the traditional Infrastructure-as-a-Service (IaaS) container-based virtualization technology. The general problem that we address is the following. Given a set of user containers, find an allocation on machines in order to maximize the user expectation and to minimize the number of resources. We derive solutions for a context of heterogeneous, dynamic and uncertain environments.

The SAFC economic model is proposed with two SLA classes: a qualitative one and a quantitative one. The qualitative SLA class may represent the satisfaction time of the container, i.e. the waiting time between the submission until the beginning of the execution of the container. Moreover, the quantitative SLA class may represent the number of required resources to execute a container. The choice of two SLAs classes is motivated by the fact that the majority of users like to have a solution with constraints related to the waiting time and the required resources. To adapt to the user needs, we propose three Quality of Service (QoS) for each SLA class: Premium (highest service), Advanced (medium service) and Best effort (lowest service).

The goal of the SAFC system is to optimize the scheduling of containers submitted online by users and to execute them as soon as possible. In our view of the problem for resources allocation, and compared to the existing frameworks, the number of required resources is not given in the container description submitted by the user. Furthermore, the number of resources allocated for each container is decided on the fly and it is comprised between hard and soft min/max bounds of cores. The hard min/max bounds of cores are set according to the selected service in the quantitative SLA class and the configuration of machines that form the cloud infrastructure. Moreover, the soft min/max bounds of cores which are inside the hard bounds, are set according to the load of parallel machines.

In a more abstract way, the core scheduling algorithm in SAFC is composed of three cascaded steps. For instance we can instantiate the framework as:

1. Container selection: It is based on a combination between the qualitative and the quantitative classes, to select the first container in the queue that must be executed using PROMETHEE II (*Preference Ranking Organization METHod for Enrichment Evaluations*) algorithm [1];
2. Container allocation: It is based on the quantitative number of resources class, to set the number of resources that must be allocated to the selected container;

3. Container placement: Here we use a bin packing strategy. This step may also be based on a dedicated quantitative replicates class, to set the number of replicates of container (not explored in this paper);

2 Experimental using Grid5000 testbed

For the emulations, we have used Linux Container (LXC) inside the Grid5000 platform. For our experimental evaluation, we reserved an infrastructure composed of 1024 cores, distributed over 32 nodes. In this experimental evaluation, each container is submitted by one of the following three users: (i) User 1: Premium service for all SLAs classes; (ii) User 2: Advanced service for all SLAs classes; and (iii) User 3: Best effort service for all SLAs classes. Each container has a Sequential Life Time (SLT) set when the container is submitted and equal to 30 minutes. Then, according to the number of cores allocated for each container (N), the Parallel Life Time (PLT) is adapted. The PLT is computed as being $PLT = \frac{SLT}{N}$.

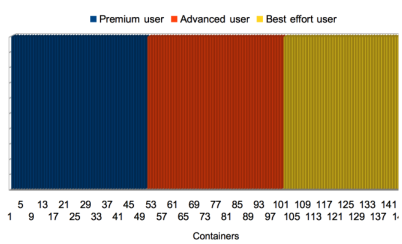

Fig. 1. Submission of 150 LXC containers at the same time

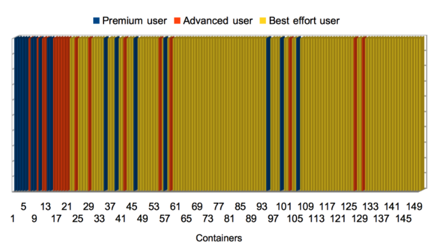

Fig. 2. Submission of 150 LXC container according to Prezi traces frequency

Figure 1 shows the order of execution of 150 containers submitted at the same time by 3 users, each user submits 50 containers. As a result, it is clear that SAFC starts by the execution, firstly containers submitted by the Premium user, then containers submitted by the Advanced user. Finally, SAFC executes containers submitted by the Best effort user. This result confirms that our framework respects the priorities of containers.

Figure 2 shows the order of execution of 150 containers submitted according to the real-world trace files of an international company called Prezi [2]. These traces represent Web oriented applications. The 150 containers are distributed as follows: (i) 17 submitted by user 1, (ii) 18 submitted by user 2 and (iii) 115 submitted by user 3. According to Figure 2, we note that there is an overlap between the execution of containers. This expected overlap is due to the fact that containers are submitted online.

References

[1] S.C.Deshmukh. Preference ranking organization method of enrichment evaluation (promethee). *International Journal of Engineering Science Invention*, 2:28–34, 2013.

[2] Prezi trace: `http://prezi.com/`.