Generating fluctuating workload for cloud elasticity simulation

Simon Bihel (Student) Computer Science Department, ENS Rennes

simon.bihel@ens-rennes.fr

***Abstract*—Cloud computing is a model that makes available infrastructures, platforms and software with a pay-as-you-go subscription. It aims to reduce the cost with a layer of visualization that allows virtual resources to be dynamically adjusted and occupied on-demand. The problem of using the minimal resources for the current demand/usage is still a research challenge that spans all layers and applications. This dynamic management of clouds is called cloud elasticity.**

***Index Terms*—Simulation; Cloud elasticity; Workload**

I. Introduction

II. Background

- What's a cloud
- Minimizing the cost (in all forms) is a research challenge, particularly for fluctuating usage
- Different types of scaling
- Present the survey that has categorized these works
- How are they categorized
- Simulation isn't used that much and it would give a lot
- Say that it gives good overview of what is needed to simulate
- Say that it covered everything

The global architecture is described in the Figure 1.

The problem of using the minimal resources for the current demand/usage is still a research challenge that spans all layers and applications. This dynamic management of clouds is called cloud elasticity.

[1] has categorized works on cloud elasticity and allows to see which elements of a cloud infrastructure, platform or application/software are impacted. As it is for now most research works are evaluated on real clouds It is interesting for a distributed systems simulator to search what is needed for simulating cloud elasticity. If it is shown that research works on cloud elasticity can be evaluated on a simulator they would benefit from cost reduction, re-runable experiments, trust in results...

In this survey proposals are categorized as follows. The scope is about what elements of a cloud the proposals work on. It can be the management of VMs, allocation of resources... Then there is the purpose of the proposal. Enhancing the *performances* (to meet the SLA), reducing the *energy consumption* footprint, being *available* when needed and reducing the overall *cost*. Another dimension is the decision making. This is what a proposal add to an existing cloud to pursue its purpose. In addition to the scope there is the elastic actions performed by the proposals. As the scope is about what elements of a cloud are concerned, the elastic action is about what is done to them. Then there is the provider dimension that tells if there is only one provider or multiple ones. At last there is the method used by the proposal to evaluate itself, through real cloud, simulation or emulation.

The survey gives a good overview on what elements of a cloud are manipulated to achieve cloud elasticity. No clue have been found to prove the opposite at the time of writing.

As the proposals are on reacting to variating usage, simulators need a way to express this fluctuating workload. We worked on elastic tasks that model tasks that are triggered regularly and with a usage that fluctuates over time.

III. State of the art

- Which elements have to be simulated and how do they work ?
- How workloads are generally modelized
- What others simulators have done
- What simulations for evaluation have been done
- What others evaluations do

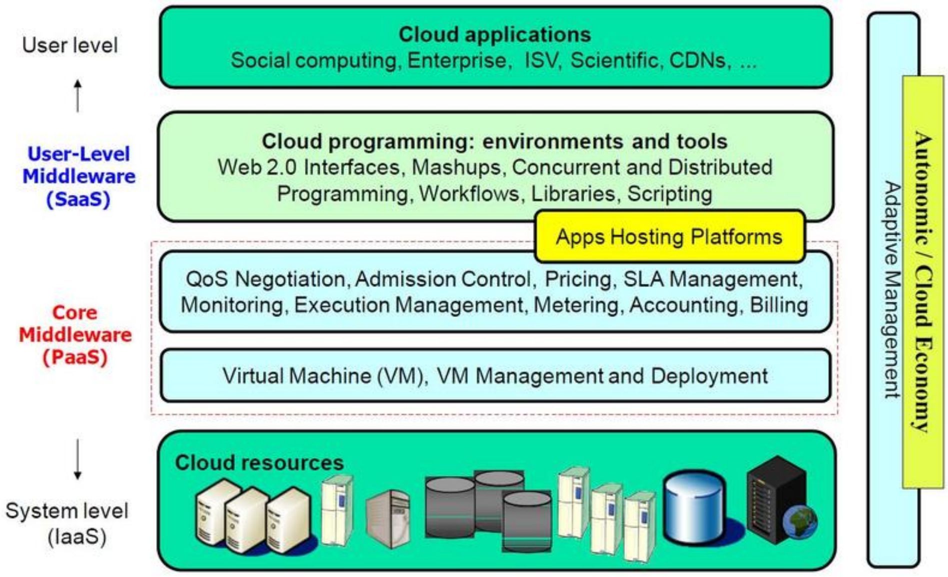
Based on the classification of the survey, a simulator should allow the manipulation of scopes, the evaluation of the different purposes, make possible the elastic actions and allow multiple providers.

At the moment no simulator article talks about dynamic workload. On the other hand in the code of DCsim [2] there was an interactive task and in the code of CloudSim [3] there was an host with dynamic workload.

// Go to contribution and go through each scope ?

For en-actor scopes, the point of elastic tasks is just to generate usage so they can chose an application type depending on what kind of usage they want (cpu, disk...). For the different kinds of application type, elastic tasks have the same mechanism it is just the inherent micro-task that is repeated

Figure 1: http://cloud-simulation-frameworks.wikispaces.asu.edu/



that changes. An elastic task will repeatedly execute an MSG task. Currently an MSG task can only simulate computing and message passing so only Multi-tier Applications can be simulated at the moment. Simulating disk and RAM usage would allow the simulation of databases, storage and thus generic application.

IV. Contribution

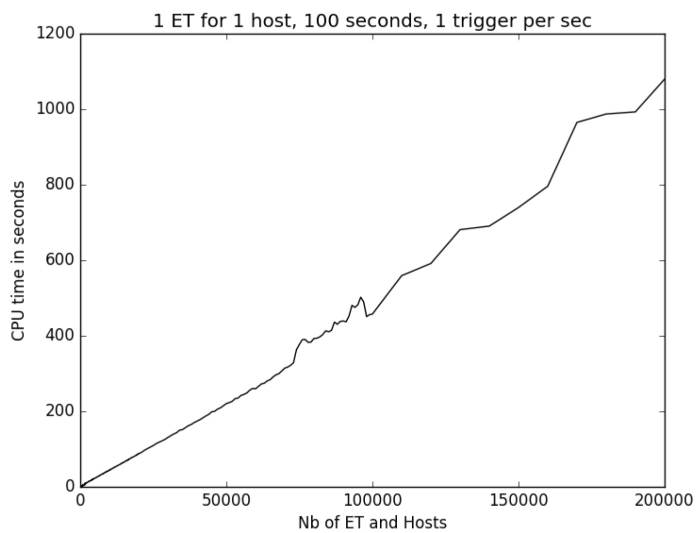
What can my contribution do that is in the survey?

- Only computational tasks for now, might be able to do storage/DBs tasks in the future. Generic ones won't be possible unless you can pass a task directly to an ET.
- Need to use the host of a task when executing to allow vertical scaling and need to manage multiple hosts to allow horizontal scaling.
- Multiple provider is possible but has to be coded.
- Purpose?

The work was done on SimGrid [4].

V. Evaluation

- How the experiments were made



A. Raw performances

B. Real traces

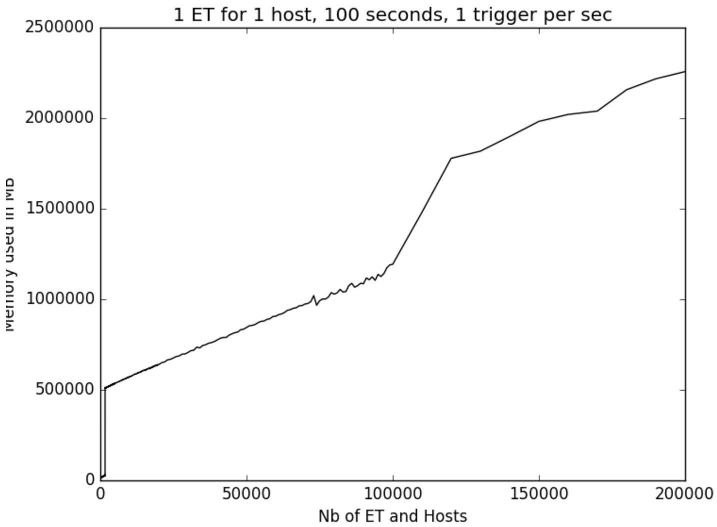
C. Functionalities

VI. Conclusion

References

[1] A. Naskos, A. Gounaris, and S. Sioutas, *Cloud Elasticity: A Survey*. Cham: Springer International Publishing, 2016, pp. 151–167. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-29919-8_12

[2] M. Tighe, G. Keller, J. Shamy, M. Bauer, and H. Lutfiyya, "Towards an improved data centre simulation with dcsim," in *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*. IEEE, 2013, pp. 364–372.



[3] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[4] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, "Versatile, scalable, and accurate simulation of distributed applications and platforms," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2899–2917, Jun. 2014. [Online]. Available: http://hal.inria.fr/hal-01017319