

Monitoring HPC applications on the cloud with Zabbix

William F. C. Tavares¹, Marcio Roberto Miranda Assis, Edson Borin¹

¹Instituto de Computação – Universidade Estadual de Campinas (Unicamp)

wtavares@lmcad.ic.unicamp.br, marcio.miranda@ic.unicamp.br, borin@unicamp.br

Abstract. *When a user overprovision cloud resources for his or her HPC application, it leads to underutilization and causes resources wastage. One solution is to monitor their usage. In this work, we propose a viable architecture to monitor HPC applications on cloud environment using the Zabbix tool. With this, we will be able to minimize cost and develop a methodology to monitor different kinds of HPC applications.*

1. Introduction

The high-performance computing (HPC) community has come to use cloud computing mainly due to accessibility and flexibility of a wide variety of computing resources. However, common users may face some issues when choosing a virtual machine for their application, which can cause overprovisioning and underutilization of resources. These two consequences may negatively affect the cost of the purchased service on the cloud. This pattern of behavior has become one of the main concerns for companies, estimating \$5.3 billion wasted on cloud resource overprovision [Chapel 2019].

Sometimes resource wastage is easy to detect, but in other cases, it becomes a particularly difficult task. One way to detect these effects is to monitor the utilization of instantiated resources so that solutions can be applied to minimize costs [Aceto et al. 2013]. However, monitoring resources is not a simple task since monitoring usage can affect application performance and user privacy.

This work proposes an architecture to monitor virtual machines and the utilization of HPC applications in the cloud using the Zabbix monitoring tool. This article is divided as follows. Section 2 presents the related works to monitor the cloud environment in the user's view. The challenges of monitoring HPC applications in the cloud are presented in Section 3, as well as the Zabbix monitoring tool. Finally, the architecture is presented in Section 4 and the conclusions and next steps in Section 5.

2. Related works

Several works address resource monitoring from the provider perspective [Al-Ayyoub et al. 2015, Gutierrez-Aguado et al. 2016], focusing on the detection of malicious users, optimizing process scheduling, and maximizing resource usage. However, recently, the user perspective has become an interesting area [Dhingra et al. 2012].

Brandt *et al.* [Brandt et al. 2009] seek to optimize scalability and guarantee reliability for HPC applications by making decisions dynamically in response to the resource utilization. A scheme for a distributed monitoring system is presented by Perez-Espinoza *et al.* [Perez-Espinoza et al. 2015], seeking to minimize bottlenecks

in centralized systems due to the high scalability of cloud. Irimie and Petcu *et al.* [Irimie and Petcu 2015] studied the cloud monitoring from a security perspective. In this work, we tend to focus on monitoring resource wastage using the utilization metrics.

3. Zabbix Monitoring Tool

The monitoring of virtual machines in cloud environments offers several challenges. Monitoring tools typically use a client-server model, in which a monitoring agent is installed on all hosts to be monitored [Fatema et al. 2014]. The first aspect to notice is related to the installation of this agent, which must be an automatic installation that takes little time and does not invade the users privacy during the configuration. The second aspect refers to the use of computing resources, the agent cannot compromise the application's performance, which may affect not only the amount of computation performed in each collection, but also the frequency of this collection. The third aspect is about security, the agent cannot allow remote access or open communication ports (it must operate on the push data mechanism, in which the agent sends its data to the server [Xu et al. 2017]).

Since we are seeking a solution that obtains utilization metrics and monitors the resource wastage in the cloud, there is a need for a triggering system (capable of notifying users about wastage) and secure communication. It is also desirable to make the monitored data persistent for posterior analysis, to be provider-independent, and to have a dashboard to visualize graphs and trends.

The Zabbix [LLC 2001] tool may offer a good solution for these constraints. Zabbix can monitor different kinds of resources, including virtual resources. It can operate on the push and/or pull data collection mechanisms [Xu et al. 2017]. It is possible to dynamically change the data collection frequency. It has a simple agent installation and configuration, a triggering system, and a storage system. It also features a dashboard with data presentation with graphs and easy configuration of triggers, users, and hosts.

However, it is necessary to extend this system so it can communicate with different providers, deal with the dynamics of the cloud (detecting started, stopped, and terminated instances), detect if a host does not have an agent installed, update instances macros and frequency collection, and create notifications automatically.

There are other monitoring tools, however, the most common tools are provider dependent [Amazon 2009] or commercial [ParkMyCloud 2015]. Some are specific for private clouds [De Chaves et al. 2011] or are generalized, going beyond monitoring [Sotomayor et al. 2009]. Others have basic features and, to refine the operation, they require a study of several modules to integrate it [Nagios 2018].

4. Proposed Architecture

Our architecture consists of the Zabbix agent installed on the hosts, the Zabbix server, and two additional components the Control System and the Audit Monitoring System, both implemented by the authors. The first one is responsible for keep consistency of the information on the Zabbix Server with the information from cloud provider; the sec-

ond detects hosts that are not sending metrics to the server. Figure 1 shows the proposed architecture.

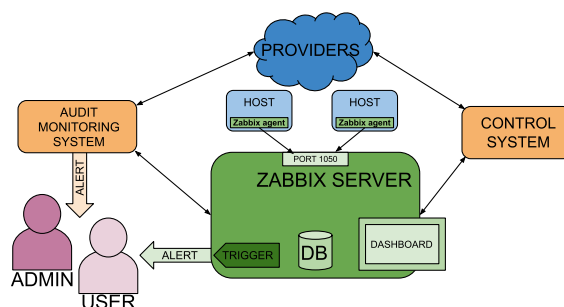


Figure 1. Proposed monitoring system architecture

The steps flow as follows. When a user starts a virtual machine instance on the cloud, it is necessary to run a script that installs and configures the Zabbix agent on the host. When the agent starts to run, it uses the auto-registration Zabbix function, which allows the agent to inform the server that it exists and register itself. Then, communication is initiated with push data collection mechanism. The Zabbix server is responsible for getting this data, storing it, showing it on the dashboard, and generating alerts based on it. When the host is terminated or stopped, the Zabbix agent stops sending data to the server.

The Control System component performs retrieves the list of started instances from the providers and then the list of registered hosts from Zabbix Server and perform the following operations: verify if the instance has been terminated and disable the monitoring of it from the server; update instances type and prices; associate registered hosts with the user that created it and the provider region.

The Audit Monitoring System is responsible for detecting hosts that have stopped sending data and for detecting new instances that have not been registered. Since the virtual machines are private and this architecture requires the user to execute the agent installation script, it is possible to prevent virtual machines from running without the agent installed. It is also possible to generate alerts for the administrator and for the user informing of the lack of host information on the server.

From monitoring virtual machines, there are native Zabbix metrics that we can explore in the context of HPC, such as CPU and memory utilization, disk read and write statistics, and network traffic statistics. Unfortunately, there are no GPU utilization metric in Zabbix, but it is possible to create new metrics using the UserParameter Zabbix feature.

5. Conclusions

Monitoring the utilization of resources is an essential task to minimize the wastage of them, which leads to optimizing the cost of running HPC applications in the cloud. There are some aspects to consider when monitoring a host with an agent installed. These aspects are simple and automatic installation, low computing resources usage, and do not make the environment vulnerable. The Zabbix monitoring tool enables these aspects to be covered and it also offers triggering, storage, and dashboard features.

In our solution to use Zabbix in the cloud, it is necessary to add two new components to allow the detection of started, stopped and terminated instances, update information from providers and perform actions based on alerts. However, it is required for the user to install the agent on the host by running a script.

The next steps are to implement the two proposed components and deploy them in a real cloud environment. It will be possible to detect if an application is not using all the cores properly, is not causing a high idle time for a node, is not being negatively affected by the network, or the GPU is being underutilized.

The authors would like to thank Petrobras, CNPq (140653/2017-1), and FAPESP (CCES 2013/08293-7) for their financial support and the Institute of Computing Multidisciplinary High Performance Computing Lab for its infrastructure and technical support.

References

- Aceto, G., Botta, A., de Donato, W., and Pescapè, A. (2013). Cloud monitoring: A survey. *Computer Networks*, 57(9).
- Al-Ayyoub, M., Jararweh, Y., Daraghme, M., and Althebyan, Q. (2015). Multi-agent based dynamic resource provisioning and monitoring for cloud computing systems infrastructure. *Cluster Computing*, 18(2).
- Amazon (2009). Amazon cloudwatch. aws.amazon.com/cloudwatch/.
- Brandt, J., Gentile, A., Mayo, J., Pebay, P., Roe, D., Thompson, D., and Wong, M. (2009). Resource monitoring and management with OVIS to enable HPC in cloud computing environments. In *IEEE International Symposium on Parallel Distributed Processing*.
- Chapel, J. (2019). Cloud waste to hit over \$14 billion in 2019. <https://devops.com/cloud-waste-to-hit-over-14-billion-in-2019/>.
- De Chaves, S. A., Uriarte, R. B., and Westphall, C. B. (2011). Toward an architecture for monitoring private clouds. *IEEE Communications Magazine*, 49(12).
- Dhingra, M., Lakshmi, J., and Nandy, S. K. (2012). Resource usage monitoring in clouds. In *13th International Conference on Grid Computing*.
- Fatema, K., Emeakaroha, V. C., Healy, P. D., Morrison, J. P., and Lynn, T. (2014). A survey of cloud monitoring tools: Taxonomy, capabilities and objectives. *Journal of Parallel and Distributed Computing*, 74(10).
- Gutierrez-Aguado, J., Alcaraz Calero, J. M., and Diaz Villanueva, W. (2016). IaaSMon: Monitoring architecture for public cloud computing data centers. *J. Grid Comput.*, 14.
- Irimie, B.-C. and Petcu, D. (2015). Scalable and fault tolerant monitoring of security parameters in the cloud. In *17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*.
- LLC, Z. (2001). Zabbix. www.zabbix.com/.
- Nagios (2018). *Nagios Core 4 Documentation*.
- ParkMyCloud (2015). Parkmycloud. parkmycloud.com/.
- Perez-Espinoza, J. A., Sosa-Sosa, V. J., and Gonzalez, J. L. (2015). Distribution and load balancing strategies in private cloud monitoring. In *12th International Conference on Electrical Engineering, Computing Science and Automatic Control*.
- Sotomayor, B., Montero, R., Llorente, I., and Foster, I. (2009). Virtual infrastructure management in private and hybrid clouds. *Internet Computing, IEEE*, 13.
- Xu, X., Chen, Y., and Alcaraz Calero, J. M. (2017). Distributed decentralized collaborative monitoring architecture for cloud infrastructures. *Cluster Computing*, 20(3).