Boletim da Sociedade Astronômica Brasileira, **30**, no. 1, 97-98


# Project and development of the architectural model for CRAAM Solar Virtual Observatory

Luis Felipe Backer[1], Israel Santos[1,2], & Leandro Pupo Natale[1]

[1] Faculdade de Computação e Informática Mackenzie
[2] Centro de Rádio Astronomia e Astrofísica Mackenzie - CRAAM

**Abstract.** Disruption of communication technologies have brought a new perspective to the way information is treated. With the adoption of new technologies it became essential the democratic access to knowledge for better results and the continuous and faster growth of the research community. In the astronomy field, the need of access to data and information impulsed labs to engineer software platforms to simplify access to collected data. In this article we will discuss about the challenges of the software development, maintenance and the adoption of new technologies to remodel and optimize the current architecture.

**Resumo.** A disrupção de tecnologias de comunicação trouxeram uma nova perspectiva no modo como abordamos o acesso a informação. Com a adoção de novas tecnologias passou a ser essencial a democratização do acesso ao conhecimento para trazer à sociedade de pesquisadores melhores resultados e garantir o crescimento contínuo da comunidade. No campo da astronomia, a necessidade de acesso a dados e informação impulsionou o desenvolvimento de plataformas de software para simplificar o acesso aos dados coletados. Neste artigo vamos discutir sobre os desafios do desenvolvimento e manutenção de software e adoção de novas tecnologias para remodelar e otimizar a arquitetura atual.

**Keywords.** Containers – Docker – Micro services – Software Architecture – SVO

## 1. Introduction

The era of communication technologies and information dissemination resulted in a new culture of sharing and democratization of knowledge, allowing users to access information from diversified sources. With this scenario it becomes essential, in the astronomy field, the development and adoption of solar virtual observatories (SVO) aiming the data sharing.

The Solar Virtual Observatory of Mackenzie University's Center of Radio, Astronomy and Astrophysics (CRAAM-SVO), initially created to structure and share collected data from the SST (Solar Submillimeter Telescope) (Kaufmann et al. 2001) and POEMAS (Polarization Emission of Millimeter) (Valio et al. 2013)telescopes, has been working with software computational resources to integrate with the International Virtual Observatory Alliance (IOVA), however the need for high availability of services and capacity to handle numerous request demand require an architectural review.

## 2. Solar Virtual Observatory Architecture

As explained by (Santos 2017) the current CRAAM-SVO architecture is composed of one application server that addresses requests from labs, partners and researchers. The development is based on Java Enterprise Edition architecture with frameworks such as JPA and Jena for a Virtuoso database integration, through SQL and RDF/XML, alongside with API's to generate FITS (Flexible Image Transport System) files for data sharing.

The software Architecture process and standards goes through constant upgrades aiming to better attend new services and to guarantee a flexible software architecture to be enhanced with the use of new disruptive technologies. The most popular approach in software development has been the use of containers united with the application development in micro services, allowing to execute in modules and gain high availability and scalability.

### 2.1. Microservices vs Service Oriented Architecture

The best choice for the SVO architecture development relies between Service Oriented Architectures (SOA), Microservices Architecture (MSA) and the analysis of which better fits with the software roadmap proposal.

(Erl et al. 2014) defines the SOA architecture model as a design paradigm model that comprises the design principles of the service orientation to enhance efficiency, productivity and agility guaranteeing a separation of concerns with low coupling services. With SOA services are agnostic, reusable and compose a service inventory where each service is responsible for a specific logic. The combination of SOA and Web Services leverage's the potential of re-usability by allowing access to the service logic through a communication framework.

The Microservices Architecture, according to (Bonerér 2016), is a system based on small, independent and connected services, this allow to divide the system into groups, removing the development complexity.

Both Microservices and SOA are verry simillar, (Bonerér 2016) says both share same goals such as decoupling, isolation, composition, integration and autonomous services, however microservices are an application of Reactive principles, this means that microservices are responsive, resilient, elastic (can adapt to larger workloads) and massage driven.

### 2.2. Containers vs Virtual Machines

According to (Dua et al. 2014), Virtual Machines (VM) are the virtualization of the operating system (OS) layer in which hypervisors allow for multiple guest OS to run in a single machine (Figure 1), which may result in performance issues due to the fact that VM's suffer from overhead in instructions translated to the host and it requires large amount of storage to maintain the OS, kernel and programs. Virtual machines are popular in the market but it's being over throned by the rise of containers.

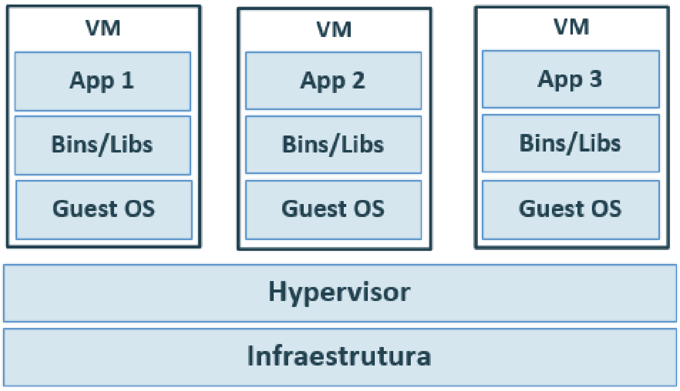


**Figure 1.** Virtual Machine architecture based on Docker.com diagram.

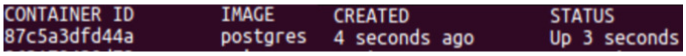


**Figure 2.** Start up time result of PostgreSQL database.

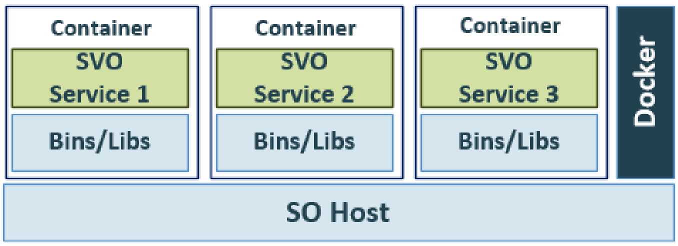


**Figure 3.** Docker architecture based on Docker.com diagram.

Containers aren't a new industry technology, but it was only with the invention of Docker platform, that the market began adhere. Containers are stateless components that works creating packages with applications, its dependencies and executable files (Dua et al. 2014). The fact that containers work as processes sharing the same OS and kernel (Figure 3) makes it a preferable option to VM's because it presents better performance in terms of hardware requirements and start up time. To take as an example, when running simple PostgreSQL data base , Docker takes only 3 seconds to start a PostgreSQL (Figure 2).

## 3. Technical approach

Each software architecture model presents it's benefits and issues, whether it's for implementation, performance or maintenance purposes. In the goal of reaching a manageable and flexible architecture, it's much needed a remodelling of the current SVO architecture so we introduce both container and micro services to build a new, flexible and optimized model CRAAM's SVO (Santos 2017).

To incorporate the new technologies into the architectural design, the study of the current application is imperative for a better understanding of the solution's functionalities and their integrations so we can build a modular version of application based on micro services.

With a good understanding of the solution, the development process will require a re-engineering of the current code in order to allow the use of micro services, guaranteeing that each service is independent and yet they interact efficiently as a single system at the user's perspective. Having a micro service architecture permits the optimized use of the container technology, by separating each service as a container, communicating between each other and enabling a faster scaling of the most requested service.

The container technology adopted will be Docker. The open source platform simplifies the build and management of containers and allows a communication between containers so the application services may interact during system execution.

## 4. Conclusion

This proposed model facilitates the DevOps practice, allowing to devide the SVO system development in groups, and simplifies upgrades and new functionality builds by initializing new containers and ending the old versions. The project reduces risks of downtimes, allowing rollback to previous stable containerized versions, and provides the choice of deployment in any environment: physical and virtual machines or public and private clouds.

*Acknowledgements.* I would like to thank MackPesquisa for the finantial support.

## References

Strimbei, C., Dospinescu, O., Strainu, R. M., Nistor, A. 2015, Informatica Economica, 19 (4), 13
Morris, D., Voutsinas, S., Hambly, N. C., Mann, R. G. 2017, Astronomy and Computing, 20, 105
Dua, R., Raja, A. R., Kakadia, D. 2014, In Cloud Engineering (IC2E), 610
De Carvalho, Reinaldo R. et al. 2010, Journal of Computational Interdisciplinary Sciences, 1, 187
Kaufmann, P. et al. 2001, IMOC 2001: Proceedings of the 2001 SBMO/IEEE MTT-S International. IEEE, p. 439.
Santos, I. F. 2017, Construção Iterativa de uma Ontologia com Integração Semântica para o Observatório Solar Virtual do Centro de Rádio Astronomia e Astrofísica Mackenzie. Tese de Doutorado, Universidade Presbiteriana Mackenzie
Valio, A. et al. 2013, Solar Physics, 283, 651
Almeida, M. B., Bax, Marcello P. 2003, Ciência da Informação, Brasília, 32 (3), 7
Berner-Lee, T., Hendler, J. 2003, Nature, 410, 1023
Borst, W. N. 1997, Construction of engineering ontologies for knowledge sharing and reuse. Universiteit Twente
Erl, T., Gee, C., Normann, H., Kress, J., & Maier, B. 2014, Next generation SOA: A concise introduction to service technology & service-orientation, Pearson Education
Bonerér, J. 2016, *Reactive microservices architecture: design principles for distributed systems*