Plug'in: A 5G Experimentation Platform for Collaborative and Reproducible Research

Ayoub Bousselmi, Sofiane Imadali, Marion Duprez
Orange Labs Networks - 44 Avenue de la République, 92320 Châtillon, France
first.second@orange.com

***Abstract*—We present Plug'in: a 5G experimentation platform to develop and test 5G software components and VNF chains. Plug'in allows researchers to collaborate on a shared platform and make their developments and experiments reproducible. The key enablers to achieve such a platform are cloud computing, virtualization and automation. Plug'in provides a sandbox where researchers can experiment on existing software components (also called atoms), chain them, and develop new ones. Atoms are deployed leveraging standard virtualization technologies regardless of their programming language. Finally, we evaluate the compute component of Plug'in.**

I. Introduction

Next generation mobile networks (5G) will leverage more software defined technologies from the radio to the network, storage, and upper layers. The goal is to accommodate numerous over-the-top applications through slicing, massive carrier aggregation, and dynamic bandwidth provisioning. Which will in turn enable faster innovation, better user experience and richer Telco use cases portfolio [1]. To support this evolution, 5G should include automation for its networks using Software Defined Networking (SDN), visibility and manageability of Virtual Network Functions (VNF), and scalability of mobile networking services using software techniques. To accelerate this phase, several stages should be taken into account: code integration, regression tests, deployment scenarios, operations feedback, and documentation. Once these requirements are met, researchers can allocate more time to develop innovative enhancements and provide them to the research community to experiment with, modify, and reproduce the claimed results. In this context, we introduce Plug'in: a 5G experimentation platform to develop and test 5G software and VNF chains. It allows researchers to collaborate on a shared platform and make their developments reproducible. It provides a sandbox environment where users can experiment existent software (also called atoms) and service chains, and develop new ones.

II. Plug'in platform overview

Plug'in provides a set of tools organized as a 5G Integrated Development Environment. It includes automatic project generation, code version control, online publication store, automatic documentation generation, and execution and experimentation environment. In the following subsections, we describe Plug'in tools from a user perspective as illustrated in figure 1.

A. AtomStore & AtomDocs

Efficiently locating all the needed tools and information for a research project can be a tedious task. It often involves exchanging emails, posting on social networks or using search engines before starting a project. AtomStore is proposed to cope with this problem. It allows researchers to publish their atoms using either AtomStore's user interface or API. AtomStore is a store for research results packaged as atoms. The goal is to gather all the needed information about research projects in a single place. It provides an ergonomic presentation of atoms in multiple categories: security, API, 5G, monitoring, networking, and IoT. Additionally, a documentation service, called AtomDocs, is a companion tool where the atom details about usage, configuration, implementation, contribution guide are included. AtomDocs provides a distraction-free portal where the user of an atom can read the detailed documentation. The documentation of an atom is written by the atom developers in the form of markdown files and included in the code repository. The documentation is automatically generated and integrated using a Continuous Integration pipeline.

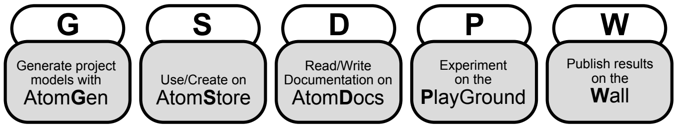

G
Generate project models with AtomGen
S
Use/Create on AtomStore
D
Read/Write Documentation on AtomDocs
P
Experiment on the PlayGround
W
Publish results on the Wall

Fig. 1: The Plug'in platform tools and their usage

B. AtomGen

Open source software with a fast growth rate may suffer from code bases divided into multiple packages (or modules) structured differently. Due to this variations enhanced by the great number of contributors (using different coding styles and practices), working on multiple modules might become a very challenging task, and intimidating to newcomers. We propose to simplify the project's management and maintainability by introducing a common project template that fits most needs, generated using our tool called AtomGen. The AtomGen allows 3 virtualization types: OS-level virtualization using Docker, hardware virtualization using Vagrant and no virtualization using the researcher's compute node as a working environment. It also proposes two different continuous integration technologies: GitLab-CI and Jenkins.

C. PlayGround

Once the researcher has found a convenient atom on the AtomStore, he/she can easily deploy it on the PlayGround.

9781538614785/18/$31.00 © 2018 IEEE

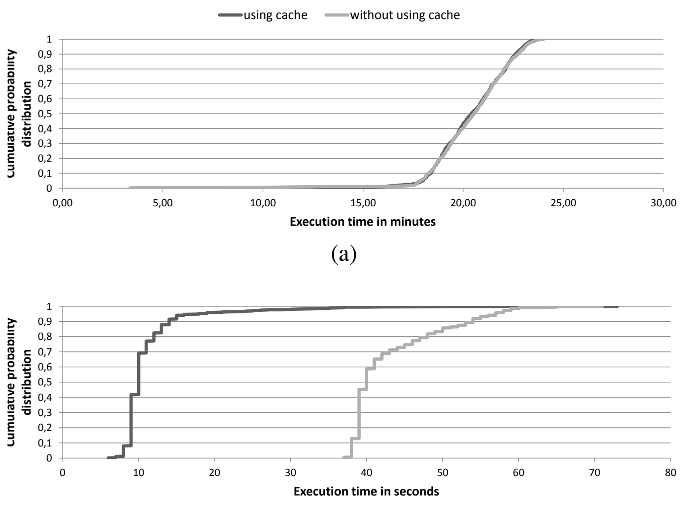
This service provides an on-demand experimentation environment using a web browser or an API. It creates time-limited sessions where users can create a number of compute instances. Researchers can use this tool to instantiate custom or pre-configured images (Sandboxes) to run experiments. It offers multiple functionalities such as: save the current session, share the experiment artifacts with the Plug'in community, upload files, and more. After finishing an experimentation, the whole environment is deleted. The PlayGround offers three sandbox images: Alpine, Ubuntu and CentOS. This service is based on an open source Docker community project called Play-with-Docker [2]. Finally, researchers can publish their results on a service called the Wall that acts as a *post-it* wall to share experimentation notes.

III. Evaluation

We propose an evaluation of the total runtime of experiments using a generic software called POV-Ray. The scenario uses POV-Ray with datasets from the Umbrella's public repository. We start 5 PlayGround sessions, 2 instances per session are launched, each instance of the experiment is executed 100 times sequentially. All instances are running concurrently, i.e. 10 tests are running simultaneously each time. The above setup is repeated twice, with and without the Docker local cache, to measure its impact on the execution time of the experiments. The impact of the Docker cache on the execution time (using the Umbrella's datasets) is presented in figure 2a. The cache has negligible impact on the total execution time. This may be explained by the fact that the execution comprises a networking phase, where the docker image of the tests is downloaded, and an actual execution on the CPU: creating the container, and executing the workloads. The caching of docker images locally on disk influences the networking phase of the experiment, which ends up being negligible compared to the computing phase in this first experiment as shown by the CDF of figure 2a. This is proven by the results of the second experiment illustrated in the CDF of figure 2b where a simpler POV-Ray script is used, consisting in less workload compared to the first test scenario.

IV. Related Work

Meng et al. [3] propose Umbrella: a framework preserving the execution environment of a computer science experimentation. Umbrella introduces an experiment specification file where dependencies such as hardware, operating system (OS), kernel, software, input data, environment variables, commands to be executed are exhaustively specified. Umbrella is designed to run single node experiments and does not provide the possibility to run multi-node scenarios. The FIT R2Lab is a wireless testbed providing an isolated experimental environment for software-based LTE based on OpenAirInterface (OAI) [4][5]. The testbed provides hardware and software support for OAI with remote access to control the entire chain in a complex 5G scenario from the Core Network to the RF frontends. Software components of the testbed are open source. At the time of writing, Plug'in does not provide a wireless experimental facility but could be connected to one. The Fraunhofer FOKUS 5G Playground [6] is a wireless testbed providing a software-based core and applications platform supporting different radio access technologies: WiFi, 4G, NB-IOT, and 5G NR. Software composing the platform is accessible through a paid license unavailable to authors at the time of writing.



(a)

(b)

Fig. 2: Impact of the docker cache on the execution time (a) using the umbrella dataset and configuration; and (b) using another dataset and configuration.

V. Conclusion

We presented Plug'in, a collaborative and integrative platform for reproducible research. Plug'in provides a set of services to publish, document and experiment a software brick. It proposes a simple method of organizing atom's code bases by providing a simple atom project generator. The PlayGround service in particular offers computing resources to test and modify atoms in a sandboxed fashion. We ran some performance tests on the PlayGround using an open source software.

References

[1] A. Gupta and R. K. Jha, "A survey of 5g network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[2] *Play with docker (pwd)*, https://labs.play-with-docker.com/, Accessed: 2018-02-13.

[3] H. Meng *et al.*, "Conducting reproducible research with umbrella: Tracking, creating, and preserving execution environments," in *2016 IEEE 12th International Conference on e-Science (e-Science)*, 2016, pp. 91–100.

[4] M. N. Mahfoudi *et al.*, "Deploy a 5g network in less than 5 minutes," in *Proceedings of the SIGCOMM Posters and Demos*, ser. SIGCOMM Posters and Demos '17, Los Angeles, CA, USA: ACM, 2017, pp. 113–115.

[5] *FIT R2lab: Reproducible research lab. at inria, sophia antipolis, france*, http://fit-r2lab.inria.fr/, 2017, Accessed: 2018-02-10.

[6] *Fraunhofer fokus 5g playground,* https://www.fokus.fraunhofer.de/go/en/fokus_testbeds/5g_playground, 2015, Accessed: 2018-02-10.