Contribution ID: **31** Type: **Presentation**

Rootless containers with udocker

Wednesday, 10 October 2018 12:00 (15 minutes)

Technologies based on Linux containers have become very popular among software developers and system administrators. The main reason behind this success is the flexibility and efficiency that containers offer when it comes to pack, deploy and run software. A containerized version of a given software can be created including all its dependencies, so that can be executed seamlessly regardless of the Linux distribution in the target hosts. Linux containers are also very well suited to the heterogeneous run-time environments that researchers face today when running complex applications across computing resources such as laptops, desktops, Linux interactive clusters, cloud providers, throughput computing and high performance computing infrastructures.

udocker is a tool developed by LIP in the context of the INDIGO-DataCloud project that addresses the problematic of executing Docker containers in user space, i.e. without installing additional system software, without requiring any administrative privileges and in a way that respects resource usage policies, accounting and process controls. udocker aims to empower users to execute applications encapsulated in Docker containers easily in any Linux system including computing clusters regardless of Docker or Linux namespaces being locally available.

udocker provides a command line interface similar to Docker and implements a subset of its commands aimed at searching, pulling, importing, loading and executing containers in a Docker like manner respecting much of the container metadata. The self installation allows a user to transfer the udocker Python script, execute it and automatically pull the required tools and libraries which are then stored in the user directory. This allows udocker to be easily deployed and upgraded by the user himself without system administrator intervention. All required binary tools and libraries are provided with udocker and compilation is not required.

udocker is an integration tool that incorporates several execution methods giving the user the best possible options to execute their containers according to the target host capabilities. Several interchangeable execution modes are available, that exploit different technologies and tools, which are integrated by udocker to enable execution both in older and newer Linux distributions. Currently four execution modes are available which can be selected dynamically, namelly:
* system call interception and pathname rewriting via PTRACE using a modified PROOT
* dynamic library call interception and pathname rewriting via ld_preload using a modified fakechroot
* Linux unprivileged namespaces using runC
* Linux namespaces using Singularity where available

Each approach has its own advantages and limitations, and therefore an integration tool offers flexibility and freedom of choice to adapt to the application and host characteristics. udocker is been successfully used to support execution of high throughput computing, high performance computing (MPI) and GPGPU based applications in many datacenters and infrastructures including EGI.

The udocker has more than 300 stars on github (https://github.com/indigo-dc/udocker). This presentation will provide further information about udocker and will highlight several user cases.

Type of abstract

Presentation

Summary

Containers are increasingly used as means to distribute and run Linux services and applications. udocker combines the pulling, extraction and execution of Docker containers without privileges. udocker is an integration tool that incorporates several approaches that enable the execution of Linux containers without requiring any priviliges across a wide-range of host systems, giving to the user the best possible options to execute their containers according to the target host capabilities. udocker is being used by many user communities to encapsulated applications and enable their seamless execution across computing sites without system administrators intervention.

Primary author: GOMES, Jorge (LIP)

Co-author: DAVID, Mario (LIP)

Presenter: GOMES, Jorge (LIP)

Session Classification: Computing Services: Part I

Track Classification: Area 3. Computing and Virtual Research Environments