CLOUD COMPUTING

Ravi Rajpurohit 1002079916

Paper 1 - Firecracker: Lightweight Virtualization for Serverless Applications

Review

In both public cloud and on-premises environments, serverless computing is getting popular for distributing and managing software and services. The serverless architecture is appealing for a variety of reasons, including interfaces with the event and streaming data sources, pay-for-use pricing, less work in maintaining servers and managing capacity, and automated scaling. Due to the ability to provision new functions and containers in a time as short as milliseconds and switch capacity between workloads as demand changes instantly, it becomes feasible to provide multitenancy, allowing servers to be shared across an excessive number of workloads. Still, they present significant challenges in isolating workloads from one another.

These problems have been resolved by cloud instance providers using hypervisor-based virtualization or by avoiding multi-tenancy and offering bare-metal instances. More workloads can be executed on a single computer thanks to serverless and container architectures, which raises any separation overhead. Common Linux container deployments, such as Docker and LXC, address this density issue by relying on isolation techniques inherent in the Linux kernel.

Firecracker could be a lightweight virtual machine which uses the Linux Kernel's KVM virtualization technology to handle modern Linux hosts as well as Linux and guest OS. The Firecracker is a Virtual Machine Monitor (VMM). It features a REST-based configuration API, serial console device emulation, disk and networking device emulation, and rate-limited network and disk output. One Firecracker process is operating in each MicroVM, providing a simple paradigm for security isolation. It offers a solution for the problems with virtualization by providing the mentioned qualities. It was targeted to create a VMM for certain objectives that could be foreseen in terms of visitor demands and characteristics than it was to create a VMM that would work for everyone. The development of Firecracker reflects these simplification presumptions. They decided to keep QEMU in Firecracker but replace it entirely with KVM. Firecracker offers a replacement foundation for providing isolation between containers and functions along with KVM.

The qualities of an ideal isolation system are performance, isolation, overhead and density, compatibility, fast switching, and soft allocation. While some of these traits like isolation are obstinately qualitative, others are converted into quantitative goals. There are three methods for segregating workloads in Linux. Firstly containers, where all workloads share a kernel and are isolated using a mix of kernel methods. Secondly, virtualization, in which workloads operate inside individual VMs under a hypervisor. And lastly, Language VM isolation, where the language VM is to blame for separating workloads from one another or from the software system.

The Firecracker is prepared to develop all of the best qualities. The Firecracker does not imitate legacy or PCI devices. It does not support VM migration nor does it offer BIOS. Also, it cannot boot arbitrary kernels. Microsoft Windows could not be booted until Firecracker was significantly modified. Moreover, Firecracker does not offer VM orchestration, packaging, administration, or any other functionality because of its process-per-VM architecture. Since 2018, this has frequently been in use in production, supporting billions of requests per month and an infinite number of production workloads.