Event-driven Computing on HPC: Experiments with Scientific Applications

Hyungro Lee
Advisor: Geoffrey C. Fox, Indiana University Bloomington
lee212@cs.indiana.edu

January 6, 2018

1 Abstract

High performance computing requires to deal with diverse applications running on different environments, and HPC containers which encapsulate required software components and configurations in an image enable reproducibility of your workloads and increase portability across platforms. Serverless computing or Function-as-a-Service (FaaS) deploys a container to execute lines of codes without management of infrastructure which is more like a simple version of Platform-as-a-Service (PaaS) of cloud computing. Commercial cloud providers such as Amazon, Google, Microsoft and IBM offer serverless computing with various function runtimes along with open source implementations e.g. OpenWhisk, Fission and Kubeless but any of these platforms are suitable for HPC applications which require particular libraries and specific hardware e.g. GPUs and interconnects. We. however, this "serverless" model may find useful in processing workloads in certain circumstances, for example, a large number of small tasks that can be run in parallel for very short periods of time or a workflow consisting of small input and output messages in each component. We investigate existing serverless computing platforms to evaluate its possible use cases on HPC sites with regarding to 1) elasticity, 2) scalability, 3) portability, and 4) continuous deployment. This work conducts a study of building reproducible application environments on two different systems; HPC and Clouds with our comparison results for serverless platforms in regard to computational performance, latency, event handler capacity, and elasticity. Our results indicate that a function invocation spawns at least a thousand of concurrent computation containers in seconds and keep alive for another tasks until its deprovisioning whereas long-running HPC applications with a large size of libraries and binary executables are not yet feasible.