HotC: Mitigating Startup Latency of Serverless Applications via Reusing Container Runtimes

Kun Suo[1], Yong Shi[1], Xiaohua Xu[1], Dazhao Cheng[2], and Wei Chen[3]

[1]*Kennesaw State University,* [2]*University of North Carolina at Charlotte,* [3]*Nvidia*

Abstract

During the past few years, serverless computing has changed the paradigm of application development and deployment in the cloud and edge due to its unique advantages, including easy administration, automatic scaling, built-in fault tolerance, etc. Nevertheless, serverless computing is also facing challenges such as long latency due to the container cold start. In this paper, we propose HotC, a container-based runtime management framework which leverages the lightweight containers to mitigate the cold start and improve network performance of serverless applications. Our evaluation results show that HotC introduces negligible overhead and can efficiently improve the performance of various applications in both cloud servers and edge devices.

1 Approach

As the traditional market of cloud computing turns mature and user requirement for microservices keeps growing, serverless computing, such as Amazon Lambda , Microsoft Azure Function and Google Cloud Function , which provides high performance, high scalability, built-in fault tolerance, is becoming increasingly popular in public clouds. Serverless infrastructure allows developers to focus on application and business logic itself instead of worrying about where to deploy their codes and how to tweak large number of servers. However, such the design might also introduce performance loss due to the cold start, especially to I/O-intensive applications. For instance, Amazon reported that every 100ms of latency costs them 1% in sales and page speed of websites is also treated by Google as one of the major ranking factors .

Another character of the serverless computing is that the packaged functions have high similarities and many of them execute in the same kind of container runtime including OS images, programming language, configuration, etc. For instance, Microsoft has revealed that about 40% of key jobs or services at Bing search rerun periodically . Besides, Cito et al. analyzed thousands of Dockerfiles from GitHub projects and they reported that both the top 100 popular and all projects

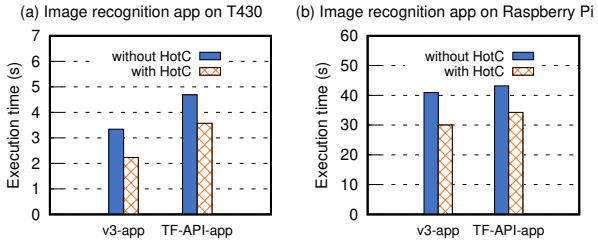

Figure 1: The image recognition application execution time w/o and w/ HotC.

are dominated by few commonly used images, which mostly contain the similar OSes, language runtimes, etc., or their combination.

There exists a semantic gap between the cold start issue and inefficient utilization of container runtime environment. To address cold start in serverless services, our key observation is that the runtime could be reused efficiently by leveraging the lightweight containers and the homogeneity of containerized serverless applications. Therefore, we proposed and developed HotC, a container-based runtime management framework which provides low-latency request handling while minimizing the performance overhead to applications.

2 Evaluation

We evaluated startup time of two image recognition applications with HotC. First, we evaluated application execution time on PowerEdge T430 server. As Figure 1(a) shows, the execution time of v3-app and TF-API-app reduced by 33.2% and 23.9% respectively compared to that without HotC. Similarly, we also evaluated the performance on Raspberry Pi. Compared to physical servers, the normal execution time of the same application prolongs more than 10 times inside edges devices and makes the cold start impact less significant among the total execution time. However, as depicted in Figure 1(b), HotC still helped reducing the execution time of v3-app and TF-API-app by 26.6% and 20.6%, respectively.