Contribution ID: **262** Type: **not specified**

Making the Kubernetes Service Abstraction Scale using eBPF

Tuesday, 10 September 2019 15:00 (45 minutes)

In this talk, we will present a scalable re-implementation of the Kubernetes service abstraction with the help of eBPF. We will discuss recent changes in the kernel which made the implementation possible, and some changes in the future which would simplify the implementation.

Kubernetes is an open-source container orchestration multi-component distributed system. It provides mechanisms for deploying, maintaining and scaling applications running in containers across a multi-host cluster. Its smallest scheduling unit is called a pod. A pod consists of multiple co-located containers. Each pod has its own network namespace and is addressed by an unique IP address in a cluster. Network connectivity to and among pods is handled by an external plugin.

Multiple pods which provide the same functionality can be grouped into services. Each service is reachable within a cluster via its virtual IP address allocated by Kubernetes. Also, a service can be exposed to outside of a cluster via the public IP address of a cluster host IP address and a port which is allocated by Kubernetes. Each request sent to a service is load-balanced to any of its pods.

Kube-proxy is a Kubernetes component which is responsible for the service abstraction implementation. The default implementation is based on Netfilter's iptables. For each service and its pods it creates couple rules in the nat table which do a load-balancing to pods. For example, for the "nginx" service which virtual IP address is 10.107.41.178 and which is running two pods with IP addresses 10.217.1.154 and 10.217.1.159 the following relevant iptables rules are created:

<pre> <code> -A KUBE-SERVICES -d 10.107.41.178/32 -p tcp -m comment –comment "default/nginx: cluster IP" -m tcp –dport 80 -j KUBE-SVC-253L2MOZ6TC5FE7P

-A KUBE-SVC-253L2MOZ6TC5FE7P -m statistic –mode random –probability 0.50000000000 -j KUBE-SEP-PCCJCD7AQBIZDZ2N -A KUBE-SVC-253L2MOZ6TC5FE7P -j KUBE-SEP-UFVSO22B5A7KHVMO

-A KUBE-SEP-PCCJCD7AQBIZDZ2N -s 10.217.1.154/32 -j KUBE-MARK-MASQ -A KUBE-SEP-PCCJCD7AQBIZDZ2N -p tcp -m tcp -j DNAT –to-destination 10.217.1.154:80 -A KUBE-SEP-UFVSO22B5A7KHVMO -s 10.217.1.159/32 -j KUBE-MARK-MASQ -A KUBE-SEP-UFVSO22B5A7KHVMO -p tcp -m tcp -j DNAT –to-destination 10.217.1.159:80 </code> </pre>

It has been demonstrated \[1\]\[2\]\[3\] that kube-proxy due to its foundational technologies (Netfilter, iptables) is one of the major pain points when running Kubernetes at large scale from performance, reliability, and operations perspective.

Cilium is an open-source networking and security plugin for container orchestration systems, such as Kubernetes. Unlike the majority of such networking plugins, it heavily relies on eBPF technology which lets one to dynamically reprogram the kernel.

The most recent Cilium v1.6 release brings the implementation in eBPF of the Kubernetes service abstraction. This allows one to run a Kubernetes cluster without kube-proxy. Thus, it makes Kubernetes no longer dependent on Netfilter/iptables. This improves scalability and reliability of a Kubernetes cluster.

No Kubernetes knowledge is required. The talk might be relevant for those who are interested in container networking with eBPF (loadbalancing, NAT).

\[1\]: https://sched.co/MPch
\[2\]: https://bit.ly/2xKk2pr
\[3\]: https://bit.ly/2WU7BCN

I agree to abide by the anti-harassment policy

Yes

I confirm that I am already registered for LPC 2019

Primary authors: Mr DANIEL, Borkmann (Cilium); Mr MARTYNAS, Pumputis (Cilium)

Presenters: Mr DANIEL, Borkmann (Cilium); Mr MARTYNAS, Pumputis (Cilium)

Session Classification: Networking Summit Track