

[TECH01]

TVP Tencent Cloud
Valuable Professional

intel

/serverless/DAYS

Techo x TVP 开发者峰会

/serverless/DAYS China
2021

无服务器，大有未来
Serverless, Empower More

microVMM —— Serverless 核心技术揭秘

费良宏

</>

β

/...

#

☰

∞

{ }



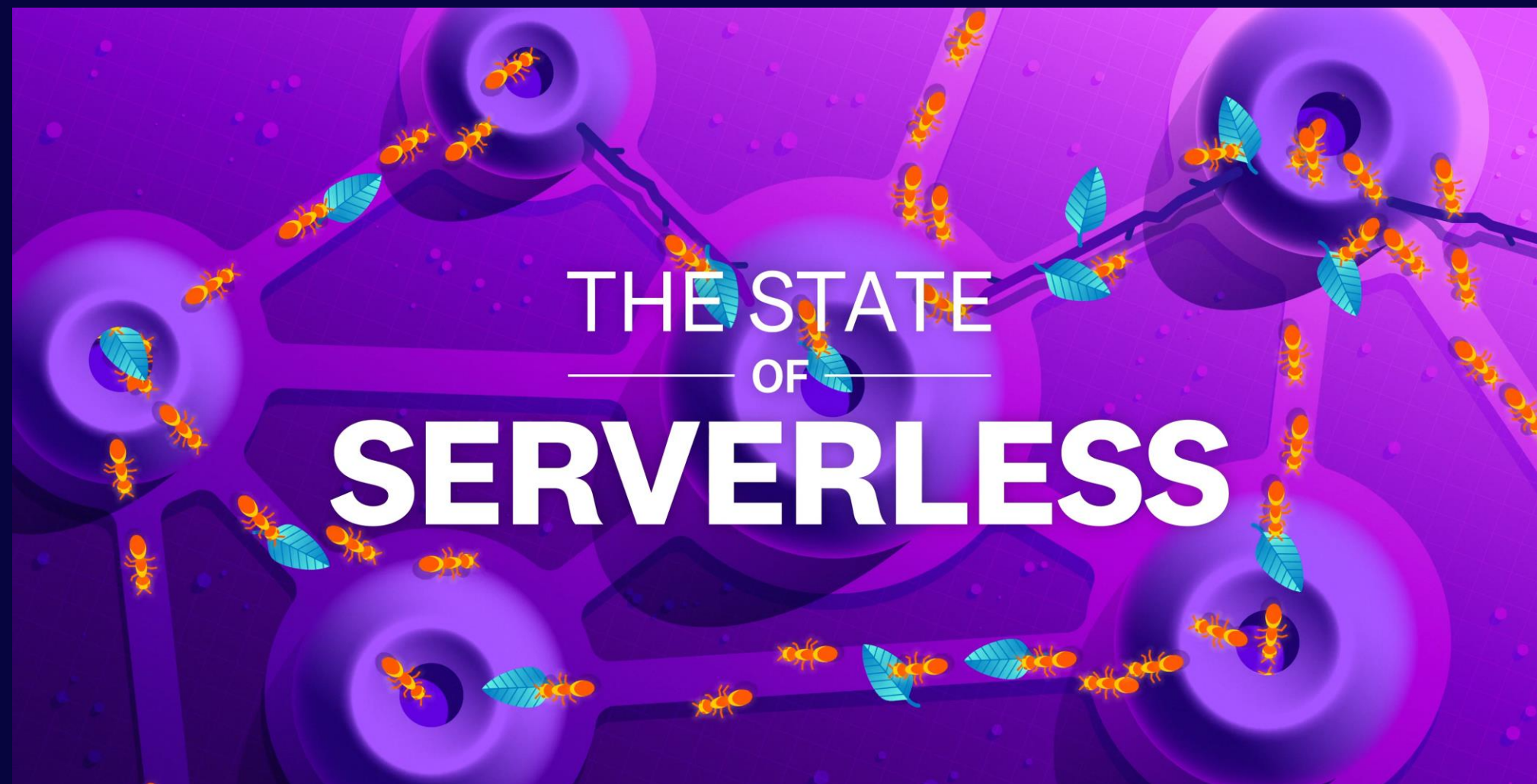
费良宏

*Amazon Web Services /
Principal Developer Advocate*

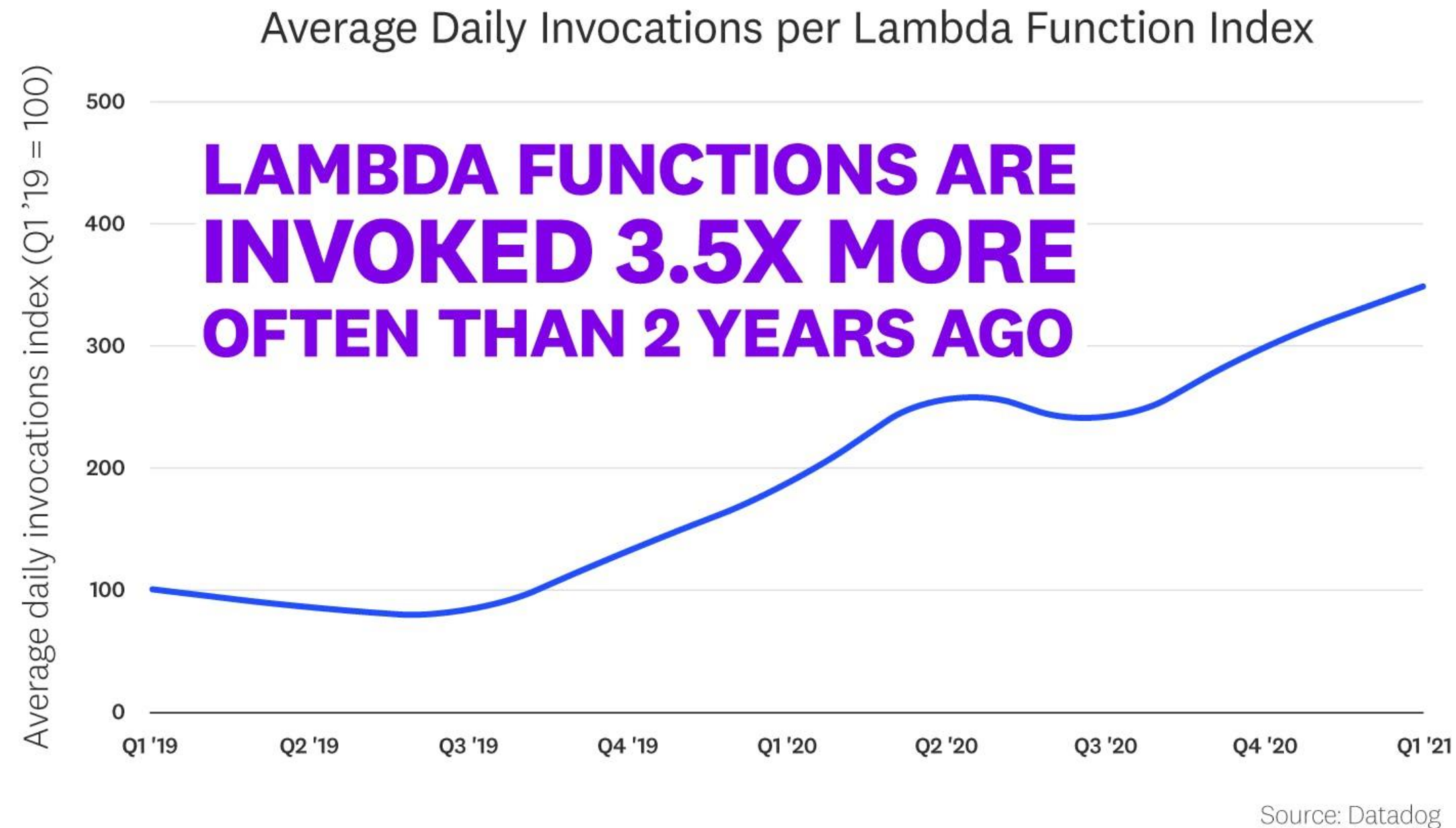


feilianghong@hotmail.com

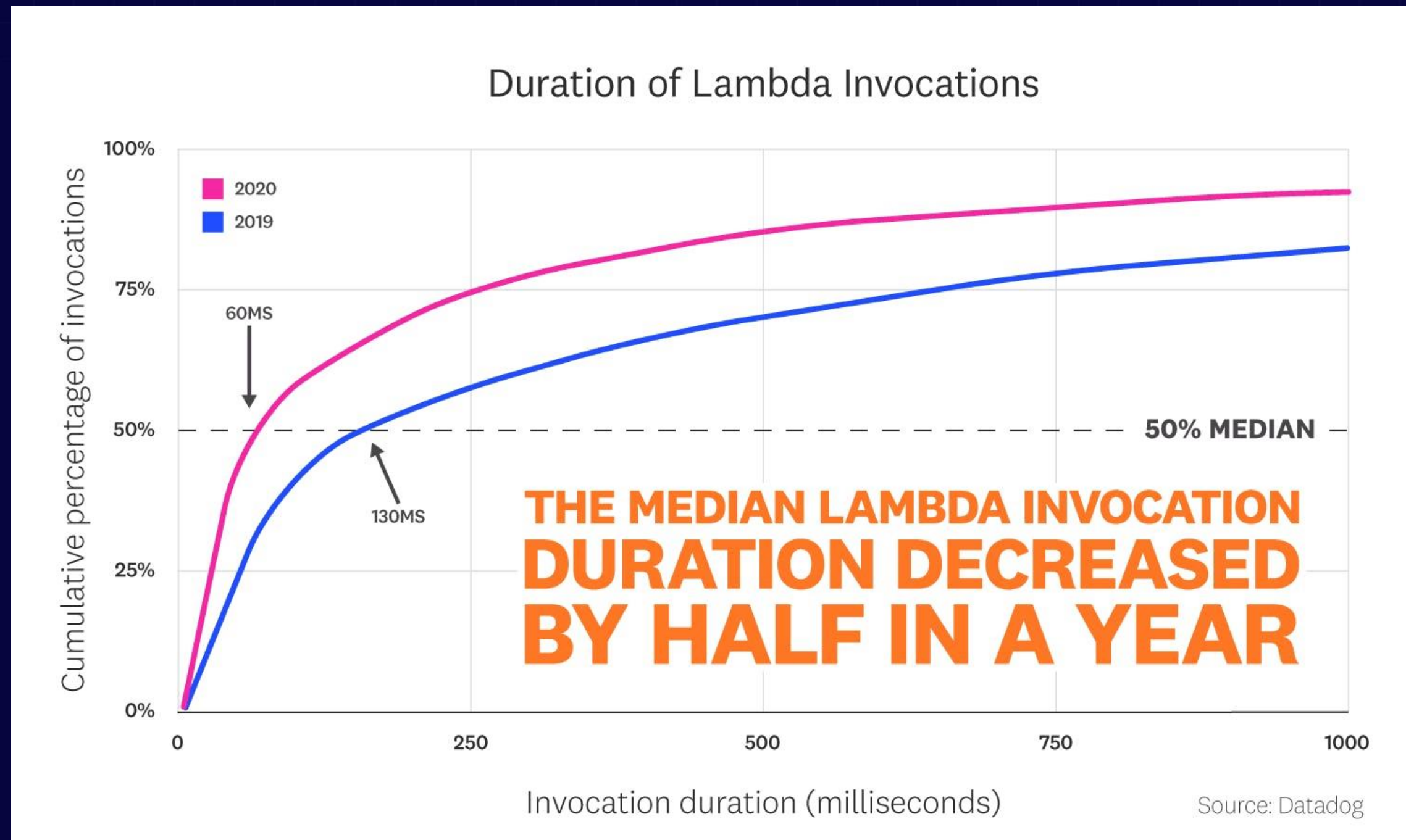
```
1 import json
2
3 def lambda_handler(event, context):
4     # TODO implement
5     return {
6         'statusCode': 200,
7         'body': json.dumps('Hello World!')
8     }
```



2021 State of Serverless Report



Amazon Lambda 函数的调用频率是两年前的 3.5 倍



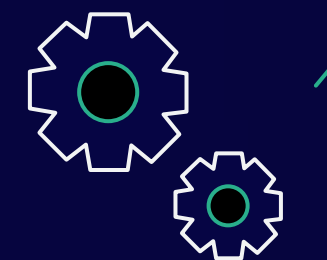
今天 *Lambda* 调用时间比一年前要短得多



安全



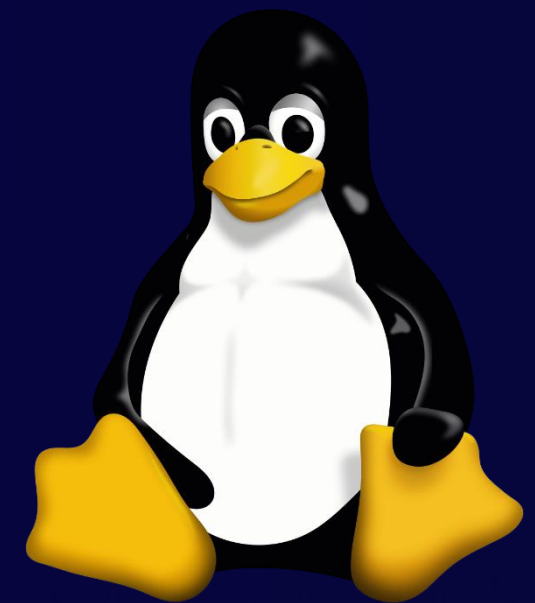
启动时间

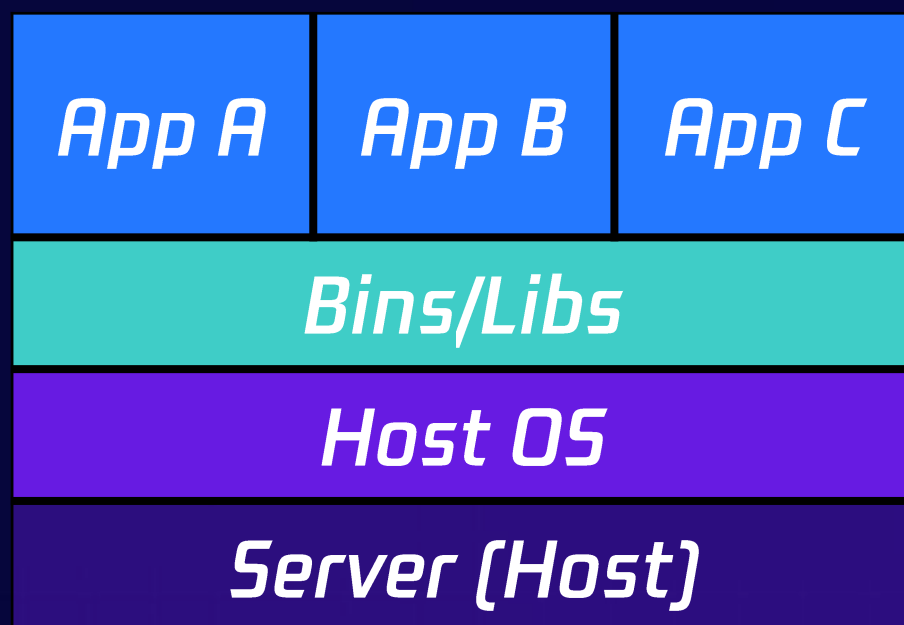


利用率

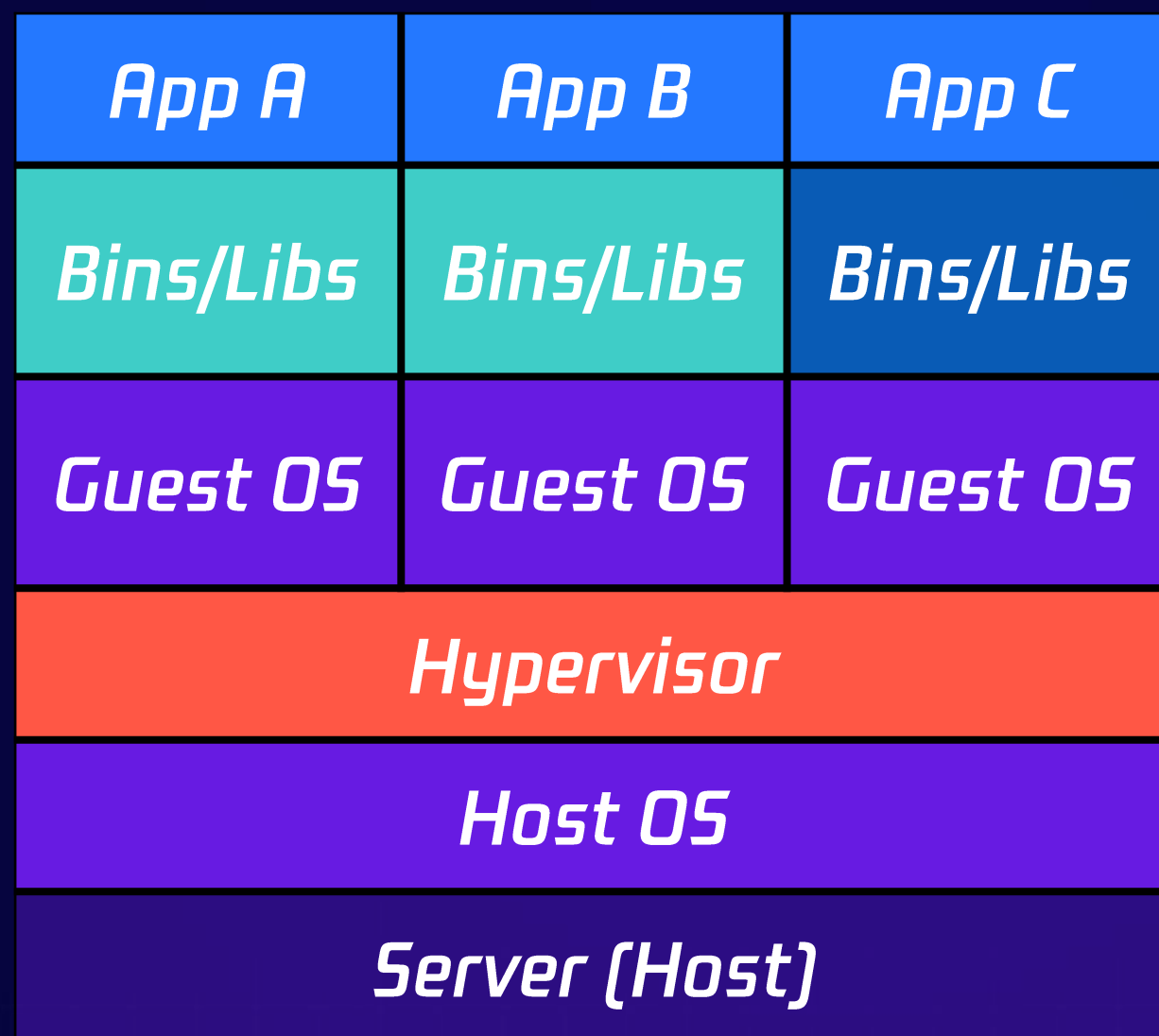


- *Namespaces* – 可见性限制
- *Control groups (cgroups)* – 资源限制
- *Capabilities* – 权限限制
- *Seccomp* – Syscall 允许/拒绝列表
- *Linux Security Modules* – 资源访问控制
- *Union filesystems* – 映像分层

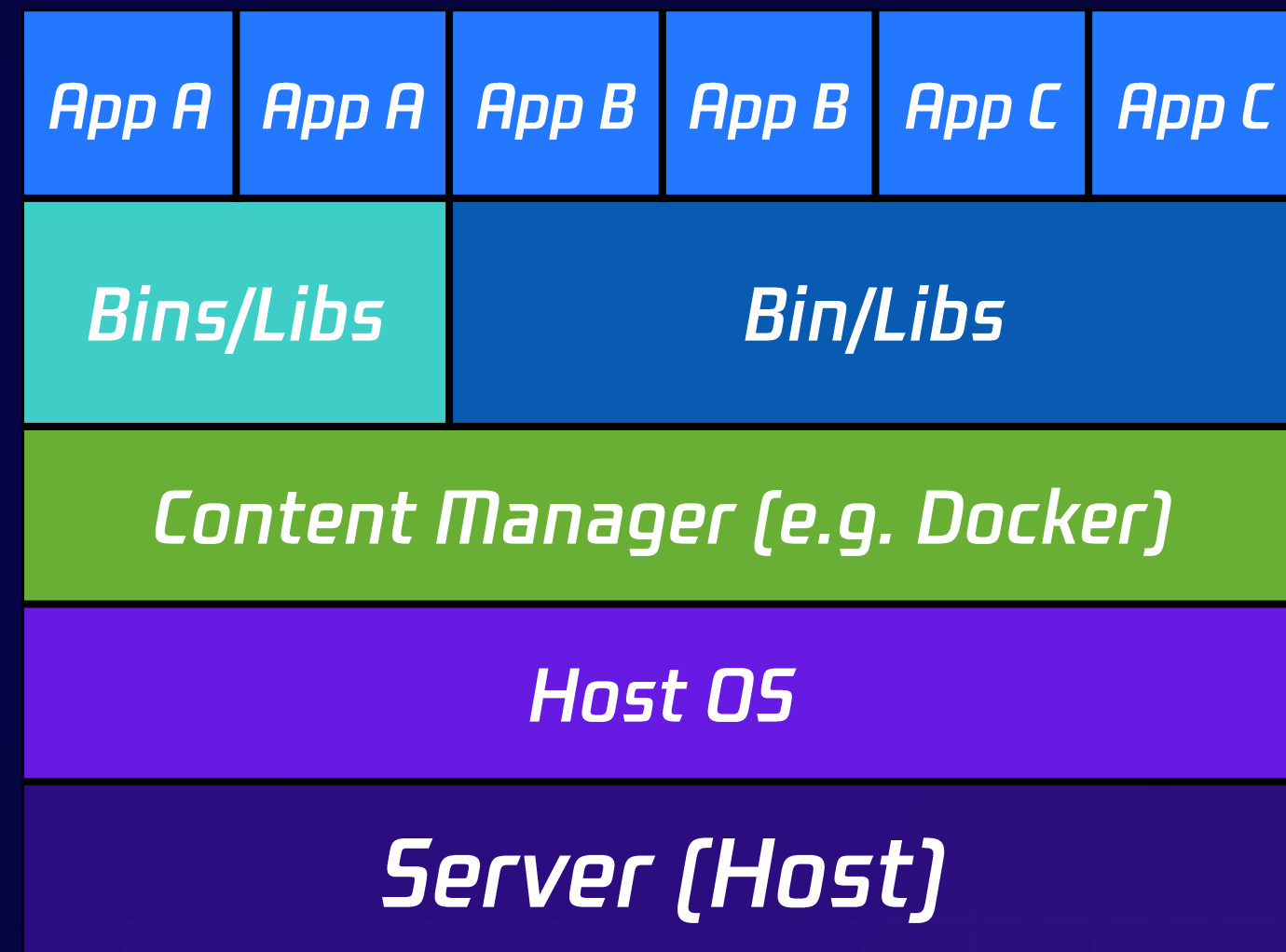




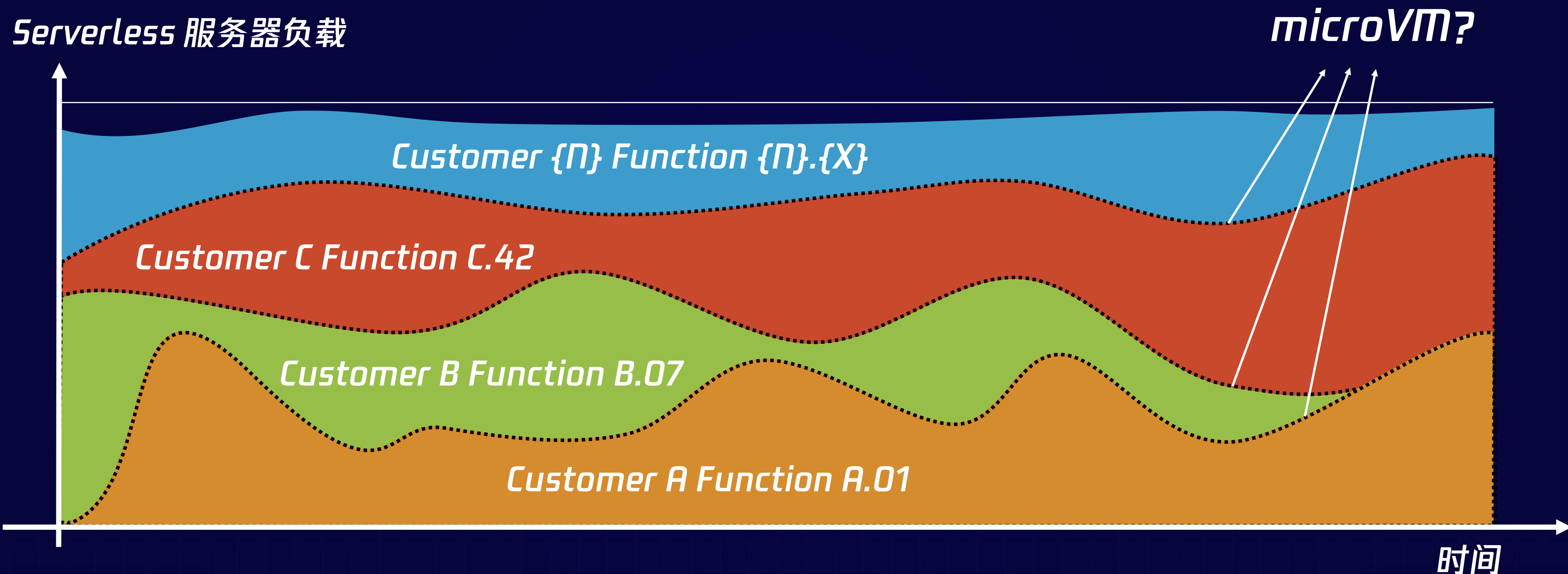
裸金属



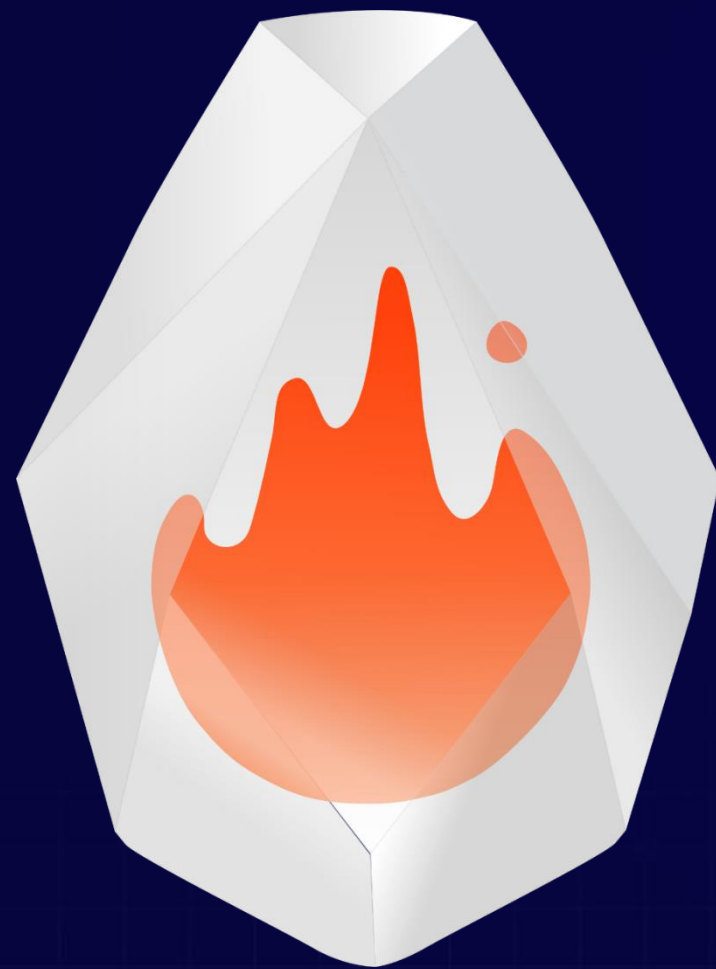
虚拟机



容器



- 为什么需要 *Firecracker* ?
- 什么是 *Firecracker*? 性能如何?
- 未来的挑战?



Firecracker

01 *Serverless* 为何需要一个 *microVMM*?

EC2 m5.metal 裸金属实例
384GB 内存

最小的Lambda函数
需要128MB内存

隔离:

在一个机器上运行数千个 *Serverless* 函数

负载 & 计算密度:

在同一个硬件环境上运行多个函数必须是安全的

性能:

Serverless 的函数的性能必须与原生运行相近

兼容性:

支持任意的 *Linux* 二进制文件和库

无需改变代码或重新编译

资源分配:

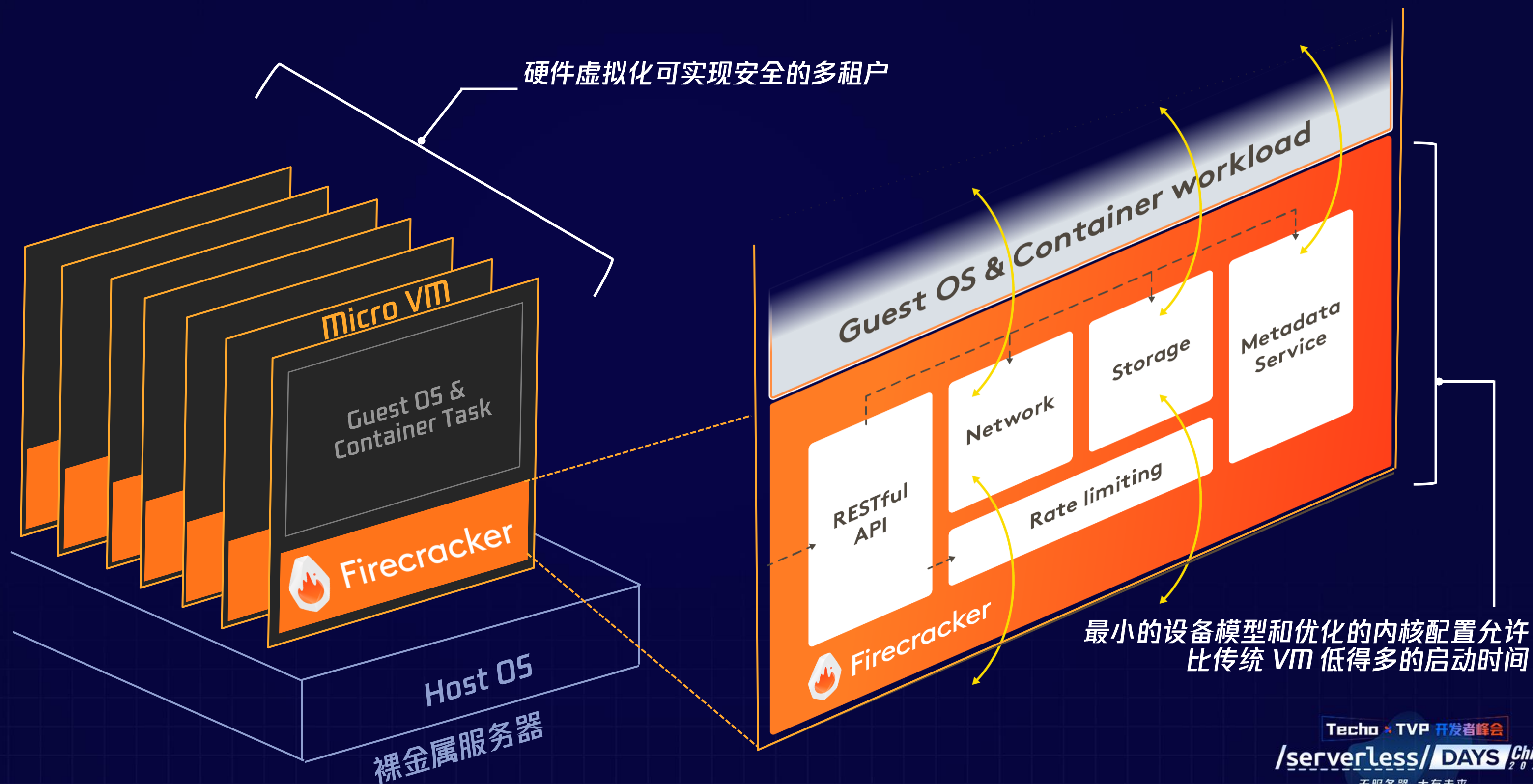
必须有可能对CPU、内存和其他资源
提供过度订阅的能力

- *QEMU/KVM*: 计算密度与负载的问题
- *Linux containers*: 隔离与兼容性的问题
- *LibOS approaches*: 兼容性问题
- *Language VM isolation*: 兼容性与隔离的问题

02 什么是 *Firecracker*?



专为 Serverless 工作负载而构建的开源虚拟化技术



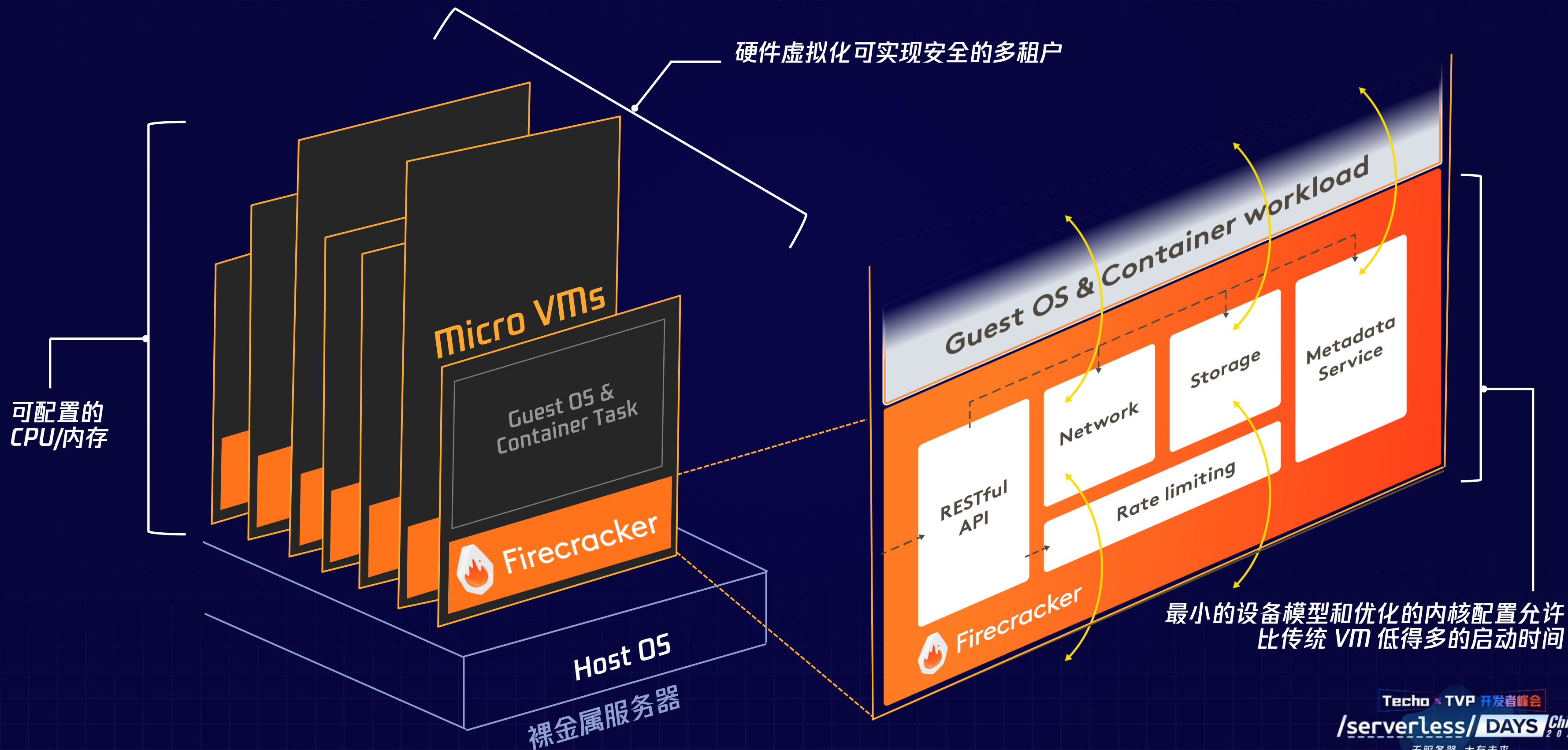


什么是 Firecracker

[TECH001]

TVP Tencent Cloud Valuable Professional

/serverless/DAYS





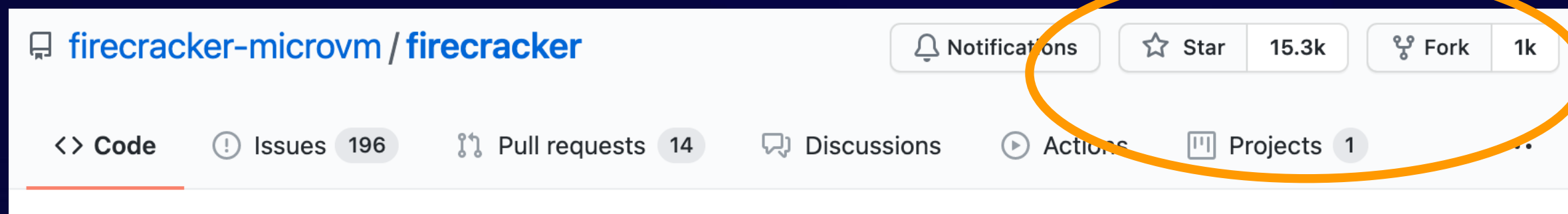
什么是 *Firecracker*

[TECH01]

TVP Tencent Cloud
Valuable Professional

/serverless/DAYS

- 基于KVM 的 VMM
- 使用 Rust 开发
- 开源, Apache 2.0
- 针对无服务器工作负载
- 不是通用 VMM

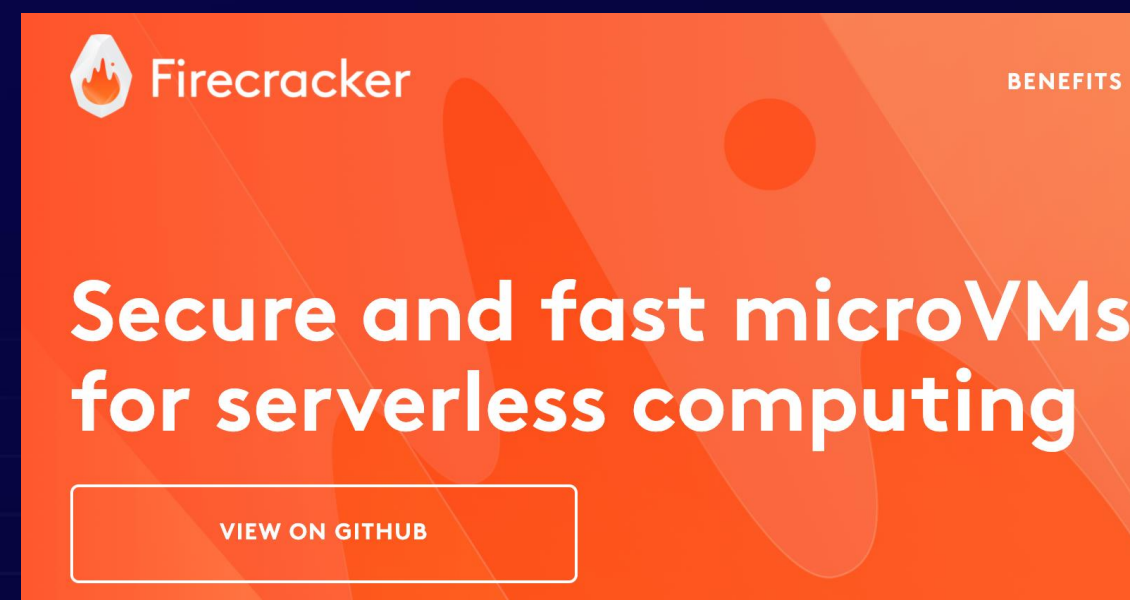


项目地址 <https://github.com/firecracker-microvm/firecracker>
当前版本 v0.24.4

在*Firecracker*的开发过程中，就已经意识到*Firecracker*对开源社区的价值，例如在容器隔离方面

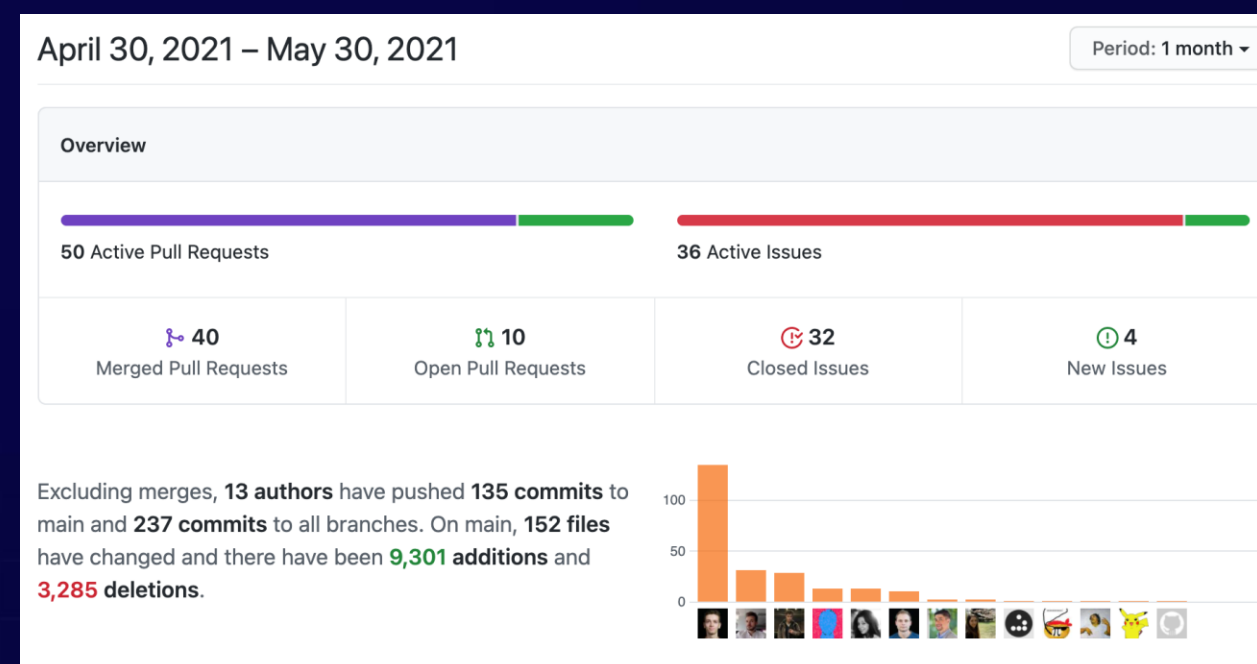
将技术分享给*Amazon Lambda*的开发者。当运行*Lambda*函数时，开发者可以有效的利用这个运行环境

在*re:Invent 2018*大会上宣布开源，许可证为*Apache 2.0*



截止到目前, 有来自48个开源社区的147个贡献者 (约占总数的18%), 加上许多社区bug报告和特性请求, 以及有价值的RFC反馈

被Weave Works Ignite/Firekube、Kata Containers、UniK、OSv 所集成, 可作为containerd的运行时 (通过 *firecracker-containerd*)





- 多租户
- 任何vCPU和内存的组合
- 允许超额
- 稳定的突变率, *100+ microVMs/host/sec*
- 仅受硬件资源限制
- 面向宿主机的 *REST API*
- 极简的 *Guest* 设备模式



安全

- 有限的设备型
- 有限的功能集
- 消除Guest与主机内核的交互
- 沙箱/jail the VMM
- 内存安全的编程语言
- 每个Firecracker进程有一个VM

效率

- 快速启动
- 低内存和CPU开销
- API 驱动

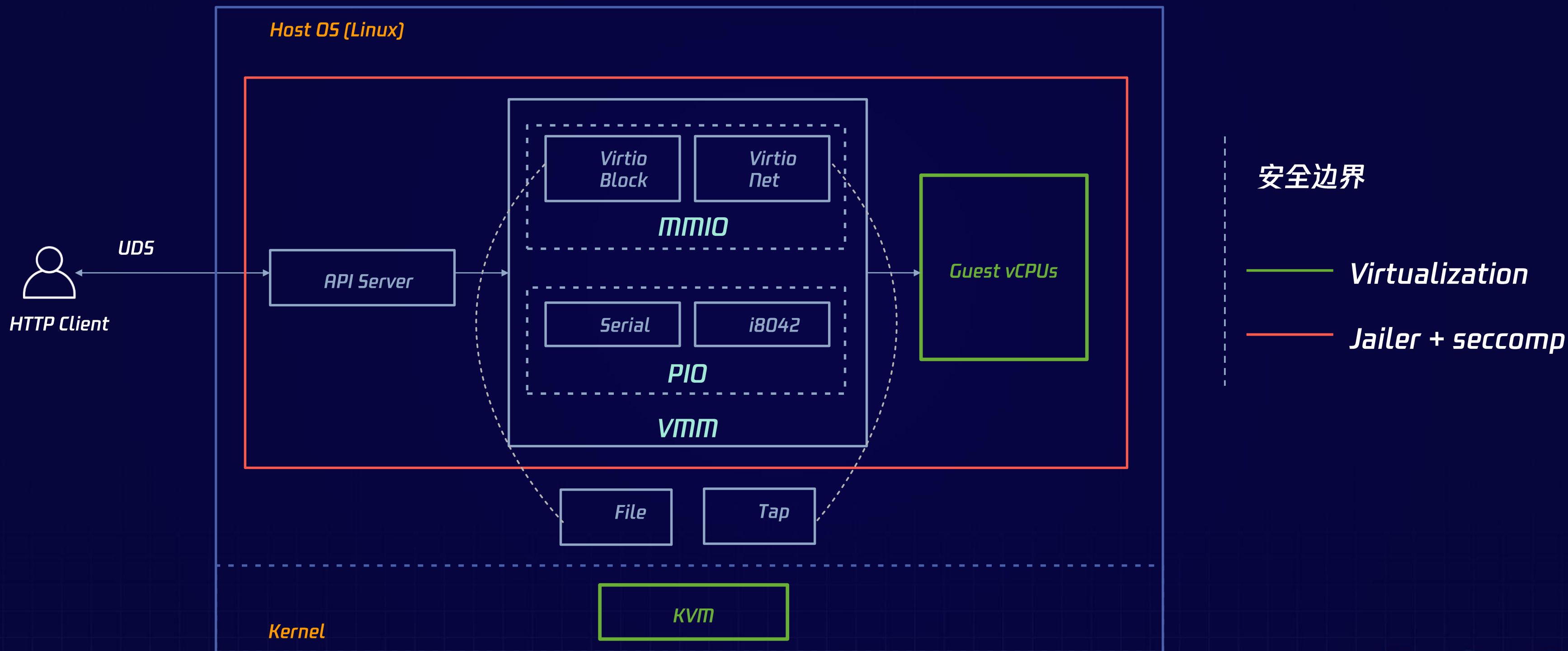


Firecracker 的关注

- 无文件系统共享
- 无动态设备附件
- 有限的网络选项
(tap, 而不是 veth)
- 与VSOCK的跨界通讯

适用于 containerd

- 块设备的快照
- 管理VMM生命周期的API
- 将"shim"分成两部分
 - Runtime 在 host上
 - Agent 在VM内
 - 通过 runc 运行容器
- 网络
 - tap 设备
 - 可与容器网络接口(CNI)插件配合用





采用 *Rust* 开发，为什么？

[TECH01]

TVP Tencent Cloud
Valuable Professional

/serverless/DAYS

内存安全

- 没有 *use-after-free*
- 不读取未初始化的内存

线程安全

- 没有数据争用





内存安全

- VM 内存模型
- 半虚拟化 (*virtio*)

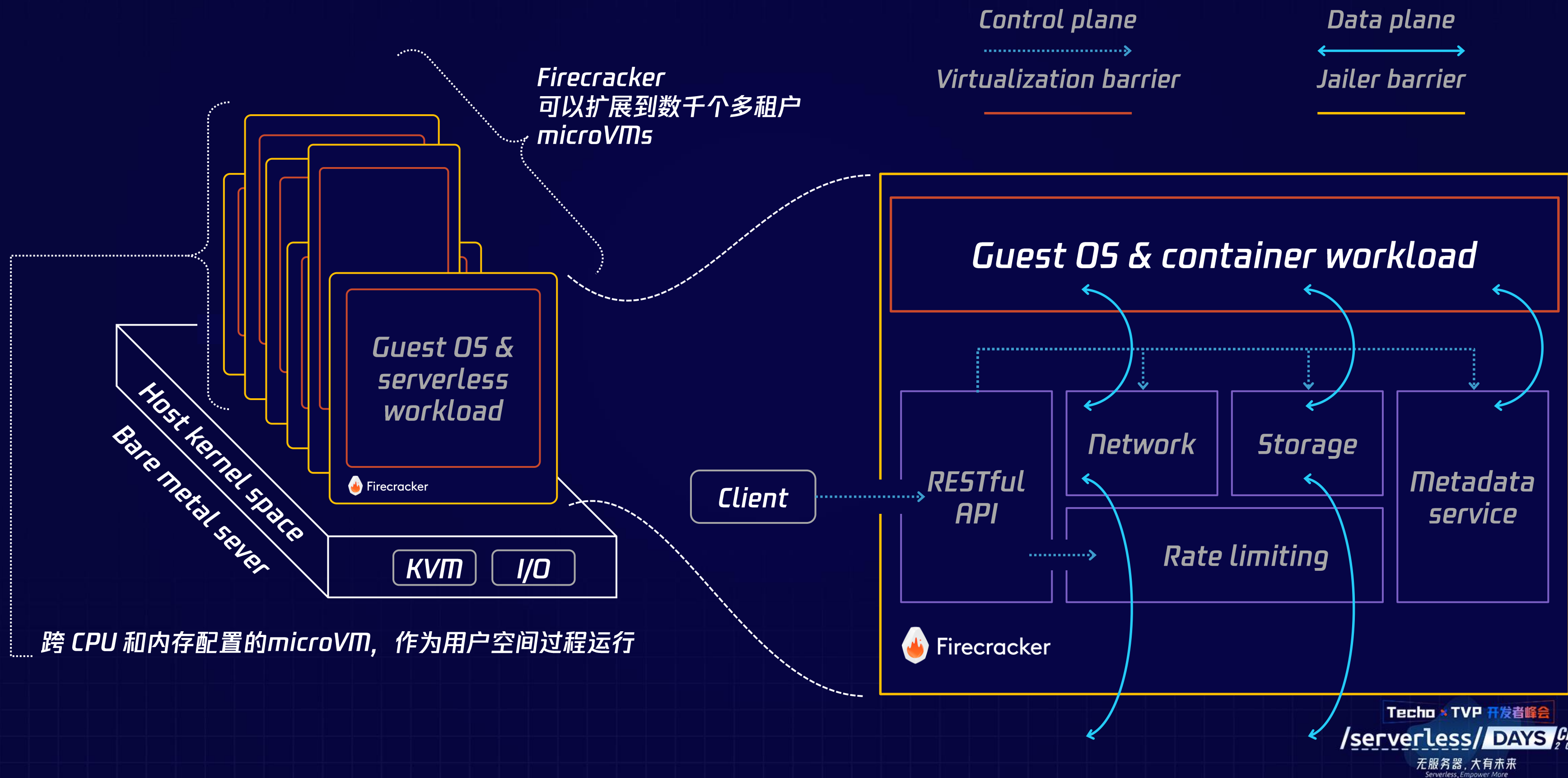
线程安全

- vCPU 线程
- 设备模拟线程程

低性能的惩罚

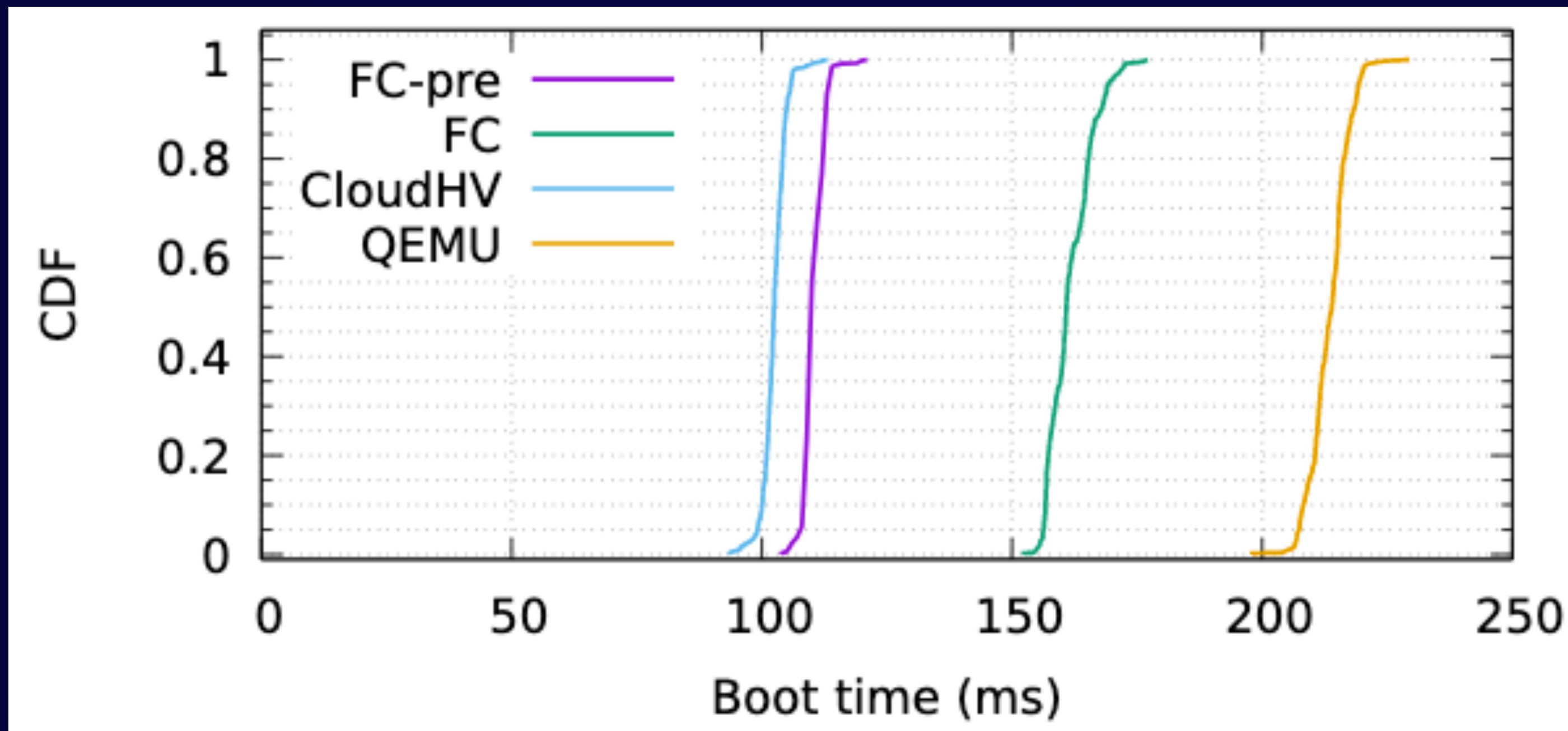
Firecracker microVMM 的特性

- 基于KVM的采用Rust语言的VMM
- 在Apache许可下开放源代码
- 针对Serverless的工作负载
- 不是一个通用的VMM
- 支持多租户
- 任何vCPU和内存组合
- 允许超额订阅
- 快速启动 [125毫秒用户空间启动]
- 快速恢复 [5毫秒到快照点]
- 稳定的突变率, 150+ microVMs/host/sec
- 更少的内存开销 (<5 MiB)
- 仅受硬件资源限制 (i3.metal中支持4000多个VM)
- 面向主机的 REST API
- 极简的客户设备模型

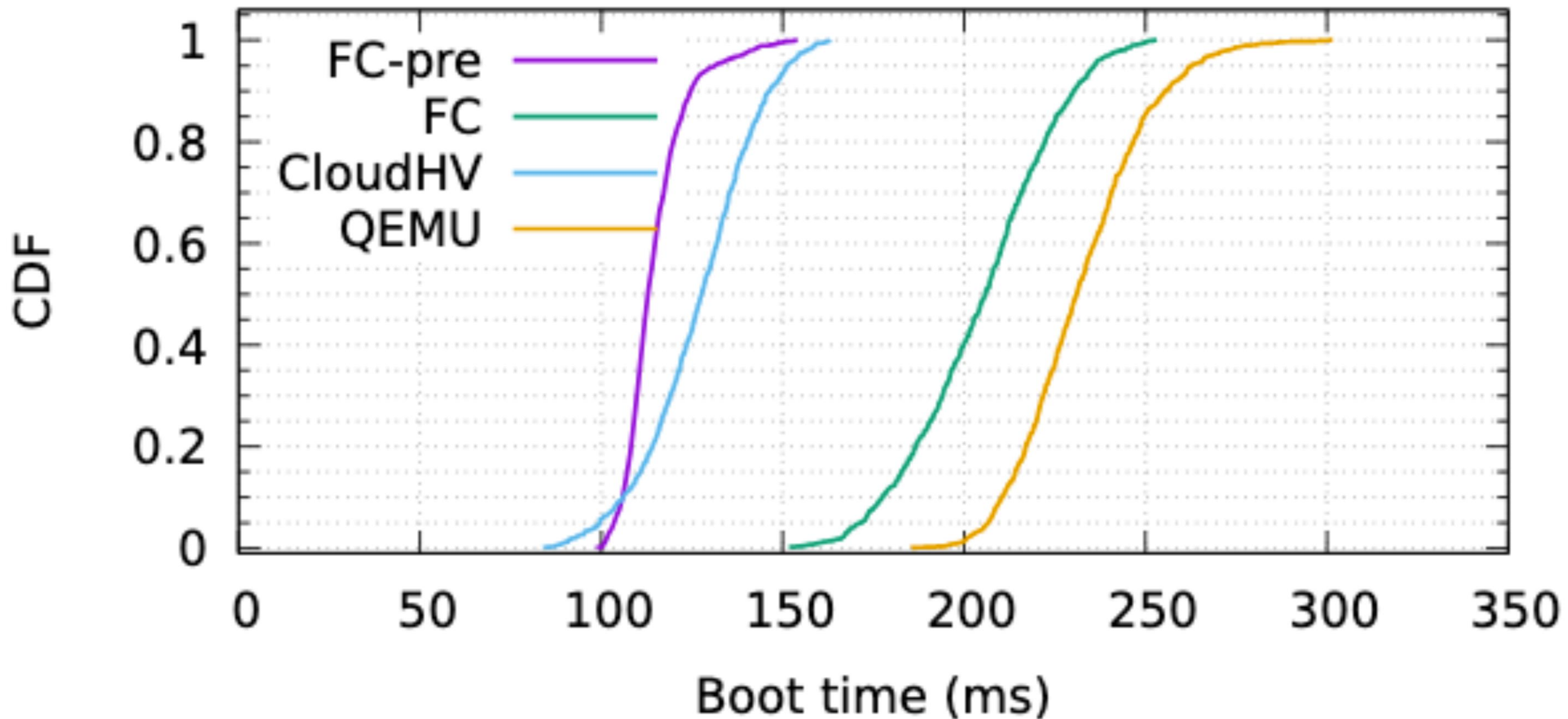


03 性能表现

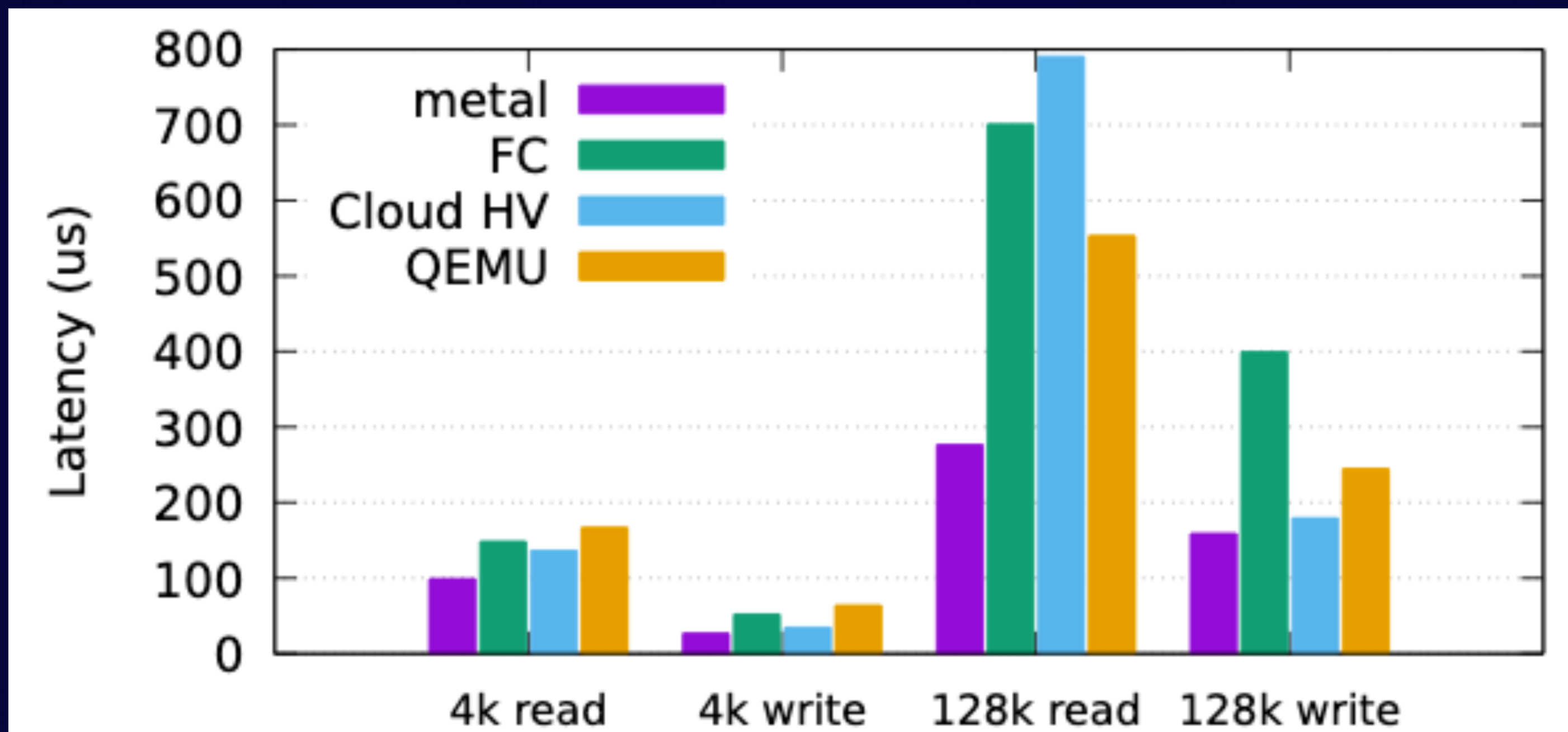




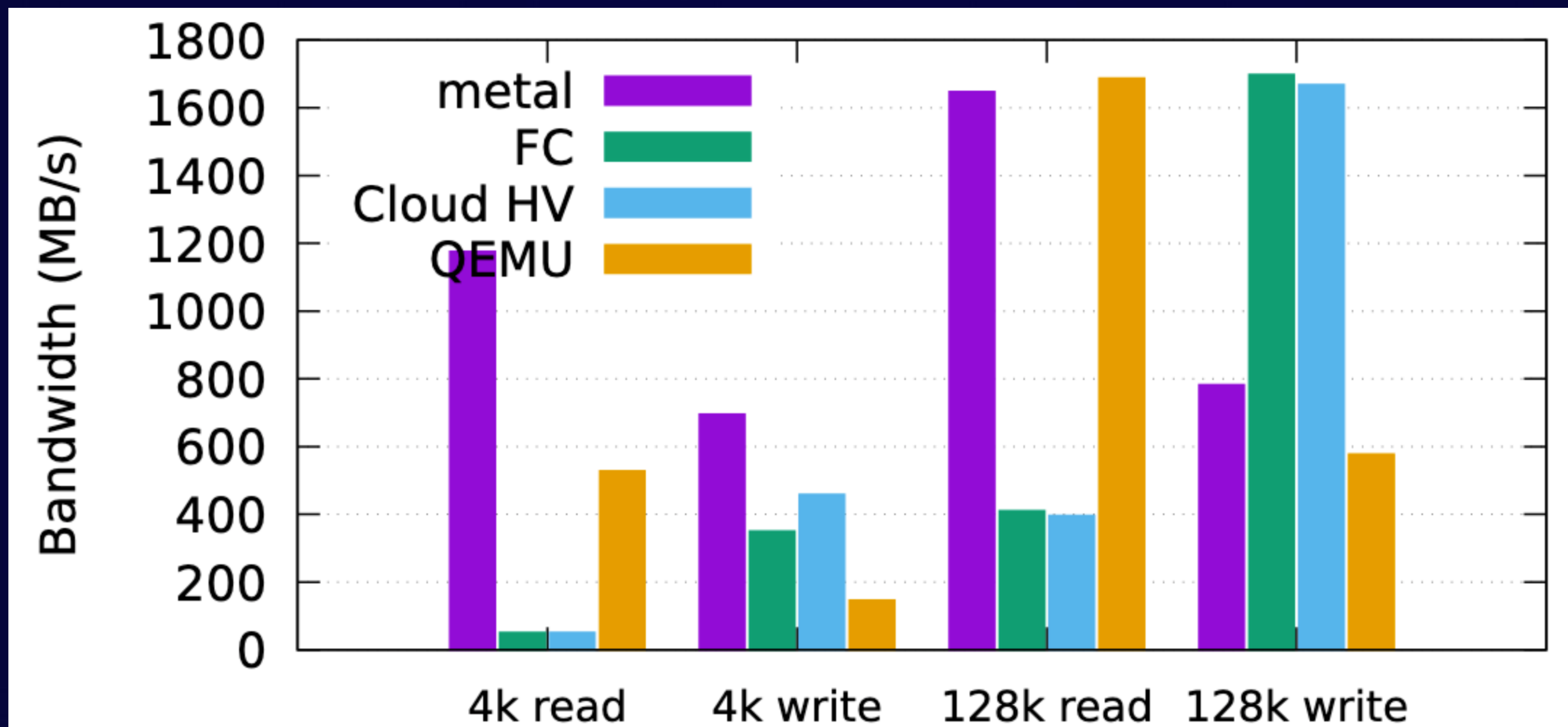
来源: <https://www.pramodb.com/index.php/2021/03/03/notes-on-the-aws-firecracker-paper/>



来源: <https://www.pramodb.com/index.php/2021/03/03/notes-on-the-aws-firecracker-paper/>

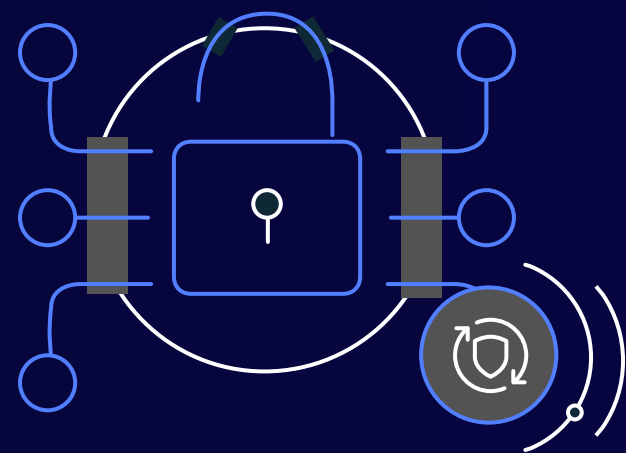


来源: <https://www.pramodb.com/index.php/2021/03/03/notes-on-the-aws-firecracker-paper/>



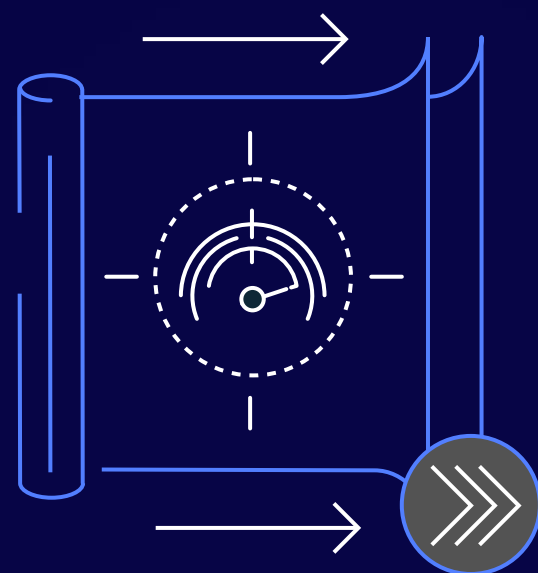
来源: <https://www.pramodb.com/index.php/2021/03/03/notes-on-the-aws-firecracker-paper/>

专为 Serverless 计算而构建的开源虚拟化技术



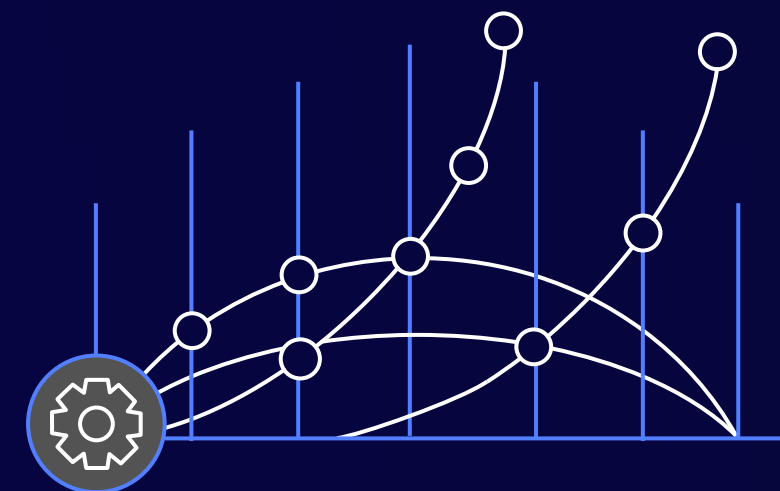
安全隔离

深度防御
最小的设备模型
速率限制
默认的安全性



因简化而快速

microVM 启动时间: 4 ms
引导进入 guest Linux user space: 125 ms



高效扩展

Firecracker 的内存开销 < 5 MiB
在每台主机上运行数以千计的 microVMs

04

经验

LESSONS
LEARNED

经验一：兼容性是硬道理

- 仅仅是禁用了超线程就发现了 *Apache Commons HTTP Client* 中的两个错误，以及自身代码中的一个错误
- 重新实现操作系统的组件会把事情变糟
- 性能兼容性也要考虑！

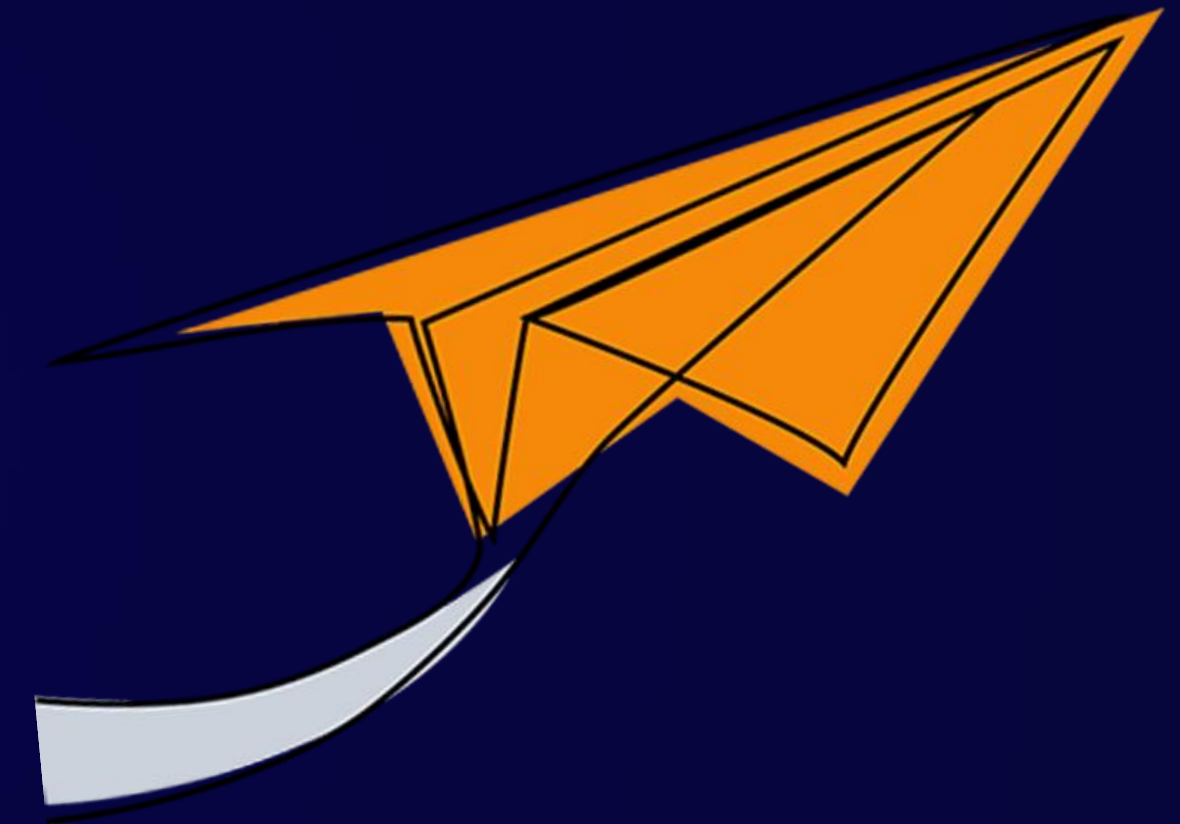
经验二：不可变的、有运行时间限制的机器

- 常见的系统管理工具，如 *RPM* 和 *DPKG* 都具有非确定性
- 限制机群的最大生命周期有助于良好的运营

经验三：要解决的问题永无尽头

- 不断变化的客户需求意味着总是需要改进

05 未来



Firecracker
项目启动

2017 年 10 月

Firecracker
开源

re:Invent 2018

rust-vmm

2018 年 12 月



- IO 性能和可扩展性 (offload)
- 调度, 特别是针对尾部延迟
- 正确性证明
- 像快照、ballooning 等功能
- rust-vmm, 以及容器社区



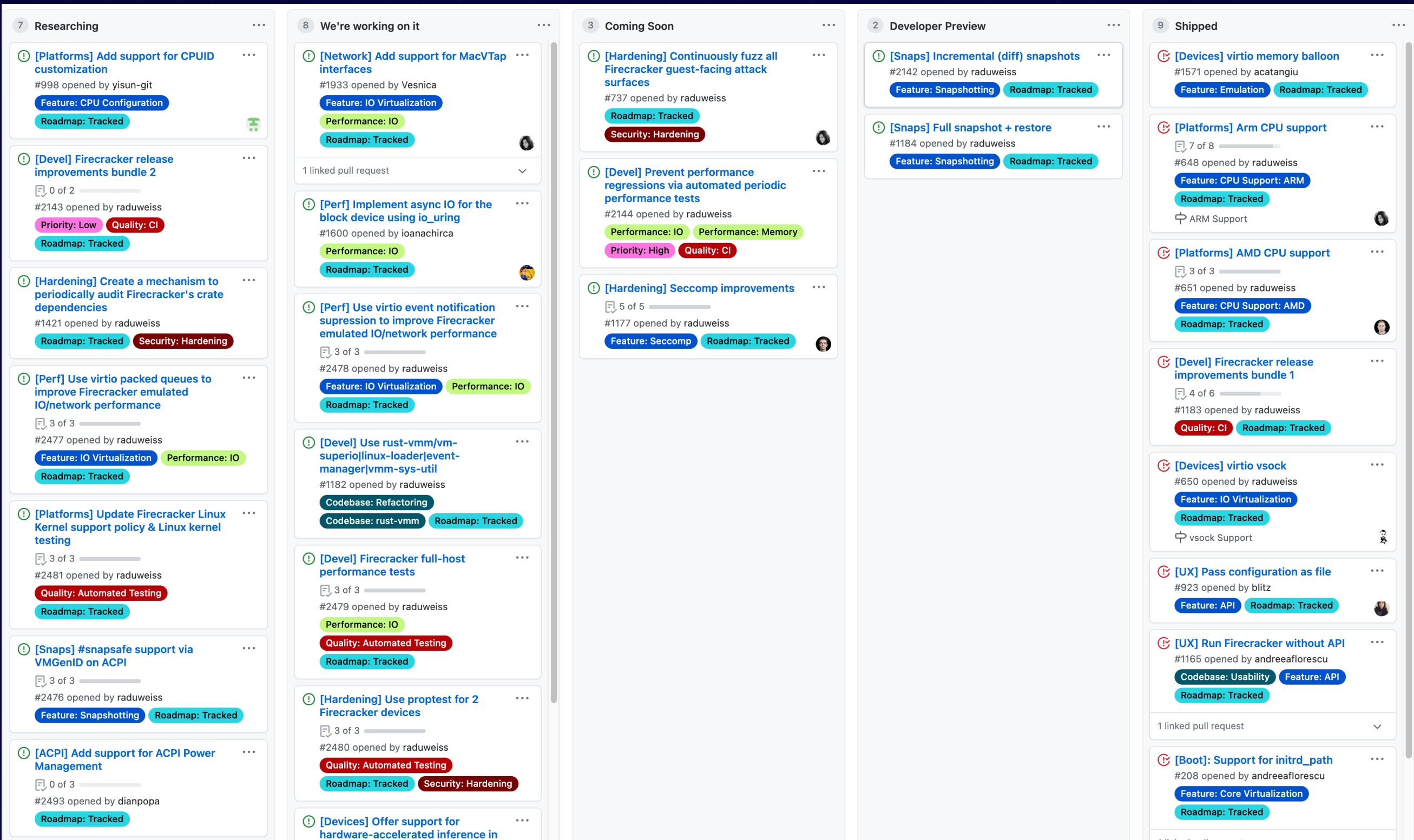
Firecracker 路线图

[TECHNO]

TVP

Tencent Cloud
Valuable Professional

/serverless/DAYS



来源: <https://github.com/firecracker-microvm/firecracker/projects/13>

Techo * TVP 开发者峰会

/serverless/DAYS China 2021

无服务器, 大有未来
Serverless, Empower More

THANK YOU!

感谢聆听!