EPPF

XDP

CPU中断 内存拷贝 等

不需要的包 会在之后被丢弃 浪费了效率

在开始就进行筛查 XDP

微服务on LINUX 用EPPF进行解决

虚拟化/分布式系统

存储/计算

存储：分布式文件系统 半结构化的存储系统 键字对形式 一致性哈希 给各个服务器分配任务

分布式计算 流式系统/批处理

虚拟化

Linux内核 容器功能 Docker on Linux kernel

Unikernel 无法共享资源

WebAssembly

设计操作系统/虚拟机/Unikernel…

Unikernel+WebAssembly

Unikernel

限制：单进程 多线程（可以将多个过程代码修改为使用多个线程/利用机器间通信 将不同进程加载在不同的Unikernel中）

单用户

有限调试

缺少足够数量的库的支持

适合使用Unikernel的情况：亚秒级启动时间、瞬时微服务、需要安全性、需要扩展到很高的数量

未来的应用：云上瞬时响应 和容器的结合

WebAssembly

WebAssembly与Host不能直接通信 由此也保证了其安全性

具有很强的可移植性 但需要glue code

因为解码更快 比JavaScript的性能更好

在非浏览器中的应用：

安全、简单、可使用多种语言、避免学习新语言的代价

区块链、云计算

缺少：

更多数据类型/垃圾收集/异常处理（中断）

分布式

去中心化/移动式…

数据在局域网内的共享 现有的协议

移动云计算

跨平台

嵌入式 虚拟机监控程序 在手机上运行

通过浏览器

智能设备算力有限 大部分在云端执行

手机进行边缘计算

小型数据中心 无线连接

跨平台 使用浏览器的计算

延迟低 AR/交互式游戏等

分布式存储

文件 存储方式 分成很多块 存在磁盘中

类存储 metadata 主服务器上存储 并行读取 Google File system 对主服务器带宽要求较大

哈希值 分散 很难增减

分布式计算

同步/异步

Net Reduce 硬盘

机器学习 速度较慢

Spark 内存存储数据

异步减少等待时间 运行精度误差

Unikernel 调试