

课程报告

一、论文总结

1.1 LegoOS

数据中心希望将CPU、内存、存储等硬件资源解耦，以提升利用率和故障隔离。

LegoOS提出“splitkernel”架构，将操作系统功能按资源类型拆分为不同的monitor，仅通过网络协同；对上层提供一个虚拟的分布式节点，使应用像在单机上运行。

远程内存访问通过RDMA与两级分布式内存管理实现；在CPU侧将cache视为按虚拟地址访问的虚拟cache，并增设小规模本地DRAM作为缓存以掩盖远程访问时延；故障按组件维度处理。

1.2 DBOS

现代的资源规模，让操作系统的资源管理变成了大数据问题。

DBOS将“操作系统状态”统一建模为数据库表，通过SQL进行访问和更新。在微内核与操作系统服务之间增加了一层分布式内存事务DBMS，操作系统服务统一用SQL + UDF操作状态表。

1.3 FlexOS

传统的操作系统在设计期固定了安全与隔离策略，难以适配不同的应用、硬件。

FlexOS基于LibOS，将组件的调用和数据的共享在代码层标注，在构建阶段选择不同的隔离机制，编写脚本自动替换源码，实现一份代码多种安全配置。并提供一个方法，自动推断出“给定性能预算下的最安全配置方式”。

二、个人理解

1. 操作系统会被拆得越来越“散”

- 硬件层面，单机能容纳的算力、内存规模有物理和成本的上限，而我们的需求仍在不断上升。因此通过高速的网络或总线，把分散的资源汇聚成资源池，以分布式的方式横向扩展，是必然趋势。LegoOS的工作说明，操作系统可以把这种“先解耦、后组合”的资源观念内建到内核层面，而不是依赖应用层调度。
- 操作系统自身层面，Linux单内核越做越大，带来了维护和安全上的压力。自然会想到将内核解耦拆分，因此有了LibOS。得益于解耦的模块，FlexOS才能把不同组件放在不同的安全隔间实现安全隔离。

获取还可以把操作系统的组件库放在更远的分布式节点上，通过网络连接各个内核库的调用，运行时按用户需求弹性拼装各个模块。或许是一个数据中心构建可扩展的大型弹性虚拟机的思路？

2. 操作系统的决策将从“设计时”后移

- 传统的OS需要在设计阶段预设好内核结构、调度策略等，力求让一个系统尽可能适配多样的需求。但是在未来，关键的决策时间可以慢慢“往后挪”。可以根据具体的需求，如应用是什么、负载有多大，在运行时随时决定需要调配多少资源、采用什么调度策略等。操作系统的构建还将越来越智能，如FlexOS的偏序搜索算法暴力枚举，AI4OS在这个地方应该很有发挥空间。

三、区块链+操作系统

我的研究领域的大背景是区块链。当前区块链系统的现实是：节点仍作为普通应用运行在通用操作系统（主要是 Linux）之上，共识、智能合约执行、状态存储、P2P 通信都由应用层自行实现，操作系统对这些区块链特有的需求几乎无感。

区块链对网络的需求非常大，如果能让内核直接暴露高性能网络服务给节点进程，减少多次拷贝带来的开销；同时让这台机器只围着“区块同步 + 共识 + 合约执行 + 状态持久化”这条链路设计，相当于“操作系统即区块链节点”。在此基础上，还可以把区块链节点内部本来就分层的功能做成可选的系统级模块，例如将合约执行或密钥管理模块放入 TEE 等强隔离环境，由操作系统调度，而不是让应用自己去隔离。

反过来，区块链能否服务于操作系统？如果未来出现“分散的、网络化的操作系统”，不同资源节点之间也会遇到信任问题，那么区块链本身也可以作为底层可信服务，为这些组件提供状态共识和资源使用的不可抵赖记录。