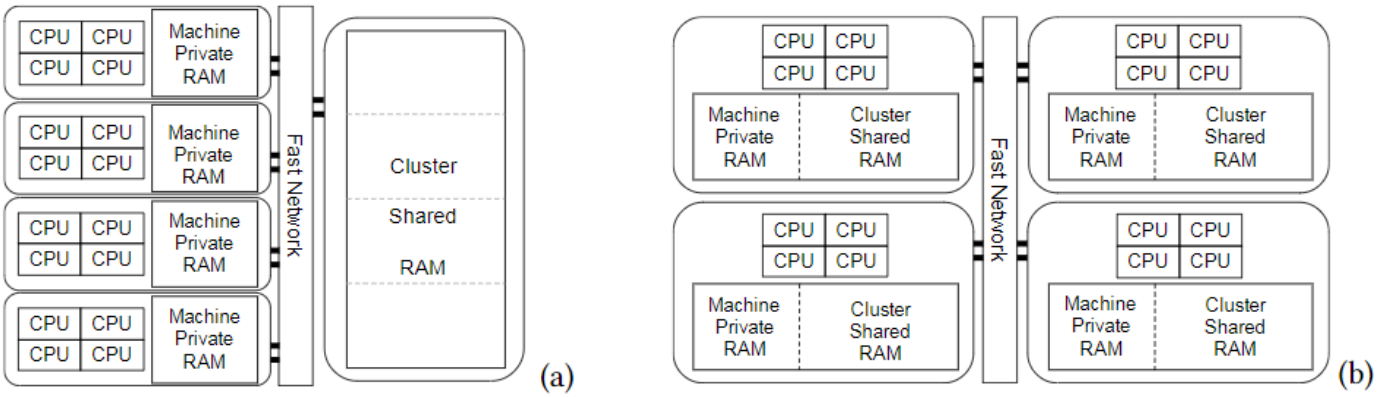
LegoOS: A Disseminated, Distributed OS for Hardware Resource Disaggregation

这篇工作发表于2018年的OSDI开创性的提出了**针对资源分解的数据中心**的操作系统内核架构Splitkernel，并实现了原型LegoOS。LegoOS在多种硬件节点组成的分布式异构系统上实现了软硬件结合的进程、内存、存储管理与故障处理，并实现了兼容现有应用的系统调用。

- 背景问题与挑战
  - 传统单体服务器带来的资源利用率低、弹性差、粗故障粒度等问题，需要资源池化。
  - 传统OS对新设备（FPGA GPU）的支持比较差。
  - 在数据中心进行资源分解非常流行，但是传统的单体OS只能管理当前机器的硬件资源，无法操作池化的硬件。
- 主要工作

这篇工作提出了LegoOS，几乎是第一个提出：**针对资源分解的数据中心设计一套操作系统**的工作。LegoOS的工作量非常庞大，实现了一整套分布式操作系统。

LegoOS建立在硬件层面的资源分解上（即在硬件层面上已经完成了资源的分解，如下图a所示，图b是软件层面上的资源分解）。在硬件上已经异化成了“内存节点/组件”、“计算节点/组件”，而非传统的单体服务器组成的集群。内存节点上几乎没有运算能力，只能在FPGA等轻量级处理器上运行最简单的内存access、alloc、translate等操作。计算节点则几乎没有存储，只有较小的缓存ExCache。



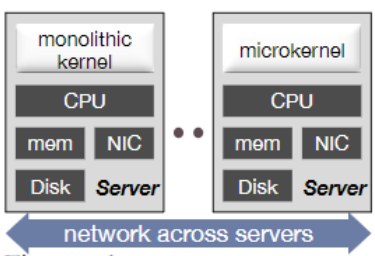


Figure 1: OSes Designed for Monolithic Servers.

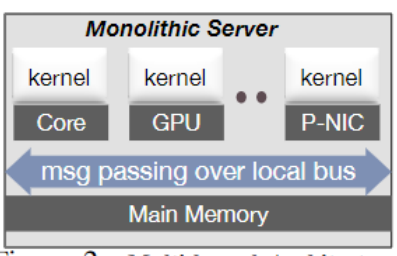


Figure 2: Multi-kernel Architecture. *P-NIC: programmable NIC.*

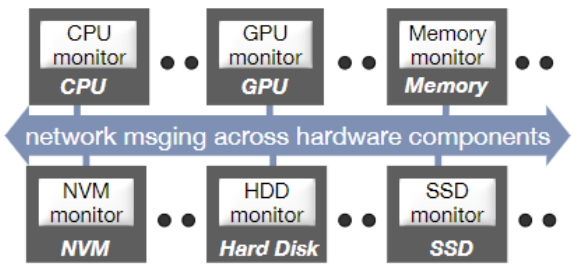


Figure 3: Splitkernel Architecture.

LegoOS提出了Splitkernel架构，相比于单机上的Multi-kernel（内核之间基于本地总线通信）而言，Splitkernel将完整内核拆分成对不同资源的Monitor，再发布到集群中各资源节点上去，这些Monitor基本上运行在FPGA等低功耗处理器上。Splitkernel基于RDMA（高速网络）来在不同的内核模块之间进行通信。

LegoOS中的硬件结构如下图，值得注意的是，在计算节点上提供了一个ExCache作为缓存以提高性能并避免cold miss。在内存上的Mem Ctrl是一个运行在FPGA上的服务。

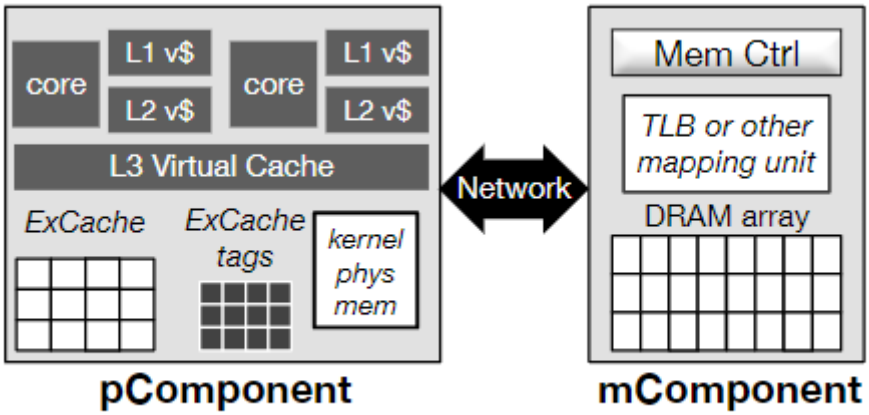


Figure 5: **LegoOS pComponent and mComponent Architecture.**

LegoOS中的进程管理与现有OS基本相同，需要注意的是，LegoOS没有实现中断，让一个进程以无调度、抢占的方式运行到结束。

LegoOS中提出了一种两级式的内存管理机制。内存节点分为主节点和普通节点，主节点分配（选择节点来提供）vRegions给用户进程（粗粒度），普通节点（vRegions所处的节点）对该vRegions执行（细粒度）管理，如内存分配等。这个设计看起来很复杂(为什么不直接让GMM来选择vRegions节点？)，因为GMM存在集中式的问题，将分配下放到主节点（只是针对一个进程而言，其实可以是任何一个节点）可以大大提高容错和处理效率。

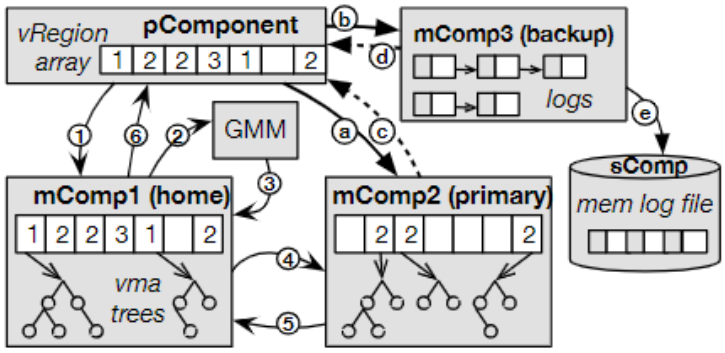


Figure 6: **Distributed Memory Management.**

LegoOS只对内存组件的容错进行了讨论，使用主备节点的方式（备节点增量存储日志）。

LegoOS实现了一系列Linux的syscall使得能够适配现有工作负载。

- 文章限制与思考
  - 硬件层面上的资源分解带来了很多限制，如在内存节点上只能运行最简单的逻辑（因为计算资源非常有限），限制了可以对内存的操作，复杂的逻辑需要上移到计算节点。

    - 在FPGA上能运行的控制器逻辑能复杂到什么程度？上面的程序和unikernel相比量级如何？能否用unikernel等技术来扩展在FPGA上的逻辑
    - 请求有控制请求和读写请求，对这两种请求能否衍生开来做？
- 由于处理能力有限，LegoOS的设计中不提供分布式系统中的一致性保证，需要应用来使用**消息传递**来实现跨组件的数据的一致性。
    - 分布式系统中的一致性、容错、选主等等主题都可以迁移过来做
- 不追求CPU的利用率，即不考虑中断，处理器会等待网络通信，可能会造成性能降低，这个问题其实也是RDMA造成的通信效率问题。
    - 通信上的问题，用CXL来做通信
- 没有在系统中实现对异构硬件的支持，只有一些描述。
    - 支持不同Splitkernel的shim层，来接管不同硬件设备
- 没有提到如何在这种分布式操作系统上进行虚拟化，目前是抽象成了一个高性能单机，不知道现有的hypervisor能不能直接在上面运行起来。
    - 运行在其上的进程的安全、隔离问题