开放原子开源基金会
OPENATOM FOUNDATION

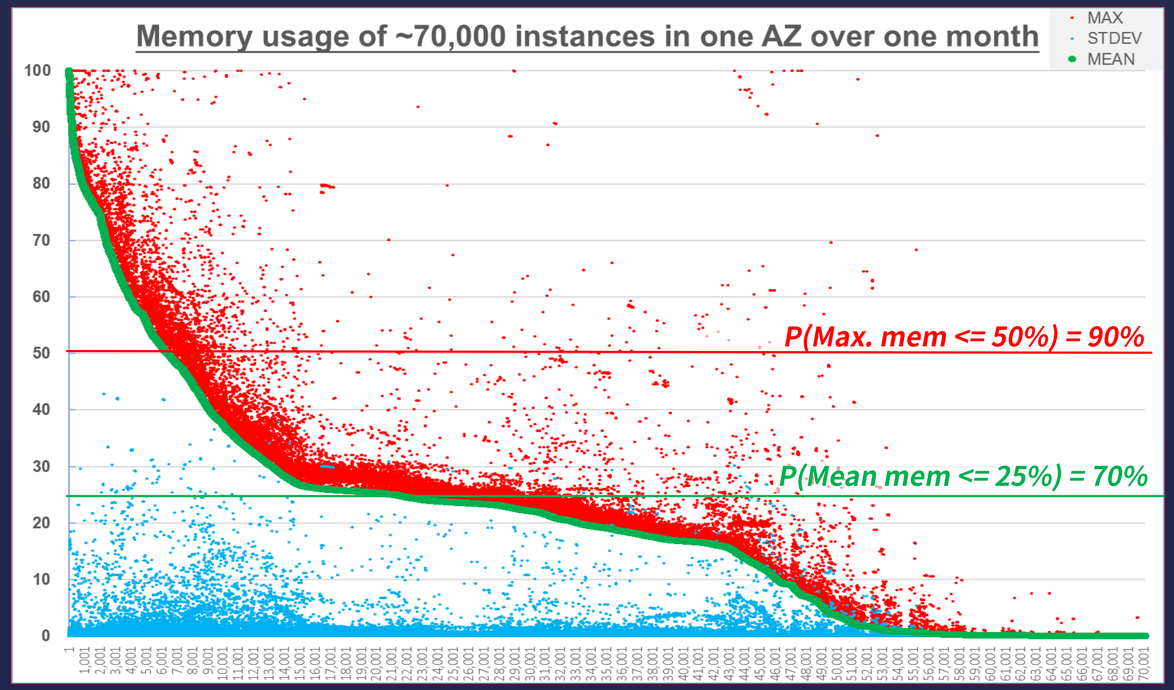
OpenEuler

灵衢内存池化关键技术与应用

演讲人
高超

演讲人职位
openEuler社区sig-UnifiedBus Committer

“内存墙”：算力规模化的终极瓶颈



任意时间内存利用率
90%虚拟机<50%

平均内存利用率
70%虚拟机<25%

“内存孤岛”：**资源无法跨节点访问**，导致资源利用率低（虚拟化场景30%-50%内存未被使用）、系统扩展性差

关键应用在集群内，资源多副本，额外资源消耗和同步开销

灵衢（UnifiedBus）互联协议与超节点架构，作为解决上述痛点的创新路径

跨越边界：灵衢互联协议如何重塑内存访问

ref:《灵衢基础规范2.0》

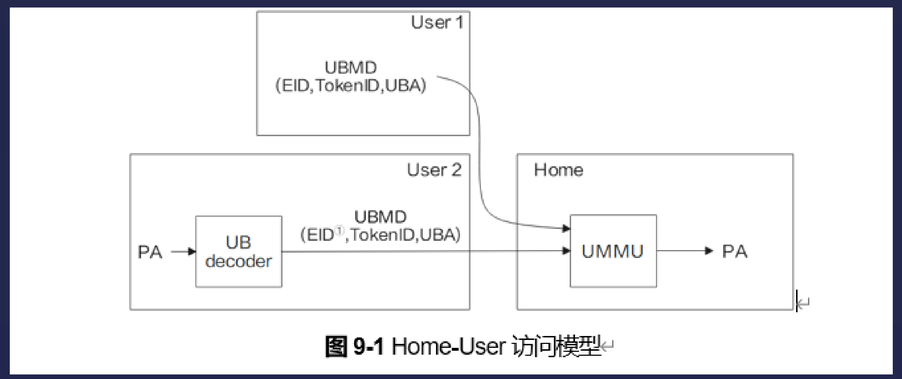


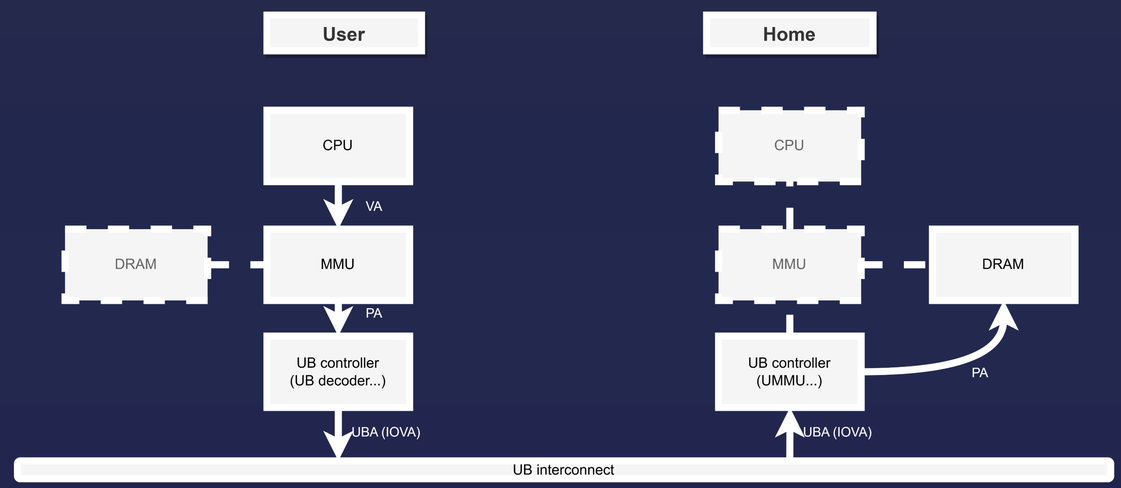
图 9-1 Home-User 访问模型

灵衢基础协议中定义的访存模型

User 向 Home 发起内存访问请求，该请求应包含 UBMD。User 发给 Home 的 UBMD 可来自于：

(1) 由 UB Decoder 基于 PA 查表得到，见 8.3 节。

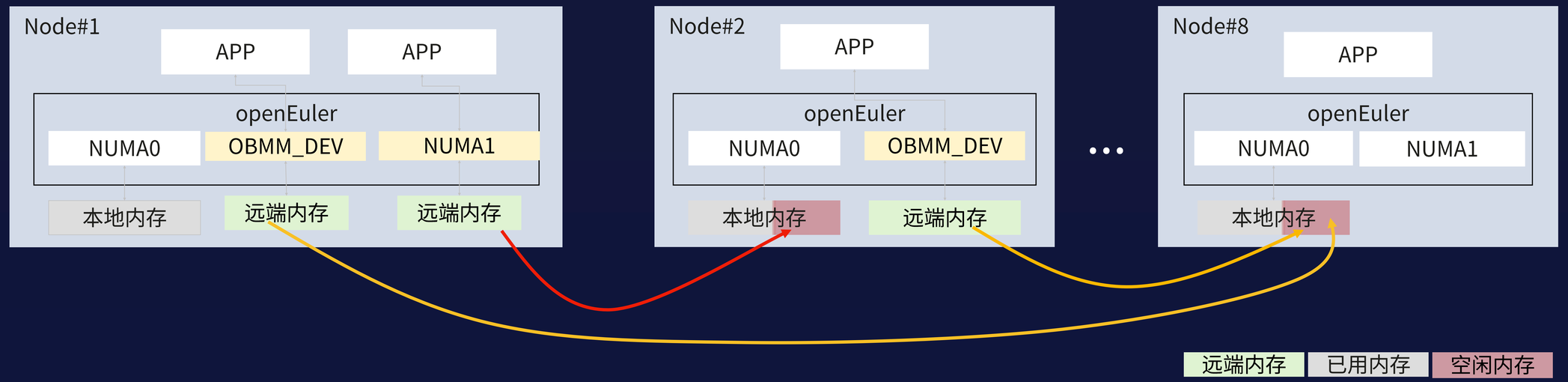
(2) 由用户通过 User 侧编程接口提供，见 8.4 节。



参照灵衢协议的物理访存路径（Load/Store）

VA->PA->UBA->PA

灵衢内存池化：可以跨节点流动的内存



Load/Store基础能力

灵衢总线定义了纯硬件实现的跨节点内存访问能力；

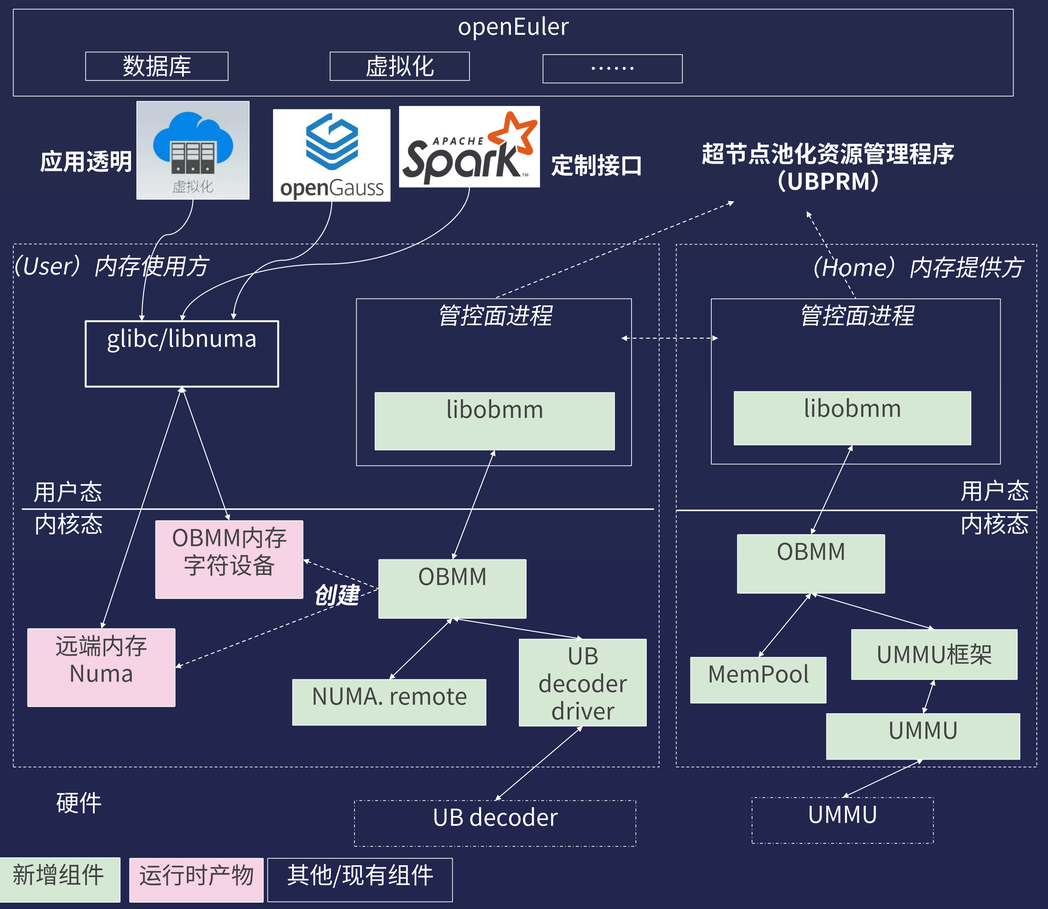
超节点内存借用

空闲资源在超节点之间动态流动，削峰填谷提高超节点资源利用率；

超节点共享内存

实现超节点内数据同步，基于Load/Store快速数据交换；

openEuler新增组件支持灵衢内存池化



管控面export/import接口

通过export/import接口完成物理内存资源在节点之间流动；

应用透明接口

透明对接当前生态，使用numa的方式管理远端内存，可以无缝使用当前的malloc, mbind, madvise, numa_alloc_onnode, move_pages等内存相关的接口；

定制接口

内存上线为类字符设备形态，通过open, mmap, unmap, close等标准接口使用，同时使用ownership接口，使用共享的方式使用远端内存；

“无”跨节点一致性：基于所有权的共享内存管理

ref:《灵衢基础规范》

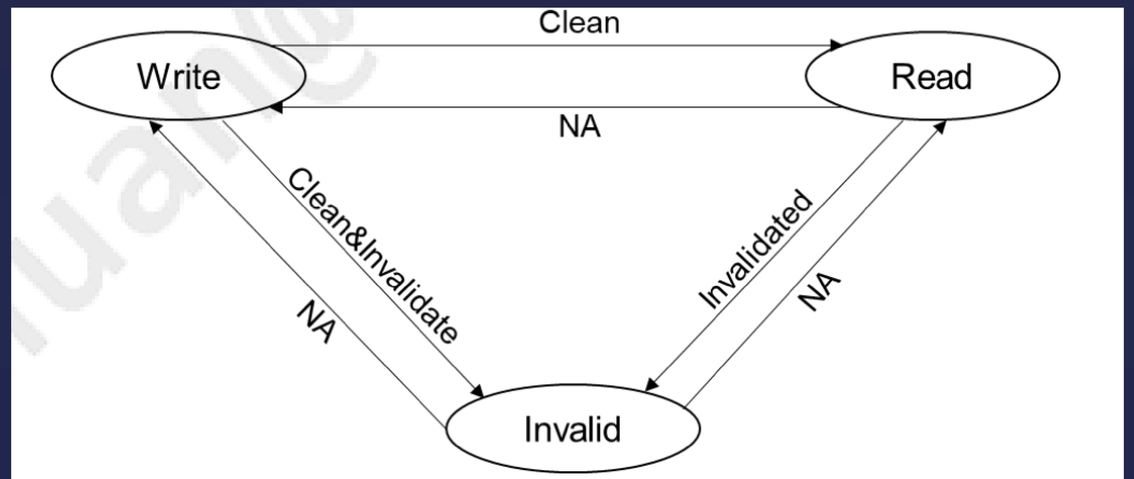


图 8-9 Ownership 状态转移图

Ownership 状态

任意粒度的内存段，在每一个节点均有本地的Ownership状态：

- Invalid：本节点不能对内存进行读写操作；
- Write：本节点可以对内存进行读或写操作；
- READ：本节点可以对内存进行只读操作；

Ownership switch接口

用户通过该接口，进行每段内存的状态维护。注意，当有一个节点为Write Ownership时，其他所有节点必须为Invalid Ownership状态。