鲲鹏服务器上优化TLBI广播提升虚拟机线性度的实践分享

陈祥

chenxiang66@hisilicon.com

开放原子开源基金会 OPENATOM FOUNDATION | OpenEuler

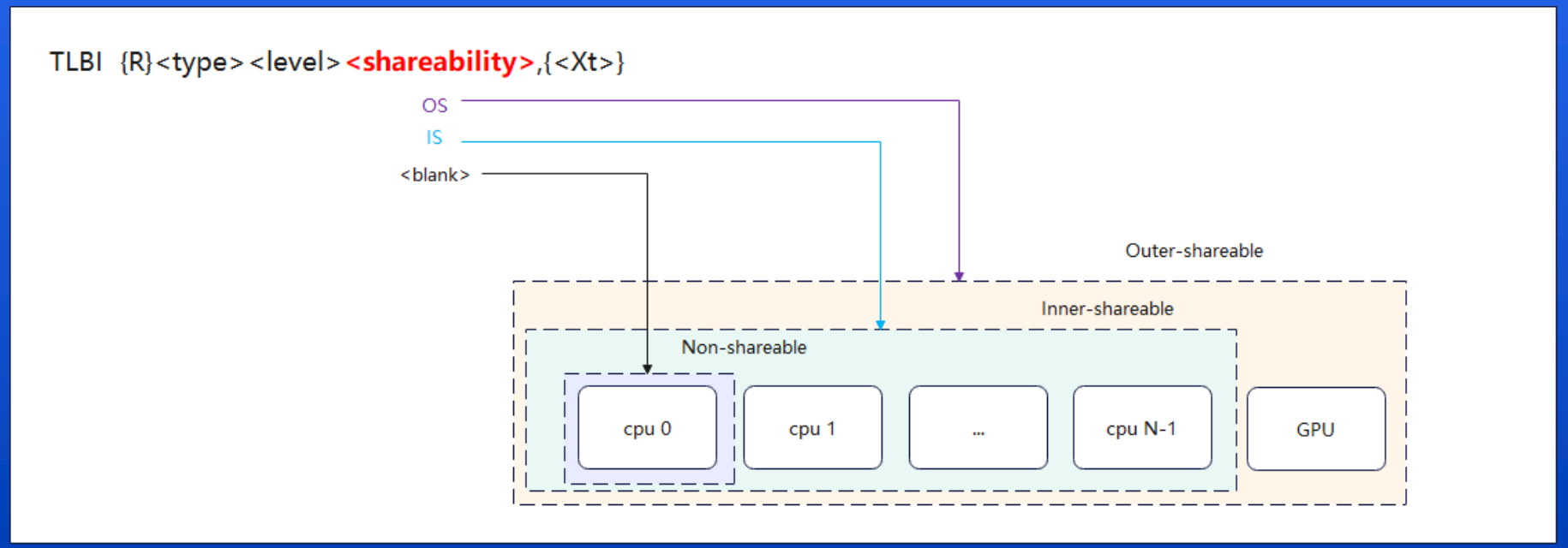
目录

TLBI广播指令介绍及问题

Linux内核中很多场景需要通过TLBI广播指令作TLB的无效化，包括：

- 页表属性变化
- 页表映射变化

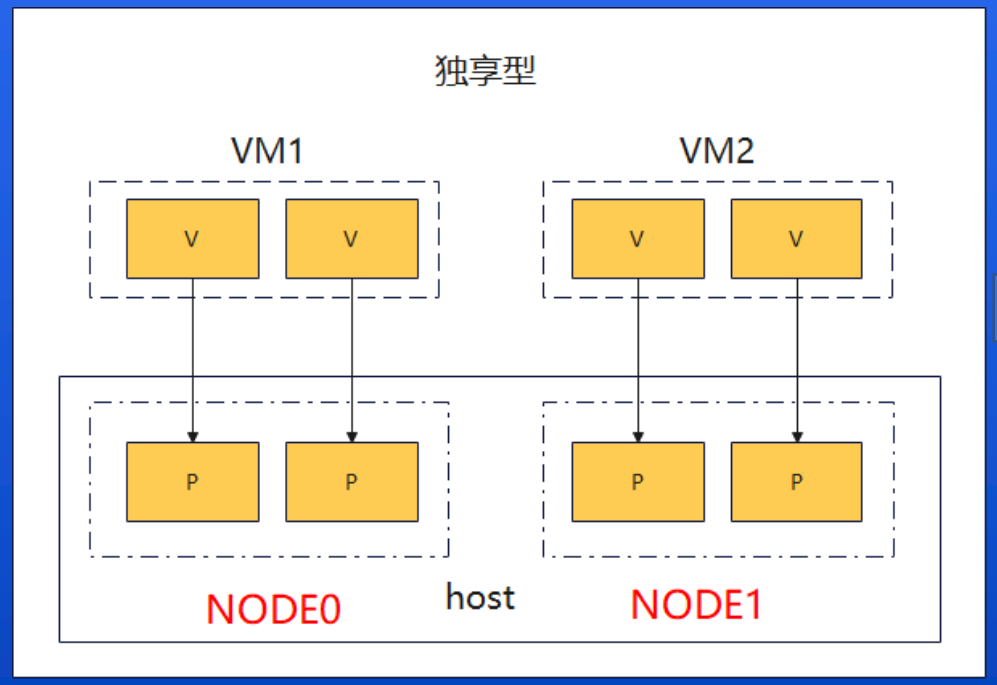
当前ARM架构中TLBI广播指令范围为IS（Inner shareable）和OS（Outer shareable），IS和OS的范围为IMPLEMENTATION DEFINED（由实现定义）。通常在实现时IS和OS范围包含全系统的CPU，即TLBI广播指令会广播到系统中所有CPU。

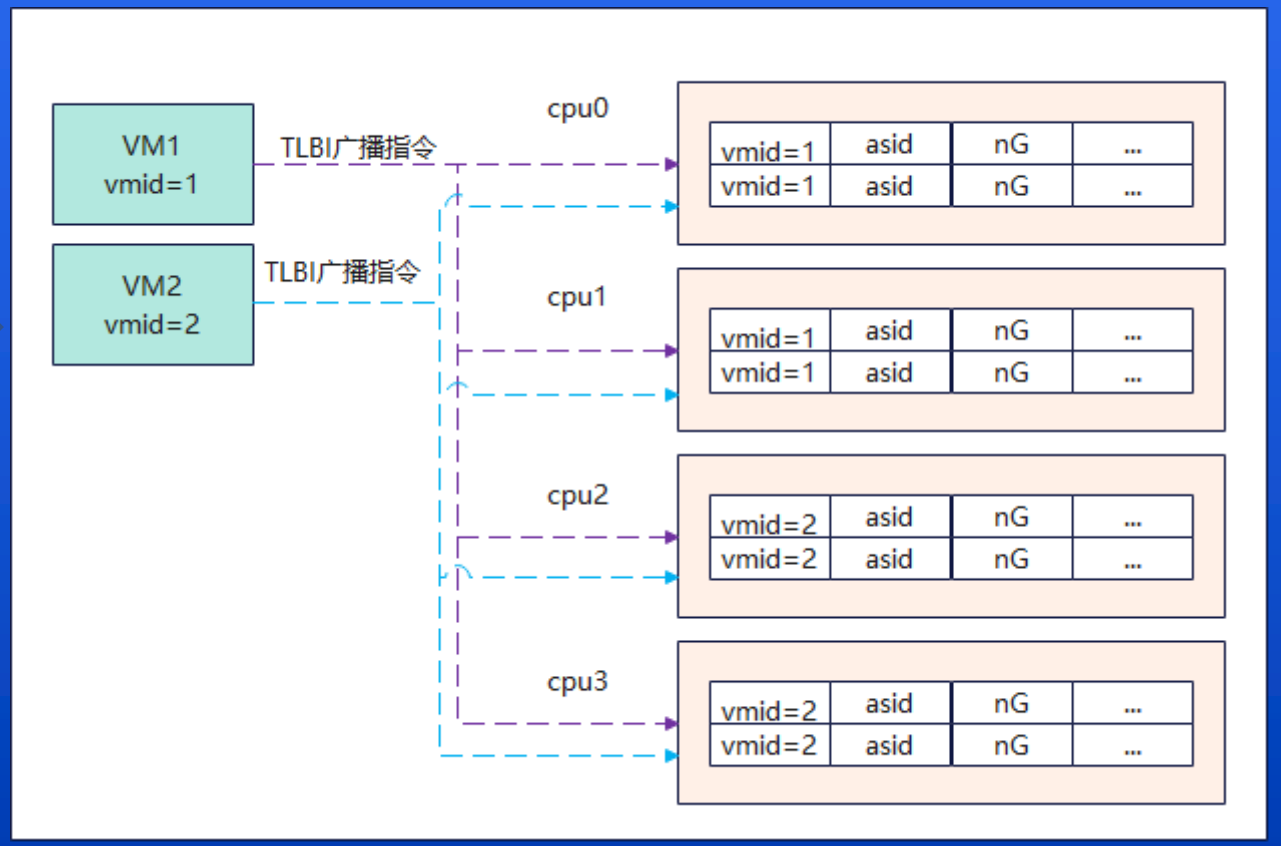


TLBI广播指令及问题（续1）

虚拟机运行场景分析

在云部署的虚拟机场景包括：独享场景和超分场景。用户通常会将虚拟机绑定在特定范围的物理CPU上，因此在这些场景中虚拟机仅运行在部分物理CPU上，只在部分物理CPU上存在对应虚拟机的TLB，但虚拟机发出的TLBI广播指令仍发往物理机所有核，这给系统性能特别是线性度造成损耗，且随着系统核数增加，损耗显著增加。





物理机：4个物理CPU

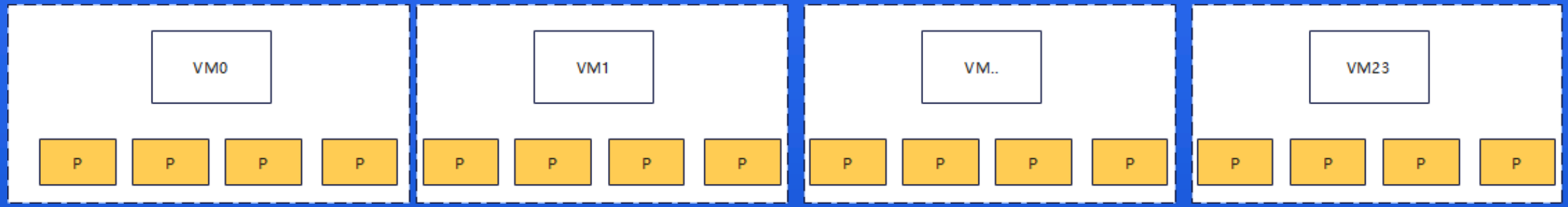
虚拟机1：2个VCPU, vmid=1

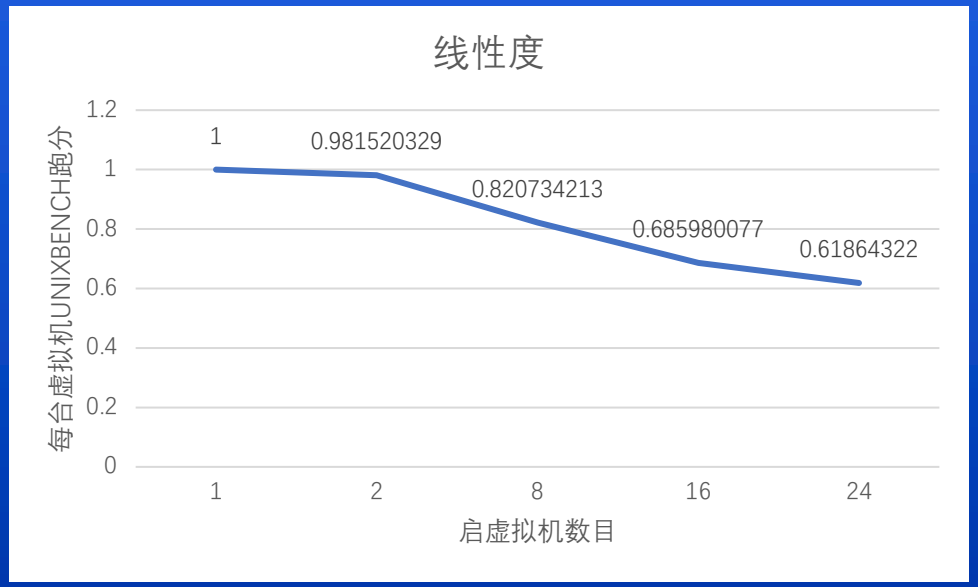
虚拟机2：2个VCPU, vmid=2

VCPU与物理CPU一一绑定

TLBI广播指令及问题（续2）

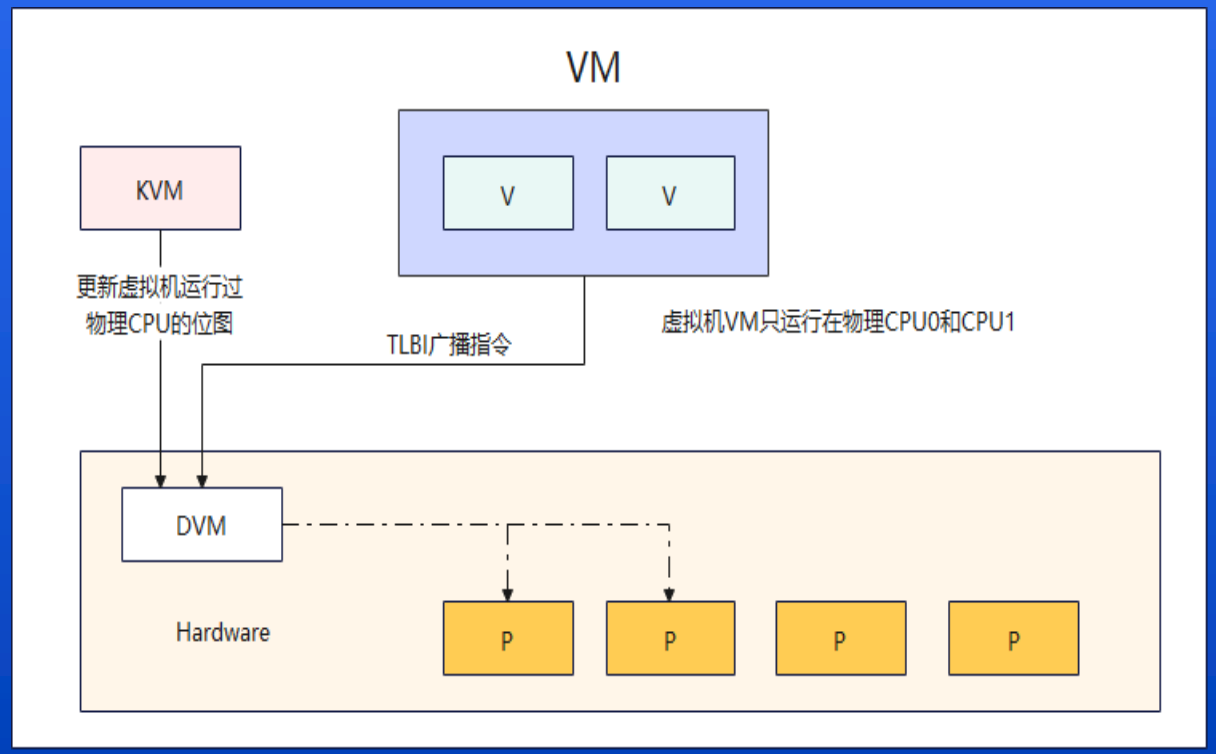
在鲲鹏服务器上，分别启1/2/4/8/16/24台相同规格的虚拟机（将虚拟机绑定到特定范围物理机上），每台虚拟机中跑Unixbench，发现线性度随着运行的虚拟机数目越多，线性度下降显著。

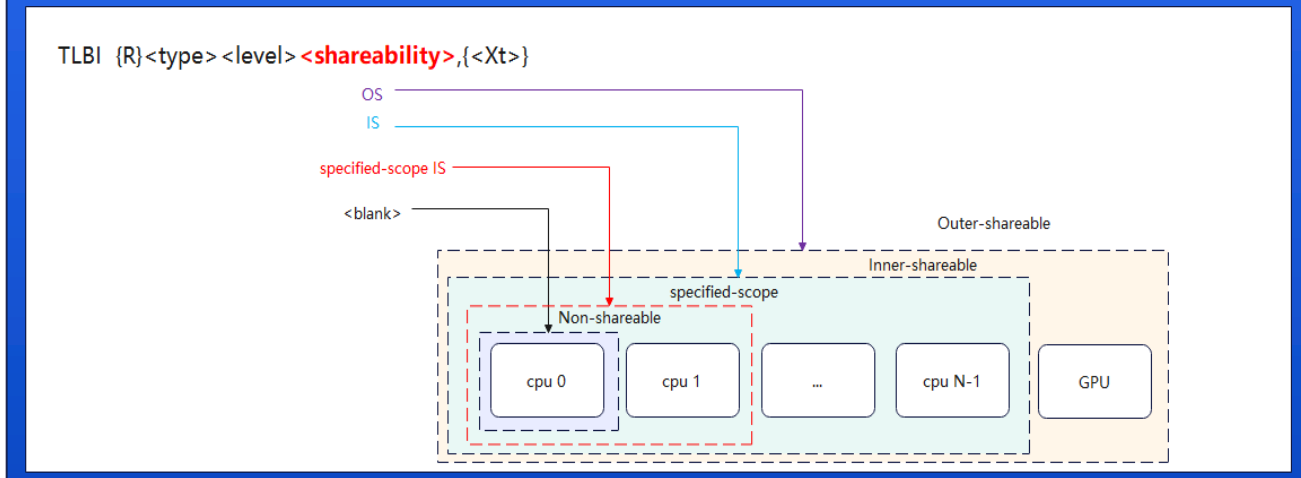




TLBI广播优化方案

为了优化上述问题，提出了TLBI广播优化方案：基于ARM架构定义的TLBI广播指令，对TLBI广播指令的广播范围作约束，广播范围不再是IS或OS所代表的全系统范围的CPU，而是虚拟机运行过的物理CPU范围。该范围由软件记录并传递给硬件，硬件通过配置的范围对DVM广播范围作限制。

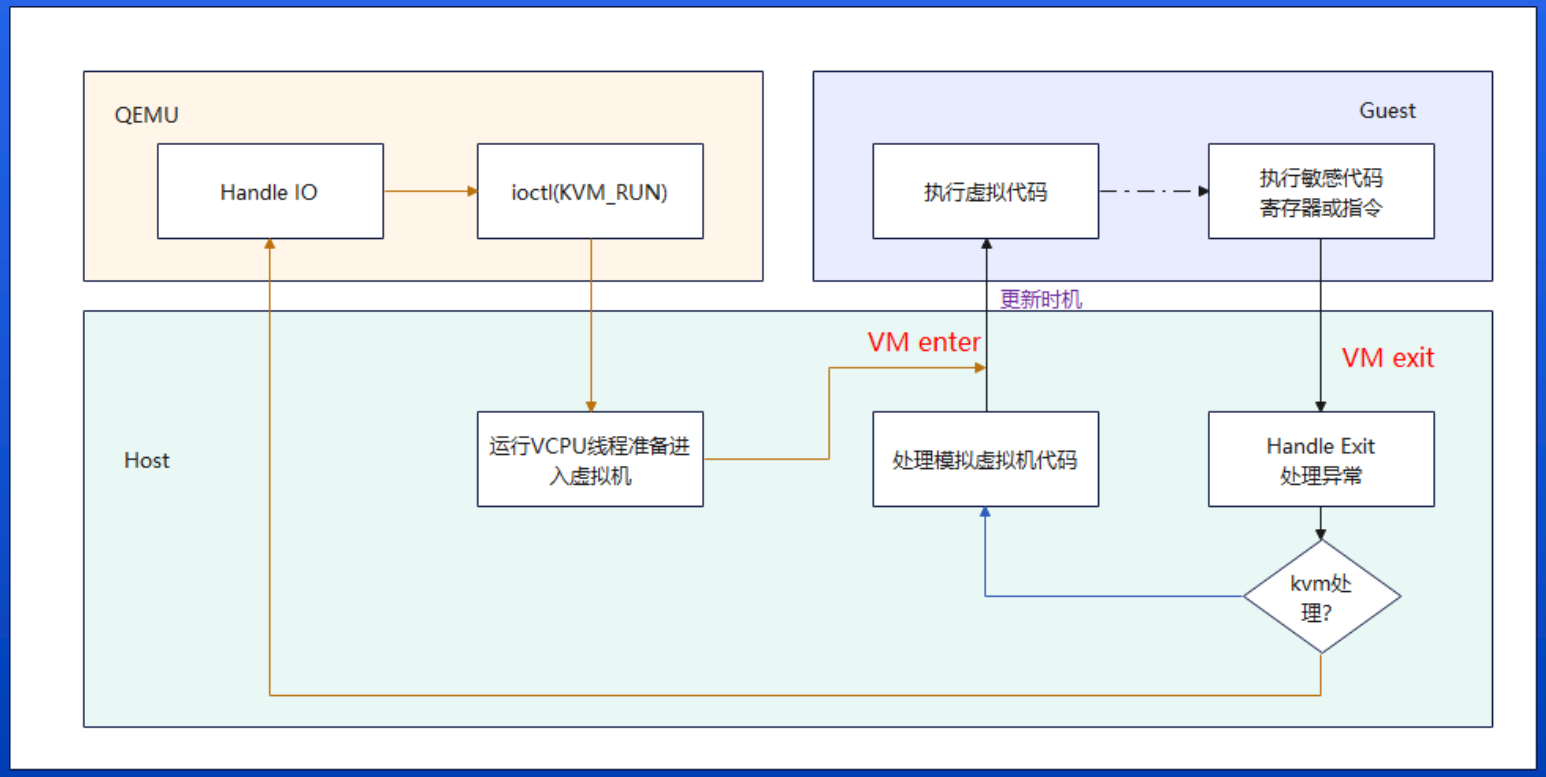




TLBI广播优化方案（续1）

软件更新时机

软件需要实时记录和更新虚拟机运行过的物理CPU范围。对于虚拟机的每个VCPU，它是HOST上的VCPU线程，它的运行视图如下图所示。在VM enter后进入虚拟机，虚拟机处于运行状态时是关调度的，因此在虚拟机中是不会改变运行物理CPU位图。只有在VM exit后退出虚拟机才有时间调度到其他物理CPU上。因此更新时机在准备进入到虚拟机时即VCPU load过程中。



软件更新过程

- VCPU load过程中检查虚拟机所运行的物理CPU位图是否发生变化，若发生变化（重新绑核），强制VCPU退出，重新加载新的位图；
- 若没有发生变化，保持之前虚拟机位图；

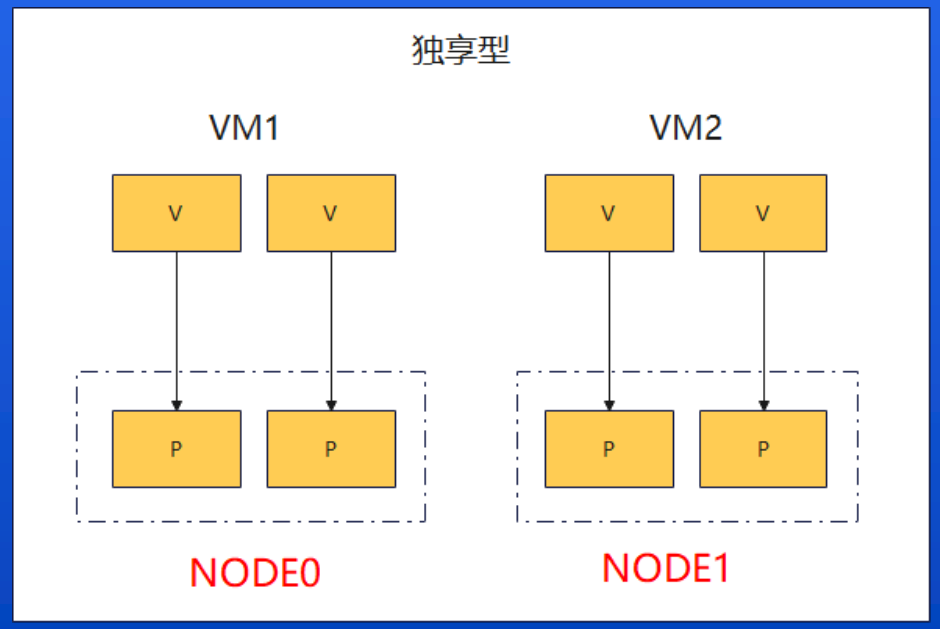
TLBI广播优化方案（续2）

物理机：4个物理CPU

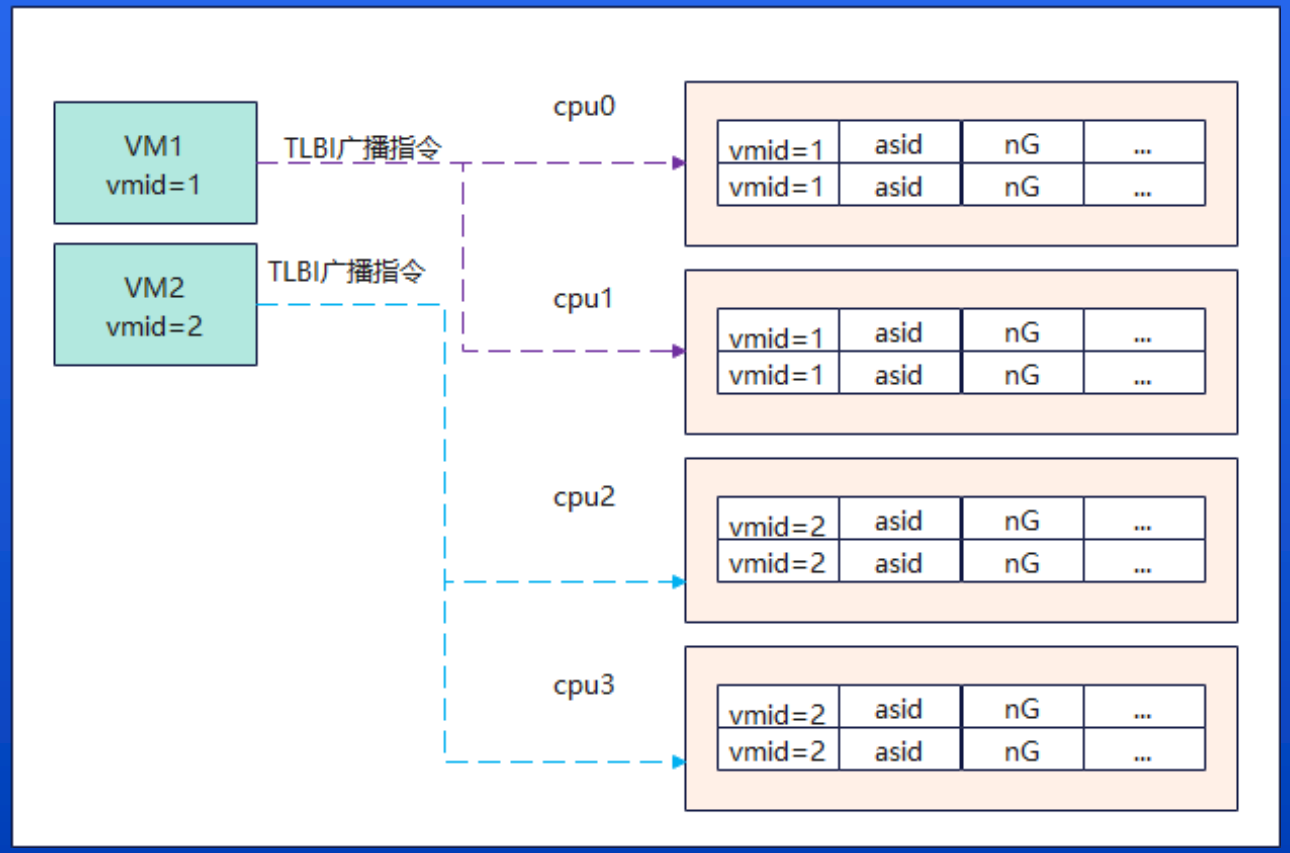
虚拟机1：2个VCPU, vmid=1

虚拟机2：2个VCPU, vmid=2

VCPU与物理CPU一一绑定

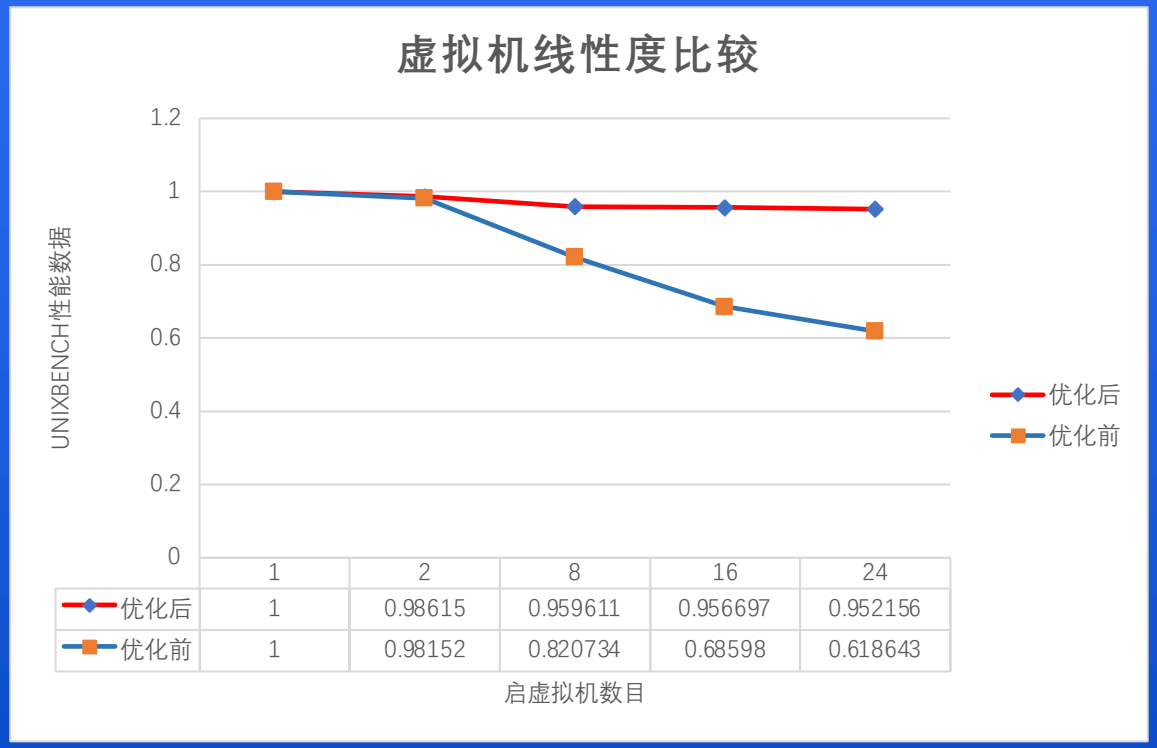


TLBI广播
优化后



优化前后性能比较

在鲲鹏服务器上，分别不使能和使能TLBI广播优化，比较两者线性度（测试过程如前所述），测试结果如下图所示。



	1	2	8	16	24
优化后	1	0.98615	0.959611	0.956697	0.952156
优化前	1	0.98152	0.820734	0.68598	0.618643

可以看出，在使能TLBI广播优化后，线性度维护在0.95以上，相比未使能情况，显著提高（24台虚拟机时提升达到50%）

TODO

- ✓ 推动特性标准化
- ✓ 考虑将该特性从VM粒度扩展到线程粒度

THANKS

THANKS

THANKS