openEuler

openRSO-解决混部应用的访存时延干扰

Resource Schedule and Orchestration

目录

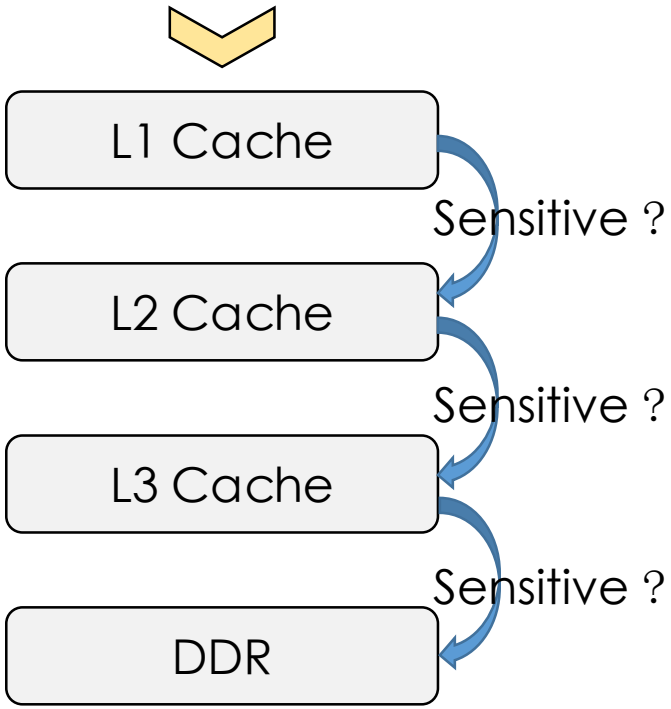
CONTENT

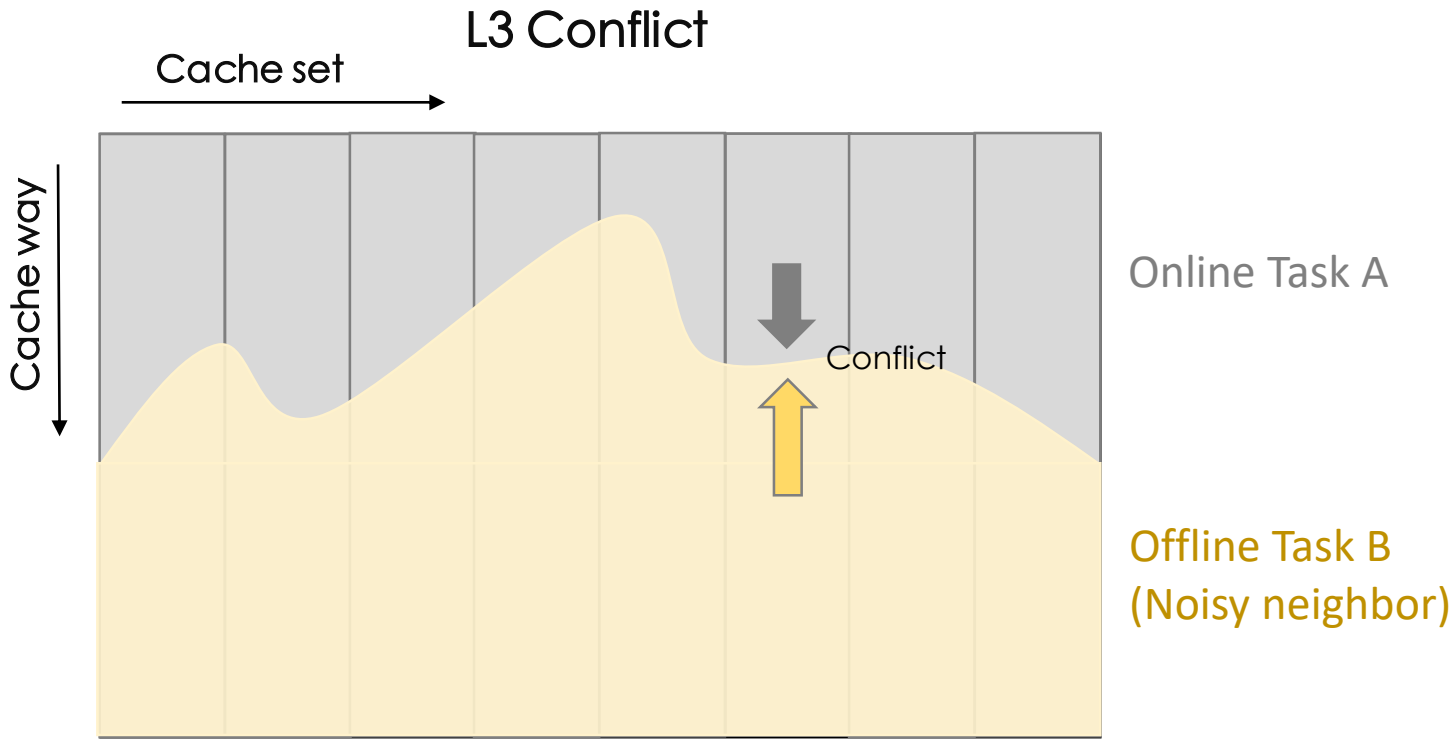
openEuler

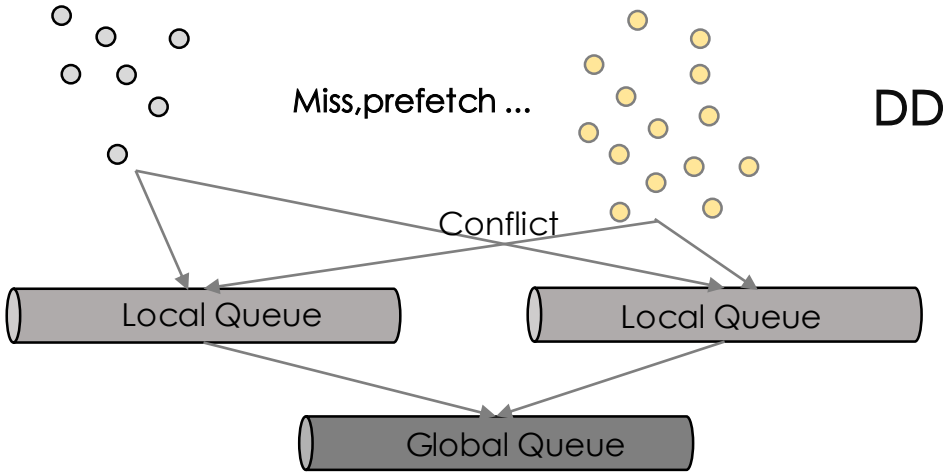
前言

- 何为混部，何为业务画像?
- 如何动态识别业务画像?
- 如何保障关键业务访存确定时延?
- 如何知道业务实际需求?

混部业务画像

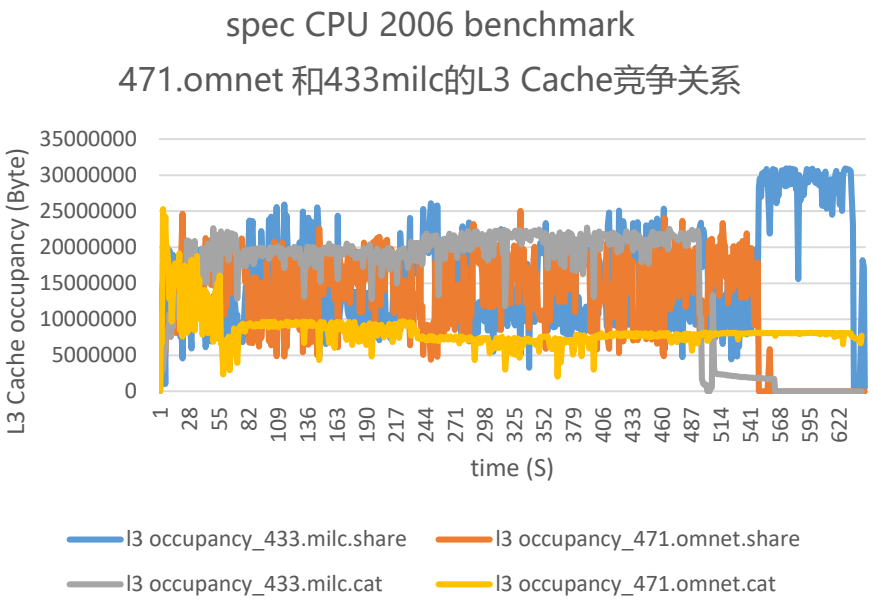




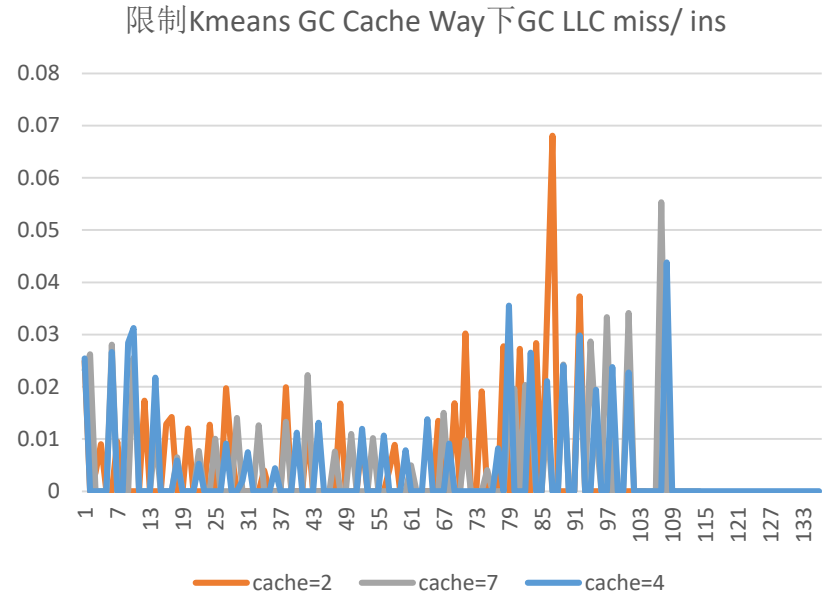


openEuler

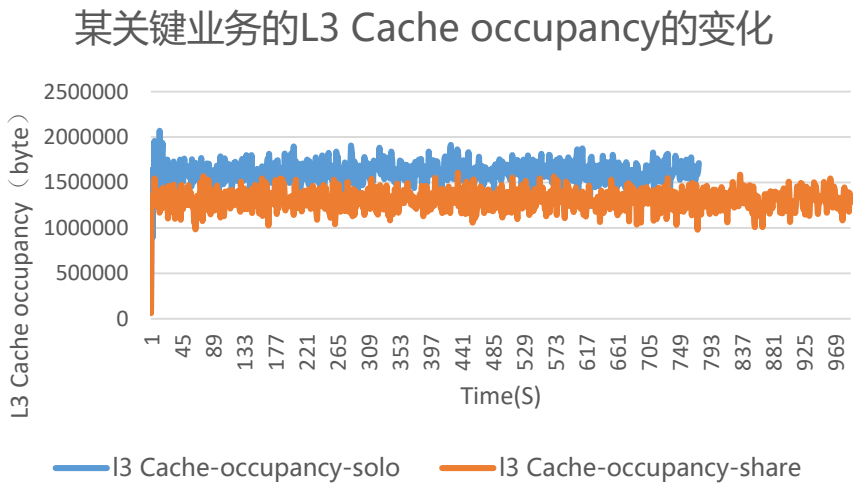
混部业务画像



计算密集型一般表现为强烈不稳定的干扰



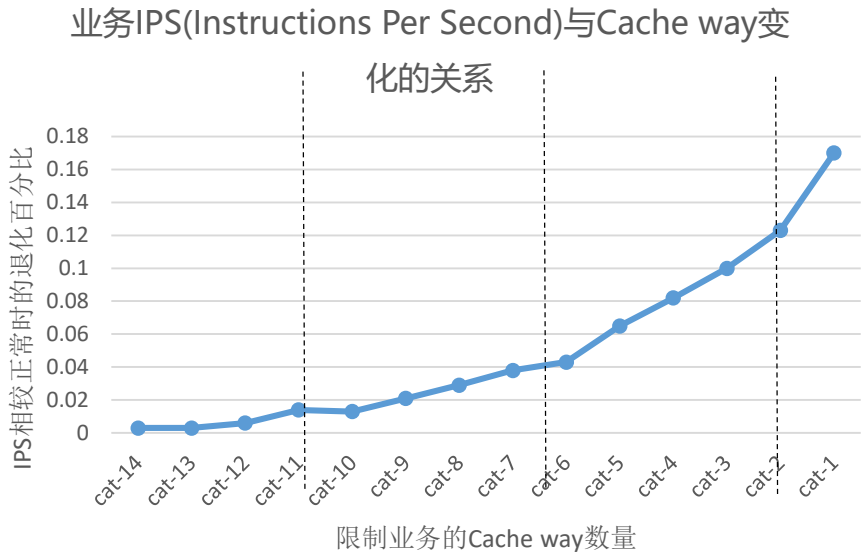
突发型业务（受）干扰有限

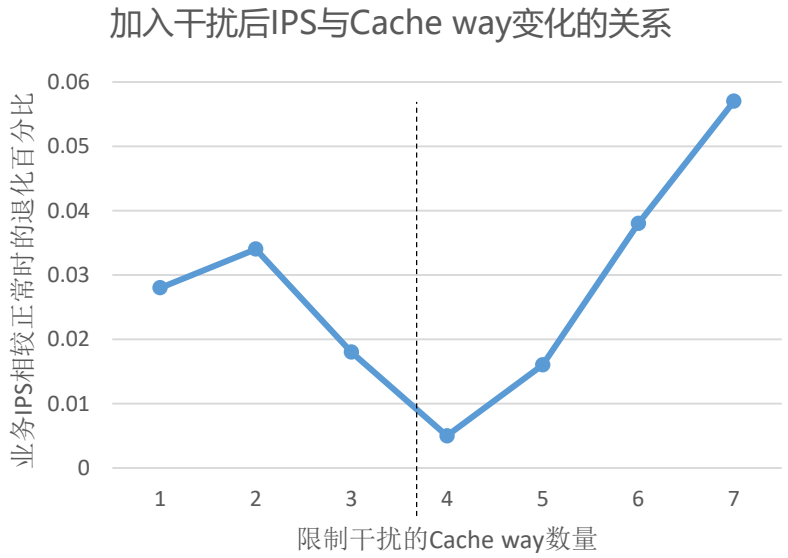


时延敏感型的业务可能有稳定的持续干扰

openEuler

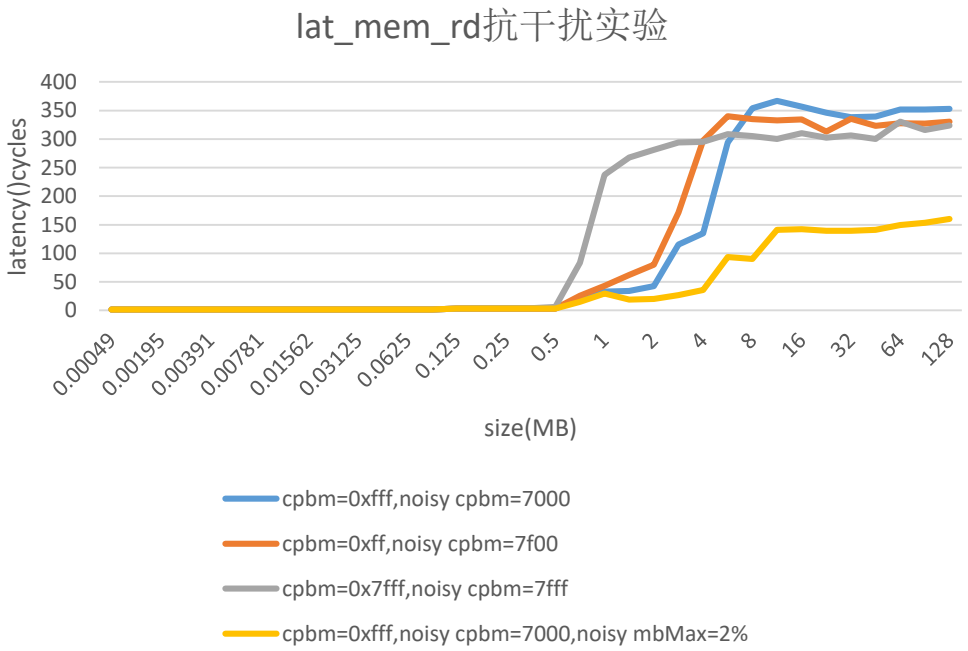
混部业务画像





动态调节资源占用需要考虑：

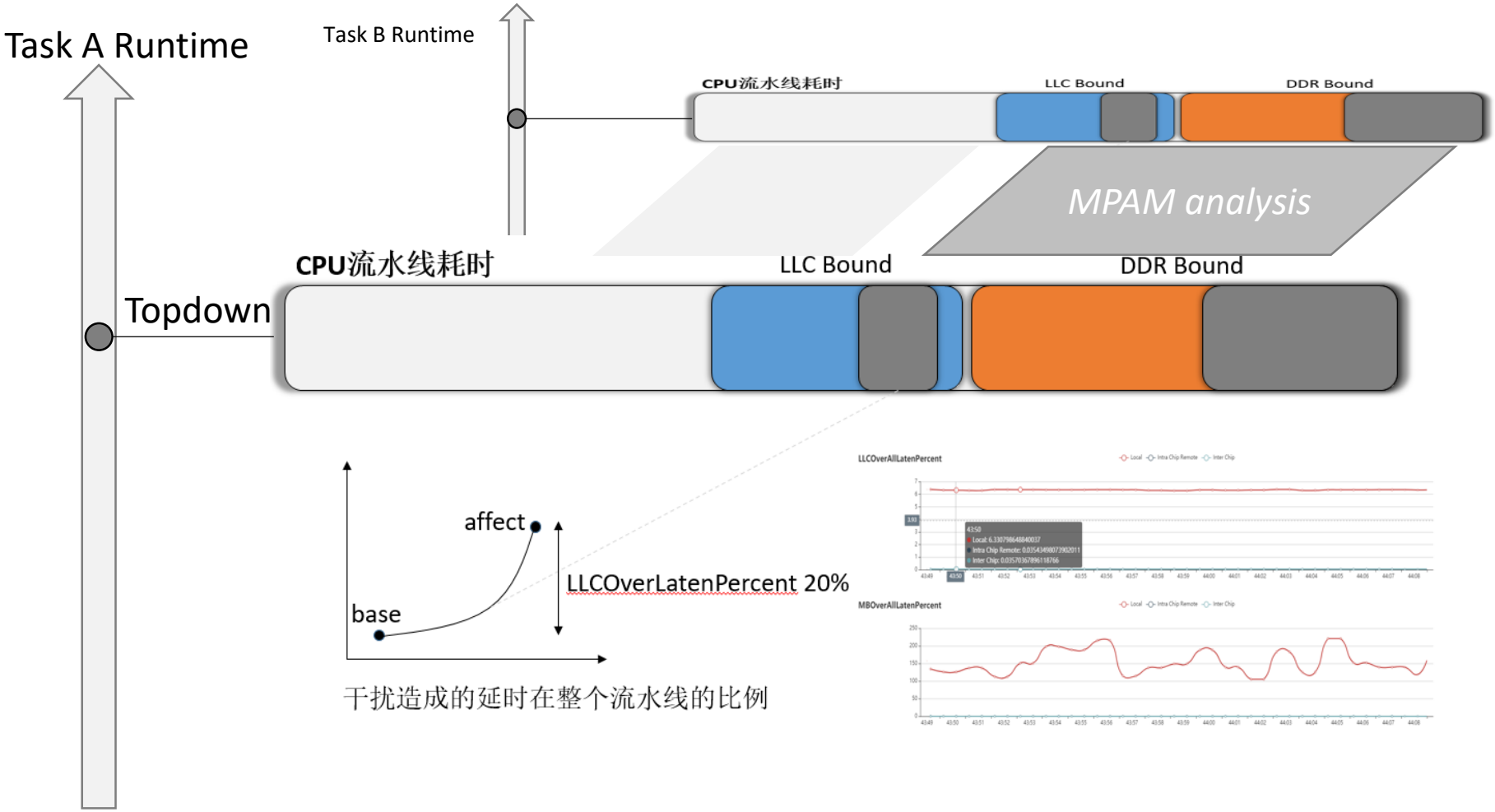
- **混合业务时的你我需求**
- **结合不同层级的资源推算时延变化大小**



openEuler

openRSO框架介绍

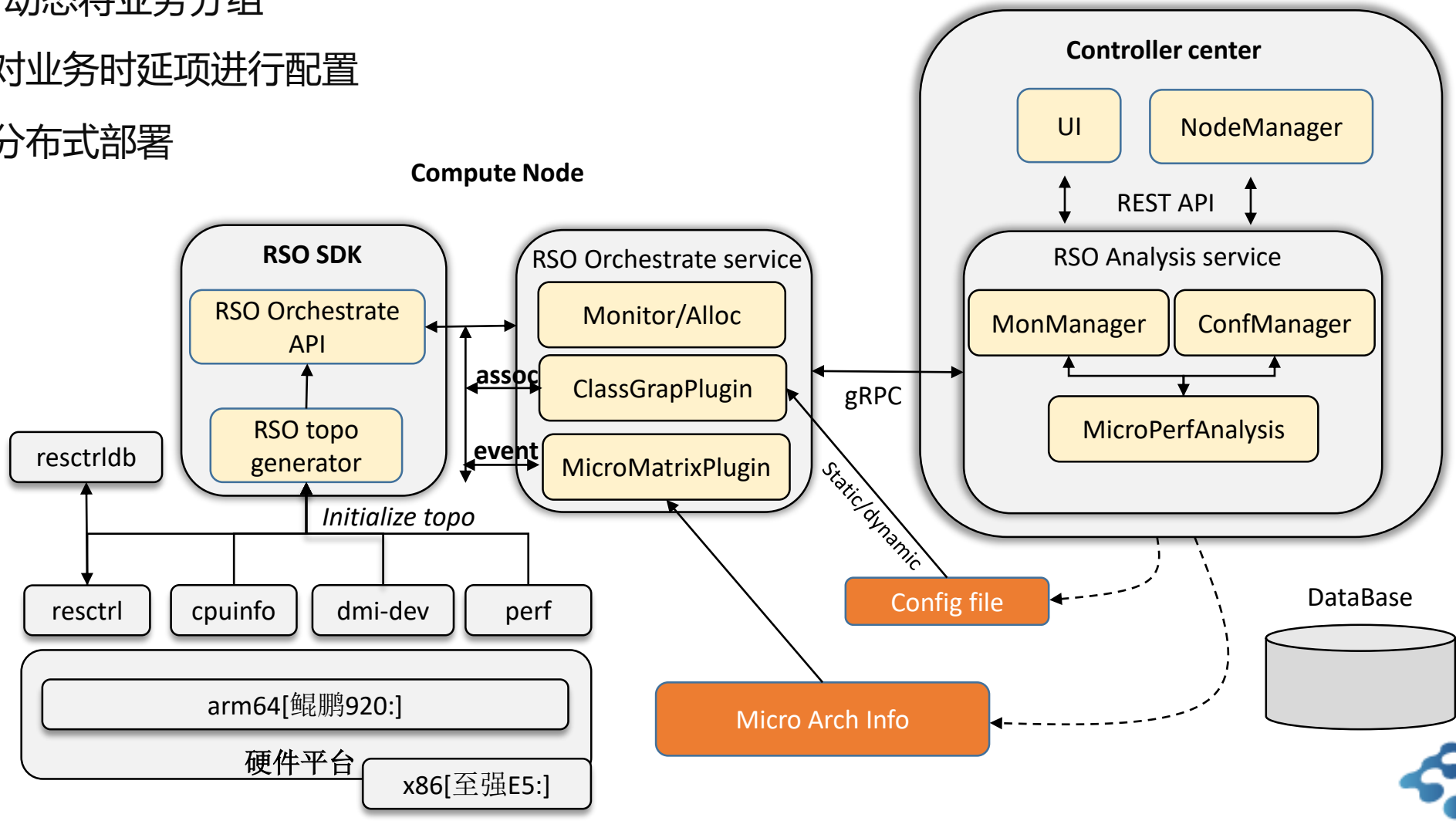
- 纵向结合Topdown模型分析Backend各个阶段的耗时
- 横向结合MPAM监控资源使用情况分析各个阶段的压力



openEuler

openRSO框架介绍

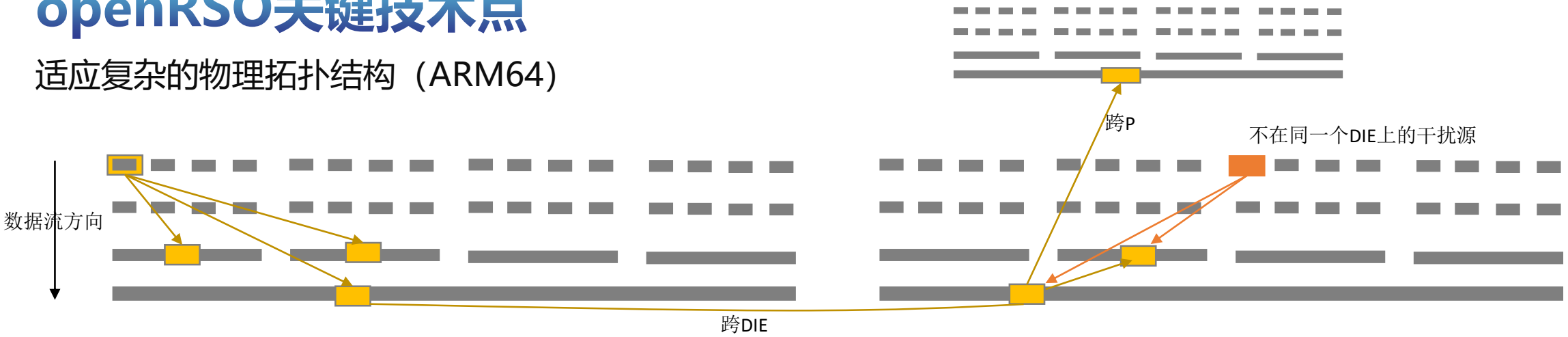
- 定制采集微架构PMU事件
- 静态/动态将业务分组
- 仅需对业务时延项进行配置
- 支持分布式部署

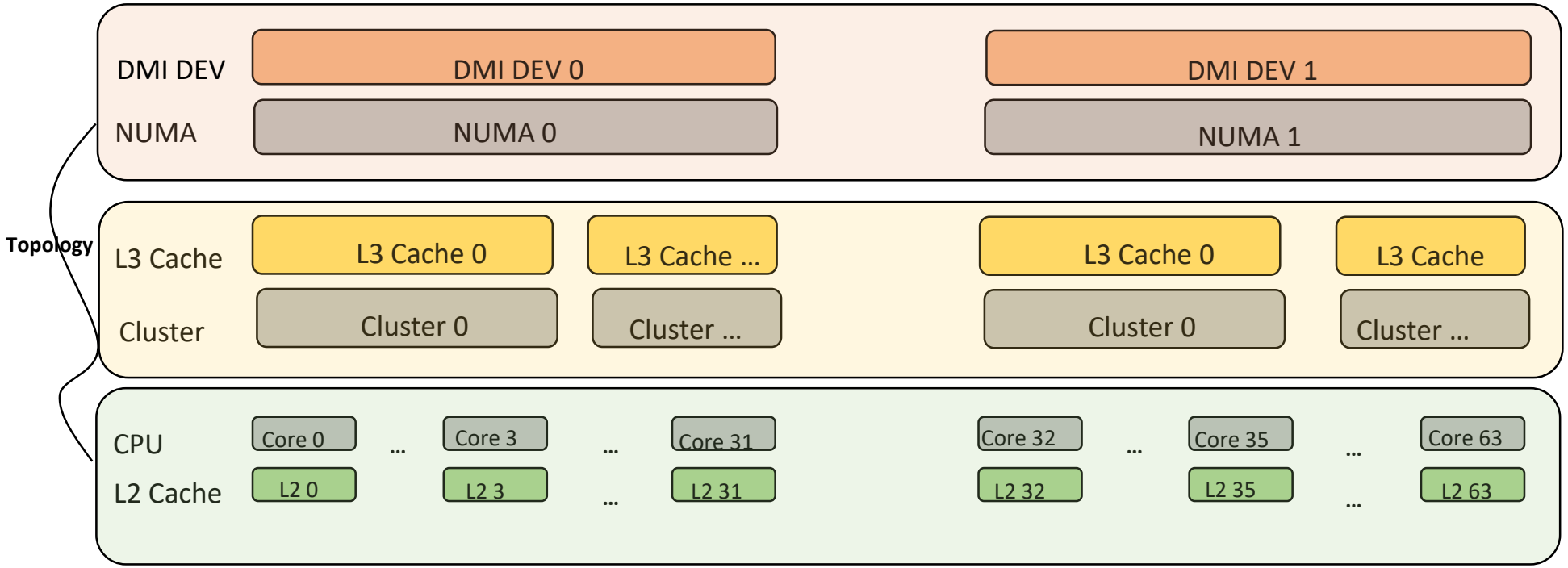


openEuler

openRSO关键技术点

适应复杂的物理拓扑结构（ARM64）

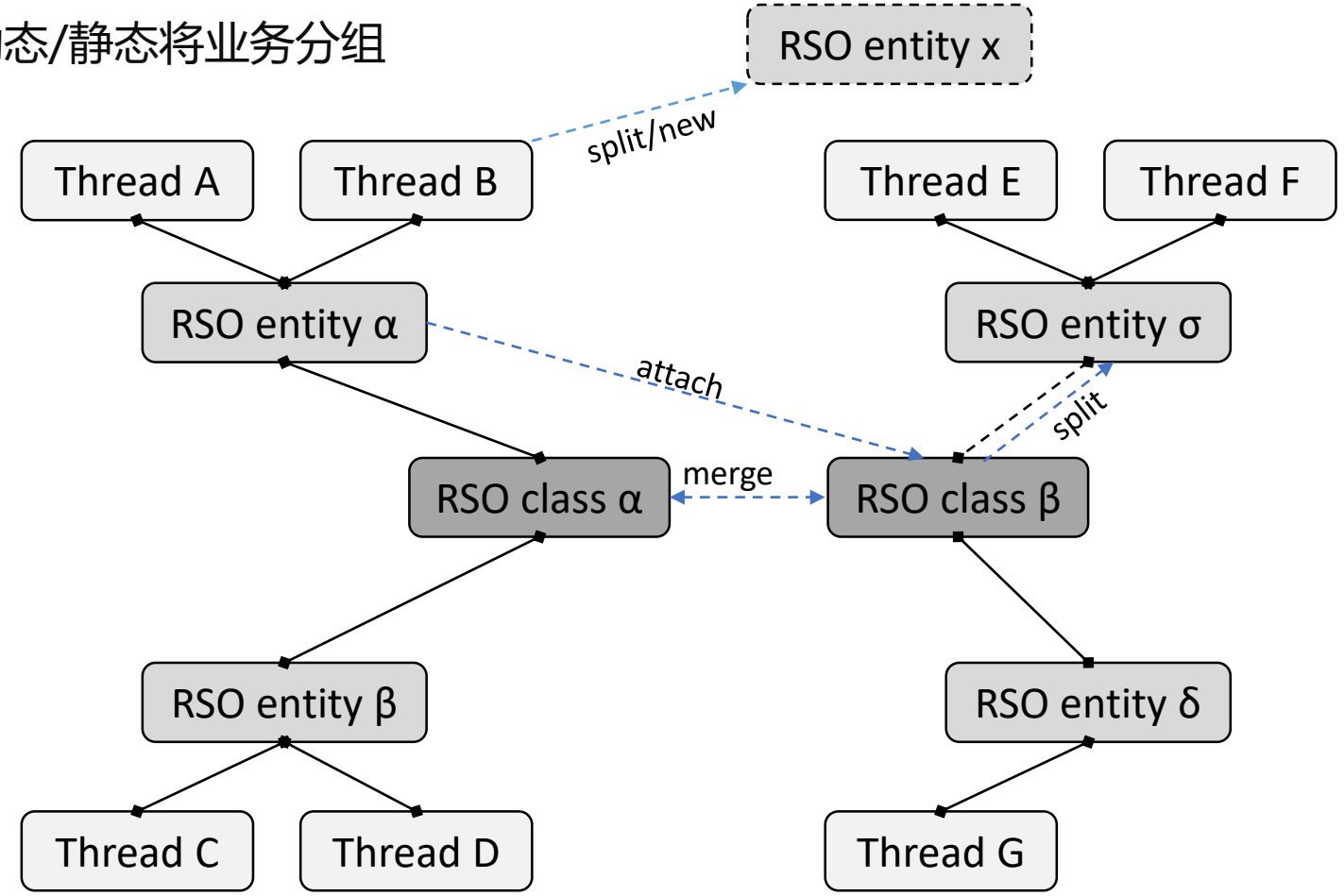




openEuler

openRSO关键技术点

动态/静态将业务分组



Operations:

Static:

- **Split B to RSO entity x**
- **Merge class** α and class β
- **Attach entity** α to class β

Dynamic:

- **Auto Split entity** α in some **conditions**
- **If latency of** class α exceeds H, merge **class** α and class β
- **If affect latency of** entity σ from entity δ exceeds I, Split it to new class

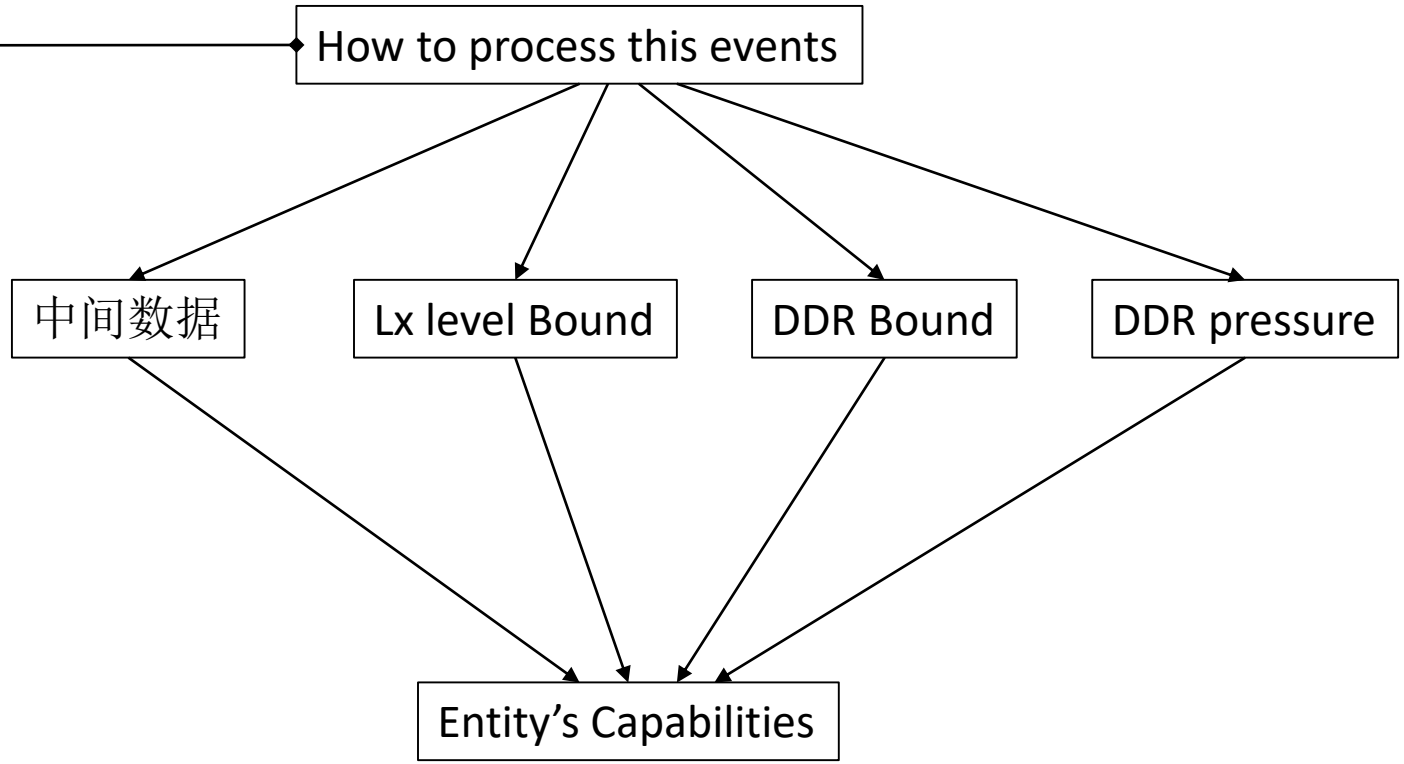
每个class为调度资源配置的最小实体，每个entity为监控的最小实体。

openEuler

openRSO关键技术点

定制微架构PMU事件

```
mapEvtName = {
    1<<3 : 'instructions',
    1<<4 : 'cycles',
    1<<5 : 'ipc',
    1<<6 : 'memstall_l2misses',
    1<<7 : 'rl2_intraChit',
    1<<8 : 'll3_intraChit',
    1<<9 : 'lddr_intraChit',
    1<<10: 'rddr_intraChit',
    1<<11: 'rddr_interChit',
    1<<12 : 'l3_refs',
    1<<13 : 'l3_misses',
    1<<14 : 'rChip_access',
    1<<15 : 'stacycbkd',
    1<<16 : 'stacycfrd',
    1<<20 : 'memband',
    1<<21 : 'l3occ',
    1<<25 : 'mbmax',
    1<<30 : 'l3pbm'
}
```



openEuler

openRSO Demo实战

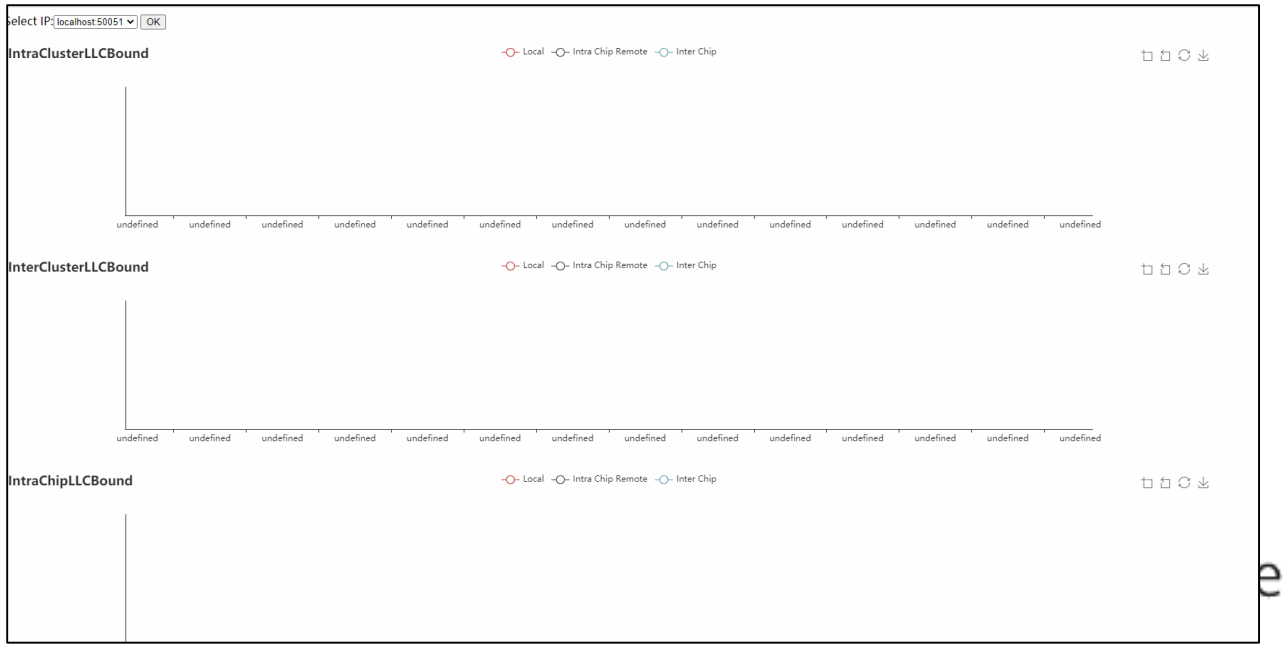
- **MPAM** 维护多业务运行时的资源调配
  - Arm v8.4 Extension特性
  - 鲲鹏920支持

场景：监控关键业务访存侧资源干扰情况，加压后观察波动，使用MPAM保护关键业务，再观察波动。

部署server

```
root@localhost server]# python3 qosos_grpc_server.py
```

获取节点业务运行情况(加压观测)



openRSO Demo实战

现场操作，对lmbench的bw_mem测试用例进行加压分析。

- 实验一，观察指定测试业务的各项数据，然后加压观察。
- 实验二，加压后限制资源前后观察各项数据，观察RSO预测所得性能劣化比。

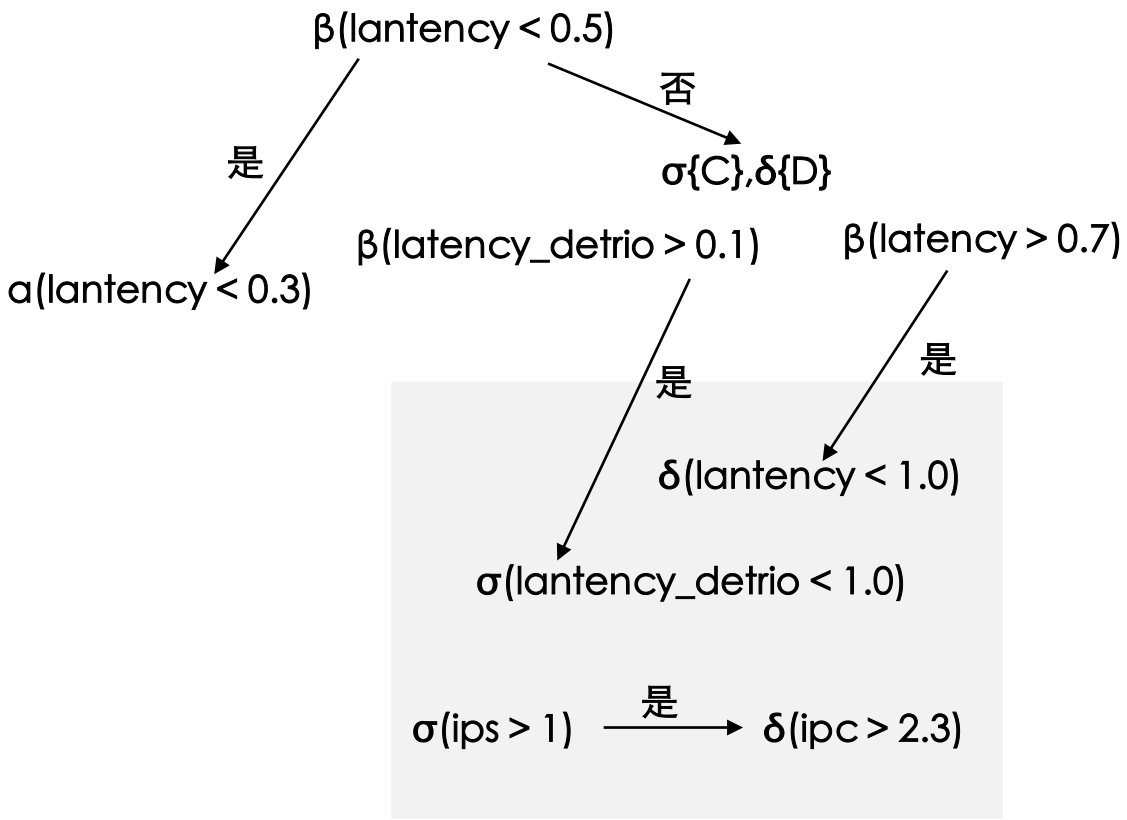
openEuler

openRSO 技术扩展

- 定义RSO原语用于设计资源配置策略

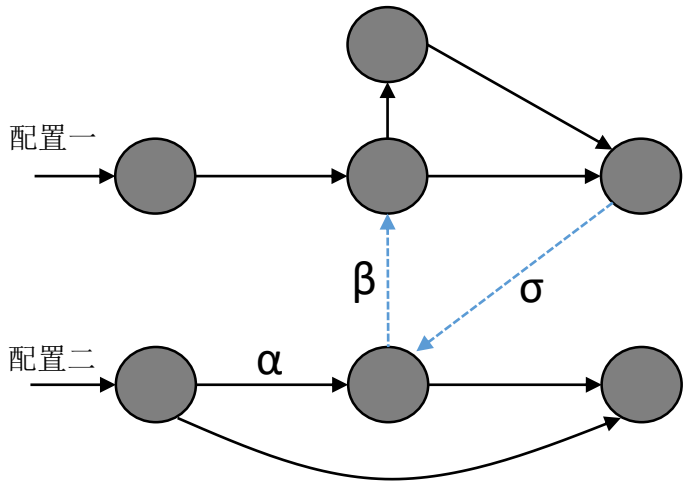
Declare:#申明
A = C{1,2,3,4} #组织CPU的最小单位
B = C{1,2}
C = T{2208,2132} #组织TASK的最小单位
D=T{2408}
{A}-> α
{C,D}-> β

Contract: #协议
β(lantency < 0.5)-> α(lantency < 0.3)
β(lantency > 0.5)->σ{C}, δ{D}
σ(ips > 1) -> δ(ipc > 2.3)
β(latency > 0.7) -> δ(lantency < 1.0)
β(lantency > 0.7&&latency_detrio > 0.1) -> σ(lantency_detrio < 1.0)

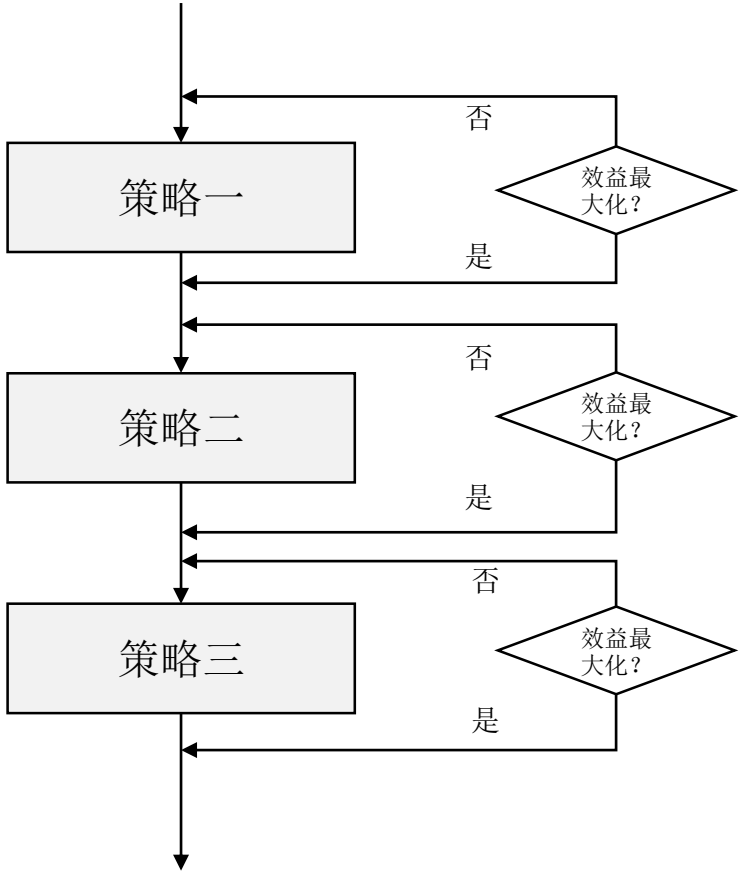


openRSO 技术扩展

- 形式化验证使用RSO原语组织的语法



- 机器学习从业务整体分析最优配置



openEuler

总结展望

- 需要内核支持MPAM/RDT，可以使用openEuler kernel的内核版本：https://gitee.com/openeuler/kernel
- RSO仓库地址，源码后续开放出去：https://gitee.com/openeuler/openRSO
- openRSO兼容x86 RDT和arm 64 MPAM提供的resctrl接口；openRSO可以定制PMU事件；openRSO可以适应复杂的物理拓扑结构；openRSO提供中间层原语定制策略，上层应用不需要关注如何分组，如何去做资源配置，例如如何调整Cache way数量。

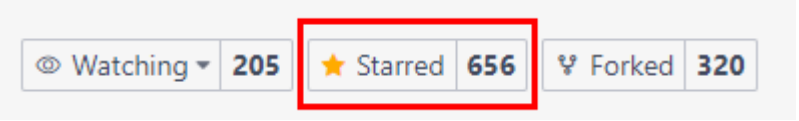
openEuler

✓ openEuler kernel gitee 仓库

源代码仓库

https://gitee.com/openeuler/kernel

欢迎大家多多 Star，多多参与社区开发，多多贡献补丁。



✓ maillist、issue、bugzilla

可以通过邮件列表、issue、bugzilla 参与社区讨论

欢迎大家多多讨论问题，发现问题多提 issue、bugzilla

https://gitee.com/openeuler/kernel/issues

https://bugzilla.openeuler.org

kernel@openeuler.org

✓ openEuler kernel SIG 微信技术交流群

请扫描右方二维码添加小助手微信

或者直接添加小助手微信（微信号：openeuler-kernel）

备注“交流群”或“技术交流”

加入 openEuler kernel SIG 技术交流群

技术交流

openEuler

Thank you

openEuler