



容器在离线混合部署技术介绍





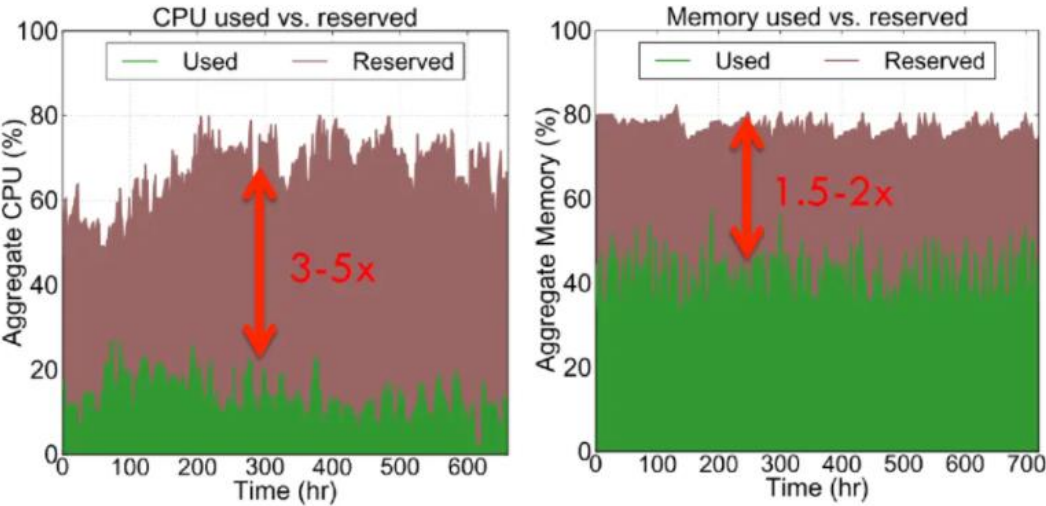
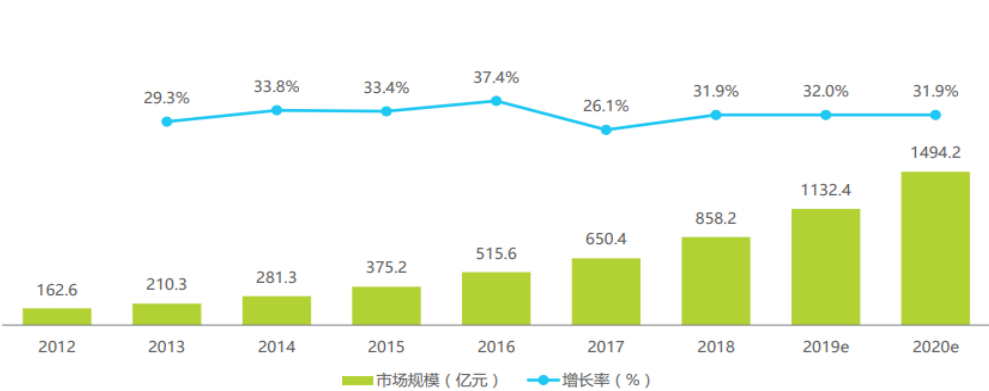
01 方案介绍与案例



容器在离线混部混合部署背景

全球数据中心规模巨大（万亿美元），并伴随高速增长（千亿美元/年），但总体资源利用率很低
据Gartner 统计，数据中心平均使用率不足15%，存在巨大的资源浪费

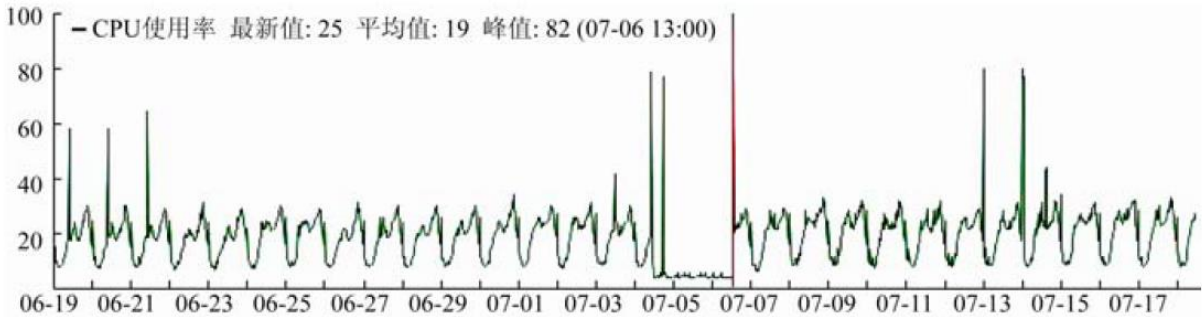
2012-2020年中国数据中心市场规模及增长率



推特数据中心资源使用情况

资源利用率低的原因

- 不同类型业务使用独立服务器资源池
- 不同类型业务使用独立调度系统
- 业务的资源预留远大于实际使用
- 在线业务存在波峰波谷，波谷时段较长

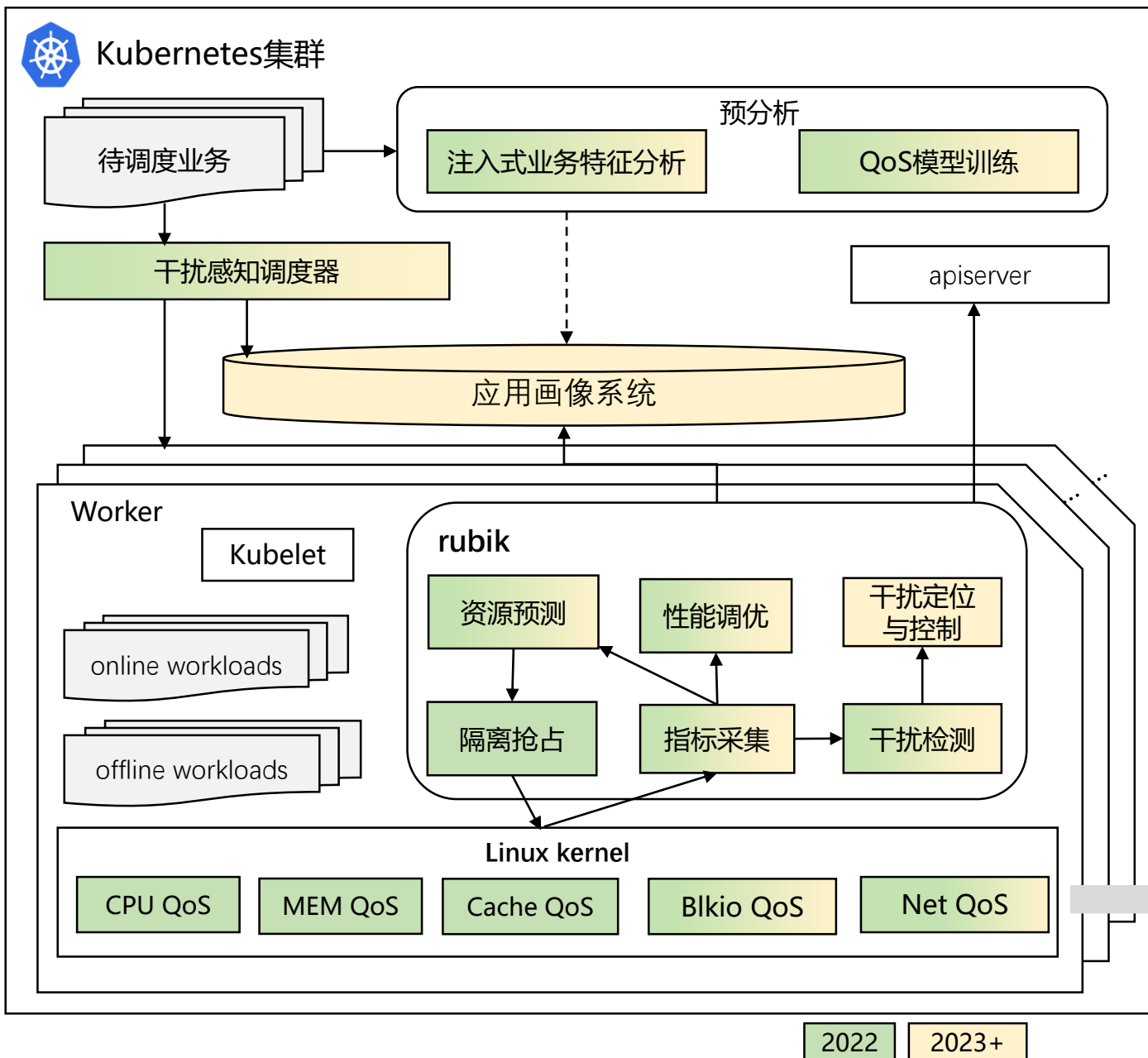


某在线作业所在服务器CPU使用率曲线（1个月）

从下表可以看出，在离线业务具有强互补性，利用离线业务填充在线波谷时段，能有效提升总体资源利用率

特征	在线业务	离线业务
质量要求	高	低
响应延迟敏感	高	低
任务运行时间	长	短
资源使用规律	动态变化	持续高位
请求响应时间	短	长
典型场景	Web、游戏、电子商务等	大数据分析、转码、AI训练等
调度系统	Kubernetes、mesos	Kubernetes、Hadoop、Spark

容器在离线混合部署总体解决方案



集群感知调度

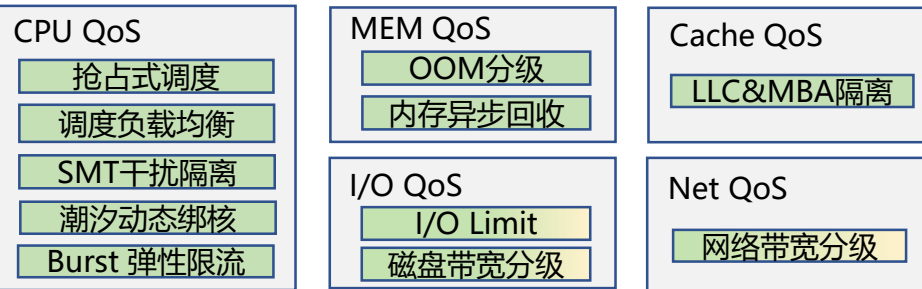
- 全维度指标收集，基于应用画像调度
- 实时资源预测调度离线作业
- 集群在离线负载均衡，反馈式重调度

混部引擎 (rubik)

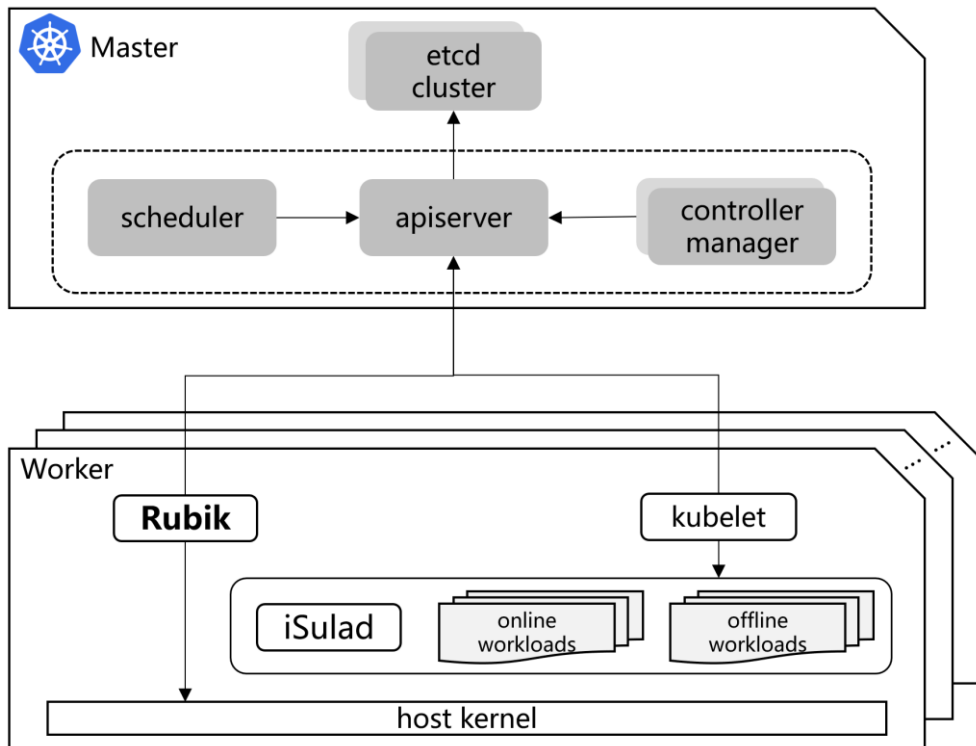
- 单机资源编排，减少性能干扰 (HT、cache、NUMA)
- 实时干扰检测，抑制离线任务扰动
- 实时健康检测、自动恢复、异常告警

隔离技术

- 内核抢占式调度、在离线负载均衡
- 内存分级回收、分级OOM
- 网络带宽抢占
- 磁盘回写限速、带宽抢占

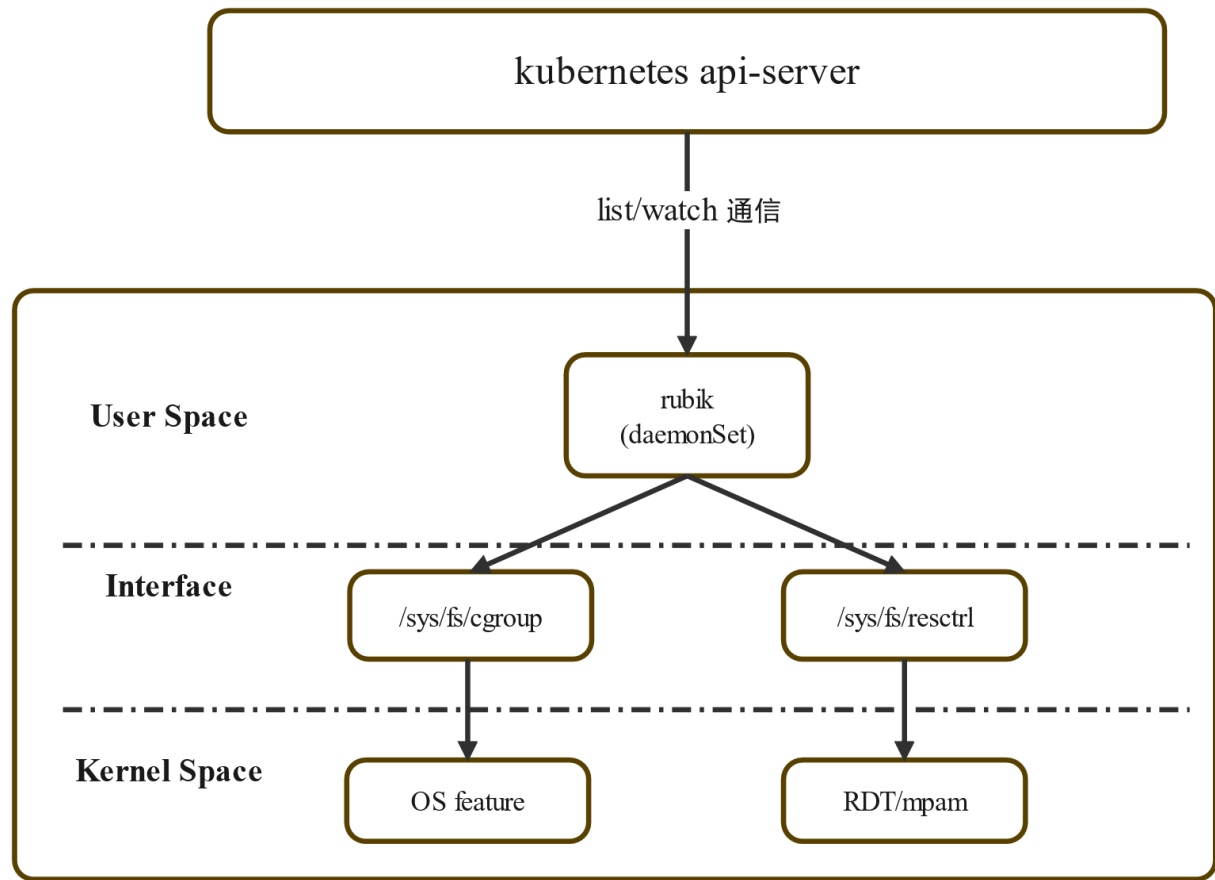


容器混部引擎 rubik : 方案架构



混部引擎rubik定位:

1. **能力定义:** 基于单机视角, 面向云场景, 支持
 - a) 单机资源编排, 减少性能干扰 (HT、cache、NUMA)
 - b) 实时干扰检测, 抑制离线任务扰动
 - c) 实时健康检测、自动恢复、异常告警
2. **部署方式:** 以daemonSet的形式运行于单机节点中。



案例

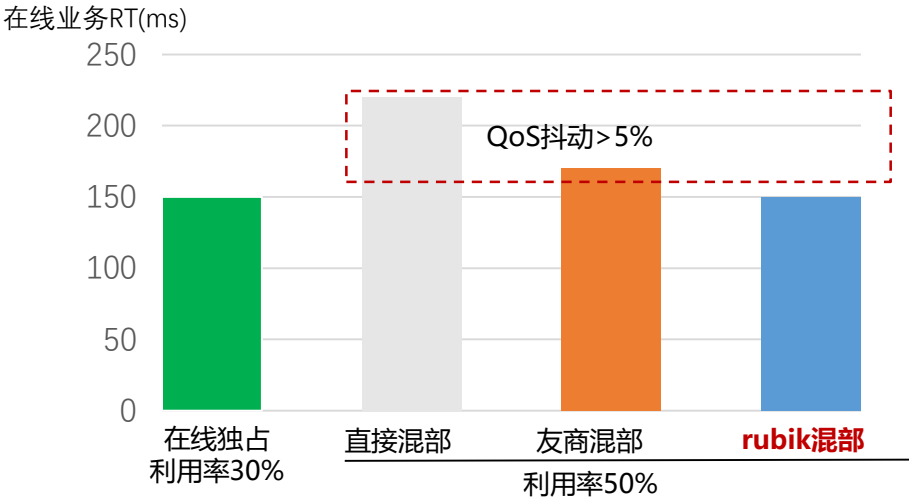
案例1

客户X对主流公有云技术测试，测试在节点的资源利用率混部到50%时，关键业务的QoS抖动情况。华为云**满分通过打标（排名第一）**

典型场景测试：

- 在线业务：在线推理
- 离线业务：大数据Spark
- 关键技术：CPU绝对抢占

打榜结果	华为云	云厂商A
资源利用率	50%	50%
在线业务QoS抖动	<1%	>5%

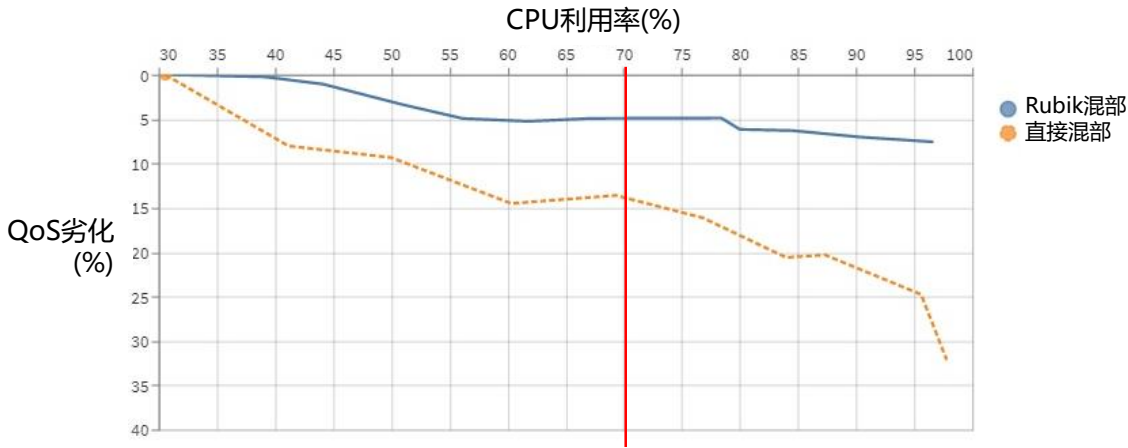


案例2

样板集群混部利用率突破70%，关键业务QoS无损（抖动<5%）

样板集群业务：

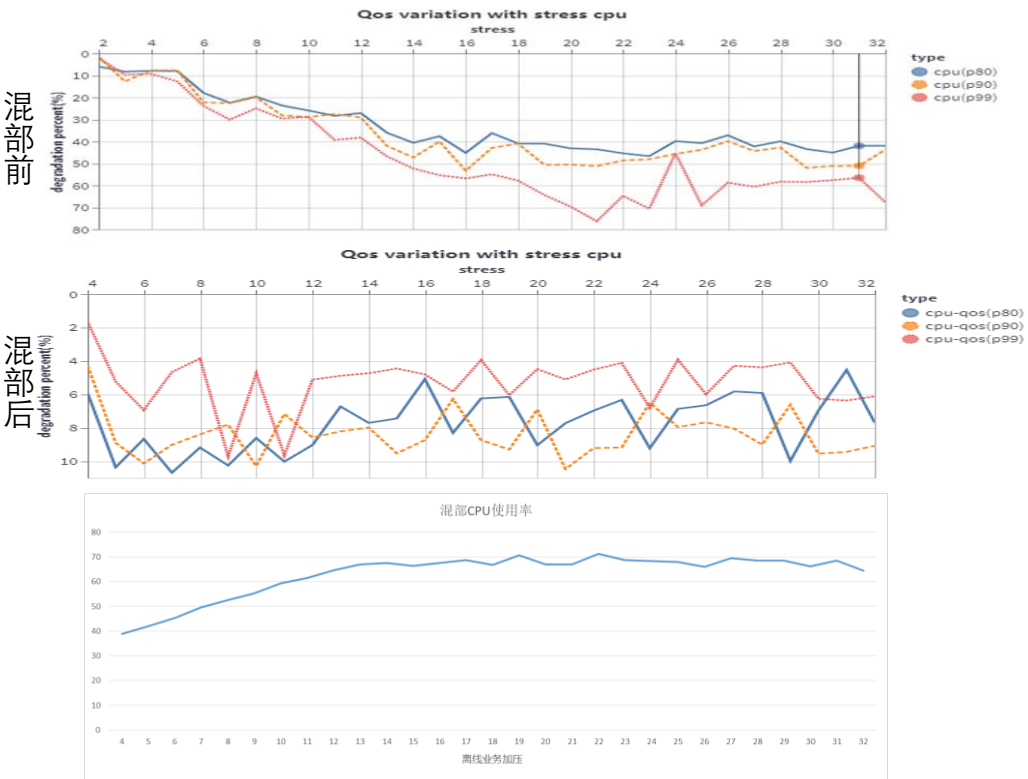
- 在线业务：云服务Y1（定位类、RT~10ms）、云服务Y2（服务发现类，RT~5ms）
- 离线业务：大数据Spark
- 关键技术：CPU绝对抢占、调度负载均衡、L3/MB干扰量化&控制、SMT干扰隔离



案例

案例3

- 在线业务：ClickHouse数据库
- 离线业务：AI机器学习
- 关键技术：CPU抢占、CPU调度负载均衡

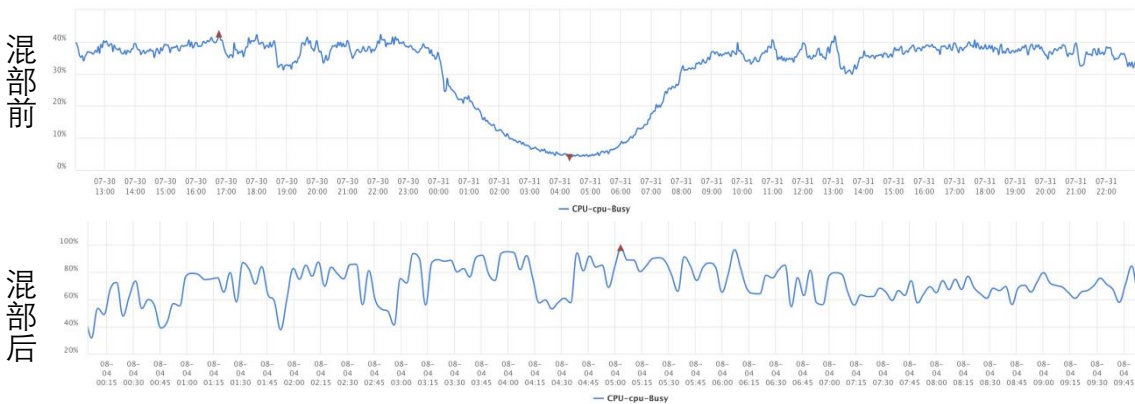


混部成果

- 混部前**：CPU利用率达到30%+，在线业务QoS下降达X%
- 混部后**：CPU利用率达到70%，在线业务QoS下降<10%

案例4

- 在线业务：新浪微博（Web服务），P99敏感（30ms）
- 离线业务：Hadoop、AI训练
- 关键技术：CPU抢占、SMT干扰隔离、CPU调度负载均衡



混部成果

- 混部前**：在线业务存在明显波峰/谷，在线节点利用率 25%
- 混部后**：整机利用率60%+，在线业务P99抖动<10%
- 现网已上线**4000+台，节省服务器采购成本数百台、降低成本数百万



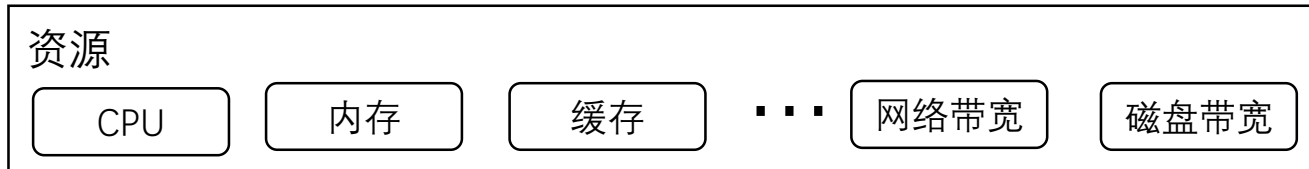
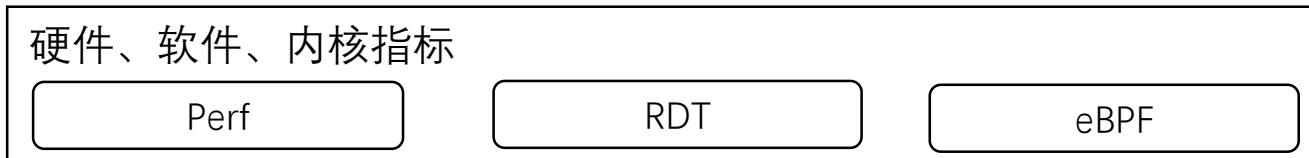
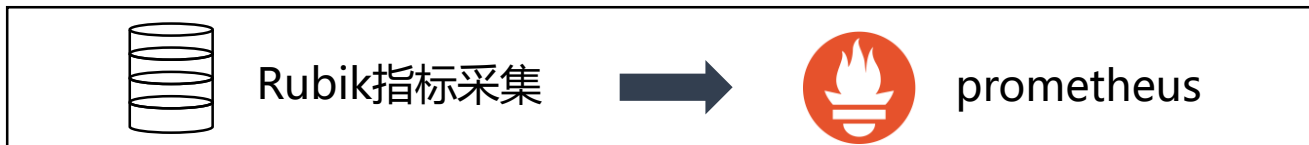
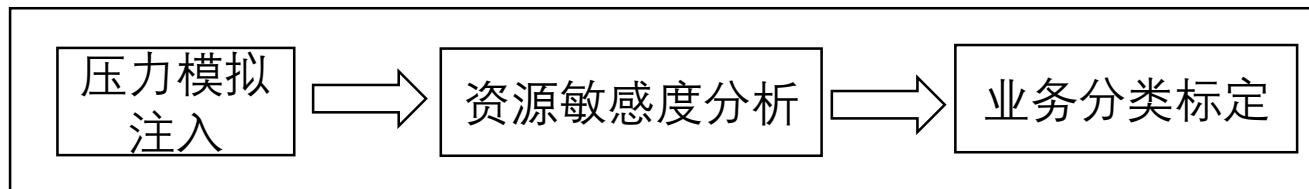
02 技术点剖析



► 应用干扰分析：QoS量化建模

回归分析建模

指标相关性分析



干扰分析

通过软硬协同分析的方式，研究负载执行时的应用级特征和系统层特征，识别节点内在线业务QoS是否符合预期以及业务之间性能干扰导致的性能下降的因素

业务分类

分析业务对资源的敏感度

指标采集

实时收集的节点内Pod资源使用量RDT、eBPF、perf等硬件、内核及QoS业务等相关指标

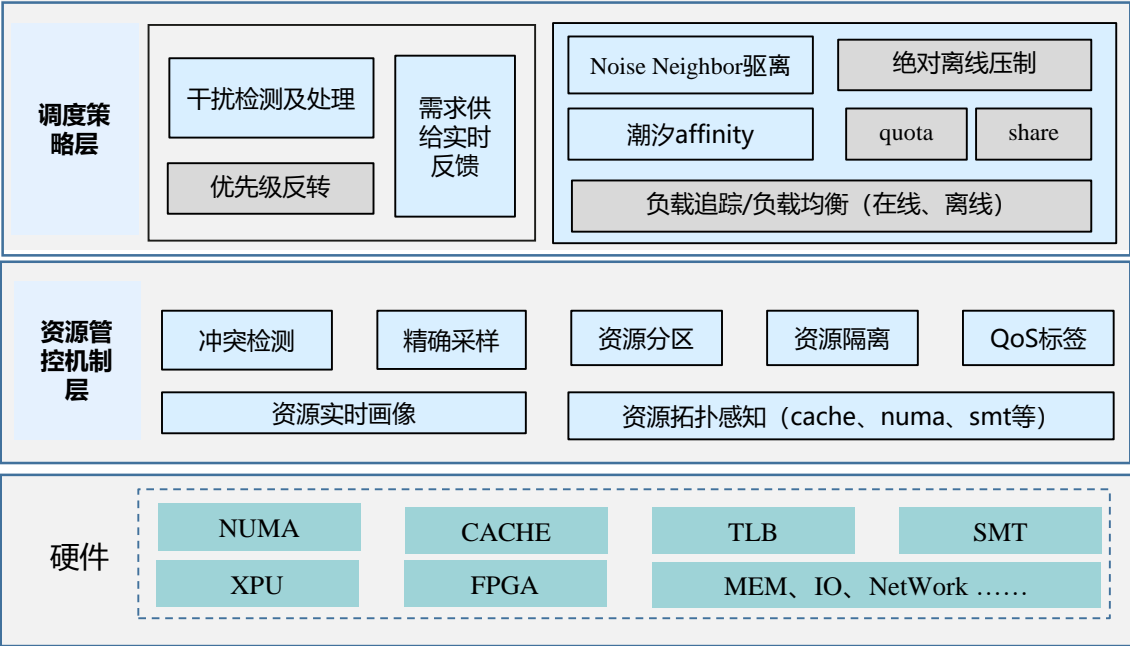
► CPU QoS（1）：CPU调度分级隔离技术

痛点：

Linux原生组调度方案提供了对不同分组任务配置CPU配额的功能，但即使在极端配置下，依然无法限制住离线任务偶发运行，从而干扰在线任务的相应时延。

场景：

离线任务在运行时，支持在线任务快速抢占离线任务及在线任务能够压制离线任务获得cpu运行。



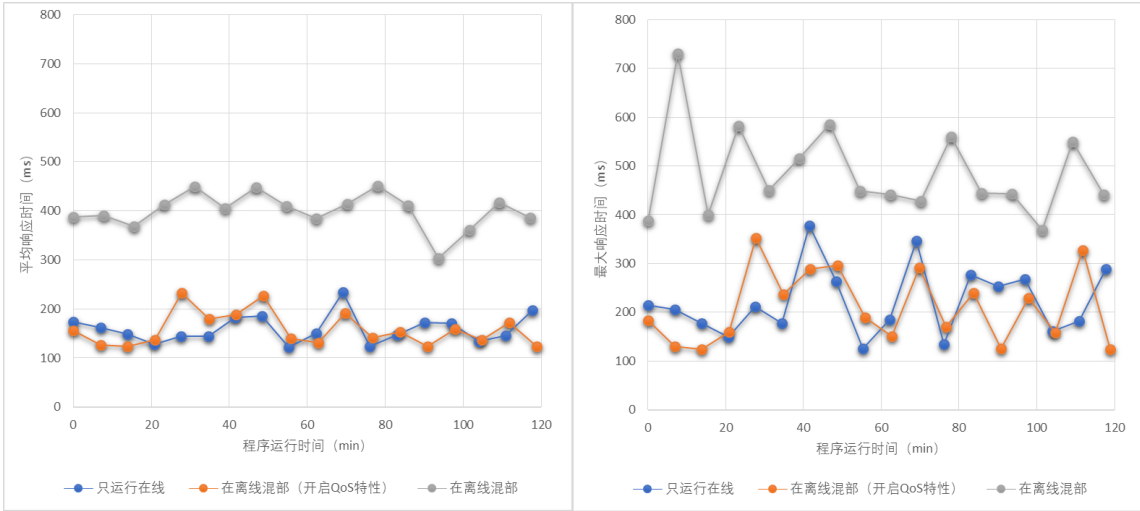
技术方案：

多维任务间CPU资源干扰隔离

- 基于标签化的抢占技术保障**us级任务抢占**；
- 混部业务CPU资源时域和空域二维压制；
- **SMT在离线强制隔离**，实时驱逐离线任务
- **在离线分层负载均衡**
- 解决优先级反转&低维任务信号响应及时性等难题

效果：

CloudSuite验证CPU隔离效果，使用web-serving模拟在线业务，in-memory-analytics模拟离线业务，开启CPU QoS隔离特性，在离线混合部署，在线业务QoS不受影响。



► CPU QoS (2) : QuotaBurst柔性限流技术

痛点:

客户因为CPU限流导致服务性能下降, 希望在保证limit配置合理的情况下, 避免CPU限流, 允许短时间突破limit

场景:

1. 单个应用实例CPU规格较小, 使用率较低, 有规律突发增长
2. 应用长期平稳运行, 运行过程中突发CPU使用飙升
3. 应用总体使用率较低, 但启动时使用率较高
4. 应用总体使用率较低, 某些实例使用率很高, 资源分配不均衡
5. 应用总体CPU使用率较高, 长期出现CPU压制
6. 应用总体CPU使用率较高, 且SLA等级很高

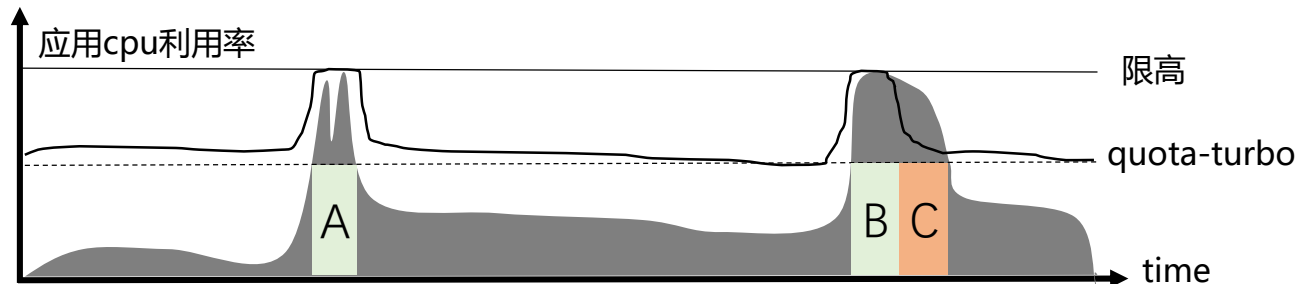


场景1



场景2

技术方案:



quota-turbo配额 (面积):

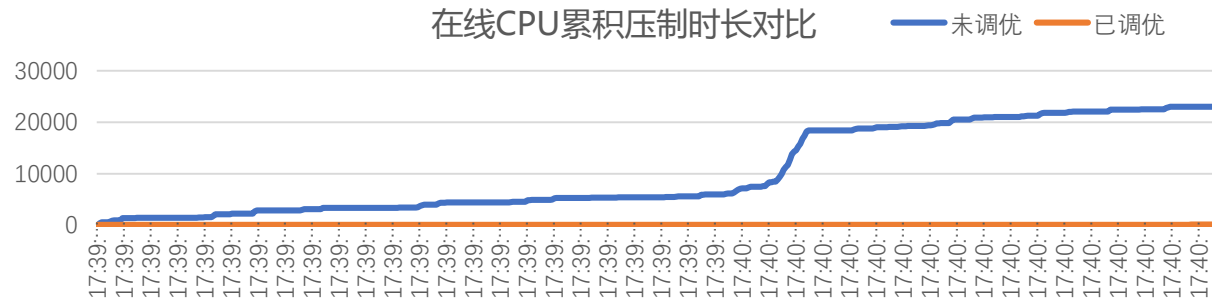
- 单个应用的整体利用率不超过限制, 允许短时间超过限制, 用完turbo配额将受到压制
- 应用启动之前利用率假定为0, 基于此给刚启动应用启动一定turbo配额

quota-turbo限高 (高度):

- 除了配额之外会限制turbo最大高度, 高度受节点整体利用率影响, 节点整体水位越高, 各应用限高线越低

效果:

使用CloudSuite web-serving模拟在线业务, stress-ng压力注入模拟离线业务, QuotaBurst柔性限流技术, 在离线混合部署, 在线业务CPU压制次数显著减少



MEM QoS (1) : 内存OOM分级

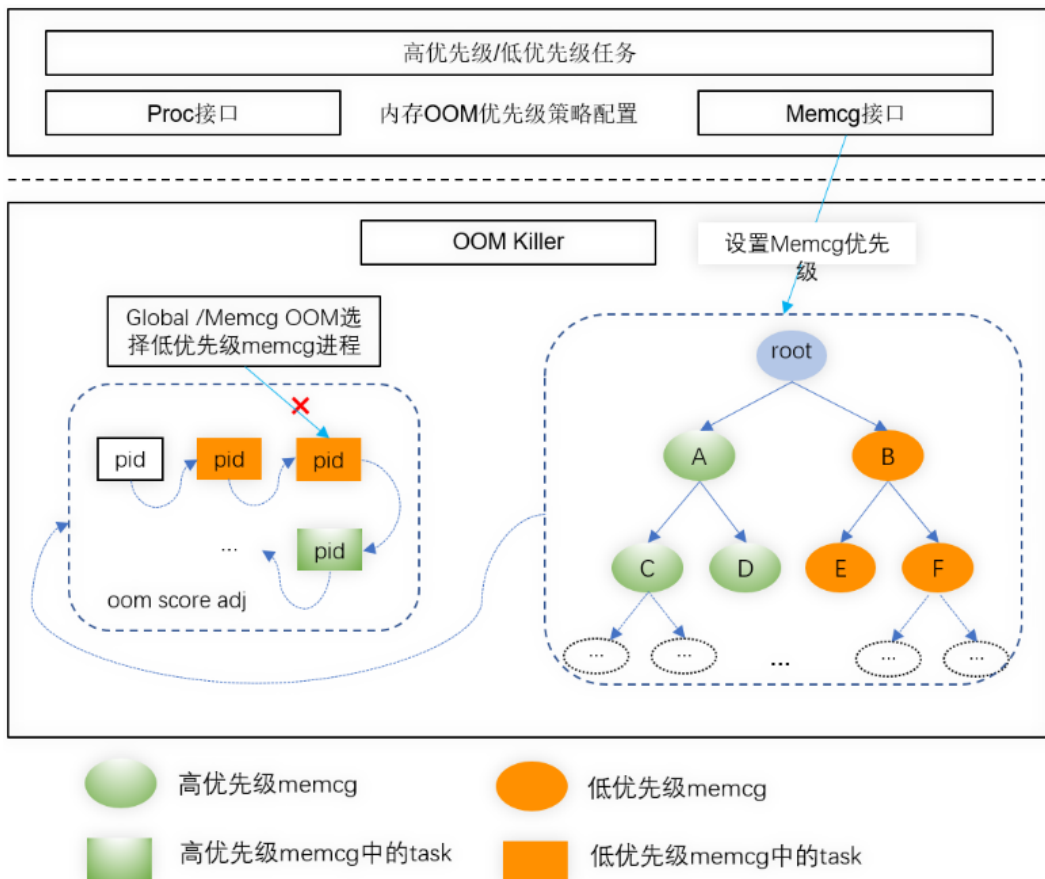
痛点:

Linux原生提供了限制应用最大使用量，但是无法在竞态下保证在线Qos的同时保证内存的整体使用率。

场景:

在离线混部时，离线应用占据大量page cache类型内存，影响在线Qos的场景。

技术方案 (1) : OOM分级



处理策略:

1. 在发生OOM时，优先终止低优先级memcg中的任务，保证高优先级memcg中的任务运行；
2. 在一个memcg中若存在多个任务，按照任务的oom_score_adj和内存使用情况评估任务的分数，终止分数最高的任务

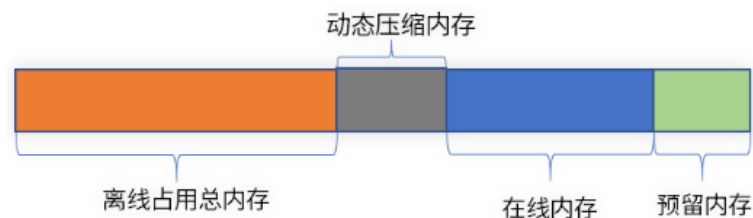
MEM QoS (2) : memcg异步回收

技术方案 (2) : memcg异步回收

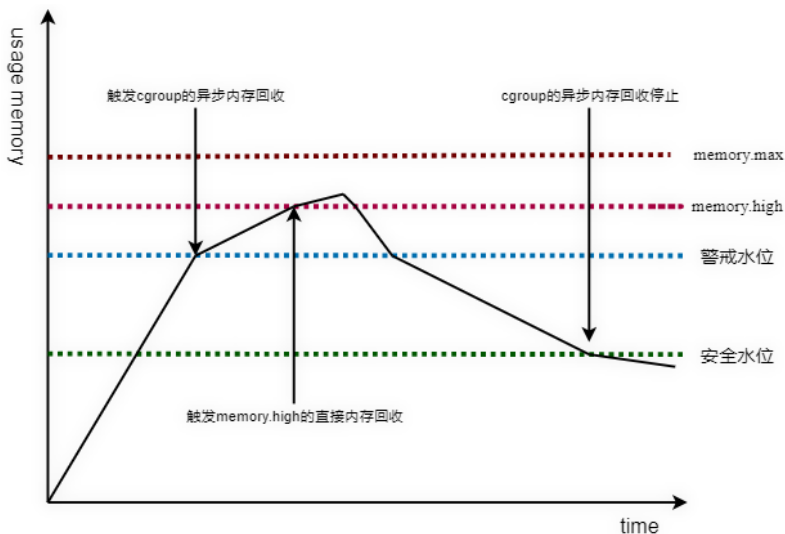
AS IS



TO BE



FSSR (快释放慢恢复) 策略

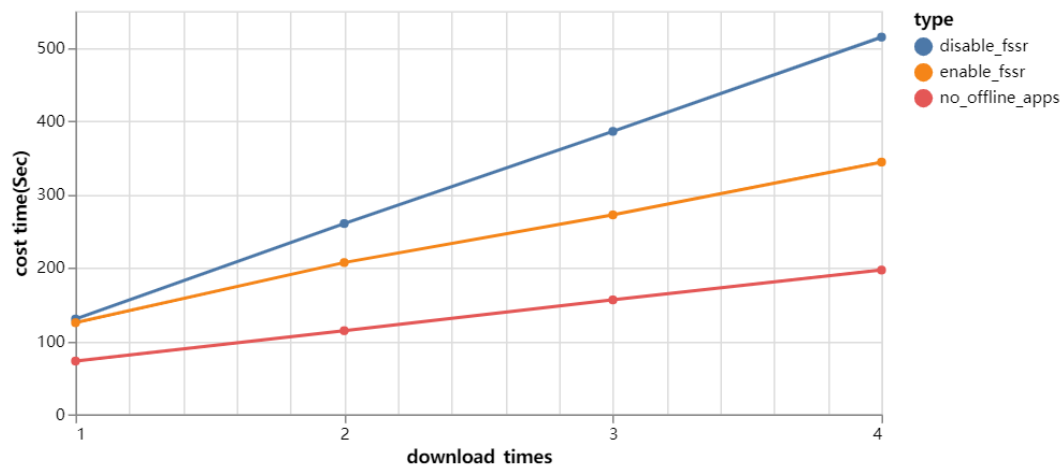


- 前提：内核支持memcg内存水位线。
- 当memcg内存使用量超过警戒水位线时，会触发异步回收内存，回收当前memcg中可以回收的内存，确保当前内存使用量低于安全水位线时停止
- 当memcg内存使用量超过设置的memory.high时触发memcg级主动回收，回收一切可以回收的内存，确保memcg内存使用量不超过memory.high

效果：

验证场景：在线应用nginx，离线应用通过stress读取文件

结果：开启内存分级策略之后效果好于没有开启策略的场景，且更接近没有离线应用的场景。



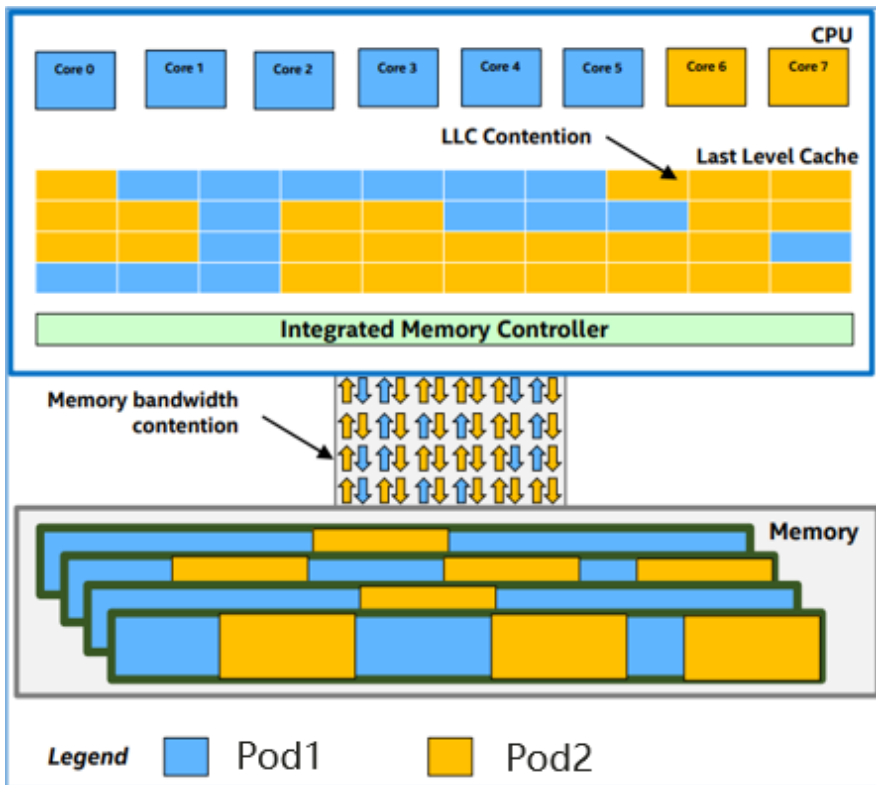
Cache QoS: L3/MB干扰量化&控制

痛点:

混部后离线业务刷cache可能导致在线业务性能下降

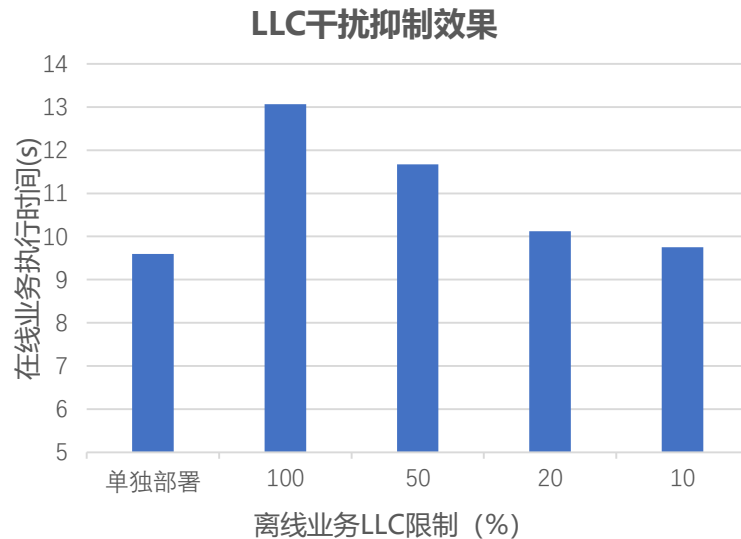
技术方案:

采用intel RDT/mpam内核技术, 限制离线业务cache使用量, 且通过检测在线业务cache指标判断其是否受到cache干扰, 动态调整离线业务的cache水位。



效果:

- 使用bzip2压缩文件 (64MiB) 模拟在线业务员, 使用stream压测内存带宽模拟离线业务, 测试离线不同水位条件下在线业务压缩时长
- 控制离线作业cache水位能有效减少LLC/MBA对在线的干扰 (<5%)



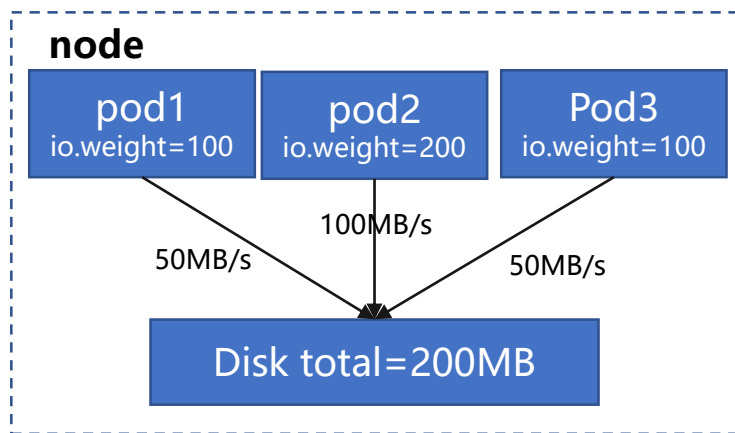
► Disk QoS: IO COST

痛点:

混合部署时，在线任务和离线任务共享IO，且IO资源有限。在线任务和离线任务相互抢占IO，导致在线Qos下降。且cgroup v1没有相关特性支持。

场景:

多个IO密集型容器混合部署时，总IO使用量超过硬件可以提供的最大值时，需要根据应用的优先级调整容器可以使用的IO。

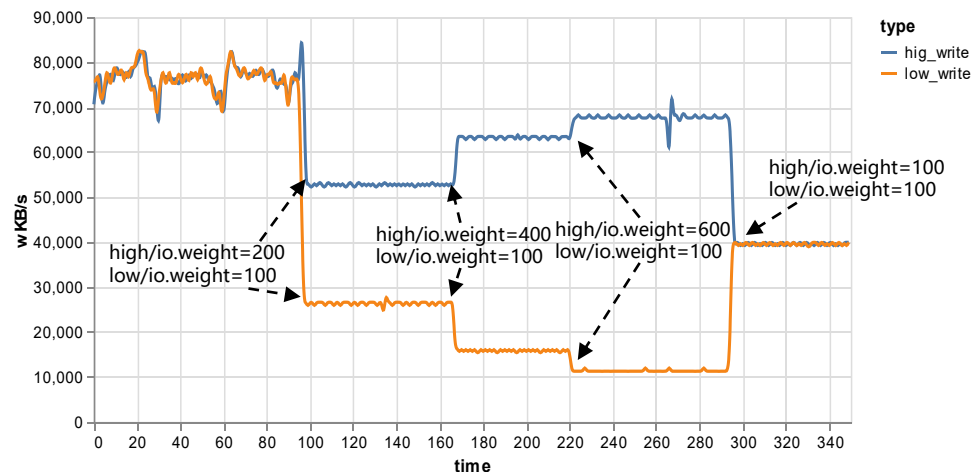


pod1/pod2/pod3同时向disk中写入数据，但disk最大带宽只有200MB，根据容器配置的权重分配相应的带宽。

技术方案:

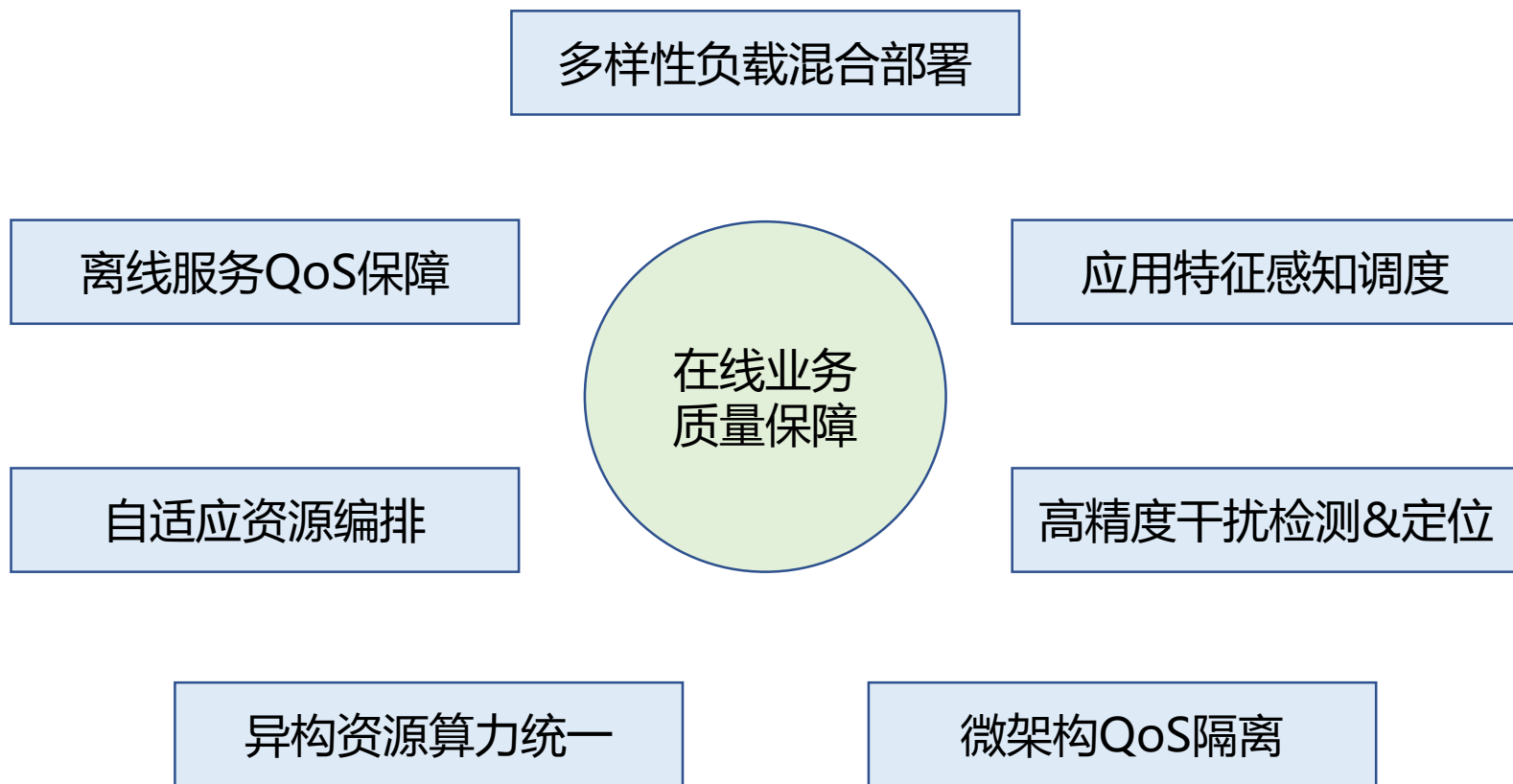
- 基于cgroup v2 io.cost特性，新增相同特性到cgroup v1中。并实现buffer io write下IO权重。
- 整体利用率不变的情况下，实现优先级写入。
- 动态调整，不需要重启容器实现IO权重动态改变。

效果:



- 蓝线为高优先级，橙线为低优先级。没有配置时两者抢占共同写入。
- 配置IO权重后，两者的实际写入速度比值接近配置的权重。

► 未来展望





THANKS

附录：混合部署特性规划时间表

现状：资源隔离技术存在部分技术难点，需要攻关，特性**分阶段交付和开源**；

建议策略：

- ① 基于测试验证场景明确所需要的特性（成熟特性），可优先进行POC测试效果验证；
- ② 基于伙伴的诉求，匹配特性时间点，进行生产环境试点
- ③ 存在技术难点等不成熟的特性，持续合作和联创，基于效果审视后续落地情况

商用支持：openEuler 社区LTS质量要求，OSV厂商或第三方提供商用保障支持。

隔离技术	特性名	openEuler主干	openEuler 22.03 (22/3/30)	openEuler 22.09 (22/9/30)	openEuler 22.03 SP1 (22/12/30)	openEuler 22.03 SP2 (23/6/30) 规划中	场景	备注
CPU	CPU调度分级抢占	√	√	√	√	√	内核态	
	SMT共享资源干扰隔离	√		√	√	√	内核态	
	CPU调度负载均衡	√		√		√	内核态	
	CPU Burst	√		√	√	√	内核态	
	QuotaBurst柔性限流技术	√				√	用户态	
MEM	OOM分级回收	√	√	√	√	√	内核态	
	内存异步回收	√			√	√	用户态	
NET	EDT网络流量限速	√				优化方案规划中	内核态	
	BWM带宽控制	√					用户态	当前方案需要上层业务适配
IO	IO Limit限速	√				按需	用户态	需适配新框架
	IO Cost限速	√			√	√	内核态	
CACHE	LLC通道隔离、MBA带宽隔离	√		√	√	√	内核态	需物理机支持
		√		√	√	√	用户态	