火龙果·整理
uml.org.cn

大规模GPU虚拟化在讯飞AI业务上的实践

火龙果·整理
uml.org.cn

目录

- 讯飞AI业务的发展
- GPU虚拟化技术
- 异构资源管理
- 业务落地方案与实践

火龙果·整理
uml.org.cn

讯飞AI业务的发展

语音识别

2018年6月12日
识别准确率提高至98%

语音合成

2018年07月14日
Blizzard Challenge
十三连冠

人脸识别

识别率高达99.4%

语义理解

自然语言各项
指标名列前列

场景识别

精准识别自然环境下
数十种场景

物体识别

全球领先的通用物体
检测算法，有效检测
图像中2万多种生活
常见物体

个性化

个性化定制场景高到171
项

火龙果·整理
uml.org.cn

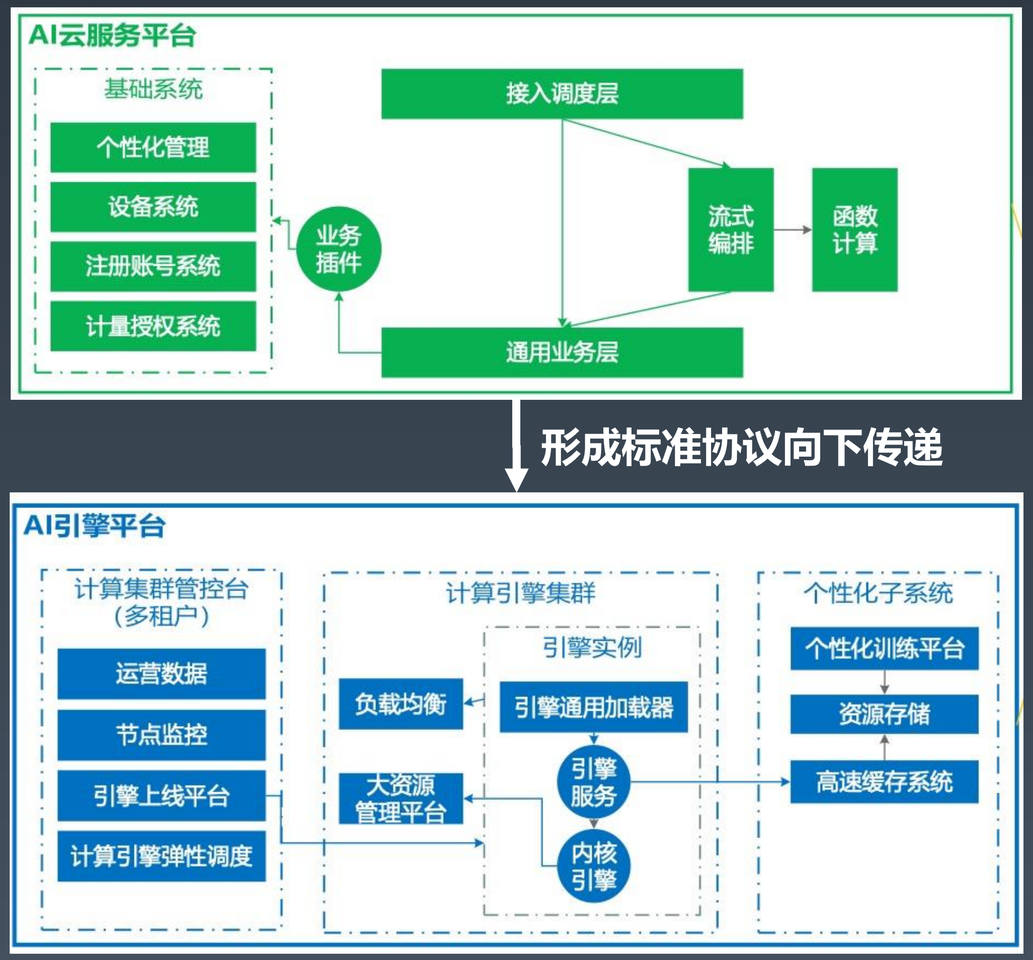
讯飞AI业务整体架构

AI云平台

用户管理
任务接入
任务编排
个性化管理

AI引擎平台

集群管控
资源管控
个性化系统



火龙果·整理
uml.org.cn

讯飞AI业务发展痛点

- 业务突增，拥有开发者90W+
- 平均日服务量，达40亿人次

随之带来的
- 资产规模扩大，管理混乱
- 资源分配不均匀
- 资源利用率较低
- 成本倍增 （尤其是cpu切换为gpu设备后）

2015年 GPU全天平均利用率48.6%

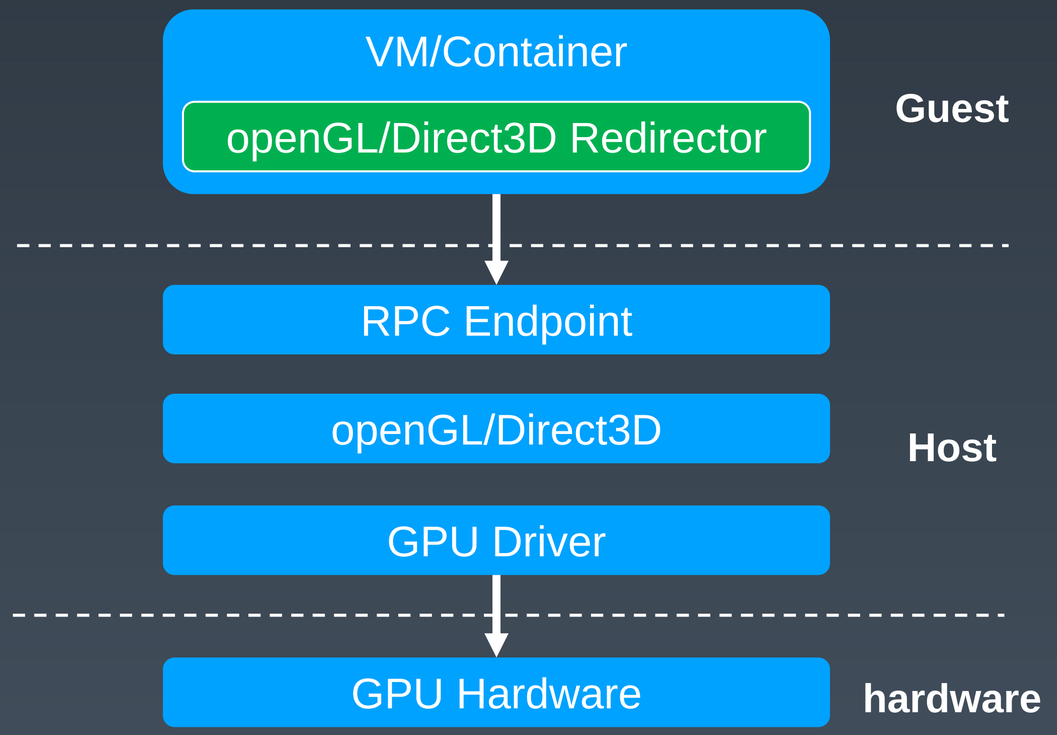
火龙果·整理
uml.org.cn

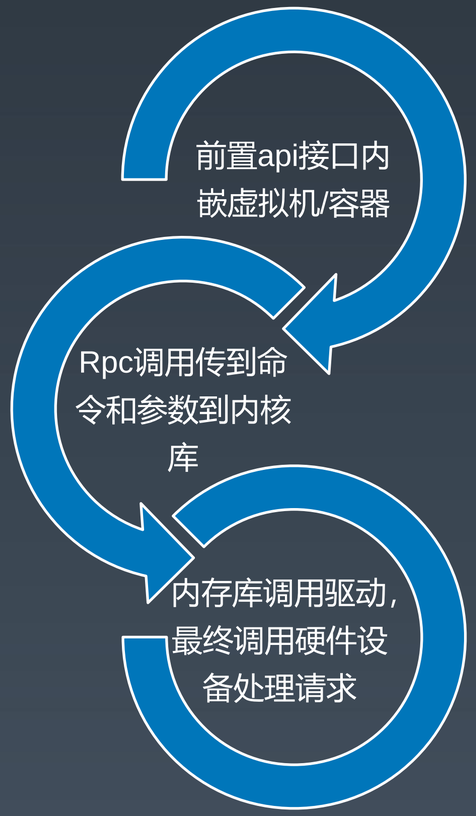
目录

- 讯飞AI业务的发展
- GPU虚拟化技术
- 异构资源管理
- 业务落地方案与实践

火龙果·整理
uml.org.cn

GPU虚拟化技术-协议传递





火龙果·整理
uml.org.cn

GPU虚拟化技术-协议传递

优点

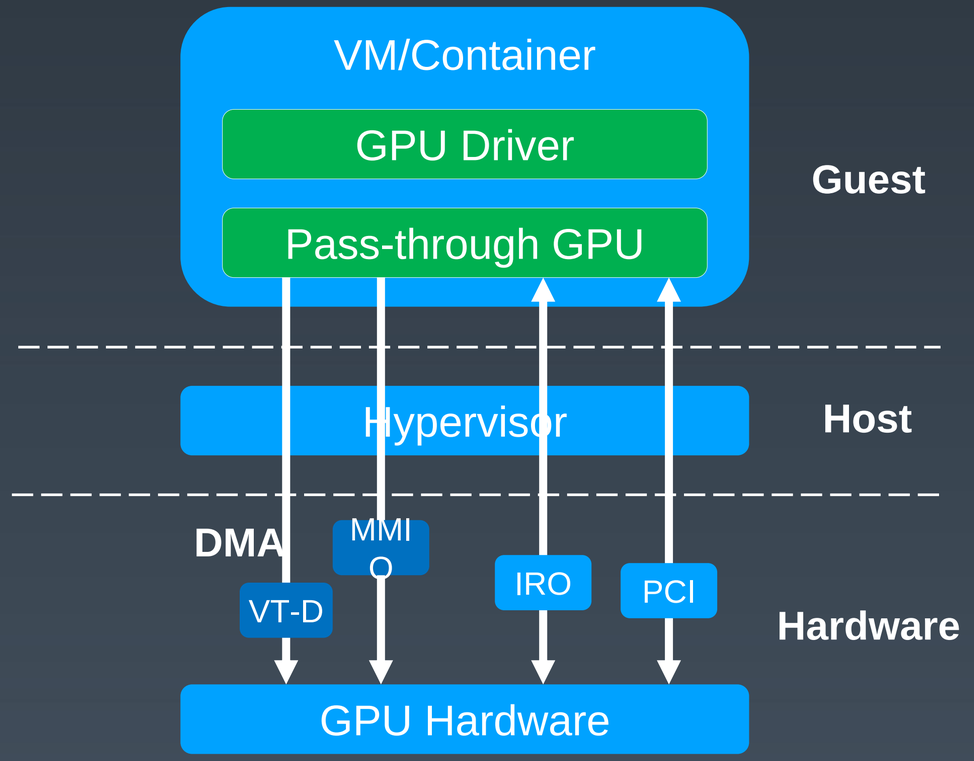
- 无需定制
- 无硬件要求
- 简单方便
- 小规模压力下，性能表现较好
- 业务无感知可任意迁移

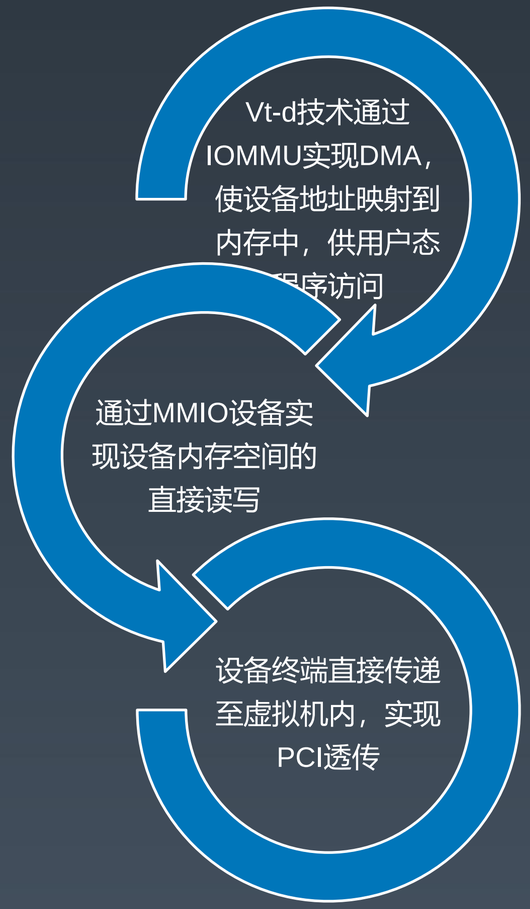
缺点

- 资源隔离差
- 多次中断切换，效率差
- 高性能计算下，性能损耗验证

火龙果·整理
uml.org.cn

GPU虚拟化技术-设备透传





火龙果·整理
uml.org.cn

GPU虚拟化技术-设备透传

优点

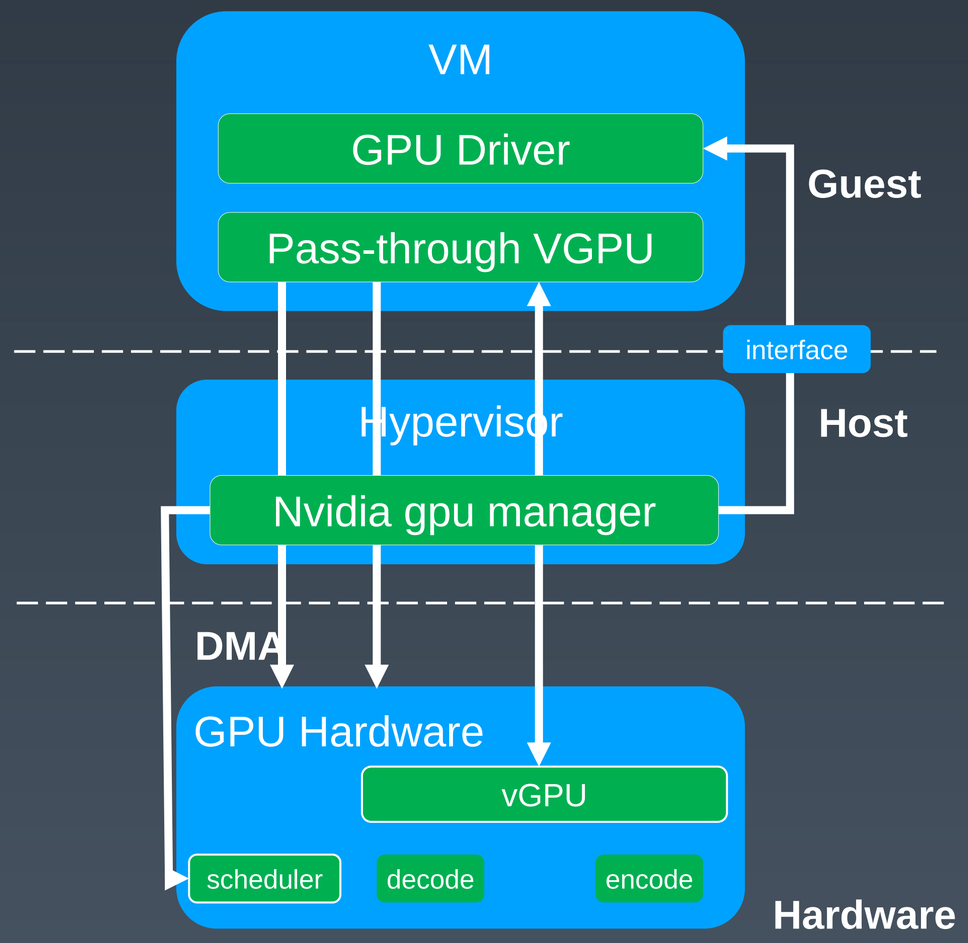
- 隔离性好
- 性能损耗低于10%

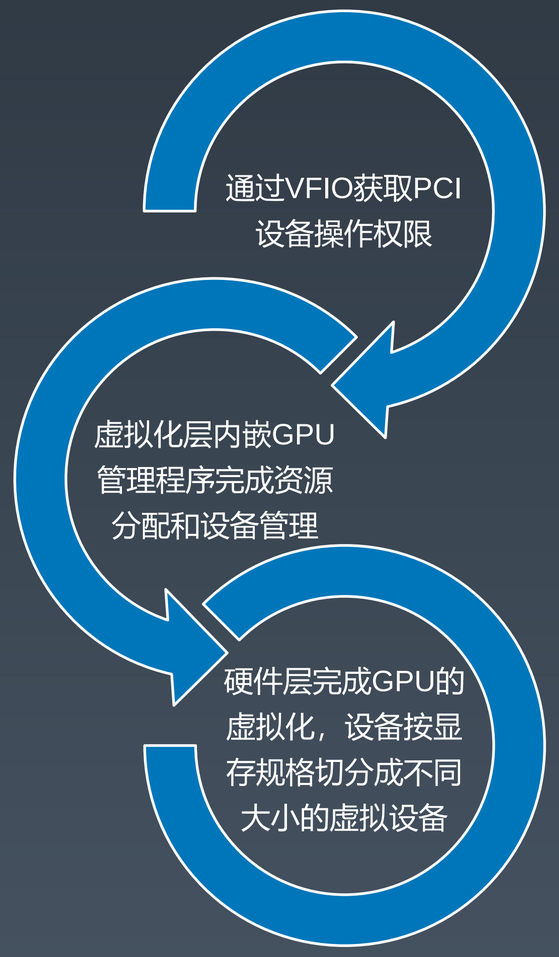
缺点

- 独占资源
- 不宜迁移
- 需要硬件进行支持

火龙果·整理
uml.org.cn

Nvidia vGPU





火龙果·整理
uml.org.cn

Nvidia vGPU

优点

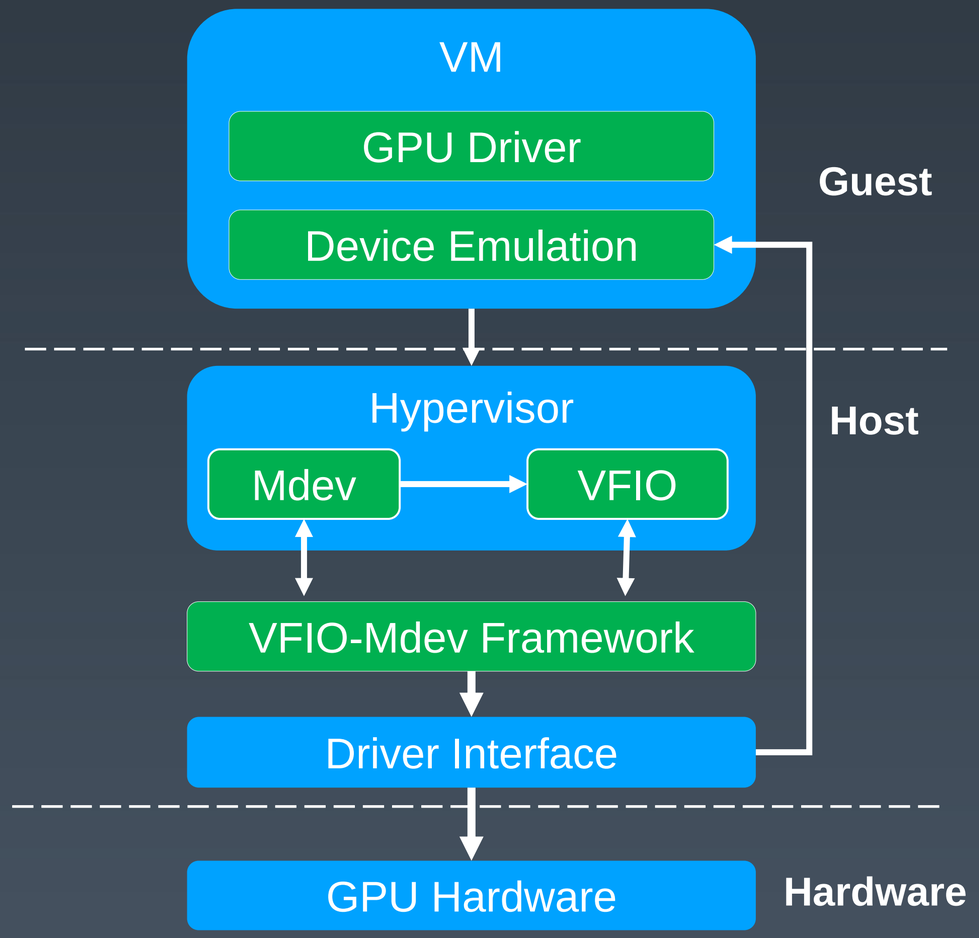
- 一虚多，资源合理利用
- 高性能计算下，性能损耗低于15%
- 可用于各个业务场景，兼容性好

缺点

- 资源隔离不完全
- 需要硬件进行支持

火龙果·整理
uml.org.cn

GPU虚拟化-模拟设备



实现

1、基于4.10内核添加GPU驱动程序
2、基于VFIO-Mdev生成中间的mediated device
3、mediated device提供用户态的接口，操作Mdev Bus
3、通过Mdev注册管理Pdev和Mdev
4、VFIO通过IOMMU管控控制设备IO
5、虚拟设备透传入虚拟机或者容器中

火龙果·整理
uml.org.cn

GPU虚拟化-模拟设备

优点

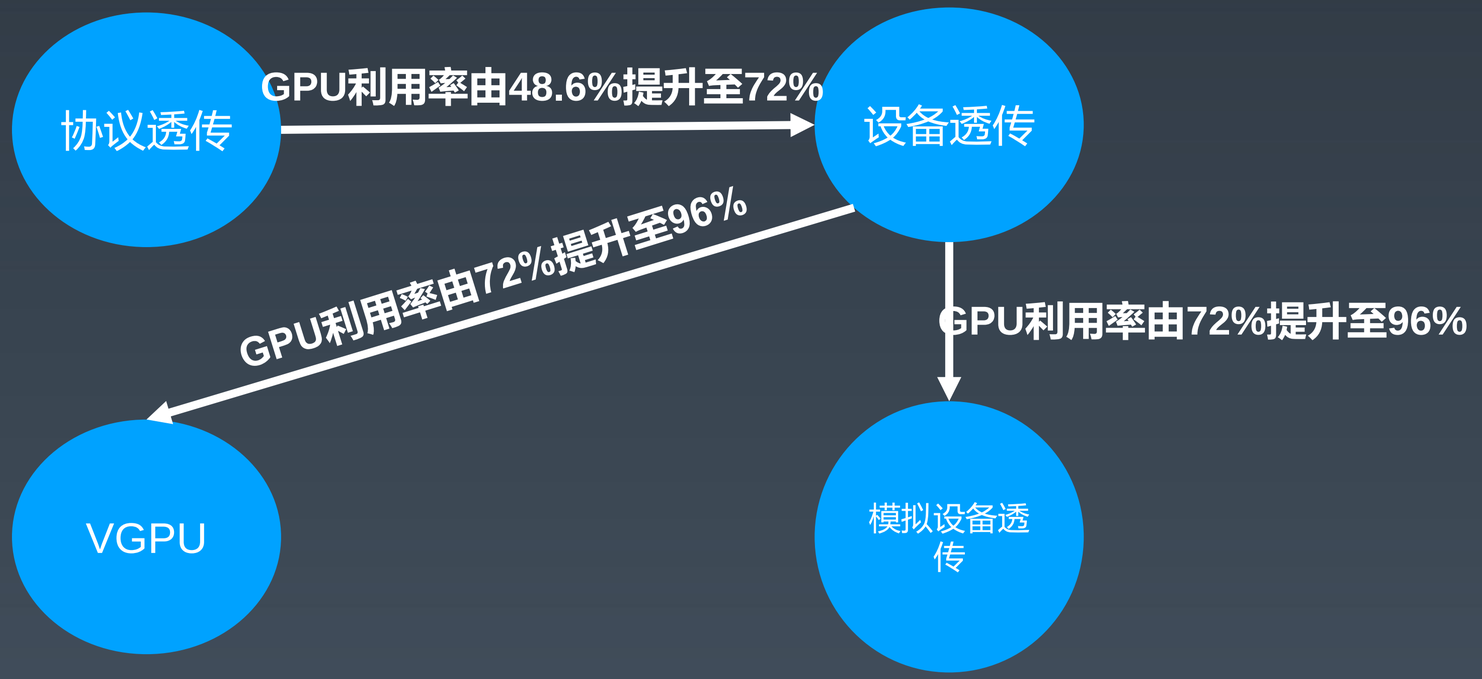
- 一虚多，资源合理利用
- 高性能计算下，性能损耗25%
- 可用于各个业务场景
- 兼容性好
- 基于VFIO，可统一设备驱动接口

缺点

- 资源隔离不完全
- 对内核版本要求较高
- 维护难度高，需要进行驱动和内核定制开发
- 性能相较VGPU方案，损失较大

火龙果·整理
uml.org.cn

总结



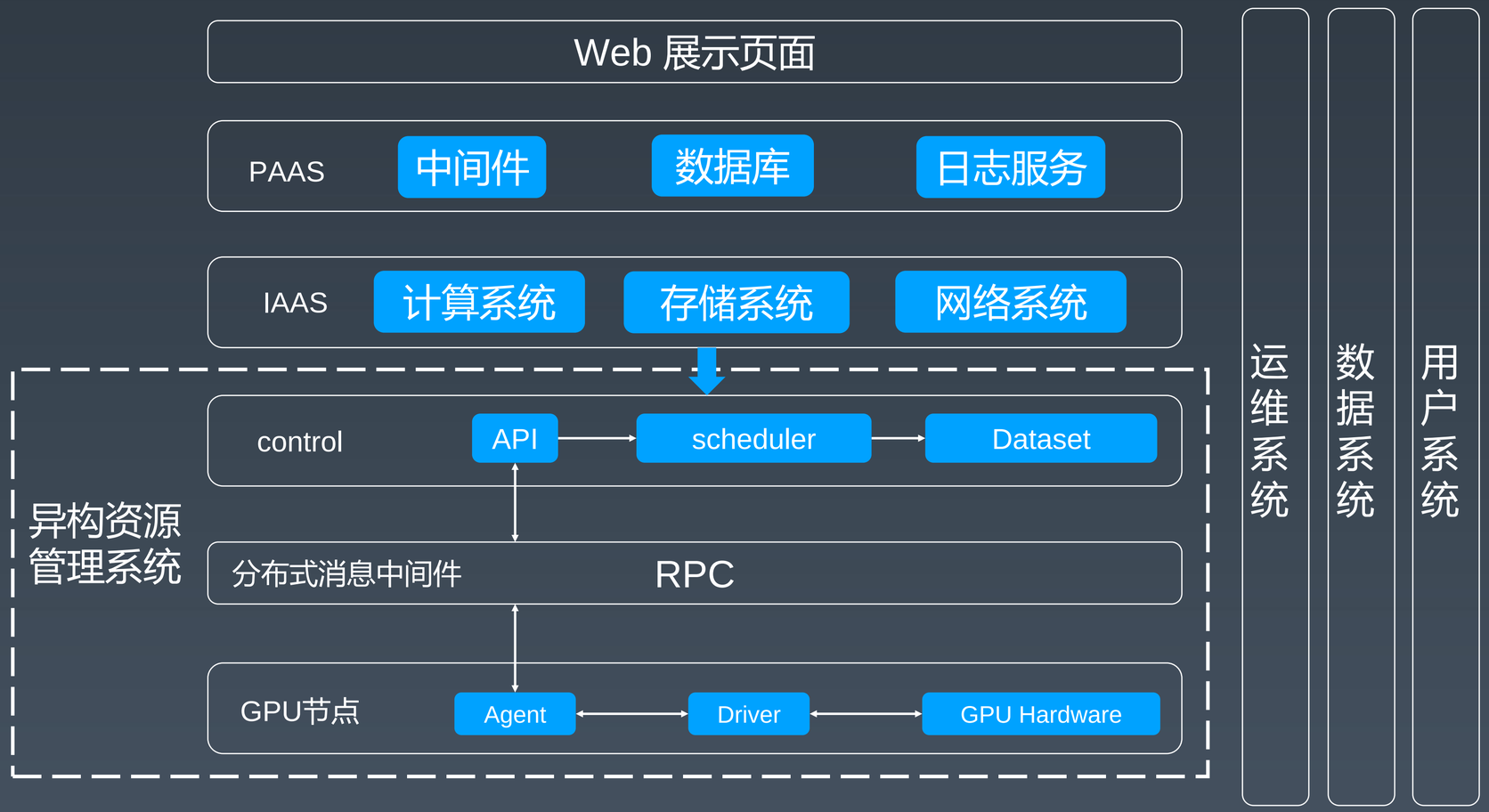
火龙果·整理
uml.org.cn

目录

- 讯飞AI业务的发展
- GPU虚拟化技术
- 异构资源管理
- 业务落地方案与实践

火龙果·整理
uml.org.cn

异构资源管理



火龙果·整理
uml.org.cn

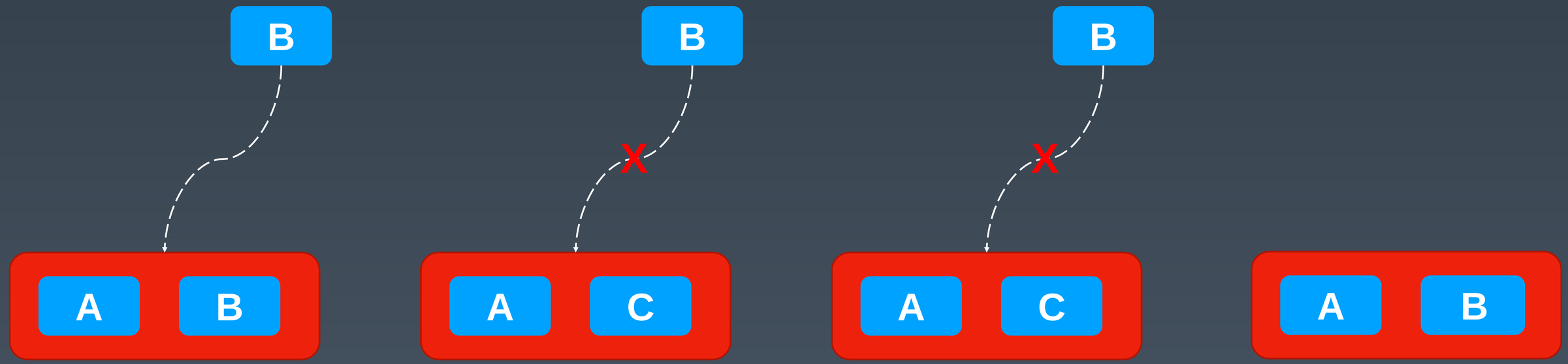
异构资源管理

B任务强烈依赖A任务处理后的数据

1、cache任务中绑定同一机器资源

2、cache任务中绑定同一机器资源失败

3、优先为B任务预留资源



火龙果·整理
uml.org.cn

异构资源管理

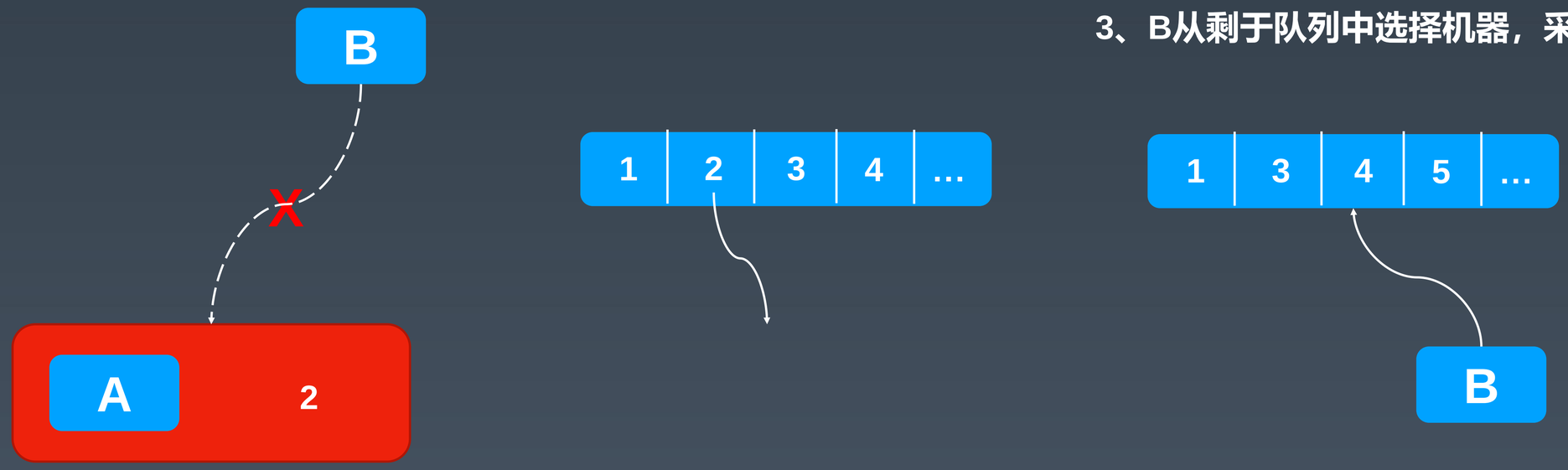
A　　B

AB任务无强烈依赖，但会抢走磁盘或者gpu算力需要分离

1、cache中A任务绑定一机器资源

2、cache中剔除之前机器列表

3、B从剩于队列中选择机器，采用资源均衡模式



火龙果·整理
uml.org.cn

目录

- 讯飞AI业务的发展
- GPU虚拟化技术
- 异构资源管理
- 业务落地方案与实践

火龙果·整理
uml.org.cn

业务落地方案于实践

问题

断点任务

批任务调度

火龙果・整理
uml.org.cn

断点任务

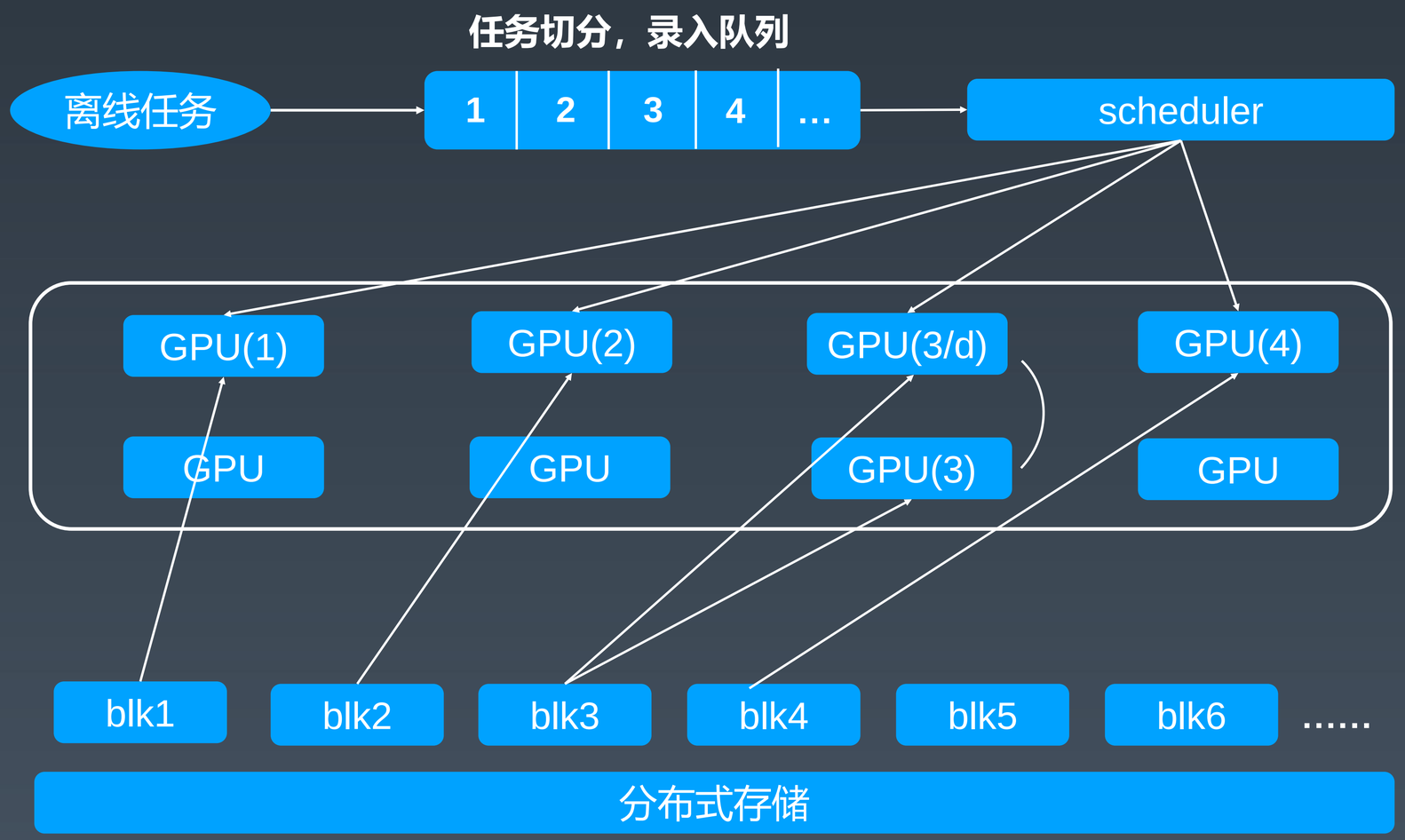
离线任务特点
- 资源大
- 任务集中
- 处理时间长
- 不需要太高的计算能力

对资源管控和调度的挑战
- 存储压力大
- 资源占用时间长
- 不易多类型集群调度

解决方案
- 分布式存储
- 分布式任务调度

火龙果·整理
uml.org.cn

断点任务



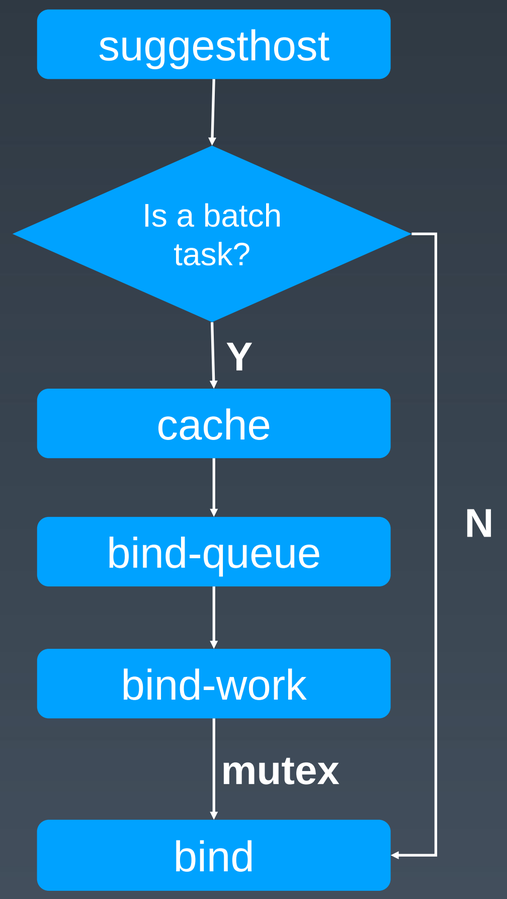
火龙果・整理
uml.org.cn

批任务调度

- 深度学习中经常会出现多机多卡的任务，也就是同时会起多个实例，多个实例属于同一个任务
- 默认调度器是一个一个进行调度的，只会检查单个实例资源够不够，这样前99个都能成功，最后一个pod调度失败。
- 这样就会造成
- 任务跑不了
- 前99个占着GPU不释放，新的任务无法调度
- 严重时整个集群死锁

火龙果·整理
uml.org.cn

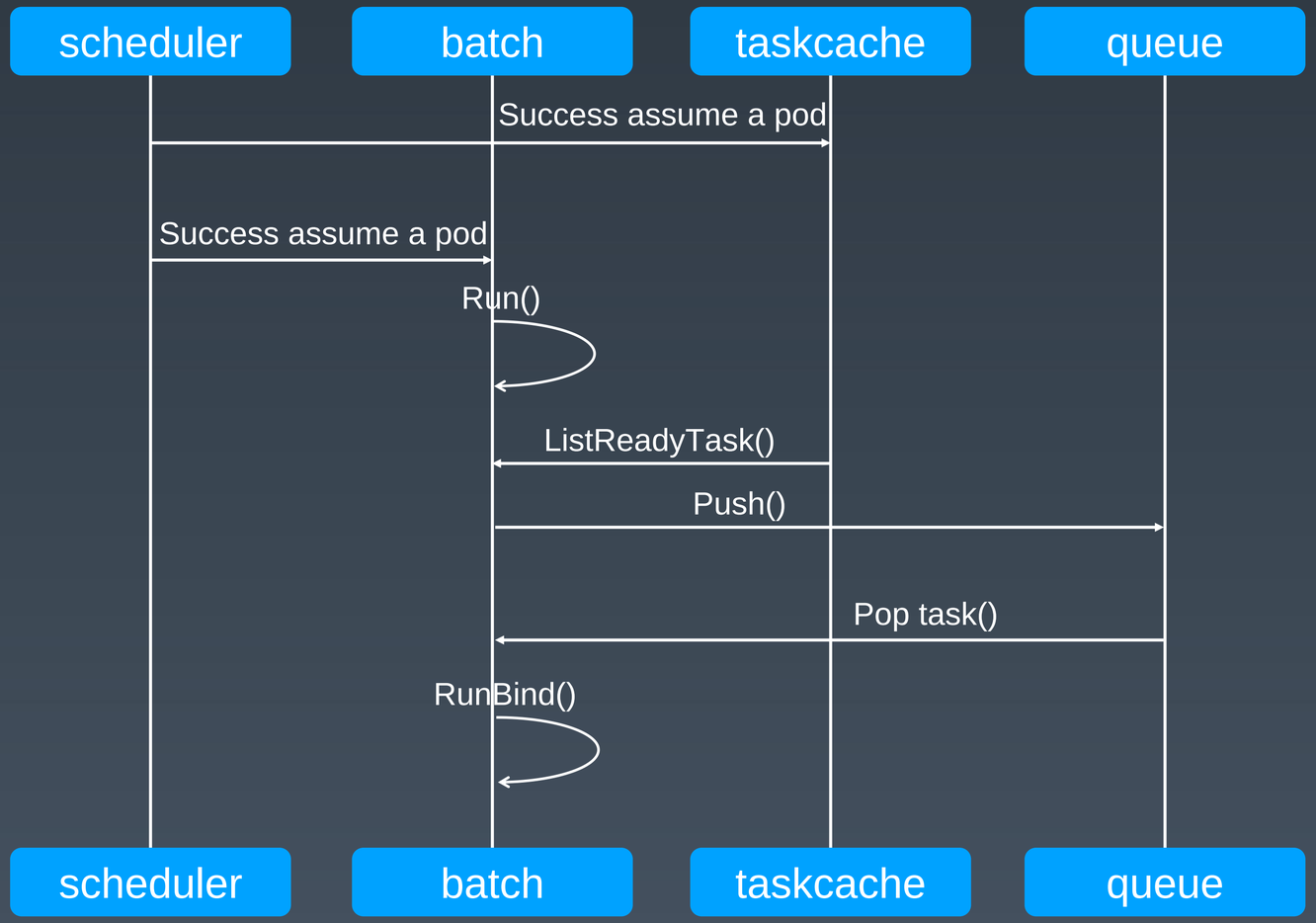
批任务调度-延迟绑定



- 如果是普通的pod，找到节点后assume就直接bind
- 如果是批处理任务，直接扔到批处理缓存中返回
- 有个协程一直检查批缓存中是否有成功的task (pod都齐了)
- 成功的task扔进binding队列，worker取成功的task进行批量绑定
- 绑定时与普通pod互斥

火龙果·整理
uml.org.cn

Batch-schedule执行流程



原生调度器在预选优选结束后交给Batch-scheduler处理

原生调度器中增加集群GPU资源检查Filter

Taskcache中的批任务pod都到齐了扔进Batch队列

Batch队列取批任务pod进行绑定

火龙果·整理
uml.org.cn

总结

讯飞在资源调度中解决的问题

- 实现6000多张物理，虚拟设备的混合管理和互调
- 实现了在线集群和离线集群的资源动态互调
- 实现了针对离线任务和在线任务的个性化调度需求

面临的挑战

- 模拟设备的性能优化
- 针对实时计算的调度能力
- 多异构资源的纳管，比如FPGA
- 个性化业务场景对调度和资源管控层面的需求

火龙果·整理
uml.org.cn

THANKS!