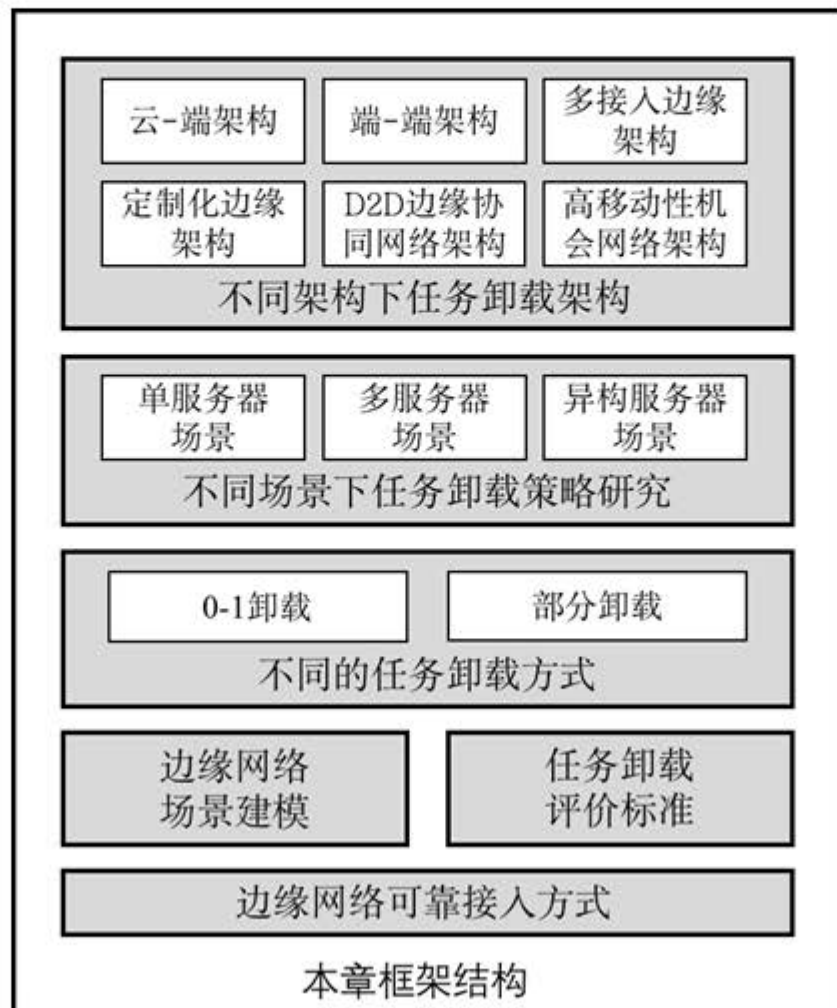


计算任务卸载

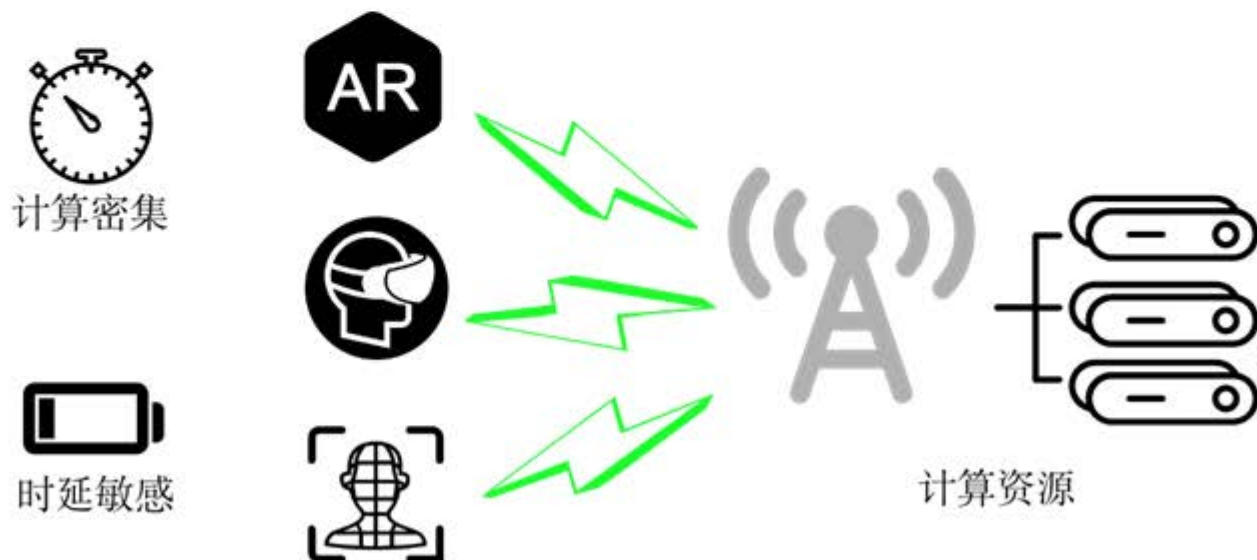
► 章节框架



- ▶ 任务卸载简介及其衡量指标
- ▶ 任务卸载方式
- ▶ 不同场景下任务卸载策略研究
- ▶ 不同架构下任务卸载策略研究
- ▶ 开源工具简介
- ▶ 边缘卸载策略展望及其挑战

- ▶ 任务卸载简介
- ▶ 边缘计算建模
- ▶ 边缘计算任务卸载评价指标

边缘计算的核心概念是将计算平台的相应功能从网络核心侧（云端）移动至网络接入侧为用户提供服务近距离服务，以减少终端用户时延，增强用户体验。



► 任务卸载的三个阶段：

- 任务上传：将任务通过移动接入点卸载至边缘服务器，根据用户服务的特点选择通信技术并且协调上传策略。
- 任务处理：服务器分配合理的资源以及处理时序对到达的任务进行处理。
- 结果回传：将计算结果回传到用户。

- ▶ 任务卸载简介
- ▶ 边缘计算建模
- ▶ 边缘计算任务卸载评价指标

任务传输建模:

	最大覆盖范围	工作频段	数据率
NFC	10 cm	13.56 MHz	106, 212, 414 Kbps
RFID	3 m	LF:120-134 KHz HF:13.56 MHz UHF:850-960 MHz	Low (LF) To High(UHF)
Bluetooth	100 m	2.4 GHz	22 Mbps
Wi-Fi	100 m	2.4GHz, 5GHz	135 Mbps
GSM	10 km	900-1800 MHz	14.4Kbps
3G	10 km	1.92-1.98 GHz	2Mbps
LTE	100 km	TDD:1850-3800 MHz FDD:700-2600 MHz	DL:300 Mbps UL:75Mbps

► 任务传输建模:

- 影响信道质量的因素为以下三点：多径衰落、信噪比和同频干扰。
- 采用香农公式去预估计信道理论的可靠传输速率。

► 任务处理建模:

- 在边缘网络中，边缘用户数量往往高于服务器数量，此时服务器中资源的竞争不可避免。
- 不同的用户将共享服务器中的CPU，高速缓存以及主存等资源，所分配资源的大小直接影响到服务处理时间。

► 任务处理建模需要考虑：

- 单个任务的处理时间建模
- 用户持续请求处理时间建模

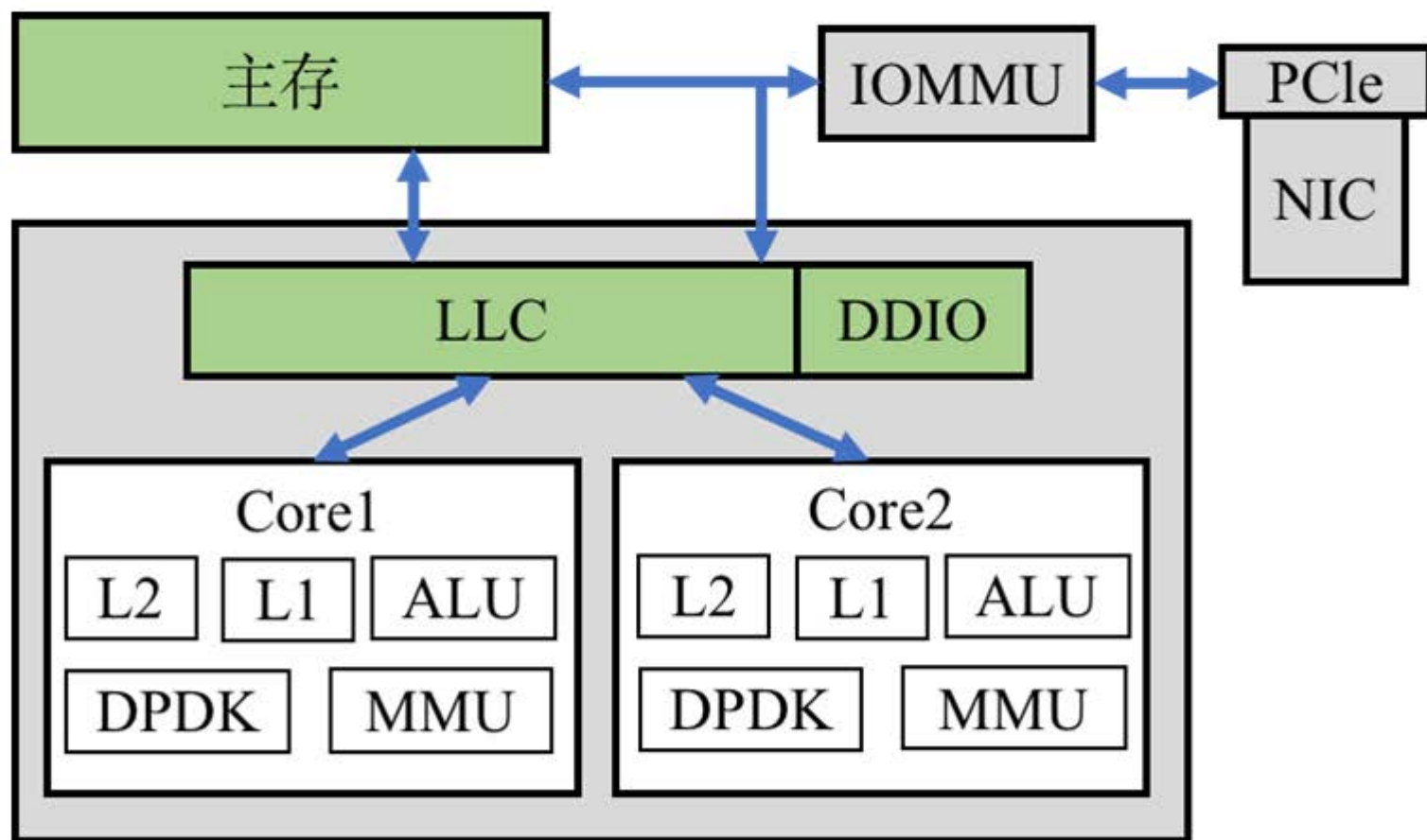
► 多服务竞争情况下的处理时间建模：

- 首先分析多服务竞争情况下，瓶颈资源在哪些位置。
- 不同的瓶颈资源会影响到哪些与处理时间相关的关键指标。
- 最后根据不同的关键指标进行处理时间的预测。

边缘计算建模 (4)

10

多服务竞争情况下的处理时间建模，瓶颈资源分析：



- ▶ **多服务竞争情况下的处理时间建模，竞争关键指标：**
 - 影响服务时间的三个“瓶颈”的位置，分别是第三级缓存、DDIO以及主存带宽资源，对三者的竞争进行量化。
- ▶ **多服务竞争情况下的处理时间建模，不同的关键指标下处理时间的预测：**
 - 根据竞争中的核心参数以及其相关度，找出核心参数与最终服务时间的关系，预测出服务处理的时间。

- ▶ 任务卸载简介
- ▶ 边缘计算建模
- ▶ 边缘计算任务卸载评价指标

► 服务时延和设备能耗

- 评价服务时延，对本地处理时延，边缘处理时延和云端处理时延进行对比，若边缘时延最优，则边缘卸载策略最优，能耗也如此。
- 在低功耗物联网场景中，存在着大量时延敏感型，计算密集型任务，此类任务需要同时考虑时延和能耗的问题。

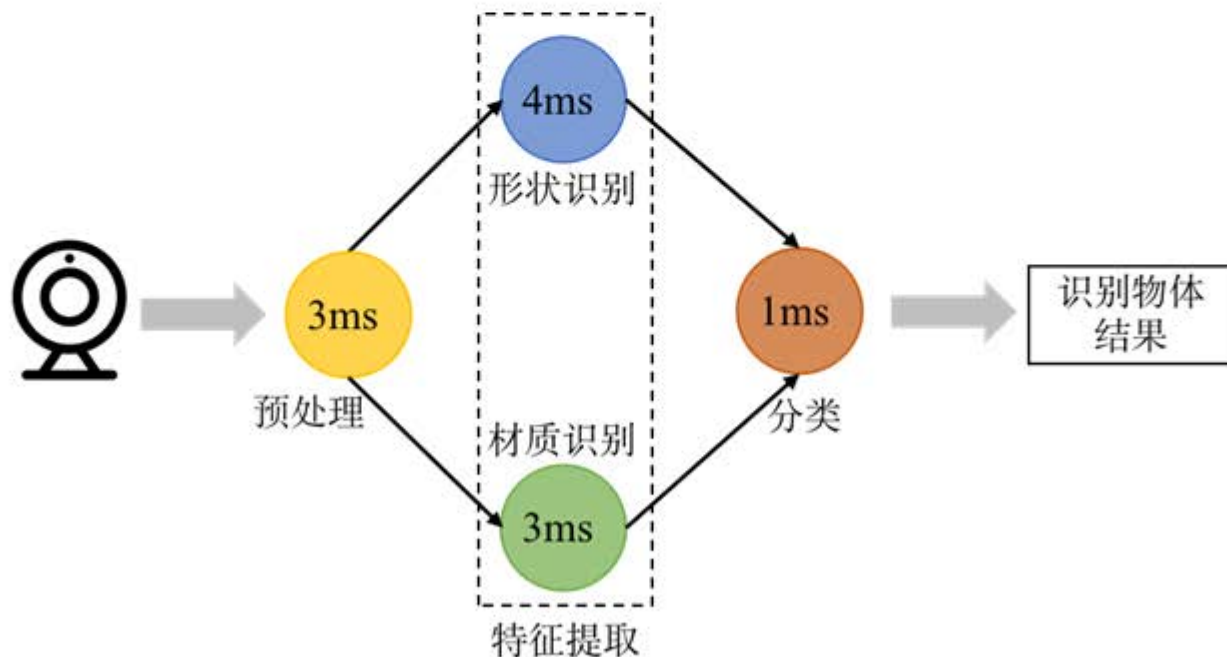
► 用户QoE

- 在流视频处理中，用户对视频的体验可以作为衡量边缘卸载策略的依据。
- 又如在AR场景中，可以通过用户的QoA来评价边缘卸载策略的好坏。

- ▶ 任务卸载简介及其衡量指标
- ▶ 任务卸载方式
- ▶ 不同场景下任务卸载策略研究
- ▶ 不同架构下任务卸载策略研究
- ▶ 开源工具简介
- ▶ 边缘卸载策略展望及其挑战

- ▶ 0表示任务在本地执行，1表示在边缘网络中执行。
- ▶ 任务通常是有一个三元组进行表示：<任务输入大小；任务截止时间；计算复杂度>。
- ▶ 用户之间的竞争增加了决策的复杂度：
 - 当用户数大于服务器数量时，用户需要共享服务器的计算资源，使得计算时间增加。
 - 同一通信范围的用户，会造成一定的通信干扰，增加了数据传输的时间。
 - 边缘环境的动态变化，很难设计中心式的卸载策略。

- ▶ 部分卸载可以从两个方面进行理解，即数据分解和任务分解。
 - 数据分解是将任务的输入进行拆分。
 - 任务分解将一个任务拆分为多个子任务。任务分解示意图如下：



► 任务分解需要克服的挑战:

- 我们需要分析任务的特点并将任务合理的划分为多个子任务。
- 如何将一系列相互关联的子任务卸载至边缘服务器。
- 多用户子任务卸载策略。

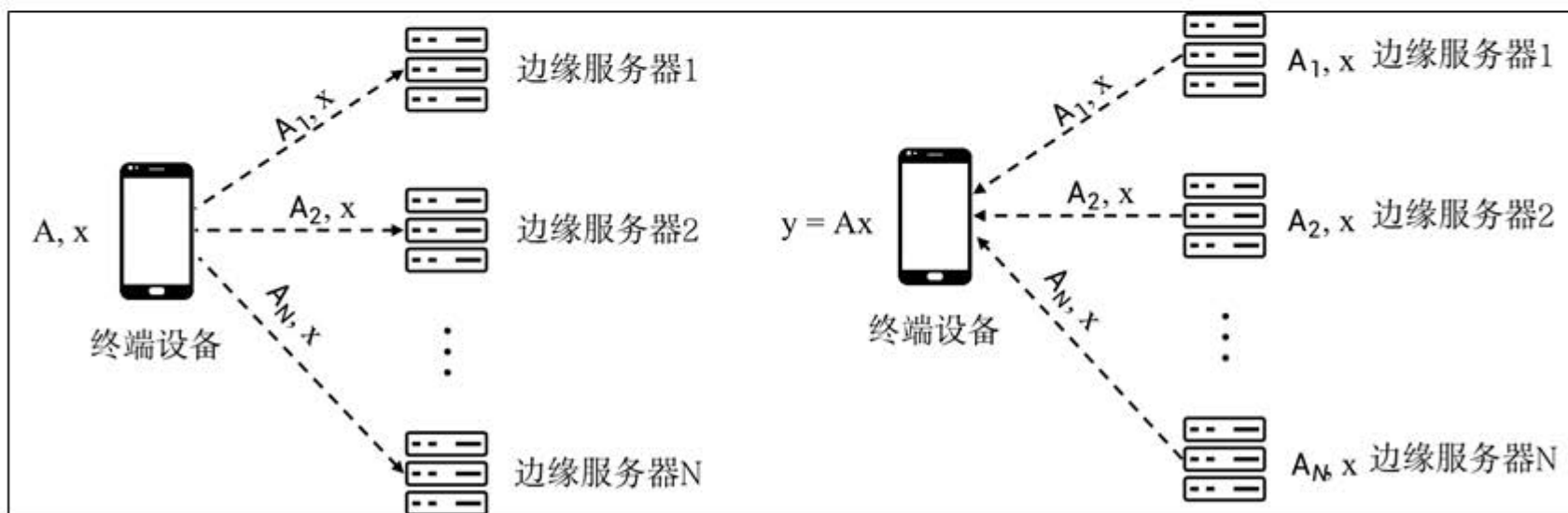
- ▶ 任务切分技术及实例分析
- ▶ 子任卸载策略

► 任务切分：CloneCloud和MAUI

- CloneCloud使用静态分析和动态分析的组合，以精细的粒度自动对目标任务进行分块，同时优化了计算和通信所需要的时间和能耗。
- MAUI是一种可将细粒度的能量感知型移动代码卸载到基础结构上的系统；它通过细粒度的代码卸载最大程度地节省了能源，同时最大程度地减少了对应用程序的更改。
- MAUI提供了四个组件：探查器和服务器端代理、周期性地求解线性方程的决策引擎、MAUI协调器。

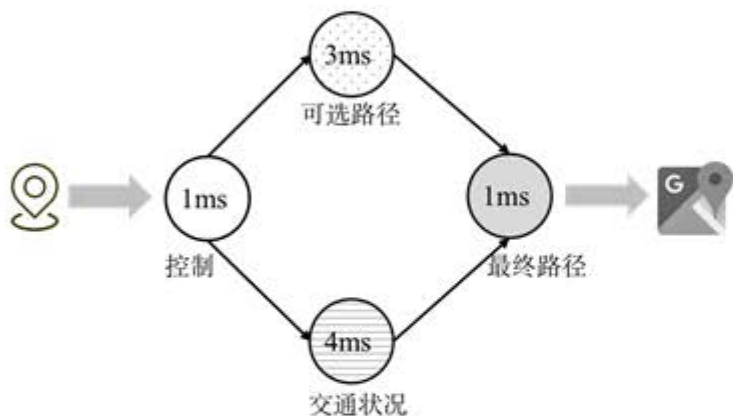
► 应用切分实例

- 较为广泛的基础应用切分方式：矩阵算法中的任务切分：

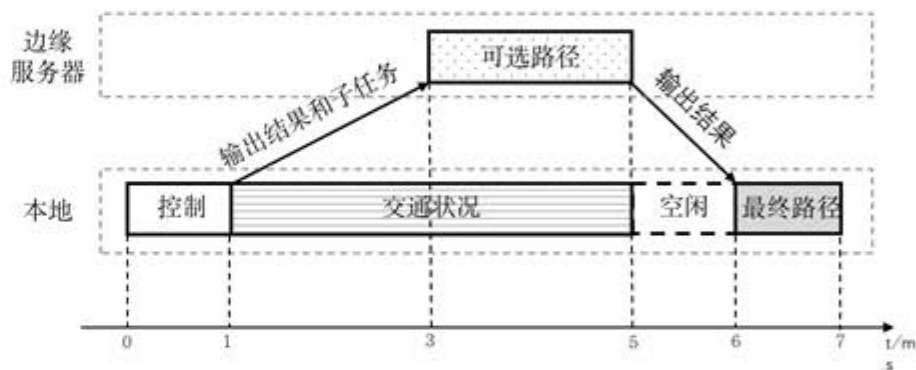


- ▶ 任务切分技术及实例分析
- ▶ 子任卸载策略

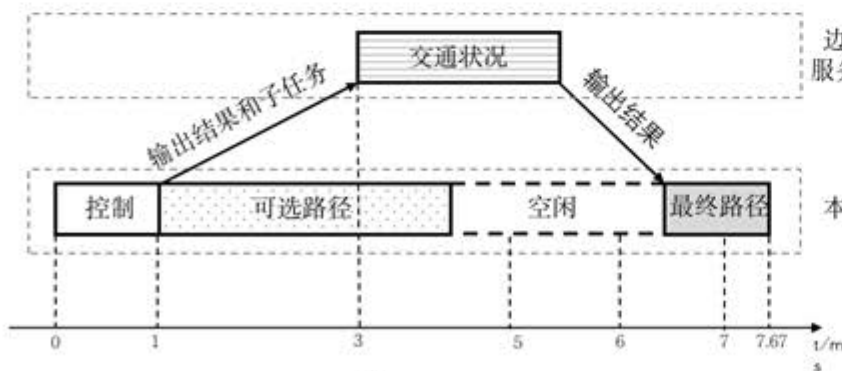
子任务卸载策略：



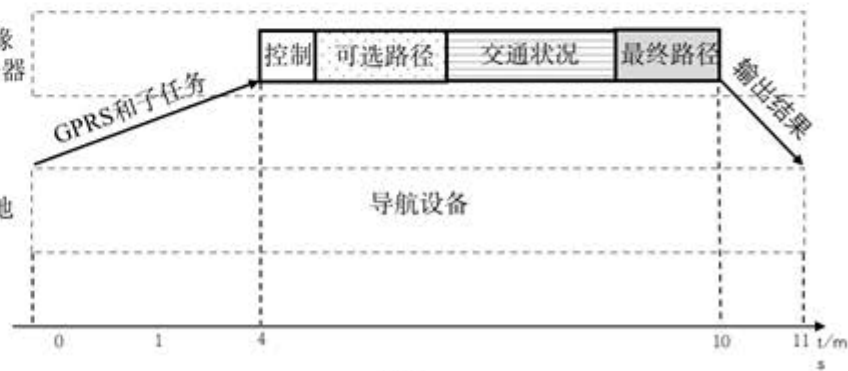
(a)



(b)



(c)



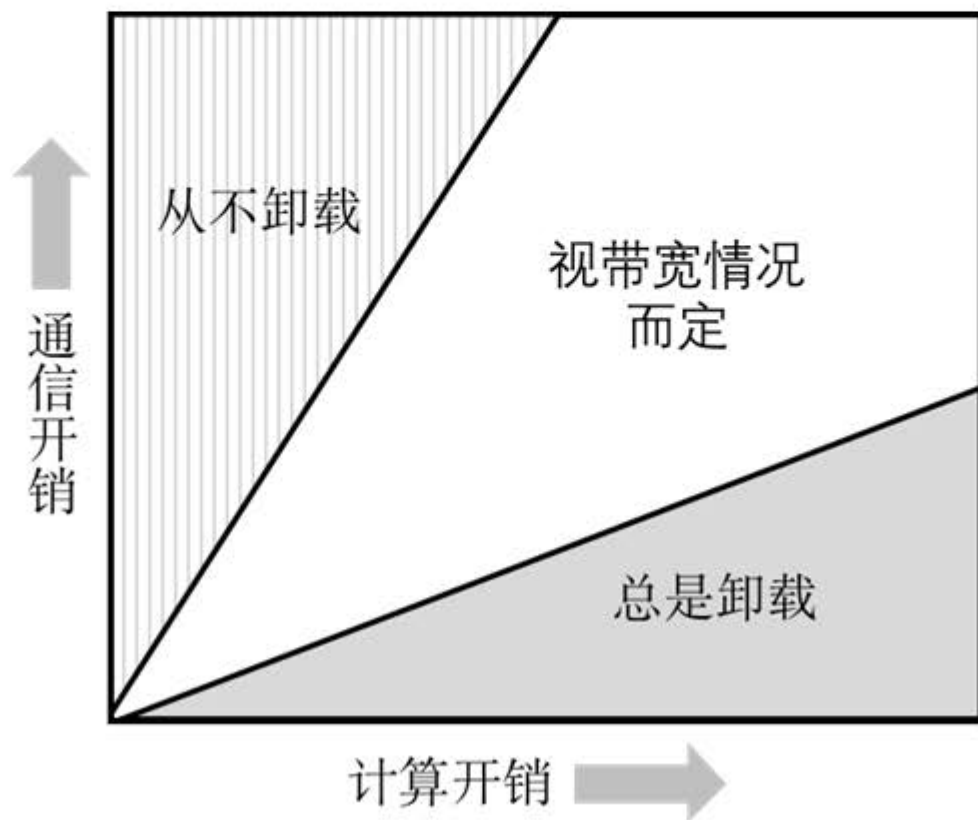
(d)

- ▶ 任务卸载简介及其衡量指标
- ▶ 任务卸载方式
- ▶ 不同场景下任务卸载策略研究
- ▶ 不同架构下任务卸载策略研究
- ▶ 开源工具简介
- ▶ 边缘卸载策略展望及其挑战

- ▶ 单用户边缘网络场景
- ▶ 多用户边缘网络场景
- ▶ 异构服务器边缘网络场景

► 0-1卸载方式

- 0-1卸载需要考虑时延大小、能量的开销等影响因素



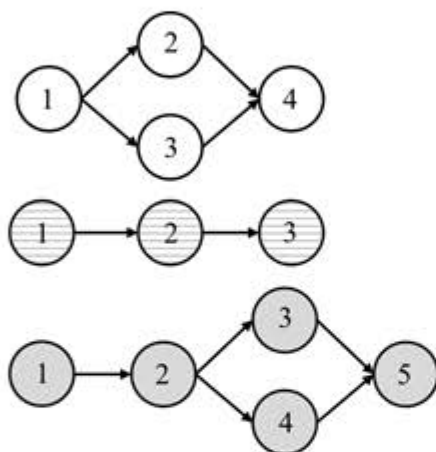
► 部分卸载方式

- 综合考虑信道质量、传输功耗、边缘服务器的运算能力和子任务之间的依赖情况，来决定各个子任务是否需要上传，并在能耗开销一定的情况下最小化任务执行时间。

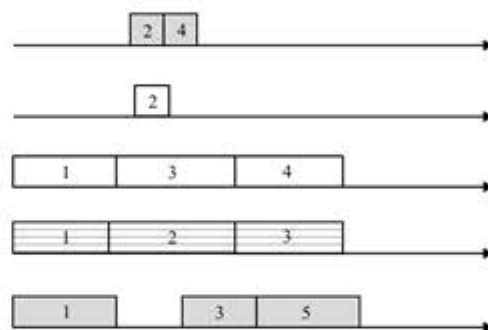
- ▶ 单用户边缘网络场景
- ▶ 多用户边缘网络场景
- ▶ 异构服务器边缘网络场景

- ▶ 多个用户将会竞争有限的传输资源以及计算资源。
- ▶ 多用户场景研究的切入点分为两个方面：传输资源与计算资源联合调度；边缘服务器任务调度。
- ▶ 可采用分布式和集中式两种方案解决将有限的无线传输资源和计算资源合理地分配给多个用户。

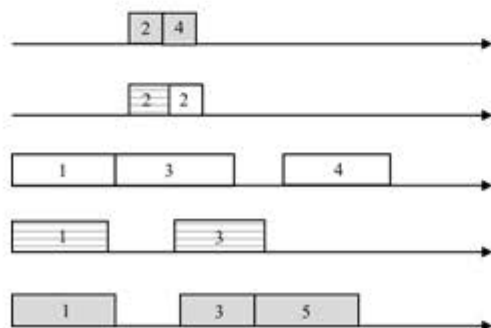
分布式博弈的方法在多用户场景中的应用：



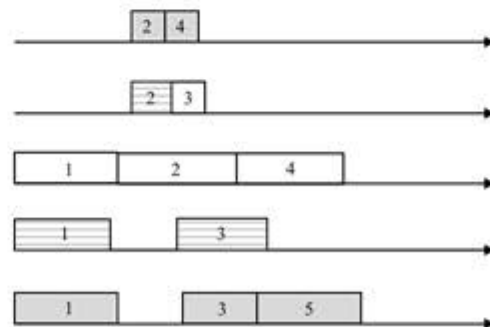
(a) 应用的DAG



(b) 第二次迭代



(b) 第三次迭代

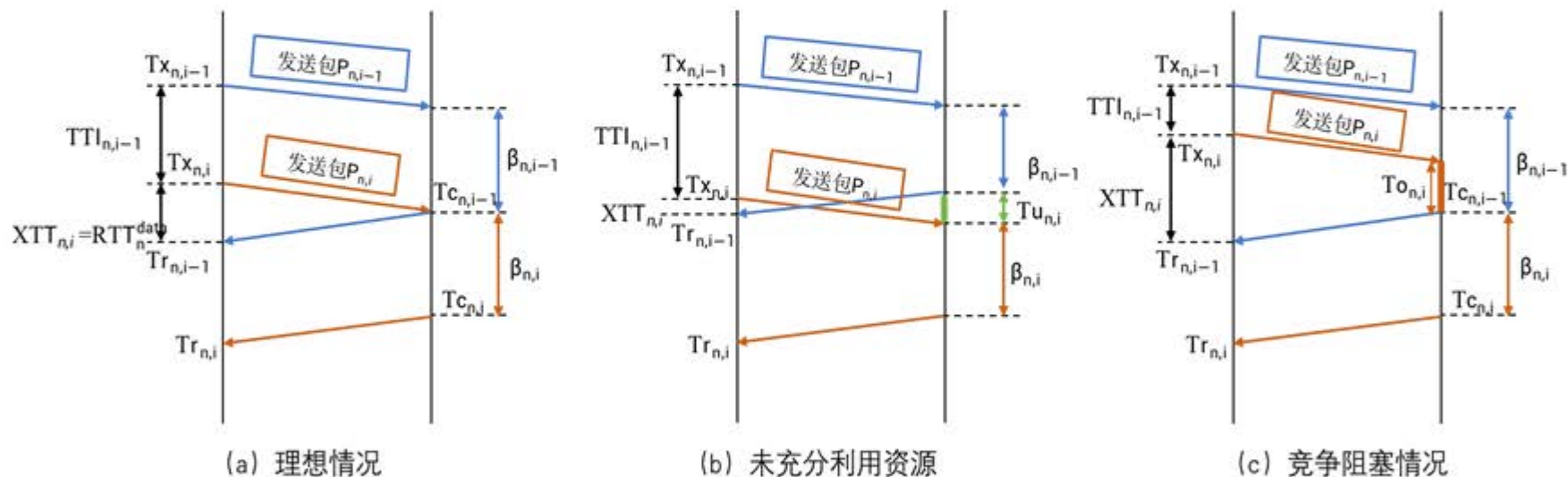


(d) 最终状态

- ▶ 单用户边缘网络场景
- ▶ 多用户边缘网络场景
- ▶ 异构服务器边缘网络场景

- ▶ **异构网络可以从两方面进行考虑：**
 - ▶ 服务器计算能力，服务器网络中包括一个中央云服务器和多个计算能力相异的边缘服务器。
 - ▶ 接入边缘网络方式的异构性，包括宏站的接入，小型蜂窝网的接入，室分基站的接入，WiFi节点的接入，或者是中继节点的接入。

卸载策略，接入方式，终端设备能耗控制是关键。异构网络中的任务卸载策略：



- ▶ 任务卸载简介及其衡量指标
- ▶ 任务卸载方式
- ▶ 不同场景下任务卸载策略研究
- ▶ 不同架构下任务卸载策略研究
- ▶ 开源工具简介
- ▶ 边缘卸载策略展望及其挑战

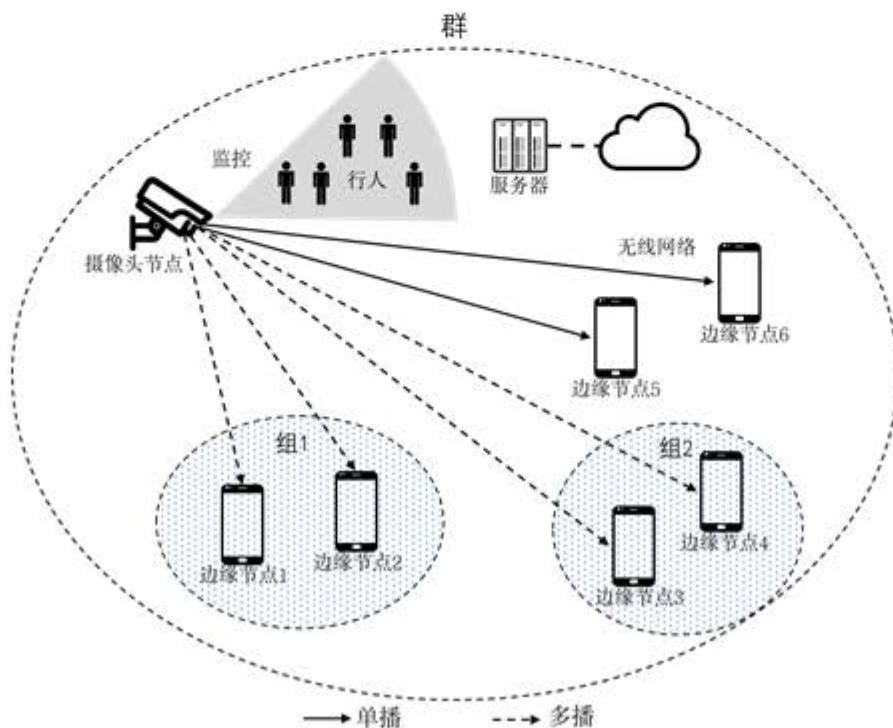
- ▶ 云-边架构
- ▶ 边-边架构
- ▶ 多接入边缘架构
- ▶ 定制化边缘架构
- ▶ 用户高移动性网络场景
- ▶ 雾计算网络场景

- ▶ **云-边架构一般分为三层：云服务中心网络、边缘网络、终端用户设备**
 - 在0-1卸载方式下，此三层架构中我们需要根据各层的计算能力，层间的传输能力，设定一定的阈值，以指导云或者边的卸载决策。
 - 在部分卸载策略下同样按照上述规则，由于需要考虑整个DAG在云-边-端网络中的部署，将一些与本地交互频繁的子任务部署在边缘网络中，以缩短两者交互的时间。

- ▶ 云-边架构
- ▶ 边-边架构
- ▶ 多接入边缘架构
- ▶ 定制化边缘架构
- ▶ 用户高移动性网络场景
- ▶ 雾计算网络场景

► 边-边架构主要针对延迟敏感型任务提出，其独特的设计特点和需要解决的诸多挑战：

- 边-边架构着重于“边-端”的协同
- 边-边架构着重于“边-边”的协同



► 重要节点介绍:

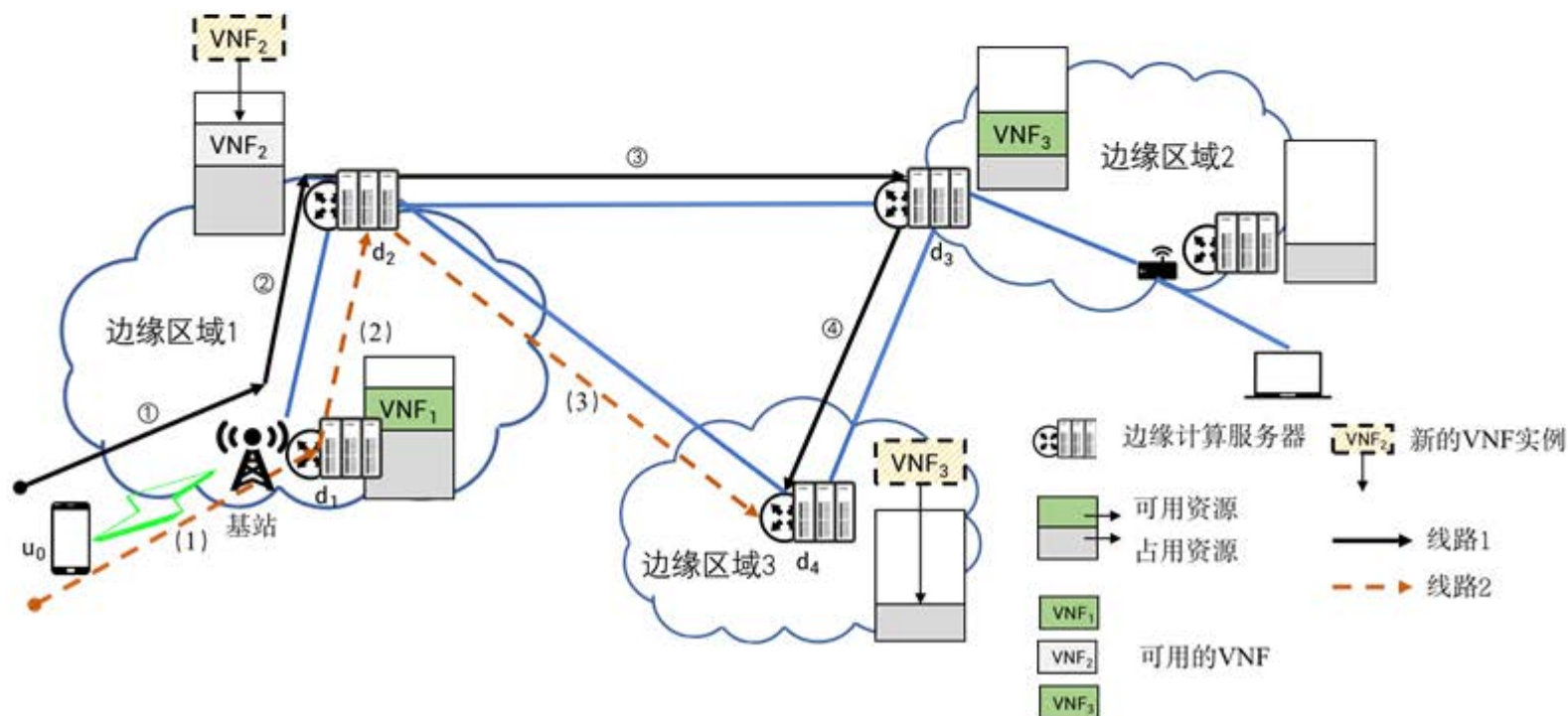
- 视频采集节点
- 边缘节点组
- IoT服务器

- ▶ 云-边架构
- ▶ 边-边架构
- ▶ 多接入边缘架构
- ▶ 定制化边缘架构
- ▶ 用户高移动性网络场景
- ▶ 雾计算网络场景

- ▶ 将移动边缘计算的接入方式从蜂窝网扩展至其它无线接入网络。
- ▶ 终端设备在同一地点可以有多个接入边缘网络的选择，在缓解无线传输侧的通信压力的同时，也提高了服务器部署的灵活性。
- ▶ 典型卸载策略：
 - 当场卸载策略：默认状态下的配置，当用户所在位置没有WiFi节点可以接入时，将通过蜂窝数据网络上传任务。
 - 在延迟卸载策略：当用户当前位置只有蜂窝网，通过预测用户轨迹，发现即将到达WiFi覆盖之处，当用户进入WiFi覆盖范围后延迟卸载。

- ▶ 云-边架构
- ▶ 边-边架构
- ▶ 多接入边缘架构
- ▶ 定制化边缘架构
- ▶ 用户高移动性网络场景
- ▶ 雾计算网络场景

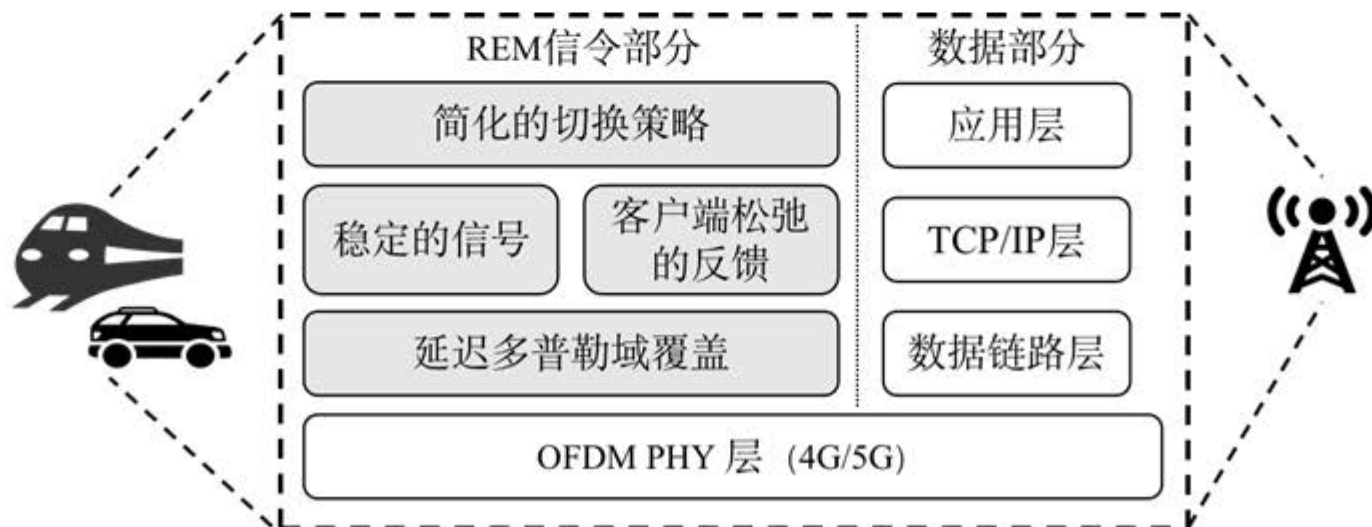
- 定制化的服务按照服务特定的需求，部署特定的服务于边缘网络中，用户在卸载过程中，只需上传相应的服务输入和服务要求。有限的资源和定制化的服务数量间的关系成为了亟需解决的首先矛盾。
- 定制化边缘网络中的任务卸载



- ▶ 云-边架构
- ▶ 边-边架构
- ▶ 多接入边缘架构
- ▶ 定制化边缘架构
- ▶ 用户高移动性网络场景
- ▶ 雾计算网络场景

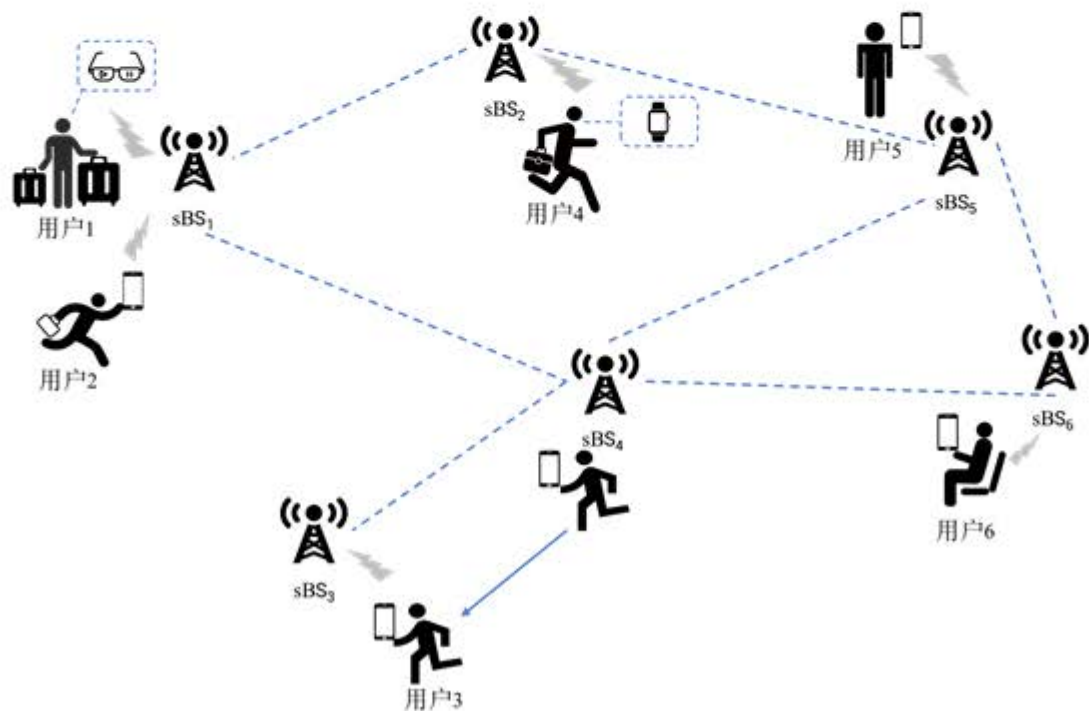
► 高移动场景中用户接入策略

- 高速铁路的4G LTE，移动性事件发生的非常频繁，且非常不稳定，平均基站之间的切换每11到20秒发生一次。
- 4G/5G在极端移动性场景下不可靠的根本原因是其控制策略基于无线信号强度的设计，通过主动策略来减轻故障。

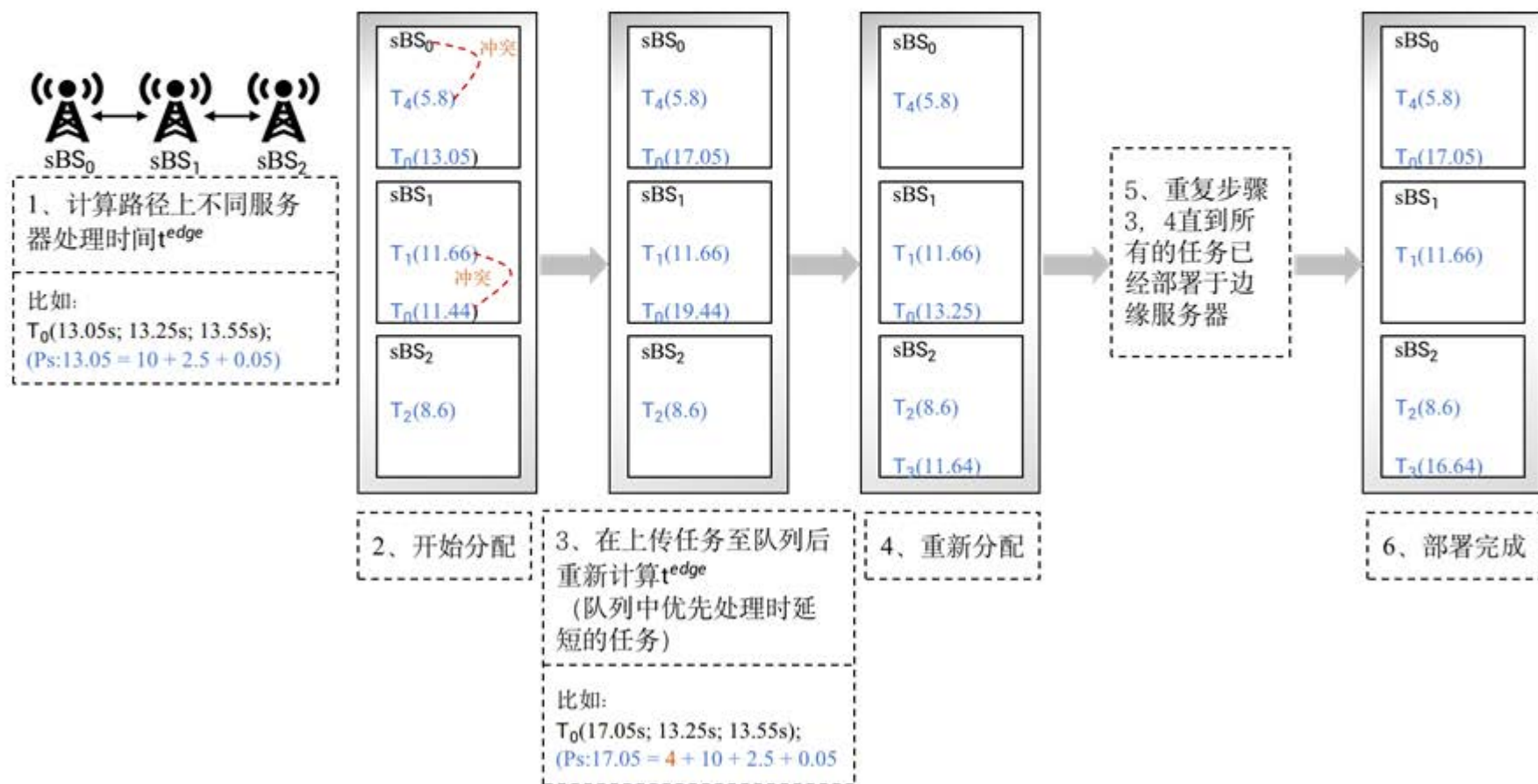


► 高移动场景中任务卸载策略

- 用户的高移动性首先移动用户导致边缘服务器工作负载实时变化，其次，其会涉及到边缘网络内的通信开销和计算资源分配问题。



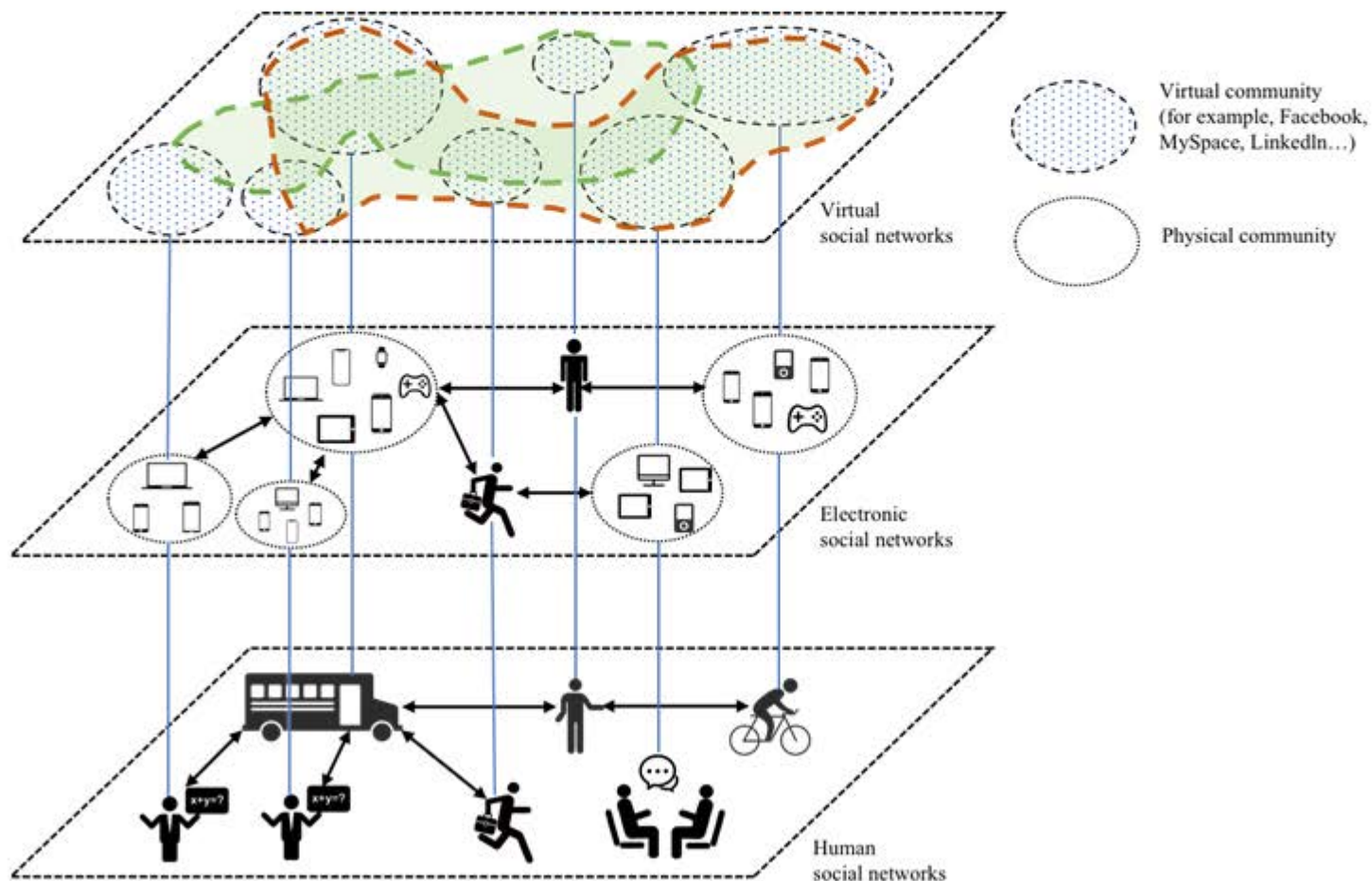
高移动场景下多任务卸载案例：



用户高移动性网络场景 (4)

47

自组织机会网络:



- ▶ 云-边架构
- ▶ 边-边架构
- ▶ 多接入边缘架构
- ▶ 定制化边缘架构
- ▶ 用户高移动性网络场景
- ▶ 雾计算网络场景

► 雾计算的定义：

- 将一些计算密集型任务分散到计算能力较弱且分布零散的设备上。
- 雾计算中的计算节点指的是位于终端设备和云端之间一切具有计算能力的设备。

► 雾计算的优势：

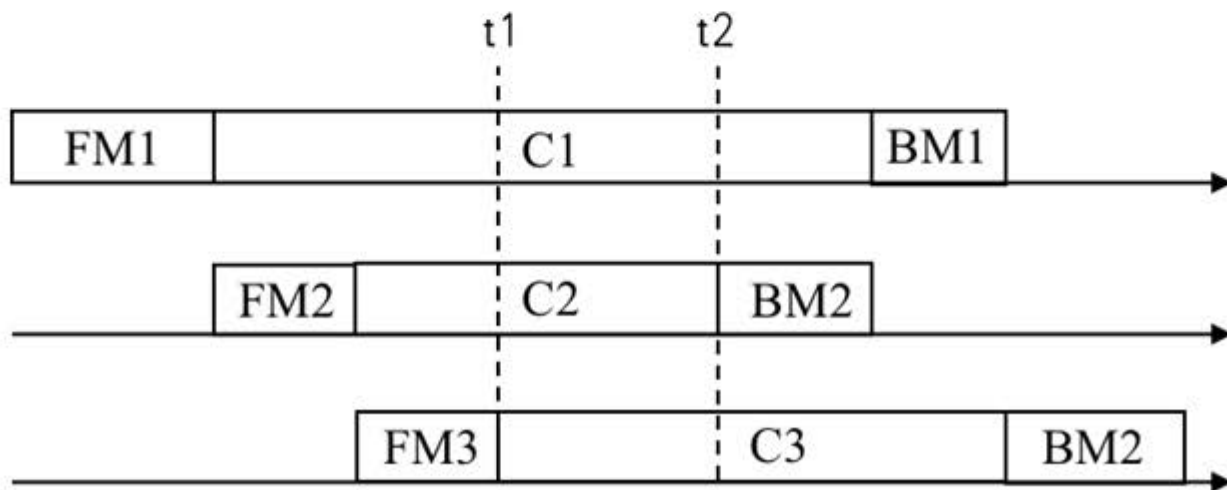
- 在D2D网络中，计算需求会随着用户数量增加而增多，但是相应的来自移动设备额外的计算能力也会相应提升。
- 在D2D网络中，设备间的近距离通信不仅充分复用了频谱资源，同时降低了数据交互的能耗。

雾计算和边缘计算的区别:

	雾计算	边缘计算
节点类型	路由器, 交换机, 接入节点, 网关	部署于基站处的服务器
节点位置	终端设备和云之间任意位置	无线接入网控制器 / 基站
软件构架	雾计算编排器	边缘计算编排器
管理策略	中心式	分布式
接入方式	蓝牙, Wi-Fi, 移动通信网络	移动通信网络
通信方式	支持多跳	单跳
节点间通信	支持	部分支持

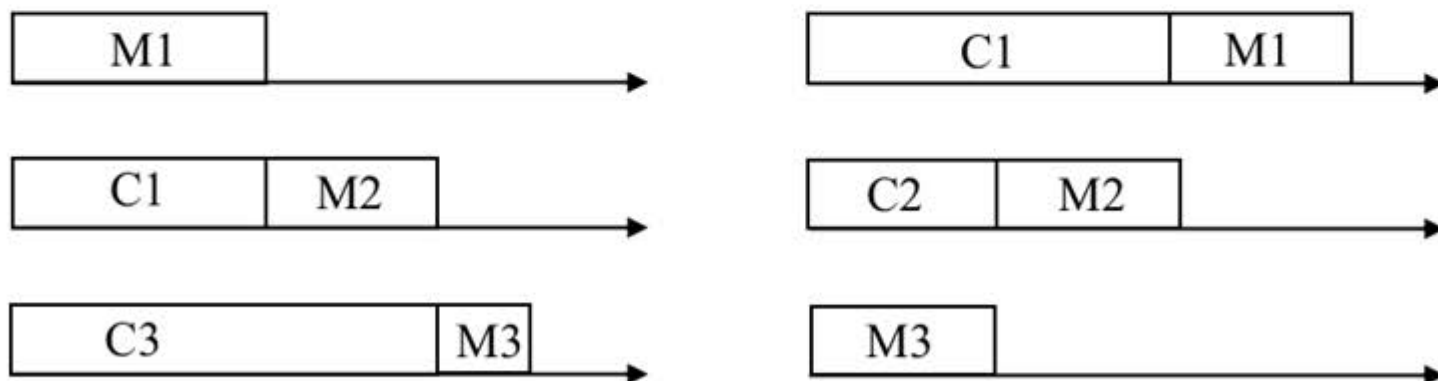
► 雾计算中的任务卸载:

- 最佳通信调度所应该满足的特性：（1）非抢占式通信调度策略是最佳的；
（2）在所有用户上传任务之后才进行任务结果回传；（3）所有用户上传任务的时间（结果回传时间）之间不应该有空闲的时间片。
- 在满足最佳通信调度特征情况下计算负载分配方法：



► 雾计算中的任务卸载：

- 最优的通信时序策略：（1）当计算节点计算能力相同，但是通信延迟不同的情况；2）当节点间传输时延相等，但节点计算能力不同的情况；（3）节点计算能力和传输能力都不同的情况：需要设计启发式算法提高策略规划时间。

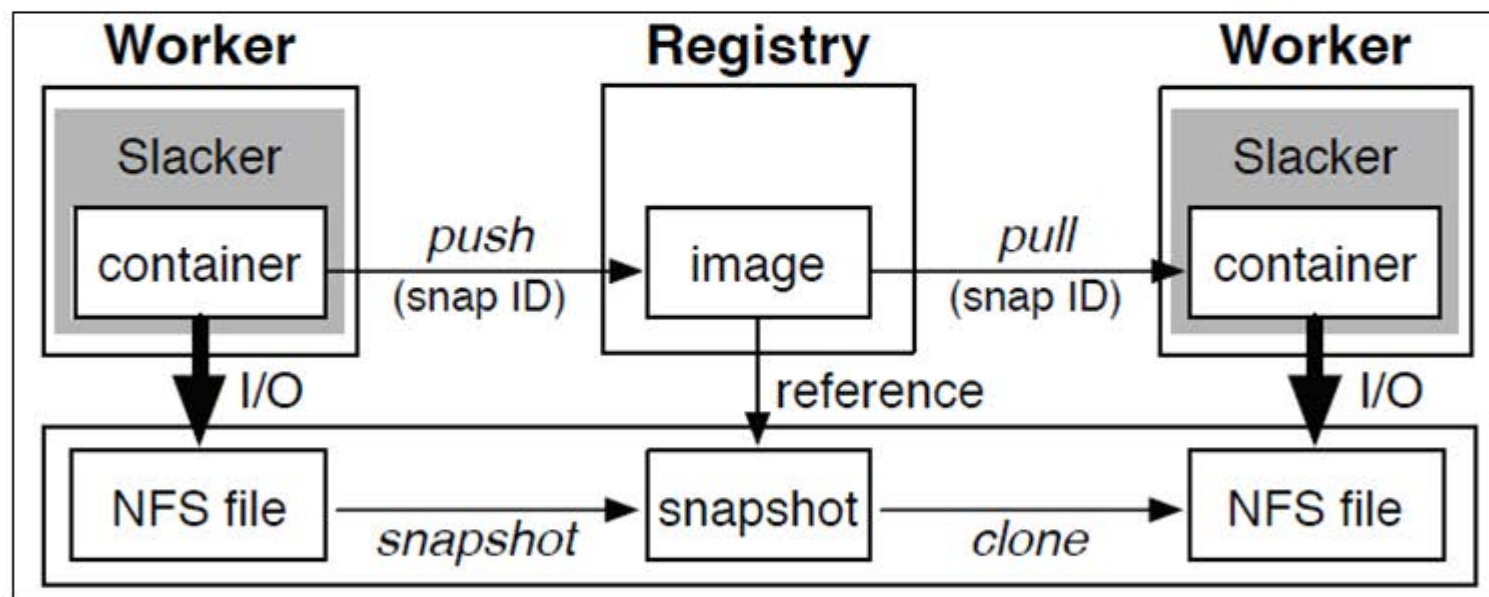


► 雾计算中的任务卸载:

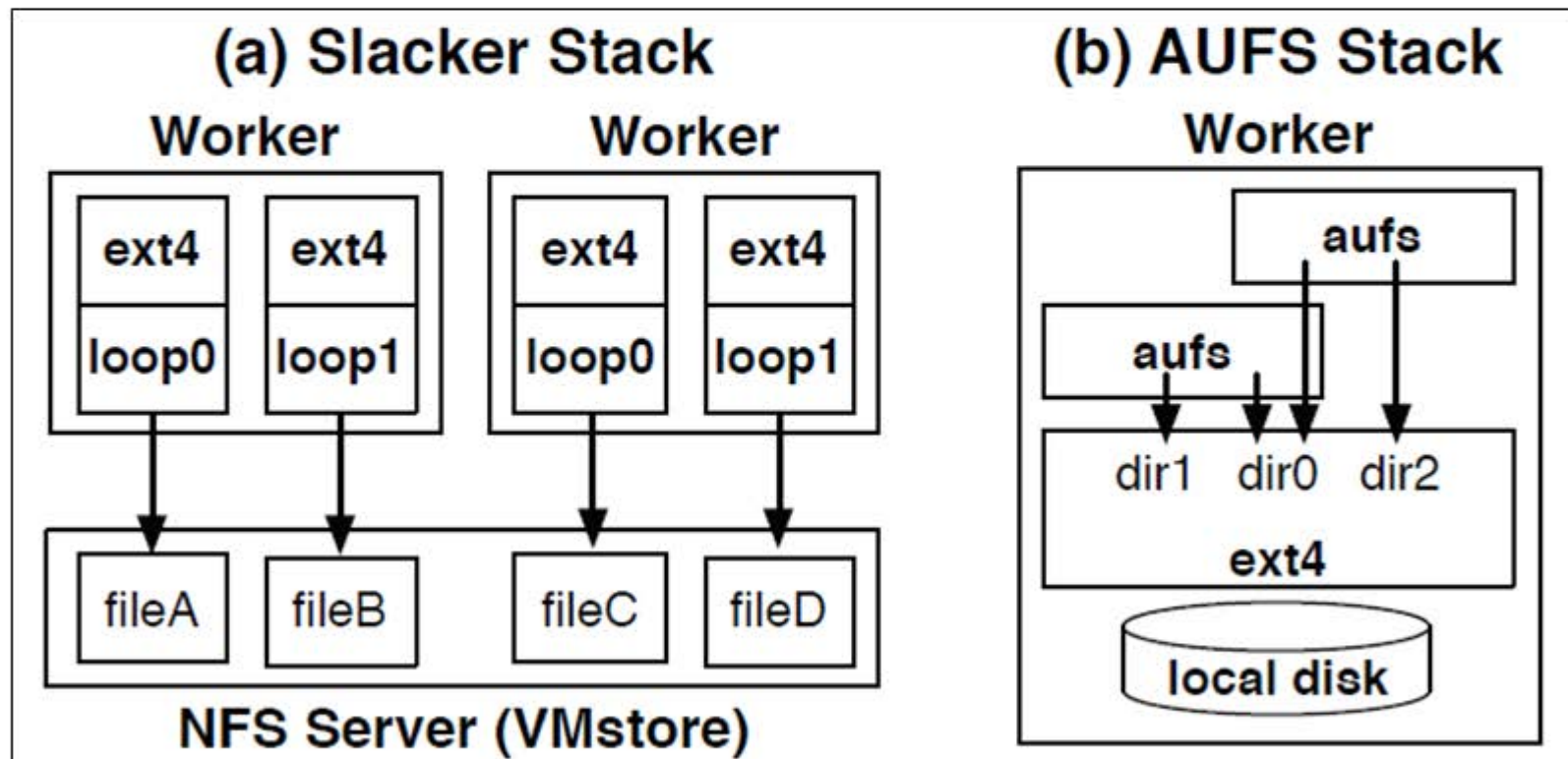
- 卸载节点数的选择: 依次去掉计算能力最弱的或者通信时间最长的节点, 如果总的服务处理时间不变的, 则在卸载节点集合中去掉该节点, 直到得到最小的子集。

- ▶ 任务卸载简介及其衡量指标
- ▶ 任务卸载方式
- ▶ 不同场景下任务卸载策略研究
- ▶ 不同架构下任务卸载策略研究
- ▶ 开源工具简介
- ▶ 边缘卸载策略展望及其挑战

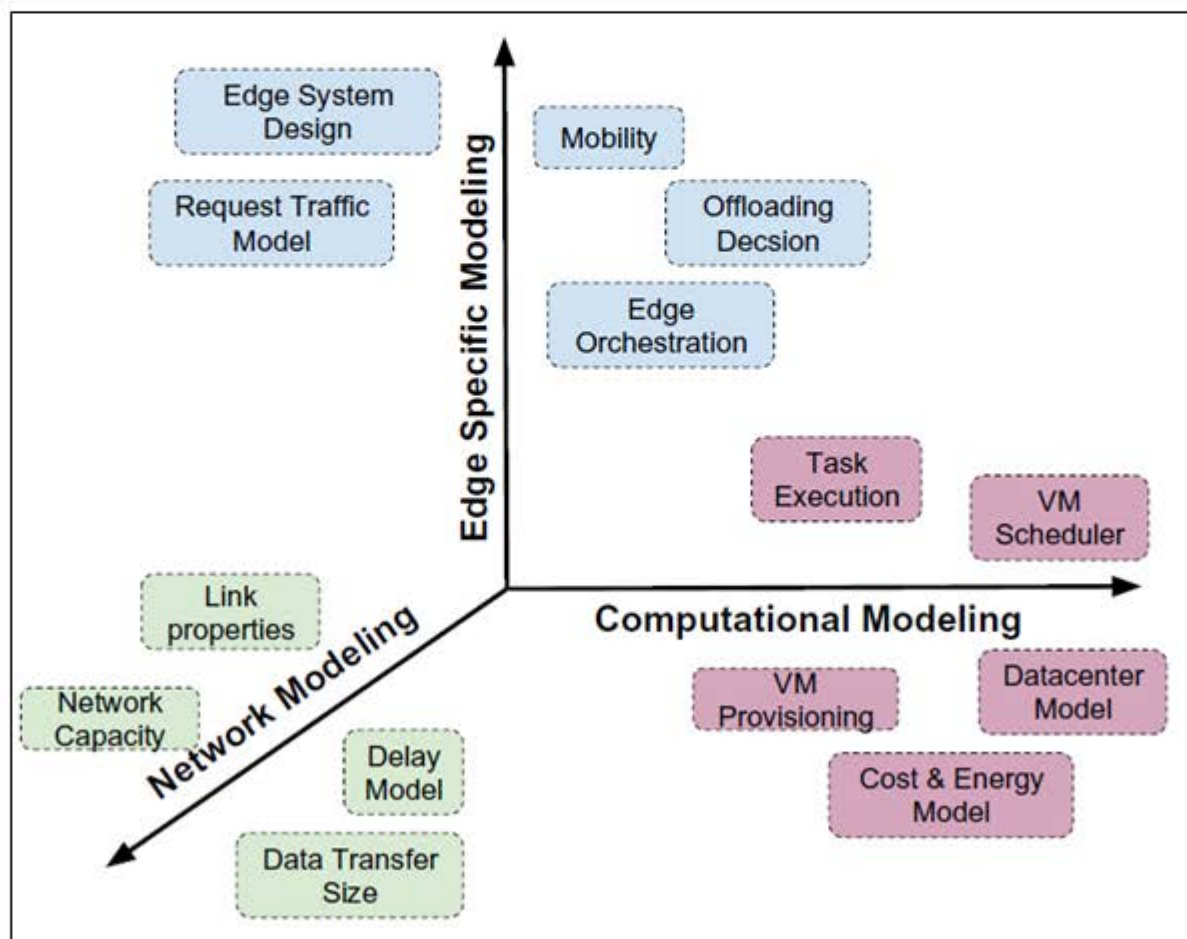
Slacker是一个针对容器快速启动优化的新型Docker存储驱动。



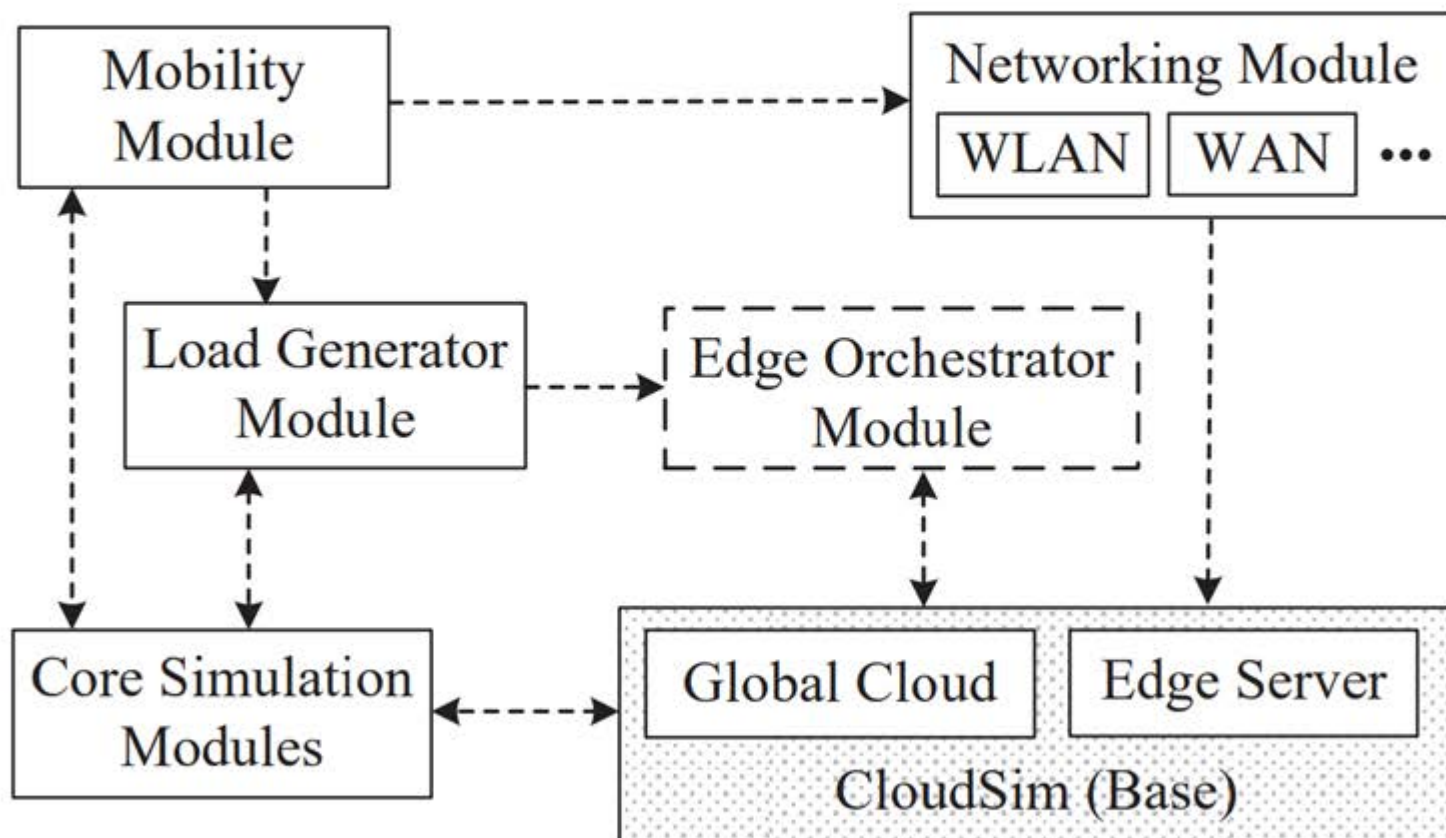
Slacker驱动栈:



EdgeCloudSim是基于CloudSim的一种边缘计算仿真工具。

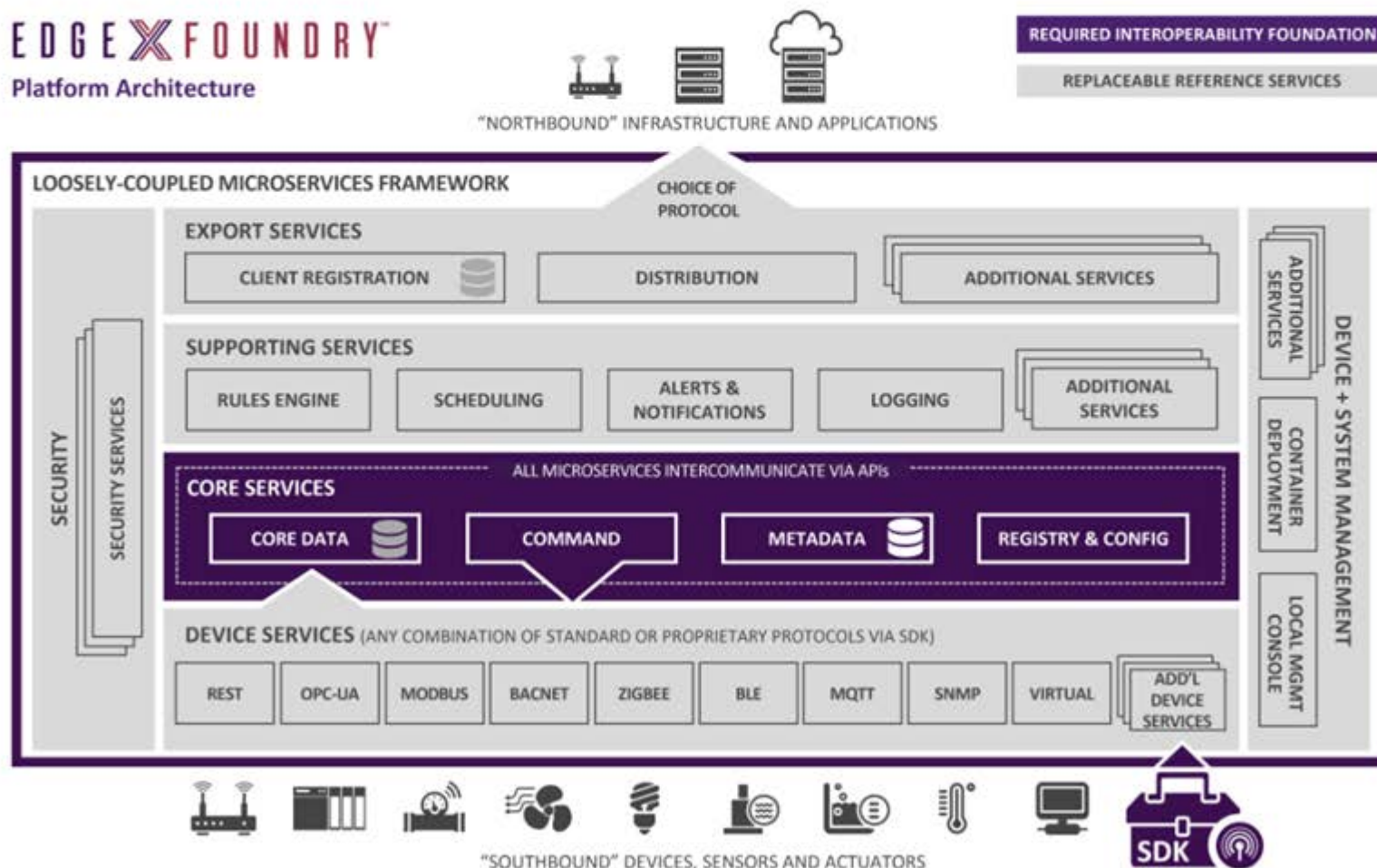


EdgeCloudSim模块之间的关系:

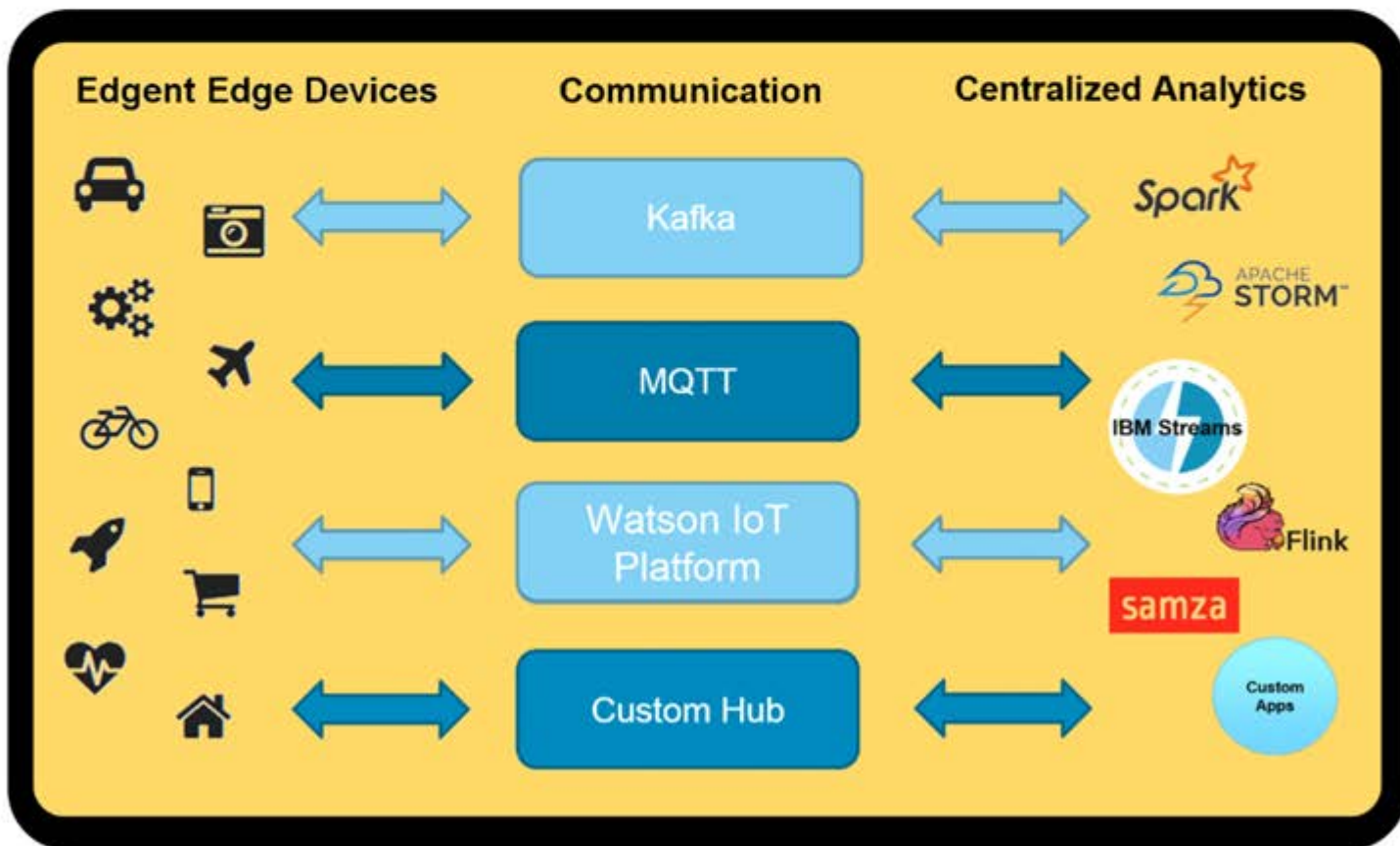


EdgeX Foundry是一款位于网络边缘，独立于供应商的开源软件平台。

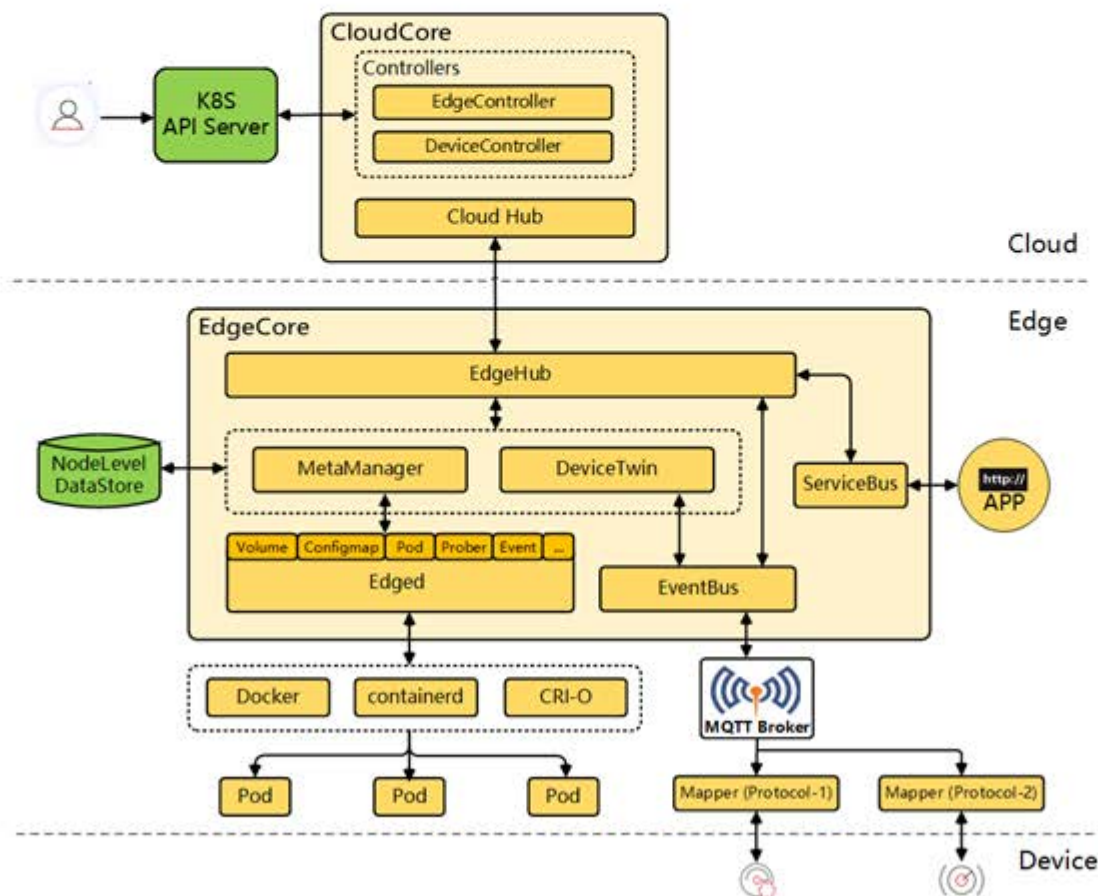
EDGE X FOUNDRY™
Platform Architecture



Apache Edgent是一种针对边缘设备的开源编程模型和运行环境，可以被部署到边缘设备上从而实现对设备上的数据和事件进行分析。



KubeEdge是一个开源系统，用于将容器化应用程序编排功能扩展到边缘的主机。



- ▶ 任务卸载简介及其衡量指标
- ▶ 任务卸载方式
- ▶ 不同场景下任务卸载策略研究
- ▶ 不同架构下任务卸载策略研究
- ▶ 开源工具简介
- ▶ 边缘卸载策略展望及其挑战

► 资源协同调度的挑战与展望：

- 在边缘计算中，虽然在基站附近部署了具有一定处理能力的服务器，但是面对服务复杂性的提高，数量的增多，动态性的提高，面临着越来越多的挑战。
- 在边缘计算中，虽然在基站附近部署了具有一定处理能力的服务器，但是面对服务复杂性的提高，数量的增多，动态性的提高，面临着越来越多的挑战。

► 用户移动性新研究机会：

- 基于用户移动轨迹的任务卸载预处理：利用用户轨迹的统计信息，预测出用户离开现有服务器覆盖范围的时间和即将进入的服务器覆盖区域，以供任务的无缝切换。
- 在雾计算中的用户移动性管理：用户节点卸载任务至边缘服务节点时，应该根据任务完成时间的估计，以及完成时间点用户-服务节点的相对位置做出合理的卸载策略。

- ▶ 简述任务部分卸载和0-1卸载的优劣，并说明部分卸载问题中需要解决的关键问题。
- ▶ 简述边缘计算与雾计算的异同以及两者需要解决的关键问题。
- ▶ 在多接入网络框架中，任务卸载应该考虑哪些因素。
- ▶ 在异构边缘网络结构中，如何充分利用边缘服务器资源。
- ▶ 试分析图 4-6子任务卸载策略中，当两个并行任务的计算量为多少时，部分卸载不再是最优选择？