ULOOF: a User Level Online Offloading Framework for Mobile Edge Computing

Jose´ L.D. Neto, Se-young Yu, Daniel F. Macedo, Jose´ M.S. Nogueira, Rami Langar, Stefano Secci

1.目标：设计一个移动计算卸载框架可以最大限度地减少远程执行开销

2.框架：用户级别的在线卸载框架（uloof）：

配备了用于估计应用方法的执行时间和能耗的新算法，以及位置感知无线网络容量估计器。

2.1 框架特点：既不需要超级用户对移动设备的特权，也不需要修改底层操作系统。

2.2 框架粒度：应用程序方法级别

2.3 考虑因素：能量和执行时间

2.4 可卸载方法选择：可卸载方法用显式注释（“@offloadcandidate”）标记。然后用注释编译应用程序，后编译器创建一个修改后的APK，将卸载逻辑集成到应用程序中。如果应用程序方法具有注释标签，则认为它是可卸载的

2.5 流程：在移动设备中，仪器中的仪器组件提供了卸载的候选方法。每当调用此类方法时，它都会截获调用，并对本地和远程方法执行情况进行执行时间和能耗估计。然后，决策引擎根据估计选择是本地执行还是远程执行。远程执行平台通过连接器模块负责可卸载计算的远程执行。它执行请求的卸载并将结果返回到移动设备。

3.算法：

3.1 决策引擎：提供准确的执行时间和能耗估计来支持卸载决策，同时对代码工具要求最少的用户工作量。

uloof决策引擎使用成本函数来估计运行这些方法所需的能量和时间

$$L(m) = \alpha \cdot t_l(m) + (1-\alpha) \cdot e_l(m)$$
$$R(m) = \alpha \cdot t_r(m) + (1-\alpha) \cdot e_r(m)$$

M是移动应用程序的一组方法，m ∈ M是卸载候选方法。L(m) 和R(m) 分别是本地和远程执行情况下的估计成本函数。

当远程执行成本低于本地执行成本触发方法卸载，即R(m) < L(m)

3.2 预测执行时间：分析计算卸载的方法执行时间的方法：根据以前的执行结果预测执行时间，并在每次调用时动态调整预测

3.2.1. 预测执行时间步骤：

（1）设置没有进行预测的初始执行，并将卸载决策作为一个随机选择

（2）第一次训练

（3）评估函数-参数：执行时间 输出-评估值

（4）本地执行时间$t_l(m)$ 定义：

$$t_l(m) = \phi_l(assess(args_m))$$

argsm是方法m中的一组参数。

（5）跟踪远程执行时间

远程执行时间$t_r(m)$预测：

$$t_r(m) = \phi_r(assess(args_m)) + d_r(m)$$

size（m，r）返回发送远程执行参数和检索结果所需的数据量。

$$d_r(m) = size(m,r)/b$$

3.3 能耗预测：

uloof能源模型的目的是基于在设备中测量的经验数据建立一个能源消耗曲线

本地执行的能耗：

$$e_l(m) = c_{cpu}(k_m, m) + c_{radio,l}(m)$$ Ccpu :是执行时间组件

$$c_{cpu}(m) = l_{cpu}(k_m/runtime_m) \cdot runtime_m$$ $runtime_m$ 是方法m的本地执行时间，lcpu是基于刻度频率的估计能耗函数

远程执行的能耗： $$e_r(m) = c_{radio,r}(m)$$

Cradio,l 和Cradio,r 分别为在本地和远程执行时给定方法m的消耗量

$$c_{radio,l}(m) = l_{radio}(\tau) \cdot d_l(m) \qquad c_{radio,r}(m) = l_{radio}(\tau) \cdot d_r(m)$$

lradio 是能量消耗。为了估计总消耗量，lradio将乘以传输数据所花费的估计时间。dl（m）和dr（m）是在本地和远程执行中传输数据所花费的时间。当方法在本地执行时： dl(m) = size(m, l)/b。 其中 size(m,l) 是在本地执行方法m时的网络流量。

因此 lcpu 和 lradio 是特定于设备和方法的功能

3.4 网络带宽估计

目的：改进执行时间和能耗预测。

传输延迟取决于调用可卸载方法时设备可用的容量，因此测量准确率网络容量是决策引擎的重要组成部分

$$\tau_{t+1} = \tau_t \cdot (1 - \beta) + \tau \cdot \beta,\ 0 \le \beta \le 1$$

τt 是在时间t估计的网络容量，$\tau t+1$是下次估计的网络容量，τ是经验计算值。β用于平滑t和t+1之间容量变化的噪声。经验容量τ的计算因Wi-Fi和蜂窝网络而异。

在Wi-Fi网络上，网络容量计算为：

$$\tau = \frac{d}{\triangle t}$$ d是△t网络传输数据的大小
△t是发送/接收数据的时间。

因为蜂窝塔的服务范围比Wi-Fi接入点大得多，所以蜂窝网络的容量

$$S = \frac{\tau}{log_2(1 + SNR)}$$ SNR是信噪比

当用户移动到同一个塔覆盖范围内的不同位置时，我们使用先前存储的S和当前位置的信噪比计算新容量 τ' 。

$$\tau' = S \cdot log_2(1 + SNR')$$

4. 实验：分为WIFI网络和蜂窝移动网络

4.1 执行时间和能耗

4.2 预测精度

4.3 系统开销
4.4 卸载决策权衡评估
4.5 不同装置之间的比较