

引用格式:裘华东,张燕,涂莹. 移动云计算嵌套式两阶段博弈计算卸载算法[J]. 科学技术与工程, 2018, 18(32): 58—63
Qiu Huadong, Zhang Yan, Tu Ying. Nested two-stage game computation offloading algorithm in mobile cloud computing[J]. Science Technology and Engineering, 2018, 18(32): 58—63

移动云计算嵌套式两阶段博弈计算卸载算法

裘华东 张燕 涂莹
(国网浙江省电力有限公司,杭州 310012)

摘要 移动云计算中,移动设备需要决定哪些应用部分卸载至云端处理,即计算卸载决策问题。针对这一问题,提出了一种嵌套式两阶段博弈算法。第一阶段中,移动设备决策其服务请求至云端处理的部分;第二阶段中,云端系统根据所有移动设备的服务请求到达率决策服务请求处理的资源分配。移动设备的目标是最小化功耗和服务请求响应时间,而云端系统的目标是最大化收益。基于向后归纳原则,利用凸优化方法求解了嵌套式两阶段博弈过程中移动设备和云端系统的最优策略,并证明算法可以产生唯一 Nash 均衡解。实验结果表明,比较基准算法,嵌套式两阶段博弈算法可以使移动设备同步降低平均功耗和平均服务请求响应时间分别约 21.8% 和 31.9%。
关键词 移动云计算 移动设备 博弈论 嵌套式博弈 资源分配
中图分类号 TP393; **文献标志码** A

移动云计算(mobile cloud computing, MCC)可以将计算、内存和存储需求从资源受限的移动设备转移至资源不受限的云端^[1],为移动设备提供了诸多优势,包括扩展了移动用户的存储能力,通过备份用户数据降低了移动设备上数据和应用的受损风险^[2]。尤其是,可以延长移动设备的电池续航时间。MCC使得移动设备可以运行计算密集型应用任务,该类应用在移动设备上局部处理时需要消耗大量电池电量^[3],而将其发送至远程云端处理可以降低能耗,该技术即为计算卸载(computation offloading)问题^[4]。

通过计算卸载可以降低移动设备功耗,但会导致服务请求需重新发送至云端,从而增加服务请求响应时间,因此,为了功耗与服务响应时间的均衡,移动设备需要合理选择是否进行应用的计算卸载,以及具体哪些应用部分卸载至云端处理。相关研究中,文献[5]将具有相互依赖关系的工作流式应用任务调度优化问题形式化为最大流/最小分割问题,通过优化在移动设备与云端中执行的每个子任务的形式进行能耗优化。文献[6]优化了具有周期性特征的任务调度问题,为了优化调度能耗,算法以任务周期足够大为假设,通过降低任务调度间隔时间达到优化能耗的目标。文献[7]利用遗传算法来搜索云端和移动终端计算资源联合执行应用的全局最优

解,重点研究了任务迁移的选择问题。文献[8]对能耗和时间进行了联合优化,通过集中式调算策略制定可以实现最优解。响应时间优化与功耗优化是相互冲突的两个目标,为了做到同步优化,本文设计了一种嵌套式两阶段博弈算法,第一阶段针对移动设备端优化,第二阶段针对云端优化。然后利用向后归纳原则,利用凸优化方法求解了嵌套式两阶段博弈过程中移动设备和云端系统的最优策略,实现了功耗与响应时间间的均衡优化。

1 移动云计算系统模型

考虑一个移动云计算 MCC 系统(即由移动设备与云计算组成的交互系统)包含 N 个移动设备,如智能手机或笔记本电脑,且通过 Wi-Fi 或 4G 网络与云端连接。MCC 系统中每个移动设备有唯一的 ID 标识,表示为 i 。如图 1 描述的是第 $i(1 \leq i \leq N)$ 个移动设备。每个移动设备 i 可执行一个应用,产生服务请求,该服务请求可在局部处理,也可通过计算卸载至远程云端执行。

为了寻找平均响应时间的分析模型,假设第 $i(1 \leq i \leq N)$ 个移动设备产生的服务请求服从平均产生速率为 λ_i 的 Poisson 分布,这样可以基于应用行为进行服务请求预测。移动设备选择卸载服务请求至云端执行的概率为 p_i^M ,上标“M”表示移动“mobile”,即可称 p_i^M 为移动设备 i 的卸载概率。移动设备的卸载概率值即为 MCC 优化框架的优化变量。根据 Poisson 分布的性质,移动设备 i 产生的服务请求卸载

2018 年 3 月 6 日收到
国家自然科学基金(57687845)资助
第一作者简介:裘华东(1962—),男,浙江嵊州人,讲师。研究方向:电力大数据及云计算系统。

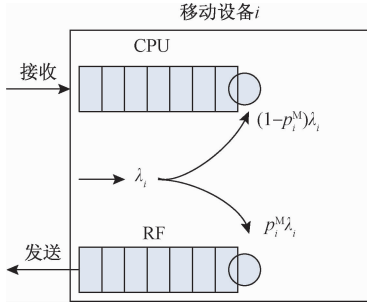


图1 移动设备:局部执行或云端执行

Fig. 1 Mobile device: Local execution or cloud execution

至云端处理服从平均速率为 $p_i^M \lambda_i$ 的 Poisson 过程,因此,移动设备 i 产生的服务请求在局部设备上处理则服从平均速率为 $(1 - p_i^M) \lambda_i$ 的 Poisson 过程。当 p_i^M 变大时,局部处理服务请求的响应时间将降低,而远程云端处理服务请求的平均响应时间将增加(由于发送/接收服务请求和云端处理服务请求的平均延时增加)。在移动设备 i 的功耗方面,移动 CPU 的功耗将降低(局部处理服务请求),而用于发送服务请求的无线射频 RF 的功耗将增加。因此,对于每个移动设备而言,需要在考虑服务请求(即计算和数据通信请求)、预期卸载速率 $p_i^M \lambda_i$ 和服务器拥塞特征的情况下,合理选择最优策略 p_i^M 。

令 μ_i^M 为移动设备 i 的平均服务请求处理速率,则移动设备 i 上局部处理服务请求的平均响应时间可计算为

$$R_i^M(p_i^M) = \frac{1}{\mu_i^M - (1 - p_i^M) \lambda_i} \quad (1)$$

令 μ_i^S 为移动设备 i 发送服务请求的平均速率,上标“S”表示发送“sending”。那么,可以计算服务请求完全发出之前服务请求在移动设备上的平均等待时间为

$$R_i^S(p_i^M) = \frac{1}{\mu_i^S - p_i^M \lambda_i} \quad (2)$$

同时, μ_i^S 正比于移动设备与访问点间的无线信道能力。

移动设备 i 的功耗由两部分组成:①局部处理服务请求时移动 CPU 的功耗;②发送服务请求至云端时 RF 组件的功耗^[9]。CPU 功耗和 RF 组件功耗可进一步划分为 CPU 或 RF 组件活动状况时(即正在处理或发送服务请求)的动态功耗部分和静态功耗部分。令 $P_{CPU,i}^{\text{dyn}}(p_i^M)$ 表示移动设备 i 的 CPU 的平均动态功耗,正比于 CPU 的活动时间,表示为 $(1 - p_i^M) \lambda_i / \mu_i^M$ 。则 $P_{CPU,i}^{\text{dyn}}(p_i^M)$ 可计算为

$$P_{CPU,i}^{\text{dyn}}(p_i^M) = \frac{(1 - p_i^M) \lambda_i}{\mu_i^M} P_{CPU,i}^{\text{dyn,max}} \quad (3)$$

式(3)中, $P_{CPU,i}^{\text{dyn,max}}$ 为移动 CPU 活动状态时的动态功耗。类似地,移动设备 i 的 RF 组件的平均动态功耗为

$$P_{RF,i}^{\text{dyn}}(p_i^M) = \frac{p_i^M \lambda_i}{\mu_i^M} P_{RF,i}^{\text{dyn,max}} \quad (4)$$

令 $P_{CPU,i}^{\text{sta}}$ 和 $P_{RF,i}^{\text{sta}}$ 分别表示移动设备 i 的 CPU 和 RF 组件的静态功耗,均为常量。那么,移动设备 i 的总体功耗为

$$P_{\text{Mobile},i}(p_i^M) = P_{CPU,i}^{\text{dyn}}(p_i^M) + P_{RF,i}^{\text{dyn}}(p_i^M) + P_{CPU,i}^{\text{sta}} + P_{RF,i}^{\text{sta}} \quad (5)$$

图2是由服务请求池、作为服务提供者的数据中心以及中间资源管理节点组成的移动云计算资源分配的目标系统,现考虑的是一个同质数据中心环境。数据中心由 M 个负责移动设备服务请求处理的同质服务器组成, j 为数据中心服务器的标识。

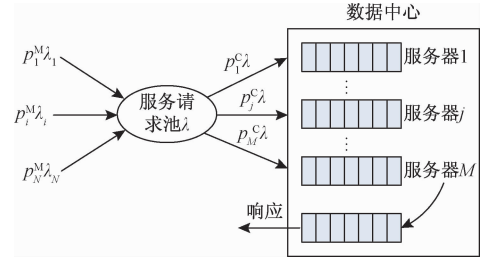


图2 移动云计算资源分配系统

Fig. 2 Resource allocation system of mobile cloud computing

服务请求池包括来自所有移动设备产生的远程服务请求。根据 Poisson 分布的属性,服务请求池的总体服务请求产生速率 λ 可计算为

$$\lambda = \sum_{i=1}^N p_i^M \lambda_i \quad (6)$$

一个服务请求可分配至数据中心的任一服务器上。令 p_j^C 为服务请求分配至服务器 j 的概率,上标“C”表示云端“cloud”。这些概率值即为云端资源分配优化框架的优化变量。根据 Poisson 分布的属性,分配至服务器 j 的移动服务请求服从平均速率为 $p_j^C \lambda$ 的 Poisson 过程,即是服务器的平均服务请求到达率。一旦服务请求被分配至服务器,服务器即可为服务创建虚拟机,开始应用执行过程。

MCC 系统中的每个服务器 j 为执行移动服务请求分配的总资源比例表示为 $\varphi_j^C (0 \leq \varphi_j^C \leq 1)$ 。利用 M/M/1 排队模型,分配至该服务器的服务请求的平均处理时间为

$$R_j^C(p_j^C; \varphi_j^C; p^M) = \frac{1}{\varphi_j^C \mu_j^C - p_j^C \lambda} \quad (7)$$

式(7)中, μ_j^C 表示当服务器的所有资源分配至该服务请求处理时的平均处理速率。由于服务器是同质的,故所有服务器的 μ_j^C 值是相等的。 λ 是 $p^M = \{p_1^M, p_2^M, \dots, p_N^M\}$ 的函数,如式(6)。

数据中心完成服务请求处理后发回对服务请求的响应。可以计算出数据中心完全发出前响应平均等待的时间为

$$R^R(p^M) = \frac{1}{\mu^R - \lambda} \quad (8)$$

式(8)中,上标“R”表示接收“receiving”,即移动设备接收数据中心的响应。

因此,移动设备 i 的一个服务请求的平均响应时间(局部处理或云端处理)为

$$R_i^{\text{Avg}}(p^M; p^C; \varphi^C) = (1 - p_i^M) R_i^M(p_i^M) + p_i^M \left[R_i^S(p_i^M) + \sum_{j=1}^M p_j^C R_j^C(p_j^C; \varphi_j^C; p^M) + R^R(p^M) \right] \quad (9)$$

式(9)中, $p^C = \{p_1^C, p_2^C, \dots, p_M^C\}$, $\varphi^C = \{\varphi_1^C, \varphi_2^C, \dots, \varphi_M^C\}$ 。

服务器功耗包括服务器活动状态(即处理服务请求)的动态功耗和静态功耗。服务器 j 的平均动态功耗正比于服务器的活动时间,表示为 $p_j^C \lambda / \varphi_j^C \mu_j^C$,和分配至服务请求处理的资源比例 φ_j^C :

$$P_{\text{Serv},j}^{\text{dyn}}(p_j^C; p^M) = \frac{p_j^C \lambda}{\varphi_j^C \mu_j^C} P_{\text{Serv},j}^{\text{dyn},\max} = \frac{p_j^C \lambda}{\mu_j^C} P_{\text{Serv},j}^{\text{dyn},\max} \quad (10)$$

式(10)中, $P_{\text{Serv},j}^{\text{dyn},\max}$ 为服务器活动状态且所有资源分配至该服务请求处理时的动态功耗。此外,服务器 j 的静态功耗为

$$P_{\text{Serv},j}^{\text{sta}}(\varphi_j^C) = \varepsilon_{\text{Serv},j} + \varphi_j^C (P_{\text{Serv},j}^{\text{sta},\max} - \varepsilon_{\text{Serv},j}) \quad (11)$$

因此,数据中心的总体功耗为所有服务器的功耗之和,即:

$$P_{\text{DC}} = \sum_{j=1}^M [P_{\text{Serv},j}^{\text{dyn}}(p_j^C; p^M) + P_{\text{Serv},j}^{\text{sta}}(\varphi_j^C)] \quad (12)$$

令 $U(R) = \beta - \gamma R$ 表示平均服务请求响应时间为 R 时 MCC 系统的效用函数。则云端的总体收益为

$$\lambda \left[\beta - \gamma \sum_{j=1}^M p_j^C R_j^C(p_j^C; \varphi_j^C; p^M) + R^R(p^M) \right] - \text{price} \sum_{j=1}^M [P_{\text{Serv},j}^{\text{dyn}}(p_j^C; p^M) + P_{\text{Serv},j}^{\text{sta}}(\varphi_j^C)] \quad (13)$$

式(13)中, price 表示单位能量的价格。

2 博弈问题形式化

设计一种两阶段博弈算法^[10],对 MCC 系统进行形式化建模。第一阶段博弈中,每个移动设备 i 决定远程云端处理的服务请求比例 p_i^M ,移动设备的目标是最小化目标函数:

$$w_1 P_{\text{Mobile},i}^{\text{Avg}}(p_i^M) + w_2 R_i^{\text{Avg}}(p_i^M; p^C; \varphi^C) \quad (14)$$

以上目标函数是移动设备功耗 $P_{\text{Mobile},i}(p_i^M)$ 与平均服务请求响应时间 $R_i^{\text{Avg}}(p_i^M; p^C; \varphi^C)$ 的线性组合。由于 p^C 和 φ^C 由云端决定,所以对于移动设备是未知

的。令 $p_{-i}^M = \{p_1^M, p_2^M, \dots, p_{i-1}^M, p_{i+1}^M, \dots, p_N^M\}$ 表示除移动设备 i 外其他所有移动设备的卸载概率。由于 p_{-i}^M 在移动设备 i 之前也未给出,因此,可以得到 $R_i^{\text{Avg}}(p_i^M; p^C; \varphi^C) = R_i^{\text{Avg}}(p_i^M, p_{-i}^M; p^C; \varphi^C)$ 。权值 w_1 和 w_2 对于单个设备不必设置相同。例如:若移动设备的电量不足,可降低 w_1 值,由于此时电池不是瓶颈;而当电量降低至关键位置时,可以增加 $P_{\text{Mobile},i}(p_i^M)$ 的权重值,以得到更多的计算卸载。

在第二个阶段,云端分配服务请求至服务器,并为处理服务请求分配总资源的一部分,即需要寻找最优的 p^C 和 φ^C ,两个值均取决于所有移动设备的服务请求卸载率。云端的目标是最大化其收益,即式(13),等同于最小化目标函数:

$$\lambda \gamma \sum_{j=1}^M p_j^C R_j^C(p_j^C; \varphi_j^C | p^M) + \text{price} \sum_{j=1}^M [P_{\text{Serv},j}^{\text{dyn}}(p_j^C | p^M) + P_{\text{Serv},j}^{\text{sta}}(\varphi_j^C)] \quad (15)$$

式(15)中, $R_j^C(p_j^C; \varphi_j^C | p^M)$ 和 $P_{\text{Serv},j}^{\text{dyn}}(p_j^C | p^M)$ 分别表示给定 p^M 时 $R_j^C(p_j^C; \varphi_j^C; p^M)$ 和 $P_{\text{Serv},j}^{\text{dyn}}(p_j^C; p^M)$ 的函数。

假设云端资源分配结果 p^C 和 φ^C 对于移动设备是可知的,则第一阶段博弈中,所有移动设备可竞争分配资源,即为标准式博弈(即所有博弈者同步选择策略)。每个移动设备 i 选择最优策略 p_i^M 以最小化代价函数:

$$w_1 P_{\text{Mobile},i}^{\text{Avg}}(p_i^M) + w_2 R_i^{\text{Avg}}(p_i^M | p^C; \varphi^C) = w_1 P_{\text{Mobile},i}^{\text{Avg}}(p_i^M) + w_2 R_i^{\text{Avg}}(p_i^M, p_{-i}^M | p^C; \varphi^C) \quad (16)$$

式(16)中, $R_i^{\text{Avg}}(p_i^M | p^C; \varphi^C)$ 和 $R_i^{\text{Avg}}(p_i^M, p_{-i}^M | p^C; \varphi^C)$ 为给定 p^C 和 φ^C 时 $R_i^{\text{Avg}}(p_i^M; p^C; \varphi^C)$ 的函数。

因此,MCC 系统本质上是一个嵌套式两阶段博弈,由于第一阶段博弈本身为标准式博弈,提出对该嵌套式两阶段博弈的求解算法。

3 嵌套式两阶段博弈优化算法

基于向后归纳原则,对 MCC 系统的嵌套式两阶段博弈进行求解,即先求解第二阶段的云端优化策略,再推导移动设备的优化策略。

3.1 云端优化

云端优化即是在给定 p^M 时寻找最优控制变量 p^C 和 φ^C ,以最小化目标函数式(15)。将该收益最优问题命名为资源分配与请求分派问题 RARD。RARD 问题的约束条件为

$$0 \leq p_j^C \leq 1, \quad \forall j \quad (17)$$

$$0 \leq \varphi_j^C \leq 1, \quad \forall j \quad (18)$$

$$\sum_{j=1}^M p_j^C = 1 \quad (19)$$

$$p_j^C \sum_{i=1}^N p_i^M \lambda_i < \varphi_j^C \mu_j^C, \quad \forall j \quad (20)$$

提出定理 1,以得到请求分派阶段的最优解,即找到 RARD 问题中的最优 p^C 。由于篇幅原因,定理证明略。

定理 1 最优请求分派。在 RARD 问题的最优解中,最优请求分派概率是相等的,即对于 $1 \leq j \leq M, p_j^C = 1/M$ 。

基于定理 1,初始的 RARD 问题转换为最优资源分配问题,即给定 p^C 的情况下,寻找最优 φ^C ,以最小化目标函数式(15)。最优资源分配问题的约束条件为式(18)和式(20)。

该问题是凸性最优化问题,由于当给定 p^C 时目标函数式(15)为凸函数,约束式(18)和式(20)为线性不等式约束。因此,可通过标准凸性最优化方法求解。同时,可得到以下最优资源分配问题最优解的推论。

推论 1 最优资源分配 在资源分配问题最优解中,最优变量矢量 φ^C 均是相等的。

证明 当给定 p^C 时,最优资源分配问题可分裂为在每个服务器上的 M 个局部优化问题集合。每个局部优化问题需要寻找最优 φ_j^C 值,使得目标函数达到最小:

$$\lambda \gamma p_j^C R_j^C(p_j^C; \varphi_j^C | p^M) + price[P_{Serv,j}^{dyn}(p_j^C | p^M)] + P_{Serv,j}^{sta}(\varphi_j^C) \quad (21)$$

$$\text{约束条件为 } 0 \leq \varphi_j^C \leq 1 \text{ 和 } p_j^C \sum_{i=1}^N p_i^M \lambda_i < \varphi_j^C \mu_j^C。$$

不同服务器上局部最优化问题是完全相同的,原因在于:① p_j^C 值是相互相等的;②服务器是同质的。证毕。

基于推论 1 可知,仅需找到一个服务器上的局部最优问题的解即可。

3.2 移动设备优化

假设云端的分配资源比例对于移动设备是可知的,即 p^C 和 φ^C 提前给出。那么,在嵌套式两阶段博弈的第一阶段中,每个移动设备 i 决定远程云端处理的服务请求比例 p_i^M ,以最小化目标函数式(16)。关于 p_i^M 的约束如下:

$$0 \leq p_i^M \leq 1, \quad \forall i \quad (22)$$

$$(1 - p_i^M) \lambda_i \leq \mu_i^M - \varepsilon, \quad \forall i \quad (23)$$

$$p_i^M \lambda_i \leq \mu_i^S - \varepsilon, \quad \forall i \quad (24)$$

$$p_j^C \sum_{i=1}^N p_i^M \lambda_i \leq \varphi_j^C \mu_j^C - \varepsilon, \quad \forall j \quad (25)$$

$$\sum_{i=1}^N p_i^M \lambda_i \leq \mu^R - \varepsilon \quad (26)$$

式中, $\varepsilon \ll 1$ 为小正实数,目的是使得 p^M 的区域为闭

合集合。约束条件式(23)~式(26)分别由式(1)、式(2)、式(7)和式(8)导出。该分布式优化问题本质上是标准化博弈,每个博弈者(即移动设备)同步选择策略 p_i^M 以最小化目标函数式(16)。命名该博弈为对于每个移动设备的卸载概率决策博弈 OPD。

由于移动设备相互之间是非合作关系,即非合作式博弈,则需要求解 Nash 均衡解的存在性和唯一性。Nash 均衡是所有博弈者的最优策略组合,处于该均衡处时,博弈者无法脱离该均衡而找到最优的策略。证明 OPD 博弈问题中的 Nash 均衡的存在性和唯一性如下。

定理 2 OPD 博弈中的 Nash 均衡。OPD 博弈中的 Nash 均衡存在且是唯一的。

证明 OPD 博弈是严格凹性 n 人博弈,原因在于:①对于每个博弈者 i ,最小化凸目标函数式(16)等同于最大化每个博弈的凹支付函数;②约束式(22)~式(26)下的策略集 p^M 的域是闭合凸集。如此,可推导出 OPD 博弈的 Nash 均衡是存在且唯一的。

移动设备通过利用标准凸优化方法可以找到相应的 OPD 博弈的 Nash 均衡,具体过程如算法 1 所示。

算法 1 Find Nash Equilibrium in the OPD Game for Each Mobile Device i

1. Initialize p_i^M (the offloading probability of the i -th mobile device itself) as well as p_{-i}^M (the anticipation of the offloading probabilities of other mobile devices)
2. Do the following procedure iteratively
3. For each $1 \leq i' \leq N$
4. Find the optimal $p_{i'}^M$ (i . e., the best response of the i' -th mobile device) with respect to $p_{-i'}^M$, by solving the convex optimization problem for i' -th mobile device with objective function Eq(16) and constraints Eq(22) ~ Eq(26)
5. Update $p_{i'}^M$ to be the new value
6. End
7. Until the solution converges

然而,在 MCC 系统中,由于 p^C 和 φ^C 值由云端决定,故移动设备并不能无法提前得到。基于顺序博弈的向后归纳原则,每个移动设备基于 p^C 和 φ^C 的预期优化其卸载概率 p_i^M 。设计了一种迭代算法 2,可以找到嵌套式两阶段博弈中每个移动设备的最优策略。在每次迭代中,算法 2 拥有一次 Nash 均衡寻找阶段和一次资源分配预期更新阶段。前一阶段中,移动设备运行算法 1 寻找对应 OPD 博弈的 Nash 均衡。后一阶段中,移动设备基于更新的 Nash 均衡对 p^C 和 φ^C 的预期值进行更新。算法 2 具体过程如下所示。

算法 2 Deriving an Optimal Strategy for Each Mobile Device in the Nested Two Stage Game

1. Initialize the anticipation of p^C and φ^C
2. Do the following procedure iteratively

3. Finding the Nash equilibrium: Find the Nash equilibrium of the OPD game based on the anticipation of p^C and φ^C , by executing Algorithm 1
4. Updating the anticipation of resource allocation results: Find and update the anticipation of p^C and φ^C , by solving the RARD problem based on p^M obtained from the Nash equilibrium
5. Until the solution converges

4 仿真实验

通过仿真实验配置多移动设备与云计算系统组合而成的移动云计算系统,并验证嵌套式两阶段博弈算法的性能。

考虑 MCC 系统由 $N = 20$ 个移动设备和云计算系统组成,云计算系统的数据中心由 10 台服务器组成。MCC 系统中的参数值以标准值取代实际值。每个移动设备的服务请求产生速率 λ_i 服从 $[1, 1.5]$ 间的均匀分布,移动 CPU 的平均服务请求处理速率 μ_i^M 设置为 1.6,每个移动设备的平均服务请求发送速率 μ_i^S 设置为 2。每个移动 CPU 和 RF 组成的最大动态功耗值 $P_{CPU,i}^{dyn,max}$ 和 $P_{RF,i}^{dyn,max}$ 分别服从 $[4, 6]$ 和 $[1, 1.5]$ 间的均匀分布,每个移动 CPU 和 RF 组成的静态功耗值 $P_{CPU,i}^{sta}$ 和 $P_{RF,i}^{sta}$ 分别服从 $[2, 3]$ 和 $[1, 1.5]$ 间的均匀分布。对于云端计算系统,每个同质服务器的最大平均服务请求处理速率 μ_j^C 设置为 3,服务器的最大动态功耗 $P_{Serv,j}^{dyn,max}$ 和最大静态功耗 $P_{Serv,j}^{dyn,max}$ 分别设置为 10 和 5,云端的平均响应发送速率 μ^R 设置为 50。云端系统中的效用函数中,参数 $\beta = 5, \gamma = 1$,单位能量价格 $price = 0.2$ 。

实验 1 测试 MCC 系统,比较平均功耗和所有移动设备的平均服务请求响应时间,选择两种基准算法。在 Baseline1 算法中,所有移动设备产生的服务请求均在移动设备上局部处理。在 Baseline2 算法中,移动设备将所有产生的服务请求均发送至云计算系统处理,且云系统基于总体服务请求到达率以最优的请求分派和资源分配进行服务请求处理。图 3 为标准化的平均功耗与平均服务请求响应时间的结果。

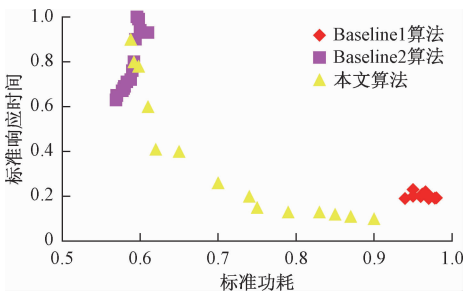


图 3 功耗与响应时间指标

Fig. 3 Index of energy consumption and response time

从图 3 可以看出,比较 Baseline1 算法,本文的博弈算法可以使移动设备可以同步降低平均功耗和平均服务请求响应时间分别约 21.8% 和 31.9%,原因在于:如果所有服务请求均在局部处理,则由于请求处理的严重拥塞,会导致移动设备在 CPU 上拥有更高功耗和更高服务请求响应时间。另一方面,比较 Baseline2 算法,博弈算法可以使得移动设备降低平均服务请求响应时间约 89%。然而,比较 Baseline2 算法,却无法使移动设备降低平均功耗。这主要是因为所有服务请求均卸载至云端处理对于移动设备而言是能效最高的策略,但会导致严重延时。

此外,实验还比较了所有移动设备的目标函数式(14)的平均值,该目标函数展示了移动设备对于功耗与服务请求响应时间间的均衡期望。表 1 是博弈算法相比 Baseline1 和 Baseline2 算法在目标函数式(14)的平均值上所得到的最小和最大降低幅度。可以看出,最大降低幅度值达到 89.5%,这表明嵌套式两阶段博弈算法是切实有效的。

表 1 博弈算法下目标函数的降低幅度

Table 1 Reduction of the objective function in the game algorithm

比较 Baseline1 降低幅度/%		比较 Baseline2 的降低幅度/%	
最小	最大	最小	最大
15.4	41.2	0	89.5

实验 2 中,设置权重因子 w_1 和 w_2 分别为 5 和 1,改变 MCC 系统中移动设备数量 N 值进行性能分析,主要测试了移动设备数量发生改变时博弈算法得到的标准化平均功耗和标准化平均服务请求响应时间的变化。图 4 的结果表明,平均功耗和平均响应时间均会随着 N 值的增加而增加,这是因为 N 值增加带来拥塞等级的增加,进而导致数据中心中平均服务请求响应时间的增加。移动设备感知到拥塞后,会分配更多的服务请求进行局部处理,这反过来会导致功耗的增加。

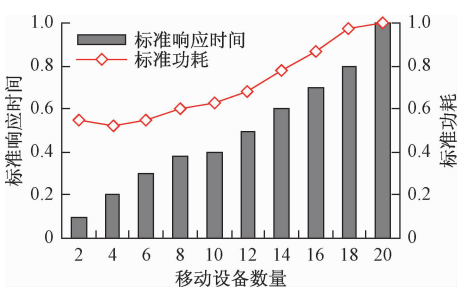


图 4 移动设备数量对性能的影响

Fig. 4 Effects of number of mobile devices on performance

5 结论

针对移动云计算中的计算卸载问题,提出了一种嵌套式两阶段博弈算法。第一阶段移动设备对应用卸载至云端做出决策,第二阶段云端对服务请求的资源分配做出决策。设计了两个阶段中的两个主体的优化目标函数,并利用向后归纳原则和凸优化方法,求解了嵌套式两阶段博弈过程中移动设备和云端系统的最优策略。结果表明,所提博弈算法可以同步降低移动设备的平均功耗和平均服务请求响应时间。

参 考 文 献

1 Zare J, Abolfazli S, Shojafar M, *et al.* Resource scheduling in mobile cloud computing: Taxonomy and open challenges. IEEE International Conference on Data Science and Data Intensive Systems. New York: IEEE Computer Society, 2015:594—603

2 Zhou B, Dastjerdi A V, Calheiros R N, *et al.* A context sensitive off-loading scheme for mobile cloud computing service. IEEE International Conference on Cloud Computing. New York: IEEE Computer Society, 2015:869—876

3 李继芯,李小勇,高云全,等. 5G 网络下移动云计算节能措施研究. 计算机学报,2017;40(7):1491—1516

Li Jixin, Li Xiaoyong, Gao Yunquan, *et al.* Energy saving research on mobile cloud computing in 5G. Chinese Journal of Computers, 2017; 40(7):1491—1516

4 Barbera M V, Kosta S, Mei A, *et al.* To offload or not to offload? The bandwidth and energy costs of mobile cloud computing. Proceedings—IEEE INFOCOM, 2015; 12(11):1285—1293

5 Terzopoulos G, Karatzas H D. Dynamic voltage scaling scheduling on Power-aware clusters under power constraints. IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications. New York: IEEE, 2013:72—78

6 Saravanan S, Venkatachalam V. Advance map reduce task scheduling algorithm using mobile cloud multimedia services architecture. Sixth International Conference on Advanced Computing. New York: IEEE, 2015:21—25

7 柳 兴,李建彬,杨 震,等. 移动云计算中的一种任务联合执行策略. 计算机学报,2017;40(2):364—377

Liu Xing, Li Jianbin, Yang Zhen, *et al.* A task collaborative execution policy in mobile cloud computing. Chinese Journal of Computers, 2017;40(2):364—377

8 胡海洋,刘润华,胡 华. 移动云计算环境下任务调度的多目标优化方法. 计算机研究与发展,2017;54(9):1909—1919

Hu Haiyang, Liu Runhua, Hu Hua. Multi-objective optimization for task scheduling in mobile cloud computing. Journal of Computer Research and Development, 2017; 54(9):1909—1919

9 Shin D, Kim K, Chang N, *et al.* Online estimation of the remaining energy capacity in mobile systems considering system-wide power consumption and battery characteristics. Design Automation Conference. New York: IEEE, 2013:59—64

10 Canterbury E R. Game theory: An introduction. Reference Reviews, 2015;14(7):35—36

Nested Two-stage Game Computation Offloading Algorithm in Mobile Cloud Computing

QIU Hua-dong, ZHANG Yan, TU Ying

(State Grid Zhejiang Electric Power Limited Company, Hangzhou 310012, China)

[Abstract] In mobile cloud computing, a mobile device should judiciously decide whether to offload computation and which portion of application should be offloaded to the cloud. For solving this problem, a nested two stage game algorithm is proposed. In the first stage, each mobile device determines the portion of its service requests for remote processing in the cloud. In the second stage, the cloud facilities allocate a portion of its total resources for service request processing depending on the request arrival rate from all the mobile devices. The objective of each mobile device is to minimize its power consumption and the service request response time, the objective of the cloud is to maximize its own profit. Based on the backward induction principle, the optimal strategy for all the mobile devices and the cloud were derived, and it was proved that the algorithm can generate an unique Nash equilibrium solution. Experimental results show, compared with the baseline algorithms, the algorithm can make mobile devices achieve simultaneous reduction in average power consumption and average service request response time by 21.8% and 31.9%, respectively.

[Key words] mobile cloud computing mobile device game theory nested game resource allocation