

## 小蜂窝云中功率与负载的联合优化分配算法

牛瑞彪,唐 伦,陈 婉

(重庆邮电大学 移动通信技术重点实验室,重庆 400065)

**摘 要:** 移动终端资源有限及本地服务基站资源不足会引起移动终端体验质量降低、卸载任务时延的问题。为此,提出一种新的联合优化分配算法。基于小蜂窝信道质量和剩余可用计算资源建立小蜂窝云(SCC),按照信道质量和剩余可用计算资源分配负载(卸载任务)到 SCC,并采用启发式算法求解发送功率的次优解。仿真结果表明,该算法在小蜂窝云计算场景中能提高无线与计算资源的利用率,同时提升用户的体验质量。

**关键词:** 移动云计算;任务卸载;小蜂窝云;剩余可用计算资源;信道质量

**中文引用格式:**牛瑞彪,唐 伦,陈 婉. 小蜂窝云中功率与负载的联合优化分配算法[J]. 计算机工程,2017,43(8): 49-55.

**英文引用格式:**Niu Ruibiao, Tang Lun, Chen Wan. Allocation Algorithm of Power and Load Jointing Optimization for Small Cell Cloud[J]. Computer Engineering, 2017, 43(8): 49-55.

## Allocation Algorithm of Power and Load Jointing Optimization for Small Cell Cloud

NIU Ruibiao, TANG Lun, CHEN Wan

(Key Lab of Mobile Communication Technology, Chongqing University of Post and Telecommunications, Chongqing 400065, China)

**[Abstract]** The Quality of Experience (QoE) of mobile terminals severely reduces caused by both local service station to a serious shortage of resources, and mobile terminal resources with limited. Furthermore, the task offloading into remote cloud brings the large time delay. This paper proposes a joint optimization of the power and load allocation algorithm for small cell cloud. Based on the channel quality and the rest of the available computing resources to build Small Cell Cloud (SCC), according to the same element, it distributes the load (offloading tasks) to SCC and uses a heuristic algorithm to seek approximate suboptimal solution to transmitting power. Simulation results show that the algorithm can improve the utilization of the radio and computing resources, and enhance the user QoE at the same time.

**[Key words]** Mobile Cloud Computing (MCC); task offloading; Small Cell Cloud (SCC); remaining available computing resources; channel quality

**DOI:** 10.3969/j.issn.1000-3428.2017.08.009

### 0 概述

在未来的 5G 网络中,业务种类和负载数量的急剧增长对移动终端和网络都是一种挑战,移动云计算(Mobile Cloud Computing, MCC)是解决移动终端资源有限的一种技术, MCC 的基本实现形式就是任务卸载,而超密集部署和新的网络技术,如移动边缘计算、雾计算是应对网络方面问题一致认同的方法。在超密集部署的网络场景下,通过结合新的网络技术和资源管理技术来提高用户的体验质量(Quality of Experience, QoE)与网络性能。任务卸载<sup>[1-2]</sup>已经成为提升移动终端性能的有效途径,通过这种途径,

使用户拥有了更多的计算能力和存储能力。同时,任务卸载还会让移动终端减少复杂的、耗能的计算任务<sup>[3-5]</sup>。任务卸载的机制是终端将所需要执行的任务通过信道上传到为其提供服务的云进行处理,处理完成后,将处理结果回传。从任务卸载的机制可以发现,任务卸载终端与云端的交互过程实质上是资源置换的过程。终端通过相应的网络通信资源,来置换云端的计算和存储资源。因此,任务卸载的资源优化可分为通信资源优化和计算资源优化两大部分。

当前已有的相关研究在这两方面均有涉及,但是大部分只简单考虑了任务卸载到中心云,而不是

**基金项目:**国家自然科学基金(6157073)。

**作者简介:**牛瑞彪(1990—),男,硕士研究生,主研方向为云计算、5G 网络;唐 伦,教授、博士;陈 婉,硕士研究生。

**收稿日期:**2017-02-15 **修回日期:**2017-03-23 **E-mail:**1406665676@qq.com

其他形式的云,例如小蜂窝云(Small Cell Cloud, SCC)。文献[6]提出一个运算卸载的数学模型,目的是在严格的时延和能耗限制下,联合优化通信和计算资源。文献[7]提出一种平衡能耗和时延的无线资源及计算资源的联合优化算法,以上2种方案也都以中心云作为研究的参考。没有考虑当任务卸载发生在不符合将任务上传到中心云的条件这个问题。针对距离中心云较远的小蜂窝密集部署的场景中任务卸载到中心云计算时会带来严重干扰这一问题,现有的相关研究大部分都是采取一种在本地部署<sup>[8-10]</sup>或者基于已有的基础设备,根据一定算法自动组织起一种本地云<sup>[11-12]</sup>来实现任务卸载。文献[11]提出了一种选择无线可用资源、基站部署场景、时延要求等不同策略时,Femto-cloud大小对性能的影响。文献[12]提出了一种分布式和集中式混合的新的小蜂窝云集群算法。随着研究中越来越多涉及到SCC这种新形式的云,而组成小蜂窝云的各个集群基站剩余计算资源和其他资源并不总是相同,具有差异性,任务卸载不再是简单地卸载到一个中心云,需要考虑任务卸载在小蜂窝云中各个集群基站之间如何分配的问题<sup>[13-15]</sup>。文献[13]提出一种在一定QoS要求下,小蜂窝云处理多用户任务请求卸载的功率优化方案,但该方案没有考虑对用户QoS的进一步优化。文献[14]提出一种基于动态图划分理论进行卸载请求的负载分配方案,但该方案只对卸载请求的分配进行了研究,没有考虑系统的功率损耗情况。

综合以上研究,本文提出一种联合优化系统功耗和用户体验质量(QoE)的资源分配方案。该方案首先根据集群算法建立SCC时各集群基站的相关信息来决定任务卸载运算量的分配,然后通过功率调整,达到优化系统功耗、提高用户体验质量的目的。

## 1 系统模型

在本文考虑的场景中,任何一个SCeNBce(Small Cell enhanced Node B cloud enabled)都具有一定的计算能力和存储能力。所有的SCeNBce通过无线链路连接在一起,每个SCeNBce的带宽为 $B$ ,本文将有任务卸载请求的SCeNBce称为服务基站,这里用集合 $S_{SC} = \{1, 2, \dots, S\}$ 表示,SCeNBce $s$ 服务的众多用户发起的卸载请求都汇集到服务基站SCeNBce $s$ 等待处理,服务基站 $s$ 由于资源不足,不能够为这些业务提供服务,所以通过组建SCC,将每个时隙到来的众多请求,打包分发到周围的SCeNBce进行计算,用 $(W, \Delta_i)$ 表示汇总后的卸载请求,其中, $W$ 表示的是该卸载请求需要处理的数据量; $\Delta_i$ 表示该请求所能容忍的最大时延。这里需要说明的是,在实际场景中,卸载请求 $(W, \Delta_i)$ 中的2个

量都是随机量,是变化的。本文为了便于建模分析,先考虑一种简单的情况,即所有的卸载请求 $(W, \Delta_i)$ 中的量都是常量。

在这种场景下,如图1所示,由SCeNBce $s$ 建立SCeNBce集群,需要说明的是,本文中SCC由服务基站 $s$ 和服务基站 $s$ 选定的SCeNBce集群基站共同组成。为了叙述简便,本文后续的服务基站 $s$ 选定的SCeNBce集群基站都称为集群基站,本文假定集群基站用集合 $SCs = \{1, 2, \dots, N\}$ 表示。

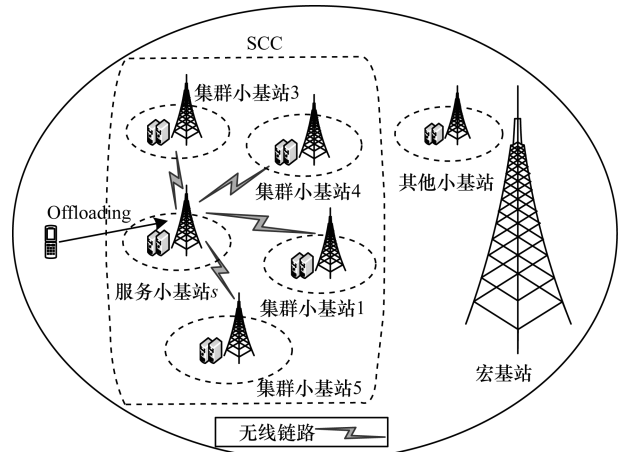


图1 SCC建立示意图

考虑到SCC建立的算法不仅要选择出较优的SCeNBce集群,还要确保算法带来的开销不能过高。在静态和动态算法的基础上,假定每一个集群建立周期,请求SCeNBce $s$ 处理的数据量可预测,本文提出一种基于信道状态和剩余可用计算资源的SCeNBce集群算法,算法过程如下:

1) 服务SCeNBce $s$ 向周边SCeNBce发送特定的参考信号,周边SCeNBce $n$ 将自己的剩余计算容量 $F_n$ 和参考信号接收功率报告给SCeNBce $s$ 。参考信号接收功率和剩余计算容量小于一个阈值的SCeNBce,主动不报告。为了避免同一个SCeNBce被多个SCC争夺,当SCeNBce收到多个参考信号接收功率时,选择最大的参考信号接收功率回复,当有多个参考信号接收功率相同时,随机选择一个回复。

2) 服务SCeNBce $s$ 收到众多SCeNBce的汇报,如表1所示,有了各个SCeNBce的参考信号接收功率和剩余计算容量,在一个时隙内根据以下优化算法选择最优的 $N$ 个SCeNBce作为这一时隙的SCeNBce集群。

$$\begin{aligned} \max \quad & \sum_{i=1}^N (\alpha \times rsrp_i + \beta \times F_i) \\ \text{s. t.} \quad & rsrp_i \geq rsrp_{th} \\ & F_i \geq F_{th} \\ & c \times \sum_{i=1}^N F_i \geq W \end{aligned} \quad (1)$$

其中, $\alpha$ 和 $\beta$ 为选取集群基站中参考信号和剩余可

用计算资源的影响因子; $rsrp_i$  表示 SCeNBce  $i$  的参考信号接收功率; $F_i$  表示 SCeNBce  $i$  的剩余计算容量; $c$  表示平均处理 1 bit 数据需要的时间,由 SCeNBce 的物理性质决定的一个常量。

表 1 上报到服务基站的参考信号接收功率和空闲的计算资源

序号	参考信号接收功率	空闲的计算资源
1	$rsrp_1$	$F_1$
2	$rsrp_2$	$F_2$
...	...	...
$i$	$rsrp_i$	$F_i$
...	...	...

3) SCeNBce  $s$  向被选择为 SCeNBce 集群的 SCeNBce 发送确定信号。收到确认信号的 SCC 在这一间隙被锁定,不会回复 SCC 发来的建立集群信号。值得注意的是, $N$  的值根据特定的部署场景设立一个确定的值。

#### 算法 1 SCC 建立算法

**步骤 1** 计算汇报到 SCC  $s$  的所有  $\alpha \times rsrp_i + \beta \times F_i$ ,  $i$  表示汇报基站的标识;

**步骤 2** 按照  $\alpha \times rsrp_i + \beta \times F_i$  降序排列,重新标识为  $(1, 2, \dots)$ , 令  $i = 1$ ;

**步骤 3** 判断  $c \times \sum_{n=1}^i F_n \leq W$ , 如果否, 转步骤 5;

**步骤 4** 令  $i = i + 1$ , 转到步骤 3;

**步骤 5** 令  $N = i$ , 算法结束。

服务基站 SCeNBce  $s$  将卸载请求  $(W, \Delta_i)$  中需要计算的所有数据按照实际情况分发给 SCC 内的集群基站 SCeNBce  $n$ , 本文用  $w_n$  表示分配到集群基站 SCeNBce  $n$  的数据量。每一个集群基站 SCeNBce  $n \in N$  的剩余计算容量为  $F_n$ , 那么运算时间就是  $\frac{w_n}{F_n}$ 。

在实际问题中, 卸载请求需要计算的数据量和计算完成后回传的数据量不是完全相等的, 本文为了方便讨论, 假设上传卸载请求到  $s$  的数据量和计算完成回传的数据量相同。

任意 2 个 SCeNBce 的无线信道链路可以实现的 最大传输速率根据香农公式可得:

$$R_{sn} = B_{sn} \text{lb} \left( 1 + \frac{a_{sn} p_{sn}}{N_0} \right) \quad (2)$$

其中,  $a_{sn}$  表示服务 SCeNBce  $s$  到集群 SCeNBce  $n$  的信道增益系数 (包含阴影衰落和路径损耗);  $N_0$  是噪声功率;  $p_{sn}$  表示小蜂窝  $s$  到  $n$  的发送总功率。

将卸载请求  $(W, \Delta_i)$  看成一个数据包, 那么当用户有很多个数据包发送到其服务的 SCeNBce 的时候, 由于无线信道容量不足以将这些数据包发送到集群蜂窝去, 那么这些数据包在服务 SCeNBce 处就会排队, 假设用户产生的数据包服从泊松分布, 平均到达率为  $\lambda$  (个/s), 平均服务速率为  $u$  (个/s), 如图 2 所示。

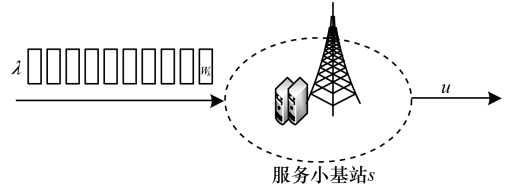


图 2 卸载任务排队模型的示意图

设允许的最大队长为  $l_{\max}$ , 则队列模型为  $M/M/1$ , 其中,  $m = 1$ 。由文献 [16] 可得, 平均队长  $\bar{l}$  为:

$$\bar{l} = \frac{\rho}{1 - \rho} \times \frac{1 - l_{\max} \rho^{l_{\max} - 1} + (l_{\max} - 1) \rho^{l_{\max}}}{1 - \rho^{(l_{\max} + 1)}} \quad (3)$$

其中,  $\rho = \frac{\lambda}{u}$ 。

值得注意的是, 平均服务速率  $u$  就是服务 SCeNBce  $s$  向集群 SCeNBce 发送数据包的平均速率。两者之间的关系如下:

假设功率控制的周期为  $n_i$  个 TTI,  $n_i$  由实际的情况来取一个定值, 则  $n_i$  个 TTI 由小蜂窝  $s$  向集群蜂窝  $n$  传输数据的平均速率  $\tilde{v}$  为:

$$\tilde{v} = \frac{1}{n_i} \sum_{i=1}^{n_i} \sum_{n \in N} R_{sn}^i \quad (4)$$

由上文的假设可知, 由 SCeNBce  $s$  向集群 SCeNBce  $n$  传输一个数据包所用的平均时间  $\tilde{t}$  为:

$$\tilde{t} = \frac{W}{\tilde{v}} \quad (5)$$

所以:

$$u = \frac{1}{\tilde{t}} = \frac{\tilde{v}}{W} \quad (6)$$

排队模型的平均等待时延  $t_{\text{delay}}$  为:

$$t_{\text{delay}} = \frac{\bar{l}}{u} \quad (7)$$

## 2 目标优化

用户通过比较本地处理运算任务和卸载任务到小蜂窝云运算这 2 种处理方式的功耗, 来决定是否进行任务卸载, 一旦决定将任务卸载到 SCC, 用户立刻生成数据包, 发起任务卸载请求。此时, 正在为用户服务的蜂窝云就会收到请求, 当该服务蜂窝云服务的同一时刻很多用户发起任务卸载请求时, 而由于小蜂窝云处理这些数据包 (即将数据包发送到集群蜂窝运算) 的速率有限, 因此会在该小蜂窝云处排队等候, 本文的目的就是在满足用户 QoS 的情况下, 尽可能降低能耗, 同时减小等待的时延, 提高用户的 QoE。

本文将问题建模如下:

$$\min \sum_{p_{sn}, w_n} \sum_{s \in S} \left\{ (p_{sn}) + \left( t_{\text{delay}} + \frac{w_n}{F_n} + \frac{2w_n}{R_{sn}} \right) \right\}$$

$$\begin{aligned}
\text{s. t. (a)} \quad & w_n \geq 0, \forall n \in N \\
\text{(b)} \quad & \sum_{n \in N} w_n = W \\
\text{(c)} \quad & 0 \leq p_{sn} \leq p_{\max}, \forall s \in S_{SC}, \forall n \in N \\
\text{(d)} \quad & 0 \leq \sum_{n \in N} p_{sn} \leq p_{\max} \\
\text{(e)} \quad & t_{\text{delay}} + \frac{w_n}{F_n} + \frac{2w_n}{R_{sn}} \leq \Delta_t, \forall n \in N
\end{aligned} \quad (8)$$

其中,(a)和(b)是确保卸载请求的所有数据都得到运算;(c)表示服务基站  $s$  向集群基站  $n$  发送数据的功率不能超过服务基站的最大发送功率;(d)表示服务基站向集群内所有基站的发送总功率不能超过服务基站的最大发送功率;(e)表示数据包的排队等待时延和传输时延与运算时延的总和不能超过所能容忍的最大时延,其中,第1项为等待时延;第2项为运算时延;第3项为传输时延。

由于直接求解原问题非常复杂,不容易求得。由建模过程可知,原问题可以分为2个子问题,优化目标的第1项为 SCeNBce 集群内服务 SCeNBce 的总发送功率,第2项为业务的总时延。

(e)可以分为两部分:一部分是队列等待时延

$t_{\text{delay}}$ ;另一部分是 SCC 中的传输和计算时延  $\frac{w_n}{F_n} + \frac{2w_n}{R_{sn}}$ 。其中,等待时延  $t_{\text{delay}}$  只与  $p_{sn}$  有关,SCC 中的

传输和计算时延  $\frac{w_n}{F_n} + \frac{2w_n}{R_{sn}}$  与  $w_n$  和  $p_{sn}$  都有关,所以求解原问题次优解可分为2步:1)得出  $w_n$  的分配值的次优解;2)固定  $w_n$ ,通过调整功率  $p_{sn}$  求得原问题的次优解。

考虑到优化目标中的发送功率  $p_{sn}$ ,传输时延和等待时延  $t_{\text{delay}}$  与信道质量和分配的数据量  $w_n$  直接相关,应尽可能地多为信道质量好的集群基站分配数据量,这样可以节省功率的消耗和传输时延,但此时可能信道质量好的集群基站剩余可用计算资源比较小,计算时延会很大。基于以上分析本文按照如下的方式分配  $w_n$ 。

由 SCC 集群建立过程可知,集群 SCeNBce  $n$  报告的参考信号接收功率  $rsrp_n$  实际就直接表示了信道质量的好坏,而  $F_n$  直接表征了剩余计算资源的数量。

所以令:

$$\begin{aligned}
\delta_n &= \rho \times \frac{-1}{rsrp_n} + \omega \times F_n \\
w_n &= W \times \frac{\delta_n}{\sum_1 \delta_n}
\end{aligned} \quad (9)$$

其中, $rsrp_n$  的单位为 dBm; $\rho$  和  $\omega$  的选取根据场景中信道质量和剩余可用计算资源的重要程度选取常量。

将  $w_n$  分配完成后,再通过调整发送功率  $p_{sn}$  就能得到优化目标的次优解。为了方便叙述,令:

$$t_{\text{cluster}}^n = \frac{w_n}{F_n} + \frac{2w_n}{R_{sn}} \quad (10)$$

$$f(p_{sn}) = (p_{sn}) + \left( t_{\text{delay}} + \frac{w_n}{F_n} + \frac{2w_n}{R_{sn}} \right) \quad (11)$$

由于 SCC 内的 SCeNBce  $n$  与服务 SCeNBce  $s$  之间的距离和信道状态不完全相同,并且各个 SCeNBce  $n$  剩余可用计算资源也不完全相同,因此为每个集群 SCeNBce  $n$  分配的数据量  $w_n$  也不相同,与每个分配的  $w_n$  对应都有一个 SCC 内传输和计算时延。对于一个数据包  $(W, \Delta_t)$  来说,所经历的时延是各个分配量  $w_n$  处理完成所经历的最大时延。即  $\max\left(\frac{w_n}{F_n} + \frac{2w_n}{R_{sn}}\right), n \in N$ ,显然,简单地将  $p_{sn}$  等量分配不能保证满足时延条件,也无法有效节省能量的损耗。基于此,本文提出一种启发式算法,可以有效寻找到较优的功率分配方案,同时满足系统时延要求,节省能耗。算法如下:

**算法2** 基于 SCC 的功率和卸载任务分配算法

1. 初始化:搜索步长  $x_0, W, p_{\max}, \Delta_t, \lambda, l_{\max}$ ;
2. 使用式(9)计算  $w_n$ ;
3. 遍历所有  $n \in \mathbb{N}$ ;
4. 令每个  $p_{sn} = x_0 \times i, i \in [1, 2, \dots, p_{\max}/x_0]$
5. 如果  $(\sum_{n \in N} p_{sn} \leq p_{\max})$
6. 对于  $\forall n \in \mathbb{N}$
7. 使用如下相应公式分别计算  $R_{sn}, t_{\text{delay}}, t_{\text{cluster}}^n$ ;
8. 式(2)  $\rightarrow R_{sn}$ , 式(3)、式(5)、式(6)、式(7)  $\rightarrow t_{\text{delay}}, t_{\text{cluster}}^n$ ;
9. 令  $t_{\text{cluster}} = \max(t_{\text{cluster}}^n)$ ;
10. 循环结束;
11. 如果结束;
12. 如果  $(t_{\text{delay}} + t_{\text{cluster}} \leq \Delta_t)$
13. 使用式(11)计算  $f(p_{sn})$ ;
14. 如果结束;
15. 循环结束。

### 3 性能仿真与结果分析

为了验证上述所提算法的有效性,参照 3GPP 标准,针对实际的网络场景,仿真参数设置如表 2 所示。

表 2 仿真参数设置

仿真参数	仿真值
微基站最大功率 $p_{\max}$ /dBm	30
系统带宽/MHz	10
数据包 $W$ /bit	$5 \times 10^6$
到达率 $\lambda$ /(个 $\cdot$ s $^{-1}$ )	5
剩余计算资源 $F_n$ /(bit $\cdot$ s $^{-1}$ )	$[1 \times 10^7, 15 \times 10^7]$
高斯噪声功率	-148.95 dBm/180 kHz
集群小蜂窝数量 $N$	5
穿透损耗/dB	15
搜索步长 $x_0$ /mW	20

仿真场景为单层建筑,双排房间,如图 3 所示,在每个房间随机部署 Femto。标号为 0 的房间部署的小基站为服务基站  $s$ ,标号为 1~5 的房间中部署的小基站为集群蜂窝基站。

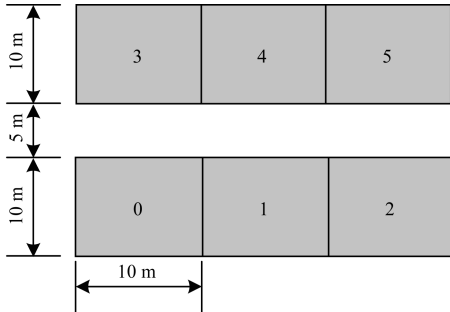


图 3 仿真场景示意图

假定 SCC 建立过程中汇报的各集群蜂窝(1~5)中的剩余计算资源为  $F = [4.5, 1.5, 3.5, 1.5, 3.5]$ ;各个集群基站的剩余计算资源  $F_n$  等于向量  $F$  中的数值乘以  $10^7$  bit/s。

首先简单说明另外 3 种方案:1)静态功率和负载分配(Static Power and Load Allocation, SPLA)方案为平均分配;2)动态负载分配(Dynamic Load Allocation, DLA)方案功率平均分配,负载按照所提算法分配;3)动态功率分配(Dynamic Power Allocation, DPA)方案功率按照所提算法分配,负载为平均分配。为了对比算法的性能,将以上 3 种方案作为小蜂窝云中功率和负载联合优化分配方案(Joint Optimization of the Power and Load allocation algorithm for Small Cell Cloud, JOPL-SCC)的对比模型。通过上述方案和本文所提方案的对比,可以看出本文算法性能有明显提升。

如前所述,按照图 3 所示的实际场景,在每个房间随机部署小基站,小基站的位置如图 4 所示。根据仿真测量结果,服务小基站 0 到其他各个集群小基站信道质量从高到低顺序为 3,1,4,2,5。

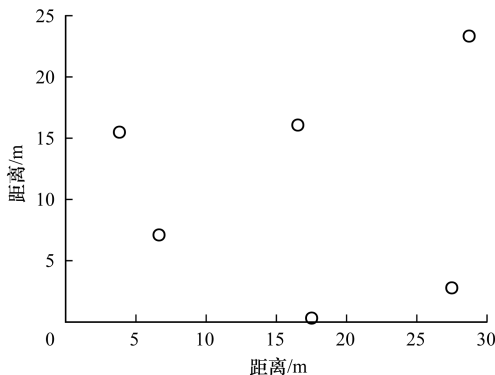


图 4 小基站部署位置示意图

图 5 所示为本文提出的卸载请求运算量分配策略在  $\rho$  和  $\omega$  取不同常量时,各个小基站分配的卸载请求运算量。当服务基站的参考信号功率设定为

300 mW 时,各个集群小基站的参考信号接收功率(RSRP)大约在  $-80$  dBm 左右,按照本文所提出的方案,当  $\rho = 1, \omega = 1$  时,由于  $\rho \times \frac{-1}{rsrp_n}$  和  $\omega \times F_n$  量级的差距非常大,从图中可以看处,卸载请求运算量的分配几乎是按照剩余计算资源比例来分配的,信道质量的影响几乎为 0。根据这种情况的分析,本文没有必要将  $\omega$  的值设置的比  $\rho$  大来进行卸载请求运算量的分配。同样,由于  $\rho \times \frac{-1}{rsrp_n}$  和  $\omega \times F_n$  量级的差异,当  $\rho$  设置相对于  $\omega$  过小时,信道质量相对于剩余计算资源的影响也不显著,所以当  $\rho = 10^3, \omega = 1$  和  $\rho = 10^5, \omega = 1$  时,信道质量的影响逐渐变大。选择其中一个简单例子说明所提算法的有效性,如集群小基站 5 所分配卸载请求运算量,随着  $\rho$  的增加,信道质量在卸载请求运算量分配中的影响逐渐增大,而服务小基站 0 到集群小基站 5 的信道质量最小,所以分配到集群小基站 5 的卸载请求运算量逐渐减小。

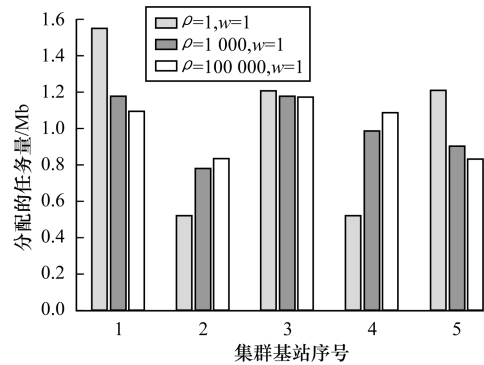


图 5 不同参数取值卸载任务量的分配比较

图 6 所示为 4 种方案的卸载请求运算量分配比较,其中动态负载分配方案(DLA)与小蜂窝云计算中功率和负载联合优化分配方案(JOPL-SCC) 2 种方案分配的卸载请求运算量相同,是按照本文所提方案,当  $\rho = 10^5, \omega = 1$  时分配的卸载请求运算量;静态功率和负载分配(SPLA)方案和动态功率分配(DPA)方案的卸载请求运算量分配都按照平均分配来进行。

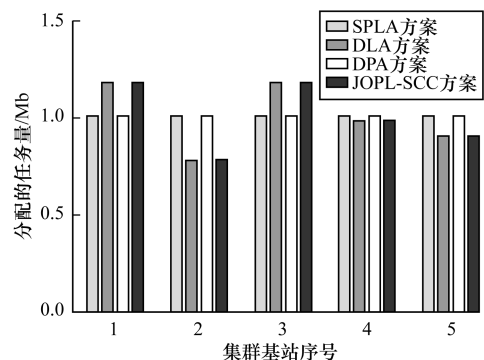


图 6 不同方案卸载任务量的分配比较

图7所示为服务小基站发送总功率为200 mW时,4种方案卸载请求运算量分量 $w_n$ 从服务基站0到各集群基站所经历的集群内总时延(传输时延与计算时延的和)。从图7可以看出,本文提出的方案使各个集群内总时延连线更平直。这是因为每个卸载请求运算量 $W$ 所经历的时延是各个分配量 $w_n$ 处理完成所经历的最大时延。所以,当各个卸载请求运算量分量 $w_n$ 所经历的集群内时延差异很小时,即各个集群内总时延连线越平直,算法的性能越好。设定搜索步长为20 mW时,算法最终搜索的功率分配如表3所示。比较图7和表3可知,当功率搜索时,尽可能地向集群内总时延大的集群小基站分配功率,减小各个集群内总时延分量的差异。结果表明本文提出的算法是有效的。而当发送总功率继续增大时,功率搜索的范围会更大,效果更显著。

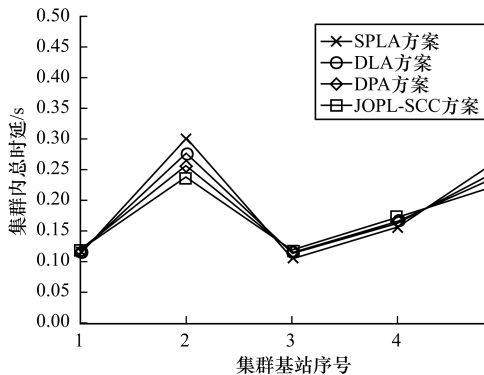


图7 不同方案集群内总时延的比较

表3 发送总功率为200 mW时各方案最终发送功率比较 mW

基站序号	SPLA	DPA	DLA	JOPL-SCC
1	40	20	40	20
2	40	80	40	80
3	40	20	40	20
4	40	20	40	20
5	40	20	40	20

图8所示为卸载任务所经历的总时延随着服务小基站 $s$ 发送总功率的变化情况。从图中可以看到2个交点分别是SPLA方案和DPA方案、DLA方案和本文JOPL-SCC方案的2个交点。这是因为SPLA方案和DPA方案、DLA方案和本文JOPL-SCC方案的卸载请求运算量分配分别相同。而在功率搜索开始时,初始功率也各自相同,所以卸载任务所经历的总时延两两相同,这是非常合理的。从图中还可以看出,SPLA会带来更大的集群时延,这是因为该方案没有通过卸载请求运算量分配和功率调整来减小集群内的最大时延。而另外其他3种方案集群内的总时延明显减少,与DLA和DPA相比,本文提出的JOPL-SCC方案对卸载请求运算量和功率

都进行了相应的调整,来减小各个卸载请求运算量分量 $w_n$ 所经历的集群内时延差异,集群内时延明显得到改善。

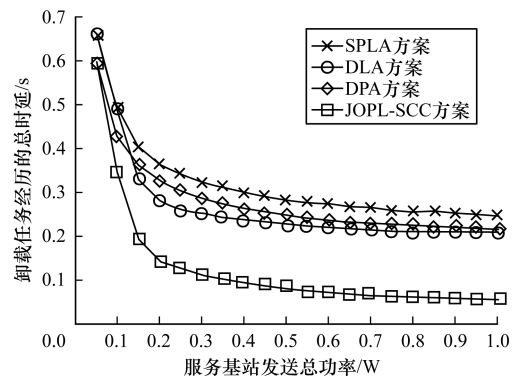


图8 不同方案卸载任务总时延的比较

图9所示为系统效用函数值随服务基站发送总功率的变化情况。如本文提出的优化目标一样,系统效用函数为服务基站发送总功率与卸载任务所经历的总时延的加权和。图中的2个交点原因与图8相同,不在赘述。从图中可以看出SPLA方案系统效用函数最大,这就表明该方案满足一定的时延条件,相比于其他3种方案,功耗就会比较大。或者在服务基站发送总功率与其他3种方案相同时,该方案带来的时延较大。而从另外3种方案可以明显看出,本文提出的JOPL-SCC方案系统效用函数值最小,性能最好。

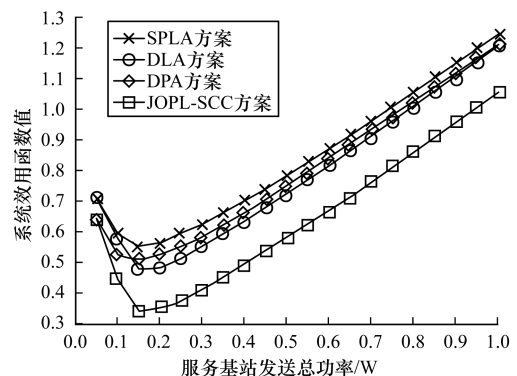


图9 不同方案系统效用函数比较

## 4 结束语

针对移动终端资源有限且本地服务基站资源严重不足的情况,以及移动终端的体验质量(QoE)严重降低的问题,本文提出一种小蜂窝云中功率和负载联合优化分配方案(JOPL-SCC)。该方案通过将移动终端的任务卸载到服务基站,由服务基站组建SCC,并将卸载的任务合理分发到SCC内运算;结合分发时的功率分配,达到减小卸载任务的处理时延,提高移动终端的QoE,同时还可以实现尽量降低系统能耗的目的。仿真结果表明,JOPL-SCC方案能够提高用户体验质量,同时节省系统能耗。

## 参考文献

- [ 1 ] Ma X,Zhao Y,Zhang L, et al. When Mobile Terminals Meet the Cloud: Computation Offloading as the Bridge[J]. IEEE Network,2013,27(5):28-33.
- [ 2 ] Chen X,Jiao L,Li W, et al. Efficient Multi-user Computation Offloading for Mobile-edge Cloud Computing [J]. IEEE/ACM Transactions on Networking,2016,24(5):2795-2808.
- [ 3 ] Fernando N, Seng W L, Rahayu W. Mobile Cloud Computing: A Survey [J]. Future Generation Computer Systems,2013,29(1):84-106.
- [ 4 ] Dinh H T,Lee C, Niyato D, et al. A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches [J]. Wireless Communications and Mobile Computing,2013,13(18):1587-1611.
- [ 5 ] Mao Y, Zhang J, Letaief K B. Dynamic Computation Offloading for Mobile-edge Computing with Energy Harvesting Devices[J]. IEEE Journal on Selected Areas in Communications,2016,34(12):3590-3605.
- [ 6 ] Barbarossa S,Sardellitti S,Lorenzo P D. Communicating While Computing: Distributed Mobile Cloud Computing over 5G Heterogeneous Networks [J]. IEEE Signal Processing Magazine,2014,31(6):45-55.
- [ 7 ] Munoz O,Pascual-Iserte A,Vidal J. Optimization of Radio and Computational Resources for Energy Efficiency in Latency-constrained Application Offloading [J]. IEEE Transactions on Vehicular Technology,2015,64(10):4738-4755.
- [ 8 ] Zhao T,Zhou S,Guo X, et al. A Cooperative Scheduling Scheme of Local Cloud and Internet Cloud for Delay-aware Mobile Cloud Computing [C]//Proceedings of IEEE GLOBECOM'15. Washington D. C., USA; IEEE Press,2015:1-6.
- [ 9 ] You C,Huang K,Chae H. Energy Efficient Mobile Cloud Computing Powered by Wireless Energy Transfer [J]. IEEE Journal on Selected Areas in Communications,2016,34(5):1757-1771.
- [ 10 ] Mao Y, Zhang J, Letaief K B. Dynamic Computation Offloading for Mobile-edge Computing with Energy Harvesting Devices [J]. IEEE Journal on Selected Areas in Communications,2016,34(12):3590-3605.
- [ 11 ] Oueis J,Strinati E C,Barbarossa S. Small Cell Clustering for Efficient Distributed Cloud Computing [C]//Proceedings of IEEE PIMRC'14. Washington D. C., USA; IEEE Press,2014:1474-1479.
- [ 12 ] Oueis J, Strinati E C, Barbarossa S. Distributed Mobile Cloud Computing: A Multi-user Clustering Solution [C]//Proceedings of ICC'16. Washington D. C., USA; IEEE Press,2016:1-6.
- [ 13 ] Oueis J, Strinati E C, Barbarossa S. The Fog Balancing: Load Distribution for Small Cell Cloud Computing [C]//Proceedings of VTC'15. Washington D. C., USA; IEEE Press,2015:51-63.
- [ 14 ] Ningning S,Chao G,Xingshuo A, et al. Fog Computing Dynamic Load Balancing Mechanism Based on Graph Repartitioning [J]. China Communications,2016,13(3):156-164.
- [ 15 ] Verma S,Yadav A K, Motwani D, et al. An Efficient Data Replication and Load Balancing Technique for Fog Computing Environment [C]//Proceedings of the 3rd International Conference on Computing for Sustainable Global Development. Washington D. C., USA; IEEE Press,2016:2888-2895.
- [ 16 ] 周炯槃. 通信网理论基础 [M]. 北京:人民邮电出版社,2009.

编辑 索书志

(上接第 48 页)

## 参考文献

- [ 1 ] 马 鸣. 云计算模式区域医疗卫生信息化平台建设探索 [J]. 医学信息杂志,2013,34(1):19-24.
- [ 2 ] 程 龙. 基于 OpenStack 的动态资源调度方法的研究与实现 [D]. 西安:西北大学,2014.
- [ 3 ] 涂聘俊. 基于 OpenStack 的业务云平台的动态可伸缩性研究与实现 [D]. 北京:北京邮电大学,2014.
- [ 4 ] 高 芳. 基于 OpenStack 的业务云平台资源分配模型的研究与实现 [D]. 北京:北京邮电大学,2013.
- [ 5 ] 张子凡. OpenStack 部署实践 [M]. 北京:人民邮电出版社,2014.
- [ 6 ] 张小斌. OpenStack 企业云平台架构与实践 [M]. 北京:电子工业出版社,2015.
- [ 7 ] 英特尔开源技术中心. OpenStack 设计与实现 [M]. 北京:电子工业出版社,2015.
- [ 8 ] 付 勋. 基于 OpenStack 的云平台资源调度算法设计与实现 [D]. 北京:北京邮电大学,2014.
- [ 9 ] 蔚欢乐. 基于 OpenStack 的资源动态分配框架的设计与实现 [D]. 西安:西北大学,2014.
- [ 10 ] 汪 楠. 基于 OpenStack 云平台的计算资源动态调度及管理 [D]. 大连:大连理工大学,2013.
- [ 11 ] 林伟伟. 云计算资源调度研究综述 [J]. 计算机科学,2012,39(10):1-6.
- [ 12 ] Padala P, Shin K G, Zhu Xiaoyun, et al. Adaptive Control of Virtualized Resources in Utility Computing Environments [J]. ACM SIGOPS Operating Systems Review,2007,41(3):289-302.
- [ 13 ] Wen Xiaolong, Gu Genqiang, Li Qingchun, et al. Comparison of Open-source Cloud Management Platform: OpenStack and OpenNebula [C]//Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery. Washington D. C., USA; IEEE Press,2012:2457-2461.
- [ 14 ] 张晋芳,王清心,丁家满,等. 一种云计算环境下大数据动态迁移策略 [J]. 计算机工程,2016,42(5):13-17.
- [ 15 ] 陈丽平,郭 鑫. 最小能耗优化云模型中的动态图挖掘方法 [J]. 计算机工程,2015,41(8):23-31.
- [ 16 ] 冯秀玲. 云计算环境下的负载均衡算法的研究与设计 [D]. 北京:北京邮电大学,2012.
- [ 17 ] 葛 新. 基于云计算集群扩展中的调度问题研究 [D]. 合肥:中国科学技术大学,2011.
- [ 18 ] 李 冰. 云计算环境下动态资源管理关键技术 [D]. 北京:北京邮电大学,2012.
- [ 19 ] 曾智斌. 云计算中高能效的虚拟资源分配策略 [J]. 计算机系统应用,2011,20(12):55-59.

编辑 金胡考