



(12) 发明专利申请

(10) 申请公布号 CN 112052092 A

(43) 申请公布日 2020.12.08

(21) 申请号 202010929293.7

(22) 申请日 2020.09.07

(71) 申请人 中山大学

地址 510275 广东省广州市海珠区新港西路135号

(72) 发明人 刘芳 梁家越 金航 肖依

(74) 专利代理机构 广州粤高专利商标代理有限公司 44102

代理人 张金福

(51) Int. Cl.

G06F 9/50 (2006.01)

G06N 3/12 (2006.01)

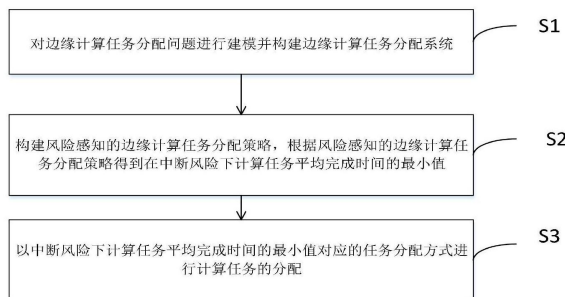
权利要求书4页 说明书11页 附图2页

(54) 发明名称

一种风险感知的边缘计算任务分配方法

(57) 摘要

本发明公开了一种风险感知的边缘计算任务分配方法,包括以下步骤:S1:对边缘计算任务分配问题进行建模并构建边缘计算任务分配系统;S2:构建风险感知的边缘计算任务分配策略,根据风险感知的边缘计算任务分配策略得到在中断风险下计算任务平均完成时间的最小值;S3:以中断风险下计算任务平均完成时间的最小值对应的任务分配方式进行计算任务的分配。本发明通过将任务分配至执行风险小的计算节点,减少任务中断发生的可能性,并最小化任务的平均完成时间,提高了系统的鲁棒性。



1. 一种风险感知的边缘计算任务分配方法,其特征在于,包括以下步骤:

S1:对边缘计算任务分配问题进行建模并构建边缘计算任务分配系统;

S2:构建风险感知的边缘计算任务分配策略,根据风险感知的边缘计算任务分配策略得到在中断风险下计算任务平均完成时间的最小值;

S3:以中断风险下计算任务平均完成时间的最小值对应的任务分配方式进行计算任务的分配。

2. 根据权利要求1所述的一种风险感知的边缘计算任务分配方法,其特征在于,所述对边缘计算任务分配问题建模包括:构建边缘云、定义边缘服务、定义任务分配决策,

其中边缘云包括:控制器、若干边缘计算节点,所述控制器与若干边缘计算节点均通过网络连接,所述控制器接收用户的任务并分配至合适的边缘计算节点,每个边缘计算节点负责执行任务并返回结果至控制器,其中边缘计算节点集合记为EN, $EN = \{en_1, en_2, \dots, en_j\}$, EN中有J个边缘计算节点,每个边缘计算节点与控制器的通信带宽表示为 tx_j , $1 \leq j \leq J$,设在每个边缘计算节点内部,维持若干个工作线程用以执行任务,并维持一个等待队列用以缓存未分配执行的任務,将边缘计算节点j, $1 \leq j \leq J$ 内部的等待队列长度表示为 $q1_j$,将每个边缘计算节点内部平均任务执行时间表示为 T_j^{Avg} ;

边缘服务:所述边缘服务由用户发起,每个边缘服务接收一个任务和该任务对应的数据文件,通过执行任务返回结果,在每一轮任务调度中,将任务集记为 $TK = \{tk_1, tk_2, \dots, tk_I\}$,其中I代表任务集中的任务总数,对应任务i, $1 \leq i \leq I$,将该任务对应的数据文件大小记为 s_i ,定义 t_{ij}^{exe} 来表示任务i在边缘计算节点j的执行时间,其中 $1 \leq i \leq I$, $1 \leq j \leq J$;

任务分配决策:对于一个任务集合和一个边缘计算节点集合,任务分配决策定义了为每一个任务选择的边缘计算节点,可使用矩阵表示为 $X = \{x_{1,1}, x_{1,2}, \dots, x_{1,I}, \dots, x_{i,j}, \dots, x_{I,1}, \dots, x_{I,J}\}$, $1 \leq i \leq I$, $1 \leq j \leq J$,其中 $x_{i,j} = 1$ 表示将任务i分配至边缘计算节点j, $x_{i,j} = 0$ 则无此分配关系。

3. 根据权利要求2所述的一种风险感知的边缘计算任务分配方法,其特征在于,所述控制器和边缘计算节点构建了边缘计算任务分配系统,所述控制器包括有:任务提交接口、第一任务管理器、调度器、任务跟踪器、预测器、节点管理器,

所述任务提交接口响应用户的服务请求,并接收任务和其他对应数据文件;

所述第一任务管理器使用等待队列管理新到达的计算任务和需要重新调度的计算任务并在每个计算任务分配周期,将所有等待计算任务提交至调度器;

节点管理器模块用于维持与边缘计算节点的组件的通信连接,并收集每个边缘节点的运行信息;

预测器用于产生计算任务被分配至边缘计算节点的完整时间的预测值和计算任务中断风险下的期望完成时间的预测值,所述预测器包括一个任务单元和一个节点预测单元,每个任务预测单元为一组具有类似软硬件配置的边缘计算节点维护已执行任务的时间分布信息并提供执行时间的预估值,使用机器类型来标记同一组边缘节点;

所述调度器在每个调度周期开始时,从预测器收集各计算任务-边缘计算节点对的完成时间预测值,并作出调度决策;

所述任务跟踪器将各计算任务分配至对应的边缘计算节点,并监控各计算任务的运行

情况；

所述边缘计算节点包含心跳线程、第二任务管理器和若干工作线程，所述第二任务管理器接收来自控制器的计算任务，并加入等待队列中；

所述若干工作线程中若有空闲的工作线程，则空闲的工作线程从等待队列中选择一个计算任务并执行；

心跳线程负责定时发送本节点的运行信息至控制器。

4. 根据权利要求3所述的一种风险感知的边缘计算任务分配方法，其特征在于，所述构建风险感知的边缘计算任务分配策略包括：

S201构建计算任务分配至边缘计算节点的完整时间的表达式；

S202以最小化计算任务的平均完成时间为目标建立目标函数和约束条件；

S203量化潜在的计算任务中断对任务完成时间带来的额外开销时间，将额外开销时间加上无风险时计算完成时间作为中断风险下的完成时间对步骤S202中的目标函数进行更新；

S204求解更新后的目标函数的最优解。

5. 根据权利要求4所述的一种风险感知的边缘计算任务分配方法，其特征在于，计算任务被分配至边缘计算节点的完整时间记为 t_{ij}^{cpt} ，

$$t_{ij}^{cpt} = t_{ij}^{tr} + t_{ij}^{wt} + t_{ij}^{ex} \quad (1)$$

每个任务的完整运行包含若干过程，分别记为：第一过程、第二过程、第三过程，所述第一过程为任务及其数据文件被传输至对应的边缘计算节点，并加入该节点内部的等待队列中；所述第二过程为如果边缘计算节点内部的没有空闲的工作线程，则任务需要在等待队列中等待直至被分配至某个空闲线程；所述第三个过程为任务被某个空闲线程调用执行；

其中，计算任务在第一个过程的传输时间记为 t_{ij}^{tr} ，其计算公式为：

$$t_{ij}^{tr} = \frac{S_i}{tx_j} \quad (2)$$

计算任务在第二个过程的等待时间记为 t_{ij}^{wt} ，其计算公式为：

$$t_{ij}^{wt} = ql_j \times T_j^{Avg} \quad (3)$$

计算任务在第三个过程的执行时间即 t_{ij}^{ex} 。

6. 根据权利要求5所述的一种风险感知的边缘计算任务分配方法，其特征在于，以最小化计算任务的平均完成时间为目标建立目标函数和约束条件，所述目标函数表达式为：

$$\min_{i,j} \sum_{i,j} (x_{i,j} \times t_{i,j}^{cpt}) + \left(\sum_i x_{i,j} - 1 \right) \times T_j^{Avg},$$

约束条件为：

$$s. t. \sum_i x_{i,j} = 1,$$

$$x_{i,j} \in \{0, 1\}, 1 \leq i \leq I, 1 \leq j \leq J \quad (4)$$

其中，目标函数中将计算任务分配问题转化为了线性规划问题，规划问题的代价为批量任务完成时间之和， $(\sum_i x_{i,j} - 1) \times T_j^{Avg}$ 为补偿项，所述补偿项表示将多个任务分配至同

一个节点时的各任务的额外等待时间之和。

7. 根据权利要求6所述的一种风险感知的边缘计算任务分配方法,其特征在于,量化潜在的计算任务中断对任务完成时间带来的额外开销时间,将额外开销时间加上无风险时计算完成时间作为中断风险下的完成时间对步骤S202中的目标函数进行优化,具体包括:

将边缘节点的预计留存时间和边缘节点在边缘云中已读过的时间两者差值定义为边缘节点的剩余存留时间,当将一个计算任务分配至一个边缘计算节点,设该边缘计算节点的剩余存留时间为一个随机变量RPT,其概率分布律表示为:

$$p_m = P(RPT = rpt_m), m = 1, 2, \dots, M, \quad (5)$$

其中, rpt_m 代表RPT的所有可能取值, M 代表这些取值的数量; 设任务的完成时间为一个随机变量CT, 其概率分布律表示为:

$$q_n = P(CT = ct_n), n = 1, 2, \dots, N, \quad (6)$$

其中, ct_n 代表CT的所有可能取值, N 代表所有可能取值的数量;

由于每次任务分配决策是相互独立的,故在量化单次任务中断的时间开销时假设任务的第二次执行必然完成,又假设计算任务在重新调度时也被分配至一个软硬件配置相似的边缘计算节点,也即第二次正常运行时间与第一次正常运行时间相同,则定义在计算任务中断风险下的期望完成时间为随机变量TCT,表达式如下:

$$TCT = \begin{cases} CT, & RPT > CT, \\ RPT + CT, & RPT \leq CT. \end{cases} \quad (7)$$

随机变量TCT的数学期望表达式为:

$$\begin{aligned} E(TCT) &= \sum_{rpt_m > ct_n} (p_m q_n ct_n) + \sum_{rpt_m \leq ct_n} [p_m q_n (rpt_m + ct_n)] \\ &= \sum (p_m q_n ct_n) + \sum_{rpt_m \leq ct_n} p_m q_n rpt_m \\ &= E(CT) + \sum_{rpt_m \leq ct_n} p_m q_n rpt_m, \end{aligned} \quad (8)$$

由公式(8)知,引入计算任务中断风险后,任务的期望完成时间从 $E(CT)$ 变成了 $E(TCT)$,后者比前者多了一项,也即是潜在计算任务中断带来的额外开销时间的期望值,故定义潜在计算任务中断带来的额外开销时间为ET,其期望值如公式(9)所示:

$$E(ET) = \sum_{rpt_m \leq ct_n} p_m q_n rpt_m, \quad (9)$$

定义 t_{ij}^{et} 为ET的预估值,定义在计算任务中断风险下的期望完成时间为 t_{ij}^{ect} ,其值等于正常完成时间 t_{ij}^{cpt} 与 t_{ij}^{et} 之和,如公式(10)所示:

$$t_{ij}^{ect} = t_{ij}^{cpt} + t_{ij}^{et} \quad (10)$$

将 t_{ij}^{ect} 取代 t_{ij}^{cpt} ,建立风险感知的任务分配策略,将线性规划问题更新如公式(11)所示:

$$\min_{i,j} \sum_{i,j} (x_{i,j} \times t_{i,j}^{ept}) + \left(\sum_i x_{i,j} - 1 \right) \times T_j^{Avg},$$

$$s.t. \sum_i x_{i,j} = 1,$$

$$x_{i,j} \in \{0,1\}, 1 \leq i \leq I, 1 \leq j \leq J, \quad (11)。$$

8. 根据权利要求7所述的一种风险感知的边缘计算任务分配方法,其特征在于,求解更新后的目标函数的最优解是采用遗传算法在设定的迭代次数内获得近似最优解,在编码过程中,对于任务集 $TK = \{tk_1, tk_2, \dots, tk_I\}$ 和边缘计算节点集 $EN = \{en_1, en_2, \dots, en_J\}$,一种任务分配决策 $X = \{x_{1,1}, x_{1,2}, \dots, x_{1,I}, \dots, x_{i,j}, \dots, x_{I,1}, \dots, x_{I,J}\}$ 被编码为一个向量 $S = \{s_i = en_j, 1 \leq i \leq I, 1 \leq j \leq J\}$,代表一条染色体;其中, $s_i = en_j$ 表示将任务 i 调度至边缘计算节点。

9. 根据权利要求8所述的一种风险感知的边缘计算任务分配方法,其特征在于,遗传算法中适应度函数表达式为:

$$Fitness(TK, EN, X) = \frac{1}{C(X)} \quad (12)$$

$$C(X) = \sum_{i,j} (x_{i,j} \times t_{i,j}^{ept}) + (\sum_i x_{i,j} - 1) \times T_j^{Avg} \quad (13)。$$

10. 根据权利要求9所述的一种风险感知的边缘计算任务分配方法,其特征在于,预测器采用黑盒方法来为每个计算任务产生概率分布和预测值。

一种风险感知的边缘计算任务分配方法

技术领域

[0001] 本发明涉及边缘计算技术领域,更具体地,涉及一种风险感知的边缘计算任务分配方法。

背景技术

[0002] 边缘计算技术作为分布式计算技术得到了迅速的发展,各种边缘计算平台或框架也被推出。边缘计算平台被部署于网络边缘上的服务器上,承担资源管理,设备管理和任务执行等功能。在此基础上,一群部署在网络边缘上的服务器组成边缘云,为用户提供低延迟的边缘服务,以满足实时交互应用的需求。

[0003] 任务分配是边缘云整体运行的关键环节,边缘云中的控制器基于任务分配策略为每个任务选择合适的计算节点。在边缘计算任务分配问题上,当前已有一些工作对用户移动性、边缘计算节点的异构性、网络带宽的动态变化性等挑战开展了研究,然而针对边缘计算节点动态性和不确定性问题的研究比较少见。不同于云计算集群调度的环境,边缘计算环境下的计算节点具有多样性,可以是边缘服务器,或云中心服务器,甚至是移动设备。这些节点处于不同的网络环境中,且部分边缘计算节点处于移动网络中,而移动网络并不如有线网络稳定,其网络中断可能性相对有线网络较大。此外,具有移动性的移动设备还可能因位置变化而不定时退出边缘云。不论是网络中断或边缘计算节点退出等情况,主节点与边缘计算节点的交互连接将断开,在边缘计算节点中未执行完毕的任务将被迫中断,调度器需要将任务重新分配至合适的边缘计算节点重新执行。尽管目前已有检查点等技术可以保存部分任务的运行状态并恢复执行,能减少任务重新执行的时间开销,但并不能减少任务中断的发生频率。

[0004] 在边缘计算中任务分配问题上,现有的多数研究工作通常假设任务在执行过程中是不会出错或者中断的,并没有考虑边缘计算节点的动态性和不稳定性。

[0005] 在现有技术中,公开号为CN110198339A的中国发明专利,于2019年9月3日公开了一种基于QoE感知的边缘计算任务调度方法。该专利提出得分方法周期性收集各设备资源状态信息和网络状态信息。在任务分配过程中,根据预测模型估计任务在不同设备的执行时间,并基于设备资源状态信息和网络状态信息,建立线性规划模型,并求解得到任务调度结果。该专利提出的方法考虑了设备的网络状态和资源状态,能实现各设备之间的负载均衡,提高任务的执行效率。但该方案并没有考虑任务分配过程的鲁棒性。在物联网场景中,边缘设备具有异构性,智能手机和移动电脑等都可以作为边缘设备。这些设备可能因位置变动而脱离原先所在的网络,导致已任务的任务发生中断,需要重新分配和执行。因此,为了提高任务分配的鲁棒性,减少任务中断情况的发生,研究边缘计算中具有风险感知的任务分配策略具有重要意义。

发明内容

[0006] 本发明为克服上述现有技术中边缘计算任务分配中没有考虑边缘计算节点的动

态性和不稳定性,鲁棒性低的缺陷,提供一种风险感知的边缘计算任务分配方法。

[0007] 本发明的首要目的是为解决上述技术问题,本发明的技术方案如下:

[0008] 一种风险感知的边缘计算任务分配方法,包括以下步骤:

[0009] S1:对边缘计算任务分配问题进行建模并构建边缘计算任务分配系统;

[0010] S2:构建风险感知的边缘计算任务分配策略,根据风险感知的边缘计算任务分配策略得到在中断风险下计算任务平均完成时间的最小值;

[0011] S3:以中断风险下计算任务平均完成时间的最小值对应的任务分配方式进行计算任务的分配。

[0012] 本方案中,所述对边缘计算任务分配问题建模包括:构建边缘云、定义边缘服务、定义任务分配决策,

[0013] 其中边缘云包括:控制器、若干边缘计算节点,所述控制器与若干边缘计算节点均通过网络连接,所述控制器接收用户的任务并分配至合适的边缘计算节点,每个边缘计算节点负责执行任务并返回结果至控制器,其中边缘计算节点集合记为 EN , $EN = \{en_1, en_2, \dots, en_J\}$, EN 中有 J 个边缘计算节点,每个边缘计算节点与控制器的通信带宽表示为 tx_j , $1 \leq j \leq J$,设在每个边缘计算节点内部,维持若干个工作线程用以执行任务,并维持一个等待队列用以缓存未分配执行的任務,将边缘计算节点 j , $1 \leq j \leq J$ 内部的等待队列长度表示为 ql_j ,将每个边缘计算节点内部平均任务执行时间表示为 T_j^{Avg} ;

[0014] 边缘服务:所述边缘服务由用户发起,每个边缘服务接收一个任务和该任务对应的数据文件,通过执行任务返回结果,在每一轮任务调度中,将任务集记为 $TK = \{tk_1, tk_2, \dots, tk_I\}$,其中 I 代表任务集中的任务总数,对应任务 i , $1 \leq i \leq I$,将该任务对应的数据文件大小记为 s_i ,由于在边缘云中的各边缘计算节点并非同构的,可能具有不同的软硬件配置,因此在同一任务在不同边缘计算节点中的执行时间也不相同。定义 t_{ij}^{exe} 来表示任务 i 在边缘计算节点 j 的执行时间,其中 $1 \leq i \leq I$, $1 \leq j \leq J$;

[0015] 任务分配决策:对于一个任务集合和一个边缘计算节点集合,任务分配决策定义了为每一个任务选择的边缘计算节点,可使用矩阵表示为 $X = \{x_{1,1}, x_{1,2}, \dots, x_{1,I}, \dots, x_{i,j}, \dots, x_{I,1}, \dots, x_{I,J}\}$, $1 \leq i \leq I$, $1 \leq j \leq J$,其中 $x_{i,j} = 1$ 表示将任务 i 分配至边缘计算节点 j , $x_{i,j} = 0$ 则无此分配关系。

[0016] 本方案中,所述控制器和边缘计算节点构建了边缘计算任务分配系统,所述控制器包括有:任务提交接口、第一任务管理器、调度器、任务跟踪器、预测器、节点管理器,

[0017] 所述任务提交接口响应用户的服务请求,并接收任务和其他对应数据文件;

[0018] 所述第一任务管理器使用等待队列管理新到达的计算任务和需要重新调度的计算任务并在每个计算任务分配周期,将所有等待计算任务提交至调度器;

[0019] 节点管理器模块用于维持与边缘计算节点的组件的通信连接,并收集每个边缘节点的运行信息;

[0020] 预测器用于产生计算任务被分配至边缘计算节点的完整时间的预测值和计算任务中断风险下的期望完成时间的预测值,所述预测器包括一个任务单元和一个节点预测单元,每个任务预测单元为一组具有类似软硬件配置的边缘计算节点维护已执行任务的时间分布信息并提供执行时间的预估值,使用机器类型来标记同一组边缘节点;

[0021] 所述调度器在每个调度周期开始时,从预测器收集各计算任务-边缘计算节点对的完成时间预测值,并作出调度决策;

[0022] 所述任务跟踪器将各计算任务分配至对应的边缘计算节点,并监控各计算任务的运行情况;

[0023] 所述边缘计算节点包含心跳线程、第二任务管理器和若干工作线程,所述第二任务管理器接收来自控制器的计算任务,并加入等待队列中;

[0024] 所述若干工作线程中若有空闲的工作线程,则空闲的工作线程从等待队列中选择一个计算任务并执行;

[0025] 心跳线程负责定时发送本节点的运行信息至控制器。

[0026] 本方案中,所述构建风险感知的边缘计算任务分配策略包括:

[0027] S201构建计算任务分配至边缘计算节点的完整时间的表达式;

[0028] S202以最小化计算任务的平均完成时间为目标建立目标函数和约束条件;

[0029] S203量化潜在的计算任务中断对任务完成时间带来的额外开销时间,将额外开销时间加上无风险时计算完成时间作为中断风险下的完成时间对步骤S202中的目标函数进行更新;

[0030] S204求解更新后目标函数的最优解。

[0031] 本方案中,计算任务被分配至边缘计算节点的完整时间记为 t_{ij}^{cpt} ,

$$[0032] \quad t_{ij}^{cpt} = t_{ij}^{tr} + t_{ij}^{wt} + t_{ij}^{ex} \quad (1)$$

[0033] 每个任务的完整运行包含若干过程,分别记为:第一过程、第二过程、第三过程,所述第一过程为任务及其数据文件被传输至对应的边缘计算节点,并加入该节点内部的等待队列中;所述第二过程为如果边缘计算节点内部的没有空闲的工作线程,则任务需要在等待队列中等待直至被分配至某个空闲线程;所述第三个过程为任务被某个空闲线程调用执行;

[0034] 其中,计算任务在第一个过程的传输时间记为 t_{ij}^{tr} ,其计算公式为:

$$[0035] \quad t_{ij}^{tr} = \frac{s_i}{tx_j} \quad (2)$$

[0036] 计算任务在第二个过程的等待时间记为 t_{ij}^{wt} ,其计算公式为:

$$[0037] \quad t_{ij}^{wt} = ql_j \times T_j^{Avg} \quad (3)$$

[0038] 计算任务在第三个过程的执行时间即 t_{ij}^{ex} 。

[0039] 本方案中,以最小化计算任务的平均完成时间为目标建立目标函数和约束条件,所述目标函数表达式为:

$$[0040] \quad \min \sum_{ij} (x_{ij} \times t_{ij}^{cpt}) + \left(\sum_i x_{ij} - 1 \right) \times T_j^{Avg},$$

[0041] 约束条件为:

$$[0042] \quad \text{s. t. } \sum_i x_{i,j} = 1,$$

$$[0043] \quad x_{i,j} \in \{0, 1\}, 1 \leq i \leq I, 1 \leq j \leq J \quad (4)$$

[0044] 其中,目标函数中将计算任务分配问题转化为了线性规划问题,规划问题的代价为批量任务完成时间之和, $(\sum_i x_{ij} - 1) \times T_j^{Avg}$ 为补偿项,所述补偿项表示将多个任务分配至同一个节点时的各任务的额外等待时间之和。

[0045] 本方案中,量化潜在的计算任务中断对任务完成时间带来的额外开销时间,将额外开销时间加上无风险时计算完成时间作为中断风险下的完成时间对步骤S202中的目标函数进行优化,具体包括:

[0046] 将边缘节点的预计留存时间和边缘节点在边缘云中已读过的时间两者差值定义为边缘节点的剩余存留时间,当将一个计算任务分配至一个边缘计算节点,设该边缘计算节点的剩余存留时间为一个随机变量RPT,其概率分布律表示为:

$$[0047] \quad p_m = P(RPT = rpt_m), m = 1, 2, \dots, M, \quad (5)$$

[0048] 其中 rpt_m 代表RPT的所有可能取值, M 代表这些取值的数量;设任务的完成时间为一个随机变量CT,其概率分布律表示为:

$$[0049] \quad q_n = P(CT = ct_n), n = 1, 2, \dots, N, \quad (6)$$

[0050] 其中 ct_n 代表CT的所有可能取值, N 代表所有可能取值的数量;

[0051] 由于每次任务分配决策是相互独立的,故在量化单次任务中断的时间开销时假设任务的第二次执行必然完成,又假设计算任务在重新调度时也被分配至一个软硬件配置相似的边缘计算节点,也即第二次正常运行时间与第一次正常运行时间相同,则定义在计算任务中断风险下的期望完成时间为随机变量TCT,表达式如下:

$$[0052] \quad TCT = \begin{cases} CT, & RPT > CT, \\ RPT + CT, & RPT \leq CT. \end{cases} \quad (7)$$

[0053] 随机变量TCT的数学期望表达式为:

$$[0054] \quad \begin{aligned} E(TCT) &= \sum_{rpt_m > ct_n} (p_m q_n ct_n) + \sum_{rpt_m \leq ct_n} [p_m q_n (rpt_m + ct_n)] \\ &= \sum (p_m q_n ct_n) + \sum_{rpt_m \leq ct_n} p_m q_n rpt_m \\ &= E(CT) + \sum_{rpt_m \leq ct_n} p_m q_n rpt_m, \end{aligned} \quad (8)$$

[0055] 由公式(8)知,引入计算任务中断风险后,任务的期望完成时间从 $E(CT)$ 变成了 $E(TCT)$,后者比前者多了一项,也即是潜在计算任务中断带来的额外开销时间的期望值,故定义潜在计算任务中断带来的额外开销时间为ET,其期望值如公式(9)所示:

$$[0056] \quad E(ET) = \sum_{rpt_m \leq ct_n} p_m q_n rpt_m, \quad (9)$$

[0057] 定义 t_{ij}^{et} 为ET的预估值,定义在计算任务中断风险下的期望完成时间为 t_{ij}^{ect} ,其值等于正常完成时间 t_{ij}^{cpt} 与 t_{ij}^{et} 之和,如公式(10)所示:

$$[0058] \quad t_{ij}^{ect} = t_{ij}^{cpt} + t_{ij}^{et} \quad (10)$$

[0059] 将 t_{ij}^{ect} 取代 t_{ij}^{cpt} ,建立风险感知的任务分配策略,将线性规划问题更新如公式(11)

所示：

$$[0060] \quad \min \sum_{i,j} (x_{i,j} \times t_{i,j}^{ept}) + \left(\sum_i x_{i,j} - 1 \right) \times T_j^{Avg},$$

$$[0061] \quad s.t. \quad \sum_i x_{i,j} = 1,$$

$$[0062] \quad x_{i,j} \in \{0,1\}, 1 \leq i \leq I, 1 \leq j \leq J, \quad (11)$$

[0063] 本方案中，求解更新后目标函数的最优解是采用遗传算法在设定的迭代次数内获得近似最优解，在编码过程中，对于任务集 $TK = \{tk_1, tk_2, \dots, tk_I\}$ 和边缘计算节点集 $EN = \{en_1, en_2, \dots, en_J\}$ ，一种任务分配决策 $X = \{x_{1,1}, x_{1,2}, \dots, x_{1,I}, \dots, x_{i,j}, \dots, x_{I,1}, \dots, x_{I,J}\}$ 被编码为一个向量 $S = \{s_i = en_j, 1 \leq i \leq I, 1 \leq j \leq J\}$ ，代表一条染色体；其中， $s_i = en_j$ 表示将任务 i 调度至边缘计算节点 j ，

[0064] 本方案中，遗传算法中适应度函数表达式为：

$$[0065] \quad \text{Fitness}(TK, EN, X) = \frac{1}{C(X)} \quad (12)$$

$$[0066] \quad C(X) = \sum_{i,j} (x_{i,j} \times t_{i,j}^{ept}) + \left(\sum_i x_{i,j} - 1 \right) \times T_j^{Avg} \quad (13)。$$

[0067] 本方案中，预测器采用黑盒方法来为每个计算任务产生概率分布和预测值。

[0068] 与现有技术相比，本发明技术方案的有益效果是：

[0069] 本发明基于构建的边缘计算任务分配系统通过风险感知的边缘计算任务分配策略进行计算任务的分配，充分考虑了边缘计算节点的动态性和不稳定性，通过将任务分配至执行风险小的计算节点，减少任务中断发生的可能性，并最小化任务的平均完成时间，提高了系统的鲁棒性。

附图说明

[0070] 图1为本发明方法流程图。

[0071] 图2为基于边缘设备组成的边缘云示意图。

[0072] 图3为边缘计算任务分配系统图。

[0073] 图4为本发明中任务分配策略各种时间值的概率分布关系图。

具体实施方式

[0074] 为了能够更清楚地理解本发明的上述目的、特征和优点，下面结合附图和具体实施方式对本发明进行进一步的详细描述。需要说明的是，在不冲突的情况下，本申请的实施例及实施例中的特征可以相互组合。

[0075] 在下面的描述中阐述了很多具体细节以便于充分理解本发明，但是，本发明还可以采用其他不同于在此描述的方式来实施，因此，本发明的保护范围并不受下面公开的具体实施例的限制。

[0076] 实施例1

[0077] 如图1所示，一种风险感知的边缘计算任务分配方法，包括以下步骤：

[0078] S1：对边缘计算任务分配问题进行建模并构建边缘计算任务分配系统；

[0079] S2:构建风险感知的边缘计算任务分配策略,根据风险感知的边缘计算任务分配策略得到中断风险下计算任务平均完成时间的最小值;

[0080] S3:以中断风险下的计算任务平均完成时间的最小值对应的任务分配方式进行计算任务的分配。

[0081] 图2示出了基于边缘设备组成的边缘云。

[0082] 本方案中,所述对边缘计算任务分配问题建模包括:构建边缘云、定义边缘服务、定义任务分配决策,

[0083] 其中边缘云包括:控制器、若干边缘计算节点,所述控制器与若干边缘计算节点均通过网络连接,所述控制器接收用户的任务并分配至合适的边缘计算节点,每个边缘计算节点负责执行任务并返回结果至控制器,需要说明的是,移动设备由于动态性可以不定期加入或退出边缘云,边缘服务器在边缘云中存留更持久和稳定,本发明使用存留时间指代边缘计算节点从加入至退出边缘云两时刻之间的时间段,所述退出广泛地指代由于边缘计算节点主动退出、关机或断开网络等原因导致的与边缘云控制器交互断开的事件。本发明并不特别区分某个边缘计算节点是服务器还是移动设备,两者仅在于计算能力和在边缘云中的存留时间有差别,但本质都是提供计算资源的计算节点。对于长期稳定的服务器,其存留时间可认为是无限大的。

[0084] 其中,边缘计算节点集合记为 EN , $EN = \{en_1, en_2, \dots, en_J\}$, EN 中有 J 个边缘计算节点,每个边缘计算节点与控制器的通信带宽表示为 tx_j , $1 \leq j \leq J$,设在每个边缘计算节点内部,维持若干个工作线程用以执行任务,并维持一个等待队列用以缓存未分配执行的任务,将边缘计算节点 j , $1 \leq j \leq J$ 内部的等待队列长度表示为 $q1_j$,将每个边缘计算节点内部平均任务执行时间表示为 T_j^{Avg} ;

[0085] 边缘服务:所述边缘服务由用户发起,每个边缘服务接收一个任务和该任务对应的数据文件,通过执行任务返回结果,在每一轮任务调度中,将任务集记为 $TK = \{tk_1, tk_2, \dots, tk_I\}$,其中 I 代表任务集中的任务总数,对应任务 i , $1 \leq i \leq I$,将该任务对应的数据文件大小记为 s_i ,由于在边缘云中的各边缘计算节点并非同构的,可能具有不同的软硬件配置,因此在同一任务在不同边缘计算节点中的执行时间也不相同。定义 t_{ij}^{exe} 来表示任务 i 在边缘计算节点 j 的执行时间,其中 $1 \leq i \leq I$, $1 \leq j \leq J$;

[0086] 任务分配决策:对于一个任务集合和一个边缘计算节点集合,任务分配决策定义了为每一个任务选择的边缘计算节点,使用矩阵表示为 $X = \{x_{1,1}, x_{1,2}, \dots, x_{1,I}, \dots, x_{i,j}, \dots, x_{I,1}, \dots, x_{I,J}\}$, $1 \leq i \leq I$, $1 \leq j \leq J$,其中 $x_{i,j} = 1$ 表示将任务 i 分配至边缘计算节点 j , $x_{i,j} = 0$ 则无此分配关系。

[0087] 如图3所示,本方案中,所述控制器和边缘计算节点构建了边缘计算任务分配系统,所述控制器包括有:任务提交接口、第一任务管理器、调度器、任务跟踪器、预测器、节点管理器,

[0088] 所述任务提交接口响应用户的服务请求,并接收任务和其他对应数据文件;

[0089] 所述第一任务管理器使用等待队列管理新到达的计算任务和需要重新调度的计算任务并在每个计算任务分配周期,将所有等待计算任务提交至调度器;

[0090] 节点管理器模块用于维持与边缘计算节点的组件的通信连接,并收集每个边缘节

点的运行信息(如等待队列长度);若节点管理器模块发现某边缘计算节点心跳超时,则将该边缘计算节点的已分配任务重新加入任务管理器,并删除该边缘计算节点信息,同时将该边缘计算节点的存留时间信息告知预测器中对应的节点预测单元。

[0091] 预测器用于产生计算任务被分配至边缘计算节点的完整时间的预测值和计算任务中断风险下的期望完成时间的预测值,所述预测器包括一个任务单元和一个节点预测单元,每个任务预测单元为一组具有类似硬件配置的边缘计算节点维护已执行任务的时间分布信息并提供执行时间的预估值,使用机器类型来标记同一组边缘节点;

[0092] 所述调度器在每个调度周期开始时,从预测器收集各计算任务-边缘计算节点对的完成时间预测值,并作出调度决策;

[0093] 所述任务跟踪器将各计算任务分配至对应的边缘计算节点,并监控各计算任务的运行情况;当发生任务中断时,将该任务重新加入调度队列中。当任务完成执行并从边缘节点返回结果时,将该任务的执行时间告知预测器中对应的任务预测单元。

[0094] 所述边缘计算节点包含心跳线程、第二任务管理器和若干工作线程,所述第二任务管理器接收来自控制器的计算任务,并加入等待队列中;

[0095] 所述若干工作线程中若有空闲的工作线程,则空闲的工作线程从等待队列中选择一个计算任务并执行;

[0096] 心跳线程负责定时发送本节点的运行信息(包括等待队列长度)至控制器。

[0097] 本方案中,风险感知的边缘计算任务分配策略基于上述边缘计算任务分配系统进行实施,在每个调度周期内,控制器内部的调度器作出任务分配决策。此后,每个任务的完整运行过程包含若干部分。首先,任务及其数据文件被传输至对应的边缘计算节点,并加入该节点内部的等待队列中。其次,如果边缘计算节点内部的没有空闲的工作线程,则任务需要在等待队列中等待直至被分配至某个空闲线程。再次,任务被某个空闲线程调用并执行。最后,当任务执行完毕,返回结果至控制器。由于任务的返回结果通常数据量较小,其占用时间相对其他过程较短,因此省略考虑此过程。

[0098] 所述风险感知的边缘计算任务分配策略构建过程包括:

[0099] S201构建计算任务分配至边缘计算节点的完整时间的表达式;

[0100] S202以最小化计算任务的平均完成时间为目标建立目标函数和约束条件;

[0101] S203量化潜在的计算任务中断对任务完成时间带来的额外开销时间,将额外开销时间加上无风险时计算完成时间作为中断风险下的完成时间对步骤S202中的目标函数进行更新;

[0102] S204求解更新后的目标函数的最优解。

[0103] 本方案中,设没有任务中断的情况发生,即在不考虑任务的重新调度和执行的情况下,计算任务被分配至边缘计算节点的完整时间记为 t_{ij}^{cpt} ,

$$[0104] \quad t_{ij}^{cpt} = t_{ij}^{tr} + t_{ij}^{wt} + t_{ij}^{ex} \quad (1)$$

[0105] 每个任务的完整运行包含若干过程,分别记为:第一过程、第二过程、第三过程,所述第一过程为任务及其数据文件被传输至对应的边缘计算节点,并加入该节点内部的等待队列中;所述第二过程为如果边缘计算节点内部的没有空闲的工作线程,则任务需要在等待队列中等待直至被分配至某个空闲线程;所述第三个过程为任务被某个空闲线程调用执

行；

[0106] 其中，计算任务在第一个过程的传输时间记为 t_{ij}^{tr} ，其计算公式为：

$$[0107] \quad t_{ij}^{tr} = \frac{s_i}{tx_j} \quad (2)$$

[0108] 计算任务在第二个过程的等待时间记为 t_{ij}^{wt} ，其计算公式为：

$$[0109] \quad t_{ij}^{wt} = ql_j \times T_j^{Avg} \quad (3)$$

[0110] 计算任务在第三个过程的执行时间即 t_{ij}^{ex} 。

[0111] 本方案中，在不考虑任务执行风险的情况下，以最小化计算任务的平均完成时间为目标建立目标函数和约束条件，所述目标函数表达式为：

$$[0112] \quad \min \sum_{ij} (x_{ij} \times t_{ij}^{cpt}) + \left(\sum_i x_{ij} - 1 \right) \times T_j^{Avg},$$

[0113] 约束条件为：

$$[0114] \quad \text{s. t. } \sum_i x_{i,j} = 1,$$

$$[0115] \quad x_{i,j} \in \{0,1\}, 1 \leq i \leq I, 1 \leq j \leq J \quad (4)$$

[0116] 其中，目标函数中将计算任务分配问题转化为了线性规划问题，规划问题的代价为批量任务完成时间之和， $(\sum_i x_{ij} - 1) \times T_j^{Avg}$ 为补偿项，所述补偿项表示将多个任务分配至同一个节点时的各任务的额外等待时间之和。

[0117] 需要说明的是，在约束条件中， $\sum_i x_{i,j} = 1$ 确保每个任务只能被调度至一个边缘计算节点和执行一次。在最小化该目标函数的代价的过程，求解算法将尽量避免将多个任务分配至同一个边缘计算节点，以获得最小的代价。

[0118] 本方案中，在上述线性规划问题中加入对任务执行风险的考虑，使用概率分布的观点以量化潜在的任务中断对任务完成时间带来的额外开销时间。量化潜在的计算任务中断对任务完成时间带来的额外开销时间，将额外开销时间加上无风险时计算完成时间作为中断风险下的完成时间对步骤S202中的目标函数进行优化，具体包括：

[0119] 将边缘节点的预计留存时间和边缘节点在边缘云中已读过的时间两者差值定义为边缘节点的剩余存留时间，当将一个计算任务分配至一个边缘计算节点，设该边缘计算节点的剩余存留时间为一个随机变量RPT，其概率分布律表示为：

$$[0120] \quad p_m = P(PRT = rpt_m), m = 1, 2, \dots, M, \quad (5)$$

[0121] 其中 rpt_m 代表PRT的所有可能取值，M代表这些取值的数量；设任务的完成时间为一个随机变量CT，其概率分布律表示为：

$$[0122] \quad q_n = P(CT = ct_n), n = 1, 2, \dots, N, \quad (6)$$

[0123] 其中 ct_n 代表CT的所有可能取值，N代表所有可能取值的数量；

[0124] 由于每次任务分配决策是相互独立的，故在量化单次任务中断的时间开销时假设任务的第二次执行必然完成，又假设计算任务在重新调度时也被分配至一个软硬件配置相似的边缘计算节点，也即第二次正常运行时间与第一次正常运行时间相同，则定义在计算任务中断风险下的期望完成时间为随机变量TCT，表达式如下：

$$[0125] \quad TCT = \begin{cases} CT, & RPT > CT, \\ RPT + CT, & RPT \leq CT. \end{cases} \quad (7)$$

[0126] 随机变量TCT的数学期望表达式为:

$$[0127] \quad \begin{aligned} E(TCT) &= \sum_{rpt_m > ct_n} (p_m q_n ct_n) + \sum_{rpt_m \leq ct_n} [p_m q_n (rpt_m + ct_n)] \\ &= \sum (p_m q_n ct_n) + \sum_{rpt_m \leq ct_n} p_m q_n rpt_m \\ &= E(CT) + \sum_{rpt_m \leq ct_n} p_m q_n rpt_m, \end{aligned} \quad (8)$$

[0128] 由公式(8)知,引入计算任务中断风险后,任务的期望完成时间从E(CT)变成了E(TCT),后者比前者多了一项,也即是潜在计算任务中断带来的额外开销时间的期望值,故定义潜在计算任务中断带来的额外开销时间为ET,其期望值如公式(9)所示:

$$[0129] \quad E(ET) = \sum_{rpt_m \leq ct_n} p_m q_n rpt_m, \quad (9)$$

[0130] 定义 t_{ij}^{et} 为ET的预估值,定义在计算任务中断风险下的期望完成时间为 t_{ij}^{ect} ,其值等于正常完成时间 t_{ij}^{cpt} 与 t_{ij}^{et} 之和,如公式(10)所示:

$$[0131] \quad t_{ij}^{ect} = t_{ij}^{cpt} + t_{ij}^{et} \quad (10)$$

[0132] 将 t_{ij}^{ect} 取代 t_{ij}^{cpt} ,建立风险感知的任务分配策略,将线性规划问题更新如公式(11)所示:

$$[0133] \quad \min \sum_{ij} (x_{ij} \times t_{ij}^{ept}) + \left(\sum_i x_{ij} - 1 \right) \times T_j^{Avg},$$

$$[0134] \quad s.t. \sum_i x_{ij} = 1,$$

$$[0135] \quad x_{i,j} \in \{0,1\}, 1 \leq i \leq I, 1 \leq j \leq J, \quad (11)$$

[0136] 需要说明的是,在本发明中,求解线性规划问题优化过程中,为了获取最小代价,一方面从总体上最小化任务的期望平均完成时间,另一方面需要为每一个任务最小化额外开销时间。从任务分配过程看,调度器将为每个任务尽量分配计算力强且风险小的边缘计算节点。

[0137] 本方案中,求解更新后目标函数的最优解是采用遗传算法在设定的迭代次数内获得近似最优解,在编码过程中,对于任务集 $TK = \{tk_1, tk_2, \dots, tk_I\}$ 和边缘计算节点集 $EN = \{en_1, en_2, \dots, en_J\}$,一种任务分配决策 $X = \{x_{1,1}, x_{1,2}, \dots, x_{1,I}, \dots, x_{i,j}, \dots, x_{I,1}, \dots, x_{I,J}\}$ 被编码为一个向量 $S = \{s_i = en_j, 1 \leq i \leq I, 1 \leq j \leq J\}$,代表一条染色体;其中, $s_i = en_j$ 表示将任务i调度至边缘计算节点j,

[0138] 本方案中,遗传算法中适应度函数表达式为:

$$[0139] \quad \text{Fitness}(TK, EN, X) = \frac{1}{C(X)} \quad (12)$$

$$[0140] \quad C(X) = \sum_{i,j} (x_{i,j} \times t_{i,j}^{ept}) + (\sum_i x_{i,j} - 1) \times T_j^{Avg} \quad (13)。$$

[0141] 本方案中,预测器采用黑盒方法来为每个计算任务产生概率分布和预测值。

[0142] 下面详细阐述黑盒方法产生概率分布和预测值的过程:

[0143] 所述黑盒方法假设大部分任务与先前所有任务的某个子集类似,且类似任务有相似的执行时间。所述黑盒方法不需要获取计算任务的结构信息或用户预先提供的信息,但需要每个计算任务的属性集合。本文中选用三个属性:提交任务的用户名、任务名和任务逻辑名(如图像识别或矩阵乘法)。对于每个属性-值对,所述黑盒方法通过直方图跟踪所有具有相同属性和值的任务的执行时间并生成一个经验分布。每个属性-值对应的直方图将使用四种点估计技术进行预估:平均值、中值、衰减率为0.5的移动平均值以及最近20个任务的平均值。同时在每次预估时维护四个点估计技术的归一化平均绝对误差(normalized mean absolute error, NMAE)。当预估一个带有若干属性的任务的执行时间时,从所有属性-值对应的直方图中比较所有点估计技术的NMAE,选择最低NMAE对应的点估计值和执行时间分布。此外,本发明实施例中新增了一个直方图以跟踪所有任务的执行时间,用于预估平均任务执行时间。根据以上原理,每个任务预测单元为一组边缘计算节点维护其任务执行时间的分布。

[0144] 每个节点预测单元为一组边缘计算节点维护其存留时间的分布,但由于边缘计算节点没有类似属性,因此并不用区分多种属性维护存留时间的分布。

[0145] 本发明采用基于流的直方图算法以构建直方图。一个直方图是一个键值对的集合,作为一个实数集合的近似代表,表示为 $\{(v_1, f_1), \dots, (v_B, f_B)\}$ 。对于每对 (v_i, f_i) , $1 \leq i \leq B$, v_i 表示一个数字的值,而 f_i 该数字的频数。本发明定义一个称作概率直方图的新概念,表示为 $\{(v_1, p_1), \dots, (v_B, p_B)\}$ 。对于每对 (v_i, p_i) , $1 \leq i \leq B$, v_i 表示一个数字的值,而 p_i 该数字的概率。因此,概率直方图是一个概率分布的近似代表。本发明新增两个算法用以将直方图转换为概率直方图,如算法1和算法2所示。

算法1 右移和概率转换过程

输入: 一个直方图 $\{(v_1, f_1), \dots, (v_B, f_B)\}$, 一个位移值 δ

输出: 一个概率直方图 $\{(u_1, p_1), \dots, (u_B, p_B)\}$

1: set $S = \sum_{i=1}^B f_i$

2: for $i = 1$ to B do

[0146]

3: $u_i = v_i + \delta$

4: $p_i = f_i \div \delta$

5: end for

算法 2 从给定值开始概率转换过程

输入：一个直方图 $\{(v_1, f_1), \dots, (v_B, f_B)\}$ ，一个位移值 δ

输出：一个概率直方图 $\{(u_1, p_1), \dots, (u_c, p_c)\}$

1: 查找 i 以满足 $f_i \leq \delta \leq f_{i+1}$

2: if $f_i = \delta$ then

3: set $S = \sum_{j=i}^B f_j$

[0147] 4: else

5: set $S = \sum_{j=i+1}^B f_j, i = i + 1$

6: end if

2: for $j = i$ to $B, k = 1$ to $B - i + 1$ do

3: $u_k = v_j$

4: $p_k = f_k \div \delta$

5: end for

[0148] 算法1首先根据一个位移值对一个直方图进行右移操作，然后将给直方图转换成概率直方图。算法2首先在某个直方图中根据某个给定值，过滤掉小于该值的点，然后将剩余的直方图转换为概率直方图。

[0149] 本实施例所述任务调度策略中各种时间值的概率分布的关系如图4所示。预测器从任务预测单元获取任务的执行时间的概率分布，并使用公式(2)和(3)计算任务的传输时间和等待时间后，使用算法1可以获得任务完成时间的概率分布和预测值。类似地，从节点预测单元获取存留时间的概率分布，在计算该节点的已存留时间后，使用算法2可以获得节点剩余存留时间的概率分布。进而根据公式 $E(ET) = \sum_{rpt_m \leq ct_n} p_m p_n rpt_m$ 计算额外开销时间，根据公式(1)和公式(10)计算任务在中断风险下的期望完成时间。

[0150] 在可扩展性方面，本发明实施例中主要使用三种设计以减少内存占用。首先，黑盒方法和直方图算法本身的设计能够使用固定大小的内存来维护一个任意键值对的直方图。其次，本发明每个任务预测单元不必仅为单个边缘计算节点维护一个执行时间直方图，可共同为一群软硬件配置相似的边缘计算节点服务，比如按照CPU计算能力分类，也可简单地分为手机、移动电脑和边缘服务器等三类。最后，类似地，本文发明每个节点预测单元为一群网络环境相似(比如在同一无线接入点下)的边缘节点维护一个存留时间直方图，而不仅是单个边缘计算节点。

[0151] 显然，本发明的上述实施例仅仅是为清楚地说明本发明所作的举例，而并非是对本发明的实施方式的限定。对于所属领域的普通技术人员来说，在上述说明的基础上还可以做出其它不同形式的变化或变动。这里无需也无法对所有的实施方式予以穷举。凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等，均应包含在本发明权利要求的保护范围之内。

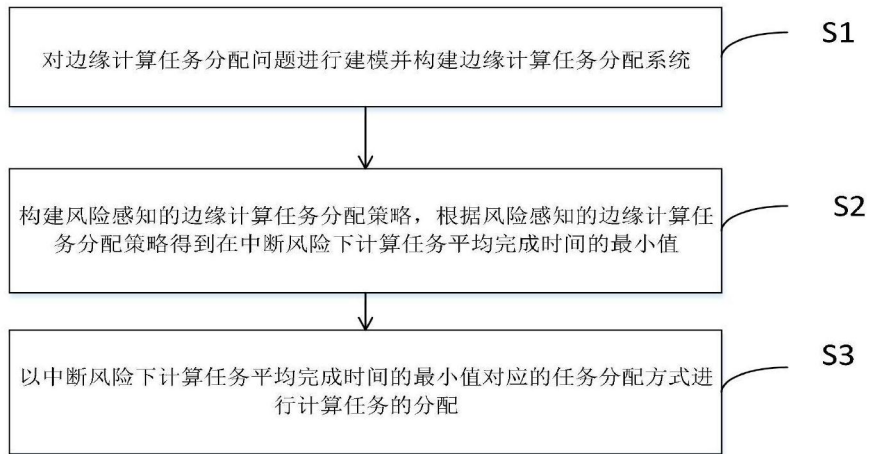


图1

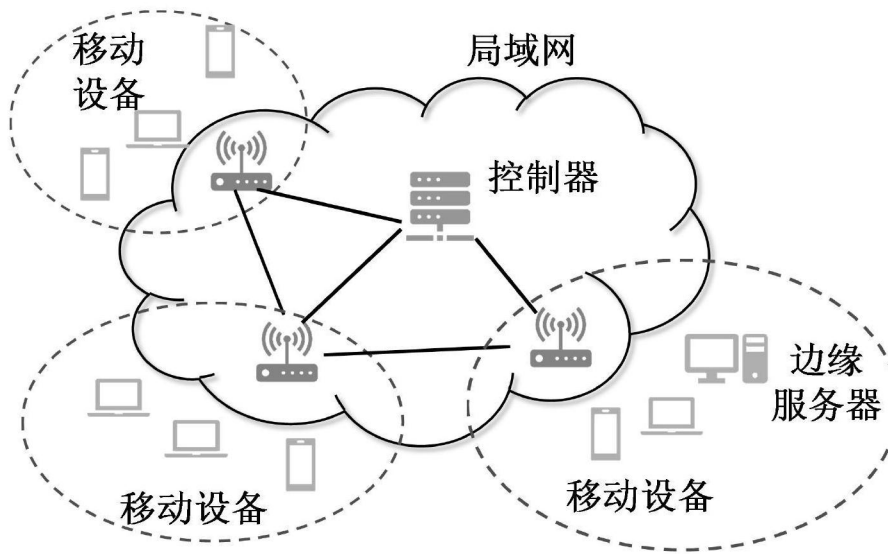


图2

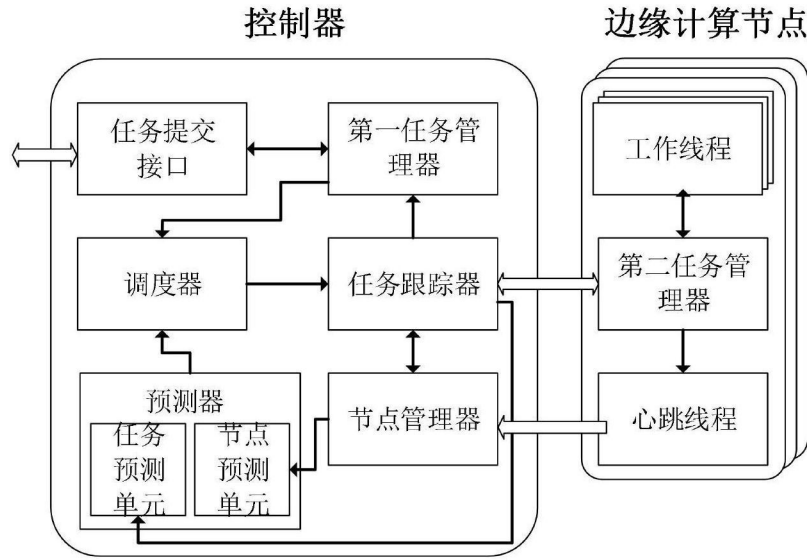


图3

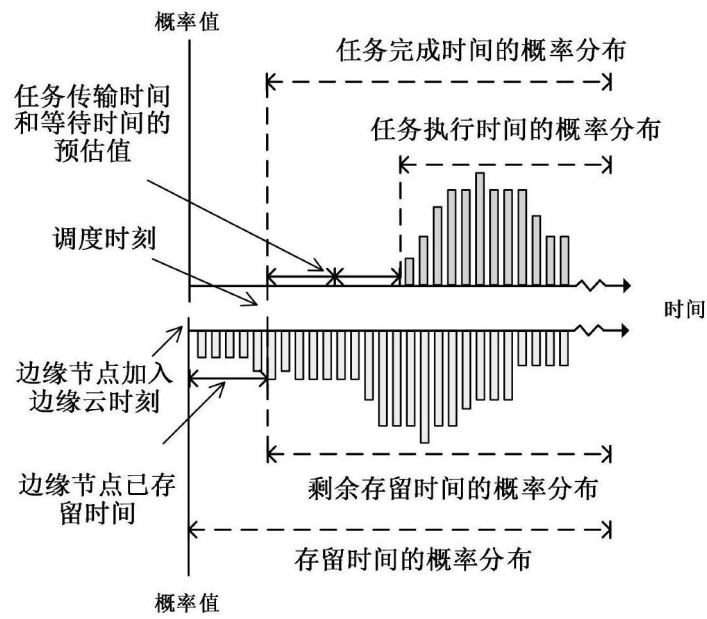


图4