



(12) 发明专利申请

(10) 申请公布号 CN 117632298 A

(43) 申请公布日 2024. 03. 01

(21) 申请号 202311663171.8

(22) 申请日 2023.12.06

(71) 申请人 江西理工大学

地址 341099 江西省赣州市红旗大道86号

(72) 发明人 陈益杉 罗显淞 王碧 李伟

徐中辉

(74) 专利代理机构 北京盛询知识产权代理有限公司

11901

专利代理师 吴立强

(51) Int. Cl.

G06F 9/445 (2018.01)

G06F 9/48 (2006.01)

G06F 9/50 (2006.01)

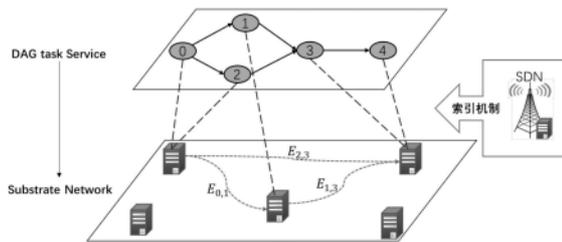
权利要求书2页 说明书7页 附图7页

(54) 发明名称

一种基于优先级列表索引机制的任务卸载及资源分配方法

(57) 摘要

本发明公开了一种基于优先级列表索引机制的任务卸载及资源分配方法,包括:通过监测器收集指定区域内边缘服务器的资源信息,根据本地设备的任务请求构建DAG模型,生成优先级索引列表;基于所述优先级索引列表,采用基于优先级的贪婪算法和资源分配算法进行目标优化,获得关联型任务卸载策略及资源分配方法;根据所述关联型任务卸载策略及资源分配方法执行用户任务卸载操作。本发明通过对任务卸载和资源分配进行预测和优化,有效地降低了总体延迟,提高了资源利用率,实现了对关联型任务的高效执行和资源的合理分配。



1. 一种基于优先级列表索引机制的任务卸载及资源分配方法,其特征在于,包括:
收集指定区域内边缘服务器的资源信息,根据本地设备的任务请求构建DAG模型,生成优先级索引列表;

基于所述优先级索引列表,采用基于优先级的贪婪算法和资源分配算法进行目标优化,获得关联型任务卸载策略及资源分配方法;

根据所述关联型任务卸载策略及资源分配方法执行用户任务卸载操作。

2. 根据权利要求1所述的基于优先级列表索引机制的任务卸载及资源分配方法,其特征在于,根据所述本地设备的任务请求构建DAG模型,生成优先级索引列表,包括:

分别设置DAG中每个子任务的优先级以及每个DAG的优先级,将DAG中每个子任务进行排序,获得所述DAG的子任务优先级列表,再将各DAG的所述子任务优先级列表进行排序,得到DAG优先级列表,即所述优先级索引列表。

3. 根据权利要求2所述的基于优先级列表索引机制的任务卸载及资源分配方法,其特征在于,定义所述DAG中每个子任务的优先级的方法为:

$$\text{rank}(t_i^x) = \frac{W(t_i^x)}{\bar{f}_x} + \max_{t_j^x \in \text{succ}(t_i^x)} \left\{ \frac{d(e_{i,j}^x)}{\bar{v}} + \text{rank}(t_j^x) \right\}$$

式中, t_i^x 为本地设备 I_x 产生的子任务, i 为子任务的序号, x 为本地设备的序号, $\text{rank}(t_i^x)$ 表示 t_i^x 的优先级, $W(t_i^x)$ 表示 t_i^x 的计算量大小, \bar{f}_x 为本地设备 I_x 与系统中边缘服务器的综合计算能力, $\text{succ}(t_i^x)$ 表示需要 t_i^x 发送依赖数据的子任务的集合, t_j^x 为 $\text{succ}(t_i^x)$ 中的一个子任务, $e_{i,j}^x$ 表示 t_i^x 向 t_j^x 发送的依赖数据, $d(e_{i,j}^x)$ 为 $e_{i,j}^x$ 的数据量大小, \bar{v} 为依赖数据在系统中的平均传输速率。

4. 根据权利要求3所述的基于优先级列表索引机制的任务卸载及资源分配方法,其特征在于,定义每个DAG的优先级的方法为:

$$\psi(x) = \text{cost}(G_x) + \frac{\text{rank}(t_0^x)^2}{\text{cost}(G_x)}$$

式中, G_x 为本地设备 I_x 产生的DAG任务, t_0^x 为 G_x 的开始子任务, $\text{cost}(G_x)$ 为 G_x 的系统代价, $\psi(x)$ 为 G_x 的优先级。

5. 根据权利要求1所述的基于优先级列表索引机制的任务卸载及资源分配方法,其特征在于,获得所述关联型任务卸载策略及资源分配方法,包括:

根据所述优先级索引列表,通过贪婪算法寻找使每个所述子任务具有最小的最早完成时间的卸载决策,获得每个边缘服务器的DAG卸载记录;

按所述优先级索引列表索引取出每个子任务,选择使得该子任务有最早完成时间的设备进行卸载,更新子任务卸载策略和边缘服务器的DAG卸载记录,按照所述边缘服务器的DAG卸载记录计算DAG在边缘服务器的计算资源占比,获得所述关联型任务卸载策略及资源分配方法。

6. 根据权利要求5所述的基于优先级列表索引机制的任务卸载及资源分配方法,其特

征在于,按照所述边缘服务器的DAG卸载记录计算DAG在边缘服务器的计算资源占比的方法为:

$$\gamma_k^x = \frac{\psi(x)}{\sum_{n=1}^N R_k^n \psi(n)}$$

式中, γ_k^x 为资源分配比例, R_k^n 为记录本地设备产生的DAG任务是否卸载子任务到边缘服务器的二进制变量, $\psi(n)$ 为DAG的优先级,N为DAG任务的总数。

7.根据权利要求1所述的基于优先级列表索引机制的任务卸载及资源分配方法,其特征在于,所述关联型任务卸载策略及资源分配方法的目标函数为系统完成所有DAG任务的平均响应时间:

$$\min \frac{1}{N} \sum_{x=1}^N ft(t_{u_{x+1}}^x)$$

式中,ft为子任务的完成时间, $t_{u_{x+1}}^x$ 为 G_x 的结束任务, G_x 为本地设备 I_x 产生的DAG任务,N为DAG任务的总数。

8.基于优先级列表索引机制的任务卸载及资源分配系统,其特征在于,包括:

系统监控单元:用于监测来自设备的任务信息和系统的资源信息;

卸载规划单元:根据所述任务信息和所述资源信息,通过DAG优先级列表索引机制规划任务卸载策略及资源分配方法;

策略实施单元:根据已拟定的所述任务卸载策略及资源分配方法执行用户任务卸载操作。

一种基于优先级列表索引机制的任务卸载及资源分配方法

技术领域

[0001] 本发明涉及边缘计算环境下DAG任务的卸载及资源分配技术领域,尤其涉及一种基于优先级列表索引机制的任务卸载及资源分配方法。

背景技术

[0002] 边缘计算将计算和数据处理分散到多个边缘服务器上,实现更加高效和快速的数据处理和分发,减少了对云计算中心的依赖,降低了数据传输的延迟和带宽消耗。在这种环境下,任务需要被卸载到多个边缘服务器上进行处理,以减少延迟和带宽消耗,提高服务质量。

[0003] 然而,在多用户多边缘服务器环境下,如何对关联型任务进行合理的卸载和资源分配仍然是一个具有挑战性的问题。本地设备产生的关联型任务通常被建模为DAG任务,一个DAG任务可能由多个子任务组成,这些子任务之间存在先后顺序和依赖关系,因此需要一个能够合理分配资源和协调任务之间执行顺序的策略。同时,由于边缘服务器之间的性能和资源利用率存在差异,因此需要一个能够动态调整资源分配和任务卸载的方案。

[0004] 在实际应用中,资源受限是一个重要的问题。由于边缘服务器的计算能力有限,因此需要一个能够优化资源利用率和任务执行效率的方案。一方面,需要尽可能地利用边缘服务器的计算资源,以提高整个系统的处理能力和效率。另一方面,需要避免资源浪费和任务阻塞等问题,以保证系统的稳定性和可靠性。

[0005] 为了解决这些问题,可以考虑引入优先级机制来合理的制定任务卸载方案和资源分配方案。

[0006] 任务卸载中的优先级机制主要通过为不同的任务划分不同的优先级,优先级机制可以在保证任务顺序和执行效率的前提下,合理的使级别优先的任务能够优先抢占更好的卸载位置,从而提高系统的整体性能。同时,为了更加准确地进行资源分配,还需要设计一种有效的资源分配方法。这种资源分配方法需要能够考虑到DAG任务整体的计算量、各设备的计算资源情况、边缘服务器的资源利用率、任务之间的依赖关系等多个因素,从而在任务卸载和资源分配方面做出更加合理的决策,是一项实际且具有挑战性的工作。

发明内容

[0007] 本发明针对边缘计算环境下DAG任务的卸载及资源分配问题,提供了一种基于优先级列表索引机制的任务卸载及资源分配方法。

[0008] 一方面,为实现上述目的,本发明提供了一种基于优先级列表索引机制的任务卸载及资源分配方法,包括:

[0009] 收集指定区域内边缘服务器的资源信息,根据本地设备的任务请求构建DAG模型,生成优先级索引列表;

[0010] 基于所述优先级索引列表,采用基于优先级的贪婪算法和资源分配算法进行目标优化,获得关联型任务卸载策略及资源分配方法;

[0011] 根据所述关联型任务卸载策略及资源分配方法执行用户任务卸载操作。

[0012] 优选地,根据所述本地设备的任务请求构建DAG模型,生成优先级索引列表,包括:
[0013] 分别设置DAG中每个子任务的优先级以及每个DAG的优先级,将DAG中每个子任务进行排序,获得所述DAG的子任务优先级列表,再将各DAG的所述子任务优先级列表进行排序,得到DAG优先级列表,即所述优先级索引列表。

[0014] 优选地,定义所述DAG中每个子任务的优先级的方法为:

$$[0015] \quad rank(t_i^x) = \frac{W(t_i^x)}{\bar{f}_x} + \max_{t_j^x \in succ(t_i^x)} \left\{ \frac{d(e_{i,j}^x)}{\bar{v}} + rank(t_j^x) \right\}$$

[0016] 式中, t_i^x 为本地设备 I_x 产生的子任务, i 为子任务的序号, x 为本地设备的序号, $rank(t_i^x)$ 表示 t_i^x 的优先级, $W(t_i^x)$ 表示 t_i^x 的计算量大小, \bar{f}_x 为本地设备 I_x 与系统中边缘服务器的综合计算能力, $succ(t_i^x)$ 表示需要 t_i^x 发送依赖数据的子任务的集合, t_j^x 为 $succ(t_i^x)$ 中的一个子任务, $e_{i,j}^x$ 表示 t_i^x 向 t_j^x 发送的依赖数据, $d(e_{i,j}^x)$ 为 $e_{i,j}^x$ 的数据量大小, \bar{v} 为依赖数据在系统中的平均传输速率。

[0017] 优选地,定义每个DAG的优先级的方法为:

$$[0018] \quad \psi(x) = cost(G_x) + \frac{rank(t_0^x)^2}{cost(G_x)}$$

[0019] 式中, G_x 为本地设备 I_x 产生的DAG任务, t_0^x 为 G_x 的开始子任务, $cost(G_x)$ 为 G_x 的系统代价, $\psi(x)$ 为 G_x 的优先级。

[0020] 优选地,获得所述关联型任务卸载策略及资源分配方法,包括:

[0021] 根据所述优先级索引列表,通过贪婪算法寻找使每个所述子任务具有最小的最早完成时间的卸载决策,获得每个边缘服务器的DAG卸载记录;

[0022] 按所述优先级索引列表索引取出每个子任务,选择使得该子任务有最早完成时间的设备进行卸载,更新子任务卸载策略和边缘服务器的DAG卸载记录,按照所述边缘服务器的DAG卸载记录计算DAG在边缘服务器的计算资源占比,获得所述关联型任务卸载策略及资源分配方法。

[0023] 优选地,按照所述边缘服务器的DAG卸载记录计算DAG在边缘服务器的计算资源占比的方法为:

$$[0024] \quad \gamma_k^x = \frac{\psi(x)}{\sum_{n=1}^N R_k^n \psi(n)}$$

[0025] 式中, γ_k^x 为资源分配比例, R_k^n 为记录本地设备产生的DAG任务是否卸载子任务到边缘服务器的二进制变量, $\psi(n)$ 为DAG的优先级, N 为DAG任务的总数。

[0026] 优选地,所述关联型任务卸载策略及资源分配方法的目标函数为系统完成所有DAG任务的平均响应时间:

$$[0027] \quad \min \frac{1}{N} \sum_{x=1}^N ft(t_{u_{x+1}}^x)$$

[0028] 式中, ft 为子任务的完成时间, $t_{u_{x+1}}^x$ 为 G_x 的结束任务, G_x 为本地设备 I_x 产生的 DAG 任务, N 为 DAG 任务的总数。

[0029] 另一方面, 为了实现上述目的, 本发明还提供了基于优先级列表索引机制的任务卸载及资源分配系统, 包括:

[0030] 系统监控单元: 用于监测来自设备的任务信息和系统的资源信息;

[0031] 卸载规划单元: 根据所述任务信息和所述资源信息, 通过 DAG 优先级列表索引机制规划任务卸载策略及资源分配方法;

[0032] 策略实施单元: 根据已拟定的所述任务卸载策略及资源分配方法执行用户任务卸载操作。

[0033] 与现有技术相比, 本发明具有如下优点和技术效果:

[0034] 本发明通过任务优先级和用户优先级的综合考虑, 能够更好地解决分布式计算环境下 DAG 任务的卸载问题, 让每个子任务都能获得当前资源情况下最早的结束时间, 最大化利用分布式计算环境的计算资源, 从而提高 DAG 任务的执行效率和性能; 通过对任务卸载和资源分配进行预测和优化, 有效地降低了总体延迟, 提高了资源利用率, 实现了对关联型任务的高效执行和资源的合理分配。

附图说明

[0035] 构成本申请的一部分的附图用来提供对本申请的进一步理解, 本申请的示意性实施例及其说明用于解释本申请, 并不构成对本申请的不当限定。在附图中:

[0036] 图1为本发明实施例的基于优先级列表索引机制的任务卸载及资源分配系统架构示意图;

[0037] 图2为本发明实施例的面向资源分配的 DAG 任务的卸载方法的流程示意图;

[0038] 图3为本发明实施例的任务响应时间与参数 α 的关系示意图;

[0039] 图4为本发明实施例的任务响应时间与算法的关系示意图;

[0040] 图5为本发明实施例的任务响应时间与边缘服务器个数的关系示意图;

[0041] 图6为本发明实施例的任务响应时间与本地设备个数的关系示意图;

[0042] 图7为本发明实施例的任务响应时间与最大逻辑核心数的关系示意图。

具体实施方式

[0043] 需要说明的是, 在不冲突的情况下, 本申请中的实施例及实施例中的特征可以相互组合。下面将参考附图并结合实施例来详细说明本申请。

[0044] 需要说明的是, 在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行, 并且, 虽然在流程图中示出了逻辑顺序, 但是在某些情况下, 可以以不同于此处的顺序执行所示出或描述的步骤。

[0045] 本发明提出了一种基于优先级列表索引机制的任务卸载及资源分配方法, 如图1,

具体包括：

[0046] 通常，任务卸载都会把更小的卸载时延作为优化目标，而环境中的计算资源是有限的，会出现一些重要的或计算量大的任务分配不到更多计算资源的情况，这对关联型任务尤为致命，一个任务的时延变长，往往会直接影响或间接影响后置任务的进行，这将提高系统整体时延。为了让计算量较大的任务或者位置特殊的任务享受到更优质的计算资源，它应该对更好的卸载位置进行优先抢占，为此，提出一种针对用户的基于优先级列表索引机制的任务卸载及资源分配方法。

[0047] 通过监测器收集指定区域内边缘服务器的资源信息，根据本地设备的任务请求构建DAG模型，生成优先级索引列表；

[0048] 分别定义DAG中每个子任务的优先级以及每个DAG的优先级，将DAG中每个子任务进行排序，获得所述DAG的子任务优先级列表，再将各DAG的所述子任务优先级列表进行排序，得到DAG优先级列表，即所述优先级索引列表。

[0049] DAG中每个子任务的优先级被定义为：

$$[0050] \quad \text{rank}(t_i^x) = \frac{W(t_i^x)}{\bar{f}_x} + \max_{t_j^x \in \text{succ}(t_i^x)} \left\{ \frac{d(e_{i,j}^x)}{\bar{v}} + \text{rank}(t_j^x) \right\}$$

[0051] 每个DAG的优先级被定义为：

$$[0052] \quad \psi(x) = \text{cost}(G_x) + \frac{\text{rank}(t_0^x)^2}{\text{cost}(G_x)}$$

[0053] 其中：

[0054] 1) G_x 表示本地设备 I_x 产生的DAG任务， t_i^x 为 G_x 中序号为i的子任务， t_0^x 为 G_x 的开始子任务， $\text{succ}(t_i^x)$ 表示需要 t_i^x 发送依赖数据的子任务的集合。

[0055] 2) $W(t_i^x)$ 为 t_i^x 的计算量大小， \bar{f}_x 为 I_x 与系统中边缘服务器的综合计算能力，表示如下：

$$[0056] \quad \bar{f}_x = \alpha f_x + (1 - \alpha) \frac{\sum_{k=1}^M F_k}{M}$$

[0057] 其中， α 表示权重参数， f_x 为 I_x 的计算能力， F_k 为边缘服务器 P_k 的计算能力， M 为边缘服务器总数。

[0058] 3) $e_{i,j}^x$ 表示 t_i^x 向 t_j^x 传输的依赖数据， $d(e_{i,j}^x)$ 为依赖数据的大小， \bar{v} 为依赖数据在系统中的平均传输速率。

[0059] 4) $\text{cost}(G_x)$ 为 G_x 的系统代价，表示如下：

$$[0060] \quad \text{cost}(G_x) = \sum_{t_i^x \in T_x} \frac{W(t_i^x)}{\bar{f}_x} + \sum_{e_{i,j}^x \in E_x} \frac{d(e_{i,j}^x)}{\bar{v}}$$

[0061] 其中， T_x 为 G_x 中子任务的集合， E_x 为 G_x 中依赖数据的集合。

[0062] 5) 将DAG中每个子任务按rank从大到小顺序排序，得到该DAG的子任务优先级列表

Q;再将各DAG的Q按照 ψ 从大到小的顺序排序,得到DAG优先级列表PL。

[0063] 基于优先级索引列表,采用基于优先级的贪婪算法和资源分配算法进行目标优化,获得关联型任务卸载策略及资源分配方法;

[0064] 图2给出了面向资源分配的DAG任务的卸载方法的流程示意图,在分布式计算环境中,监测器首先收集区域内边缘服务器的资源信息,根据本地设备的任务请求建立DAG模型,为这些DAG生成优先级索引列表。根据列表顺序,结合不同设备的计算资源以及DAG优先级为每个子任务选择最佳的卸载位置并分配合适的计算资源。

[0065] 在得到卸载决策的过程中,进一步构建的优化目标为系统完成所有DAG任务的平均响应时间:

$$[0066] \quad \min \frac{1}{N} \sum_{x=1}^N ft(t_{u_{x+1}}^x)$$

[0067] 其中,

[0068] 1) ft 表示子任务的完成时间, $t_{u_{x+1}}^x$ 为 G_x 的结束任务,故 $ft(t_{u_{x+1}}^x)$ 也表示 G_x 的响应时间, ft 的计算公式如下:

$$[0069] \quad ft(t_i^x) = \begin{cases} st(t_i^x) + \frac{W(t_i^x)}{f_x}, & k = 0 \\ st(t_i^x) + \frac{W(t_i^x)}{F_k \cdot \gamma_k^x}, & k = \{1, \dots, M\} \end{cases}$$

[0070] f_x 为本地设备 I_x 的计算能力, F_k 为边缘服务器 P_k 的计算能力, γ_k^x 为 G_x 在边缘服务器 P_k 上的计算资源占比。

[0071] 2) 在 P_k 接收到执行 t_i^x 所需的依赖数据并且 P_k 可用之前,不能开始执行 t_i^x 。因此, t_i^x 的开始时间表示如下:

$$[0072] \quad st(t_i^x) = \max\{ava(G_x, P_k), \max_{t_j^x \in pred(t_i^x)} \{ft(t_j^x) + ct(e_{j,i}^x)\}\}$$

[0073] 其中, $pred(t_i^x)$ 表示向 t_i^x 发送依赖数据的子任务集合, $ava(G_x, P_k)$ 表示 P_k 对 G_x 的子任务的开始可用时间,它与 P_k 处理的上一个 G_x 的子任务有关,且 $ava(G_x, P_0)$ 表示本地设备 I_x 的开始可用时间, $ct(e_{j,i}^x)$ 为依赖数据的传输时间。

[0074] 3) G_x 在边缘服务器 P_k 上的计算资源分配比例 γ_k^x 与DAG优先级有关,表示为:

$$[0075] \quad \gamma_k^x = \frac{\psi(x)}{\sum_{n=1}^N R_k^n \psi(n)}$$

[0076] 其中, R_k^n 为记录 G_n 是否卸载子任务到 P_k 的二进制变量, N 为DAG任务的总数。

[0077] 本发明针对多用户DAG任务的卸载问题,采用基于优先级的贪婪算法和资源分配算法和进行目标优化。将关联型任务卸载问题映射到DAG任务模型中,DAG任务产生之后,通过计算子任务优先级和DAG优先级得到一个DAG之间有序,每个子任务之间有序的优先级索

引列表。按列表索引顺序,通过贪心算法得到每个子任务的卸载决策。最后,通过卸载决策得到边缘服务器的DAG卸载记录,分配每个子任务的计算资源占比。整个优先级列表索引机制过程如下:

[0078] 1) 初始化

[0079] 对所有关联型任务的数据元素进行收集和处理,建模其所对应的DAG任务图,收集当前情况下边缘服务器资源信息。

[0080] 2) 索引操作

[0081] 计算每个DAG及其子任务的优先级,构建优先级列表PL。

[0082] 3) 贪婪选择

[0083] 按PL索引取出每个子任务,选择使得该子任务有最早完成时间的设备进行卸载,更新子任务卸载策略和边缘服务器DAG卸载记录。

[0084] 4) 资源分配

[0085] 按照边缘服务器DAG卸载记录计算DAG在边缘服务器的计算资源占比:

$$[0086] \quad \gamma_k^x = \frac{\psi(x)}{\sum_{n=1}^N R_k^n \psi(n)}$$

[0087] 通过上述贪婪算法和资源分配,本发明能够高效地解决多用户DAG任务的卸载问题,并实现了资源的优化利用,从而提高了边缘计算环境下的任务执行效率。

[0088] 卸载操作

[0089] 根据已拟定的卸载策略和资源分配方法执行用户任务卸载操作。

[0090] 采用本发明进行任务卸载和资源分配的过程如下:

[0091] 1) 初始化阶段:处理器接收来自各设备的任务处理请求;

[0092] 2) 任务卸载决策制定阶段:根据DAG优先级和子任务优先级构造DAG优先级列表,通过贪婪算法寻找使每个子任务具有最小的最早完成时间的卸载决策,得到每个边缘服务器的DAG卸载记录。

[0093] 3) 计算资源分配阶段:根据得到的DAG卸载记录进行边缘服务器计算资源分配。

[0094] 4) 执行阶段:根据任务卸载决策以及计算资源分配策略,进行所有DAG任务的卸载。

[0095] 本实施例还提供了一种基于优先级列表索引机制的任务卸载及资源分配系统,用于实施基于优先级列表索引机制的任务卸载及资源分配方法,包括:

[0096] 系统监控单元:用于监测来自设备的任务信息和系统的资源信息;

[0097] 卸载规划单元:根据所述任务信息和所述资源信息,通过DAG优先级列表索引机制规划任务卸载策略及资源分配方法;

[0098] 策略实施单元:根据已拟定的所述任务卸载策略及资源分配方法执行用户任务卸载操作。

[0099] 仿真模拟

[0100] 对比本发明方法(PBGTS)与差分进化算法(MDE)、异构最早完成时间算法(MHEFT)、多应用多任务调度算法(MAMTS)、粒子群算法(PSO)和遗传算法(GA)的数据结果。由图3中的对比可知, α 对系统平均响应时间的影响控制在2s以内,经过对比分析,将轻量级(Lw)DAG的 α 设置为0.9,计算密集型(Cpi)DAG的 α 设置为0.875,通信密集型(Cmi)DAG的 α 设置为0.7。由

图4中的对比可知,本发明的单个DAG的响应时间和系统平均响应时间要优于其他算法。由图5和图6中的对比可知,本发明的优势在计算环境发生变化时并不受影响,无论是改变边缘服务器的数量还是改变本地设备的数量,本发明都能得到一个较好的结果。由图7可知,最大逻辑核心的数量影响了本发明和其他算法的效果,经过对比分析,本发明将处理轻量级DAG、计算密集型DAG、通信密集型DAG的最大逻辑核心数分别设置为10、7、6能得到最好的效果。

[0101] 本发明主要面向边缘计算环境下的DAG任务的卸载决策及资源分配问题,设计了一种基于优先级的任务卸载决策系统,在降低总体延迟进行任务卸载的前提下,以每个子任务都能获得当前情况下最早开始时间为目标,在计算资源受限的分布式计算环境中找到了满意的任务卸载方案和资源分配方案,提出基于优先级列表索引机制的任务卸载及资源分配方法。

[0102] 本发明通过任务优先级和用户优先级的综合考虑,能够更好地解决分布式计算环境下DAG任务的卸载问题,让每个子任务都能获得当前资源情况下最早的结束时间,最大化利用分布式计算环境的计算资源,从而提高DAG任务的执行效率和性能;通过对任务卸载和资源分配进行预测和优化,有效地降低了总体延迟,提高了资源利用率,实现了对关联型任务的高效执行和资源的合理分配。

[0103] 以上,仅为本申请较佳的具体实施方式,但本申请的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本申请揭露的技术范围内,可轻易想到的变化或替换,都应涵盖在本申请的保护范围之内。因此,本申请的保护范围应该以权利要求的保护范围为准。

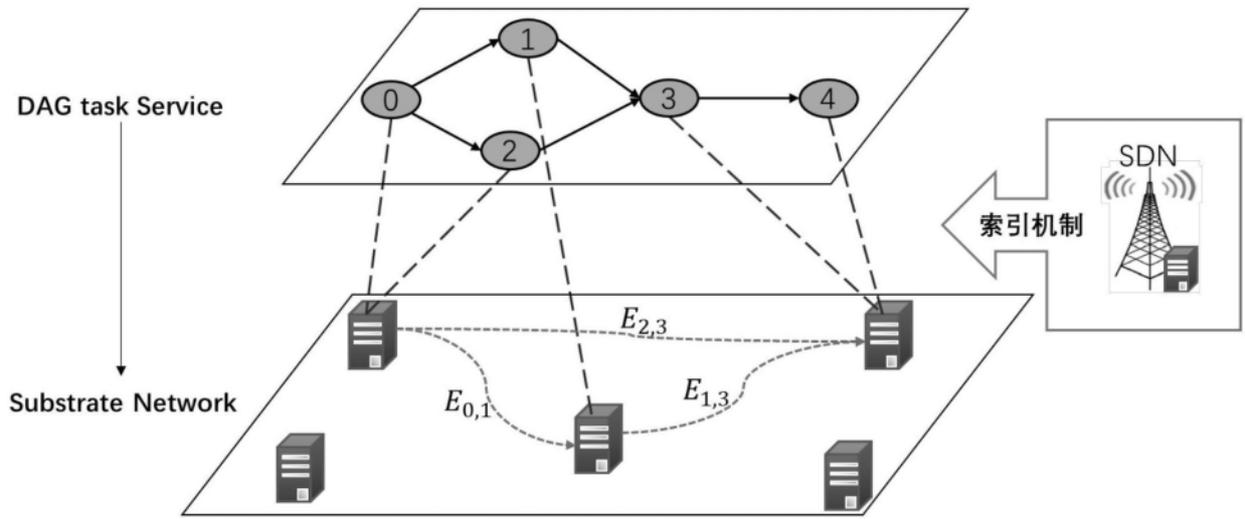


图1

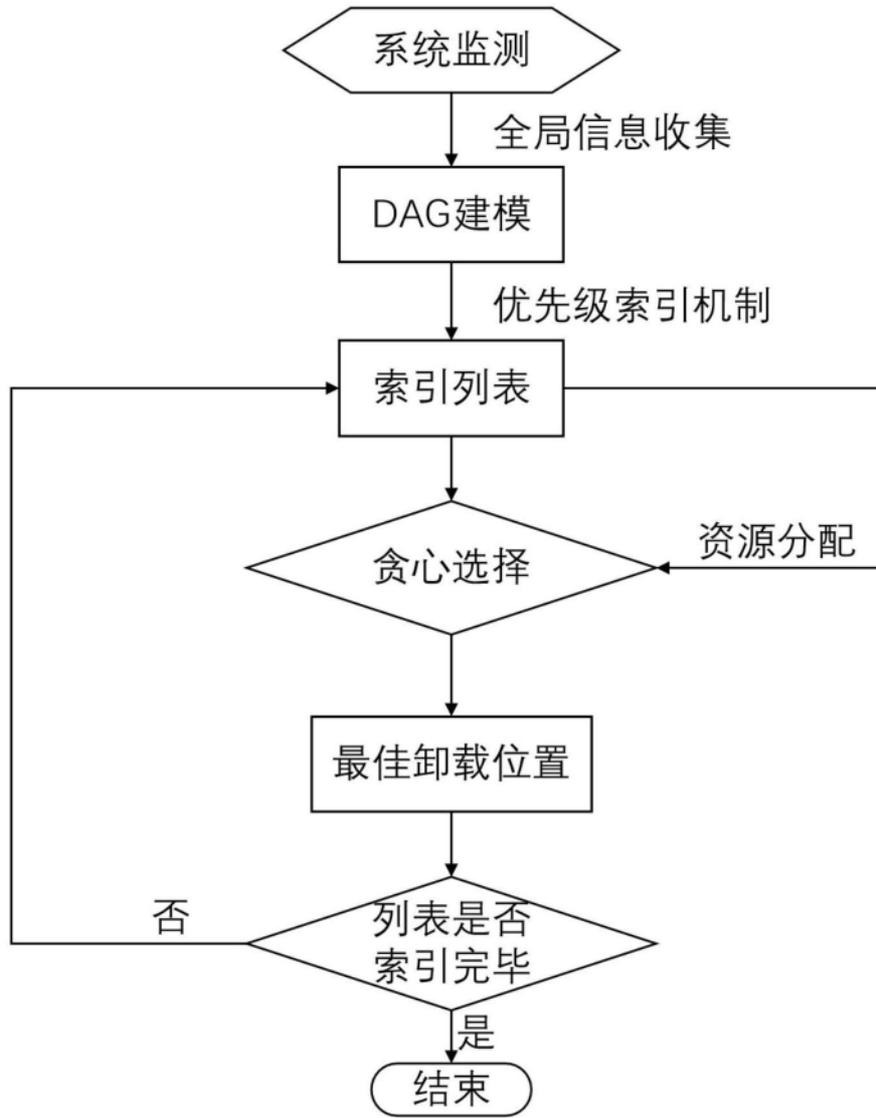


图2

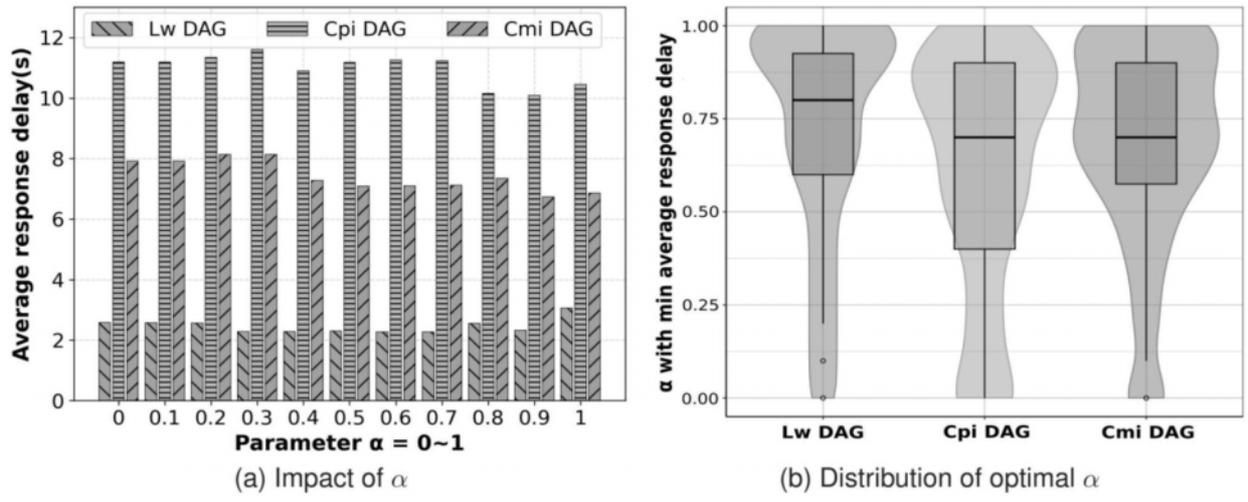


图3

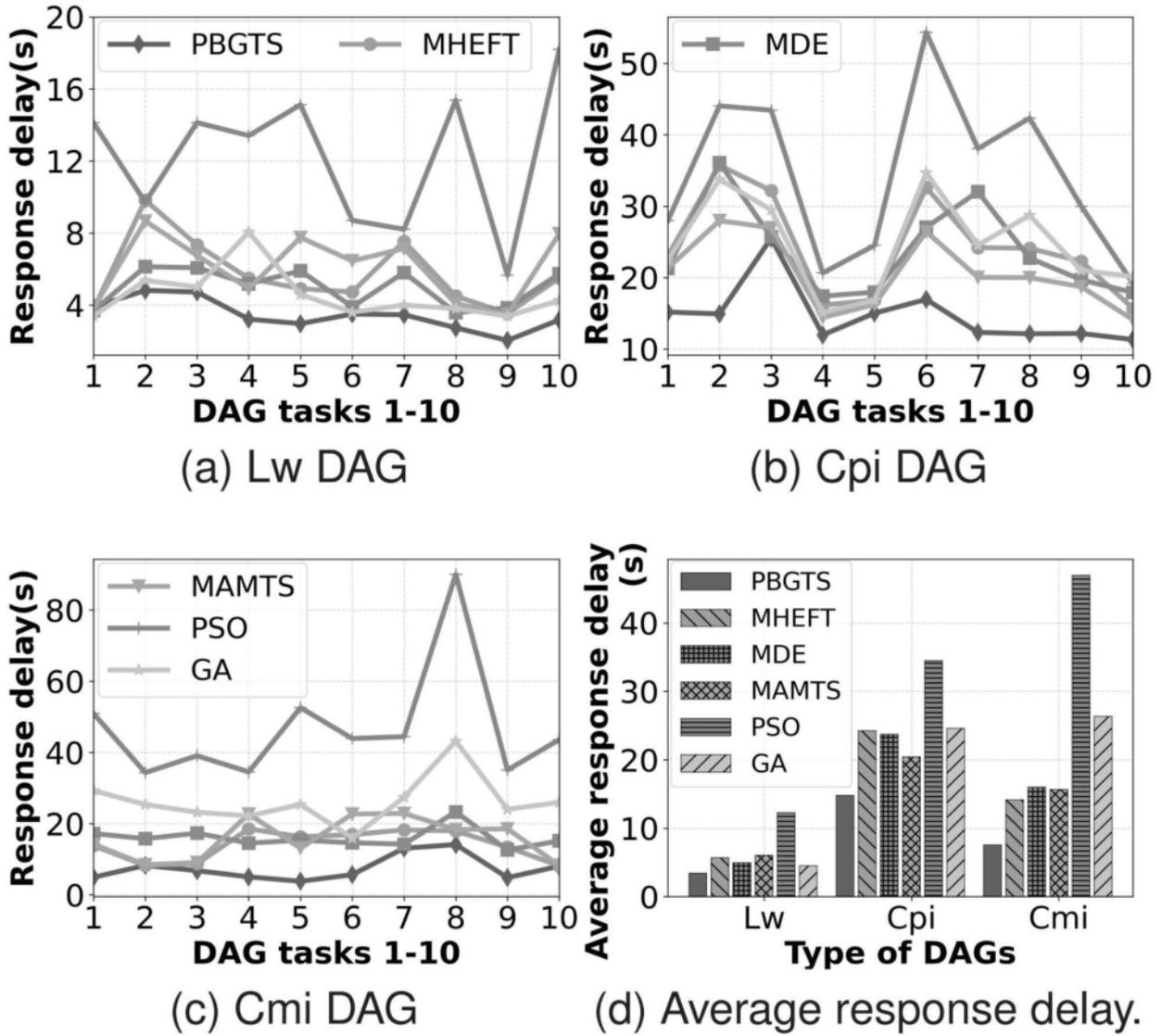


图4

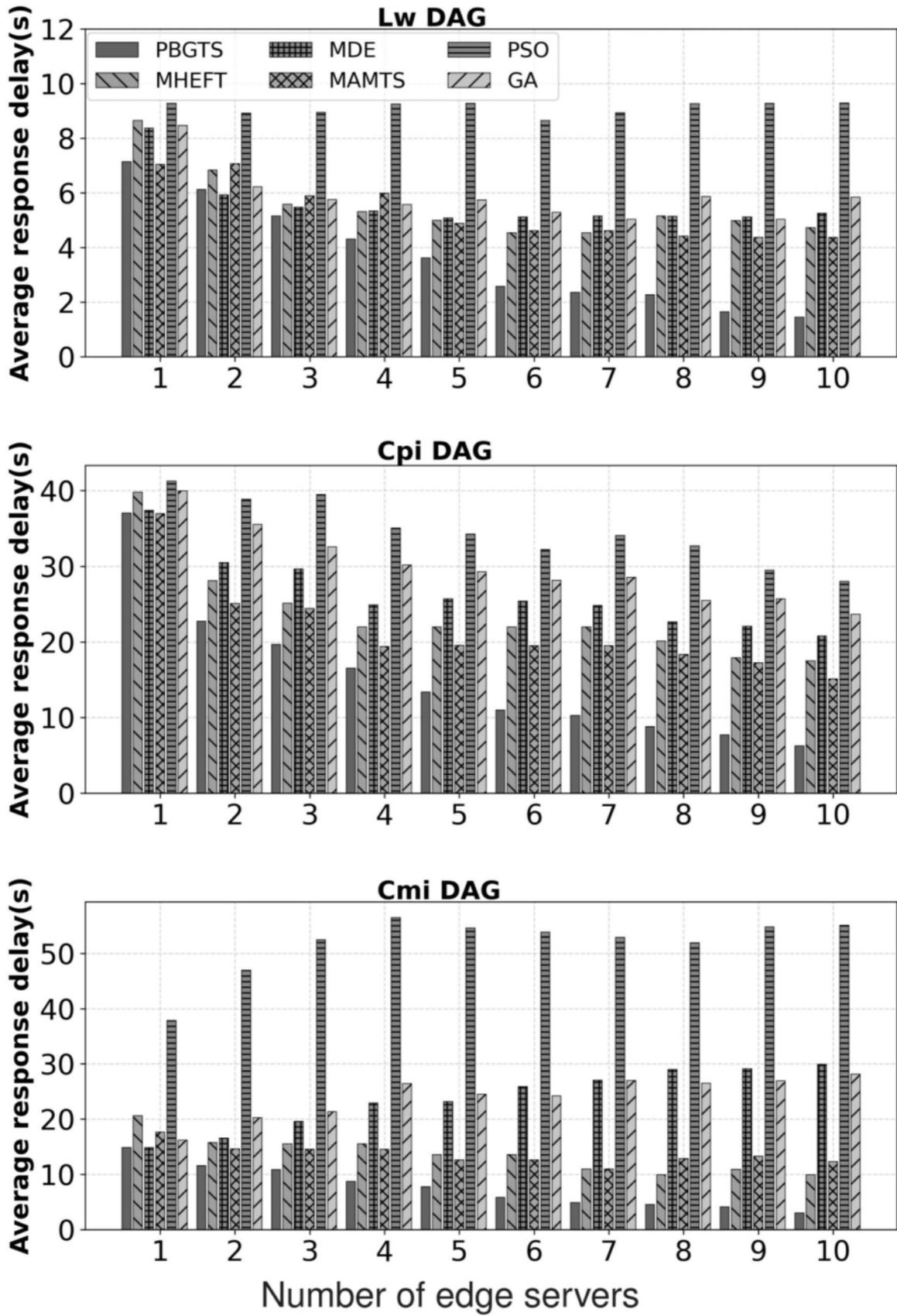


图5

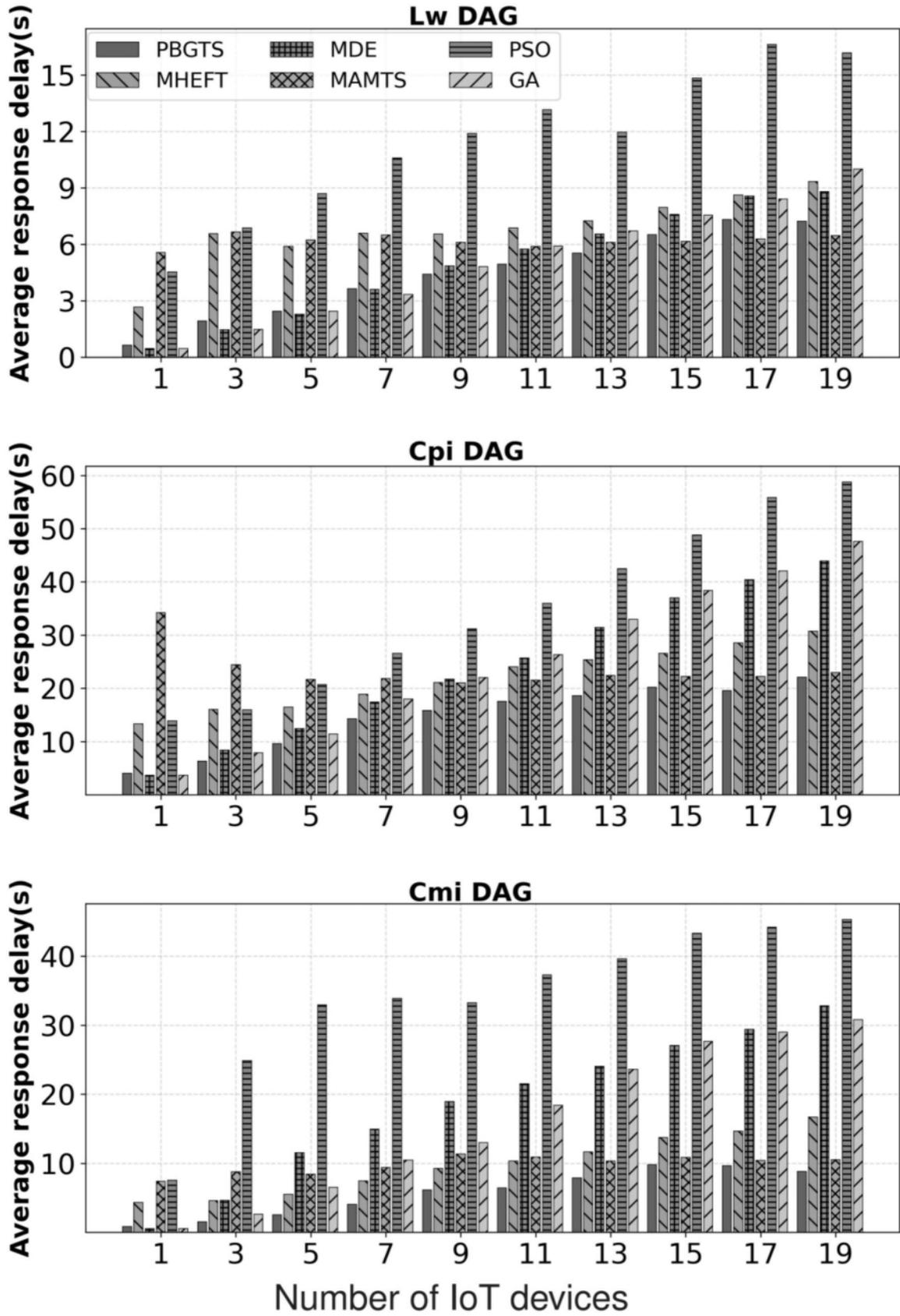


图6

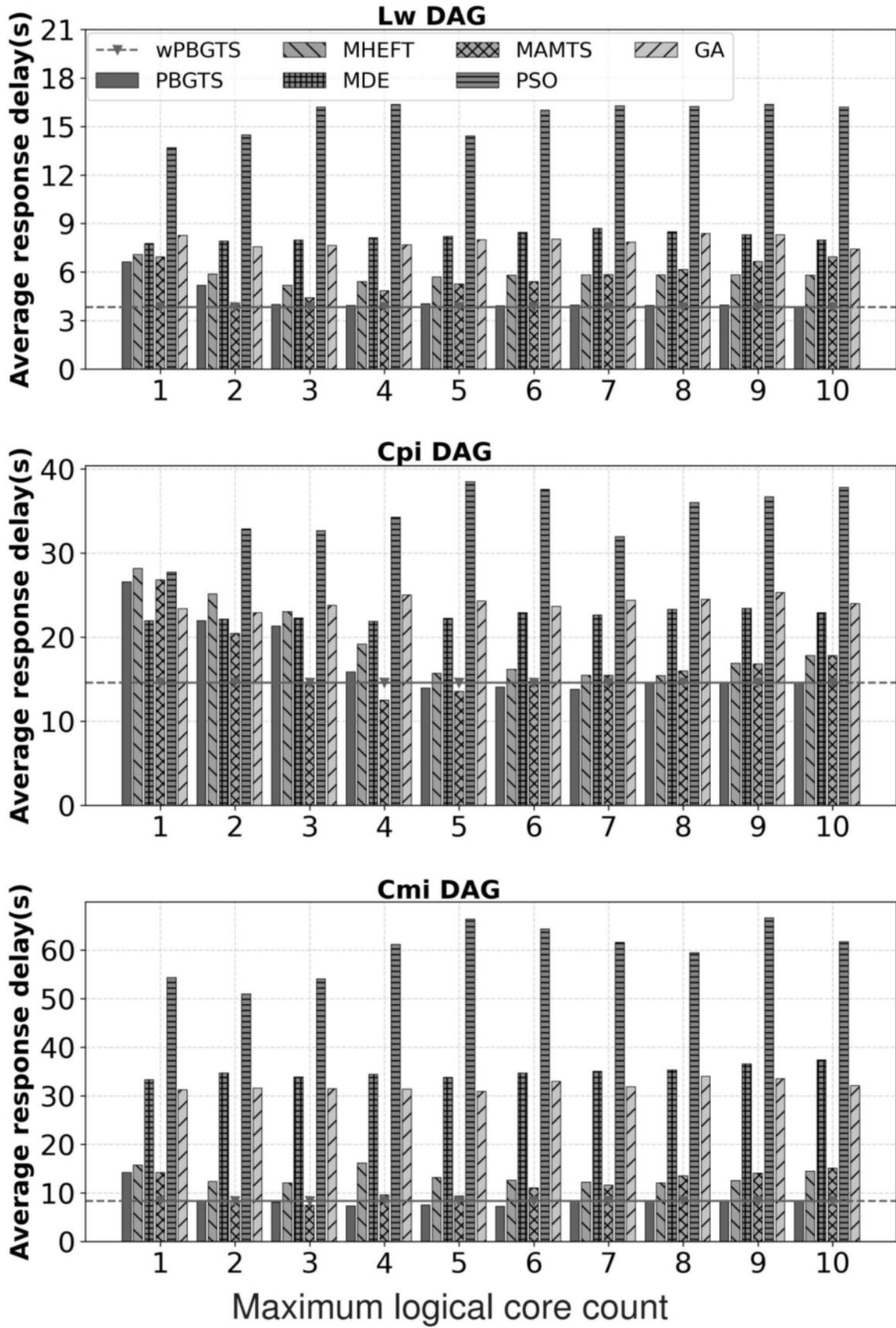


图7