

移动云环境下多 DAG 节能调度算法

薛慧丽^{1,2}, 邵孟良¹

1. 广州南洋理工职业学院 信息工程学院, 广东 广州 510925

2. 广东技术师范学院 电子与通信工程学院, 广东 广州 510665

摘要: 移动云环境下多 DAG 节能调度 MEO 与 MES 算法主要是基于智能手机能耗模型, 通过充分地利用集群任务间的松弛时间, 在执行任务许可的情况下, 通过任务合并的方式尽可能减少服务器的使用数目并降低传输能耗, 最终实现总能量的节约与开销。特别是随着任务数的增多, MEO 算法与 MES 算法更能突显节能的效果, 其最高节能效果可达 20% 以上, 节能优势最明显。

关键词: 移动云计算; 多个 DAG; 节能调度

中图分类号: TP306

文献标识码: A

文章编号: 1000-2324(2017)03-0327-07

Multi DAG Energy Saving Scheduling Algorithm under Mobile Cloud Environment

XUE Hui-li^{1,2}, SHAO Meng-liang¹

1. School of Information Engineering/Guangzhou Nanyang Polytechnic College, Guangzhou 510925, China

2. School of Electronic and Communication Engineering/Guangdong Polytechnic Normal University, Guangzhou 510665, China

Abstract: Under the cloud environment, the MEO and MES algorithms with multi-DAG energy saving scheduling are mainly on basis of the energy consumption model of smart phones to use the relaxation time among the cluster tasks fully and reasonably. If circumstances permitting, the number of the servers can be reduced through the combination of tasks to reduce the transmission energy consumption, finally, the total energy savings and spending are achieved. With the increase of tasks, MEO and MES algorithms can highlight the energy saving effect as much more as 20%, energy saving advantages are most obvious.

Keywords: Mobile cloud computing; multi-DAG; energy saving scheduling

随着移动互联网的迅速发展, 主要针对 PC 机的云计算服务正在逐渐实现从 PC 机到移动终端的战略转移。据工业和信息化部权威统计^[1], 2015 年前三个季度, 全国移动电话用户累计达到 13.01769 亿户, 其中 3G 用户 4.08 亿户, 4G 用户已达 3.56 亿户。现在移动终端的计算能力、存储容量, 以及远程网络所提供的一些服务与功能, 已经满足不了目前 3G、4G 移动用户的需求, 而云计算则因无前期投资、操作成本低、计算能力强、存储容量大、可扩展性强、易于访问等特点^[2]而能很好地改善移动用户的体验。所以, 移动云计算 (Mobile Cloud Computing, MCC) 目前已经成为很多研究人员关注的热点问题。

云计算虽然有很明显的优势, 但是也面临着很重要的问题。根据中国云计算的相关数据显示^[3], 我国数据中心 2009 年的总耗电量就达到了 364 亿 kWh, 2015 年估计要达到 1 000 亿 kWh, 这相当于三峡电站一年的总发电量, 2020 年将超过 2500 亿 kWh, 届时或将超过当前全球数据中心的能耗总量。这就是说, 如果云计算数据中心的高能耗所带来的问题不能得到根本性的解决, 那么将会严重抑制云计算、移动云计算的进一步健康发展。

云计算数据中心节能调度的主要目标是尽可能提高资源的利用率, 同时减少资源的占用时间和占用数目, 从而使整个任务集群系统既可以发挥良好性能, 又可以节省能耗。文献[4]研究发现, 利用动态电压频率调整 (Dynamic Voltage and Frequency Scaling, DVFS) 机制, 结合 EES (Energy-Efficient Scheduling) 节能调度算法, 能够在松弛时间内或任务容忍限度下, 降低 CPU 运行频率, 并降低处理器能耗开销, 从而达到节能目的。文献[5]则指出, 对于通信密集型任务, 基于 DVFS 机制的 EES 算法是很难获得良好节能效果的。文献[6]认为, 在处理器松弛时间内, 通过复制、执行先驱任务, 不但可以减少任务间的传输能耗, 而且可以缩短任务的总执行时间, 从而达到整体节能优化的效果。

收稿日期: 2015-11-19

修回日期: 2015-12-24

基金项目: 2016 年广州市科技计划项目产学研重大专项(201604010103); 广东高职教育研究会 2015 年课题 (GDGZ15Y134); 2016 年广东高等教育学会高职高专云计算与大数据委员会项目 (GDYJSKT16-07)

作者简介: 薛慧丽(1976-), 女, 硕士研究生, 讲师。主要从事计算机图形学、智能信息处理、云计算研究。E-mail: 1398804839@qq.com

文献[7]还提出了 EAD-PRO (Energy Aware Duplication Processor Reduction Optimizing) 能量感知处理器合并优化算法, 能够通过能量感知, 对处理器运行的任务进行合并优化, 从而实现节能目的。因为这些研究考虑的基本上只是某一种任务、单 DAG (Directed Acyclic Graph, 有向无环图) 的情况, 而云计算数据中心往往都是多 DAG 复杂任务并存的状态, 所以文献[8]提出了云计算环境下抢占式多 DAG 工作流的动态调度模型以提高资源利用率的方法。文献[9]试图用 EFRD 算法来解决云计算环境下多 DAG 共享资源调度的资源分配优化问题。文献[10]提出了一种考虑在虚拟机之间链路通信竞争的动态多 DAG 分层调度来计算传输能耗的方法。文献[11]则基于计算密集型和通信密集型这两种任务特点分析的基础之上, 提出了 MREO 算法, 这种算法主要是通过整合独立任务, 优化路径进程, 减少处理器数量, 从而降低系统的计算和通信能量开销, 实现节能效果。

本文主要基于以上的相关研究基础, 提出两种专门针对移动云环境下多 DAG 节能优化算法——MEO (Multi Energy Optimization) 与 MES (Multi Energy Scheduling)。这两种算法主要是通过充分合理地利用集群任务间的松弛时间, 在执行任务许可的情况下, 通过任务合并的方式尽可能减少服务器的使用数目并降低传输能耗, 最终实现总能量的节约与开销。这两种方法都主要是通过两个模块实现: 第一是任务合并模块, 根据任务的依赖关系确定处理器的合并方案; 第二是松弛时间优化模块, 通过任务合并之后的残留松弛时间进行优化处理。

1 多 DAG 节能调度模型

在图论中, 有向无环图 (DAG 图) 所表示的是从某个顶点出发, 经过若干条边无法再回到出发点, 因为在 DAG 图中从某一个点到达另一个点所经过的两种路线不一定是形成环状。在通常情况下, 一个 DAG 工作流表示若干个具有依赖关系的并行任务, 而多个不同有向无环的 DAG 工作流是相互独立的, 并没有相互约束的关系, 其模型基本参数如表 1 所示。

表 1 多 DAG 工作流节能调度模型基本参数

Table 1 Basic parameters of energy saving dispatching under the multi DAG workflow

参数 Parameter	含义 Meaning	参数 Parameter	含义 Meaning
D	DAGs 工作流所有任务的集合	$AST(n_{ij})$	任务最早开始时间
m	集合 D 中 DAG 工作流任务的总数	$AFT(n_{ij})$	任务最早完成时间
n_{ij}	第 i 个 DAG 工作流中的第 j 个任务	$AST'(n_{ij})$	任务最晚开始时间
n_{ik}	第 i 个 DAG 工作流中的第 k 个任务	$AFT'(n_{ij})$	任务最晚完成时间
n_i	第 i 个 DAG 工作流中的任务总数	$T_{excc}(n_{ij})$	任务执行时间
e_{ijk}	在第 i 个 DAG 任务 j 与任务 k 之间的关系	$Slack(n_{ij})$	任务松弛时间

定义: 一个有向无环 DAG 图 $D, D = (V, E), V$ 为顶点集合, 代表将被处理的任务集合, $V = \{n_{ij} | 1 \leq i \leq m, 1 \leq j \leq n_i\}$; E 为向边集合, 代表 DAG 工作流中具有依赖关系的各并行任务, $E = \{e_{ijk} | 1 \leq i \leq m, 1 \leq j \leq n_i, 1 \leq k \leq n_i\}$ 。

每一个 $e_{ijk} \in E, c_{ijk}$ 表示任务 n_{ij} 通过网络传输到任务 n_{ik} 所用的时间, 如果任务 n_{ij} 和任务 n_{ik} 在同一个处理节点上, $c_{ijk} = 0$ (网络传输耗时为 0)。如果任务 n_{ij} 和任务 n_{ik} 不在同一个理节点上, $c_{ijk} \geq 0$ (需要一定的传输时间)。

如: 在图 1 有向无环 DAG 工作流示意图中, 共有 A、B、C、D、E、F、G、H、I 等九个具有依赖关系的并行任务, 从 A 到 H 或者从 A 到 I, 都会出现不一样的工作流, 如果以从 A 到 I 为例, 则有 $A \rightarrow D \rightarrow G \rightarrow I, A \rightarrow B \rightarrow D \rightarrow G \rightarrow I, A \rightarrow B \rightarrow E \rightarrow I$ 与 $A \rightarrow B \rightarrow E \rightarrow G \rightarrow I$ 四条不同的工作路径。比如, 任务 B 和任务 D 因为不在同一个节点, 如果要把任务 B 的结果传输给任务 D, 那么需要 2 个单位时间。同理, 如果要把任务 B 的结果传输给任务 E, 则需要 3 个单位时间。

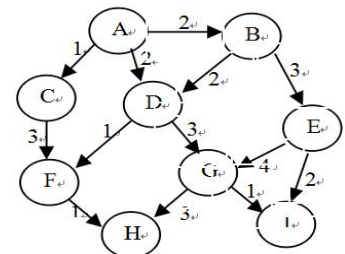


图 1 有向无环 DAG 工作流示意图
Fig.1 The running process of DAG

2 多 DAG 节能调度算法

针对移动计算节点的多 DAG 工作流, 我们主要采用合并调度算法。首先, 通过松弛时间计算公式 $Slack(n_{ij}) = AFT(n_{ij}) - AST(n_{ij}) - T_{excc}(n_{ij})$, 把没有松弛时间的移动节点, 以及松弛时间不合要求的移动

节点进行界限分支,从而降低不必要的资源开销。其次,通过回溯算法找到能量消耗最小的合并方式进行最终合并。

算法 1 MEO 算法合并调度部分

```

Input: DAGs
Output: S
D←{DAGs}; Makespan←Cmakespan(D);
Re←Allocate(D);Ge←Prio(Re);S'←null;
MEO(A)
  Root(A0);A'←A - {A};
  For i=0; i<=length(A');i++ do
  If ismerge(root A'i) = true & time(root A'i) <=makepan then
  J←{root & A'中的所有作业};Job(J);
  RT←{root + A'i} + {A' - {A'i}};
  Prio(RT);S'←RT; MEO(RT);
  End
  Else
  If i >= length(A') then;Temp←A';
  For j = 1; j< length(Temp); j++ do
    root←root(Temp[j]);
  End
  For k = j++; K< length(Temp); k++ do
  If ismerge(root + Temp[k]) == true then
  RT←{root + Temp[k]} + {Temp - Temp[k]};
  Delete(Temp[k]);
  End
  End
  End
  if RT∈S' then return;
  End
  Else
  S'←S' + RT;
  If Cenergy(RT) < Cenergy(S) then
  S = RT
  End
  End
return;
```

算法 2 MES 算法合并调度部分

```

Input: DAGs
Output: S
D←{DAGs}; Makespan←Cmakespan(D);
B1←Allocate(D);B2←Prio(B1);A'←null;
EMS(B2)
  Root(B2); B3←B2 - {B2};
  For i=0; i<=length(B3);i++ do
```

```

If ismerge( $root + B_3$ ) = true && time( $root B_3$ )  $\leq$  makepan then
 $J \leftarrow \{root \&\& B_3 \text{ 中的所有作业}\}; Job(J);$ 
 $RT \leftarrow \{root + B_{3i}\} + \{B_2 - \{B_{3i}\}\};$ 
 $Prio(RT); A' \leftarrow RT; MEO(RT);$ 
End
Else
If  $i \geq length(B_3)$  then;  $Temp \leftarrow B_3;$ 
For  $j = 1; j < length(Temp); j++$  do
     $root \leftarrow root(Temp[j]);$ 
End
For  $k = j++; K < length(Temp); k++$  do
If ismerge( $root + Temp[k]$ ) == true then
 $RT \leftarrow \{root + Temp[k]\} + \{Temp - Temp[k]\};$ 
    Delete( $Temp[k]$ );
End
End
End
if  $RT \in A'$  then return;
End
Else
 $A' \leftarrow A' + RT;$ 
If  $Cenergy(RT) < Cenergy(S)$  then
 $S = RT$ 
End
End
return;

```

算法 3 MEO 算法松弛时间处理部分

Input: S

Output: T

```

If slacking( $S$ ) == ture then
    For  $S! = \Phi$  do
         $T1 \leftarrow EAD - PRO(S); T2 \leftarrow EES(S);$ 
    End
If  $Cenergy(T1) \leq Cenergy(T2)$  then;  $T \leftarrow T2;$ 
End
Else
 $T \leftarrow T1;$ 
End
End
Return  $T;$ 

```

算法 4 MES 算法松弛时间处理部分

Input: S

Output: T

```

If slacking( $S$ ) == ture then

```

```

For  $S' = \Phi$  do
    T1  $\leftarrow$  EADUS(S);
    T2  $\leftarrow T'_{excc}(S_{ni}) = T_{excc}(S_{ni}) + Slack(S_{nij})$ 
     $U'_{ij} = \frac{T_{excc}(S_{ni})}{T'_{excc}(S_{ni})} \times U_{ij}$ ;
End
If Cenergy(T1)  $\leq$  Cenergy(T2) then;  $T \leftarrow T2$ ;
End
Else
 $T \leftarrow T1$ ;
End
End
Return T;
```

3 多 DAG 节能实验结果与分析

为了更好地评估与分析本文所提出的两种算法的节能效果，笔者主要通过模拟程序随机生成有利于验证实验数据所需的各种形态的 DAG 图，并将本文提出的算法与 EES、EAD-PRO 这两种算法进行对比分析。同时，为了计算与分析方便，DAG 工作流程中相关任务的执行与传输时间，都取 1 到 10 区间中的整数。

3.1 移动云环境 MEO 算法下多 DAG 节能优化实验分析

依据第 2 节的“多 DAG 节能调度模型”以及图 1 所示的分配策略，首先进行处理器占用数目对比分析实验：通过模拟程序随机生成第一张有向无环 DAG 图，并记录处理器最初被占用的数目。接着，再通过模拟程序随机生成第二张有向无环 DAG 图，进而把这两张不同的 DAG 图，利用上文所述的 MEO 算法合并调度部分，计算出合并优化之后处理器被占用的数目，并与最初记录的处理器被占用的数目进行对比。依此类推，所有的实验数据都是基于上一张 DAG 图，与新添加的下一张 DAG 图进行对比。图 2 就是通过 MEO 合并优化算法与合并前处理器被占用数目的对比分析数据。

基于以上处理器占用数目对比分析实验，笔者针对移动云环境下多 DAG workflow 节能优化问题，利用 MEO 算法进行了总完成时间、平均执行时间、资源利用率、总节能效果等方面与 EES、EAD-PRO 两种算法进行对比分析，进而得到以下系列结果。

3.1.1 MEO 节能算法在总完成时间方面的优势分析 在执行处理任务时，多种不同形态的 DAG，利用 MEO 算法与 EES 和 EAD-PRO 两种算法进行对比，分析计算处理器完成所有任务的时间总和，即总完成时间，三种算法的总完成时间如图 3 所示。

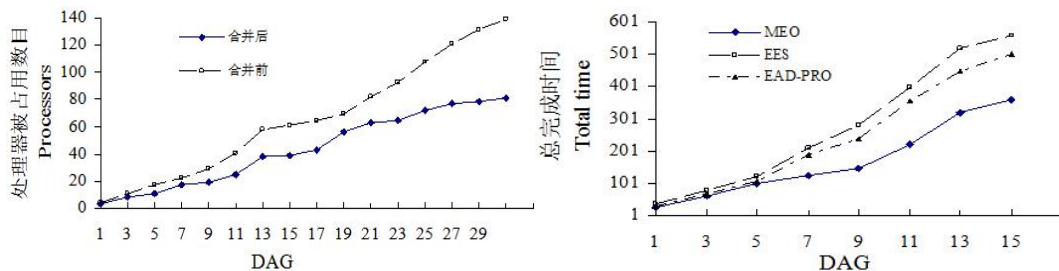


图 2 处理器被占用数目与有向无环 DAG 图数关系 图 3 MEO 节能算法在总完成时间方面的优势分析图

Fig.2 The relationship between processors and DAG Fig.3 The advantages of MEO algorithm in terms of completion

由图 3 可知，EES 算法所用的总执行时间最长，主要是因为 EES 只考虑了 DVFS 机制；EAD-PRO 算法所用的总执行时间比 EES 算法少一点，但是区分度不是很明显，因为 EAD-PRO 只合并了有依赖关系的处理器的任务；而 MEO 算法所用的总执行时间，特别是 DAG 图数越大，其所用的时间明

显少于前面两种算法, 因为 MEO 算法可以合并多种 DAG 集合中轻负载处理器的任务。

3.1.2 MEO 节能算法在平均执行时间方面的优势分析 在 MEO、EES 和 EAD-PRO 三种算法所求出总完成时间的基础上, 同时记录模拟程序生成的随机任务数, 通过总执行时间除以任务总数, 最终计算出平均执行时间, 三种算法的平均执行时间如图 4 所示。

EAD-PRO 算法与 MEO 算法, 因为都应用了处理器合并机制, 所以, 任务平均完成时间都比 ESS 算法低。但是, EAD-PRO 算法单一, 对于复杂的多 DAG 工作流, 特别是在任务数数目众多的情况下, 其平均完成时间还是高于 MEO 算法。

从图 4 中可以看到, ESS 算法的资源利用率, 明显要低于 MEO 和 EAD-PRO 两种算法。在 DAG 图数不多时, MEO 和 EAD-PRO 算法的优劣不是很明显, 但是, MEO 算法在复杂的多 DAG 图数的情况下, 其资源利用率明显优于 EAD-PRO 算法与 ESS 算法。实验证明, MEO 算法不但可以更准确地整合 DAG 工作流中的轻负载处理器, 而且能够有效提高资源利用率。

3.1.3 MEO 节能算法在资源利用率方面的优势分析 在 3.1.1 总完成时间的基础之上, 把 \sum CPU 数、 \sum 时间与 CPU 利用率三者的乘积, 除以总完成时间, 就可以得到 MEO、EES 和 EAD-PRO 三种算法下处理器的资源利用率, 结果如图 5 所示。

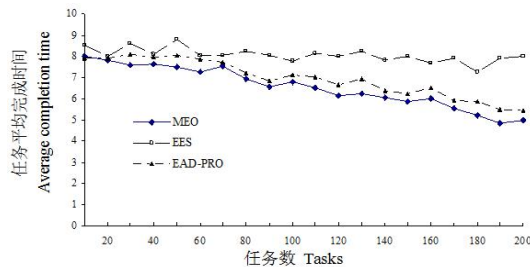


图 4 MEO 节能算法在平均执行时间方面的优势分析图 Fig.4 The advantages of MEO algorithm in average execution time

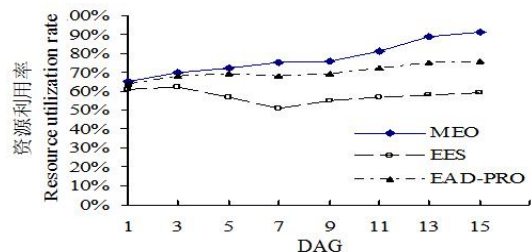


图 5 MEO 节能算法在资源利用率方面的优势分析图 Fig.5 The advantages of MEO algorithm in resource utilization rate

如图 5 所示, 与传统耗能相比, MEO、EES 和 EAD-PRO 三种算法均有较明显的节能效果, 而 MEO 算法在任务数繁多、DAG 工作流复杂的情况下, 节能效果达到 20% 以上, 节能优势最明显。

3.1.4 MEO 节能算法在总节能效果方面的优势分析 为说明 MEO 算法在节能效果方面的优势, 在预先准备的硬件环境下模拟同构集群环境, 对实验中随机生成的任务进行能耗计算, 并与 EES 和 EAD-PRO 算法对比, 通过算法总耗能除以传统总耗能对节能效果计算, 得到如图 6 的结果。

3.2 移动云环境 MES 算法下多 DAG 节能优化实验分析

MES 算法实验所采用的还是 3.1 中所介绍的方法, 在移动云环境下将 MES 算法的合并优化策略, 与 EADUS、TEBUS 等传统方法进行对比, 分别在移动计算节点的占用数目、总执行时间、总节能效果三个方面进行了对比, 其优势分析如下。

3.2.1 MES 节能算法在移动节点占用数目方面的优势分析 在移动计算云环境正常运行的情况下, 移动计算节点数目通常都会随着并行任务集群数目的增加而增加。但是, 如果用 MES 算法对多 DAG 工作流中的任务进行合并调度后, 占用移动计算节点的数目将会大幅度减少, 其分析结果如图 7。

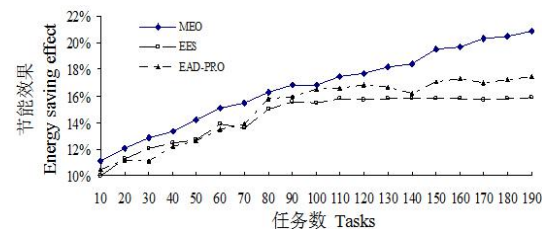


图 6 MEO 节能算法在总节能效果方面的优势分析 Fig.6 The advantages of MEO algorithm in energy-saving effect

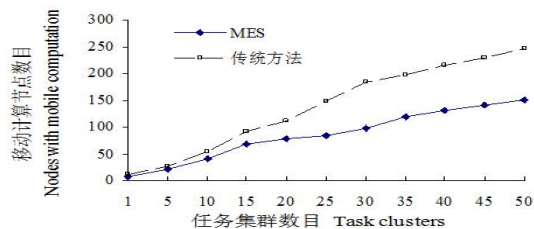


图 7 MES 节能算法在移动节点占用数目方面的优势分析 Fig.7 The advantages of MES algorithm in mobile nodes

在 DAG 工作流中的任务数较多的时候, 使用 MES 算法最大的优势就是占用移动计算节点数目

比传统算法可减少 60%以上。可见, MES 算法对整个节点集群整体节能提供了实现的途径。

3.2.2 MES 节能算法在总执行时间方面的优势分析 在移动私有云环境下, 通过 EADUS、TEBUS 算法与 MES 算法在多 DAG 调度总执行时间方面进行了对比, 得到如图 8 所示结果。

据图 8 可知, EADUS 算法的总完成时间一直高于其他两个算法。在执行任务比较少的情况下, MES 与 TEBUS 的差别并不大, 有时 TEBUS 算法比 MES 算法还要省时。但是随着任务数的增加, MES 算法的优势就突显出来了。这主要是因为 MES 算法合并了多 DAG workflow 状态下的多种依赖关系的轻负载计算节点, 而 TEBUS 算法虽然对性能与能耗都有考虑与兼顾, 但对多 DAG workflow 状态下的多依赖关系的复杂情况并没有考虑。

3.2.3 MES 节能算法在总节能效果方面的优势分析 为了说明 MES 算法在节能效果方面的优势, 笔者基于前文所提出的智能手机能耗模型, 在预先准备的硬件设备下模拟同构集群环境, 充分利用模拟程序所生成的大量随机任务进行能耗计算, 并与 EADUS 和 TEBUS 两种算法进行对比, 通过算法总耗能除以传统总耗能对总节能效果进行计算, 得到如图 9 所示结果。

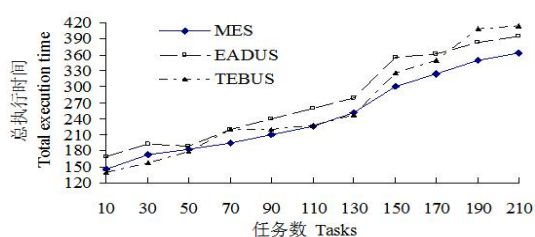


图 8 MES 节能算法在总执行时间方面的优势分析
Fig.8 The advantages of MES algorithm in total execution time

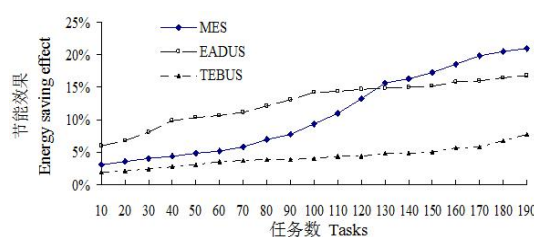


图 9 MES 节能算法在总节能效果方面的优势分析
Fig.9 The advantages of MES algorithm in energy-saving effect

由图 8 可知, MES 与 EADUS 算法的总节能效果比 TEBUS 算法都有优势, 在任务数不是太多的情况下, EADUS 算法的总节能效果比 MES 算法还好, 但是, 随着任务数越来越多、DAG workflow 越来越复杂, MES 算法的优势才能突显出来, 其最高节能效果可以达到 20%以上, 节能优势最明显。

4 小结

基于智能手机能耗模型, 在预先准备的硬件设备下模拟同构集群环境, 充分利用模拟程序所生成的大量随机任务进行能耗计算, 通过 MEO 算法与 EES、EAD-PRO 算法的对比, 以及 MES 算法与 EADUS、TEBUS 算法的对比, 最后得出结论: 随着任务数增多, 最能突显多 DAG workflow 节能的算法是 MEO 算法与 MES 算法, 其最高节能效果均可达 20%以上, 节能优势最明显。

参考文献

- [1] 中华人民共和国工业和信息化部.2015 年 10 月电话用户分省情况[EB/OL].<http://yxj.miit.gov.cn/n1146312/n1146904/n1648372/c4432696/content.html>.
- [2] 薛慧丽.云计算的架构及核心技术[J].智能计算机与应用,2014(4):63-67
- [3] 中国云计算.数据中心火热耗能问题或成最大阻碍[EB/OL].<http://www.chinacloud.cn/show.aspx?id=16978&cid=14>
- [4] Huang QJ, Su S, Li J, *et al.* Enhanced energy-efficient scheduling for parallel applications in cloud[C]//Proceedings of the 12th IEEE /ACM International Symposium on Cluster, Cloud and Grid Computing. New York:ACM Press, 2012:781-786
- [5] Shu C, Feng W. A power-aware run-time system for high-performance computing[C]//Proceedings of the 2005 ACM/IEEE Conference on Supercomputing. Washington, DC:IEEE Computer Society, 2006: 258-267.
- [6] Zong ZL, Manzanares A, Ruan XJ, *et al.* EAD and PEBD: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters[J]. IEEE Transactions on Computers, 2011,60(3):360-374
- [7] 李新,贾智平,鞠雷.一种面向同构集群系统的任务节能调度优化方法[J].计算机学报,2012,35(3):591-602
- [8] 孙月,于炯,朱建波.云计算中一种多 DAG workflow 可抢占式调度策略[J].计算机科学,2014,41(3):145-148
- [9] 田国忠,肖创柏,赵娟娟.云计算环境下多 DAG 调度的资源分配进化算法[J].计算机应用研究,2014,31(9):2798-2802
- [10] 景维鹏,吴智博,刘宏伟.云计算环境下动态多 DAG workflow 可靠调度方法[J].西安电子科技大学学报:自然科学版,2016,43(2):92-97
- [11] 刘丹琦,于炯,英昌甜.云计算环境下多 DAG workflow 的节能调度算法[J].计算机应用,2014,33(9):2410-2415