



## (12) 发明专利申请

(10) 申请公布号 CN 103885839 A

(43) 申请公布日 2014. 06. 25

(21) 申请号 201410136320. X

(22) 申请日 2014. 04. 06

(71) 申请人 孙凌宇

地址 343000 江西省吉安市吉州区安宁路  
15 号 6 栋 101 室

申请人 冷明

冷子阳

(72) 发明人 冷明 孙凌宇 冷子阳

(51) Int. Cl.

G06F 9/50 (2006. 01)

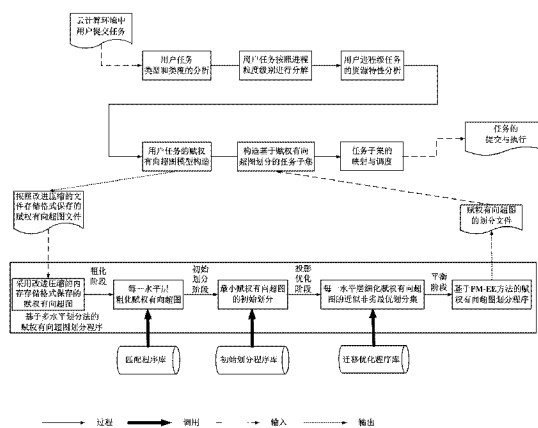
权利要求书5页 说明书12页 附图3页

## (54) 发明名称

基于多水平划分法和赋权有向超图的云计算任务调度方法

## (57) 摘要

本发明涉及一种云计算环境下的基于多水平划分法和赋权有向超图的云计算任务调度方法,其采用赋权有向超图描述任务的资源需求及依赖关系,并生成相应的赋权有向超图文件;然后启动基于多水平划分法的赋权有向超图划分程序,对生成的赋权有向超图进行划分;最后依据赋权有向超图的划分结果构造任务子集,通过 MapReduce 任务调度模型对其进行映射和调度。采用本发明的基于多水平划分法和赋权有向超图的云计算任务调度方法,不仅仅有效地提高了任务调度的效率,还显著地提高了任务划分的性能,具有较好的实用性。



1. 一种基于多水平划分法和赋权有向超图的云计算任务调度方法,其特征在于,具体步骤如下:

步骤 1,类型类度分析,输入云计算环境下用户提交的任务,并对其进行类型和类度的分析,确定任务的并行化程度和特点;

步骤 2,进程粒度分解,根据用户任务的并行化程度和特点,以及云计算的资源共享分配方式等独特性质,对用户任务按照进程粒度级别进行分解;

步骤 3,资源特性分析,根据云计算的资源共享分配方式等独特性质,对分解后的任务进行资源特性分析;

步骤 4,赋权有向超图文件生成,依据对任务资源特性的分析结果,建立描述其资源需求及依赖关系的赋权有向超图模型,并按照改进压缩的文件存储格式保存为赋权有向超图文件;

步骤 5,赋权有向超图划分,启动基于多水平划分法的赋权有向超图划分程序,读取赋权有向超图文件,采用改进压缩的内存存储格式对赋权有向超图进行存储,对生成的赋权有向超图进行划分,将最终得到的划分结果存储在赋权有向超图划分文件中;

步骤 6,任务子集构造,在检测到基于多水平划分法的赋权有向超图划分程序完成划分之后,从赋权有向超图划分文件中读取相应的划分结果,依据赋权有向超图的划分结果构造进程级任务子集;

步骤 7,任务映射调度,通过 MapReduce 任务调度模型,对基于赋权有向超图优化划分构造的任务子集进行映射和调度,实现在云计算环境中的任务提交与执行,有效地均衡云计算平台的负载和缩短整个任务完成的时间跨度;

上述的步骤 4 中,所述的赋权有向超图的改进压缩的文件存储格式的如下:

步骤 4.1,文件格式的第 1 行第 1 个参数代表着有向赋权超边的数目  $m$ ,第 2 个参数代表着赋权结点的数目  $n$ ;

步骤 4.2,文件格式的第 2 行开始到第  $m+1$  行的每行代表着一一条有向赋权超边的相关信息,第 1 个数值为有向赋权超边的权值信息,其余数值为有向赋权超边的结点信息,其中每行的最后一个数值代表有向赋权超边的尾端结点信息,且有向赋权超边的源端结点信息处于有向赋权超边的权值信息和尾端结点信息之间;

步骤 4.3,文件格式的第  $m+2$  行开始到第  $m+n+1$  行的每行代表着一个赋权结点的权值信息;

上述的步骤 5 中,所述的基于多水平划分法的赋权有向超图划分程序的步骤如下:

步骤 5.1,读取赋权有向超图文件,采用改进压缩的内存存储格式对赋权有向超图进行存储;

步骤 5.2,进入到多水平划分法的粗化阶段,采用赋权有向超图的结点匹配程序将当前水平层粗化赋权有向超图的某些结点结合在一起,得到下一水平层的粗化赋权有向超图,重复此过程直到粗化赋权有向超图足够小为止,即得到一个最小赋权有向超图;

步骤 5.3,进入到多水平划分法的初始划分阶段,运行基于 FM 方法的赋权有向超图划分程序,得到最小赋权有向超图的初始划分;

步骤 5.4,多水平划分法优化阶段的离散群智能搜索程序初始化,初始化全体群智能粒子的全局历史最优位置向量和每个群智能粒子的自身位置向量、自身速度向量、自身历史

最优位置向量；

步骤 5.5, 进入到多水平划分法的优化阶段, 从最小赋权有向超图投影回初始赋权有向超图, 在每一水平层的细化赋权有向超图中, 采用离散群智能搜索程序对细化赋权有向超图的投影划分进行优化；

步骤 5.6, 进入到平衡阶段, 运行基于 FM-EE 方法的赋权有向超图划分程序；由于在基于多水平划分法的赋权有向超图划分过程中, 可能违背赋权有向超图划分问题的平衡约束条件, 因此在基于多水平划分法的赋权有向超图划分所求解的基础上, 运行基于 FM-EE 方法的赋权有向超图划分方法, 使划分解满足平衡约束条件, 从而得到赋权有向超图划分问题的划分解；

步骤 5.7, 将最终得到的赋权有向超图划分结果存储在赋权有向超图划分文件中；

上述的步骤 5.1 中, 所述的赋权有向超图的改进压缩的内存存储格式如下；

步骤 5.1.1, 使用 vwgt 数组存储赋权有向超图中结点的权值信息, 且 vwgt 数组的大小为赋权有向超图中的结点个数；

步骤 5.1.2, 使用 xadj 数组存储每个结点所有邻接有向超边列表的起始位置信息, 即第  $i$  个结点的终止位置为第  $i+1$  个结点的起始位置减 1, 且 xadj 数组的大小为赋权有向超图中的结点个数加 1, xadj 数组最后一个元素用于存放最后一个结点的终止位置；

步骤 5.1.3, 使用 adjncy 数组存储每个结点所有邻接有向超边的列表信息, 第  $i$  个结点的邻接有向超边列表存储在 adjncy 数组中, 从 adjncy[xadj[i]] 到 adjncy[xadj[i+1]-1]；

步骤 5.1.4, 使用 eptr 数组存储每条有向超边所包含的结点列表的起始位置信息, 即第  $j$  条有向超边的终止位置为第  $j+1$  条有向超边的起始位置减 1, 且 eptr 数组的大小为赋权有向超图中的有向超边个数加 1, eptr 数组最后一个元素用于存放最后一条有向超边的终止位置；

步骤 5.1.5, 使用 eind 数组存储每条有向超边所包含结点的列表信息, 其中每条有向超边的尾端结点只有 1 个, 且每条有向超边尾端结点的所有直接前驱结点都包含在该有向超边的源端子集中；第  $j$  条有向超边的结点列表存储在 eind 数组中, 从 eind[eptr[j]] 到 eind[eptr[j+1]-1], 其中第  $j$  条有向超边的源端结点为 eind[eptr[j]] 到 eind[eptr[j+1]-2], 第  $j$  条有向超边的尾端结点为 eind[eptr[j+1]-1]；

步骤 5.1.6, 使用 hewgt 数组存储有向超边的权值信息, 且 hewgt 数组的大小为赋权有向超图中的有向超边数目；

上述的步骤 5.2 中, 所述的当前水平层粗化赋权有向超图的结点匹配程序的步骤如下：

步骤 5.2.1, 标注当前水平层粗化赋权有向超图中所有结点处于未匹配状态；

步骤 5.2.2, 运行赋权有向超图的结点核值计算程序, 基于结点属性函数值进行当前水平层粗化赋权有向超图中所有结点的核值求解, 并按照结点的核值进行非严格降序排序；

步骤 5.2.3, 按照结点核值的非严格降序访问处于未匹配状态的结点  $v$ ；如果结点  $v$  还有邻接结点未匹配, 那么选取处于没有匹配状态的且权值最大的邻接超边的邻接结点  $u$  和结点  $v$  匹配, 且标注这两个结点为匹配状态；如果结点  $v$  所有邻接匹配结点处于匹配状态, 那么结点  $v$  仍处于未匹配状态, 在粗化过程中直接拷贝到下一水平层的粗化赋权有向超图

中；

步骤 5.2.4, 步骤重复上一步, 直至所有结点访问结束；

上述的步骤 5.2.2 中, 所述的赋权有向超图的结点核值计算程序的步骤如下：

步骤 5.2.2.1, 计算出所有结点的属性函数值；

步骤 5.2.2.2, 对所有结点的属性函数值进行非严格降序排序；

步骤 5.2.2.3, 按照结点属性函数值的非严格降序次序访问每个结点, 计算每个结点的核值；

上述的步骤 5.2.2.2 中, 所述的结点属性函数值的非严格降序排序的步骤如下：

步骤 5.2.2.2.1, 根据结点的属性函数值属于在一定范围内的整数, 扫描所有结点的属性函数值, 统计每一种属性函数值的结点个数, 存储在计数辅助数组 bin 中；

步骤 5.2.2.2.2, 针对每一种属性函数值, 借助计数辅助数组 bin, 计算出在所有结点的属性函数值中, 小于该属性函数值的结点个数, 存储在位置辅助数组 pos 中；

步骤 5.2.2.2.3, 扫描所有结点的属性函数值, 针对每一个结点的属性函数值, 借助位置辅助数组 pos, 得到该结点的属性函数值在非严格降序排序的次序, 并将该次序存储在次序辅助数组 vert 中；

上述的步骤 5.2.2.3 中, 所述的结点 v 的核值计算的步骤如下：

步骤 5.2.2.3.1, 将结点 v 的属性函数值作为核值输出；

步骤 5.2.2.3.2, 标记结点 v 从所在的超边 e 中删除；

步骤 5.2.2.3.3, 如果超边 e 删除结点 v 后, 仍包含两个及以上未被标记删除的结点, 则超边 e 仍然存在, 否则删除超边 e；

步骤 5.2.2.3.4, 重新计算结点 v 的邻接结点 u 的属性函数值；

步骤 5.2.2.3.5, 如果邻接结点 u 的属性函数值大于结点 v 的属性函数值, 更新邻接结点 u 的属性函数值, 并且借助计数辅助数组 bin、位置辅助数组 pos 和次序辅助数组 vert 的信息, 快速更新邻接结点 u 在所有结点的属性函数值非严格降序排序的次序；否则不更新邻接结点 u 的属性函数值及其排序的次序；

上述的步骤 5.4 中, 所述的离散群智能搜索程序的初始化步骤如下：

步骤 5.4.1, 初始化全体群智能粒子的全局历史最优位置向量, 设定全局历史最优位置向量为最小赋权有向超图的初始划分；

步骤 5.4.2, 初始化每个群智能粒子的自身位置向量, 遍历每个群智能粒子并设定每个群智能粒子的自身位置向量为最小赋权有向超图的初始划分, 即群智能粒子在最小赋权有向超图的每个维度空间的位置；群智能粒子在最小赋权有向超图的某维度空间的位置, 代表着该维度空间对应的最小赋权有向超图结点所处划分的结点子集；

步骤 5.4.3, 初始化每个群智能粒子的自身速度向量, 遍历每个群智能粒子并在速度阈值范围内, 随机给定每个群智能粒子在每个维度空间的速度向量；

步骤 5.4.4, 初始化每个群智能粒子的自身历史最优位置向量, 遍历每个群智能粒子并设定每个群智能粒子的自身位置向量为自身历史最优位置向量；

上述的步骤 5.5 中, 所述的离散群智能搜索程序的步骤如下：

步骤 5.5.1, 投影每个群智能粒子的自身位置向量, 遍历每个群智能粒子并依据每个群智能粒子的上一水平层粗化赋权有向超图的自身位置向量投影到当前水平层的细化赋权

有向超图上,得到每个群智能粒子在当前水平层细化赋权有向超图的自身位置向量,即群智能粒子在当前水平层的细化赋权有向超图的每个维度空间的位置;群智能粒子在当前水平层的细化赋权有向超图的某维度空间的位置,代表着该维度空间对应的当前水平层的细化赋权有向超图结点所处划分的结分子集;

步骤 5.5.2,投影每个群智能粒子的自身速度向量,遍历每个群智能粒子并依据每个群智能粒子的上一水平层粗化赋权有向超图的自身速度向量投影到当前水平层的细化赋权有向超图上,得到每个群智能粒子在当前水平层细化赋权有向超图的自身速度向量,即群智能粒子在当前水平层的细化赋权有向超图的每个维度空间的速度;

步骤 5.5.3,计算每个群智能粒子的二维辅助数组  $EDG[n][m]$ ,遍历每个群智能粒子并依据每个群智能粒子的当前水平层细化赋权有向超图的自身位置向量,计算每个群智能粒子的二维辅助数组  $EDG[n][m]$ ;

步骤 5.5.4,快速计算每个群智能粒子在当前水平层的细化赋权有向超图的自身位置向量的割切值,遍历每个群智能粒子并依据每个群智能粒子的二维辅助数组  $EDG[n][m]$ ,快速计算每个群智能粒子在当前水平层的细化赋权有向超图的自身位置向量的割切值;

步骤 5.5.5,循环初始化,初始化循环计数器 COUNT 为 0;

步骤 5.5.6,更新每个群智能粒子的自身速度向量,遍历每个群智能粒子并依据每个群智能粒子自身的历史最优位置向量、全体群智能粒子的全局历史最优位置向量,在速度阈值范围内更新自身速度向量;

步骤 5.5.7,更新每个群智能粒子的自身位置向量和二维辅助数组  $EDG[n][m]$ ,遍历每个群智能粒子并计算每个群智能粒子自身位置向量和自身速度向量的向量和,更新自身位置向量,进而依据群智能粒子的自身位置向量计算群智能粒子的二维辅助数组  $EDG[n][m]$ ;

步骤 5.5.8,快速计算每个群智能粒子的自身位置向量的割切值,遍历每个群智能粒子并依据每个群智能粒子的自身位置向量和二维辅助数组  $EDG[n][m]$ ,快速计算每个群智能粒子的当前水平层细化赋权有向超图的自身位置向量的割切值;如果该群智能粒子的自身位置向量的割切值小于自身历史最优位置向量的割切值,则更新该群智能粒子的历史最优位置向量为当前的自身位置向量;如果该群智能粒子的自身位置向量的割切值小于全体群智能粒子的全局历史最优位置向量的割切值,则更新全体群智能粒子的全局历史最优位置向量为该群智能粒子当前的自身位置向量;

步骤 5.5.9,重复步骤 5.5.6、5.5.7、5.5.8,直至循环计数器 COUNT 到达给定的上限;

上述的步骤 5.5.3 和步骤 5.5.7 中,所述的计算群智能粒子的二维辅助数组  $EDG[n][m]$  的步骤如下:

步骤 5.5.3.1,二维辅助数组  $EDG[n][m]$  清零;

步骤 5.5.3.2,读取  $eptr$  数组和  $eind$  数组存储的每条超边所包含的结点信息,基于群智能粒子的当前水平层细化赋权有向超图的自身位置向量,计算每条超边在  $n$  个划分子集  $V_1 \cdots V_n$  的结分子集,即二维辅助数组  $EDG[n][m]$  的两行分别存放  $m$  条超边在  $n$  个划分子集的结分子集;

上述的步骤 5.5.4 和步骤 5.5.8 中,所述的快速计算群智能粒子的自身位置向量的割切值的步骤如下:

步骤 5.5.4.1, 划分割切值清零;

步骤 5.5.4.2, 遍历每条超边是否结束, 如果访问未结束, 即存在超边  $e$  未被访问, 则转步骤 5.5.4.3; 否则访问结束, 返回划分割切值;

步骤 5.5.4.3, 如果满足  $EDG[i][e] \geq 1$  的条件 1 和  $EDG[j][e] \geq 1$  的条件 2 时, 意味着超边  $e$  在划分子集  $V_i$  和  $V_j$  的结点个数都大于等于 1, 即可判定超边  $e$  是两栖边, 并将划分割切值累加上当前超边的权值; 否则判定超边  $e$  不是两栖边, 划分割切值不变;

步骤 5.5.4.4, 转步骤 5.5.4.2;

上述的步骤 5.5.6 中, 所述的群智能粒子的自身速度向量的更新步骤如下:

步骤 5.5.6.1, 计算出群智能粒子的当前自身速度向量与惯性权重参数的乘积;

步骤 5.5.6.2, 计算出群智能粒子的历史最优位置向量和群智能粒子的当前自身位置向量的向量差, 再和认知参数相乘;

步骤 5.5.6.3, 计算出全体群智能粒子的全局历史最优位置向量和群智能粒子的当前自身位置向量的向量差, 再和社会参数相乘;

步骤 5.5.6.4, 计算出步骤 5.5.6.1、步骤 5.5.6.2、步骤 5.5.6.3 得到的三个乘积的向量和, 更新群智能粒子的自身速度向量。

## 基于多水平划分法和赋权有向超图的云计算任务调度方法

### 技术领域

[0001] 本发明涉及一种云计算环境下的基于多水平划分法和赋权有向超图的云计算任务调度方法。

### 背景技术

[0002] 云计算作为分布式计算、并行计算、网格计算等传统技术和网络编程模型、分布式数据存储技术、虚拟化技术等新型技术融合发展的产物,是引领未来信息产业创新的关键战略性技术和手段,将对我国发展高新技术产业具有重要的战略意义。云计算通过将计算任务划分在大规模的廉价服务器集群上,使得人们能够利用分布在各地的闲散资源来处理较为复杂的应用程序,以极低的成本投入获得极高的计算品质。

[0003] 在满足云计算环境要求的前提下,将大量分散的应用程序任务划分成多个具有一定约束关系的任务子集,调度到不同的虚拟机上,获得比其他一些针对网格计算或并行计算的任务调度算法更短的时间跨度和更好的运行质量,是实现云计算高性能的关键核心技术。

[0004] 现有技术的划分系统中有若干种任务的划分法,这些划分法从依赖关系数目最小、划分后任务子集的任务数目均匀分布等不同的方面来实现,主要有基于迁移的划分法、水平嵌套划分法、多水平划分法等方法。

[0005] 基于迁移的划分法。该方法首先产生任务的随机初始划分,同一个任务不能同时属于两个任务子集。在迁移优化阶段,从两个任务子集中各选取一个任务进行成对交换,这两个任务分别属于两个不同的任务子集且收益最大,从而每次都利用交换过程最大限度地改进任务划分质量。记录割切达到最小值时刻的任务划分结果,且一旦交换了所选择的两个任务,在整个迁移过程余下的优化改进中,将这两个任务锁定使得它们不再被选中。重复上述过程直到所有可能的任务都经过迁移之后,回滚到累计收益最大值即割切最小值的时刻。该划分法得到的任务划分结果不稳定,离散性很大,因此限制了该划分法所能解决问题的规模。

[0006] 水平嵌套划分法。该方法首先选择一个任务,将这个任务标上号码 0,然后将所有和这个任务相连的任务标上号码 1,之后对于那些还未标上号码,但是和已经标上号码的任务相连的任务,将其标号为相连任务的号码上加 1。直到一半的任务标上号码,标号过程才结束。那些已经标上号码的任务集合设为一个任务子集,其他任务为另一个任务子集。该划分法只有在选取的初始任务接近外围时,得到的任务划分结果相对较好,总的来说该任务划分结果也不稳定。

[0007] 多水平划分法。Karypis 针对结点规模达到几百万的划分问题,提出了多水平划分的概念,在相对较短的时间内可以得到高质量的划分。该方法包含粗化、初始划分和迁移优化三个阶段。首先,它采用随机匹配策略将某些任务结合在一起,得到下一水平层的粗化任务图,重复此过程直到粗化任务图足够小为止,即得到一个最小任务图。然后,采用划分法对最小任务图进行对分,得到一个初始划分。之后,将最小任务图投影回初始任务图,在每

一水平层的细化任务划分中,按照贪心原则选择收益值最大的任务进行迁移优化,得到最后的任务划分结果。

[0008] 多水平划分法在电路划分中的应用。自多水平划分的概念提出以来,得到了广泛地重视,并应用在电路划分等许多研究领域。

[0009] 2008 年中国专利局公告的由冷明、郁松年和孙凌宇申报,中国专利号为 200710043765.3 号《基于多水平划分法的大规模集成电路划分方法》的发明专利,针对现有技术方案中因采用随机策略进行匹配和贪心原则进行迁移优化,导致无法逃离局部最优的划分,提供了一种改进的基于多水平划分法的大规模集成电路划分方法,有效地提高了大规模集成电路划分的效率和性能。该发明专利在多水平划分法的粗化阶段,通过对结点属性进行赋权无向图中所有结点的核值求解排序,按照基于结点核值的非严格降序访问处于未匹配状态的结点,依据一定规则对其进行匹配,从而将连接性好的结点合并在一起;在多水平划分法的优化阶段,采用免疫克隆优化程序改进贪心原则的局部搜索方法,对在每一水平层投影的划分进行优化,借助克隆操作、克隆变异操作、接种免疫疫苗操作、克隆选择操作,使得改进后的方法在利用启发信息搜索局部最优解的同时,更自由地对具有潜力的解空间进行搜索,增加全局搜索能力。

[0010] 2012 年中国专利局公告的由孙凌宇、冷明和冷子阳申报,中国专利号为 201210155738.6 号《基于多水平划分法和赋权超图的大规模集成电路划分方法》的发明专利,针对采用赋权无向图作为大规模集成电路划分问题的数学模型,存在着赋权无向图最优划分和大规模集成电路最优划分的不一致性,提供了一种基于多水平划分法和赋权无向超图的大规模集成电路划分方法,进一步提高了大规模集成电路划分的效率和性能。该发明采用赋权无向超图对电路划分问题进行数学建模,其中电路逻辑单元表示为赋权无向超图中的结点,电路单元间的连线表示为赋权无向超图中的超边。相比赋权无向图而言,赋权无向超图为电路提供了更为精确的模型:每条超边可以连接两个以上的结点,对应于电路单元间的信号可以连接两个以上的电路逻辑单元。该发明将大规模集成电路划分问题转换为赋权无向超图划分问题,其中大规模集成电路划分问题要求每个电路子集所包含的电路逻辑单元数目相等,对应于赋权无向超图划分问题的平衡约束条件,划分结果使得这些电路子集之间的内连线数据达到最小,对应于赋权无向超图划分问题的最小化总割切。

[0011] 2012 年中国专利局公告的由孙凌宇、冷明和冷子阳申报,中国专利号为 201210150329.7 号《基于结点属性函数的大规模集成电路的核值计算方法》的发明专利,在采用多水平划分法求解赋权无向超图划分问题的粗化阶段中,提供了所需的基于结点属性函数的大规模集成电路的核值计算方法。该发明专利有利于在赋权无向超图的结点匹配过程中,发挥结点核值导向性作用,使得粗化后超图中结点权值趋于大小一致,并最大程度地减少超边的数目以及超边权值之和,为大规模集成电路多水平划分的后续阶段提供更优的粗化超图。

[0012] 2012 年中国专利局公告的由冷明、孙凌宇和冷子阳申报,中国专利号为 201210154573.0 号《基于元胞自动机和赋权超图的大规模集成电路划分方法》的发明专利,采用元胞自动机对赋权无向超图划分问题进行数学建模。其中,元胞对应于赋权无向超图中的结点,邻接元胞对应于邻接超边所包含的结点,元胞的状态对应于所在的划分子集,进而采用快速的元胞收益值和划分割切值的计算方法,有效地降低了大规模集成电路划分方



法的空间复杂度和时间复杂度。

## 发明内容

[0013] 本发明涉及的云计算环境中任务调度包括元任务的调度和依赖任务的调度。元任务之间相互独立,其调度不考虑任务间的数据关联与优先约束关系,因此它只是部分地解决了资源异构性和可用性问题的,缺乏普遍适用性。而依赖任务之间存在先后依赖关系,要求一个任务必须接收到它的所有前驱任务消息后才能开始执行。

[0014] 本发明采用赋权有向超图来构造任务划分问题的数学模型,任务表示为赋权有向超图的结点,任务结点间的先后依赖关系表示为赋权有向超图中的有向超边。赋权有向超图的多对多关系提供了精确描述用户任务的手段,其结点对应于分解后的进程级用户任务,有向超边对应于任务结点间的先后依赖关系,任意超边的尾端结点所对应任务的全部前驱任务都包含在该超边的源端子集中。相比赋权有向图和赋权无向超图,赋权有向超图为依赖任务的调度提供了更为精确的模型,能全面地表示云计算环境的异构性、分布性、广域性等特点,从而提高任务调度的准确性和执行效率。

[0015] 相比赋权有向图而言,赋权有向超图为任务结点的依赖关系提供了更为精确的模型:每条边仅连接两个结点,对应于任务结点之间的依赖关系只能连接两个任务,而每条超边可以连接两个以上的结点,对应于任务结点之间的先后依赖关系可以连接两个以上的任务,即任意超边的尾端结点所对应任务的全部(两个以上)前驱任务都包含在该超边的源端子集中。

[0016] 相比赋权无向超图而言,赋权有向超图为任务结点的先后依赖关系提供了更为精确的模型:无向超边连接两个以上的结点,但无法表示任务结点间的先后依赖关系,而每条有向超边可以连接两个以上的结点,超边尾端结点的所有直接前驱结点都包含在该超边的源端子集中,对应于任务结点间的先后依赖关系。

[0017] 本发明的目的在于针对已有技术存在的不足,提供一种基于多水平划分法和赋权有向超图的云计算任务调度方法,有效地缩短了任务完成的时间跨度,实现了云计算资源的合理利用,为云计算提供高效的任务调度机制。为达到上述目的,本发明的构思如下。

[0018] 一、在云计算环境中将大量分散的任务按照特定的约束条件划分为多个规模较小的任务子集,使划分后的各个任务子集计算规模相当,但任务子集之间的依赖关系数目达到最少,从而最大程度地实现云计算平台的负载均衡,并缩短整个任务完成的时间跨度。任务子集的优化划分作为云计算任务调度机制的关键环节,其结果对整个云计算环境的运行效率有着重要的影响,可有效地减少资源空闲时间,提高资源的利用效益。

[0019] 二、云计算任务调度的任务子集的优化划分问题转换为赋权有向超图划分问题,即任务子集的优化划分问题要求每个任务子集所包含的任务数目相等,对应于赋权有向超图划分问题的平衡约束条件,划分结果使得任务子集之间的依赖关系数目达到最小,对应于赋权有向超图划分问题的最小化总割切。其中,赋权有向超图的划分要求每个任务子集所包含的任务数目相等,对应于多资源约束条件下的平衡约束条件,并且划分结果使得这些任务子集之间的依赖关系数目达到最小,对应于多目标优化划分问题的最小化总割切。

[0020] 三、采用赋权有向超图描述任务的资源需求及依赖关系,并生成相应的赋权有向超图文件;然后启动基于多水平划分法的赋权有向超图划分程序,对生成的赋权有向超图

进行划分;最后依据赋权有向超图的划分结果构造任务子集,通过 MapReduce 任务调度模型对其进行映射和调度。

[0021] 四、在多水平划分法的粗化阶段,通过对赋权有向超图中所有结点进行基于结点属性函数的核值求解排序,进而基于结点核值的非严格降序访问处于未匹配状态的结点,依据一定规则对其进行匹配,从而将连接性好的结点合并在一起,为多水平划分的后续阶段提供更优的粗化赋权有向超图。

[0022] 五、在多水平划分法的优化阶段,采用多目标的离散群智能搜索程序改进贪心原则的局部搜索方法,对在每一水平层粗化赋权有向超图投影的划分进行优化,其中群智能粒子所处  $|V|$  维空间位置 ( $V$  代表赋权有向超图中结点集合) 对应于一个划分方案,即群智能粒子在每个维度空间的位置,代表该维度空间对应结点所处划分的结点子集;伴随着优化阶段的细化过程,每个群智能粒子解决方案代表的划分投影到当前水平层的细化赋权有向超图上,且群智能粒子的自由度随着空间维度  $|V|$  的增加而增大;群智能粒子之间进行直接通信或者间接通信,利用它们聚集协同表现出的智能行为,有效地搜索多约束条件下、多目标组合的 Pareto 有效解,使非劣迁移优化解向着 Pareto- 最优面逼近,增强迁移优化算法的多目标搜索能力。

[0023] 根据上述的发明构思,本发明的技术方案是这样实现的:一种基于多水平划分法和赋权有向超图的云计算任务调度方法,其特征在于,具体步骤如下。

[0024] 步骤 1, 类型类度分析, 输入云计算环境下用户提交的任务, 并对其进行类型和类度的分析, 确定任务的并行化程度及特点。

[0025] 步骤 2, 进程粒度分解, 根据用户任务的并行化程度及特点, 以及云计算的资源共享分配方式等独特性质, 对用户任务按照进程粒度级别进行分解。

[0026] 步骤 3, 资源特性分析, 根据云计算的资源共享分配方式等独特性质, 对分解后的任务进行资源特性分析。

[0027] 步骤 4, 赋权有向超图文件生成, 依据对任务资源特性的分析结果, 建立描述其资源需求及依赖关系的赋权有向超图模型, 并按照改进压缩的文件存储格式保存为赋权有向超图文件。

[0028] 步骤 5, 赋权有向超图划分, 启动基于多水平划分法的赋权有向超图划分程序, 读取赋权有向超图文件, 采用改进压缩的内存存储格式对赋权有向超图进行存储, 对生成的赋权有向超图进行划分, 将最终得到的划分结果存储在赋权有向超图划分文件中。

[0029] 步骤 6, 任务子集构造, 在检测到基于多水平划分法的赋权有向超图划分程序完成划分之后, 从赋权有向超图划分文件中读取相应的划分结果, 依据赋权有向超图的划分结果构造进程级任务子集。

[0030] 步骤 7, 任务映射调度, 通过 MapReduce 任务调度模型, 对基于赋权有向超图优化划分构造的任务子集进行映射和调度, 实现在云计算环境中的任务提交与执行, 有效地均衡云计算平台的负载和缩短整个任务完成的时间跨度。

[0031] 上述的步骤 4 中, 所述的赋权有向超图的改进压缩的文件存储格式的如下。

[0032] 步骤 4.1, 文件格式的第 1 行第 1 个参数代表着有向赋权超边的数目  $m$ , 第 2 个参数代表着赋权结点的数目  $n$ 。

[0033] 步骤 4.2, 文件格式的第 2 行开始到第  $m+1$  行的每行代表着一一条有向赋权超边的相

关信息,第1个数值为有向赋权超边的权值信息,其余数值为有向赋权超边的结点信息,其中每行的最后一个数值代表有向赋权超边的尾端结点信息,且有向赋权超边的源端结点信息处于有向赋权超边的权值信息和尾端结点信息之间。

[0034] 步骤4.3,文件格式的第 $m+2$ 行开始到第 $m+n+1$ 行的每行代表着一个赋权结点的权值信息。

[0035] 上述的步骤5中,所述的基于多水平划分法的赋权有向超图划分程序的步骤如下。

[0036] 步骤5.1,读取赋权有向超图文件,采用改进压缩的内存存储格式对赋权有向超图进行存储。

[0037] 步骤5.2,进入到多水平划分法的粗化阶段,采用赋权有向超图的结点匹配程序将当前水平层粗化赋权有向超图的某些结点结合在一起,得到下一水平层的粗化赋权有向超图,重复此过程直到粗化赋权有向超图足够小为止,即得到一个最小赋权有向超图。

[0038] 步骤5.3,进入到多水平划分法的初始划分阶段,运行基于FM方法的赋权有向超图划分程序,得到最小赋权有向超图的初始划分。

[0039] 步骤5.4,多水平划分法优化阶段的离散群智能搜索程序初始化,初始化全体群智能粒子的全局历史最优位置向量和每个群智能粒子的自身位置向量、自身速度向量、自身历史最优位置向量。

[0040] 步骤5.5,进入到多水平划分法的优化阶段,从最小赋权有向超图投影回初始赋权有向超图,在每一水平层的细化赋权有向超图中,采用离散群智能搜索程序对细化赋权有向超图的投影划分进行优化。

[0041] 步骤5.6,进入到平衡阶段,运行基于FM-EE方法的赋权有向超图划分程序。由于在基于多水平划分法的赋权有向超图划分过程中,可能违背赋权有向超图划分问题的平衡约束条件,因此在基于多水平划分法的赋权有向超图划分所求解的基础上,运行基于FM-EE方法的赋权有向超图划分方法,使划分解满足平衡约束条件,从而得到赋权有向超图划分问题的划分解。

[0042] 步骤5.7,将最终得到的赋权有向超图划分结果存储在赋权有向超图划分文件中。

[0043] 上述的步骤5.1中,所述的赋权有向超图的改进压缩的内存存储格式如下。

[0044] 步骤5.1.1,使用vwgts数组存储赋权有向超图中结点的权值信息,且vwgts数组的大小为赋权有向超图中的结点个数。

[0045] 步骤5.1.2,使用xadj数组存储每个结点所有邻接有向超边列表的起始位置信息,即第 $i$ 个结点的终止位置为第 $i+1$ 个结点的起始位置减1,且xadj数组的大小为赋权有向超图中的结点个数加1,xadj数组最后一个元素用于存放最后一个结点的终止位置。

[0046] 步骤5.1.3,使用adjncy数组存储每个结点所有邻接有向超边的列表信息,第 $i$ 个结点的邻接有向超边列表存储在adjncy数组中,从adjncy[xadj[i]]到adjncy[xadj[i+1]-1]。

[0047] 步骤5.1.4,使用eptr数组存储每条有向超边所包含的结点列表的起始位置信息,即第 $j$ 条有向超边的终止位置为第 $j+1$ 条有向超边的起始位置减1,且eptr数组的大小为赋权有向超图中的有向超边条数加1,eptr数组最后一个元素用于存放最后一条有向超边的终止位置。

[0048] 步骤 5.1.5, 使用 `eind` 数组存储每条有向超边所包含结点的列表信息, 其中每条有向超边的尾端结点只有 1 个, 且每条有向超边尾端结点的所有直接前驱结点都包含在该有向超边的源端子集中。第  $j$  条有向超边的结点列表存储在 `eind` 数组中, 从 `eind[eptr[j]]` 到 `eind[eptr[j+1]-1]`, 其中第  $j$  条有向超边的源端结点为 `eind[eptr[j]]` 到 `eind[eptr[j+1]-2]`, 第  $j$  条有向超边的尾端结点为 `eind[eptr[j+1]-1]`。

[0049] 步骤 5.1.6, 使用 `hewgts` 数组存储有向超边的权值信息, 且 `hewgts` 数组的大小为赋权有向超图中的有向超边数目。

[0050] 上述的步骤 5.2 中, 所述的当前水平层粗化赋权有向超图的结点匹配程序的步骤如下。

[0051] 步骤 5.2.1, 标注当前水平层粗化赋权有向超图中所有结点处于未匹配状态。

[0052] 步骤 5.2.2, 运行赋权有向超图的结点核值计算程序, 基于结点属性函数值进行当前水平层粗化赋权有向超图中所有结点的核值求解, 并按照结点的核值进行非严格降序排序。

[0053] 步骤 5.2.3, 按照结点核值的非严格降序访问处于未匹配状态的结点  $v$ ; 如果结点  $v$  还有邻接结点未匹配, 那么选取处于没有匹配状态的且权值最大的邻接超边的邻接结点  $u$  和结点  $v$  匹配, 且标注这两个结点为匹配状态; 如果结点  $v$  所有邻接匹配结点处于匹配状态, 那么结点  $v$  仍处于未匹配状态, 在粗化过程中直接拷贝到下一水平层的粗化赋权有向超图中。

[0054] 步骤 5.2.4, 步骤重复上一步, 直至所有结点访问结束。

[0055] 上述的步骤 5.2.2 中, 所述的赋权有向超图的结点核值计算程序的步骤如下。

[0056] 步骤 5.2.2.1, 计算出所有结点的属性函数值。

[0057] 步骤 5.2.2.2, 对所有结点的属性函数值进行非严格降序排序。

[0058] 步骤 5.2.2.3, 按照结点属性函数值的非严格降序次序访问每个结点, 计算每个结点的核值。

[0059] 上述的步骤 5.2.2.2 中, 所述的结点属性函数值的非严格降序排序的步骤如下。

[0060] 步骤 5.2.2.2.1, 根据结点的属性函数值属于在一定范围内的整数, 扫描所有结点的属性函数值, 统计每一种属性函数值的结点个数, 存储在计数辅助数组 `bin` 中。

[0061] 步骤 5.2.2.2.2, 针对每一种属性函数值, 借助计数辅助数组 `bin`, 计算出在所有结点的属性函数值中, 小于该属性函数值的结点个数, 存储在位置辅助数组 `pos` 中。

[0062] 步骤 5.2.2.2.3, 扫描所有结点的属性函数值, 针对每一个结点的属性函数值, 借助位置辅助数组 `pos`, 得到该结点的属性函数值在非严格降序排序的次序, 并将该次序存储在次序辅助数组 `vert` 中。

[0063] 上述的步骤 5.2.2.3 中, 所述的结点  $v$  的核值计算的步骤如下。

[0064] 步骤 5.2.2.3.1, 将结点  $v$  的属性函数值作为核值输出。

[0065] 步骤 5.2.2.3.2, 标记结点  $v$  从所在的超边  $e$  中删除。

[0066] 步骤 5.2.2.3.3, 如果超边  $e$  删除结点  $v$  后, 仍包含两个及以上未被标记删除的结点, 则超边  $e$  仍然存在, 否则删除超边  $e$ 。

[0067] 步骤 5.2.2.3.4, 重新计算结点  $v$  的邻接结点  $u$  的属性函数值。

[0068] 步骤 5.2.2.3.5, 如果邻接结点  $u$  的属性函数值大于结点  $v$  的属性函数值, 更新邻

接结点  $u$  的属性函数值, 并且借助计数辅助数组  $\text{bin}$ 、位置辅助数组  $\text{pos}$  和次序辅助数组  $\text{vert}$  的信息, 快速更新邻接结点  $u$  在所有结点的属性函数值非严格降序排序的次序; 否则不更新邻接结点  $u$  的属性函数值及其排序的次序。

[0069] 上述的步骤 5.4 中, 所述的离散群智能搜索程序的初始化步骤如下。

[0070] 步骤 5.4.1, 初始化全体群智能粒子的全局历史最优位置向量, 设定全局历史最优位置向量为最小赋权有向超图的初始划分。

[0071] 步骤 5.4.2, 初始化每个群智能粒子的自身位置向量, 遍历每个群智能粒子并设定每个群智能粒子的自身位置向量为最小赋权有向超图的初始划分, 即群智能粒子在最小赋权有向超图的每个维度空间的位置。群智能粒子在最小赋权有向超图的某维度空间的位置, 代表着该维度空间对应的最小赋权有向超图结点所处划分的结点子集。

[0072] 步骤 5.4.3, 初始化每个群智能粒子的自身速度向量, 遍历每个群智能粒子并在速度阈值范围内, 随机给定每个群智能粒子在每个维度空间的速度向量。

[0073] 步骤 5.4.4, 初始化每个群智能粒子的自身历史最优位置向量, 遍历每个群智能粒子并设定每个群智能粒子的自身位置向量为自身历史最优位置向量。

[0074] 上述的步骤 5.5 中, 所述的离散群智能搜索程序的步骤如下。

[0075] 步骤 5.5.1, 投影每个群智能粒子的自身位置向量, 遍历每个群智能粒子并依据每个群智能粒子的上一水平层粗化赋权有向超图的自身位置向量投影到当前水平层的细化赋权有向超图上, 得到每个群智能粒子在当前水平层细化赋权有向超图的自身位置向量, 即群智能粒子在当前水平层的细化赋权有向超图的每个维度空间的位置。群智能粒子在当前水平层的细化赋权有向超图的某维度空间的位置, 代表着该维度空间对应的当前水平层的细化赋权有向超图结点所处划分的结点子集。

[0076] 步骤 5.5.2, 投影每个群智能粒子的自身速度向量, 遍历每个群智能粒子并依据每个群智能粒子的上一水平层粗化赋权有向超图的自身速度向量投影到当前水平层的细化赋权有向超图上, 得到每个群智能粒子在当前水平层细化赋权有向超图的自身速度向量, 即群智能粒子在当前水平层的细化赋权有向超图的每个维度空间的速度。

[0077] 步骤 5.5.3, 计算每个群智能粒子的二维辅助数组  $\text{EDG}[n][m]$ , 遍历每个群智能粒子并依据每个群智能粒子的当前水平层细化赋权有向超图的自身位置向量, 计算每个群智能粒子的二维辅助数组  $\text{EDG}[n][m]$ 。

[0078] 步骤 5.5.4, 快速计算每个群智能粒子在当前水平层的细化赋权有向超图的自身位置向量的割切值, 遍历每个群智能粒子并依据每个群智能粒子的二维辅助数组  $\text{EDG}[n][m]$ , 快速计算每个群智能粒子在当前水平层的细化赋权有向超图的自身位置向量的割切值。

[0079] 步骤 5.5.5, 循环初始化, 初始化循环计数器  $\text{COUNT}$  为 0。

[0080] 步骤 5.5.6, 更新每个群智能粒子的自身速度向量, 遍历每个群智能粒子并依据每个群智能粒子自身的历史最优位置向量、全体群智能粒子的全局历史最优位置向量, 在速度阈值范围内更新自身速度向量。

[0081] 步骤 5.5.7, 更新每个群智能粒子的自身位置向量和二维辅助数组  $\text{EDG}[n][m]$ , 遍历每个群智能粒子并计算每个群智能粒子自身位置向量和自身速度向量的向量和, 更新自身位置向量, 进而依据群智能粒子的自身位置向量计算群智能粒子的二维辅助数组  $\text{EDG}[n]$

[m]。

[0082] 步骤 5.5.8, 快速计算每个群智能粒子的自身位置向量的割切值, 遍历每个群智能粒子并依据每个群智能粒子的自身位置向量和二维辅助数组  $EDG[n][m]$ , 快速计算每个群智能粒子的当前水平层细化赋权有向超图的自身位置向量的割切值。如果该群智能粒子的自身位置向量的割切值小于自身历史最优位置向量的割切值, 则更新该群智能粒子的历史最优位置向量为当前的自身位置向量。如果该群智能粒子的自身位置向量的割切值小于全体群智能粒子的全局历史最优位置向量的割切值, 则更新全体群智能粒子的全局历史最优位置向量为该群智能粒子当前的自身位置向量。

[0083] 步骤 5.5.9, 重复步骤 5.5.6、5.5.7、5.5.8, 直至循环计数器 COUNT 到达给定的上限。

[0084] 上述的步骤 5.5.3 和步骤 5.5.7 中, 所述的计算群智能粒子的二维辅助数组  $EDG[n][m]$  的步骤如下。

[0085] 步骤 5.5.3.1, 二维辅助数组  $EDG[n][m]$  清零。

[0086] 步骤 5.5.3.2, 读取  $eptr$  数组和  $eind$  数组存储的每条超边所包含的结点信息, 基于群智能粒子的当前水平层细化赋权有向超图的自身位置向量, 计算每条超边在  $n$  个划分子集  $V_1 \cdots V_n$  的结点个数, 即二维辅助数组  $EDG[n][m]$  的两行分别存放  $m$  条超边在  $n$  个划分子集的结点个数。

[0087] 上述的步骤 5.5.4 和步骤 5.5.8 中, 所述的快速计算群智能粒子的自身位置向量的割切值的步骤如下。

[0088] 步骤 5.5.4.1, 划分割切值清零。

[0089] 步骤 5.5.4.2, 遍历每条超边是否结束, 如果访问未结束, 即存在超边  $e$  未被访问, 则转步骤 5.5.4.3; 否则访问结束, 返回划分割切值。

[0090] 步骤 5.5.4.3, 如果满足  $EDG[i][e] \geq 1$  的条件 1 和  $EDG[j][e] \geq 1$  的条件 2 时, 意味着超边  $e$  在划分子集  $V_i$  和  $V_j$  的结点个数都大于等于 1, 即可判定超边  $e$  是两栖边, 并将划分割切值累加上当前超边的权值; 否则判定超边  $e$  不是两栖边, 划分割切值不变。

[0091] 步骤 5.5.4.4, 转步骤 5.5.4.2。

[0092] 上述的步骤 5.5.6 中, 所述的群智能粒子的自身速度向量的更新步骤如下。

[0093] 步骤 5.5.6.1, 计算出群智能粒子的当前自身速度向量与惯性权重参数的乘积。

[0094] 步骤 5.5.6.2, 计算出群智能粒子的历史最优位置向量和群智能粒子的当前自身位置向量的向量差, 再和认知参数相乘。

[0095] 步骤 5.5.6.3, 计算出全体群智能粒子的全局历史最优位置向量和群智能粒子的当前自身位置向量的向量差, 再和社会参数相乘。

[0096] 步骤 5.5.6.4, 计算出步骤 5.5.6.1、步骤 5.5.6.2、步骤 5.5.6.3 得到的三个乘积的向量和, 更新群智能粒子的自身速度向量。

[0097] 本发明与现有技术相比较, 具有如下显而易见的突出实质性特点和显著优点。

[0098] 1、提高了任务调度的效率。

[0099] 本发明基于多水平划分法和赋权有向超图的云计算任务调度方法, 一方面通过任务到赋权有向超图文件的转换, 启动基于多水平划分法的赋权有向超图划分程序, 对生成的赋权有向超图进行划分, 避免了划分方法直接在任务上进行划分; 另一方面可以通过设

置划分方法中的惯性权重参数、认知参数和社会参数来获取较优的划分结果后,再进行任务的映射和调度,从而有效地提高了任务调度的效率,缩短了任务完成的时间跨度,实现了云计算资源的合理利用,为云计算提供高效的任务调度机制。

[0100] 2、提高了任务划分的性能。

[0101] 本发明基于多水平划分法和赋权有向超图的任务优化划分方法,借助二维辅助数组  $EDG[n][m]$  存储每条超边在划分子集的结点数,实现了划分割切值的计算方法,有效地避免了遍历超边中的结点。相比赋权无向超图划分方法和赋权有向图划分方法,该方法有效地找到比现有技术更优的任务划分结果,降低了空间复杂度和时间复杂度,最终显著地提高了任务优化划分的性能。

[0102] 通过以下对本发明云计算环境下的基于多水平划分法和赋权有向超图的云计算任务调度方法的实例结合其附图的描述,可以进一步理解本发明的目的、具体结构特征和优点。

[0103] 图 1 是本发明云计算环境下的基于多水平划分法和赋权有向超图的云计算任务调度方法的流程图。

[0104] 图 2 是本发明的赋权有向超图的改进压缩的内存存储格式。

[0105] 图 3 是本发明基于多水平划分法和赋权有向超图的云计算任务划分过程中优化阶段的离散群智能搜索程序的流程图。

[0106] 具体实施方式。

[0107] 为了能够更清楚地理解本发明云计算环境下的基于多水平划分法和赋权有向超图的云计算任务调度方法的技术内容,特举以下实例详细说明。

[0108] 本实施例的基于多水平划分法和赋权有向超图的云计算任务调度方法的流程图如图 1 所示。在云计算环境下,输入用户提交的任务 101,对用户任务进行类型和类度的分析 102,确定任务的并行化程度和特点;根据用户任务的并行化程度和特点,以及云计算的资源共享分配方式等独特性质,对用户任务按照进程粒度级别进行分解 103;进而对分解后的任务进行资源特性分析 104;依据对任务资源特性的分析结果,建立描述其资源需求及依赖关系的赋权有向超图模型 105;按照改进压缩的文件存储格式保存为赋权有向超图文件 109;启动基于多水平划分法的赋权有向超图划分程序 115,读取赋权有向超图文件 109,得到采用改进压缩的内存存储格式的赋权有向超图 111;进入到多水平划分法的粗化阶段,调用赋权有向超图的结点匹配程序 116,将某些赋权有向超图的结点结合在一起,得到每一水平层的粗化赋权有向超图 112;进入到多水平划分法的初始划分阶段,调用最小赋权有向超图的初始划分程序 117 对最小赋权有向超图进行划分,得到最小赋权有向超图的初始划分 113;进入到多水平划分法的投影优化阶段,从最小赋权有向超图投影回初始赋权有向超图,在每一水平层的细化赋权有向超图中,调用细化赋权有向超图的迁移划分优化程序 118 对划分进行优化,得到每一水平层细化赋权有向超图的近似非劣最优划分 114;进入到平衡阶段,运行基于 FM-EE 方法的赋权有向超图划分程序 115,使初始赋权有向超图的划分结果满足平衡约束条件,并将最终得到的划分结果存储为赋权有向超图划分文件 110;在检测到基于多水平划分法的赋权有向超图划分程序 115 完成划分之后,从赋权有向超图划分文件 110 中读取相应的划分结果,依据赋权有向超图的划分结果构造进程级任务子集 106;通过 MapReduce 任务调度模型,对基于赋权有向超图优化划分构造的任务子集

进行映射和调度 107 ;在云计算环境中,对基于赋权有向超图优化划分构造的任务子集中的任务提交与执行 108,有效地均衡云计算平台的负载并缩短整个任务完成的时间跨度。

[0109] 本实施例的赋权有向超图改进压缩的文件存储格式参见在先技术 [1] “G. Karypis and V. Kumar. HMetis 1.5.3: A Hypergraph Partitioning Package [R]. Technical report, Department of Computer Science, University of Minnesota, 1998.”和在先技术 [2] “孙凌宇,冷明,郭恺强,朱平. 一种 VLSI 设计到超图的转换系统[J]. 计算机工程与应用, 2012, Vol. 29, Issue. 2, Pages 7-16.”。与在先技术 [1,2] 相同点:文件格式的第 1 行第 1 个参数代表着有向赋权超边的数目  $m$ ,第 2 个参数代表着赋权结点的数目  $n$ ;文件格式的第 2 行开始到第  $m+1$  行的每行代表着一一条有向赋权超边的相关信息;文件格式的第  $m+2$  行开始到第  $m+n+1$  行的每行代表着一个赋权结点的权值信息。与在先技术 [1,2] 区别点:文件格式的第 2 行开始到第  $m+1$  行中,除第 1 个数值之外的其余数值为有向赋权超边的结点信息,其中每行的最后一个数值代表有向赋权超边的尾端结点信息,且有向赋权超边的源端结点信息处于有向赋权超边的权值信息和尾端结点信息之间。

[0110] 本实施例的赋权有向超图的改进压缩的内存存储格式如图 2 所示。存储结构使用 `adjncy` 数组 204 存储每个结点所有邻接有向超边的列表信息。使用 `xadj` 数组 203 存储每个结点所有邻接有向超边列表的起始位置信息,即第  $i$  个结点的终止位置为第  $i+1$  个结点的起始位置减 1,且 `xadj` 数组 203 的大小为赋权有向超图中的结点个数加 1, `xadj` 数组 203 最后一个元素用于存放最后一个结点的终止位置。使用 `eind` 数组 207 存储每条有向超边所包含结点的列表信息。使用 `eptr` 数组 206 存储每条有向超边所包含的结点列表的起始位置信息,即第  $j$  条有向超边的终止位置为第  $j+1$  条有向超边的起始位置减 1,且 `eptr` 数组 206 的大小为赋权有向超图中的有向超边条数加 1, `eptr` 数组 206 最后一个元素用于存放最后一条有向超边的终止位置。使用 `vwgts` 数组 202 存储结点的权值信息,且 `vwgts` 数组 202 的大小为赋权有向超图中的结点个数。使用 `hewgts` 数组 205 存储有向超边的权值信息,且 `hewgts` 数组 205 的大小为赋权有向超图中的有向超边条数。假设数组地址从零开始,结点编号从零开始,则第  $i$  个结点的邻接有向超边列表存储在 `adjncy` 数组 204 中,从 `adjncy[xadj[i]]` 到 `adjncy[xadj[i+1]-1]`;第  $j$  条有向超边的邻接结点列表存储在 `eind` 数组 207 中,从 `eind[eptr[j]]` 到 `eind[eptr[j+1]-1]`,其中第  $j$  条有向超边的源端结点为 `eind[eptr[j]]` 到 `eind[eptr[j+1]-2]`,第  $j$  条有向超边的尾端结点为 `eind[eptr[j+1]-1]`。图例 201 包含总共 7 个结点和 8 条有向超边,其中第 6 个结点的权值为 7,有 2 条邻接有向超边  $f$ 、 $h$ ,其中有向超边  $f$  对应的权值为 4,且相应的邻接结点分别为结点 7、3、6,源端结点为结点 7 和 3,尾端结点为结点 6;有向超边  $h$  对应的权值为 1,且相应的邻接结点分别为结点 4、6,源端结点为结点 4,尾端结点为结点 6。

[0111] 本实施例的赋权有向超图的结点核值计算程序参见在先技术 [3] “孙凌宇,冷明,冷子阳. 云计算环境中的基于结点属性函数的任务核值计算方法 [P]. 2014. 发明专利. 2014 年 4 月 6 日提交申请”。

[0112] 本实施例的基于 FM 方法的赋权有向超图划分程序参见在先技术 [4] “Fiduccia C, Mattheyses R. A linear-time heuristics for improving network partitions[C]. Proceedings of the 19th Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 1982, Pages 175-181.”。



[0113] 本实施例的基于 FM-EE 方法的赋权有向超图划分程序参见在先技术 [5] “Karypis G, Aggarwal R, Kumar V, Shekhar S. Multilevel hypergraph partitioning: Applications in VLSI domain[J]. IEEE transactions on very large scale integration systems, 1999, Vol.7, Issue.1, Pages 69-79.”。

[0114] 本实施例的基于多水平划分法的赋权有向超图划分过程中优化阶段的离散群智能搜索程序的流程图如图 3 所示,步骤如下。

[0115] A01:初始化全体群智能粒子的全局历史最优位置向量,设定全局历史最优位置向量为最小赋权有向超图的初始划分。

[0116] A02:初始化每个群智能粒子的自身位置向量,遍历每个群智能粒子并设定每个群智能粒子的自身位置向量为最小赋权有向超图的初始划分,即群智能粒子在最小赋权有向超图的每个维度空间的位置。群智能粒子在最小赋权有向超图的某维度空间的位置,代表着该维度空间对应的最小赋权有向超图结点所处划分的结点子集。

[0117] A03:初始化每个群智能粒子的自身速度向量,遍历每个群智能粒子并在速度阈值范围内,随机给定每个群智能粒子在每个维度空间的速度向量。

[0118] A04:初始化每个群智能粒子的自身历史最优位置向量,遍历每个群智能粒子并设定每个群智能粒子的自身位置向量为自身历史最优位置向量。

[0119] A05:投影每个群智能粒子的自身位置向量,遍历每个群智能粒子并依据每个群智能粒子的上一水平层粗化赋权有向超图的自身位置向量投影到当前水平层的细化赋权有向超图上,得到每个群智能粒子在当前水平层细化赋权有向超图的自身位置向量,即群智能粒子在当前水平层的细化赋权有向超图的每个维度空间的位置。群智能粒子在当前水平层的细化赋权有向超图的某维度空间的位置,代表着该维度空间对应的当前水平层的细化赋权有向超图结点所处划分的结点子集。

[0120] A06:投影每个群智能粒子的自身速度向量,遍历每个群智能粒子并依据每个群智能粒子的上一水平层粗化赋权有向超图的自身速度向量投影到当前水平层的细化赋权有向超图上,得到每个群智能粒子在当前水平层细化赋权有向超图的自身速度向量,即群智能粒子在当前水平层的细化赋权有向超图的每个维度空间的速度。

[0121] A07:计算每个群智能粒子的二维辅助数组  $EDG[n][m]$ ,遍历每个群智能粒子并依据每个群智能粒子的当前水平层细化赋权有向超图的自身位置向量,计算每个群智能粒子的二维辅助数组  $EDG[n][m]$ 。

[0122] A08:快速计算每个群智能粒子在当前水平层的细化赋权有向超图的自身位置向量的割切值,遍历每个群智能粒子并依据每个群智能粒子的二维辅助数组  $EDG[n][m]$ ,快速计算每个群智能粒子在当前水平层的细化赋权有向超图的自身位置向量的割切值。

[0123] A09:循环初始化,初始化循环计数器 COUNT 为 0。

[0124] A10:更新每个群智能粒子的自身速度向量,遍历每个群智能粒子并依据每个群智能粒子自身的历史最优位置向量、全体群智能粒子的全局历史最优位置向量,在速度阈值范围内更新自身速度向量。

[0125] A11:更新每个群智能粒子的自身位置向量和二维辅助数组  $EDG[n][m]$ ,遍历每个群智能粒子并计算每个群智能粒子自身位置向量和自身速度向量的向量和,更新自身位置向量,进而依据群智能粒子的自身位置向量计算群智能粒子的二维辅助数组  $EDG[n][m]$ 。

[0126] A12:快速计算每个群智能粒子的自身位置向量的割切值,遍历每个群智能粒子并依据每个群智能粒子的自身位置向量和二维辅助数组  $EDG[n][m]$ ,快速计算每个群智能粒子的当前水平层细化赋权有向超图的自身位置向量的割切值。如果该群智能粒子的自身位置向量的割切值小于自身历史最优位置向量的割切值,则更新该群智能粒子的历史最优位置向量为当前的自身位置向量。如果该群智能粒子的自身位置向量的割切值小于全体群智能粒子的全局历史最优位置向量的割切值,则更新全体群智能粒子的全局历史最优位置向量为该群智能粒子当前的自身位置向量。

[0127] A13:重复步骤 A10、A11、A12,循环计数器 COUNT 加 1,直至循环计数器 COUNT 到达给定的上限。

[0128] A14:投影到上一水平层的粗化赋权有向超图。

[0129] A15:重复步骤 A05、A06、A07、A08、A09、A10、A11、A12、A13、A14,直至投影回到初始赋权有向超图。



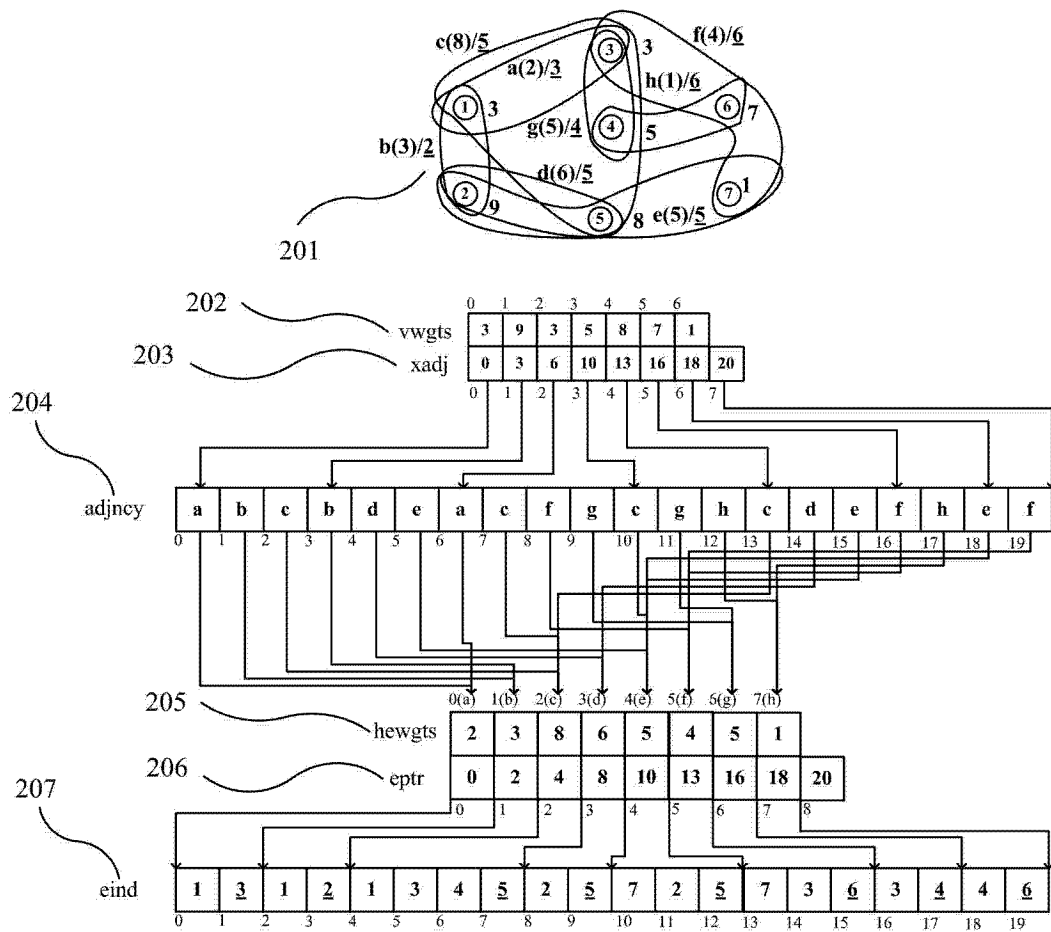


图 2

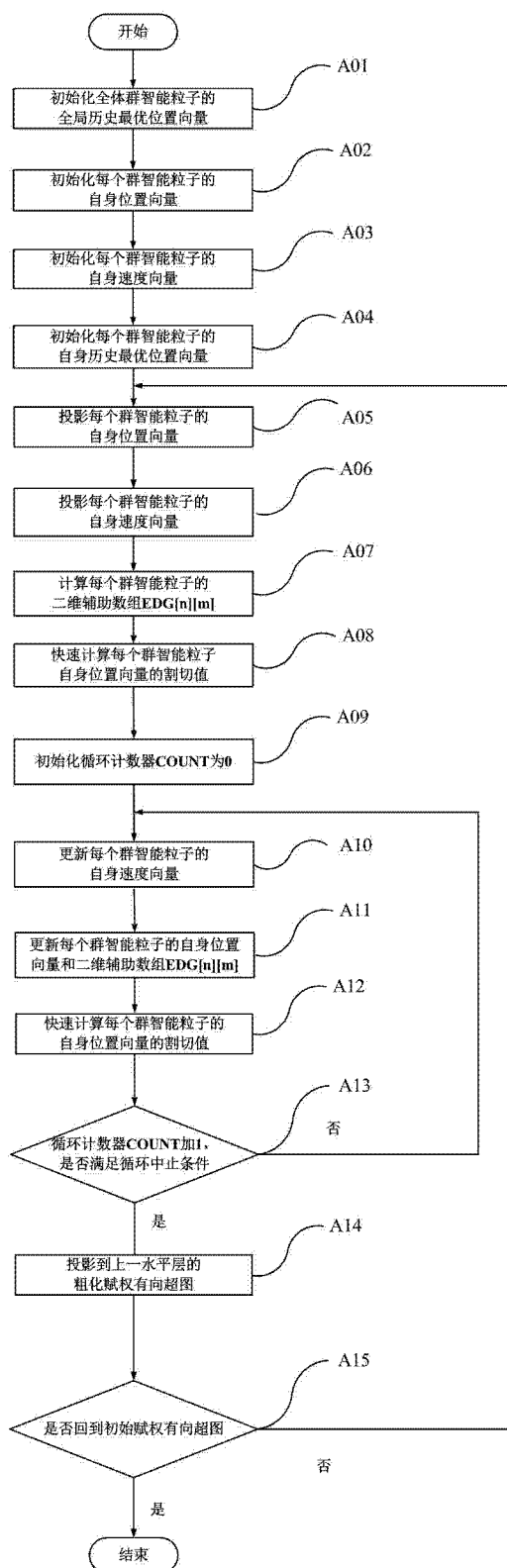


图 3