

云计算环境下的大规模图状数据处理任务调度算法

李健, 黄庆佳, 刘一阳, 苏森

(北京邮电大学网络与交换技术国家重点实验室, 10087, 北京)

摘要: 针对云计算环境下调度算法必须考虑资源租赁成本的问题, 提出一种新的基于粒子群优化的大规模图状数据处理任务调度算法(LGPPSO). 首先, 该算法将图状数据处理任务调度方案编码为粒子群中粒子的位置, 并利用任务的调度长度和资源租赁成本建立适应度函数来评价当前粒子的优劣程度, 然后重新定义粒子群的参数和相关操作, 最后在算法的每一次迭代过程中, 粒子不断更新自身的速度和位置, 以获得任务调度的近似最优解. 模拟实验结果表明: 在仅以调度长度为目标时, LGPPSO 算法的调度长度比异构最早完成时间任务调度算法(HEFT)平均降低约 12.3%; 在以调度长度和资源租赁成本为目标时, 与成本感知任务调度算法(CCSH)相比, 在资源租赁成本基本一致的情况下, LGPPSO 算法的调度长度平均降低约 9.97%.

关键词: 大规模图状数据处理; 调度算法; 粒子群优化; 云计算

中图分类号: TP311 **文献标志码:** A **文章编号:** 0253-987X(2012)12-0116-07

A Task Scheduling Algorithm for Large Graph Processing Cloud in Computing

LI Jian, HUANG Qingjia, LIU Yiyang, SU Sen

(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 10086, China)

Abstract: A new task scheduling algorithm for large graph processing based on particle swarm optimization (short for LGPPSO) is proposed to take the monetary cost in cloud computing into account. The schedule plan for large graph processing task is expressed as position of particles, and both the monetary cost and the schedule length are used in the fitness function. The parameters and operations of the particles in LGPPSO are then redefined. The velocity and position of particles are updated at each iteration to get a near-optimal solution. Simulation results show that the average schedule length of LGPPSO algorithm is reduced by about 12.3% compared to the heterogeneous earliest finish time algorithm, and is reduced by about 9.97% compared to the cost conscious scheduling heuristic algorithm with similar resource rental cost.

Keywords: large graph processing; scheduling algorithm; particle swarm optimization; cloud computing

近年来,随着互联网和 Web 2.0 技术的高速发展,许多应用都涉及图状数据处理,如交通路线图、科学文献引用关系图和社交网络等^[1-2]. 由于图状

数据规模的不断增大,对计算能力的需求远远超出本地数据中心的处理能力,这就需要不断加大本地数据中心的基础设施投入来扩展现有系统. 然而,

收稿日期: 2012-06-27. 作者简介: 李健(1988—),男,博士生; 苏森(通信作者),男,教授,博士生导师. 基金项目: 国家自然科学基金资助项目(61170274).

网络出版时间: 2012-09-23

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20120923.1734.007.html>

扩展本地数据中心存在资源利用率低、配置成本高、维护困难等问题,越来越多的公司选择在云计算环境下处理大型图状数据。云计算是分布式计算、网络计算和效用计算的进一步发展,是以网络为载体,以虚拟化技术为基础,根据用户需求动态配置资源的新兴计算模式^[3-4],其主要优势包括:海量图数据存储能力、强大的计算处理能力以及良好的可伸缩性和灵活性。如何利用云计算的优势,高效调度大规模图状数据处理任务是当前的研究热点问题。

一般来说,大规模图状数据处理任务采用有向无环图(directed acyclic graph, DAG)来表示,图中的节点表示一个处理任务,边表示节点之间的前驱后继关系。针对 DAG 模型最小化任务调度时间的问题在绝大多数情况下是 NP 完全问题^[5]。文献[6]设计了异构最早完成时间任务调度算法(HEFT),利用图中节点的计算负载和传输负载计算节点的优先级,并排序构建调度列表,根据优先级大小,将任务调度到最小化任务完成时间的机器执行。文献[7]设计了任务复制调度算法(DBUS),利用任务复制技术,根据处理器当前的空闲时间,重复映射任务图中的一些前驱任务,减少节点之间的传输开销。文献[8]提出支配队列任务调度算法(DSC),利用图中节点的计算负载和传输负载动态跟踪图的关键路径,将关键路径上的节点进行聚簇在同一处理器上执行。文献[9]设计并实现了 DAGMan 系统用以在网格环境下处理大规模图状数据,其主要调度思想是针对网格资源的动态加入和退出,建立可靠性模型,降低因资源动态退出所造成的性能损失。文献[10]针对微软处理图状数据的性能需求开发了 Dryad 系统,达到最大化系统吞吐量的目标。文献[11]首次在云计算环境下开发大规模图状数据处理系统,其调度思想主要满足处理任务的数据本地化需求。但是,以上这些方法均从大规模图状数据处理性能角度出发,设计启发式方法求解问题模型,而云计算资源即付即用的付费模式,使得资源的租赁成本成为调度算法必须考虑的因素。文献[12]针对此问题,提出成本感知任务调度算法(CCSH),建立云计算环境下大规模图状数据处理任务的调度模型,并设计了降低成本的任务调度算法,但该算法在降低资源租赁开销的同时延长了任务的调度长度,其求解质量仍存在较大的提升空间。

借鉴上述相关研究成果,本文提出一种新的基于粒子群优化的任务调度算法(large graph pro-

cessing based on particle swarm optimization, LG-PPSO),用于解决云计算环境下大规模图处理任务调度问题。该算法将图状数据处理任务调度方案编码为粒子群中粒子的位置,并利用图状数据处理的调度长度和资源租赁成本建立适应度函数来评价当前粒子的优劣程度。在每一次迭代中,粒子通过跟踪个体极值和全局极值来更新速度和位置,以获得图状数据处理任务调度的近似最优解。实验结果表明,与已有研究成果相比,本文所提出的算法,可显著降低调度长度和资源租赁成本。本文的主要贡献为:①根据云计算环境下大规模图状数据处理任务的调度模型重新定义粒子群的相关操作,缩短大规模图状数据处理任务调度长度的同时,降低资源租赁成本;②通过对随机生成图和真实应用图进行大规模实验测量,验证算法的通用性。

1 问题描述

1.1 大规模图状数据处理任务模型

借鉴微软的相关定义^[12],大型图状数据可以用有向无环图 DAG 来表示相互之间通信的任务集。图中顶点集 N 为大规模图状数据处理任务 n_i 的集合,有向边集 E 为图中任务之间边 e_{ij} 的集合,边上的权重值表示任务 n_i 和任务 n_j 不再同一资源上执行时的传输时间。如果这 2 个任务被调度到同一资源上执行,则传输时间为 0。图中没有任何前驱结点的任务标记为入口任务 n_e ,图中没有任何后继结点的任务标记为出口任务 n_l 。任务 n_i 的直接前驱任务集合表示为 $\text{pred}(n_i)$,直接后继任务集合表示为 $\text{succ}(n_i)$,任务 n_i 调度到虚拟机 m_k 上的执行时间表示为 $t(n_i, m_k)$ 。图中出口节点 n 的完成时间表示为 $t(n_l)$ 。因此,一个大规模图处理任务的调度长度 T 可定义为所有出口任务完成时间的最大值

$$T = \max\{t(n_l)\} \quad (1)$$

1.2 云计算资源计价模型

现有云计算供应商配置不同类别的虚拟机供用户使用。这些虚拟机彼此之间的处理能力和计价模式互不相同。例如:亚马逊^[13]目前针对计算密集型和内存密集型等应用提供 11 种不同类型的虚拟机,并根据用户租用的时长按照小时计费,但这种计费模式将导致用户租用资源不足 1 小时也将支付足额的费用,并不符合云计算即付即用的理念。本文选用 Google App Engine 的计价模式作为资源的计价模型^[14],即根据计算资源的处理能力的不同,按照使用时间进行计费。假设 c_b 是每个时间周期处理能

力最差虚拟机的计费, r_j 表示第 j 台虚拟机和速度最慢虚拟机的速度比率. 根据文献[15], 第 j 台虚拟机的计费可表示为

$$c_j = c_b \exp r_j \quad (2)$$

处理大规模图状数据所需要资源租赁成本 C 可以表示为所有执行图状数据处理任务的费用总和

$$C = \sum c_j \quad (3)$$

1.3 大规模图状数据处理任务调度问题描述

云计算环境下, 大规模图状数据处理任务调度问题是一个多目标是优化问题, 其主要目标最小化任务调度长度 T 和资源租赁成本 C . 对于一个大规模图状数据处理任务, 不同的调度方案将导致不同调度长度和资源租赁成本, 而这 2 个目标相互影响. 例如, 调度任务给处理能力强的虚拟机执行将减少运行时间, 但却增加资源租赁成本. 针对此多目标优化问题, 将存在一组帕累托(Pareto)最优解集^[16], 解集中任何解都不可能在降低一个目标的前提下, 提高另一个目标. 为有效均衡云计算环境下大规模图状数据处理任务的调度长度和资源租赁成本, 将下式作为该模型的目标函数

$$\min \left\{ \alpha \frac{T' - T_{\min}}{T_{\max} - T_{\min}} + (1 - \alpha) \frac{C' - C_{\min}}{C_{\max} - C_{\min}} \right\} \quad (4)$$

$\alpha \in [0, 1]$

式中: T' 表示当前策略的任务调度长度; C' 表示当前策略的资源租赁成本; $T_{\min(\max)}$ 表示当前解集中最小(大)的任务调度长度; $C_{\min(\max)}$ 表示当前解集中最小(大)的资源租赁成本; 系数 α 表示调度长度和资源租赁成本之间均衡关系, α 值变小将得到资源租赁成本较小但调度长度延长的调度方案.

2 LGPPSO 任务调度算法

2.1 PSO 相关参数和算子重定义

粒子群优化算法(particle swarm optimization, PSO)^[17]是由 Kennedy 和 Eberhart 提出的一种新的全局优化进化算法. 该算法首先随机初始化一群粒子, 每个粒子的位置对应于优化问题的一个解, 粒子的速度指导其在迭代过程中向最优解运动的方向. 粒子同时具有记忆功能, 可根据迭代过程中自身达到的最优值 $\mathbf{X}_p$ 和整个群体所到达过的最优值 $\mathbf{X}_g$ 来不断修正自己位置和速度, 从而形成群体寻优的进化. 由于 PSO 算法概念简单, 因此在解决连续域内多目标优化问题获得了成功的应用^[18-19], 但是在解决离散域内的问题时无法达到理想的效果, 需要针对问题特性重新定义粒子群中粒子的位置、速

度和相关操作.

定义 1 在 PSO 算法中, 粒子的位置对应于待解决问题的一个潜在解. 设在第 t 次迭代时粒子 i 的位置为

$$\mathbf{X}_i(t) = [x_{i1}(t), x_{ij}(t), \dots, x_{im}(t)] \quad (5)$$

式中: j 表示大规模图状数据的第 j 个任务; $x_{ij}(t)$ 表示第 j 个数据任务从资源列表中选择虚拟机编号.

定义 2 粒子群中每个粒子都由一个速度矢量决定其飞行方向和速率大小, 相应的飞行速度表示为

$$\mathbf{V}_i(t) = [v_{i1}(t), v_{i2}(t), \dots, v_{im}(t)] \quad (6)$$

式中: $v_{ij}(t)$ 定义为一个二进制变量, 如果取值为 0, 则表示第 j 个图状数据处理任务需要从云计算环境中重新选择虚拟机执行.

定义 3 减法 $\mathbf{X}_i - \mathbf{X}_j$ 表示 2 个大规模图状数据处理任务的差异性. 若当前 2 个调度方案在同一维度上具有相同的值, 则减法操作结果取 1, 否则取 0. 例如, $(6, 9, 1, 3, 2, 5, 7) - (7, 9, 2, 3, 5, 5, 7) = (0, 1, 0, 1, 0, 1, 1)$

定义 4 加法 $P_i \mathbf{V}_i \oplus P_j \mathbf{V}_j$ 表示粒子更新过程中, 通过跟踪自身最优位置和全局最优位置来更新自身的速度, 即在粒子的对应维度上以 P_i 的概率按照 $\mathbf{V}_i$ 各维度的值更新和以 P_j 的概率按照 $\mathbf{V}_j$ 各维度的值更新, 且 $P_i + P_j$ 的概率总和为 1. 例如, $0.1(0, 1, 1, 0, 1, 1, 0) \oplus 0.9(1, 1, 1, 1, 0, 1, 1) = (x, 1, 1, x, x, 1, x)$, 其中 x 表示此维度的值取 0 取 1 并不确定. 本例第一维中 x 表示此维度以 0.1 的概率取 0, 以 0.9 的概率取 1, 依此类推.

定义 5 乘法 $\mathbf{X}_i \otimes \mathbf{V}_j$ 表示在粒子根据当前速度更新位置. 即当前位置 $\mathbf{X}_i$ 按照 $\mathbf{V}_j$ 的策略对当前大规模图状数据处理任务调度策略进行调整. 此乘法操作仅针对位置和速度之间的相互操作, 例如 $(7, 3, 2, 1, 6, 5, 4) \otimes (1, 1, 0, 1, 1, 1, 1) = (7, 3, x, 1, 6, 5, 4)$, 表示当前调度方案中第 3 个大规模图状数据处理任务的调度方案需要调整.

因此, 针对大规模图状数据处理任务调度问题, 在 $t+1$ 次迭代计算时, 粒子 i 根据式(7)规则进行速度更新

$$\mathbf{V}_i(t+1) = P_1 \mathbf{V}_i(t) \oplus P_2 (\mathbf{X}_p - \mathbf{X}_i(t)) \oplus P_3 (\mathbf{X}_g - \mathbf{X}_i(t)) \quad (7)$$

根据式(8)规则进行位置更新

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) \otimes \mathbf{V}_i(t+1) \quad (8)$$

式中: P_1 、 P_2 、 P_3 为常量, 且 $P_1 + P_2 + P_3 = 1$.

2.2 LGPPSO 算法流程

LGPPSO 算法将每个粒子的位置 $\mathbf{X}_i$ 对应于当前大规模图状数据处理任务的调度策略,将目标函数式(4)标记为 $f(\mathbf{X})$. 其基本思想是根据大规模图状数据处理任务的先后继关系,初始化粒子群体的位置和速度,设置适应度函数值最小的粒子为最优粒子. 算法流程如下.

(1)读入大规模图状数据处理任务图,按照层次遍历进行任务序号标记,并计算任务的总数 n . 读入当前云计算环境的虚拟机处理能力和计价模式.

(2)初始化 m 个粒子的位置 $\mathbf{X}_i$ 和 $\mathbf{V}_i$,并保证这些粒子的资源租赁成本满足约束条件,算法执行的最大迭代次数设为 T .

(3)根据粒子的相关信息计算所有粒子的适应度 $f(\mathbf{X}_i)$,计算出其相应的个体极值 $\mathbf{X}_p$,选取个体极值中最小的作为全局极值 $\mathbf{X}_g$.

(4)根据粒子的适应度函数值,利用式(7)进行速度更新,用式(8)进行位置更新,位置更新中随机选取云计算中候选虚拟机列表,对每个粒子同时更新粒子的个体极值和全局极值.

(5)如未达到预先设定的迭代次数,则返回步骤 3,否则停止计算.

LGPPSO 算法的伪代码描述如下:

- 1. begin;
- 2. input large graph and resource model
- 3. initialize swarm with position $\mathbf{X}_i$ and velocity $\mathbf{V}_i$ that meet the monetary cost constraint;
- 4. do
- 5. set $t=1$
- 6. for each particle i
- 7. compute evaluation value;
- 8. if ($f(\mathbf{X}_i)<f(\mathbf{X}_{pb})$)
- 9. $\mathbf{X}_{pb}=\mathbf{X}_i$;
- 10. end
- 11. if ($f(\mathbf{X}_i)<f(\mathbf{X}_{gb})$)
- 12. $\mathbf{X}_{gb}=\mathbf{X}_i$;
- 13. end
- 14. update position and velocity using equation;
- 15. end
- 16. $t++$;
- 17. while $t<T$

大规模图状数据处理任务的总数为 n ,计算一

次调度长度的时间复杂度为 $O(n^2)$. 因粒子群迭代过程中粒子数目为 m ,迭代次数为 T ,故 LGPPSO 算法的时间复杂度为 $mTO(n^2)$.

3 性能评估与分析

针对大规模图状数据处理任务调度算法,文献[6]提出的 HEFT 算法能够得到较短的调度长度,并且得到广泛地应用. 针对云计算环境下资源按需租赁的特点,文献[9]提出的 CCSH 算法可均衡大规模图状数据处理的调度长度和资源租赁成本. 所以,为验证 LGPPSO 算法有效性,选择文献[6]和文献[9]中的启发式方法做为基线算法进行比较. 主要的评价指标包括任务的调度长度和资源租赁开销.

3.1 实验设置

云计算仿真实验环境设置为 100 个处理器性能有差异的虚拟机和全连接网状拓扑结构. 处理器频率服从 1.33 GHz~4.0 GHz 的均匀分布,网络带宽为 100 Mb/s. 速度最慢的虚拟机计价为 0.5 元/计算单元. 本文分析算法性能所选用的测试数据为随机生成图和真实应用图. 所有图中节点的计算负载随机在区间 200~1 000 节点内生成. 传输与计算负载比 R_{cc} 取 0.1、0.3、0.5、1、5 进行实验测试. 类似于文献[20],本文选用 DAG Generator^[21] 随机生成异构的大规模图状数据. 真实应用图选用高斯消去图模型,该模型在图像聚类分析和网络拓扑结构分析上有广泛应用.

对于 LGPPSO 算法,设定粒子群初始种群数为 8,算法的最大迭代次数设为 30. 式(8)中的 P_1 、 P_2 、 P_3 分别被设置为 0.1、0.2 和 0.7,并且所有实验结果均重复运行取平均值. LGPPSO 算法在以调度长度为适应度函数的情况下,随机生成图和真实图状结构的实验结果如图 1~图 4 所示. LGPPSO 在以调度长度和资源租赁成本为适应度函数的情况下,随机生成图和真实图状结构的实验结果如图 5~图 9 所示.

3.2 仿真实验结果与分析

图 1 和图 2 表明,与 HEFT 算法相比,针对随机生成大规模图状数据,LGPPSO 算法显著降低了大规模图状数据的调度长度. 在随机 200 和 400 节点任务情况下,调度长度平均降低了约 8.3%和 9.7%. 图 3 和图 4 表明,本文所设计的 LGPPSO 算法同样适用于真实图状结构应用,在高斯 100 个节点和 200 个节点的情况下,调度长度平均降低 14.6%和 16.9%,其主要原因是 HEFT 仅将当前优

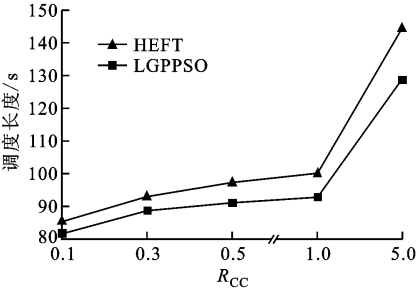


图 1 随机 200 节点调度长度比较

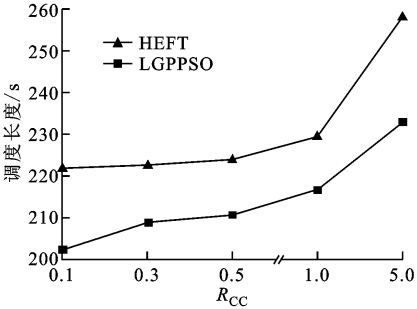


图 2 随机 400 节点的调度长度比较

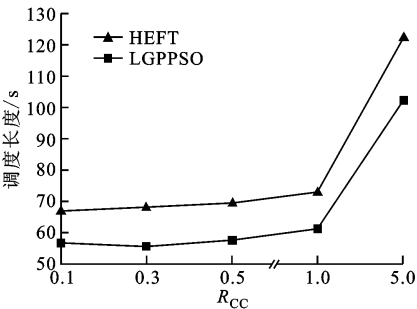


图 3 高斯 100 节点的调度长度比较

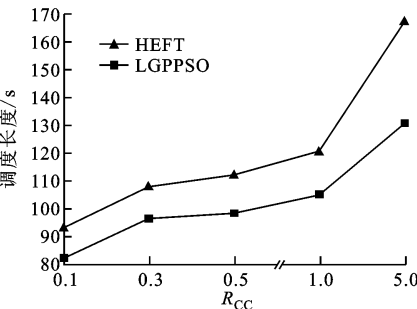


图 4 高斯 200 节点的调度长度比较

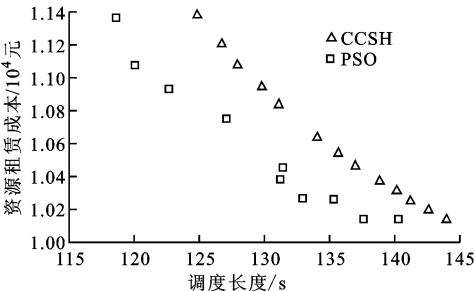


图 5 随机 200 节点资源租赁成本和调度长度比较

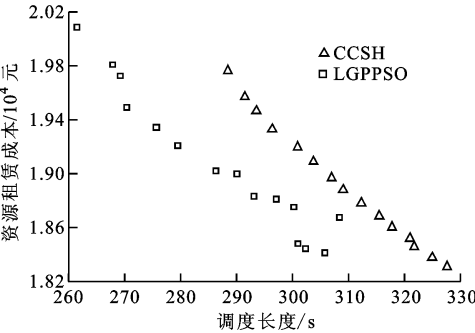


图 6 随机 400 节点资源租赁成本和调度长度比较

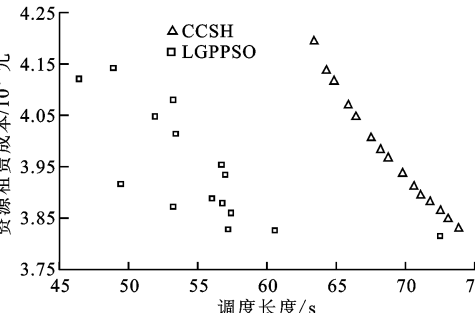


图 7 高斯 100 节点资源租赁成本和调度长度比较

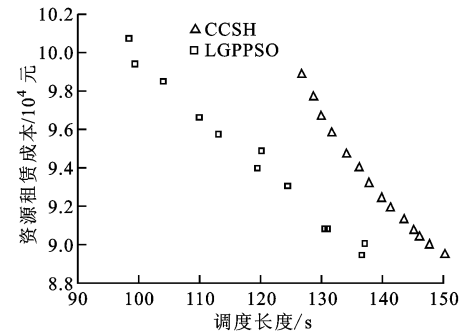


图 8 高斯 200 节点资源租赁成本和调度长度比较

优先级最高的任务调度到最早完成的机器执行,未从图状数据结构的整体进行考虑,而本文提出的基于粒子群优化的任务调度算法初始化多个粒子,增大了解空间搜索范围,提高了获得近似最优解的可能性。

文献[9]设计的 CCSH 算法和 HEFT 算法相

比,能够节省大约 40%左右的资源租赁成本,但是却延长了 10%左右的调度长度,而本文利用帕勒托最优理论设计的目标函数,能够有效解决此问题。图 5 和图 6 表明,与 CCSH 算法相比,在 $R_{cc}=1$ 和随机生成节点数为 200 和 400 的情况下,本文算法

的资源租赁成本降低约 0.16% 和 0.28%, 调度长度分别平均降低了约 9.3% 和 5.6%。图 7 和图 8 表明, 本文所设计的 LGPSO 算法同样适用于真实图状结构应用。在 $R_{cc}=1$ 和高斯消去图节点数为 200 和 100 时, 本文算法的资源租赁成本降低约 0.13% 和 0.27%, 调度长度分别平均降低了 11.14% 和 13.85%。

4 结 论

云计算环境下的大规模图状数据处理任务调度算法, 需要在缩短调度长度的同时, 降低处理这些任务的资源租赁成本。但是, 目前的调度方法仅考虑调度大规模图状数据处理任务的长度, 而忽略了云计算环境下的资源租赁成本。本文利用帕勒托最优理论, 建立了最小化调度长度和资源租赁成本的目标函数, 重新定义了粒子群优化算法中粒子的位置、速度和相关操作, 通过粒子群算法的迭代过程, 得到最小调度长度和资源租赁成本的任务调度方案。模拟实验结果表明, 本文所设计的基于粒子群优化技术的大规模图处理任务调度算法, 能够针对任意的图状结构, 达到优化调度长度和降低资源租赁成本的目标。

参考文献:

- [1] MALEWICZ G, AUSTERN M H, BIK A J, et al. Pregel: a system for large-scale graph processing[C] // Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. New York, USA: ACM, 2010:135-146.
- [2] 于戈, 谷峪, 鲍玉斌, 等. 云计算环境下大规模图数据处理技术[J]. 计算机学报, 2011, 34(10):1753-1767.
YU Ge, GU Yu, BAO Yubin, et al. Large scale graph data processing on cloud computing environments [J]. Chinese Journal of Computers, 2011, 34(10):1753-1767.
- [3] ARMBRUST M, FOX A, GIFFITH R, et al. Above the clouds: a Berkeley view of cloud computing [EB/OL]. (2009-10-08) [2012-04-03]. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>.
- [4] BUYYA R, YEO C S, VENUGOPAL S, et al. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility [J]. Future Generation Computer Systems, 2009, 25(6):599-616.
- [5] ULLMAN J K. NP-complete scheduling problems[J]. Journal of Computer and Systems Sciences, 1975, 10(3): 498-500.
- [6] TOPCUOGLU K, HARIRI S, WU M. Performance-effective and low-complexity task scheduling for heterogeneous computing [J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(3): 260-274.
- [7] BOZDA D, ZGNER F, CATALYUREK R U. A task duplication based bottom-up scheduling algorithm for heterogeneous environments [C] // Proceedings of the 20th International Parallel and Distributed Processing Symposium. Piscataway, NJ, USA: IEEE, 2006:160-172.
- [8] YANG T, GERASOULI A. DSC: scheduling parallel tasks on an unbounded number of processors [J]. IEEE Transactions on Parallel and Distributed Systems, 1994, 5(9):951-967.
- [9] DEELMAN E, SINGH G, SU M H, et al. Pegasus: a framework for mapping complex scientific workflows onto distributed systems[J]. Scientific Programming, 2005, 13(3): 219-237.
- [10] ISARD M, BUDIU M, YU Yuan, et al. Dryad: distributed data-parallel programs from sequential building blocks [C] // Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems. New York, USA: ACM, 2007: 59-72.
- [11] WARNEKE D, KAO O. Nephele: efficient parallel data processing in the cloud [C] // Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers. New York, USA: ACM, 2009: 1-10.
- [12] LI Jian, SU Sen, CHENG Xiang, et al. Cost-conscious scheduling for large graph processing in the cloud [C] // Proceedings of the 13th International Conference on High Performance Computing and Communications. Piscataway, NJ, USA: IEEE, 2011: 808-813.
- [13] Amazon Amazon Elastic Compute Cloud [EB/OL]. (2009-03-01) [2012-05-17]. <http://calculator.s3.amazonaws.com/calc5.html>.
- [14] Google. Google app engine [EB/OL]. (2010-04-20) [2011-07-03]. <http://cloud.google.com/pricing/compute-engine.html>.
- [15] ZHU Q, AGRAWAL G. Resource provisioning with budget constraints for adaptive applications in cloud environments [C] // Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing. New York, USA: ACM, 2010: 304-

- 307.
- [16] ULUNGU G, TEGHEM J. Multiobjective combinatorial optimization problems: a survey[J]. Journal of Multi-Criteria Decision Analysis, 1994, 3(2): 83-104.
- [17] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of the International Conference on Neural Networks. Piscataway, NJ, USA: IEEE, 1995: 1942-1948.
- [18] HOLLAND J. Adaptation in natural and artificial systems [D]. Boston, MA, USA: Massachusetts Institute of Technology, 1992.
- [19] 程祥, 张忠宝, 苏森, 等. 基于粒子群优化的虚拟网络映射算法[J]. 电子学报, 2011, 39(10): 2240-2244.
- CHENG Xiang, ZHANG Zhongbao, SU Sen, et al. Virtual network embedding based on particle swarm optimization [J]. Chinese Journal of Electronics, 2011, 39(10): 2240-2244.
- [20] BARBOSA J G, MOREIRA R. Dynamic scheduling of a batch of parallel task jobs on heterogeneous clusters [J]. Parallel Comput, 2011, 37(8): 428-438.
- [21] Institut National de Recherche en Informatique et en Automatique. DAG generation program [EB/OL]. (2005-04-09) [2012-03-12]. <http://www.loria.fr/~suter/dags/html>.

(编辑 刘杨 赵大良)