

基于安全性约束和协同禁忌优化算法的 云计算 workflow 调度

邢立峰

(河套学院 理学系, 内蒙古 巴彦淖尔 015000)

摘要: 针对现有的云计算 workflow 调度方法侧重于节能和可靠性等方面进行研究, 忽略了安全性约束的要求, 设计了一种基于协同禁忌算法的具有安全型约束的能实现云计算 workflow 高效调度的方法; 首先, 定义了云计算 workflow 调度的 DAG 图, 对安全性约束进行了形式化描述, 并建立了云计算 workflow 调度的数学模型; 然后, 在经典的协同禁忌算法的基础上, 通过设计解的编码方式、适应度函数、变邻域结构和双禁忌表, 实现对经典的协同禁忌算法进行改进; 最后, 定义了基于改协同禁忌算法实现对云计算 workflow 调度的具体算法; 在云计算的仿真环境 Cloud-Sim 下进行实验, 实验结果证明了文中方法不仅具有较快的收敛速度, 而且能寻找到比其它方法更优的调度方案, 能满足安全性约束要求, 是一种可行实用的调度方法。

关键词: workflow 调度; 虚拟机; 安全性约束; 云计算

Work Flow Scheduling Based on Safety Constraint and Cooperative Tabu Optimization Algorithm in Cloud Computing

Xing Lifeng

(Hetao College Department of Physiology, Bayannaoer 015000, China)

Abstract: Aiming at the given cloud computing method focus on saving power or reliability, not paying attention to the safety constraint, a work flow scheduling method based on cooperative tabu algorithm with safety constraint is proposed. Firstly, the DAG figure of work flow scheduling in cloud computing is defined, the safety constraint is described formally, the mathematic model of cloud computing work flow scheduling is built. Then on the basis of the classic cooperative tabu algorithm, the code of solution, fitness function, variable neighbor structure and double tabu table are designed, to realize the improvement of the classic tabu algorithm. Finally, the specific algorithm base on cooperative tabu algorithm is defined. The experiment is operated in the Cloud-sim environment, the result shows the method in this paper not only has the quicker convergence rate, and can get the more optimal scheduling scheme, satisfying the safety constraint need. Therefore, it is a feasible and applicable scheduling method.

Keywords: work flow scheduling; virtual machine; safety constraint; cloud computing

0 引言

云计算是在并行计算、网格计算、分布式计算的基础上发展而来^[1-2], 现有的云计算能提供比网格计算和并行系统更高的可靠性, 但是仍然存在一些问题, 如: 难以在保证用户服务质量的前提下, 压缩云计算的运行成本, 降低云计算中心的能耗, 使得服务提供商能实现收益的最大化^[3-4]。

云计算 workflow 的调度效率的高低受到硬件性能、调度的可靠性和完成时间长短的影响^[5]。目前对云计算 workflow 的调度往往着力于节能和可靠性等方面进行研究, 如文献 [6] 提出了一种基于可靠性的 workflow 调度方法, 减少传输失败率和传输时间, 提高整体的可靠性。文献 [7] 对网络资源属性进行量化和规范化, 使得任务在调度时能选择综合性能高的资源类簇, 缩短任务资源匹配时间。文献 [8] 采用回溯和分支限界整合云计算任务路径优化选择。文献 [9] 根据 DAG workflow 按照 QOS 需求进行优先级划分, 对于优先级高的作业分配资源。

以上工作均研究云计算 workflow 调度, 但未考虑安全性约

束, 因此, 本文在上述工作的基础上提出了一种云计算环境下的满足安全性约束的云计算流动态调度方法。

1 云计算 workflow 调度 DAG 图

采用有向无环图 (directed acirclic graph, DAG) 来描述云计算 workflow 任务结构, 一个网格任务依赖型有向无环图 DAG 描述如图 1 所示。

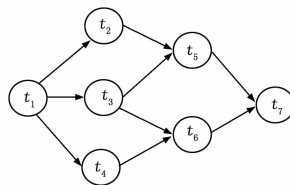


图 1 workflow 调度 DAG 图

在图 1 中, workflow 调度 DAG 图可以采用二元组表示 $DAG = \{T, E\}$, 其中:

1) $T = \{t_1, t_2, \dots, t_n\}$ 表示 DAG 中的节点集, 其含义是用户任务对应的若干子任务集, $n = |T|$ 为任务总数, 采用 $W(t_i)$ 表示任务 t_i 的计算量;

2) $E = \{e_{ij} = \langle t_i, t_j \rangle, e_{ij} \in T \times T\}$ 为有向边集合, 表示任务 t_i 和 t_j 之间的依赖关系, t_j 必须在任务 t_i 完成后才开始处理。

收稿日期: 2014-09-19; 修回日期: 2014-10-17。

作者简介: 邢立峰 (1975-), 女, 内蒙古五原人, 硕士研究生, 讲师, 主要从事数据库与计算机应用方向的研究。

采用矩阵 C 表示任务之间的通信关系 $C = \{c_{ij} = (t_i, t_j), c_{ij} \in T \times T\}$, c_{ij} 表示任务 t_i 和 t_j 分配到不同的资源上时的通信量, 当 t_i 和 t_j 分配到相同的资源上时, $c_{ij} = 0$ 。

DAG 图中的任意任务的前驱节点表示 $Pred(t_i) = \{t_j \mid t_j \in T, e_{ji} \in E\}$ 任务 t_i 的前驱任务集, $Succ(t_i) = \{t_j \mid t_j \in T, e_{ij} \in E\}$ 表示任务 t_i 的后继任务。

2 基于 DAG 和安全性约束的工作流调度

2.1 工作安全性约束

根据使用技术的安全性强度, 可以将虚拟机划分为相应安全性级别, 根据操作的敏感程度, 访问控制和用户需求, 采用 r-risk 型安全访问控制方法, 即在对工作流进行调度时, 设定工作流的冒险水平阈值 τ , 即安全水平高于工作流阈值的虚拟机可以用于进行分配。下面从任务的安全性级别和多个子任务构成的 DAG 图的整体安全性建模如下:

1) 单个任务 t_i 满足安全性约束的分配: 其冒险水平阈值 τ_i , 分配给其的虚拟机为 v_j , 虚拟机的安全性级别为 s_j , 当 $s_j \geq \tau_i$ 时, 该虚拟机满足安全性约束要求, 可以分配给任务 t_i 。例如对于安全需求为 3 的操作, 可以分配到安全性级别分别为 4 以上的虚拟机上。

2) 整个任务集即 DAG 满足安全性约束的调度: 任务集 $T = \{t_1, t_2, \dots, t_n\}$ 的分配方案 $P = \{p_1, p_2, \dots, p_n\}$ 的整体风险概率可以通过下式计算:

$$p(risk) = 1 - e^{-\sum_{i=1}^m (s_i - v_i)} \quad (1)$$

当 $p(risk)$ 大于所有任务的冒险水平阈值 τ_i 的最大值时, 则认为分配方案 $P = \{p_1, p_2, \dots, p_n\}$ 是满足安全性约束的。

2.2 数学模型的定义

对于任意任务, 其操作时间可以分为两部分, 即接收其所需数据所需要的时间开销和将任务分配到所需虚拟机上所需要的时间, 对于任意任务, 满足其安全性约束的虚拟机集为 M , 其完成时间 $Finish_i$ 可以通过下式计算:

$$Finish_i = \max_{pre_i} Finish_{pre_i} + \frac{Cm_i}{ban_{ij}} + \frac{L_i}{p_i} \quad (2)$$

$s, tj \in M$

在式 (2) 中, $\max_{pre_i} Finish_{pre_i}$ 是当前任务节点的任意前驱节点完成时间的最大值。 Cm_i 是当前任务节点所需要传输的数据量, ban_{ij} 为数据来源虚拟机与当前任务分配所在虚拟机之间的带宽, L_i 为当前节点的工作量, p_i 为当前节点所分配的虚拟机的处理速度。

所有任务即整个 DAG 的任务完成时间为耗时最长的任务所需要的时间, 即:

$$Finish_{DAG} = \max_i Finish_i \quad (3)$$

3 基于禁忌优化算法的工作流调度

3.1 禁忌优化算法

禁忌算法是一种由 Glover 在 1986 年提出的优化算法, 禁忌算法引入了人工智能的记忆机制和禁忌表, 因此能表现出极强的全局寻优能力。

为了防止算法陷入局部最优, 禁忌优化算法采用禁忌表存储已经获得局部最优解, 并设置每个局部最优解的禁忌长度, 在下次搜索中, 利用禁忌表中的信息来不再选择这些点, 从而跳出局部最优。同时禁忌算法还能根据藐视准则赦免被禁忌的

优良状态, 选择禁忌表中已有最优解作为当前解, 最终实现全局最优解的获取。经典的禁忌优化算法的流程如图 2 所示。

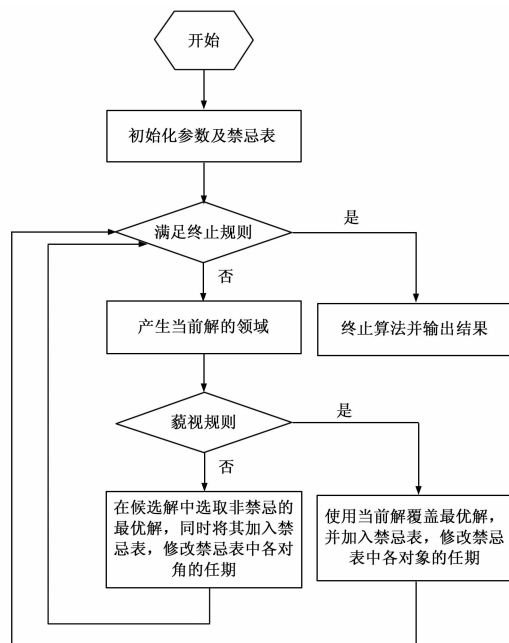


图 2 传统禁忌算法流程图

3.2 解的编码方式和适应度函数

采用向量 $P = \{p_1, p_2, \dots, p_n\}$ 来表示当前解, 对于向量中的每个元素 p_i , 均表示 t_i 分配的虚拟机 v_i , 因此, 当前 DAG 工作流对应的编码长度即为该用户任务对应的子任务总数 n 。

适应度函数即在禁忌算法寻优的过程中最大化的目标函数, 由于式 (4) 所示的目标函数是最小化式 (3) 所示的 DAG 的任务完成时间, 即适应度函数可以表示为:

$$Fitness = 1/Finish_{DAG} \quad (4)$$

3.3 邻域结构设计

在图 2 所示的经典禁忌优化算法中, 由当前解生成候选解的方法需要设计邻域结构, 如果邻域解和当前解差别过大时, 会使得禁忌搜索过程为随机搜索, 而变化太小又会减慢收敛速度, 可能陷入局部最优, 因此, 需要设计合理的邻域结构, 在保证解的优越性的同时, 加快算法的收敛。

假设基本邻域结构为: 为当前任务节点随机选择一个满足安全性约束的虚拟机, 这可以防止随机搜索, 使得新解在合理的范围选择, 为了防止算法早熟收敛, 设计两种变邻域结构, 当达到算法指定迭代次数后, 如果当前解的适应度还未明显改善, 则依次采用下面的基于参考值 S 的变邻域结构 1 和变邻域结构 2 产生新的候选解。

变邻域结构 1: 每次从当前解产生一个候选解, 可以通过连续调用 S 次基本邻域结构产生;

变邻域结构 2: 当对当前解产生邻域解释时, 需要改变当前解附近的 $2S$ 范围内所有节点的虚拟机编号。

3.4 基于改进禁忌优化的工作流调度算法

采用改进的禁忌算法实现工作流调度的过程可以描述为:

输入: DAG 工作流的所有任务集 $T = \{t_1, t_2, \dots, t_n\}$, 当前虚拟机集合 $V = \{v_1, v_2, \dots, v_n\}$, 最优解最久未变化次数 r_{max} , 禁忌表长度 L , 参考值 S , 领域集合 $S_{neighbor}$ 的元素个数

最大值为 N ，候选解集 S_{selected} 中元素个数最大值为 M ，算法最大迭代次数 T ，候选集元素个数 K ；

输出：全局最优解 best-far ；

步骤 1：随机产生满足安全性约束的解作为当前解 x_{cur} ，初始化最优解 $\text{best-far} = x_{\text{initial}}$ ，当前迭代次数 $t=1$ ，最优解未变化次数 $r=1$ ；

步骤 2：将当前解 x_{cur} 放入禁忌表 TB 中，将移动量 (0, 0) 放入禁忌表 TW 中，设置其禁忌长度为 L ；

步骤 3：判断 $t \leq T$ 是否成立：如果成立，则算法结束，输出最优解 best-far ；否则 $t=t+1$ ；

步骤 4：根据 3.3 节所示邻域结构生成当前解 x_{cur} 的邻域 S_{neighbor} ，直到 S_{neighbor} 中元素的个数为 N ，选择其中最优的 K 个作为候选解，加入选解集 S_{selected} ；

步骤 5：将候选解 S_{selected} 中的最优候选解 $s_{\text{selected-best}}$ 与最优解 best-far 比较：

If $\text{Fitness}(s_{\text{selected-best}})$ 大于 $\text{Fitness}(\text{best-far})$

$r = r + 1$ ；

$\text{best-far} = s_{\text{selected-best}}$

$x_{\text{cur}} = s_{\text{selected-best}}$

If $s_{\text{selected-best}}$ 不在禁忌表中，则将 $s_{\text{selected-best}}$ 加入禁忌表 TB 中，设置其禁忌长度为 L ，将其移动方式加入禁忌表 TW 中，将禁忌表中其他元素的禁忌长度设置为 -1；

Else 选择未被禁忌的下个较优候选解 $s_{\text{selected-second}}$ 作为当前最优解 x_{cur} ，将其加入到禁忌表中，将其移动方式加入禁忌表 TW 中，更新禁忌表中其他元素的禁忌长度 -1；

步骤 6：当前迭代次数 $t = t + 1$ ，并转入步骤 3 继续执行。

4 仿真实验

在云计算的仿真平台 Cloud Sim 下进行仿真试验，工作流任务实例如图 3 所示，与方便与文献 [10] 进行比较，实验参数均与文献 [10] 保持一致，椭圆内表示每个子任务，任务的工作量满足 [50, 500] 内的随机均匀分布，虚拟机的个数为 4，虚拟机之间带宽矩阵为：

$$Mv = \begin{bmatrix} 0 & 40 & 120 & 80 \\ 40 & 0 & 60 & 100 \\ 120 & 60 & 0 & 90 \\ 80 & 100 & 90 & 0 \end{bmatrix} \quad (5)$$

所有任务集中的任务仅第 6 个安全性约束阈值为 4，其余任务均为 2，虚拟机的安全性等级取值分别为 2、2、3、4。

协同禁忌算法的参数设置如下：优解最久未变化次数 $r_{\text{max}} = 4$ ，禁忌表长度 $L=4$ ，参考值 $S=1$ ，领域集合 S_{neighbor} 的元素个数最大值为 $N=5$ ，候选解集 S_{selected} 中元素个数最大值为 $M=3$ ，算法最大迭代次数 $T=200$ ，候选集元素个数 $K=6$ 。

将文中方法、文献 [9] 方法和文献 [10] 方法对图 3 所示的工作流实例进行仿真，得到的工作流的完成时间随着算法迭代次数变化的收敛情况如图 4 所示。

从图 4 中可以看出，文中方法在 140 代时就已经收敛，收敛的工作流调度完成时间为 178.4，其它两种方法最终得到的工作流调度完成时间分别为 195.1 和 210，文献 [9] 方法在整个仿真期间都未能收敛，而文献 [10] 在 180 代时收敛，但没有收敛到全局最优解，显然文中方法不仅具有较快的收敛速度，而且收敛的解更优。

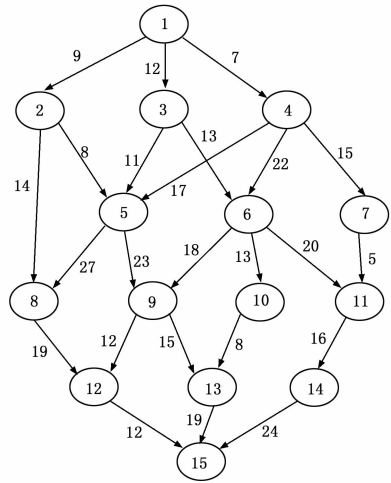


图 3 工作流任务实例

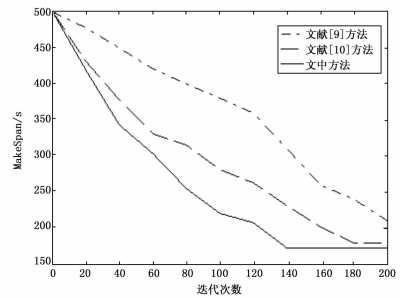


图 4 3 种算法收敛曲线比较

为了 3 种方法对安全性约束的敏感程度，将三种方法在有安全性约束和无安全性约束下得到的平均完成时间进行仿真，结果如图 5 所示。

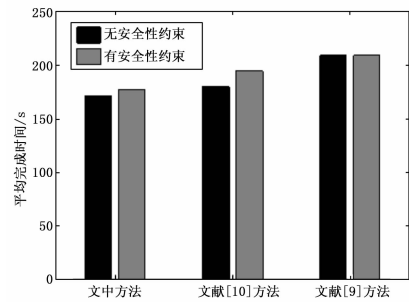


图 5 有/无安全性约束的平均完成时间

从图 5 可以看出，文中方法和文献 [10] 方法在算法中考虑了安全性约束，且在安全性约束时获得的平均完成时间与无安全型约束时相差较小，而文献 [9] 方法由于在算法中没有考虑安全性约束，因此不能解决安全型约束的云工作流调度问题。

5 结语

为了对云计算中的任务进行合理高效调度，需要首先解决的关键问题就是云计算工作流调度，为此，本文设计了一种具有安全型约束的能实现云计算工作流高效调度的方法。在文中设计了 DAG 工作流调度模型和具有安全型约束的目标数学模型。采用协同禁忌算法实现对目标数学模型的寻优。在云计算 Cloud-Sim 工具下进行仿真，通过实验证明了文中方法不仅具有较快的收敛速度，而且能寻找到比别的方法更优的调度方案。

(下转第 316 页)

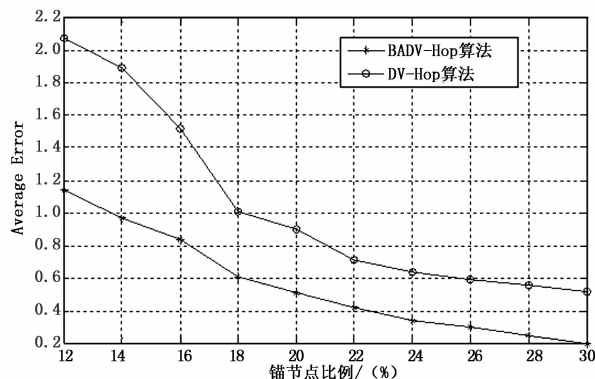


图 4 不同锚节点个数下的定位性能

得到大幅度提高,有效降低了无线传感器网络的节点定位误差。

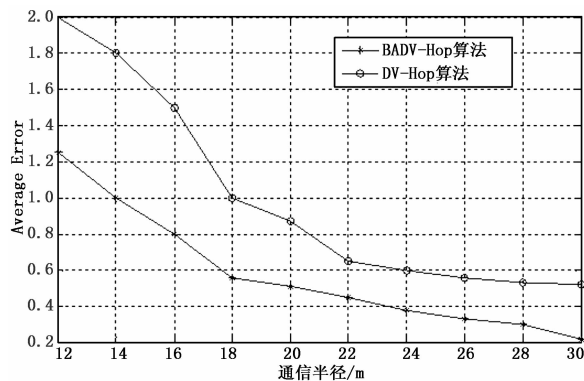


图 5 不同通信半径下的定位性能

3.2.4 定位结果比较

DV-Hop 算法和 BADV-Hop 算法的定位结果如图 6 所示。从图 6 可知, BADV-Hop 算法的定位误差小于 DV-Hop 算法,且定位效果明显得以提高。

4 结束语

为了提高无线传感器网络定位精度,提出一种蝙蝠算法和 DV-Hop 算法相融合的节点定位方法,采用蝙蝠算法修正 DV-Hop 算法的定位误差,仿真结果表明,相比于 DV-Hop 算法,

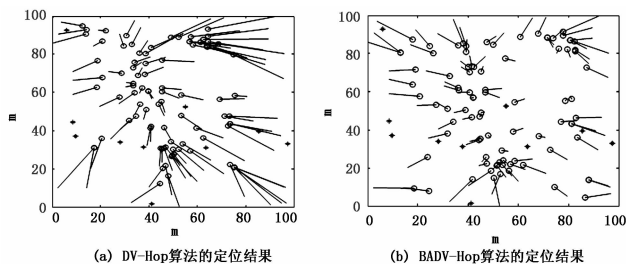


图 6 两种算法的定位结果对比

BADV-Hop 算法获得优的节点定位结果,优势十分明显,在无线传感器网络中具有广泛的应用前景。

参考文献:

- [1] 王福豹,史 龙,任丰原. 无线传感器网络中的自身定位系统和算法 [J]. 软件学报, 2005, 16 (5): 857-868.
- [2] 赵仕俊,孙美玲,唐懿芳. 基于遗传模拟退火算法的无线传感器网络定位算法 [J]. 计算机应用与软件, 2009, 26 (10): 189-192.
- [3] 孙泽宇,魏 巍. 一种改进无线传感器网络定位算法的研究 [J]. 计算机仿真, 2010, 27 (9): 125-127.
- [4] 衣 晓,王梓有,薛兴亮. 一种基于次锚节点的无线传感器网络质心定位算法 [J]. 计算机应用与软件, 2013, 30 (6): 116-120.
- [5] Geng Y, He J, Pahlavan K. Modeling the Effect of Human Body on TOA Based Indoor Human Tracking [J]. International Journal of Wireless Information Networks, 2014, 20 (4): 306-317.
- [6] 刘运杰,金明录,崔承毅. 基于 RSSI 的无线传感器网络修正加权定位算法 [J]. 传感技术学报, 2009, 23 (5): 717-721.
- [7] 李 辉,熊盛武,刘 毅,等. 无线传感器网络 DV-Hop 定位算法的改进 [J]. 传感技术学报, 2011, 24 (12): 1782-1786.
- [8] 毛科技,赵小敏,何文秀,等. WSN 中基于区域划分的半自动 DV-Hop 定位算法 [J]. 计算机科学, 2012, 39 (3): 39-42.
- [9] 叶 蓉,赵灵错. 基于蚁群粒子群混合的无线传感器网络定位算法 [J]. 计算机测量与控制, 2011, 19 (3): 732-735.
- [10] 陈星舟,廖明宏,林建华. 基于粒子群优化的无线传感器网络节点定位改进 [J]. 计算机应用, 2010, 30 (7): 1736-1739.
- [11] Yang X S, Gandomi A H Bat algorithm: a Novel Approach for Global Engineering Optimization [J]. Engineering Computations, 2012, 29 (5): 464-483.
- [12] IEEE press [C]. 2012: 194-200.
- [13] Kang Q, He H, Wei J. An effective iterated greedy algorithm for reliability-oriented task allocation in distributed computing systems [J]. Journal of parallel and distributed computing, 2013, 73 (8): 1106-1115.
- [14] 闫 歌,于 炯,杨兴耀. 基于可靠性的云计算工作流调度策略 [J]. 计算机应用, 2014, 34 (3): 673-677.
- [15] 郭凤羽,禹 龙,田生伟,等. 云计算环境下对资源聚类的工作流任务调度算法 [J]. 计算机应用, 2013, 33 (8): 2154-2157.
- [16] 刘丹琦,于 炯,英昌甜. 云计算环境下多有向无环图工作流的节能调度算法 [J]. 计算机应用, 2013, 33 (9): 2410-2415.
- [17] 孙 月,于 炯,朱建波. 云计算中一种多 DAG 工作流可抢占式调度策略 [J]. 计算机科学, 2014, 3 (41): 145-148.
- [18] 马俊波,殷建平. 云计算环境下带安全约束的工作流调度问题的研究 [J]. 计算机工程与科学, 2014, 4 (36): 607-614.

(上接第 306 页)

参考文献:

- [1] Du Z H, Hu J K, Chen Y N, Wang X Y. Optimized QoS-aware replica placement heuristics and applications in astronomy data grid [J]. The journal of Systems and software, 2011, 84 (7): 1224-1232.
- [2] 刘少伟,孙令梅,任开军,等. 云计算下优化科学工作流执行性能的两阶段数据放置与任务调度策略 [J]. 计算机学报, 2011, 11 (34): 2121-2130.
- [3] Vaquero L M, Roderio Merino L, Moran D. Locking the sky, a survey on IaaS cloud security [J]. Computing, 2011, 91 (1): 93-118.
- [4] Malik S, Huef F, Caromel D. Reliability aware scheduling in cloud computing [A]. Proceedings of the 2012 international conference for internet technology and secured transactions. Piscataway, Nj: