

# 云计算任务冗余调度优化模型<sup>①</sup>

孙健美, 曾碧卿

(华南师范大学 计算机学院, 广州 510631)

**摘要:** 为了解决现有云计算平台的高度动态性和异构性带来的可靠性低的问题, 提出了一种基于冗余调度的可靠云计算模型。针对云计算按需付费的特点, 基于该可靠模型, 建立了基于服务费用约束的冗余调度优化模型, 以保证在该费用约束下获得最大的云计算服务可靠性。最后, 用遗传算法解决该最优冗余分配问题。

**关键词:** 云计算; 可靠性; 冗余调度; 最优冗余分配; 遗传算法

## Optimal Model Redundant Task Scheduling for Cloud Computing

SUN Jian-Mei, ZENG Bi-Qing

(School of Computer, South China Normal University, Guangzhou 510631, China)

**Abstract:** In order to solve the problem of low reliability of Cloud Computing Platform for its own dynamic and heterogeneous, a reliable Cloud Computing model based on redundant scheduling is proposed. For the pay-as-need features of Cloud Computing, an optimal model redundant task scheduling for Cloud Computing based on the given cost of services is established to maximize the reliability of Cloud Computing. Finally, Genetic Algorithms is used to solve the problem of optimal redundancy allocation.

**Key words:** cloud computing; reliability; redundant scheduling; optimal redundancy allocation; genetic algorithm

## 1 引言

云计算是网格计算、分布式计算、并行计算、效用计算、网络存储、虚拟化、负载均衡等传统计算机和网络技术发展融合的产物, 是一种新兴的商业计算模式, 是目前国内外商业和科研机构研究的热点之一, 其核心思想是通过互联网将超大规模的计算存储资源整合起来, 形成一个虚拟的计算资源池, 并以服务的形式按需提供给用户, 简化了任务调度的过程, 将原来网格计算的任务调度问题的复杂性留给了虚拟化技术来解决。云计算作为网格计算的发展, 虽然它的规模更大, 但是它仍存在一些网格计算, 甚至是其它所有计算系统都存在的问题——服务应用程序处理失败。如 Amazon 的 S3 服务的中断、Google 的 Apps 服务和 Gmail 服务的中断等, 这无疑让许多专家学者对云计算的可靠性和安全性产生怀疑。为此, 研究高效率的云计算资源调度算法, 尤其是具有容错机制<sup>[1-3]</sup>

和失败容忍能力的可靠云计算服务调度策略, 对切实提高云计算服务的可靠性具有重要意义。冗余调度作为软件容错技术的一种常用的手段, 本文将运用于云计算环境中, 并通过实验证实了加入一定的冗余既保证了服务可靠性又加快了任务的执行速度。此外, 本文又结合了实际生活中用户对各种资源节点的支付能力及对各种调度方案的满意程度的不同, 提出了服务费用约束下的冗余优化调度方案, 并用遗传算法求解该优化模型。

## 2 基本定义

定义 1.  $T = \{t_1, t_2, \dots, t_n\}$  代表一个任务, 由  $n$  个相互独立的子任务构成的任务集合。

定义 2.  $R = \{r_1, r_2, \dots, r_m\}$  表示由  $m$  个异构的计算资源构成的资源集合。

<sup>①</sup> 基金项目: 广东省自然科学基金(8151063101000040)

收稿时间: 2011-10-14; 收到修改稿时间: 2011-11-05

定义 3.  $ETC = \begin{bmatrix} et_{11}, \dots, et_{1m} \\ \dots, \dots, \dots \\ et_{n1}, \dots, et_{nm} \end{bmatrix}$  表示预期执行时间矩阵

阵,  $et_{ij}$  表示第  $j$  个资源上执行第  $i$  个任务所要用的时间。根据相应的调度方案可以计算出每个资源完成分配给它的所有任务所要用的时间, 求出最大值即最大执行时间跨度 (也称最优跨度)。

定义 4. 资源节点的稳定性<sup>[4]</sup>定义为: 资源正常运行的时间 / (资源正常运行的时间 + 资源出错运行的时间)。

定义 5.  $Allocation = \begin{bmatrix} al_{11}, \dots, al_{1m} \\ \dots, \dots, \dots \\ al_{n1}, \dots, al_{nm} \end{bmatrix}$  表示任务资源

分配矩阵,  $al_{ij}=1$  表示将任务  $i$  分配到资源  $j$  上执行,  $al_{ij}=0$  表示不将任务  $i$  分配到资源  $j$  上执行。

### 3 基于冗余调度的可靠云计算模型的建立

#### 3.1 可靠云计算模型的提出

冗余调度是指为了能够确保任务的顺利执行, 将一个任务分配到多个资源节点上进行执行, 只要其中一个资源节点完成了任务, 那么任务就执行成功, 当然, 现状最好的资源节点将最先返回执行结果, 这样就可以以较快的速度获得结果。此外, 冗余调度还可以实现结果验证的功能。

在进行任务冗余调度时, 资源的冗余度关系到任务的执行效率, 冗余度过大意味着占据过多的资源, 这不仅仅造成了许多不必要的花费也大大降低了资源的利用率; 若冗余度过小, 则云计算平台提供的服务可靠性就会降低。另外资源节点的稳定性 (包括硬件、软件、通信链路等) 也会影响到任务执行成功的概率。任务调度的冗余度和资源的稳定性是影响服务可靠性的两个关键因素<sup>[5]</sup>, 因此云计算服务的可靠性可定义为资源冗余度和稳定性的函数  $R = F(r, s)$ , 其中  $r$  表示任务的冗余度,  $s$  表示资源节点的稳定性。若用  $s_k$  表示资源  $k$  的稳定性, 则  $1-s_k$  表示资源  $k$  的失效率, 当一个子任务在  $r$  个资源上冗余执行时 ( $r \leq r_{max}$ ,  $r_{max}$  为限定的最大冗余度, 为了方便编码), 执行失败的概率为  $\prod_{k=1}^r (1-s_k)$ , 那么该子任务执行成功的概率为  $1 - \prod_{k=1}^r (1-s_k)$ , 于此, 整个任务在云计算平台上执行成

功的概率为  $R = \prod_{i=1}^n (1 - \prod_{k=1}^r (1-s_k))$ 。

#### 3.2 几种冗余调度模型的分析比较

本文就无冗余 (不保证所有子任务都能可靠执行)、固定冗余 (每个子任务的冗余度是一个固定值, 冗余度太小则不能保证所有子任务的可靠性, 冗余度过大则花费也更多)、保证可靠性的冗余 (保证每个子任务的成功率大于 0.9, 每个子任务冗余度不一定相等) 进行了实验, 并考虑到云计算平台的复杂性, 本文按照任务数  $n$  和资源数  $m$  的相对大小, 分为三种情况: (1) 负载较轻, 取  $n=10, m=5$ ; (2) 负载一般, 取  $n=50, m=10$ ; (3) 负载偏重, 取  $n=256, m=16$ 。执行结果如图 1, 图 2 和图 3。

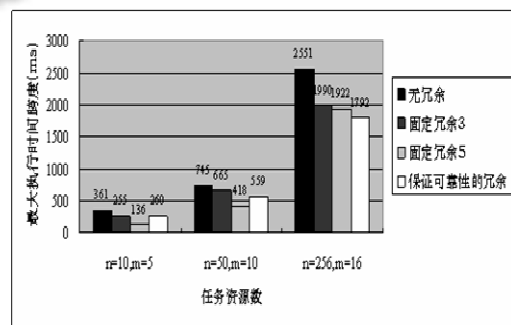


图 1 四种调度算法在不同负载下最大执行时间对比图

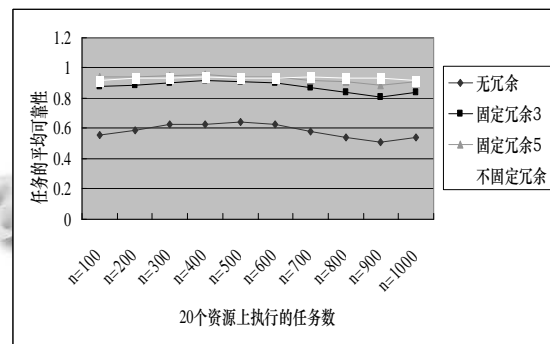


图 2 四种调度算法将任务调度到 20 个资源上执行的可靠性对比图

以上实验结果均是经过 10 次重复实验得出的结果。从图 1 可以看出有冗余的调度比无冗余的调度需更少的执行时间, 并随着负载的增加, 保证可靠性的冗余调度性能越来越优于固定冗余调度。从图 2 可以看出, 无冗余调度的可靠性远远低于有冗余调度的可靠性, 刚开始保证可靠性的冗余调度的可靠性介于固定 3 冗余和固定 5 冗余调度的之间并随着任务数的增

加, 当任务数达到一定值后, 固定冗余调度的可靠性开始降低。从图 3 可以看出, 保证可靠性的冗余调度所需的资源数总比无冗余的要少, 也比固定冗余的少; 固定冗余所需的资源数总比无冗余的要少, 并且固定的冗余度越大所需的资源数也越多。因此, 保证可靠性的冗余调度算法将更适合于云计算的任务调度。

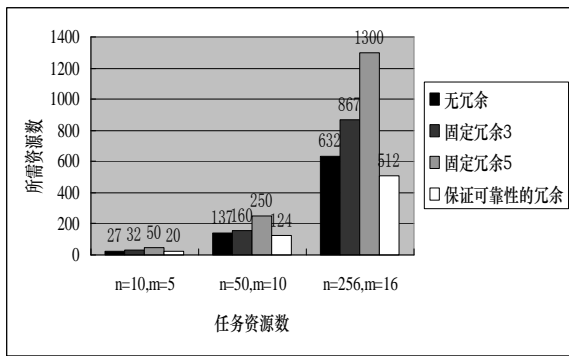


图 3 四种调度算法在不同负载下所需资源数对比图

## 4 云计算任务冗余调度优化模型的构建及求解

### 4.1 问题的提出

通过任务的冗余调度可以提高服务的可靠性, 但由于云计算本身是按需支付模式的, 不同任务对资源的需求不同, 不同资源的单价也是不同的, 因此过多的占用云计算资源意味着更高的花费, 因此本文将在上文给出的保证可靠性的冗余调度的基础上建立基于服务费用的冗余调度优化模型, 即在规定的服务费用范围内获取最大的可靠性。此模型可归结为最优冗余问题, 属于典型的 NP-hard 问题<sup>[6,7]</sup>, 遗传算法作为一种启发式智能优化搜索算法, 已经被广泛用于解决可靠性优化、任务调度等许多优化问题, 并且被相关的研究者证明其大大降低了计算的复杂性, 所以本文将采用遗传算法来求解该优化模型。在目标值

$$\sum_{i=1}^n \sum_{k=1}^r c_{ik} \leq C^* \text{ 下求解最大可靠性 } R_{\max}。$$

### 4.2 遗传算法求解优化模型的实现

遗传算法是根据自然进化论与遗传变异理论为基础求解全局最优解的仿生型算法, 其本质是一种求解问题的高效并行全局搜索算法。它根据问题的目标函数构造一个适值函数 (Fitness Function), 对一个由多个解 (每个解对应一个染色体) 构成的种群进行评估、遗传运算、选择, 经多代繁殖, 获得适应值最好的个

体作为问题的最优解。

#### 4.2.1 编码方案

初始种群是随机产生的, 具体的产生方式依赖于编码方法。在遗传算法中需要对染色体进行编码, 使一个染色体对应优化问题的一个解。本文采用的是多参数级联编码方法, 其中每一个参数采用整数编码。假设一个任务被划分成 5 个相互独立的子任务, 有 10 个可用资源, 限定的最大冗余度为 4, 则某冗余调度方案对应的一个染色体为 (1 6 0 0/3 8 0 0/5 4 0 0/2 0 0 0/10 9 0 0), 表示子任务 1 在资源 1 和资源 6 上执行, 任务 2 在资源 3 和资源 8 上执行, ..., 第 5 个任务在资源 10 和资源 9 上执行。

#### 4.2.2 适应度函数

遗传算法对一个个体 (解) 的好坏用适应度函数值来评价, 适应度函数值越大, 解的质量越好。适应度函数是遗传算法进化过程的驱动力, 也是进行自然选择的唯一标准, 它的设计应结合求解问题本身的要求而定。

本文求可靠性越大越好, 即  $R_{\max}$ , 所以采用对数标定法<sup>[8]</sup>设置适应度函数为  $f = -\frac{1}{\ln[R]}$ , 其中,  $f$  为个体的适应值,  $R$  为个体的可靠性, 对  $R$  取对数后再求负倒数, 这样保证了个体的可靠性越大, 适应值就越大, 满足适应度函数的设计要求。

#### 4.2.3 选择操作

选择操作就是用来确定如何从父代群体中按某种方法选取那些个体遗传到下一代群体中的一种遗传运算, 用来确定重组或交叉个体, 以及被选个体将产生多少个子代个体。本文采用最常用的轮盘赌方法来选择个体。

#### 4.2.4 交叉与变异操作

交叉是指对两个相互配对的染色体依据交叉概率按某种方式相互交换其部分基因, 从而形成两个新的个体。交叉运算在遗传算法中起关键作用, 是产生新个体的主要方法。本文采用了最常用的单点交叉法, 但是交叉点是设置在任务之间的, 这样避免了同一任务因为交叉操作而分配到相同的资源上。

变异是指依据变异概率将个体编码串中的某些基因值用其它基因值来替换, 从而形成一个新的个体。变异运算是产生新个体的辅助方法, 它决定了遗传算法的局部搜索能力, 同时保持种群的多样性。交叉运

算和变异运算的相互配合,共同完成对搜索空间的全局搜索和局部搜索。本文采用了基本位变异,随机选择一个新的未分配的资源去替换原有的资源。此外,本文中的交叉概率  $P_c$  和变异概率  $P_m$  均采用了自适应的选定方式,即通过个体自身适应值的大小和整个种群性能的比较来选定,  $P_c$ ,  $P_m$  分别为:

$$P_c = \begin{cases} k_1(f_{\max} - f') / (f_{\max} - f_{\text{avg}}), f' \geq f_{\text{avg}} \\ k_2, f' < f_{\text{avg}} \end{cases}$$

$$P_m = \begin{cases} k_3(f_{\max} - f) / (f_{\max} - f_{\text{avg}}), f \geq f_{\text{avg}} \\ k_4, f < f_{\text{avg}} \end{cases}$$

其中,  $f_{\max}$  为群体中最大的适应度值,  $f_{\text{avg}}$  为每代群体的平均适应度值,  $f'$  为要交叉的两个个体中较大的适应度值,  $f$  为要变异个体的适应度值。

## 5 实例分析

本文所采用的实验环境: Windows XP 的操作系统, Intel PM 的处理器, 2G 的内存, 80G 的硬盘, 模拟仿真在 Eclipse 平台下使用 Java 语言进行。

本文取  $n=4$ ,  $m=12$ , 约束条件  $C^* = 30$ ,  $r_{\max} = 3$ , 资源属性见表 1。

表 1 资源属性表

属性 编号	稳定性 (S)	价格 (C)
1	0.97	6
2	0.96	7
3	0.94	7
4	0.91	6
5	0.89	6
6	0.86	3
7	0.75	3
8	0.72	4
9	0.70	2
10	0.67	2
11	0.66	2
12	0.65	2

针对该优化模型的求解若采用穷举搜索方法将花费大量的时间,本文用遗传算法来实现该优化模型的求解大大减少了计算时间,得到在给定费用为 30 的情况下,无冗余调度时云计算服务可靠性最大值为 0.797,对应的调度方案为子任务 1 在资源 3 上执行,

子任务 2 在资源 1 上执行,子任务 3 在资源 2 上执行,子任务 4 在资源 4 上执行,此时的服务花费为 26;引入冗余机制后云计算服务可靠性最大值为 0.877,对应的调度方案为子任务 1 在资源 11、资源 7 和资源 5 上执行,子任务 2 在资源 2 上执行,子任务 3 在资源 6 和资源 12 上执行,子任务在资源 1 上执行,与此相应的服务费用为 29,时间费用为 49。由此可见,增加一定的服务花费,引入适当的冗余可以大幅度提高云计算服务可靠性。

## 6 总结

本文在研究了云计算许多独特的特点及其不可靠性后,建立了基于冗余调度的可靠云计算模型,证实了通过加入适量的冗余不仅能保证云计算服务的可靠性,还大大加快了任务的完成时间;并结合了现实生活,在服务费用的约束下建立了冗余调度优化模型,即在用户能承受的负担范围内通过对每个任务加入适量的冗余来提高整个云服务的可靠性,针对该优化问题,实验结果再次验证了遗传算法能以较快的速度求得问题的最优解。

## 参考文献

- 刘鹏.云计算.北京:电子工业出版社,2010.13-14.
- Slawinska M, Slawinski J, Sunderam V. Unibus: Aspects of heterogeneity and fault tolerance in cloud computing. Proc. of IEEE International Conference. USA. 2010.
- Deng J, Huang SCH, Han YSS, Deng JH. Fault-Tolerant and reliable computation in cloud computing. IEEE Globecom 2010 Workshop on Web and Pervasive Security. USA, 2010: 1601-1605.
- 刘宴兵,尚明生,肖云鹏.网格高性能调度及资源管理技术.北京:科学出版社,2010.145-149.
- 刘波,林伟伟,齐德昱.一种冗余调度的可靠网格计算模型.小型微型计算机系统,2010,31(3):515-518.
- 程世娟,卢伟,何平.蚁群算法在冗余系统可靠性最优分配上的应用.计算机工程与应用,2009,45(15):64-66.
- 郭凤昌,黄洪钟,许焕卫,等.失效恢复机制下的网格任务冗余调度优化.机械工程学报,2010,46(23):154-160.
- 汪定伟,王俊伟,王洪峰,等.智能优化方法.北京:高等教育出版社,2007.46-48.
- Gan GN, Huang TL, Gao S. Genetic simulated annealing algorithm for task scheduling based on cloud computing environment. Proc. of IEEE International Conference. Guilin, China. 2010: 60-63.