

相对最小执行时间方差的云计算任务调度算法

李水泉^{1,3}, 邓泓^{2,3}

(1. 深圳大学 计算机与软件学院, 广东 深圳 518060;

2. 江西农业大学 软件学院, 江西 南昌 330045;

3. 江西省高等学校农业信息技术重点实验室, 江西 南昌 330045)

摘要:随着越来越多的计算任务投入到云计算系统中,如何实现较好的任务调度和分配对于任务完成时间以及云计算系统的负载问题具有重要的作用。为使云计算平台的任务调度有较好的负载均衡和较早的最早完成时间,提出相对最小执行时间方差的云计算调度算法 min-variance。由于各计算资源执行完任务的执行时间方差能够在一定程度上反映负载均衡和最早完成时间的问题,因此算法利用任务在各个计算资源之间的位置不同能产生不同执行时间方差的事实,通过一定规则和次数的任务位置变换以达到相对最小执行时间方差,从而在负载均衡和最早完成时间上都达到较好的效果。在 CloudSim 仿真平台上进行了实验,结果表明,与现有一些调度算法相比, min-variance 算法不仅具有较好的负载均衡,同时也有较早的最早完成时间。

关键词:云计算;任务调度;执行时间;方差

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2018)07-0034-04

doi: 10.3969/j.issn.1673-629X.2018.07.008

Min-variance Tasks Scheduling Algorithm for Cloud Computing System

LI Shui-quan^{1,3}, DENG Hong^{2,3}

(1. School of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, China;

2. School of Software, Jiangxi Agricultural University, Nanchang 330045, China;

3. Key Laboratory of Agricultural Information Technology of Jiangxi College, Nanchang 330045, China)

Abstract: As more and more computing tasks are put into the cloud computing system, how to achieve better task scheduling and allocation plays an important role for load balancing and earliest completion time in cloud computing system. In order to get better load balancing and earliest completion time in task scheduling of cloud computing platforms, we propose a task scheduling algorithm on cloud computing platforms with relative minimum variance of execution time, called min-variance. Because the variance of execution time of the execution tasks from each computing resource can reflect the load balancing and earliest completion time to a certain extent, this algorithm moves the tasks to other computing resources within a rule and a certain number of times to achieve relative minimum variance of execution time, based on the fact that it can produce different variances of execution time when the tasks move to other computing resources. The algorithm is experimented on the simulation platform CloudSim, which shows that the min-variance algorithm not only can get better load balancing, but also achieve the earliest completion time compared with some existing scheduling algorithms.

Key words: cloud computing; task scheduling; execution time; variance

0 引言

云计算是近年来新兴的一种商业计算模型^[1],它将用户提交的任务分布到由大量计算资源构成的资源池中,在计算完成后将结果返回给用户。由于云计算具有计算能力强、价格低廉、使用简单等特点,使得云

计算用户的数量迅速增加,同时也使计算任务数量与日剧增。如果对数量庞大的任务调度不合理,将导致完成任务的时间长以及计算资源的浪费。针对任务调度属于 NP 难^[2]的问题,且任务数量庞大时不能通过暴力求解得到最早的任务最早完成时间和负载最均

收稿日期:2017-05-24

修回日期:2017-10-27

网络出版时间:2018-03-08

基金项目:江西省科技支撑计划项目(20141BBF60051);江西省科技指导项目(2010ZDG05600);江西农业大学大学生创新创业训练计划项目(201610410063)

作者简介:李水泉(1993-),男,硕士研究生,研究方向为时空大数据;邓泓,硕士,副教授,研究方向为农业信息化与图像处理。

网络出版地址: <http://cnki.net/kcms/detail/61.1450.tp.20180307.1416.010.html>

衡^[3]策略,研究如何合理调度云计算任务,以达到较好的最早完成时间和负载均衡,是具有重要意义的。目前,针对最早完成时间或负载均衡提出的云计算任务调度算法主要有以下几种:传统网格计算中的调度算法,如 min-min 算法^[4]和 max-min 算法^[5]。这两种算法都采用贪心策略,将最小或最大的任务分配到可以最早完成它们的机器上,实现局部最小完成时间,但它们的最早完成时间一般不是最好的,同时 min-min 算法没有考虑负载均衡;现有成熟的云计算平台的调度算法,如 Hadoop 中的默认任务调度算法^[6]。该算法使用 FIFO 队列策略,采用较公平的原则,将先提交的且优先级高的任务分配到空闲的机器上,使得在最后一个任务分配前所有机器都处于工作状态,但它没有考虑最早完成时间问题;启发式智能算法,如遗传算法、粒子群算法、蚁群算法及对它们的混合算法等^[7-13]。这些算法虽然在选取了合适参数和迭代次数的基础上能获得较优的解,但是算法复杂,在实际的实现过程中参数和迭代次数的选取困难,并且它们对于大量任务的处理时间较长,因此算法的稳定性不高。

针对上述算法的不足之处,考虑如何调度任务以达到较早的最早完成时间和较好的负载均衡,文中提出相对最小执行时间方差的云计算任务调度算法 min-variance。该算法在 max-min 算法分配任务的基础上,利用任务位置的转移能够改变各计算资源的执行时间方差,增加了有限次数的任务在计算资源上的位置转移,从而达到相对最小的执行时间方差,实现最早完成时间较早和负载较均衡。

1 云计算任务调度问题

目前,云计算的任务调度问题在云计算系统中具有重要的研究意义。在云计算系统中,存在一个若干不同性能的计算资源构成的资源池,云计算的任务调度即是从资源池中选择合适的计算资源处理不同长度的计算任务。随着云计算技术的成熟与发展,云计算任务数量大量增加,因此将云计算任务合理分配给各计算资源对提高处理效率、节省电力等具有重要作用。

文中研究的目的是实现负载较均衡和达到较早的最早完成时间,在现有的计算任务调度算法中,无论是传统的贪心算法,如 min-min 和 max-min 算法,还是智能型启发式算法,如遗传算法和蚁群算法等,其计算结果往往不是最优。这是因为贪心策略的思想是从局部最优解得到全局最优解,这样的解并不是最优的;而智能启发式算法大多是从全局最优出发,进而得到调度的解,这种方法理论上能得到最优解,但是收敛速度却不能控制,因此其解也不是最优。所以在得到其任务调度结果后,往往很容易找到一个任务,转移这个任

务到另一个计算资源后将得到更早的最早完成时间,或者使各计算资源的任务完成时间更加接近,即得到一个更优的解,文中即根据这一事实提出 min-variance 算法,通过一定次数变换任务在计算资源的位置从而得到完成时间的相对最小方差,进而求得最优解。

2 min-variance 算法

2.1 算法的体系架构

min-variance 算法在 CloudSim^[14]上实现。CloudSim 是由墨尔本大学推出的云计算仿真软件,主要用于模拟云计算基础架构和服务,在云计算的研究中广泛使用,因此在该仿真软件上实现的 min-variance 算法可用于真实的云计算环境中。min-variance 算法在 CloudSim 工作模型中的关系如图 1 所示。

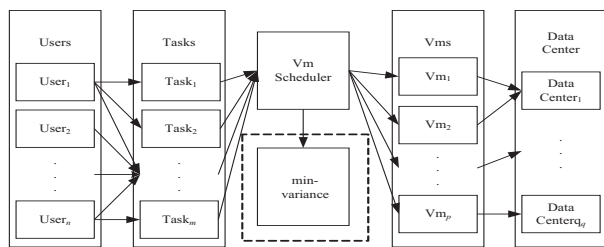


图 1 min-variance 算法与 CloudSim 工作模型关系图

图 1 中左边两列表示每个云计算用户提交若干个计算任务,右边两列表示虚拟机与数据中心的映射关系(虚拟机存在于硬件层之上,便于对硬件的操作),中间列表示通过一个调度器进行任务调度,使任务与虚拟机发生映射,而调度器使用的调度算法即 min-variance 算法,在算法执行完后,得到任务的分配调度结果,再由虚拟机调度任务。

2.2 相对最小执行时间方差

将 n 个任务划分到 k 个计算资源中,则每个计算资源有一个任务集合,计算资源执行完任务集合中的所有任务有一个执行时间 t ,所有的 t 构成集合 T ,执行时间方差就是集合 T 的方差。因为方差衡量的是一组离散数据的偏离程度,所以执行时间的方差在一定程度上反映负载均衡的问题,即执行时间的方差越小,说明负载越均衡。文中算法就是在初始执行时间方差的基础上,不断减小方差,以达到负载较均衡和较早完成时间。执行时间方差的计算公式如下:

$$E(t) = \frac{1}{k} \sum_{i=0}^{i=k} t_i \quad (1)$$

$$D(T) = \frac{1}{k} \sum_{i=0}^{i=k} (t_i - E(t_i)) \quad (2)$$

任务调度问题是一种组合问题,通过计算所有组合可以找到一个最小的执行时间方差,但任务数量往往较大,不可能通过暴力求解所有组合,因而也不可能计算出执行时间方差的最小值,所以文中算法的执行

结果只能在某一区间内最小,该区间值域由任务、计算资源和算法共同决定。

经过分析发现,执行时间方差的值可以由以下两种方式改变:将一个计算资源的任务集合中的一个任务放到另一个计算资源中;将一个计算资源的任务集合中的一个任务与另一个计算资源任务集合中的一个任务进行位置变换。因此,通过这两种方式可以使方差不断减小。另外,通过这两种方式减小的方差一定能在某个值终止,这是因为任务和计算资源在计算机上都是离散的且数量是一定的,因此不可能使方差一直减小,即通过一定次数的迭代找到最小的方差。基于以上分析,min-variance 算法具有可行性,并能够达到负载的较均衡和较早的最早完成时间。

2.3 max-min 算法

min-variance 算法的核心部分在于有限次数的任务位置转移从而达到相对最小的执行时间方差,因此任务的初始化分配很大程度决定了算法核心部分的运算时间,所以采用合适的方法进行任务的初始化分配具有重要作用。与其他初始化任务分配方法相比,max-min 算法同时具备以下几个优势:算法复杂度不高,能够快速进行初始化;先将较大的任务分配到能够最早完成它的计算资源上,这种方式能够实现较好的负载均衡;采用贪心策略以取得较早的完成时间。这三点优势能够使得 min-variance 不但能快速完成初始化分配,还能在算法核心部分运算之前就有了较好的负载均衡和较早的最早完成时间,从而减小了后续的计算量,因此 min-variance 使用 max-min 算法作为任务的初始化分配方法。

max-min 算法采用贪心策略,先将较大的任务分配到能够最早完成它的计算资源上。首先将任务按指令长度由大到小进行排列,同时将计算资源按计算速度由小到大排列,再计算出各个任务在各虚拟机上的执行时间。用二维数组 $\text{Time}[j][i]$ 表示第 j 个任务在第 i 个计算资源的执行时间,MI 表示任务指令长度,MIPS 表示计算资源的执行速度,计算方法如下:

$$\text{Time}[j][i] = \text{MI}[j] / \text{MIPS}[i] \quad (3)$$

在计算出 Time 后,从任务 0 开始,依次将各个任务加入到能够最早完成该任务的计算资源的任务列表中。

2.4 算法流程

假设有 n 个任务将分配到 k 个计算资源中,令 $\text{tasknum}(i)$ 表示计算资源 i 中的任务数量,则 min-variance 算法流程如下:

(1) 用 max-min 算法初始化任务分配。

(2) 利用式 1 和式 2 计算执行时间方差 $D(T)$, 记为 \min_dt ,同时计算最早完成时间 \min_v 。

(3) 依次选取计算资源 i 中的任务 j , 记为 task。若 $i = k - 1, j = \text{tasknum}(i)$, 则转到步骤 7。

(4) 将 task 加入到计算资源 $l(0 \leq l < k, l! = i)$ 中,计算每次加入的新执行时间方差 $D(T)$ 和最早完成时间 v , 若 $D(t) < \min_dt \& \& v \leq \min_v$, 则更新最好情况 resu 为将 task 加入到计算资源 l , 同时更新 \min_dt 为 $D(t)$, \min_v 为 v 。

(5) 将 task 与计算资源 $l(0 \leq l < k, l! = i)$ 中的任务 $h(0 \leq h < \text{tasknum}(l))$ 互换位置,计算每次互换位置后的新执行时间方差 $D(T)$ 和最早完成时间 v , 若 $D(t) < \min_dt \& \& v \leq \min_v$, 则更新最好方法 resu 为将 task 与计算资源 l 中的任务 h 互换位置,同时更新 \min_dt 为 $D(t)$, \min_v 为 v 。

(6) 转到步骤 3。

(7) 若 resu 为空时,则算法结束,否则重复步骤 3~6。

min-variance 算法的程序流程如图 2 所示。

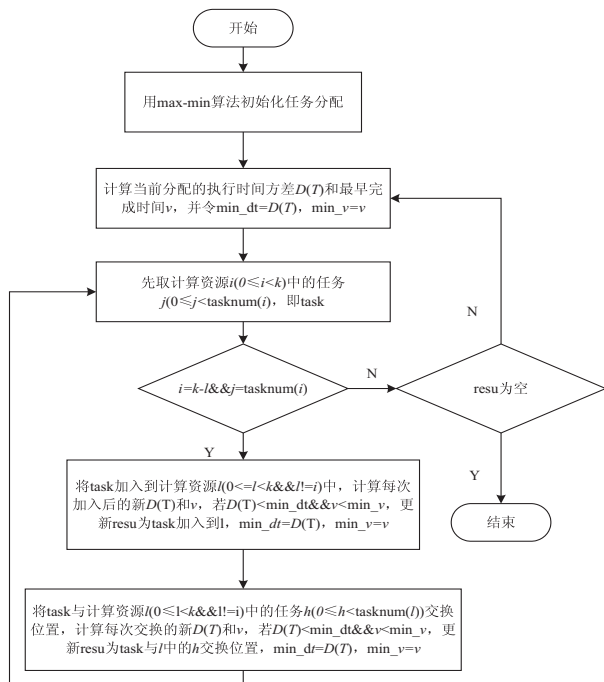


图 2 min-variance 算法的程序流程

3 实验结果

实验在 CloudSim 仿真器上进行。由于现实条件下各种云计算任务数量的差别较大,同时各任务指令长度的长短差距也很大,为了更好地模拟真实条件下的云计算任务调度,实验设置了 5 组不同数量的任务,使用同一组虚拟机 (CloudSim 中虚拟机的执行速度相当于真实云环境中计算资源的执行速度) 分别对其进行处理。其中虚拟机共有 8 个,执行速度 MIPS 都在 10 到 1 000 之间,而任务的长度 MI 都在 100 到 100 000 之间。实验结果如表 1 和表 2 所示。

表 1 不同数量的任务在同一组虚拟机上的最早完成时间

调度算法	任务数				
	20	40	60	80	100
min-min	365	732	1 136	5 117	6 269
蚁群算法	494	625	1 128	4 843	6 241
遗传算法	329	839	1 538	7 632	9 970
max-min	277	627	1 038	4 556	5 919
文中	269	620	1 032	4 549	5 891

表 2 各虚拟机执行时间的最大差

调度算法	任务数				
	20	40	60	80	100
min-min	365	732	400	3 235	2 073
蚁群算法	494	327	495	875	875
遗传算法	329	441	1 020	4 369	6 164
max-min	73	14	40	66	63
文中	17	6	9	17	7

与 min-variance 算法相比较的其他四种算法都属于以最早完成时间或负载均衡为目标的任务调度算法,其中蚁群算法的蚂蚁个数设置为任务数,迭代次数为 35,遗传算法交叉概率设置为 0.25,变异概率设置为 0.04,迭代次数为 100。从表 1 和表 2 可以看出,min-min、蚁群和遗传算法进行的任务调度结果互有优劣,但与 max-min 和 min-variance 算法相比,虽然在任务数为 40 时蚁群算法的最早完成时间较 max-min 算法好,但在其余情况下最早完成时间较晚,且各虚拟机执行时间的最大差较大,这是因为智能型的蚁群和遗传算法受参数影响较大,且由于它们没有考虑负载均衡,使得负载不均衡。min-min 算法优先放置较短的指令到能最快执行完它的计算资源上,不但使得负载不均衡,而且当出现有某个任务远长于其他任务时则表现出最早完成时间较晚。min-variance 算法是在 max-min 的基础上继续提前最早完成时间和实现更好的负载均衡,由数据可以看出,min-variance 算法在最早完成时间上都早于 max-min 算法,而随着任务的增长,所提前的最早完成时间可能更多,并且其虚拟机执行时间的最大差比 max-min 算法小很多,实现了较好的负载均衡。因此与其他四种算法相比,min-variance 算法在最早完成时间和负载均衡上都有更好的效果。

4 结束语

云计算任务调度算法对于云计算的计算效率具有重要影响,而任务调度问题是一个 NP 难问题,因此只有不断探索更高效的任务调度算法才能应对不断增长的

云计算用户和任务数量。为实现更早的最早完成时间和负载较均衡,提出了 min-variance 算法,在 max-min 算法的基础上不断缩小各虚拟机执行时间的方差,取得了较好的效果。

min-variance 算法虽然能够实现较好的最早完成时间和负载较均衡,但仍有许多工作需要进一步研究:现实中的云计算资源在内存等方面各不相同,因此需要在文中研究的基础上更真实地模拟实际中的云环境;一些任务存在依赖问题,故需要添加任务的依赖关系。

参考文献:

[1] 刘 鹏. 云计算[M]. 第 2 版. 北京:电子工业出版社, 2011.

[2] TAYAL S. Task scheduling optimization for the cloud computing systems [J]. International Journal of Advanced Engineering Sciences and Technologies, 2011, 5(2): 111-115.

[3] 顾崇林. 虚拟机集群负载均衡的研究[D]. 哈尔滨:哈尔滨工业大学, 2010.

[4] PATEL G, MEHTA R, BHOI U. Enhanced load balanced min-min algorithm for static meta task scheduling in cloud computing [J]. Procedia Compute Science, 2015, 57: 545-553.

[5] 史少锋, 刘宴兵. 基于动态规划的云计算任务调度研究 [J]. 重庆邮电大学学报:自然科学版, 2012, 24(6): 687-692.

[6] SHARMA S, SINGH M. A Modified FIFO with distributed and pipelining scheduling in Hadoop [J]. Imperial Journal of Interdisciplinary Research, 2016, 10(2): 1839-1842.

[7] 于国龙, 崔忠伟, 左 羽. 基于离散粒子群优化的 MPSoC 节能调度算法 [J]. 江西师范大学学报:自然科学版, 2016, 40(3): 307-311.

[8] XUE Shengjun, LI Mengying, XU Xiaolong, et al. An ACO-LB algorithm for task scheduling in the cloud environment [J]. Journal of Software, 2014, 9(2): 466-473.

[9] 王 波, 张晓磊. 基于粒子群遗传算法的云计算任务调度研究 [J]. 计算机工程与应用, 2015, 51(6): 84-88.

[10] 曹 阳, 刘亚军, 俞 琰. 基于遗传-蚁群算法的云计算任务调度优化 [J]. 吉林大学学报:理学版, 2016, 54(5): 1077-1081.

[11] GUO Lizheng, ZHAO Shuguang, SHEN Shigen, et al. Task scheduling optimization in cloud computing based on heuristic algorithm [J]. Journal of Networks, 2012, 7(3): 547-553.

[12] 查安民, 谭文安. 融合粒子群与蚁群的云计算任务调度算法 [J]. 计算机技术与发展, 2016, 26(8): 24-29.

[13] 张兴国, 周东健, 李成浩. 基于粒子群-蚁群融合算法的移动机器人路径优化规划 [J]. 江西师范大学学报:自然科学版, 2014, 38(3): 274-277.

[14] 何倩媛. 云计算仿真工具 CloudSim 的研究与应用 [J]. 科技资讯, 2016, 14(2): 32-33.