

网格依赖任务重调度资源选择算法研究

孟湘来

(天津广播电视大学,天津 300191)

摘要:为网格依赖任务选择稳定资源是减少重调度触发频繁的关键。针对稳定资源选取的问题,文中提出了资源成功执行任务标记方法,依据标记值对资源进行动态分级,在此基础上提出了基于税收机制的成功标记重调度方法。在该方法中引入了税收征收不同税率的思想,对资源的标记值按大小分段,根据标记值来为任务分配当前较稳定的资源。对于不同分段中的标记值,资源在成功完成一项任务或失败时,对标记值的增加或减少程度会不同。实验结果表明,该方法可以提高资源利用率及负载均衡,有效提高资源稳定性判断的准确性,从而减少重调度触发的频率。

关键词:网格;重调度;成功标记;依赖任务

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2015)09-0169-05

doi:10.3969/j.issn.1673-629X.2015.09.036

Research on Resource Selection Algorithm in Grid Dependent Tasks Rescheduling

MENG Xiang-lai

(Tianjin Radio & TV University, Tianjin 300191, China)

Abstract: Selecting a stable resource for grid-dependent task is the key issues to reduce the frequency of tasks rescheduling. Aiming at this problem, present a marking method that count the number of resources to perform tasks successfully. Based on this tag value classify resource dynamically. Then put forward the improved method based on the tax mechanism successful index rescheduling method. This method uses different tax rates for different tag value range, segmenting the tag values of resources according to the size, allocating the relatively stable resources for task according to the current tag values. For the tag value in different segments, when resources are in the successful completion of a task or fails, the tag value will be different for increasing or decreasing degree of tag values. Experiment results show that the method can make the resources fully utilized and improve load imbalance, effectively improving the accuracy of current stability judgment. The possibility of rescheduling trigger is reduced.

Key words: grid; rescheduling; successful index; dependent tasks

0 引言

网格计算中存在具有依赖关系的任务,对网格任务调度提出了挑战。进行依赖任务调度时,由于任务间依赖关系的存在,某一项任务的资源分配将会影响其他任务的资源分配。因此,为了提高整体应用的性能,在网格依赖任务时,采用能够进行全局优化的静态调度的策略^[1-7]。全局优化的静态调度策略实际是基于预知的应用、资源信息在运行前所制定一个全局调度计划;在资源执行任务期间发生错误或资源性能变化不能按期完成任务时进行重新调度。

但是,网格中的资源具有动态性的特征,资源性能

会随时间变化而变化,而且还有可能在任务执行期间发生资源失效。这种情况都会造成资源不能按期完成任务。为了保证任务的正常完成,必须频繁进行重调度。而频繁的重调度将会严重加大任务调度过程中总的执行代价,而重调度发生在应用的运行时刻,势必会对应用性能造成较为明显的影响。为了保证应用过程的整体性能,必须解决依赖任务调度中重调度触发频繁问题^[8-10]。

在资源规模和任务规模很大时,采用启发式算法来提高重调度的执行效率的同时,还可以通过提高资源稳定性以减少重调度触发的频率。因此,在依赖任

务调度时,从备选资源中选择稳定性高的资源是一种减少重调度触发频繁的有效方法,可以提高应用的整体性能^[11-17]。

基于以上分析,文中介绍了成功标记重调度方法,通过为重调度任务选择备选资源中最稳定的资源的方式来降低重调度发生的频率;提出了基于税收机制的成功标记重调度方法,可以在成功标记重调度方法的基础上进一步提高对资源稳定性预测的准确度,从而达到进一步减少重调度触发频率的目的,同时可以使资源负载均衡。

1 相关工作

从静态调度方法在网格环境下依赖任务调度所面临的问题来看,现有的研究大致可分为两种不同的解决方法。一种是以提高预测精度、多资源备份为目标。Plan Switching 方法^[1]就属于此种,该方法是一种以多个备用方案来提高静态调度计划可行性与最优性的方法。它是在应用执行前构建一系列的活动图,每个活动图即代表一个可行的调度计划,在运行过程中如若一个活动图不能正常执行,它将会切换到其他活动图。该方法由于需要获得真实运行时的任务、资源信息,代价较大。因此,文中不采用。

另一种解决方法的思路是通过减弱对预测精度依赖、适应动态性为目标。主要是通过重调度这样的计划调整策略为手段,具体算法包括:

MQD、SIL 算法^[2]对计算密集型和数据密集型应用都有很好的表现,在一定程度上解决了静态调度算法对性能预测精确性的依赖问题,但是该算法是面向 Bag-of-Tasks 类型的独立网格应用,它不能应用于具有依赖关系的任务调度。

Low-Cost 重调度策略^[3]所面向的依赖任务调度问题,需要在应用运行时的一些关键点上考虑重调度,其主要目的就是解决静态全局调度对任务预测的不精确而造成的契约冲突问题,因此,它没有考虑资源的变化。

AHEFT 算法^[5]在处理高并行度的数据密集型应用时会有较好体现。它考虑了资源的多种变化和性能预测精度对调度最优性的影响,并按照 HEFT 的启发式算法实现了调度计划调整的策略,但该方法并没有考虑到重调度触发频繁问题。

FIBR 算法^[7]是一种基于错误标记的重调度算法。针对重调度触发频繁问题,在进行依赖任务重调度资源选择时选择最稳定的资源,从而减少了重调度触发的频率。在该算法中,为每个可用资源设定一个错误标记 tag,将它存储于信息服务器 IS 中。tag 值用于标识一个资源的稳定性。tag 值越低,系统认为该资源的

稳定性越好。当一个运行着某一个任务的资源发生错误时,就要对运行在其上的任务进行重调度。其算法如下:

(1)用户在提交任务时,要给定该任务必须完成的最终期限和预计的执行时间。系统以此来为任务选取相应的资源。当任务分配给一个资源,资源适配器需要一个时间间隔后收到任务执行情况的响应。这个时间间隔要由资源的处理速度和资源与资源适配器之间的通信延时决定。

(2)如果资源不能在网格管理器给定的时间间隔内完成任务,就认为资源发生了错误。那么该资源的标记 tag 值加 1,相反如果能顺利完成该任务,tag 值减 1。

(3)当资源发生了错误,运行在出错资源上的任务需要重调度。此时对可用资源的选择就要根据各资源的信息服务器中存放的 tag 值来判断。根据 tag 值,任务分配另一个最稳定的备选资源上。

通过以上算法,在对任务进行重调度时,对备选资源进行了选择。选择最稳定的任务来执行,可以增大任务顺利完成的概率,从而减小了发生重调度的可能性。但是这种算法没有考虑到资源的负载均衡,并且在系统对资源性能变化做出快速判断方面并不理想。

综上所述,在现有的依赖任务重调度方法的研究上,对重调度所面临的触发频繁问题,仍未有一个最优的解决方案。

文中提出的基于税收机制的成功标记重调度方法是对 FIBR 算法的改进,可以在 FIBR 方法的基础上进一步提高对资源稳定性预测的精度,从而达到进一步减少重调度触发频率的目的,同时可以使资源的负载均衡。

2 基于税收机制的成功标记重调度

首先,给出文中的网格组成结构(见图 1)。

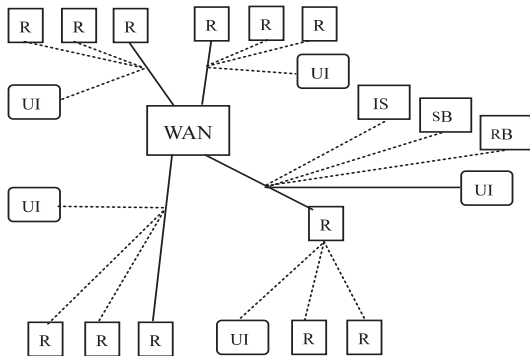


图 1 网格结构图

其中,R 为各站点资源;UI 为用户接口;IS 为信息服务器;SC 为调度器;RB 为资源适配器。

地理上分布的各站点上有多个计算资源 R,在它

们的末端是 UI,任务通过 UI 提交给网格系统;RB 用于发现所有的可用资源;SC 用于将任务调度给可用资源;IS 用于存储调度器需要的任务和资源的状态信息,它存储着每个资源的历史信息。文中认为 SC、RB 和 IS 都是非常稳定的,只有 R 是动态的。

借鉴 FIBR 算法的思想,文中提出了成功标记重调度算法。为了方便后面的计算,这里记录资源状态 tag 值的方法与 FIBR 算法中记录 tag 值的方法相反,即:如果资源能在网格管理器给定的时间间隔内完成任务,就认为资源能够成功完成任务,那么该资源的标记 tag 值加 1;相反如果不能完成该任务,tag 值减 1。对于以上介绍的成功标记重调度算法,优点是非常简单,但却存在着以下不足:

(1)当采用这种成功标记来表示各资源的稳定性时,由于资源性能上存在的差异经过标记 tag 值的累计会使整体可用资源根据 tag 值的大小而形成一种接近菱形的结构。每次发生重调度时,由于系统会选择 tag 值较大的资源为其分配任务,就会造成位于菱形顶端的极少数资源负载过重。而这些资源负载过重又会影响这些资源的性能,从而影响正在处理的任务的顺利完成,反而会增加这些资源出错的可能性。与此同时,位于菱形中上部的大量资源却因分配不到任务而造成资源的浪费。

(2)使用一个性能随时间变化的动态资源的最理想效果是:在资源的性能处于最佳状态时为其分配任务。然而资源的稳定性是根据历史记录来判断的,所以不可能达到这种理想的情况,只能想办法接近。这就要求对资源性能的变化要做出快速反应。也就是说,对于一个稳定性评价很高的资源,在它的性能处于不稳定状态时应该使系统对其尽快做出新的判断,将新到的任务分配给当前更稳定的资源;对于一个稳定性评价处于菱形虚线以上、阴影部分以下的稳定性较高资源,在其性能处于非常稳定时,系统也能尽快对其做出新的判断,为其分配任务。但文中的成功标记方法在这方面做的并不理想。

考虑以下两种情况:

(1)资源 R 执行任务情况(1)如图 2 所示。

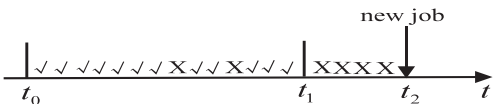


图 2 资源 R 执行任务情况(1)

(对号表示 R 执行任务成功,叉号表示失败)
这是某资源 R 在 t_0 时刻到 t_2 时刻执行任务的情况。从 t_0 到 t_1 ,R 完成任务的数量远大于失败的数量,所以它的 tag 值会非常高,这使它位于菱形的顶端。而在 t_1 到 t_2 时间段内,假定 R 由于某些原因性能下

降,导致连续做任务失败,但由于之前 tag 的累积值非常大,即使连续失败导致 tag 值减小,也不会改变系统对它稳定性的判断。由于 tag 值依然很大,所以在 t_2 时刻新任务到来时,系统依然会优先选择资源 R 来执行,而此时 R 并不适合执行任务。这样会增加重调度产生的可能性。

(2)资源 R 执行任务情况(2)如图 3 所示。

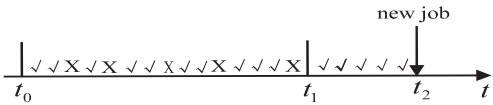


图 3 资源 R 执行任务情况(2)

这是某资源 R 在 t_0 到 t_2 时刻执行任务的情况。从 t_0 到 t_1 ,R 成功完成任务的情况并不是非常理想,所以它的 tag 不会特别高,这使它处于菱形的虚线以上、阴影部分以下。而在 t_1 到 t_2 之间,假定 R 性能很稳定,连续成功完成任务。但由于之前 tag 的累积值并不高,即使连续成功导致 tag 值增大也不会改变系统对其稳定性的判断,新任务到达依然不会优先考虑它。而 R 此时很适合执行这次任务。

通过以上两种情况可以看出,成功标记重调度方法虽然可以根据历史记录对资源的总体性能给出判断,但对资源的动态性变化的判断过慢。用简单的 tag++或 tag--并不能让系统对资源的性能变化做出尽快的反应。这里提出一种改进的成功标记重调度方法—基于税收机制的成功标记方法。在该方法中引入税收征收不同税率的思路,对资源的 tag 值按大小分段。对于不同分段中的 tag 值,资源在成功完成一项任务或失败时,对 tag 值的增加或减少程度不同。处于 tag 值越高分段中的资源,成功完成一次任务其 tag 值增加的程度越小,而执行失败时,tag 值减小的程度越大;处在 tag 值越小的分段中的资源,成功完成一次任务其 tag 值增加的程度越大,而执行失败时,tag 值减小的程度越小。

其算法如下:

(1)用户在提交任务时,要给定该任务必须完成的最终期限和预计的执行时间。系统以此来为任务选取相应的资源。当任务分配给一个资源,资源适配器需要一个时间间隔后收到任务执行情况的响应。这个时间间隔要由资源的处理速度和资源与资源适配器之间的通信时延决定。

(2)将可用资源的 tag 值按大小分成若干段。如果资源不能在给定时间间隔内给出结果,那么认为执行失败。判断该资源的 tag 值位于哪一分段中,进而为 tag 减掉相应的值,然后重新判定 tag 值属于哪一分段。而在给定时间内执行成功,根据 tag 所在的分段增加相应的值,然后重新判定 tag 值属于哪一分段。

(3)当资源发生了错误,运行在出错资源上的任务需要重调度。此时对可用资源的选择就要根据各资源的信息服务器中存放的 tag 值来判断。根据 tag 值,任务重调度到另一个系统认为最稳定的备选资源上。

通过这种方法,可以达到资源负载均衡的目的,使大部分较稳定资源都能得到充分利用,而且可以有效加快系统对资源性能变化情况的判断。使系统能够尽快在资源稳定的时候增加分配任务的机会,在不稳定的时候减少分配任务的机会。进一步减少了重调度发生的可能性。

3 实验环境

文中使用文献[8]中所设计的方案进行模拟实验。它是通过使用网络中间件 Globus Toolkit4.0 设计的模拟环境。任务、资源的生成则是借鉴文献[10]所提出的方法,按照特征参数随机生成测试用例,包括任务图以及资源环境,在资源环境模拟上做了更进一步的完善,通过资源动态性模拟器来体现资源的动态性。

为了减少实验的复杂性和突出核心问题,本实验进行如下假定:

- 忽略掉资源的网络延迟;
 - 不涉及其他应用对资源共享冲突的因素,考虑资源的可用时间段为 $[0, \infty)$;
 - 假设任务图具有单一入口、出口节点。
- DAG 任务图与资源环境随机生成过程中所需的特征参数具体有:

- V :任务图中的任务个数。
 - α :并行度因子(即任务图形状控制参数)。任务图的高度(H)则是由均数为 \sqrt{v}/α 的正态分布随机生成的,在此对 H 取整运算。
 - out_degree:即任务节点出边的最大值,用 v 的百分比来表示。节点出边是在 $[1, out_degree]$ 区间内随机生成。
 - CCR:即平均通信花费与平均计算花费之比,表明应用是数据密集型、计算密集型的一个参数。
 - β :资源能力异构因子。 β 取值在 $[0, 2)$ 区间内变化,其值越高表明资源处理能力的差异越大。当 $\beta = 0$ 时表示资源同构。
 - γ :资源动态因子,资源变化时间间隔在 $[0, \gamma]$ 之间随机选择。
 - δ :网络性能异构因子。
 - n :初始应用可用的资源数量。
 - INR:可用资源加入事件占有所有事件的比例。文中所考察的是可用资源加入与退出事件。
- 文中所模拟实验中的参数设定如表 1 所示。

表 1 任务图、资源随机生成参数表

参数	值
V	10, 20, 50
α	0.5, 1, 2
out_degree	0.1, 0.5, 1
CCR	0.1, 0.5, 1, 5, 10
β	0.2, 0.6, 1, 1.4, 1.8
δ	2, 3, 5, 8, 10
γ	100, 50, 20, 10, 5
n	10, 20, 50
INR	0, 0.2, 0.25, 0.8, 1

4 实验分析

为了验证提出的基于税收机制的成功标记重调度算法的有效性,文中采用文献[8]中提出的基于 IHEFT 重调度算法对其进行验证。将此算法应用于 IHEFT 算法中,记为 M-IHEFT 算法。并将 M-IHEFT 算法与经典的动态调度算法 Max-Min、Min-Min、Max-Int 进行对比,实验结果如图 4 所示。

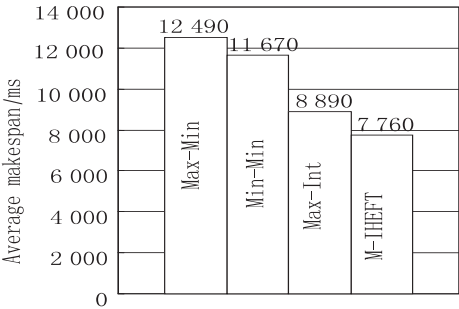


图 4 M-IHEFT 算法与经典的动态调度算法性能对比

通过比较,在基于税收机制的成功标记方法的支持下,静态调度策略的性能优势依然明显,整体上验证了基于税收机制成功标记方法的支持下网格依赖任务重调度机制的有效性。

IHEFT 算法的性能优势在于基于税收机制的重调度算法降低了重调度触发的频率。文中对 IHEFT 算法和没有使用税收机制重调度方法情况下的 IHEFT 算法(M-IHEFT)在性能方面做了比较,如图 5 所示。其中, r 为资源动态因子,资源性能变化时间间隔在 $[0, r]$ 之间随机选择。

实验结果表明,随着资源动态性的增强,应用的性能受到了较大影响。IHEFT 算法与没有税收成功标记的 M-IHEFT 算法相比,性能下降幅度要小。这是因为在资源动态性增强的情况下,以往的算法会因为重调度的频繁发生而使性能受到影响。而 M-IHEFT 算法中由于使用了基于税收机制的成功标记重调度算法,

有效减少了重调度的触发频率,使其性能达到了提升。

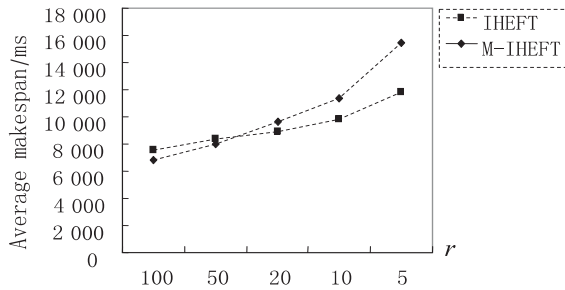


图5 M-IHEFT 算法与没有使用税收机制重调度算法性能比较

5 结束语

文中针对网格依赖任务重调度目前面临的触发频繁问题,介绍了一种简单的成功标记重调度算法。随后提出了它的改进方法—基于税收机制的成功标记重调度方法。通过该方法,网格系统可以根据成功标记tag 的值为任务分配当前较稳定的资源。同时该方法可以使大部分可用资源得到充分利用,大大减少了资源负载不均衡情况的出现,有效提高了利用历史记录对资源当前稳定性情况判断的准确性。实验结果也表明,该方法能够在动态网格条件下,使静态调度策略保持对动态调度策略性能的比较优势,同时也验证了此方法在解决处理网格依赖任务重调度时所面临的触发频繁问题上的有效性问题。

参考文献:

[1] Yu H,Marinescu D C,Wu A S,et al. Plan switching:an approach to plan execution in changing environments[C]//Proceedings of the 2006 international parallel and distributed processing symposium. [s. l.]:[s. n.],2006:33-41.

[2] Lee Y C,Zomaya A Y. Practical scheduling of bag-of-tasks applications on grids with dynamic resilience[J]. IEEE Transactions on Computers,2007,56(6):815-825.

[3] Sakellariou R,Zhao H. A low-cost rescheduling policy for efficient mapping of workflows on grid system[J]. Scientific Programming,2004,12(4):253-262.

[4] Imamagic E,Radic B,Dobrenic D. An approach to grid scheduling by using condor-g matchmaking mechanism[J]. Journal of Computing and Information Technology,2006,14(4):329-

336.

[5] Yu Zhifeng, Shi Weisong. An adaptive rescheduling strategy for grid workflow applications parallel and distributed processing symposium [C]//Proceedings of the 2007 international parallel and distributed processing symposium. Long Beach, CA:IEEE,2007:1-8.

[6] Sulistio A, Schiffmann W, Buyya R. Advanced reservation-based scheduling of task graphs on clusters[C]//Proceedings of the 13th international conference on high-performance computing. [s. l.]:[s. n.],2006:60-71.

[7] Therasa A L S,Sumathi G, Antony D S. Dynamic adaptation of checkpoints and rescheduling in grid computing[J]. International Journal of Computer Applications,2010,2(3):95-99.

[8] 陈廷伟,郝宪文,张斌,等. 资源动态组织支持的网格依赖任务重调度机制[J]. 小型微型计算机系统,2009,30(12):2389-2397.

[9] 郝宪文,代钰,张斌,等. 可迁移网格依赖任务重调度模型及算法[J]. 沈阳工业大学学报,2008,30(1):81-89.

[10] Topcuoglu H, Harir S, Wu M Y. Performance-effective and low complexity task scheduling for heterogeneous computing [J]. IEEE Trans on Parallel and Distributed Systems,2002,13(3):260-274.

[11] 张伟哲,刘欣然,云晓春,等. 信任驱动的网格作业调度算法[J]. 通信学报,2006,27(2):73-79.

[12] 刘宴兵,陈杰,熊仕勇. 基于 QoS 相似度的网格任务调度算法[J]. 重庆邮电大学学报:自然科学版,2009,21(3):416-420.

[13] 童毅,杨济安. 网格机制下的一种光网络资源调度策略[J]. 重庆邮电大学学报:自然科学版,2008,20(6):673-677.


[14] 杜晓丽,蒋昌俊,徐国荣,等. 一种基于模糊聚类的网格 DAG 任务图调度算法[J]. 软件学报,2006,17(11):2277-2288.

[15] 袁禄来,曾国荪,姜黎立,等. 网格环境下基于信任模型的动态级调度[J]. 计算机学报,2006,29(7):1217-1224.

[16] 陈晶,孔令富,潘勋. 结合预测机制的 QoS 约束的网格资源调度算法研究[J]. 计算机研究与发展,2008,45(41):11-16.

[17] 涂刚,阳富民,卢炎生. 基于动态优先级策略的最优软非周期任务调度算法[J]. 计算机研究与发展,2004,41(11):2026-2034.

网格依赖任务重调度资源选择算法研究

作者: [孟湘来](#), [MENG Xiang-lai](#)
作者单位: [天津广播电视大学, 天津, 300191](#)
刊名: [计算机技术与发展](#) 
英文刊名: [Computer Technology and Development](#)
年, 卷(期): 2015 (9)

引用本文格式: [孟湘来](#), [MENG Xiang-lai](#) [网格依赖任务重调度资源选择算法研究](#) [期刊论文] - [计算机技术与发展](#)
2015 (9)