第 26 卷第 1 期　　四川理工学院学报(自然科学版)　　Vol. 26　No. 1
2013 年 2 月　　Journal of Sichuan University of Science & Engineering(Natural Science Edition)　　Feb. 2013
文章编号:1673-1549(2013)01-0047-04　　DOI:10.3969/j. issn. 1673-1549. 2013. 01. 011

基于 QoS 约束的网格任务调度算法

王 浩, 李 飞

(成都信息工程学院网络工程学院, 成都 610225)

摘　要:针对网格环境下不同类型的用户任务执行时间差异较大的问题,在对现有网格调度算法研究之后,基于 Min – min 算法和 Sufferage 算法提出了基于任务 QoS 约束与任务损失度的最小最早完成时间算法 QDSM。算法克服了 Min – min 算法仅追求局部最优而忽视了全局的缺点。分析测试结果表明,算法实现了调度跨度与负载均衡、用户 QoS 约束的统一,在综合性能上有较大提高。

关键词:任务调度; 时间跨度; Min – min 算法; QoS 约束

中图分类号:TP393　　**文献标志码**:A

引　言

网格[1]环境下的任务调度问题是网格技术中的一个基本问题。由于网格环境中的资源所具有的异构性、动态性和分布性等特点,使得如何对任务进行调度以期满足各方面(系统用户、资源提供者、系统管理者等)的需求成为一个极具挑战性的问题。网格任务调度[2]的实质是将其环境中的 m 个需要调度的任务合理的分配到 n 个系统可用资源主机上去执行。由于现实环境中的网格系统规模一般都比较大,这样就决定了 m 和 n 都比较大,因此问题则转变成为一个 NP 难问题,即需要在有限时间内,在 2^n 个资源集合中寻找出最优任务 – 资源匹配方案,然而这又是不可能实现的,因此,一般都采用启发式任务调度算法获取近似最优配对。

目前国内外所研究的调度算法一般分为在线模式(on – line mode)和批处理(batch mode)模式两类。在线模式在任务到达的第一时间执行映射。批处理模式则需将任务收集到一定数量(系统设定的一个参数数值),等待映射事件发生后才开始映射所收集的任务。

1　相关工作

本文主要研究的是批处理模式下的启发式任务调度算法,并且已假定各任务之间相互独立,各任务在不同的资源上运行的预测执行时间可知。目前,经典的批处理模式下的调度算法有:Max – min[3]算法、Min – min[3-6]算法、GA[7-8]算法、Sufferage[9-10]算法和 SA[11](Simulated Annealing)等。Max – min 算法是基于 MCT(Minimum Completion Time,最小完成时间)的改进算法,算法选取最早完成时间最大的任务进行优先调度。GA 算法与 SA 算法,其复杂度一般认为都比较高。对于 QoS 约束[12-14]下的任务调度算法,国内外已有不少研究成果,其都考虑了多 QoS 约束的问题,但是对于大量的 QoS 约束(来自于系统用户、资源提供者、系统管理者等方面的)并未进行细分讨论;并且考虑的 QoS 约束一般都比较少,其对新约束条件的扩展性也比较差。

Min – min 算法也是基于 MCT 的改进算法,算法优先选择最早完成时间最小的任务进行调度,其以最快的速度减少任务调度队列中的待调度任务,以缩短任务的完成时间。但是 Min – min 算法同时也是一种贪心算法,仅追求任务完成时间最早的局部最优解,使得系统

收稿日期:2013-01-05
基金项目:四川省科技支撑项目(2011GZ0195)
作者简介:王 浩(1986-),男,四川平昌人,硕士生,主要从事基于云存储的计算机应用方面的研究,(E-mail)wang130_hao@126. com

负载不均衡,导致时间跨度值变大。尤其当任务的执行时间差异较大的时候,产生的负面效应就越突出。任务的损失度值反映出任务在资源主机上的执行完成时间差异,反映了环境的异构性。Sufferage 算法的时间跨度较小并且系统负载基本均衡,算法在减小任务调度跨度上的性能优于其它批处理算法,其表现出良好的综合性能;而对于具有 QoS 需求的任务的情况,基本欠缺考虑,并且任务本身也可能被多次进行分配。

在对多种异构环境下的启发式任务调度算法进行了研究、分析对比以后,结合 Min - min 算法和 Sufferage 算法的思想,提出了基于任务 QoS 约束和任务调度损失度的最小最早完成时间算法(QoS Dimensions and Sufferage Min - min,QDSM)。本文将任务 QoS 约束与任务损失度引入 Min - min 算法中,在综合权衡任务最早完成时间、任务 QoS 约束与调度损失的基础上进行任务调度,使得算法更加适应于异构环境。仿真测试表明,QDSM 算法具有较好的综合性能。

2 QDSM 算法

2.1 参数定义

为了更好的说明 QDSM 算法,本文使用以下一般性定义:

定义 1 集合 $T = \{t_1, t_2, \cdots, t_m\}$ 包含 m 个相互独立的任务。

定义 2 集合 $H = \{h_1, h_2, \cdots, h_n\}$ 包含 n 个可用资源。

定义 3 任务集合 T 所包含的 m 个任务在可用资源集合 H 所包含的 n 个主机上的预测执行时间(Expected Time to Compute, ETC)结果为 $m \times n$ 的矩阵:

$$ECT = \begin{bmatrix} e_{11} & \cdots & e_{1n} \\ \vdots & e_{ij} & \vdots \\ e_{m1} & \cdots & e_{mn} \end{bmatrix}$$

元素 e_{ij} 表示待执行任务 t_i 在可用主机资源 h_j 上的预测执行时间。

定义 4 m 个待执行任务在 n 个可用资源上面的预测最小完成时间 MCT 结果为 $m \times n$ 的矩阵:

$$MCT = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & c_{ij} & \vdots \\ c_{m1} & \cdots & c_{mn} \end{bmatrix}$$

元素 c_{ij} 表示待执行任务 t_i 在可用主机 h_j 上的预测最小完成时间。

定义 5 目前研究的用户 QoS 约束,考虑了 4 个维度:安全性、费用、成功率以及稳定性,映射为 $m \times 4$ 的用户 QoS 约束(User QoS Dimensions, UQD)矩阵:

$$UQD = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ \vdots & \vdots & \vdots & \vdots \\ q_{m1} & q_{m2} & q_{m3} & q_{m4} \end{bmatrix}$$

定义 6 集合 V 为待调度分配任务集合,其中,所有元素均属于 T 并且尚未被分配。

定义 7 第 i 个任务 t_i 的损失度(sufferage)为任务的最优最早完成时间与次优最早完成时间之差。即:

$$sufferage(t_i) = c_{iy} - c_{iz}$$

m 个任务的 *suffer* 值构成了维度为 m 的向量 $S = \{s_1, s_2, \cdots, s_m\}$。

定义 8 $m \times k$ 维矩阵 MT,用于储存每个任务在各个资源主机上的前 k 个最小执行时间,其中,元素 mt_{ij} 表示任务 t_i 的最小完成时间,参数 k 为可调节参数,取值范围为 $[1, n]$。

2.2 算法描述

根据前述参数定义,首先对 UQD 矩阵进行归一化处理,计算权重,选取权重最大者存入向量 Q;分别选取待调度任务中的最小执行时间任务与 *suffer* 值最大的任务,分别进行标记;计算权衡系数 α,根据权衡系数,选取相应的任务进行调度,同时更新 MCT 矩阵。

对于用户 QoS 约束矩阵 UQD 和预测执行时间矩阵 ECT 均已知,并已初始化;MCT 矩阵和集合 V 均为空。算法的详细步骤如下:

(1)输入矩阵 ECT 和 UQD。

(2)对矩阵 MCT 和集合 V 进行初始化,其中,$MCT = ECT$, $V = T$, 对矩阵 UQD 进行归一化处理。

(3)在矩阵 ECT 中,查找每个任务的最小执行时间,并选取前 k 个元素存入矩阵 MT。

(4)当 V 非空时,循环执行步骤(5)~步骤(9)。

(5)根据 MCT 矩阵,计算任务集合 V 的 *suffer* 值,并从中找出任务的最大 *suffer* 值,记为 s_a, 其对应的任务 $t_a \in V$。

(6)在矩阵 MT 中,查找对应于任务集合 V 的最大执行时间任务,记为 mt_b, 其对应的任务 $t_b \in V$, *suffer* 值记为 s_b。

(7)对归一化后的 UQD 矩阵,计算任务的各维 QoS 约束在待调度任务中所占权重,选取所占权重最大者存入向量 $Q = \{mq_1, mq_2, \cdots, mq_m\}$。

(8)求解权衡系数 α,

$$\alpha = \frac{mq_b}{mq_a} \times \frac{mt_b}{\text{集合 } V \text{ 中任务最小执行时间均值}}$$

(9)如果($s_a \geq (\alpha \times s_b)$)

将任务 t_a 分配到资源主机 r_a 上执行,并依据 t_a 的执行时间更新 *MCT* 矩阵,从集合 *V* 中删除任务 t_a, 否则,将任务 t_b 分配到资源主机 r_b 上执行,并依据 t_b 的执行时间更新 *MCT* 矩阵,从集合 *V* 中删除任务 t_b。

3　性能分析

在多任务、多资源的网格模拟环境 GridSim[15] 中使用随机产生的 *ECT* 和 *UQD* 矩阵作为仿真输入,分别针对 Min－min 算法、Sufferage 算法、SMM 算法和 QDSM 算法进行任务调度仿真,采集模拟数据,分析每个算法的性能,针对性的验证了 QDSM 算法在最优跨度、资源平均利用率等方面的性能。其中,资源平均利用率按下式计算。

$$\text{资源平均利用率} = \sum_{i=1}^{m} \frac{\text{资源 } r_i \text{ 上的任务的任务运}}{\text{资源 } r_i \text{ 实际占用时间}} / m$$

实验使用统计数据均值对算法性能进行分析,分成 2 组进行实验仿真验证。

3.1　时间跨度

图 1 为资源数为 10 时,在不同任务数下进行的仿真实验得到的 makespan 均值,资源数与任务数分别按 10∶100,10∶200 和 10∶300 进行匹配。

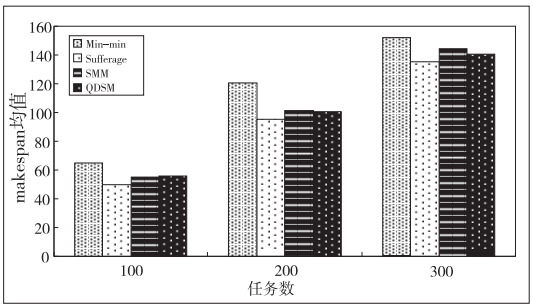

图 1　makespan 均值

分析图 1 数据可知,Sufferage 算法、SMM 算法和 QDSM 算法的 makespan 均值均少于 Min－min 算法。随着任务数量的增加,QDSM 算法的性能略有下降,但与 Min－min 算法和 SMM 算法相比仍有较好的性能。

3.2　资源平均利用率

图 2 所示为,资源数为 10 时,在不同任务数下进行的仿真实验得到的资源平均利用率。

由图 2 分析可知,QDSM 算法使得系统的资源平均利用率比 SMM 算法略有提升,较 Min－min 算法和 Sufferage 算法都表现出较好的性能。

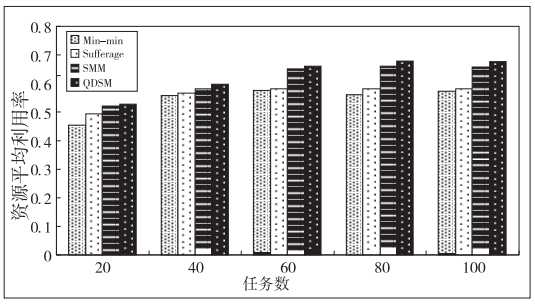

图 2　资源平均利用率

4　结束语

本文在研究了多种启发式网格任务调度算法以后,提出了适合于异构环境的独立任务调度算法:基于任务 QoS 约束和任务调度损失度的最小最早完成时间算法(QoS Dimensions and Sufferage Min－min,QDSM)。所提算法克服了 Min－min 算法的局限性,更适应于异构环境下的任务调度。然而,对于网格中资源的异常处理,需要分析异常产生的原因,根据原因有针对性的提出解决方案;对于任务约束的动态可扩展性,则需要对 QoS 约束、资源系统等各方面进行深入的分析与研究。

参 考 文 献:

[1] 都志辉,陈 渝,刘 鹏.网格计算[M].北京:清华大学出版社,2002.

[2] 王相林,张善卿,王景丽.网格计算核心技术[M].北京:清华大学出版社,2006.

[3] Chauhan Sameer Singh,Joshi R C.A weighted mean time Min-Min Max-Min selective scheduling strategy for Independent Tasks on Grid[C].//Deepak Garg.Proceeding of IACC 2010,Patiala,India,February,19-20,2010:4-9.

[4] Braun T D,Siegel H J,Beck N.A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems[J]. Journal of Parallel and Distributed Computing,2001,61(6):810-837.

[5] 周 洋,蒋昌俊,方 钰.异构环境下的独立任务调度算法的研究[J].计算机科学,2008,35(8):90-92.

[6] 莫 赞,谢 娜,贾功祥,等.基于多 QoS 需求驱动的网格资源调度研究[J].计算机应用研究,2012,29(10):3904-3907.

[7] Braun T D,Siegel H J,Maciejewski A.Static mapping heuristics for tasks with dependencies,priorities,dead-

lines, and multiple versions in heterogeneous environments[C]. Ibarra O H,Olariu S,Nakano K,et al. Proceedings of the 16th Int'l Parallel and Distributed Processing Symposium,F.L.,Florida,USA,April,15-19,2002:78-85.

[8] 朱 海,王宇平.多目标约束的网格任务安全调度模型及算法研究[J].电子与信息学报,2010,32(4):988-992.

[9] He Xiaoshan, Sun Xianhe, von Laszewskig G. QoS Guided Min-min Heuristic for Grid Task Scheduling [J].The Journal of Computer Science and Technology, 2003,18(4):442-451.

[10] 李 炯,卢显良,董 仕.基于 GridSim 模拟器的网格资源调度算法研究[J].计算机科学,2008,35(8):95-97.

[11] 薛胜军,徐钧磊,刑国稳.一种用于网格任务调度的退火进化算法[J].计算机应用研究,2011,28(11):4049-4052.

[12] Dogan A, Ozguner F. On QoS-based scheduling of a meta-task with multiple QoS demands in heterogeneous computing[C]. Ibarra O H, Olariu S, Nakano K, et al. Proceedings of the 16th Int'l Parallel and Distributed Processing Symposium,F.L.,Florida,USA,April1,5-19,2002:50-55.

[13] Chauhan Sameer Singh,Joshi R C.QoS guided heuristic Algorithms for grid task scheduling[J]. International Journal of Computer Applications,2010,2(9):24-31.

[14] Castillo C,Rouskas G N,Harfoush K.Online algorithms for advance resource reservations[J].Journal of Parallel and Distributed Computing,2011,71(7):963-973.

[15] Buyya R, MURSHED M. GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing[J].Concurrency and Computation:Practice and Experience,2002, 14(13):1175-1220.

Grid Task Scheduling Algorithm Based on QoS Dimensions

WANG Hao, LI Fei

(School of Network Engineering, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: Aiming at the problem that the grid environment of different types of user task execution time different, and research on the existing grid heuristic scheduling algorithm, the proposed a new algorithm based on task QoS dimensions and task sufferage, called QoS Dimensions and Sufferage Min-min (QDSM), which is based on both Min-min algorithm and Sufferage algorithm. The new algorithm overcomes the Min-min algorithm only to pursue local optimal of the shortcomings and ignore the global optimal. The statistic from the experiments reveal that QDSM algorithm has merits of the better executing efficiency, less time to complete the task, high load balance degree and user QoS dimensions by comparing with Min-min algorithm and Sufferage algorithm.

Key words: Tasks scheduling; Makespan; Min-min algorithm; QoS dimensions