

文章编号:1001-9081(2007)03-0541-02

一种基于效用最优的计算网格资源调度算法

余建军,郑月斋,杨明霞

(浙江工业大学浙西分校 信息与电子工程系,浙江 衢州 324000)

(yjj691121@sohu.com; yjj671121@126.com)

摘要:针对具有时间和费用约束的网格资源调度问题,通过分析任务的执行时间和执行费用与用户效用函数的关系,建立了效用函数模型,在此基础上提出了一种针对 task-farming 应用程序模式的基于效用函数最优的启发式调度算法——效用最优调度算法。实验表明,与费用-时间均衡调度算法相比,本算法能给用户带来更高的效用。

关键词:计算网格;效用函数;资源调度;启发式算法;task-farming

中图分类号: TP393.07 **文献标识码:** A

Resource scheduling algorithm for computing grid based on utility optimization

YU Jian-jun, ZHENG Yue-zhai, YANG Ming-xia

(Department of Information & Electronic Engineering, West Branch of Zhejiang University of Technology, Quzhou Zhejiang 324000, China)

Abstract: Concerning the time and budget restriction in resource scheduling, a utility function model was proposed by analyzing the relationship between the time and budget of task and the user utility function. Then, based on the utility function model, a utility optimization based heuristic resource scheduling algorithm was put forward for task farming application model. Simulation results show that our proposed algorithm provides user with more utilities than cost-time optimization algorithm.

Key words: computational grid; utility function; resource scheduling; heuristic algorithm; task-farming

0 引言

在网格^[1]环境中,由于资源的广域性、自治性和动态性,传统的集中资源分配方式既难以实现也不能满足不同应用对资源需求的差异性。由于网格资源分配和社会资源配置具有相似性,借助市场机制解决网格资源分配具有可行性和优越性:首先,网格资源的提供者和使用着往往分属不同的组织,这必然涉及到费用问题,借助经济的力量可以自动均衡对资源的需求;其次,每个参与者可以根据自身偏好自主决策,实现分散的以用户为中心的网格资源分配模式;最后,市场中每个参与者追求个体利益最大化的行为会使得整个网格资源匹配趋于最优,这是传统的资源分配方式所追求但却难以实现的。将市场机制应用于网格资源分配的研究中,有影响力的一个原型系统是澳大利亚 Monash 大学 Rajkumar Buyya 开发的 Nimrod/G^[2]。

Rajkumar Buyya 等在文献[3]和文献[4]中论述了所开发的支持任务最终完成期限和费用预算约束(Deadline and Budget Constraint, DBC)的调度算法,具体包括费用最优调度、时间最优调度、费用-时间均衡调度等。

本文提出了一种新的基于效用函数^[5-7]的启发式调度算法——效用最优调度算法,该算法在调度时使用效用函数法将任务完成时间和运行费用综合考虑,在进行任务分配时,将任务分配给效用最大的资源。扩充了 GridSim^[2]工具包中的网格资源调度器——基于计算经济的网格资源中介(Economic Grid Resource Broker, EGRB),以支持效用最优调度算法。

1 资源调度问题的描述

资源调度的主要任务是根据任务信息采用恰当的策略把不同任务分配到相应的资源上。

同 EGRB 一样,这里效用最优调度算法主要针对 task-framing 的应用程序模式^[3,8],即资源调度仅考虑独立任务集的情况。当用户向资源调度器提交独立任务集时,同时给出整个任务集的 deadline 和 budget,表示整个任务集的运行时间和费用不能超过 deadline 和 budget。

调度器的调度目标是在满足 deadline 和 budget 约束的前提下,优化调度性能,本文中使用的性能指标同 EGRB,即:

- 1) 任务完成率。满足 deadline 和 budget 约束,成功完成的任务占所提交任务的比例。
- 2) 任务执行时间。用户向调度器提交任务的时间与调度器完成向用户返回所完成的所有任务的时间差。
- 3) 任务执行费用。每个完成的的任务的费用之和。每个完成任务的费用是执行该任务的资源的价格和任务的执行时间之积。

即我们采用多指标来综合评价调度算法的性能,是一个多目标综合优化问题。

2 效用最优调度算法

根据多目标决策理论^[6],具有 k 个目标的多目标问题能够线性加权转化为单目标的首要条件是具有相同的量纲,否则可用效用函数法来完成。

收稿日期:2006-09-18

作者简介:余建军(1969-),男,浙江江山人,讲师,硕士研究生,主要研究方向:算法设计和分析、智能优化算法、网格作业调度; 郑月斋(1969-),女,浙江永康人,高级工程师,硕士研究生,主要研究方向:软件工程、数据库应用技术、现代网络技术; 杨明霞(1979-),女,浙江衢州人,助教,硕士研究生,主要研究方向:软件工程、数据库应用技术。

本文中基于如下假设:用户在给定 deadline 和 budget 并提交任务集后,任务完成率是用户满意程度的关键;然后根据 deadline 和 budget 相对大小来考虑对任务执行时间和任务执行费用的满意程度,即如 budget 与 deadline 比较,相对较大,则用户偏好于减少任务的执行时间,反之偏好于尽量减少费用。

2.1 效用函数

多目标的效用函数设计^[5],首先要要求出每个目标的效用函数,然后按照一定的合成模型加权合成。

2.1.1 任务执行时间和任务执行费用的效用函数

1) 任务执行时间的效用函数

我们定义任务 T_i 在资源 R_i 上执行的时间效用函数为:

$$U_t(T_i, R_i) = \begin{cases} 0, & T_{\text{completion_time}} > T_{\text{remain_time}} \\ T_{\text{completion_time}}/T_{\text{remain_time}}, & T_{\text{completion_time}} < T_{\text{remain_time}} \end{cases} \quad (1)$$

其中 $T_{\text{completion_time}}$ 是任务 T_i 在资源 R_i 上执行时间的预算, $T_{\text{remain_time}}$ 是任务集执行到当前还剩下的预算时间。也就是在满足 deadline 的前提下,尽量在速度慢的资源上执行,这样有益于提高任务集的完成率,且速度慢的资源一般价格较便宜。

2) 任务执行费用的效用函数

显然,任务执行费用越小用户的满意度越高。所以我们定义任务 T_i 在资源 R_i 上执行的费用效用函数为:

$$U_c(T_i, R_i) = \begin{cases} 0, & C_{\text{completion_time}} > C_{\text{remain_time}} \\ 1 + (C_{\text{min_cost}} - C_{\text{completion_time}})/C_{\text{remain_time}}, & C_{\text{completion_time}} < C_{\text{remain_time}} \end{cases} \quad (2)$$

$$C_{\text{remain_cost}} = \text{budget} - C_{\text{expenses}} - C_{\text{OnHold}}$$

其中 $C_{\text{completion_cost}}$ 是任务 T_i 在资源 R_i 上执行费用的预算; $C_{\text{remain_cost}}$ 是任务集执行到当前还剩下的费用; $C_{\text{min_cost}}$ 是任务 T_i 在最便宜的资源上执行的花费; C_{expenses} 是当前已经花费的费用, C_{OnHold} 是当前在执行但尚未完成所有任务的费用预算之和。

2.1.2 合成模型

当任务 T_i 在资源 R_i 上的执行时间或费用的效用函数为 0 时,那么任务 T_i 就不能分配在资源 R_i 上执行,否则不能在 deadline 和 budget 之内完成该任务,既任务 T_i 在资源 R_i 上执行的效用为 0。根据乘法规则^[6],我们定义任务 T_i 在资源 R_i 上执行的效用函数为:

$$U(T_i, R_i) = U_t(T_i, R_i)^{w_1} \times U_c(T_i, R_i)^{w_2} \quad (3)$$

其中 $w_1 + w_2 = 1, w_1 \geq 0, w_2 \geq 0$ 。

2.1.3 权重的确定

w_1 和 w_2 分别代表时间效用和费用效用的相对重要性。体现了如 budget 与 deadline 比较,相对较大,则用户偏好于减少任务执行时间,反之偏好于尽量减少费用。具体定义如下:

$$\begin{aligned} w_t &= \text{deadline}/T_{\text{expect}} \\ w_c &= \text{budget}/C_{\text{expect}} \\ w_1 &= w_t / (w_t + w_c) \\ w_2 &= w_c / (w_t + w_c) \end{aligned} \quad (4)$$

其中 T_{expect} 是用户提交任务集的运行时间估计值(取所有任务的总长度除以计算资源的平均计算速度的值与最长的任务的长度除以最快的资源的计算速度的值两者之间的大值), C_{expect} 是用户提交任务集的运行费用估计值(取任务的总长度除以计算资源的平均价格)。

2.2 启发式调度算法

假设: N 表示用户提交任务集中任务的数目, T_i 表示第 i

个任务, M 表示资源的数目, R_j 表示第 j 个资源。

```

scheduleWithDBC_UtilityOptimisation() {
  for (i = 0; i < N; i++)
  {
    utility = 0.0;
    for (j = 0; j < M; j++)
    {
      temp = U( T_i, R_j );
      if( temp > utility ) { utility = temp; br = j; }
    }
    if ( utility > 0.0 )          将任务 T_i 分配到资源 T_br ;
  }
}

```

3 仿真试验

3.1 仿真环境

扩充源代码公开的 GridSim 工具包中的网格资源调度器 EGRB,主要是给类 Broker 增加了四个成员函数: Utility_Time (BrokerResource br, Gridlet g), Utility_Cost (BrokerResource br, Gridlet g), Utility (BrokerResource br, Gridlet g) 和 scheduleWithDBC_UtilityOptimisation(), 分别完成任务执行时间效用函数的计算、任务执行费用效用函数的计算、综合效用函数的计算和效用最优调度算法的实现。

3.2 实验数据

我们采用与文献[4]中相同的资源模型,即网格中包含具有不同特性的 11 个资源。

实验时我们构造了由 200 个任务组成的任务集,每个任务要求处理的长度为 10000 加上一个 0~1000 的随机数,每个任务的输入和输出长度为任务长度的 10%。由于在本算法中没有考虑网络传输时间,所以把用户和资源的带宽设置为 11000 和 2000000,即任务在资源和用户之间的网络传输时间可以忽略不计。

我们针对上述任务集,完成了不同 deadline 和 budget 下的实验。具体是 deadline 按每次递增 500 从 100 逐渐递增到 3100; budget 是按每次递增 2000 从 5000 逐渐递增到 19000。

3.3 仿真结果和分析

Rajkumar Buyya 等所提出的费用最优调度、时间最优调度、费用-时间均衡调度等调度算法中,费用-时间均衡调度具有最好的性能,所以我们选择把效用最优调度算法同费用-时间均衡调度算法进行比较研究,实验结果见表 1。

表 1 中的对比数据值是指效用最优调度算法同费用-时间均衡调度算法相比,各项性能指标值提高的百分比。在实验中我们发现,在费用-时间均衡调度算法中,在不同 budget 和 deadline 下执行费用主要在 5500 左右,也就是费用预算相对充足;而任务执行时间主要分布在 100 与 1400 之间,也就是说 deadline 使用的相对充分。通过实验,我们发现效用最优调度算法与费用-时间均衡调度算法相比较,任务完成率平均提高 8.1%,任务执行时间平均增加 3.4%,任务执行费用平均增加 10.1%,也就是在费用预算相对充足的条件下(费用从 5000 到 19000)效用最优调度算法能通过多增加费用、少增加执行时间,达到增加任务完成率的效果。

4 结语

仿真实验表明本文所提出的效用最优调度算法具有较好的调度性能,但算法中没有考虑网络传输的时间和延迟、网络传输费用等因素。

(下转第 549 页)

立与 ACS 连接,代理 ACS 调用 UPnP 设备服务动作,并将网关状态变化信息及时通知给 ACS,实现 ACS 对家庭网关的配置管理。

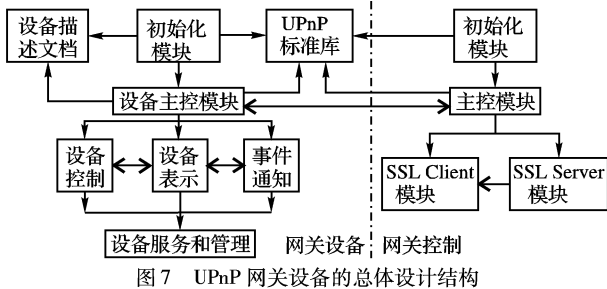


图 7 UPnP 网关设备的总体设计结构

7 结语

在对 UPnP 家庭网关配置管理方法的设计中,本文始终围绕家庭网关立足家庭、面向运营的特性,这种家庭网关不仅满足家庭网络设备的互操作性,同时具有可运营、可管理的应用模式。在具体技术实现上,将电信网络的 TR-069 终端管理技术与家庭网络的 UPnP 协议框架相结合。这在家庭网关乃至家庭终端的实现方法上是一种创新的尝试,为数字家庭产品研发者建立真正具有互操作性的家庭网络提供借鉴。

在家庭网关上实现配置管理功能,旨在为家庭网络引入并开展宽带服务,使大规模服务的自动部署成为可能。本文提供多种配置方法,可分别从家庭网关的 LAN 侧接口和 WAN 侧接口进行配置管理;也可通过家庭网关的 NAT 透传与地址解析,将远程管理扩展到家庭网络内部设备上,从而实现 ACS 与家庭终端的直接关联与配置管理,在技术上为电信

运营商将宽带服务延伸到家庭网络内部提供了理论依据,更好地体现了家庭网关可运营、可管理的特性。

致谢:本文的设计工作是在深圳风华比特通讯技术有限公司的家庭网关开发实验室进行的,工作中得到了企业领导的大力支持。在此向他们表达衷心的感谢!

参考文献:

- [1] UPnP Device Architecture, Version 1.0[S]. UPnP Forum, 2003.
- [2] SHAH A, JOHNSON K, SIBBITT M. DSL Service Flow-Thru Fulfillment Management Overview, TR-038[R]. DSL Forum, 2001.
- [3] BEMSTEIN J, SPETS T. CPE WAN Management Protocol, TR-069[R]. DSL Home Technical Working Group, 2004.
- [4] STARK B. LAN-Side DSL CPE Configuration Specification, TR-064[R]. DSL Home Technical Working Group, 2004.
- [5] JERONIMO M, WEAST J. UPnP Design by Example, A software Developer's Guide to Universal Plug and Play[M]. INTEL PRESS,2003.
- [6] 杨士元. 数字家庭网络技术及应用[M]. 北京: 电子工业出版社,2002.
- [7] Home Networked Device Interoperability Guidelines V1.0[S]. Digital Living Network Alliance, 2004.
- [8] IYER P, WARRIER U. UPnP Home Gateway Device, For UPnP Version 1.0[S]. UPnP Forum, 2001.
- [9] OOGHE S, BATHRICK G, STANKUS T. Auto - Configuration for the Connection Between the DSL Broadband Network Termination (B-NT) and the Network using ATM, TR-062[R]. DSL Auto-Configuration Working Group, 2003.
- [10] Auto-Configuration Architecture Framework, TR-046[R]. DSL Forum, 2002.
- [11] Auto-Configuration for Basic Internet (IP-based) Services, TR-044[R]. DSL Forum, 2001.

(上接第 542 页)

表 1 效用最优调度算法与费用-时间均衡调度算法性能比较

dead-line	指标	budget							
		5000	7000	9000	11000	13000	15000	17000	19000
100	任务完成率	7.3	2	-5.1	-3.4	-6.3	-2.0	0	1.0
	执行费用	8.1	0	0	0	0	0	0	0
	执行时间	7.8	6.5	-11.2	-11.6	-9.5	-5.8	7.7	0.8
600	任务完成率	20.3	22.5	25.8	-6.5	37.1	-7.8	7.4	33.1
	执行费用	25.7	35.1	37.2	-4.6	18.1	-3.4	3.5	47.3
	执行时间	9.5	29.6	19.8	-7.4	32.8	-10.7	11.4	50.1
1100	任务完成率	15.6	29.3	40.3	26.7	7.4	3.5	-1.1	-2.7
	执行费用	21.2	20.5	51.5	63.1	11.9	2.9	-0.8	-4.5
	执行时间	-4.3	22.2	8.1	-9.5	4.9	7.6	-2.4	1.7
1600	任务完成率	4.6	66.7	28.6	47.7	68.1	10	11.3	6.5
	执行费用	-0.1	70.7	35.9	74.4	-6.8	17.5	9.5	5.5
	执行时间	2	4.7	13.6	9.2	8.8	35.5	20.5	12.7
2100	任务完成率	-5.7	-4.5	-9.9	-3.7	-4.8	-1.7	0	1.1
	执行费用	0	3.5	1.1	1.4	-22.7	-1.3	0.7	0.8
	执行时间	-13.7	1.5	-12.5	-5.7	4.8	-2.9	-0.7	2.4
2600	任务完成率	-2.7	-3.5	-3.4	2.7	1.7	2.2	-1.4	-1.1
	执行费用	0	1.3	3.3	3.7	2.5	2.1	-1.1	-0.4
	执行时间	-4.4	-4.3	-13.7	1.3	0.5	2.3	-0.5	-1.1
3100	任务完成率	1.1	0	0	0	0	0	0	0
	执行费用	-0.1	-2.3	5.5	5.5	5.5	5.5	5.5	5.5
	执行时间	1.9	1.7	-5.8	-5.8	-5.8	-5.8	-5.8	-5.8

参考文献:

- [1] FOSTER I, KESSELMAN C, TUECKE S. The anatomy of the grid: Nabling scalable virtual organizations[J]. International Journal of High Performance Computing Applications, 2001, 15(3):200 - 222.
- [2] BUYYA R. Economic - based distributed resource management and scheduling for grid computing [D]. Melbourne: Monash University, 2002.
- [3] BUYYA R, GIDDY J, ABRAMSON D. An Evaluation of Economy-based Resource Trading and Scheduling on Computational Power Grids for Parameter Sweep placements[A]. Proceeding of the 2nd International Workshop on Active Middleware Services(AMS 2000) [C]. Pittsburgh, USA, Kluwer Academic Press, 2000.221 - 230.
- [4] BUYYA R, MURSHED M, ABRAMSON D. A Deadline and Budget Constrained Cost-Time Optimization Algorithm for Scheduling Task Farming Applications on Global Grids[A]. The 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02) [C]. Las Vegas, Nevada, USA, 2002.
- [5] 苏为华. 多指标综合评价理论与方法问题研究 [D]. 福建: 厦门大学, 2003.
- [6] 宣家骥. 多目标决策[M]. 长沙: 湖南科学技术出版社, 1988.
- [7] 张尧庭, 陈慧玉. 效用函数及优化[M]. 北京: 科学出版社, 2000.
- [8] (美)MATTSON TG, SANDERS BA, MASSINGILL BL. 并行编程模式[M]. 敖富江, 译. 北京: 清华大学出版社, 2005.