

一种网格工作流动态调度算法^①

Dynamic Scheduling for Grid Workflow Algorithm

桑莉莉 (桂林电子科技大学 计算机与控制学院 广西 桂林 541004)

摘 要: 由于网格系统异构和资源动态变化, 网格工作流多个任务对资源的不同需求, 以及任务之间的时序、因果和数据依赖关系, 使得网格工作流调度问题非常复杂, 低性能的资源 and 任务调度策略, 将会增加任务的执行时间并降低整个网格系统的吞吐量。本文针对网格工作流的特点提出了一种动态调度算法, 该算法追求优化执行时间和系统负载均衡的双重目的, 最后通过实验验证了该算法的可行性和优越性。

关键词: 网格 工作流 动态调度

1 引言

由于网格应用越来越复杂, 包含时间、空间和资源方面的约束, 因此一些网格项目, 例如 **Phy-GridN** 和 **GridAnt** 采用工作流和具有工作流特征的服务来管理网格中的应用, 提高了网格效率和监控管理。随着网格研究和应用的不断深入, 网格工作流的调度已经成为网格领域的研究热点。

网格是一个动态变化的环境, 当前网格调度中的大部分调度策略和算法都是针对元任务的, 即调度任务相互之间没有关联, 这显然不满足网格工作流中任务之间的时序或因果的约束条件。为了满足大规模应用中复杂的任务调度关系, 必须研究分布的网格体系结构中的工作流调度处理机制, 动态地选择合适的调度策略和规则, 实现工作流的动态优化和自动处理。综合以上分析, 本文针对网格应用和工作流的特点, 提出了一种网格工作流的动态调度算法。

2 网格工作流动态调度算法

2.1 网格工作流动态调度机制

网格任务调度问题的最一般的目标函数是 **Makespan**, 即调度系统有效的分配网格资源, 实现在整个系统内网格应用任务的完成时间最小。但是对于网格工作流的调度来说, 由于任务之间存在一些时序和因果上的依赖, 因此在调度的时候需要考虑的约

束条件比一般的网格任务调度要复杂。

网格工作流由若干关联任务构成, 每个任务都是这个流程中的一个环节, 并分散在不同的网格结点上运行, 每个任务的执行不仅受到其他执行结果的影响, 而且这个任务的执行结果也将影响其他任务的执行。任务的这种依赖关系使得一些任务必须在其他任务完成后才能开始执行。因此, 网格工作流的调度通常对任务间的依赖关系进行分析, 遵循只有任务的全部先导任务都完成后才能被调度的原则, 从所有尚未调度的任务集中选取满足约束条件的任务形成调度队列, 选取合适的调度算法进行资源的定位和任务的分派, 直到全部任务调度完毕为止。

传统工作流往往在建模过程中实现任务和所需资源的绑定, 在模型解释执行的过程中, 若最初映射的资源不再满足用户需求或变得不可用, 将导致意外的产生。网格环境动态变化, 这种方法显然无法适应松散耦合的网格工作流应用。为了解决可调度任务动态调整和资源动态变化的问题, 采用动态调度机制, 即在建模时不绑定具体资源, 而是绑定资源描述, 调度时根据运行的实际情况, 选择合适的资源执行任务。

同时, 网格工作流调度还要考虑资源负载的问题。如果任务提交者不管自己的任务具体情况如何, 都一味地追求使用世界上最快、最好的资源来计算自己的

^① 基金项目: 广西研究生创新计划资助项目(2008105950812M428)
收稿时间: 2008-10-30

问题,将导致某些资源负载过重,不堪重负,而某些资源却空闲的情况。

因此,网格工作流的调度应能有效地应对动态变化的网格计算环境,网格工作流任务调度在考虑任务间依赖关系的同时,必须结合网格结点的实时性能进行综合评价,同时力求达到系统负载均衡,以获得高的任务吞吐率和资源利用率,追求任务高效执行和网格充分利用的双赢目的。

下面先给出本文在调度过程中用到的一些参数。

2.2 相关参数

• 关键路径

在网格工作流对应的过程模型中,具有最大时间开销的工作流执行路径称为该工作流的关键路径。

• 关键因子

设关键路径预期的执行时间为 T_{key_road} ,待调度任务所在路径预期的执行时间为 T_i ,则关键因子 $key(T_i)$ 为

$$key(T_i) = \frac{Time_{T_i_road}}{Time_{key_road}} \quad (1)$$

• 优先因子

事先给定的任务的优先程度,例如根据服务质量,可以给定任务的优先因子。设任务的优先因子为 $Prio(T_i)$ 。

• 紧迫因子

任务所在应用的紧迫程度可以用紧迫因子 $Pressure(T_i)$ 来衡量:

$$Pressure(T_i) = \begin{cases} 1 - k \times \frac{T_{ideadline} - T_{inow}}{Time_{key_road}} & (\text{任务执行时间充足}) \\ 1 + k \times \frac{T_{inow} - T_{ideadline}}{Time_{key_road}} & (\text{任务已经延期}) \end{cases} \quad (2)$$

其中, k 为一个常数,用于调整参数范围, $T_{ideadline}$ 为用户对任务 T_i 给出的时间需求, T_{inow} 为当前时间。

• 时间开销

将任务从资源调度到资源的时间开销可表示为:

$$Cost(T_i, R_i, R_j) = ND_{R_i, R_j} + \frac{DT(R_i, R_j)}{BW(R_i, R_j)} + rate \times (p_j \times Wait_j + \frac{Cost_{T_i}}{Power_j}) \quad (3)$$

其中, ND_{R_i, R_j} 表示网络延迟, $DT(R_i, R_j)$ 表示任务 T_i 在资源 R_i 和 R_j 间的数据传输量, $BW(R_i, R_j)$ 表示数据

传输时的带宽, p_j 为资源的等待概率参数, $Wait_j$ 为任务 T_i 在资源 R_j 上开始执行的等待时间, $Cost_{T_i}$ 表示任务 T_i 的计算量, $Power_j$ 表示资源 R_j 的计算性能参数。

2.3 调度算法

综合 2.1 节网格工作流动态调度机制的研究,本文提出网格工作流的动态调度算法追求优化执行时间和系统负载均衡的双重目的,根据网格工作流的自身特点,从待处理的网格工作流任务集中选取可调度的任务,即先导任务全部执行完毕的任务,依据任务的关键因子、优先因子和紧迫因子将待处理任务集分为高优先就绪队列和低优先就绪队列,对高优先就绪队列中的任务采取尽快处理原则,选择执行时间最短的资源进行任务处理,而对低优先就绪队列中的任务,采用负载均衡的资源分配策略,分配低负载的网格资源进行任务处理,达到提高系统吞吐量和资源利用率的目的。具体的调度算法如下:

Step1. 生成网格工作流任务的动态就绪队列 **high_ready** 和 **low_ready**

Initialize 网格工作流实例,得到对应的过程模型,假设任务集为 $T=\{T_1, T_2, \dots, T_n\}$;

Initialize **high_ready**= ϕ , **low_ready**= ϕ ;

Repeat

Find T_i **from** T ; // 找到可调度的任务,即先导任务全部执行完毕

Count $key(T_i)$; // 计算关键因子

Count $prio(T_i)$; // 计算优先因子

Count $Pressure(T_i)$; // 计算紧迫因子

Insert(T_i , **high_ready**/**low_ready**); // 根据关键因子、优先因子和紧迫因子将任务插入相应的就绪队列

Until($T=\phi$)

Step2. 从资源集合 $R=\{R_1, R_2, \dots, R_n\}$ 中为任务 T_i 选择合适的资源,生成任务 T_i 的资源队列 **RList_i**

If R_i **satisfied** T_i

Resource requirements;

Organization role;

Qos requirements;

$R_i \rightarrow RList_i$

Step3. 任务动态调度

```

While(high_ready ≠ ∅)
{ //对高优先就绪队列实施调度
  select head of high_ready; //选取高优先
  就绪队列的首元素任务
   $\forall R_j \in RList_{head} : \text{Count Cost}(\text{head}, R_{pre}, R_j);$ 
  //预测该任务在每个可用资源上的执行开销
  schedule(head,  $R_{min}$ ); //将任务调度到执行
  开销最小的资源上运行
  remove head from high_ready;
}

While(low_ready ≠ ∅)
{ //对低优先就绪队列实施调度
  select head of low_ready; //选取低优先
  就绪队列的首元素任务
   $\forall R_j \in RList_{head} : \text{Count Load}_j = \frac{\sum Power}{TotalPower_j};$ 
  //计算每个可用资源上的负载
  select head of low_ready; //选取低优先
  就绪队列的首元素任务
  schedule(head); //将任务调度到负载最
  轻的资源上运行
  remove head from low_ready;
}

```

其中, $\sum Power$ 表示资源 R_j 上所有任务的计算开销, $TotalPower_j$ 为资源的计算力。

3 实验结果

本次实验包括两部分: ①该算法能否尽量缩短整个应用的完成时间; ②是否达到系统负载均衡。下面分别进行介绍。

① 仿真一个网格工作流应用的例子, 分别在 2、4、6、8 个节点上进行实验, 用本文提出的算法与不设置优先因子、关键因子、紧迫因子的随机调度方法进行比较, 结果如图 1 所示。从图可以看出, 本文提出的算法, 大大提高了应用的执行时间。

② 实验分别对包含 10、20、30、40、50、60、70、80、90 和 100 个任务的网格工作流在 3 个网格节点上进行测试的负载情况如图 2 所示。从图可以看出, 3 个网格节点的资源负载比较均衡。

由此得出结论: 本文提出的网格工作流动态调度算法, 达到了优化执行时间, 均衡资源负载的目的。

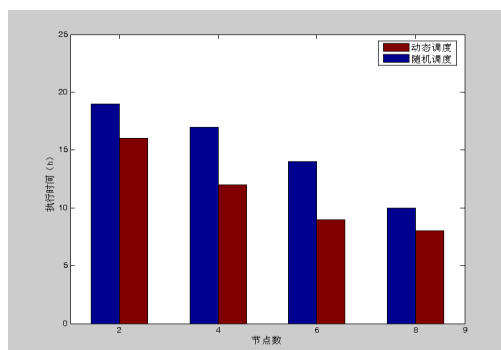


图 1 应用执行时间

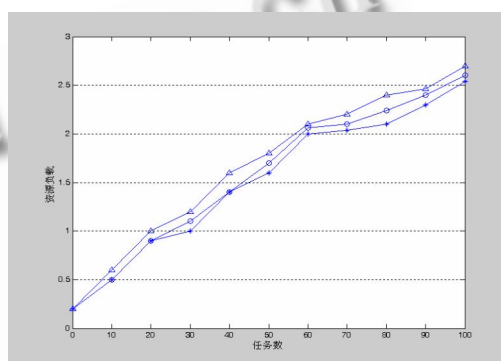


图 2 3 个节点的资源负载

4 结论

本文针对网格工作流的特点提出了一种动态调度算法, 即在建模时不绑定具体资源, 而是绑定资源描述, 调度时根据运行的实际情况, 选择合适的资源执行任务。该算法追求优化执行时间和系统负载均衡的双重目的, 并通过实验证明了提出的调度算法的可行性和优越性。

参考文献

- 1 Tannenbaum T, Wright D, Miller K, Livny M. Condor -A Distributed Job Scheduler, Beowulf Cluster Computing with Linux. The MIT Press, MA, USA, 2002.
- 2 Fahringer T, et al. ASKALON: a tool set for cluster and Grid computing. Concurrency and Computation: Practice and Experience, Wiley InterScience, 2005, 17: 143 - 169.
- 3 Cooper K, et al. New Grid Scheduling and Rescheduling Methods in the GrADS Project. NSF Next Generation Software Workshop, International Parallel and Distributed Processing Symposium, Santa Fe, (下转第 66 页)

- IEEE CS Press, Los Alamitos, CA,USA, April 2004.
- 4 Mayer A, et al. ICENI Dataflow and Workflow: Composition and Scheduling in Space and Time. UK e-Science All Hands Meeting, Nottingham, UK, IOP Publishing Ltd, Bristol, UK, September 2003.
 - 5 Birnbaum A, et al. Grid workflow software for High-Throughput Proteome Annotation Pipeline. 1st International Workshop on Life ScienceGrid (LSGRID 2004), Ishikawa, Japan, June 2004.
 - 6 Blythe J, et al. Task Scheduling Strategies for Workflow-based Applications in Grids. IEEE International Symposium on Cluster Computing and Grid (CCGrid), 2005.
 - 7 Deelman E, et al. Mapping Abstract Complex Workflows onto Grid Environments. Journal of Grid Computing, 2003,1:25 – 39.
 - 8 张绍华,顾宁,刘家茂,等.基于 D-Petri Net 和动态调度的网格工作流.计算机辅助设计与图形学学报, 2005, 17(6):1146 – 1151.
 - 9 郑凯.网格工作流的研究与完善[硕士学位论文].2006.
 - 10 邱俊珺,杨扬,郭文彩,丛林.网格 workflow 管理系统的研究与设计.微计算机信息,2006,27.
 - 11 Yu J, Venugopal S, Buyya R. A Market-Oriented Grid Directory Service for Publication and Discovery of Grid Service Providers and their Services. Journal of Supercomputing, Kluwer Academic Publishers, USA, 2005.