计算机科学2004Vol.31№.9A

基于服务质量的网格工作流算法优化

张云锋 葛 玮
(西北大学计算机科学系 西安710069)

摘 要 本文介绍了网格的基本概念,结合工作流任务的服务质量(QoS)需求,提出了基于服务质量的网格工作流调度算法,对GGWF算法中的LGSS算法做了改进,提出了ILGSS算法,对该算法的算法复杂度进行了分析,并在局域网环境下做了仿真实验,并给出了实验结果和分析,提出了在网格环境下探索自适应的工作流事务机制这一十分重要的研究方向,为网格环境下工作流的调度提供了一种新的解决方案。

关键词 网格,工作流,自适应,OGSA,QoS

An Improvement on Algorithm of Grid-Workflow Based on QoS

ZHANG Yun-Feng GE Wei
(Department of Computer Science,Northwest University,Xi'an 710069)

Abstract With the emergence of grid computing,new challenges have arisen in workflow tasks scheduling. The goal of grid-workflow task scheduling is to achieve high system throughput and to match the application needs with the available computing resources. This matching of resources in a non-deterministically share heterogeneous environment leads to concerns on Quality of Service (QoS). Grid concept is presented in this paper,coupled with the QoS requirement of workflow task,an improved algorithm—ILGSS algorithm,have been brought out. The complications of the improved scheduling algorithm have been analyzed. The experiment results show that the improved algorithm can lead to significant performance gain in various applications. An important research domain—adaptive workflow transaction in grid computing environment,has been explored,a new solution for the scheduling of distribute workflow has been bring forward in grid environment.

Keywords Grid,Workflow,Adaptive,OGSA,QoS

1 引言

网格(Grid)概念产生于20世纪90年代中期[1],是从电力网(Power Grid)概念借鉴过来的。网格的最终目的,是希望大家能够象使用电力一样方便地使用分布在网络上强大而丰富的计算能力。到目前为止,比较重要的网格体系结构有两个,五层沙漏模型和结合Web Service技术提出的开放网格服务体系结构OGSA(Open Grid Services Architecture)。由于传统的工作流在网格环境下遇到很多问题,本文旨在网格环境下的工作流任务调度提供一种近似最优的算法。

2 基于服务质量的工作流在网格环境中的应用

由于资源具有广域分布、动态、异构、自治等特性,传统的工作流机制在网格环境下遇到许多问题。首先,网格工作流的流程跨越多个管理域和组织,由于缺少统一的管理和控制,导致产生了信息缺少和不确定性因素;其次,网格资源不完全忠实于环境,计算和网络能力会随着时间变化很大,很难预测应用性能,在一个大型的网格环境下的实时资源信息的更新变得几乎不可能。网格环境下的资源映射是众所周知的NP完全且难问题,在网格环境下将工作流任务映射到网格资源是一个非常复杂且具有挑战性的问题,基于DAG任务映射的研究现在都集中在单管理域的异构集群模式。

我们提出的启发式算法称为效益函数启发式,传统映射算法的目标一般都是得到最短的任务完成时间(Makespan),然而在网格环境下,工作流的行为和系统资源状况都相当复杂,简单的追求最短完成时间有时并不合适,我们应对工作流进行区分,该启发式算法通过定义效益函数,在追求最短完成时间的同时,兼顾任务的服务质量(QoS)需求。因此,我们对前人研究的局域网格工作流调度算法做了一些改进。该算法通过定义任务的效益函数来评估任务的QoS需求。为了反映网格环境的分布性和动态性,每当工作流任务到达时,网格资源调度器通过网格信息查询服务获得当前网格可用资源的状态信息。在这样的环境中,网格资源发现机制不仅应考虑资源的位置信息,而且应考虑资源的属性语义的信息,将资源的位置信息以及资源的属性等语义信息考虑在我们的资源描述语言中,将网格环境下的各种资源(包括计算资源和)以统一的语言格式来描

张云锋 硕士研究生,主要研究方向:网格计算、工作流;葛 玮 副教授,硕士生导师,主要研究方向:软件工程、分布式计算、工作流技术。

述。

2.1 广域网格工作流调度算法

广域网格工作流管理主要接收来自用户的请求，一个工作流 W 可以被定义为一组子工作流的集合 $S_i(i=1,2,\cdots,n)$，包括两个检查点，S_1 和 S_n，分别代表开始和结束节点，我们定义 p_i 为 S_i 前面子工作流的数目，q_i 为 S_i 后面子工作流的数目，假定广域网格 G 是局域网格 $L_j(j=1,\cdots,n)$ 的集合，工作流管理的主要目的是寻找一种近似最优的调度策略 A，它是一个三元组 $(\pi^s,\pi^e,\zeta)_i$，π_i^s，π_i^e，ζ_i 分别定义为子工作流 S_i 的开始时间，结束时间和为子工作流 S_i 分配的局域网格，每一工作流都有一开始时间和结束时间，模拟就是在一个时间根据算法执行一个子工作流。

```
FOR i=1 TO i=n DO
πi^s=NULL,πi^e=NULL,ζi=NULL,Ri=FALSE,
ENDDO
π1^s=π1^e=CurrentTime ();R1=TRUE,
FOR lp=2 TO lp=n DO
 FOR i=1 TO i=n DO
  IF Ri=FALSE AND ALL Rip=FALSE (p=1,…,pi)
  BREAK
  ENDIF
  ENDDO
  πi^s=latest {πip^e|p=1,…,pi}
  IF i=n πi^e=πi^s
  ELSE
  (πi^e,ζj)=earliest {ILGSSj(Si,πi^s)|j=1,…,m};
  ENDIF
  Ri=TRUE;
  ENDDO
  FOR i=2 TO i=n-1 DO
  πi^s=earliest {πip^s|p=1,…,pi};
  ENDDO
  END
```

图1 GGWM 算法

2.2 改进的局域网格工作流调度算法

在一个局域网格中将工作流的任务流在网格资源上有序地执行就好像将整个工作流在局域网格中执行。一个很重要的区别就是局域网格的子工作流可能调度属于不同子工作流的多种任务。当多个任务同时请求相同的资源时，我们必须考虑处理冲突。

一个子工作流可以定义为一组任务的集合，表示为 $T_k(k=1,\cdots,l)$，每一任务请求特定的资源 R_k，定义 π_k^s 和 π_k^e 分别表示任务 R_k 的开始时间和结束时间。当发生资源冲突时，我们定义 π_k^s 为任务的可能开始时间，它不同于开始时间和结束时间，这样我们计算任务 T_k 的结束时间就有两个可能的结果，π_k^{e1} 和 π_k^{e1}，当没有冲突发生时，任务的可用时间等同于开始时间，两个可能的结束时间也不再有效，假如 T_c 是另一个与任务 T_k 有冲突的子工作流，T_0 为任务流的开始执行时间。我们有如下定义：

(1)D_i：用户定义的完成任务 T_i 的最终期限。

(2)$BTC(t_i,t_j,R)$：任务 t_i 和任务 t_j 在顺序存取临界网格资源 R 的效益函数。

(3)β_i 和工作流任务 t_i 相联系的效益函数，它定义工作流任务在一定时间内完成后，用户所得到的收益。用户可根据自己的需求定义各种效益函数。

ILGSS 算法的目的是在给定工作流 S 和开始时间 π^s 的前提下，综合考虑到效益函数的情况，对任务的完成时间作一个估算，即计算 π_i^e，实现 GGWM 算法中 ILGSS 函数的功能。

```
FOR k=1 TO k=l DO
πk^s=NULL,πk^e=NULL,Ri=FALSE,
ENDDO
π0^s=π0^e=π^s;R0=TRUE,
FOR lp=1 TO lp=l DO
  FOR k=1 TO k=l DO
    IF Rk=FALSE AND ALL Kkp=TRUE (p=1,…,pk)
      BREAK
    ENDIF
  ENDDO
  IF Rk=Rc
  πk^s=latest {πkp^s|p=1,…,pk}
  πc^e=latest {πcp^e|p=1,…,pc}
  πk^s=min {πk^s,earliest {πk^s,πc^e}};
  πc^e=min {πc^e,earliest {πk^s,πc^e}};
  if(πk^d> Dk)
    Modify Dk and make a next schedule
  if (πc^d> Dc)
    Modify Dc and make a next schedule
  //if Tk first execute
  πk^e1=sum {πk^s,πk^d};
  πc^e1=sum {latest {πc^e,πk^e1},πc^d};
  //if Tc first execute
  πc^e2=sum {πk^s,πc^d};
  πk^e2=sum {latest {πk^s,πc^e2},πk^d};
  βk^e= BTC (Tk,Tc,Rk);
  βc^e= BTC (Tc,Tk,Rk);
  πc^e1=πc^e1+ βk^e;
  πc^e2=πc^e2+ βc^e;
  πc^e=max {πc^e1,πc^e2};
  πk^e1=πk^e1+ βk^e;
  πk^e2=πk^e2+ βc^e;
  πk^e=max {πk^e1,πk^e2};
  IF Rk≠Rc
  πk^s=latest {πkp^s|p=1,…,pk};
  πk^e=sum {πk^s,πk^d};
  πc^e=latest {πcp^e|p=1,…,pc};
  πc^e=sum {πc^e,πc^d};
  ENDIF
  Rk=TRUE;
  Rc=TRUE;
  ENDDO
  FOR k=1 TO k=l-1 DO
    πk^e=earliest {πkq^e|q=1,…,qk};
  ENDDO
  FOR c=1 TO c=l-1 DO
    πc^e=earliest {πcq^e|q=1,…,qc};
  ENDDO
  END
```

图2 ILGSS 算法

局域子工作流的调度包括前进和回退的操作，ILGSS 与 GGWM 的最大区别是 ILGSS 存在冲突。在这种情况下，我们所选择任务的开始时间不能简单地认为是前驱任务的最迟完成时间，因为可能有其他的任务同时使用同一资源，本算法不同于以前算法的重要之处在于在追求最短完成时间的同时，兼顾任务的服务质量(QoS)需求，当存在冲突时，如果随着调度任务的执行，其他冲突任务的效益函数值下降超过了我们服务质量(QoS)的阈值，则要重

新进行调度，并对正在执行的任务执行回退操作。

3 时间复杂度分析

在网格计算环境中，对任务调度和分配的复杂度进行分析，往往只是局限于时间复杂度的分析，而通常不对算法的空间复杂度予以分析。在本文中我们仅限于时间复杂度的分析。设 n 为网格子工作流的个数，ILGSS 算法的时间复杂度为 n^2，而 GGWM 算法的时间复杂度为 n * ILGSS 算法的时间复杂度，GGWM 算法在最坏情况下即产生冲突的情况下的时间复杂度为 n^3。

4 算法仿真结果

为了评估我们的算法的执行效果，我们设计了一个模拟程序，模拟环境由13台主机分布在两个局域网内，两个局域网内分别由5台主机和8台主机组成，相当于两个局域网格，并且有5个数据源，模拟有个两个工作流 W1和 W2分别由5个子工作流 S1、S2、S3、S4、S5组成，每个任务必须存取一定的数据源来协同完成办公自动化的功能，任务的运行时间已知，假设 W1中的子工作流 S3中的 A2现在执行，而同时 W2中的 S3被调度，这样在 A3和 A4中就存在资源冲突。由于在网格环境下任务的到达时间是不可预测的，事件系统之间存在着交互、并发以及状态和事件之间的冲突限制。

在实验中我们比较了经过改进的 GGWM 算法和传统启发式算法的性能，图3显示的是在映射资源时考虑到任务的 QoS 需求，优先映射接近最终时限的任务和效益函数较高的任务，但是以 Makespan 的增加为代价的。但和传统的启发式算法相比增幅不大，在照顾效益函数的同时取得了良好的效果。

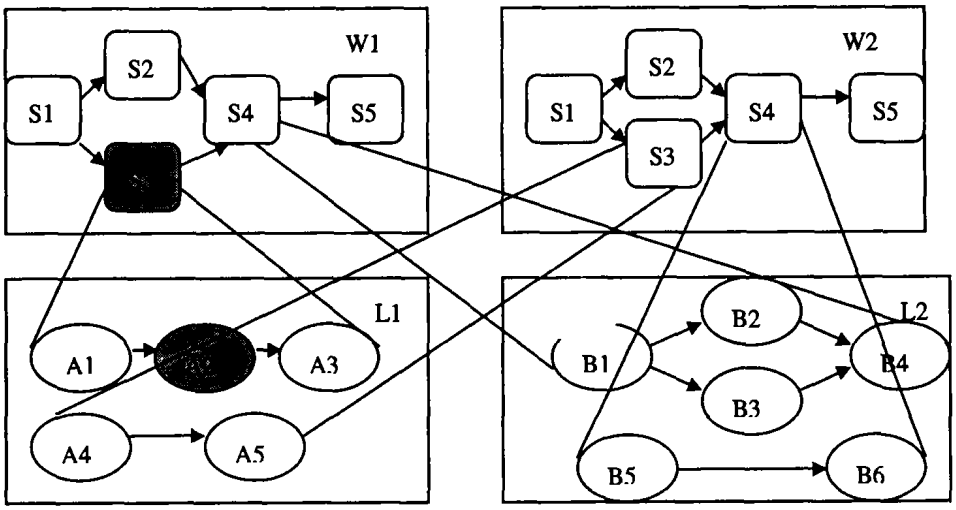

图3 实验的拓扑结构

我们定义 $\pi_i^s(\tau)$为任务 T_i 的可能开始时间，$\pi_i^s(\tau)$为任务 T_i 的开始时间，$\pi_i^e(\tau)$是任务 T_i 的结束时间，$\pi_i^{ei}(\tau)$是任务 T_i 未考虑效益函数的可能结束时间，$\pi_i^{ai}(\tau)$是任务 T_i 考虑效益函数的可能结束时间，在实验中，我们预定义如下：

$\pi_3^s(\tau)=(3,5,5,7)$ $\pi_3^d(\tau)=(5,6,7,8)$ $\pi_4^s(\tau)=(0,3,3,5)$ $\pi_4^d(\tau)=(10,12,14,16)$ $\pi_5^d(\tau)=(3,6,4,9)$BTC $(t_3,t_4,R,)=(3,5,7,9)$BTC $(t_4,t_3,R,)=(4,6,5,3)$ $D_3=24$ $D_4=30$ $D_5=35$

则具体的调度过程如下：

$\pi_3^s(\tau)=\min\{(3,5,5,7),\text{earliest}\{(3,5,5,7),(0,3,3,5)\}\}=(0,3,3,5)$

$\pi_4^s(\tau)=\min\{(0,3,3,5),\text{earliest}\{(3,5,5,7),(0,3,3,5)\}\}=(0,3,3,5)$

$\pi_3^{e1}(\tau)=\text{sum}\{(0,3,3,5),(5,6,7,8)\}=(5,9,10,13)$

$\pi_4^{e1}(\tau)=\text{sum}\{\text{latest}\{(5,9,10,13),(0,3,3,5)\},(10,12,14,16)\}=(15,21,24,29)$

$\pi_4^{e2}(\tau)=\text{sum}\{(0,3,3,5),(10,12,14,16)\}=(10,15,17,21)$

$\pi_3^{e2}(\tau)=\text{sum}\{\text{latest}\{(10,15,17,21),(3,5,5,7)\},(5,6,7,8)\}=(15,21,24,29)$

$\pi_3^{a1}(\tau)=\text{sum}\{(5,9,10,13),(3,5,7,9)\}=(8,14,17,22)$

$\pi_4^{a1}(\tau)=\text{sum}\{(15,21,24,29),(3,5,7,9)\}=(18,26,31,38)$

$\pi_3^{a2}(\tau)=\text{sum}\{(15,21,24,29),(4,6,5,3)\}=(19,27,29,32)$

$\pi_4^{a2}(\tau)=\text{sum}\{(10,15,17,21),(4,6,5,3)\}=(14,21,22,24)$

$\pi_3^a(\tau)=\max\{\pi_3^{a1}(\tau),\pi_3^{a2}(\tau)\}=(8,14,17,22)$

$\pi_4^a(\tau)=\max\{\pi_4^{a1}(\tau),\pi_4^{a2}(\tau)\}=(14,21,22,24)$

$\pi_5^a(\tau)=\text{sum}\{(14,21,22,24),(3,6,4,9)\}=(17,27,26,33)$

由于服务质量需求将会对任务调度的性能产生很大影响，对于实际的应用背景，在实验中我们模拟

了三种情形：

(1)90%的工作流任务要求在36个小时内完成，而其他的任务没有服务质量要求。

(2)70%的工作流任务要求有1.0Gigabit 的网络带宽，而其他任务没有网络带宽的要求。

(3)50%的工作流任务要求在距离最近的主机上执行，而其他任务没有这方面的要求。

(4)30%的工作流任务要求在满足特殊要求的主机上执行，而其他任务没有这方面的要求。

(5)10%的工作流任务要求在经济效益函数较高的主机上执行，而其他任务没有这方面的要求。

我们对传统的基于遗传算法的启发式算法和基于服务质量的遗传算法进行了比较，实验中我们分别创建了100个工作流任务，然后对实验结果的Makespan 取平均值。(单位：秒)

表1 实验结果的比较

序号	未考虑效益函数	考虑效益函数	改进度
(1)	432.16	412.30	4.59%
(2)	500.60	474.60	5.19%
(3)	512.86	478.68	6.66%
(4)	574.52	524.36	8.73%
(5)	654.76	583.94	10.81%

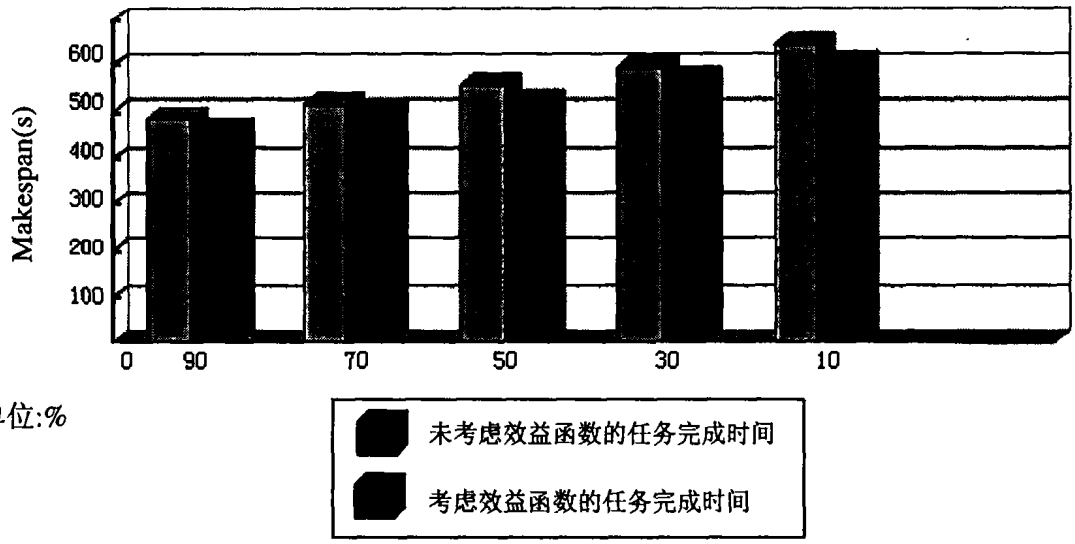

图4 实验结果比较图

5 进一步要研究的问题

现有的大多数工作流系统提供管理业务过程的功能，支持自动执行，监控活动状态，但缺乏一致性控制，并发控制和错误恢复的能力。然而这些特性对复杂业务中的活动却尤为重要。特别在网格环境下，一个事务可能持续很长时间，同时由于资源的自治性，用户可能无法锁定需要的资源，因此其事务处理必须放松 ACID 语义。上海交大李明禄教授提出了网格环境下的一种事务协调机制及其补偿技术，同时具备管理短时操作和长时间的商业活动的能力，通过提交前的回滚和提交后的故障恢复机制，在网格环境下探索自适应的工作流事务机制，已经成为一个十分重要的研究方向。

结论 网格计算是一个正在迅速发展的研究领域，本文结合工作流任务的服务质量(QoS)需求，提出了基于服务质量的网格工作流调度算法，对该算法的算法复杂度进行了分析，并在局域网环境下做了仿真实验，并给出了实验结果和分析，为网格环境下分布式工作流的调度提供了一种解决方案。不管是网格计算还是分布式工作流还很不成熟，随着研究的深入与不断发展，基于网格的工作流技术会不断得到完善和发展，这也会直接推动网格计算及其应用的发展。

参 考 文 献

1 Basney J, Livny M. Deploying a High Throughput Computing Cluster. High Performance Cluster Computing, Prentice Hall, 1999

2 Bhatia D, Burzevski V, Camuseva M, Fox G, Furmanski W, Premchandran G. WebFlow-a Visual Programming Paradigm for Web/Java Based Coarse Grain Distributed Computing [J]. Concurrency: Practice and Experience, In: Proc. of Supercomputing 2001. 1997,9(6):555～577

3 GridFlow: Workflow Management for Grid Computing [J]. Junwei Cao, Stephen A. Jarvis, Subhash Saini and Graham R. Nudd C&C Research Laboratories, NEC Europe Ltd., Sankt Augustin, Germany Department of Computer Science, University of Warwick, Coventry, UK NASA Ames Research Centre, Moffett Field, California, USA 2003

4 Murata T. Temporal Uncertainty and Fuzzy-Timing High-Level Petri Nets[J]. invited paper In: Proc. Of Application and Theory of Petri Nets, LNCS 1091. 1996. 11～28

5 Workflow Management Coalition Workflow Standard-Interoperability Abstract Specification. http://www.wfmc.org 1999

6 都志辉，陈渝，刘鹏，王小鸽，杜江，周念生. 以服务为中心的网格体系结构 OGSA. 清华大学计算机系，2003

7 都志辉，陈渝，刘鹏. 网格计算，清华大学出版社，2002

8 丁箐，陈国良，顾钧. 计算网格下一个统一的资源映射策略. 软件学报，2002

9 唐飞虎，李明禄，曹键. 网格环境下的一种事务协调机制及其补偿技术. 计算机研究与发展，2003