

文章编号: 1007- 2985(2008) 04- 0042- 03

# 计算网格中基于负载平衡的调度模式\*

徐洪智, 黄天强, 张彬连

(吉首大学信息管理与工程学院, 湖南 张家界 427000)

**摘要:** 对资源进行有效的管理和调度可以提高网格系统的利用率. 针对计算网格中的负载平衡问题, 为减少网络通信量, 提出一种分布式的网格作业调度模型, 并给出了其相关算法: 将轻负载节点逐个收集到一个队列, 并设定一个可调节的刷新时间, 当重负载节点提出调度请求时, 直接从轻载节点队列中取一轻载节点并把负载转移到该节点上, 从而解决了资源调度中的负载平衡问题.

**关键词:** 计算网格; 负载; 调度模式

**中图分类号:** TP393.01

**文献标识码:** A

网格技术是当今计算机领域研究的热点, 把计算资源、数据存储资源、广域网络、仪器设备等连成有机的整体, 方便用户使用网格中的任何资源. 任务调度是网格计算的关键技术之一, 对网格用户而言, 只需向网格系统提交任务, 由网格系统的调度程序按照某种调度策略把用户提交的任务分配给网格系统中的可用资源去完成. 作为调度的目标之一, 负载平衡是计算网格的重要特性. 针对负载平衡, 人们进行了大量的研究, 并取得了很多成果<sup>[1-3]</sup>, 但这些都是针对特定网格环境, 主要对调度算法本身进行的研究. 也有一些针对调度模式进行研究, 如文献[4]提出资源调度的三层模式, 但层次之间的通信会影响系统的总体性能, 文献[5]采用主分发器和次调度器的两层调度模式, 加快了调度速度, 但没有充分考虑计算结点上的负载平衡, 文献[6]考虑了负载平衡, 但主调度器查找轻载结点过于频繁, 影响了系统的性能. 基于以上分析, 笔者提出一种新的负载平衡调度模式, 由从调度节点自动发出轻载信号, 主调度节点事先收集这些信息, 当重载节点提出负载转移请求时, 不必每次都查询从调度节点负载情况, 有效减少了网络通信量.

## 1 网格系统模型

为了建立网格系统模型, 使用基于网格性能的主机资源聚类<sup>[7]</sup>, 根据网格应用对链路状态(带宽、延迟等)的 QoS 要求进行资源划分, 这样资源集合内部主机间连接为高带宽、低延迟, 而不同资源集合主机间的连接是相对低速的. 由此, 每一个广域节点是资源聚类产生的一个节点集合, 它相当于一个局域网格, 假设广域节点(局域网格)本身具有自治性, 即根据自身的特点能有效地对内部资源进行管理. 通过此类资源聚类方法, 在全局范围内, 笔者可以将网格抽象为若干个广域节点组成的分布系统, 然后对此分布系统进行资源管理和调度.

**定义 1** 广域节点  $N_i = \{n_{i1}, n_{i2}, \dots, n_{im}\}$ , 通过资源聚类产生的一个资源节点集合. 所有广域节点集合组成网格计算系统  $G = \{N_1, N_2, \dots, N_k\}$ .

\* 收稿日期: 2008-05-04

基金项目: 湖南省教育厅科学研究项目(07C523)

作者简介: 徐洪智(1974-), 男, 湖南望城人, 吉首大学信息管理与工程学院讲师, 硕士, 主要从事网格计算、算法设计与分析研究.

定义 2 广域节点能对内部资源进行管理和调度, 文中称之为调度节点. 在整个网格系统中, 对各广域节点进行负载分配和管理的节点称为主调度节点.

实际上, 调度节点相当于局域网络群首, 可以通过选举算法<sup>[8]</sup>产生, 主调度节点也可以通过同样的算法产生. 根据以上描述, 网格系统模型如图 1 所示.

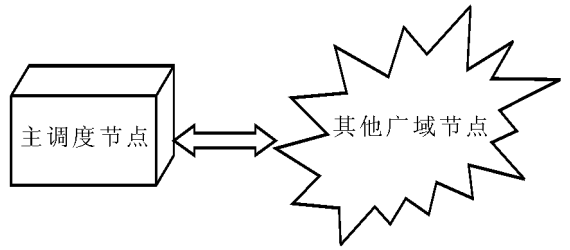


图 1 网格系统模型

为了更好地描述网格负载均衡算法, 可将各节点的负载状态刻画为 3 种: 轻载、适度、重载.

定义 3 一个计算节点是重载的, 主要指该节点的 CPU 利用率高、CPU 就绪队列长度长等; 相应地, 一个计算节点是轻载的主要指该节点的 CPU 利用率低、CPU 就绪队列长度短等. 一个计算节点是适度的主要指该节点处于重载和轻载之间的一种状态.

系统初始时, 主调度节点组播信息到系统中的各个调度节点, 表明自己是主调度节点, 其它调度节点在需要调度负载的情况下将请求负载转移的信息发送给主调度节点.

## 2 调度策略

在主调度节点上建立一张全局轻载节点表 LightWeightNode, 存放网格系统中轻载从调度节点信息, 并把这个表组织成优先队列的方式. 每个从调度节点隔一个时间间隔取该节点的负载信息, 当确认自身负载是重载时, 发送“重载”信息到主调度节点, 要求转移自身的负载到其它节点; 当确认自身目前是轻载时, 则发送“轻载”信息到主调度节点, 表明该节点还可以继续加载其它的任务, 该信息被主调度节点确认后, 只要该节点上没有负载波动, 该节点将不再往主调度节点发送任何信息, 如算法 1 所示. 主调度节点收到从调度节点发送的信息后, 如果发来的是轻载信息则记录到全局表, 如果发来的是重载信息, 则查找全局表, 发现有轻载节点才可能进行任务调度, 否则不进行任何调度, 如算法 2 所示.

### 算法 1 从调度节点发送负载信息算法

```

if( 确认自身重载)                                     { if( LoadState! = "轻载" )
{ if( LoadState! = "重载" )                           { 发送轻载信息到主调度节点, 说明该节点还可以
    { 发送重载信息到主调度节点, 要求转移自身负载;   执行其它的任务;
    }                                                    LoadState= "轻载";
}                                                         }
if( 确认自身轻载)                                     }
    
```

算法 1 中, 如果从调度节点由其它状态变为重负载情况, 则发送负载转移请求, 如果负载没有被转移, 则隔一段时间还可以继续提出请求; 如果从调度节点由其它状态变为轻载状态, 则只发一次轻载信号到主调度节点, 以减少网络通信量.

### 算法 2 主调度节点调度任务算法

```

if( 收到调度节点发来的负载转移请求)                 }
{ if( 轻载节点队列 LightWeightNode 为空)             else
  { 发送拒绝负载请求信息到请求节点;                 { 组播轻载确认信息到队列中所有节点;
    return;                                           根据返回的信息重建队列;
  }                                                    取队列中第一个节点, 发送该节点的信息到请求节点,
  }                                                    并将请求节点的部分任务调度到该节点上.
  CurNode= 从 LightWeightNode 队列中取第一个节点;
  if( 刷新时间没有失效)                               }
  { 发送 CurNode 的信息给请求节点, 并将请求节点
    }
    
```

的部分任务调度到 CurNode 上;

考虑到网格中各节点的负载是动态变化的, 计算网格中的某个节点的负载可能随时发生变化, 但可以假定在一个有效的时间内, 各节点的负载是相对稳定的, 即算法 2 中设定的刷新时间之内(刷新时间可以

根据实际网格情况进行调节),各节点的负载是相对稳定的,这时直接在队列中取出的轻载节点就是有效的节点,运用算法 2 能把重载节点上的负载转移到轻载节点上.对于算法 2 中优先队列的构造,笔者把最有效的轻载节点放到最前面,具体是把最后一次发送轻载信号的节点放到队列的最前面,这样可以保证每次取出的轻载节点是有效的.在进行负载转移时,为了减少转移本身带来的开销,笔者采用转移作业的方式,即只转移重载节点上尚未执行的作业,通过作业再分配,使网格各节点的负载达到平衡.

### 3 结语

针对广域范围内分布、异构、动态的资源调度问题,笔者应用资源聚类的方法构造了一个网格系统模型,该模型主要由主调度节点和其它从调度节点构成,系统运行时从调度节点主动发送轻载信号到主调度节点,主调度节点收集这些信息,当某个节点提出负载转移请求时,主调度节点即将负载转移到已收集的轻负载节点上,实现了网格负载平衡,并有效减少了网络通信.

### 参考文献:

- [1] BANINO C, BEAUMONT O, CARTER L. Scheduling Strategies for Master-Slave Tasking on Heterogeneous Processor Platforms [J]. IEEE Transactions on Parallel and Distributed Systems, 2004, 15(4): 319-330.
- [2] 林伟伟,齐德昱,李拥军,等. 树型网格计算环境下的独立任务调度 [J]. 软件学报, 2006, 17(11): 2352-2361.
- [3] 徐洪智,李仁发,张彬连,等. 一种快速平衡任务的网格调度算法 [J]. 系统仿真学报, 2007, 19(11): 2437-2439.
- [4] 熊磊,李元哲. 网格计算资源调度策略的三级模式 [J]. 计算机工程与应用, 2005, 41(1): 96-97.
- [5] 黄瑾,金海,谢夏. 网格系统中的层次化资源分配与任务调度 [J]. 华中科技大学学报(自然科学版), 2006, 34(10): 51-54.
- [6] 韩向春,潘勋,陈晶. 计算网格中动态负载平衡的分布调度模式 [J]. 计算机工程与设计, 2007, 28(12): 2845-2847.
- [7] 桂小林,王庆江,龚文强,等. 面向网格计算的机器选择算法研究 [J]. 计算机研究与发展, 2004, 41(12): 2189-2194.
- [8] 李航,赵志刚,王光兴. 基于代理群的网络管理中群首选举算法的研究 [J]. 计算机科学, 2005, 32(5): 64-66.

## A Scheduling Pattern Based on Task Balancing in Computational Grid

XU Hong-zhi, HUANG Tian-qiang, ZHANG Bin-lian

(School of Information Management and Engineering, Jishou University, Zhangjiajie 427000, Hunan China)

**Abstract:** Effective management and scheduling of resources can increase the utilization of grid system. For the issues of task-balancing in grid computing, in order to reduce the communication, a distributed scheduling pattern in grid is proposed, and the relevant algorithm is given. The algorithm collects each light load node to a queue, and sets an adjustable refresh time, when the heavy load nodes requests to schedule, the task will be transferred to the node which is taken out from the queue so as to solve the issues of task-balancing in grid.

**Key words:** computational grid; task; scheduling pattern

(责任编辑 陈炳权)