

光网络环境中最先开始路径优先的自适应路由算法

刘波涛

(湖南文理学院 计算机科学与技术学院, 湖南 常德 415000)

摘要: 为了优化光网络环境下分布式计算系统的资源调度性能,提出了一种最先开始路径优先的自适应路由算法。该算法基于 Dijkstra 最短路径优先算法,通过引入一个时间标记变量来估计从源节点到当前目标节点的最先可用时间,绕过调度过程中产生拥堵的链路,选择能够最先开始通信的路由,从而减小通信竞争冲突,缩短了调度长度。仿真结果表明,该算法能够使用较少的网络链路资源来获得最短的调度长度。

关键词: 光网络; 自适应路由; 分布式计算系统; 列表调度算法

中图分类号: TP18 **文献标志码:** A **文章编号:** 1001-3695(2012)07-2621-03

doi:10.3969/j.issn.1001-3695.2012.07.060

Novel adaptive routing algorithm with strategy of earliest start path first in optical network

LIU Bo-tao

(College of Computer Science & Technology, Hunan University of Arts & Science, Changde Hunan 415000, China)

Abstract: To improve the resource scheduling performance of distributed computing system in the optical network, this paper proposed a novel adaptive routing algorithm with the strategy of earliest start path first in the optical network. The proposed adaptive routing algorithm could bypass the link congestion in the process of scheduling produces, chose the route that was able to communicate the very first by introducing a time mark variable to estimate the headmost available time from the source node to the goal node, reduced communication competition conflicts, and shorter the scheduling length. The simulation results show that the proposed algorithm can use less network link resources to get the shortest scheduling length.

Key words: optical networks; adaptive routing; distributed computing system; list scheduling algorithm

随着科学计算、协同设计、虚拟现实等新型应用的出现,传统的基于 IP 分组交换网络的分布式计算系统越来越无法满足这种数据密集型的应用需求。智能光网络可为数据密集型应用提供大带宽、低延迟的传输能力,为此,研究人员将目光转向了直接在光网络上构筑分布式计算系统^[1]。目前,列表调度算法是求解光网络环境下分布计算系统资源调度问题的主流算法^[2]。列表调度算法在任务调度过程中能够对通信端口、链路资源竞争等约束条件进行判断和处理,从而实现任务与通信资源的联合调度;但该算法无法避免或减少通信竞争冲突^[3]。另外,由于通信竞争冲突的存在,当前通信请求只有等待冲突资源被释放后才能得到调度,这样容易导致较长的调度长度。可以从路由计算方面来实现对光网络环境下分布计算系统资源调度性能的优化:文献[4]提出了基于负载均衡的路由算法的思想能够缓解网络因为过多地使用某条链路而引起的拥塞和较少使用某条链路而导致的资源浪费等问题,并把此算法运用到智能光网络中;文献[5]提出了一种基于 QoS 的最短路径优先算法,该算法在路由计算之后,对路径的 QoS 系数进行比对,选择能满足更高 QoS 要求的路径作为工作路径来提高网络资源的利用率;文献[6]提出了一种基于矩阵运算的路由算法来解决全交叉光互连网络中光信号路由的选择和控制问题。

不同于上述文献,本文提出了一个自适应路由算法,该算

法在调度某条光路时能够绕过拥塞链路并选取一条最先开始的最短路由,从而优化资源调度。

1 路由选择策略

光网络中建立光路一般有三种类型的路由选择策略:

a) 固定路由策略 (fixed routing)。网络中任意节点对间每次建立光路连接均采用相同的路径。该路径为源宿节点间最短路径,通常以离线的方式预先计算好,常用的最短路径算法有 Dijkstra 算法和 Bellman-Ford 算法。该路由策略实现简单,但易引起某些链路资源过度消耗,从而增加链路的阻塞率。

b) 固定备选路由策略 (fixed-alternate routing)。每个网络节点到任意其他节点间存在多条固定路由,这些路由按优先级顺序存放在一张路由列表中,列表中的第一条路由为主用路由 (primary route),其他为备选路由。备选路由与主用路由间不能占用任何相同的链路。当连接请求达到时,源节点从路由列表中按照从主用路由到备用路由的顺序依次建立连接直到成功;如果所有备选路由都无法建立连接,则当前请求被阻塞。与固定路由策略相比,该类策略可以显著降低连接阻塞概率,并可对链路故障实现一定的容错性。

c) 自适应路由策略 (adaptive routing)。该策略根据网络当前状态动态计算源宿节点间的路由。网络状态是由当前已建立的连接决定的。自适应路由策略一般有两种形式:(a) 自

收稿日期: 2011-11-18; 修回日期: 2011-12-29

作者简介: 刘波涛 (1976-), 男, 湖南常德人, 讲师, 硕士, 主要研究方向为人工智能算法、计算机应用技术等 (hncdlbt@163.com)。

适应最小代价路径策略,即为网络的每条链路按照某种策略计算出相应的代价(如链路代价为当前剩余带宽的倒数),然后确定源宿节点的最小代价路径;(b)最小拥塞路径策略,即每个源宿节点对事先计算出一组备选路由,当连接请求到达时,在备选路由中挑选带宽占用最少(或剩余带宽最多)的路由。自适应路由策略比固定备用路由策略获得更低的阻塞概率。

2 自适应最先开始路径优先算法

列表调度算法是一种静态调度算法,在任务执行之前根据当前系统状态得到预期调度结果,然后系统根据调度表进一步将任务迁移到指定的资源上完成执行过程。在调度计算过程中,每完成一次边调度,就会改变未来网络链路资源的预留情况。而已有的路由算法只能用于当前时刻下的网络,即基于当前时刻的网络资源在空间上的分布情况计算路由。因此在联合调度过程中为了避免未来调度时刻网络资源的拥塞和冲突,需要路由算法能够根据已调度的网络资源状态计算路由。

列表调度算法属于贪婪算法,即在调度每个任务时,总是将任务调度到使之最先开始或最先完成的资源上。很多自适应路由算法都是基于 Dijkstra 最短路径优先算法,算法中通过一个距离变量 $distance(n)$ 记录当前节点到源节点 s 的距离估计,从而在空间维度找到最短路由。为了在时间维度找到最先开始的路由,本文在 Dijkstra 算法中引入一个时间标记变量 $time(n)$ 用于记录当源节点 s 到当前节点 n 的最先可用时间的估计。本文将该算法命名为自适应最先开始路径优先算法(adaptive earliest start path first, AESPF)。算法中用到的符号定义如下:

$G_{ON} = (S, L_T, d)$:表示扩展资源图 $G_R = (N, L, type, bw, d)$ 中的光网络部分,其中 $S \subset N, L_T \subset L$ 。

$time(n)$ 为时间标记变量,记录从源节点到当前节点 $n \in S$ 的最先可用时间。

$distance(n)$ 为距离标记变量,记录源节点到当前节点 $n \in S$ 的最短距离。

$pred(n)$ 为记录从源节点到当前节点 $n \in S$ 的最短路径上 n 的前继节点。

F 为节点集合,该集合中的节点到源节点间最短路径已经确认。

U 为节点集合,该集合中的节点到源节点间的最短路径还未找到。

本文设计的自适应路由算法如下:

```
//ssrc is source switch node
//sdst is destination switch node
//B is the request bandwidth
begin
  for all  $s \in S$  do
     $time(s) = \infty$ 
     $distance(s) = \infty$ 
     $pred(s) = NULL$ 
  end for
  add  $s_{src}$  to  $U$ 
   $distance(s_{src}) = 0$ 
   $time(s_{src}) = 0$ 
  while  $U$  is not empty do
     $s_i = \text{extract\_minimum}(U)$ 
    add  $s_i$  to  $F$ 
```

```
    relax_neighbors( $s_i$ )
  end while
  according to  $F$  and  $pred()$ ,
  output the route from  $s_{src}$  to  $s_{dst}$ 
end
```

说明:2~10 行对相关变量进行初始化操作;while 循环体(11~15 行)中 extract-minimum 子程序每次从集合 U 中取出一个具有最小 time 值的节点 s_i ,即 $time(s_i) = \min_{s \in U} \{time(s)\}$ 。如果有多个节点具有相同的 time 值,则从中选择具有最小 distance 值的节点。然后将 s_i 从集合 U 中删除,并插入到集合 F 中;之后由 relax-neighbors 子程序遍历 s_i 的所有邻接节点 s_j 间的链路 l_{ij} ,如果到达 s_j 的路径的最先开始时间被延迟,则更新相应的 $time(s_j)$ 、 $distance(s_j)$ 以及 s_j 的前继节点 $pred(s_j)$ 。AESPF 算法最终可找到从源节点到任意目的节点间的最先开始的路由(如果存在多条开始时间相同的路由,则选择其中的最短路由)。

算法复杂度分析:AESPF 算法是对 Dijkstra 算法的扩展,由于引入的 time 变量与之前的 distance 变量操作相同,并没有增加原有算法的计算复杂度。因此 AESPF 算法的时间复杂度与 Dijkstra 算法相同,为 $O(|S| \log |S| + |L_T|)$,其中 $|S|$ 为光网络交换节点数量, $|L_T|$ 为光网络传输链路数量。

3 实验及分析

为了验证本文算法性能,下面对固定路由算法(fixed routing algorithm, FRA)、固定备选路由算法(fixed 2-alternate routing algorithm, F2ARA)与本文提出的自适应最先开始路径路由算法(alternate earliest start path first routing algorithm, AESPFRA)进行对比实验。图 1 是调度示例的配置情况,图 1(a)表示一个包含七个计算任务的 DAG 图,其中所有节点上均标记了平均计算代价和计算类型,所有边上标记了传输数据量的大小;(b)中有六个计算资源,由一个四节点的光网络互连。计算资源上标记了与计算任务相匹配的计算类型。为了突出通信竞争冲突的效果,本文假定资源图中所有链路上均有一个单位的带宽。本文还假定所有计算资源的异构因子为 1,即相同类型的资源具有相同的计算能力。

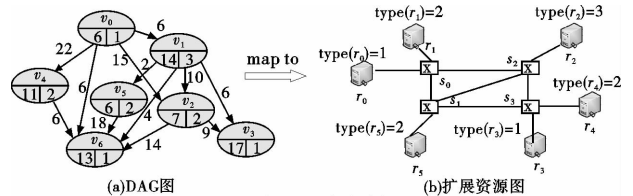


图1 调度实例

三种路由算法用甘特图表示的调度结果分别如图 2~4 所示。

从图 2~4 可以看出:三种路由算法在前六个任务($v_0 \sim v_5$)的调度结果是一样的,不同之处发生在调度最后一个任务 v_6 上。根据类型约束条件, v_6 只能调度到资源 r_0 和 r_3 , v_6 的所有入边按照 $e_{56}, e_{16}, e_{06}, e_{26}, e_{46}$ 的顺序调度。

在固定最短路由算法中, v_6 调度到 r_3 上,边 e_{56} 使用固定路由 $\langle 101, 113 \rangle$ 。由于链路 101 已经被边 e_{04} 占用,所以 e_{56} 只能延迟 15 个时间单位才能开始通信,从而使得总调度长度为 110。当 v_6 调度到 r_0 上时, e_{56} 的通信代价为 0,因此调度长度减少为 109,比调度到 r_3 上短,最终选择 r_0 。

在固定备选路由算法下, v_6 调度到 r_3 上反而获得更短的

调度长度为 99。这是因为 e_{56} 可以调度到另一个备选路径 $\langle 102, 123 \rangle$ 上。尽管备选路径上仍存在来自 e_{12} 的冲突,但仅造成 4 个时间单位的延迟。

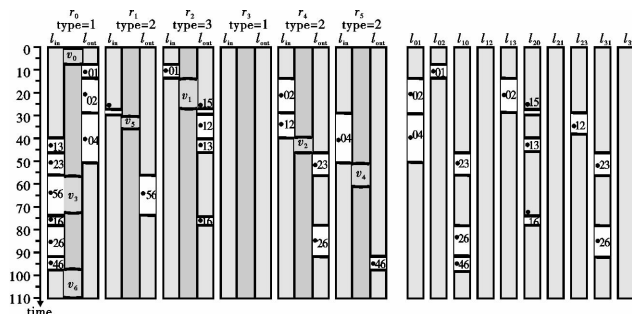


图2 固定最短路由算法的调度结果

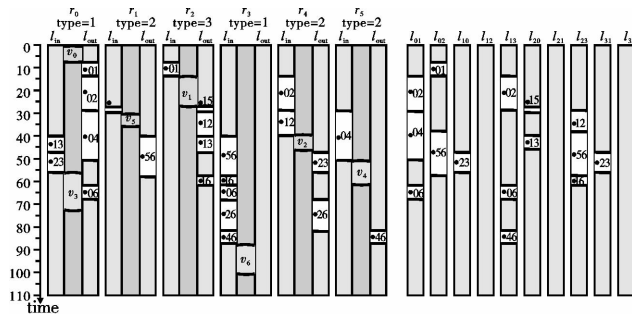


图3 固定备选路由算法的调度结果

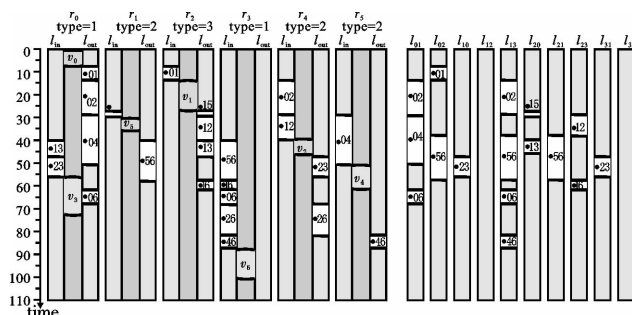


图4 自适应最先开始路径路由算法的调度结果

在自适应最先开始路径路由算法下,当 v_6 调度到 r_3 上时, e_{56} 可以调度到一条新计算的路由 $\langle 102, 121, 113 \rangle$ 上,同时避免了来自链路 101 和 123 的资源竞争冲突,使得 e_{56} 能够在 v_5 结束后立即开始通信。所以自适应最先开始路径路由算法获得了最短的调度长度 95。

对调度结果的另一个测度是资源使用情况(这里只考察 DAG 调度对光网络资源的使用情况)。根据调度结果可知,对

光网络资源使用应包含两个维度,即占用的链路带宽和带宽的使用时间。因此本文定义网络资源使用值(network resource usage, NRU)为调度期间内使用到的链路带宽与其被使用时间的乘积,即

$$NRU = \sum_{l \in L_T} \sum_{s=1}^{z_{\max}-1} [bw(l) - A(t, l)] \cdot (t_{z+1} - t_z)$$

本例中:固定最短路由算法调度后的光网络的资源使用值为 131;固定备选路由算法调度后的光网络的资源使用值为 171;而本文算法调度后的光网络的资源使用值为 148。这说明本文算法能够使用较少的网络链路资源来获得最短的调度长度。

4 结束语

由于光网络巨大的传输能力和较低的传输时延,光网络是最合适的网格基础传输网络,在光网络上构筑的分布式计算系统能够满足数据密集型的分布式科学计算应用。由于通信竞争冲突的存在,当前通信请求只有等待冲突资源被释放后才能得到调度,这样容易导致较长的调度长度。本文基于 Dijkstra 最短路径优先算法,提出了一个自适应最先开始路径优先算法。该算法绕过调度过程中产生拥堵的链路,选择能够最先开始通信的路由,从而减小通信竞争冲突,缩短了调度长度。仿真结果表明本文算法能够使用较少的网络链路资源来获得最短的调度长度。

参考文献:

- [1] GUO Wei, JIN Yao-hui, SUN Wei-qiang. Distributed computing over optical networks [J]. *Optical Communications*, 2011, 35(3): 123-133.
- [2] 韦乐平. 光网络的发展、演进和面临的挑战[J]. *中兴通讯技术*, 2002, 8(4): 18-23.
- [3] WANG Ying, JIN Yao-hui, GUO Wei. Joint scheduling for optical grid applications [J]. *Journal of Optical Network*, 2007, 6(3): 304-318.
- [4] 黄铁瑛, 王辉. 基于负载均衡的智能光网络路由算法[J]. *微电子学与计算机*, 2010, 27(3): 130-132.
- [5] 李思衡, 王辉. 基于 QoS 的智能光网络中约束路由算法研究[J]. *光通信技术*, 2010, 34(1): 15-17.
- [6] 杨俊波, 杨建坤, 李修建, 等. 全交叉光互连网络的路由选择与控制[J]. *光学精密工程*, 2010, 18(6): 1249-1257.