

基于存储的光电路交换网络资源调度

张涛, 郭薇

(上海交通大学 电子信息与电气工程学院, 上海 200240)

摘要: 在链路带宽具有不均衡性的传输网络中, 传统电路交换(TCS)端到端的建路机制会造成带宽资源浪费。针对此问题, 提出了一种基于存储的资源调度算法, 即中间存储转发调度(ISTS)算法。仿真结果表明, 与TCS相比, ISTS能够有效提高传输成功率和链路利用率。

关键词: 大数据传输; 光电路交换; 资源调度; 静态业务; 数据存储

中图分类号: 915.41 **文献标识码:** A **文章编号:** 1002-5561(2018)02-0005-04

DOI: 10.13921/j.cnki.issn1002-5561.2018.02.002

Resource scheduling of optical circuit switching network based on storage

ZHANG Tao, GUO Wei

(School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: In transport networks with unbalanced link bandwidth, the traditional circuit switching (TCS) end-to-end path establishment mechanism will cause bandwidth resource wastage. For this problem, a storage-based resource scheduling algorithm, Intermediate Store and Transfer Schedule(ISTS) algorithm has been proposed in this paper. The simulation results show that, compared with TCS, ISTS algorithm can effectively improve the transmission success rate and link utilization.

Key words: bulk data transfer; optical circuit switch; resource scheduling; static traffic; data storage

0 引言

亚马逊、微软和 Facebook 等云服务提供商为了追求所提供的服务更靠近用户且能耗更低, 通常会在全球建立多个数据中心^[1]。为了保证服务质量和提高访问的容错率, 这些云服务提供商需要定期同步和备份数据中心之间的数据, 而这些数据量通常在 TB 到 PB 范围之间^[2], 这也意味着大数据传输会占据大量的带宽资源, 并花费高昂的传输成本。光电路交换因为其低能耗、大带宽和高可靠性等优点成为大数据传输的有效方案。

在数据中心间的传输网络中, 由网页或者在线应用所产生的交互式流量会被优先安排, 其次是同步数据中心间的数据所产生的弹性流量, 这种流量通常只

需要在给定的时间内完成传输, 对数据整体的传输时延要求宽松^[3,4]。因此, 用于部署弹性流量的带宽具有不均衡性。而本文关注的就是带有截止时间的大数据传输任务。对于电路交换, 其端到端的建路机制, 在一条传输路径上所有链路预约的带宽必须保持严格一致, 因此在链路带宽具有不均衡性的网络中, 势必造成带宽资源浪费, 且还有可能导致请求无法在规定时间内完成传输。

为了解决上述问题, 本文提出在光电路交换网络的中间节点引入存储设备, 主要针对大数据传输问题中的静态业务。即: 请求的信息是提前已知的, 考虑在一个云服务供应商的数据中心间的传输网络中进行大数据传输, 在给定请求和网络资源信息情况下, 以尽可能多的请求在规定时间内完成传输为目标, 使得端到端的传输路径上允许运行不均匀的带宽速率, 从而有效提高带宽资源利用率。

收稿日期: 2017-10-11。

基金项目: 国家自然科学基金(61471238, 61433009)资助。

作者简介: 张涛(1993-), 男, 硕士研究生, 主要研究方向为传输网路由与资源调度。

1 问题描述

大数据传输的资源调度,需要解决以下3方面的问题:①时间窗的选择,每个请求都有各自的到达时间和截止时间,组成了相应的时间窗口,需要在时间窗口内选择进行数据传输的时间段;②路由计算,根据当前网络资源状况,选择一条合适的从源点到目的点的路径;③带宽分配,根据所选择的传输路径,为请求在相应链路上分配带宽。

给定一个网络拓扑,用图 $G=(V,E)$ 表示,其中 V 表示节点集合, E 表示链路集合。因为本文研究的大数据传输业务为静态业务模型,因此传输请求的信息是提前已知的。一个数据传输请求信息定义为:

$$r_i=(s_i,d_i,F_i,A_i,D_i) \quad (1)$$

表示在请求集 $R=\{r_i|1 \leq i \leq n\}$ 中的请求 r_i 需要从源点 s_i 传输 F_i 大小的数据至目的点 d_i ,请求的到达时间为 A_i ,请求需要在 A_i+D_i 之前传输完成。相比于传输时延,本文忽略传播时延。

本文把静态请求的资源调度问题目标定为容纳尽可能多的请求,即最大化请求整体传输成功率:

$$\text{最大化传输成功率} = \frac{\text{成功传输的请求数}}{\text{到达的请求总数}}$$

2 中间存储转发调度 (Intermediate Store and Transfer Schedule, ISTS) 算法

因为静态大数据传输的资源调度问题牵涉到时间窗的选择、路由选择和带宽分配,属于 NP-Complete 问题,无法找到一个算法在有限的多项式时间内寻得最优解。因此,本文提出一种启发式算法——ISTS 算法,来尝试为该问题求得可行解。

ISTS 算法以全局的角度处理多个静态业务请求,该算法主要分为三个阶段。第一阶段,对于给定的请求按照某种规则进行排序,本文采用以 F_i/D_i 为系数从大到小的排序策略,系数越大的请求说明更加紧急,优先安排此类请求的资源调度。第二阶段,对于排序好的请求依次在各自到达时刻的网络状况下进行路由搜索,这里使用 Dijkstra 最短路径算法来获得传输路径,若在到达时刻无法搜索到路由,则将请求编入等待队列。第三阶段,在上一步搜索到的路径上进行带宽的预约,这里采用贪心预约方法 (Greedy Reservation, GR)。根据 GR 策略,若请求能在规定时间内完成传输,则更新带宽资源;否则,将当前请求编入到等待队列。在对排序好的请求完成以上三个阶段后,再对等待队列里的请求执行第二、三阶段的处理。

为了方便对 ISTS 算法的描述,我们对符号做如下定义:

BW : 网络中链路的剩余带宽集合;

SR : 成功调度的请求集合;

QR : 经排序之后的有序请求集合;

WR : 需要等待的请求集合;

P_i : 请求 r_i 的传输路由;

T_i : 请求 r_i 完成传输的时刻;

T : 请求推迟的时间单位;

τ : 时间段;

S : 请求传输路径上中间节点存储情况;

reserve BW : 请求传输路径上每条链路带宽预约情况。

基于以上 ISTS 算法的符号定义,我们用表 1 中的伪代码来描述 ISTS 算法的流程。

表 1 用伪代码描述 ISTS 算法的流程

ISTS 算法
Require: $r_i=(s_i,d_i,F_i,A_i,D_i), BW$
Step 0. 初始化 $SR=\emptyset$
Step 1. 对 N 个请求 ($R=\{r_i 1 \leq i \leq N\}$) 按照文件大小与截止时间的比值从大到小排序得到 QR
Step 2. for 对于每个 QR 中的请求 r_i do
2.1 以 A_i 时刻网络的链路剩余带宽的倒数为代价值,生成代价矩阵
2.2 运用 Dijkstra 最短路径算法在代价矩阵上获得路由 P_i ; 否则加入等待队列 WR
2.3 根据 P_i , 利用 Greedy Reservation(BW, P_i) 方法进行带宽预约, 并求得请求完成传输时刻 T_i
2.4 若 $T_i < A_i + D_i$, 说明路由和带宽分配策略可行, 更新 BW , 将请求加入 SR , 否则加入等待队列
end for
Step3. for 对于每个 WR 中的请求 r_i do
3.1 以 $A_i + T_i$ 时刻网络的链路剩余带宽的倒数为代价值, 生成代价矩阵
3.2 运用 Dijkstra 最短路径算法在代价矩阵上获得传输路由 P_i ; 否则返回 3.1
3.3 根据 P_i , 利用 GreedyReservation(BW, P_i) 方法进行带宽预约, 并求得请求完成传输时刻 T_i
3.4 若 $T_i < A_i + D_i$, 说明路由和带宽分配策略可行, 更新 BW , 将请求加入 SR , 否则返回 3.1
3.5 如果在 $[A_i, A_i + D_i]$ 之间没有合适的带宽分配策略, 那么 r_i 就不会被部署
end for
return SR

GR 的目标是充分利用链路中的剩余带宽,假设在某一时段对传输路径上的链路带宽进行预约。首先,我们会预约满第一段链路的剩余带宽;其次,对于后面的链路,视前一段链路的预约情况和当前节点的存储数据的情况,综合考虑来预约合适的带宽大小。我们用表 2 中的伪代码来描述 GR 算法的流程。

表 2 用伪代码描述 GR 算法的流程

GR 算法
Require: $P=(l_1, l_2, l_3, \dots, l_n), S=(s_1, s_2, s_3, \dots, s_n), BW=(bw_{l_1}, \dots, bw_{l_n}), \tau$
Step 0. reserveBW = $(r_{l_1}, r_{l_2}, r_{l_3}, \dots, r_{l_n}), r_{l_i} = bw_{l_i}$
Step 1. for P_i 中除了 l_i 之外的所有链路 do
if $bw_{l_i} < r_{l_{i-1}} \parallel < r_{l_{i-1}} \times \tau + s_i > bw_{l_i} \times \tau$ then
$r_{l_i} = bw_{l_i}$
else
$r_{l_i} = (r_{l_{i-1}} \times \tau + s_i) / \tau$
end if
end for
return reserveBW

3 仿真实验及其结果分析

我们以传统电路交换 (Traditional Circuit Switching, TCS) 的带宽分配方法和 ISTS 算法来处理静态业务请求,并比较两种方法在网络性能上的差异。TCS 与 ISTS 的差异仅在于:TCS 要求传输路径上每条链路预约的带宽大小保持一致。除此之外,两种算法的排序策略和路由策略都是采用同一种方法。

仿真研究在 14 个节点的 NSFNET 网络上进行。假设网络上每个节点的存储资源都是充足的、网络中每条链路是双向的,且初始带宽为 10Gb/s。此外,为了使仿真更加贴近实际传输情况,需要考虑交互式流量的存在,交互式请求数量大约是大数据传输的 9 倍,约占据每条链路带宽的 5%~15%^[3]。因此,对于交互式请求,源点目的点从 14 个节点随机选取,请求到达满足泊松分布,需求的带宽满足在 1Gb/s 和 2Gb/s 均匀分布,请求持续时间满足平均值为 90s 的指数分布。而对于大数据传输请求,源点目的点均分布在网络节点,传输数据大小均匀分布在 200~800Gb 之间,截止时间对应于文件大小,在一定范围内产生。除特殊说明外,每个仿真图表都是 20 次仿真取其平均值的结果。在上述条件下,我们从传输成功率、网络链路利用率、请求平均传输时间和节点峰值存储使用情况 4 个方面来研究分析。

①传输成功率。图 1 比较了 TCS 和 ISTS 两种情

况下,传输成功率随请求个数变化的关系,可以看出,在请求较少的时候,网络链路带宽基本都能满足请求在给定时间内完成传输,而在请求数量增加的时候,链路带宽就相对匮乏,TCS 需要传输路径上每条链路所能预约的带宽一致,增加了传输时间;而 ISTS 因为引入了存储,允许把数据先传输至中间节点暂存,等到后面链路有空闲带宽再继续传输至终点,减少了请求传输的时间,提高了整体传输成功率。

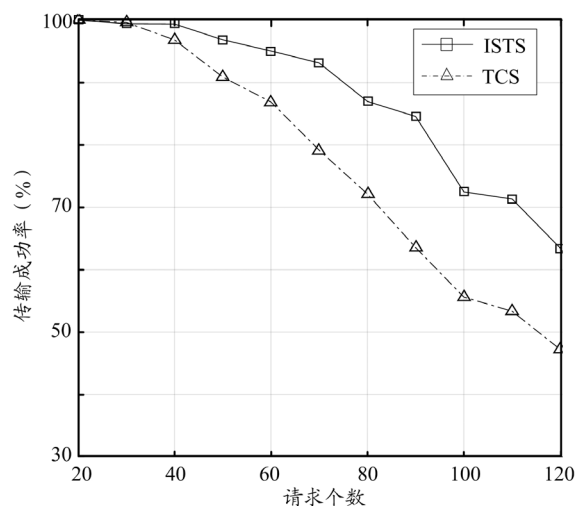


图 1 传输成功率 vs 请求数

②链路利用率。TCS 与 ISTS 在链路利用率方面的差异如图 2 所示,是在网络中有 100 个请求的情况下仿真的结果。本文对链路利用率的定义,是在整个传输时间跨度中所用的链路带宽与所有链路的总带宽的比值。由图 2 中可以得出,ISTs 的链路利用率可达 55%,而 TCS 大约在 50%。ISTs 之所以有更高的链路利用率,是因为其在传输路径上的不同链路可以预约

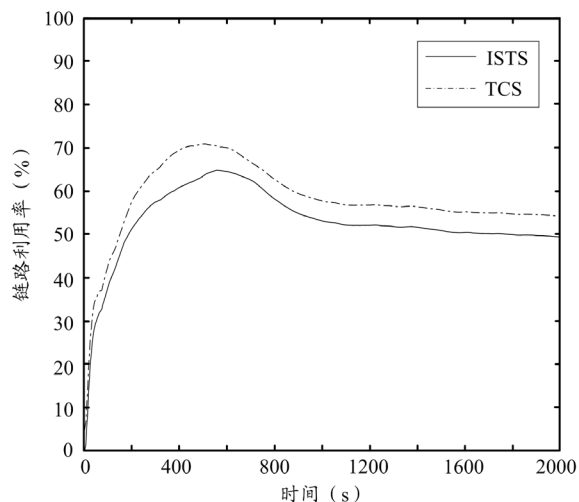


图 2 链路利用率 vs 仿真时间

不同速率的带宽,以更加充分地利用带宽资源。而TCS的带宽预约受限传输路径上链路的瓶颈带宽。

③请求平均传输时间。TCS与ISTS在请求平均传输时间上随请求数的差异如图3所示,此处针对成功传输请求的平均传输时间。从图3可以看出,ISTS在平均传输时间方面要略低于TCS,这也意味着ISTS中请求在截止时间之前传输完成的前提下还能够更快地完成传输。由于TCS的带宽预约方式受限传输路径上链路的带宽瓶颈,而ISTS又允许传输路径上运行不均匀的带宽速率,当传输路径上前一段链路预约的带宽大于后一段链路预约的带宽时,多余的数据就存储在中间节点。当后一段链路所能预约的带宽大于前一段链路时,就把暂存在中间节点的数据继续传输至终点。因此,ISTS能更快地将数据传输完,ISTS中请求所花的平均传输时间更短。

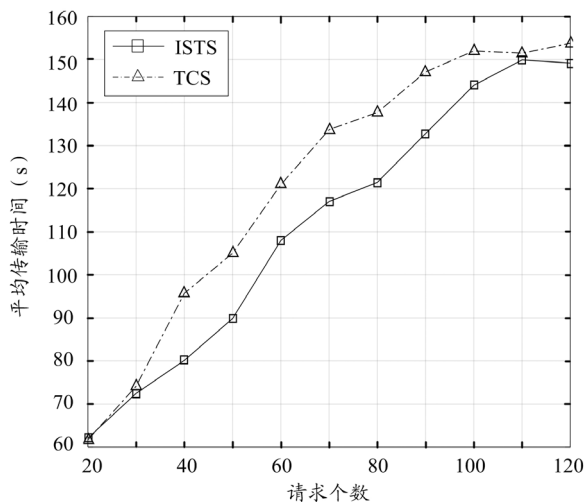


图3 请求平均传输时间 vs 请求数

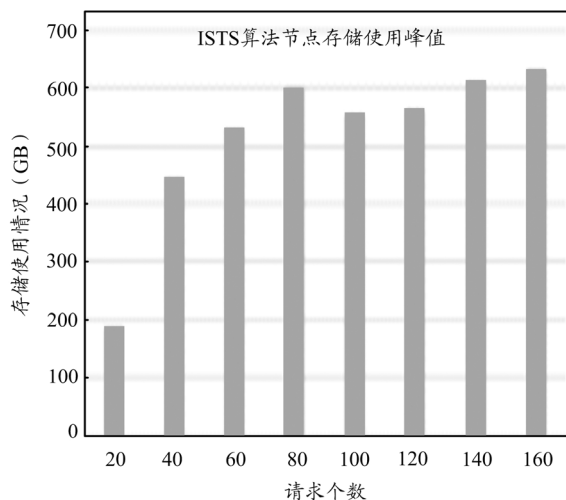


图4 ISTS节点峰值存储 vs 请求数

④节点峰值存储。图4是对ISTS算法在不同请求数的情况下,节点存储使用峰值的统计,从中可以看出,存储峰值用量大致维持在600Gb左右,这给在节点部署合适的存储容量提供了参考。由于本文是以地域分布的数据中心之间的传输网络为研究背景,因此我们可以利用数据中心丰富的存储资源来实现在中间节点的存储。随着云存储和软件定义存储技术的日渐成熟,存储设备将会以更低的成本提供高效的服务。

4 结束语

由于数据中心间传输网络中交互式流量的存在,对于带有截止时间的大数据传输任务而言,网络链路带宽资源具有不均衡性,会导致利用TCS方式进行传输造成带宽资源浪费的情况。本文提出在传输网络的中间节点引入存储设备来暂存时延容忍的数据,既能充分利用网络的带宽资源,又能提高请求的传输成功率。相比于传统的TCS方法,本文提出的ISTS算法有效提高了请求传输成功率和网络整体链路带宽利用率。除此之外,在平均传输时间方面,ISTS也略低于TCS。

事实上,存储在中间节点的数据可能会有安全方面的隐患,这可以通过数据加密和数据访问控制等方法得到有效解决,但具体方法不在本文研究范围内。后续工作,我们会将不同的排序策略和不同比例的带宽分配方式对网络性能的影响作为主要研究内容。

参考文献:

- [1] WU Y, ZHANG Z, WU C, et al. Orchestrating bulk data transfers across geo-distributed datacenters [J]. IEEE Transactions on Cloud Computing, 2017, 5(1): 112–125.
- [2] SU S, WANG Y, JIANG S, et al. Efficient algorithms for scheduling multiple bulk data transfers in inter-datacenter networks [J]. International Journal of Communication Systems, 2014, 27(12): 4144–4165.
- [3] NOORMOHAMMADPOUR M, RAGHAVENDRA C S, RAO S, et al. RCD: Rapid Close to Deadline Scheduling for datacenter networks [C] // World Automation Congress (WAC), July 31–August 4, 2016, Rio Grande. Piscataway: IEEE, 2016.
- [4] ZHANG H, CHEN K, BAI W, et al. Guaranteeing Deadlines for Inter-Data Center Transfers [J]. IEEE/ACM Transactions on Networking, 2017, 25(1): 579–595.