基于紧迫度分组调度算法

刘桐羽[1],陈翰雄[2],陈春玲[1]
(1.南京邮电大学 计算机学院、软件学院,江苏 南京 210003;
2.南京邮电大学 通信与信息工程学院,江苏 南京 210023)

摘 要:在分布式存储系统的数据中心,数据块通常都是以条状的方式分散存储在不同的服务器中。当客户请求数据时,多个服务器将会同时响应,将分散存储在服务器中的数据传输给客户。当传输的数据量超过交换机缓冲区大小就会在数据中心网络中发生TCP Incast现象。为改善数据中心网络发生TCP Incast吞吐量崩溃问题,提出基于紧迫度分组调度算法。通过在传输过程中将并发传输的数据流分组,并且按照紧迫度顺序对其传输,将并发传输数据进行分组调度传输,实现并发数据的串行传输。仿真实验表明,该算法大大降低了TCP Incast发生率,在服务器数量增加的情况下依然可以保证吞吐量没有出现大幅下降现象。基于紧迫度分组调度算法在大多数用例中可以避免Incast问题,并且提升网络传输性能。

关键词:数据中心网络;Incast;TCP;交错流;基于紧迫度分组调度算法
中图分类号:TP301.6 **文献标识码**:A **文章编号**:1673-629X(2015)10-0097-04
doi:10.3969/j.issn.1673-629X.2015.10.021

Group Scheduling Algorithm Based on Pressing Degree

LIU Tong-yu[1],CHEN Han-xiong[2],CHEN Chun-ling[1]
(1. School of Computer Science and Technology,School of Software,Nanjing University of Posts and Telecommunications,Nanjing 210003,China;
2. College of Telecommunications and Information Engineering,Nanjing University of Posts and Telecommunications,Nanjing 210023,China)

Abstract:In a data center of distributed storage system,the data blocks are usually stored in stripe as SRUs in these different servers. When multiple data senders simultaneously communicate with a single receiver,namely in many-to-one communication pattern,the burst data overloads the receiver's switch buffers,which leads to TCP throughput collapse,that's so-called TCP Incast problem. In order to improve the throughput collapse problem caused by TCP Incast in the data center networks,propose the algorithm Based on Pressing degree Group Scheduling (BPGS). Transfer the flows by grouping the flows and ordering them by the pressing degree. Simulation results show that the algorithm reduces the incidence of TCP Incast greatly,even when the number of servers increase,this algorithm can also ensure the throughout. BPGS can avoid the TCP Incast in most cases,what is more,it can also improve the network transmission performance.

Key words:data network center;Incast;TCP;staggered flow;BPGS

0 引 言

近年来,数据中心已经在悄悄地改变企业开展业务进程的方法。典型的,成千上万的数据中心主机通过使用高速的网络互联来进行通信。随着越来越多的应用部署,数据中心使用一个多层模型,比如Web服务器、应用服务器、数据库和存储服务器都一起工作来响应客户端的请求。因此,在数据中心中,总体应用性能很大程度上取决于该中心通信架构的效率。对于在数据中心建立通信架构来说,有两个比较好的选择。第一种选择是改变特定的硬件和通信协议,比如Infiniband,FibreChannel以及Myrinet;第二种选择是利用廉价的现成的产品,比如基于以太网的交换机路由器。第一种选择有能力扩展到数千个节点,但是它通常更加昂贵并且对于TCP/IP应用程序本身不兼容。

收稿日期:2014-11-04 **修回日期**:2015-02-05 **网络出版时间**:2015-08-26
基金项目:国家自然科学基金资助项目(61373137);华为科研基金项目(YB2013050012)
作者简介:刘桐羽(1990-),女,硕士研究生,研究方向为计算机通信网络;陈春玲,教授,研究方向为软件技术及其在通信中的应用、网络信息安全等。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20150826.1558.058.html

然而,第二个选择支持一个熟悉的管理基础设施,并且也不需要对应用程序、操作系统或者系统硬件进行修改,但是在节点的可扩展数上会比较少。基于成本以及兼容性,很多数据中心都选择第二种方案来作为它们的基线通信架构。

随着 TCP/IP 以及以太网技术的发展,由于 TCP/IP 的廉价以及公用性,被广泛应用于现代数据中心网络[1]。TCP 是一种成熟的技术,并且也经受住了时间的考验。然而 TCP 协议有时也会导致数据中心性能问题。比如 Amazon、Facebook、Google 以及 Microsoft 等网上服务者大量地投入到建立数据中心以此来支撑大规模数据的运算及存储[2]。这些数据中心应用,通常会使用分布/汇总的通信模式,一个请求数据会以分散数据块的形式分布存放在多个服务器中。因此,当客户端发出数据请求时,多个存放该数据的服务器就会同时做出响应,发送自己所存储的该部分数据块到同一个客户端。因此,客户端只有收到所有的分散数据块以后才会发送下一个数据请求,这种情况下很容易发生 Incast 问题[3]。

综上,导致在低延迟、高带宽和多对一的网络环境中经常发生 Incast 问题有以下几个原因[4]:首先,在日常操作系统中,最小重传时延(RTO min)通常被设定为 200 ms 或是 300 ms,而数据中心网络的往返时延(RTT)低于 250 μs[5]。这个最小重传时延通常是数据中心往返时延的 8 倍。其次,交换机通常采用缓冲区很小的交换机,并以此来降低成本。而这也会导致缓冲区溢出,进而导致数据包丢失[6]。最后,在分布/汇总的通信模式中,客户端必须要接收到所有来自服务器端的所有回应数据包以后才会发出下一个数据请求。因此,传输效率通常取决于一个或者几个数据包的传输过程[7]。

文中提出了基于紧迫度分组调度算法(Based on Pressing Group Scheduling algorithm,BPGS),通过控制待传输数据流的传输顺序以及传输时间来改善传输性能。在传输过程中引入该算法后,能够保证最需要的数据可优先占用信道进行传输,并能够有限避免发生 Incast 现象。通过 NS-2 模拟可以看到,BPGS 可避免 Incast 拥塞现象发生,并且吞吐量明显得到提升。

1 动 机

当多个发送者与一个接收者进行通信时,它们共同发送数据,这些数据足够超过接收者的以太网交换机缓冲能力,这时在高带宽低延迟的环境中就会发生 Incast 问题,而这个问题对于 TCP 吞吐量来说是灾难性的崩溃。问题来自于受限的以太网交换机缓冲区大小,TCP 恢复机制以及传输模式间的微妙影响,这也是数据中心应用的特点。当多个服务器同时发送数据流时,小的以太网缓冲区将会耗尽,这也导致了数据包的丢失以及发生一次或多次的 TCP 超时。这些超时会在网络上强加一个几百毫秒的延迟,而它们的往返时间通常被测量为几十到几百微秒。作为结果,在接收应用程序中得有效吞吐量比它的链路容量低几个数量级。这样的 Incast 模式可能出现在很多数据中心应用以及服务中。

在先前许多文献中都对 TCPIncast 问题有过研究[8]。这些工作中,有人提出将数据交错开来传输以避免数据并发传输。本节首先介绍交错流传输算法,并分析该算法的局限性。

1.1 交错流传输基本思想

从应用层角度来看,目标是避免 Incast 问题的发生。通过以上的分析可知,导致 Incast 问题发生的关键点在于交换机数据包的丢失造成 TCP 超时。因此,如果可以防止交换机上数据包的丢失,TCP 超时就不会发生,也就能避免 Incast 问题。同样的,当与同时进站的数据流相比,出站的链接带宽太窄导致了数据包的丢失。有限的交换机缓冲不能在一个相对长的时间里将所有的进站数据包保留下来,只能在转发它们之前将它们其中一部分丢弃。一些改进过的拥塞控制机制(比如快速重传机制)确实改善了性能,但是不可恢复的重传还是会发生,并且当并行发送的服务器数量增加时,性能变得更加难以让人接受。更重要的是,并发流可能导致发送方的拥塞窗口产生并发变化。这将会使问题变得更加严重,尤其是这些发送者都处在慢启动阶段。

由于 TCPIncast 现象是由大量数据并行传输导致交换机缓冲区溢出[9],进而引起数据包丢失,发生超时[10]。因此,交错流传输的主要思想为:在并行传输的数据流中加入时间片,使得原本并行传输的数据可交错开来实现串行传输,避免发生 TCP Incast 现象[11]。

1.2 交错流算法的不足

虽然通过插入时间片可以将并行传输的数据交错开来实现串行传输,但该算法存在一定不足。首先,交错流的思想为在传输数据中插入一个时间片,将数据严格交错开来传输。这样做虽然可以避免由于数据并行传输发生 Incast,但是损失了数据传输效率[12]。在数据中心网络,数据传输速度极快,简单将数据交错开来串行传输,就等同于放弃原本较快的传输速度。虽然可以避免 TCP Incast 现象发生,但也损失了传输速率[10]。其次,这样无序的串行传输数据,不能保证最被需要的数据首先得到信道进行传输[13]。将数据串行传输时,只是简单进行传输,不能保证数据传输顺序。

2　紧迫度分组调度算法

由于上述方法存在一些不足,因此文中提出基于紧迫度分组调度算法,通过该算法来避免 Incast 问题,提高传输性能。该算法主要在两个方面提升原算法。首先,提高数据传输效率,在保证避免 Incast 前提下,将多条数据流分为一组。其次,通过引入网络紧迫度对待传输数据进行调度。

2.1　算法思路

为了提高无线网络中数据分组的传输效率,出现了大量针对无线网络的数据分组调度算法,用以决定数据分组传输的顺序或时序,保证传输公平性或提高吞吐量。文中首先引入网络中传输紧迫度(Pressing Degree)概念,其定义如下:

$$P_i^k = D_i^k - d_i^k - L_k/r_i$$

其中,P_i^k 表示第 i 条数据流第 k 个数据分组的紧急度;D_i^k 表示第 i 条数据流第 k 个数据分组的延时范围;d_i^k 表示第 i 条数据流第 k 个分组积累延时;L_k 表示第 k 个数据分组大小;r_i 表示第 i 条数据流的发送速率。其中:$d_i^k = S(t) - A_i^k$,$S(t)$ 表示当前时间,A_i^k 表示第 i 条数据流第 k 个分组到达队列时间。通过对每一条数据流紧迫度进行计算。

接下来根据 P_i^k 值对待传输数据流按照从小到大的顺序进行排序,排序好以后对其进行分组。首先传送 P_i^k 值最小的一组,接下来传送 P_i^k 值次小的一组,依次类推,直到所有数据流传输完毕。在每一组里都有 l(l 值需要实验进行测试选取)条数据流。传送过程中,在每一组中间插入一固定时延 2 ms(2 ms 是一个 SRU 在 1 GigE 环境中的传输时间)。通过插入该时延可以保证每一组在传送时不会并行传输,避免在传输过程中发生 Incast。

通过该算法,按照紧迫度值从小到大的顺序对数据进行传输,这样就可以保证在串行传输的时候,将最紧急的数据首先传送到客户端。并且在每一组中间插入一个固定时延,可以保证组与组之间交错开来传输,避免并发传输引起 Incast 问题。

2.2　算法伪代码

输入:待传输的数据流;

输出:分组以后数据流。

1: for 所有传输的数据流

2:根据 $P_i^k = D_i^k - d_i^k - L_k/r_i$ 计算每一条数据流 P_i^k 值

3:对数据流按照 P_i^k 从小到大进行排序

4: end for

5:按照排好的顺序对数据流进行分组,l 条流一组

6:在组与组之间插入延时 2 ms

7:按组依次传输

3　仿真实验及结果分析

本节将重现 TCP Incast 现象,在此基础上使用文中提出的 BPGS 对该现象进行改善。通过观察吞吐量以及发生 Incast 概率来验证该算法的性能。

3.1　模拟环境

使用面向对象的网络仿真器—NS2 进行仿真实验。它可以用于仿真各种不同的 IP 网以及一些已经实现的网络传输协议,如 TCP、UDP[14]。它以单个数据包的粒度为单位来构建网络应用和协议模型。默认模拟配置包括一个集群存储系统,其中,模拟客户端和模拟服务器都连接到同一个交换机上。在这个环境中,数据块以条状的方式散布在许多服务器中。客户端向所有包含这个特定数据块 SRU 的服务器发送请求数据包来请求一个数据块;这个客户端只有接收到所有当前请求块的数据以后才会请求下一个数据块。也就是说,如果客户端从 n 个服务器中请求数据块,它只有在接收到总共 n * SRU 字节数据以后才会发送下一个数据块请求。这个简单的环境抽象了实时存储系统的许多细节,比如每个数据块的多个来自客户端的块请求,以及在一个单一的交换机上单用户发送一个通过共享子集的服务器的请求。

为了规范 Incast 现象的真实效果,在客户端通过改变参与数据传输的存储服务器数量来测量 TCP 的吞吐量。实验中,网络的拓扑结构为一个客户端以及多个服务器连接到一个交换机上。拓扑图如图 1 所示。将链路容量设置为 1 Gps,将服务器与客户端往返传输时延设置为 100 μs,将 SRU 大小设置为 256 kB,RTOmin 设置为 200 ms。

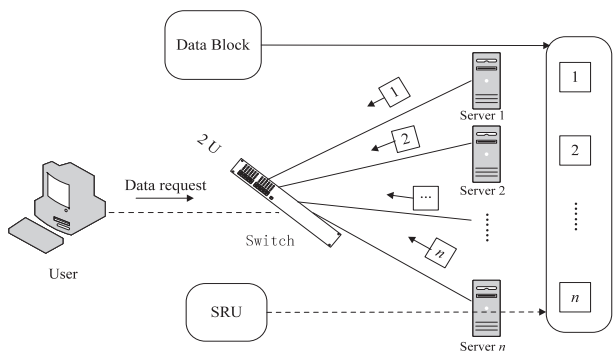

图 1　一个典型 Incast 的情形

3.2　Incast 现象重现

在这一部分中,通过改变参与传输服务器的数量来模拟 Incast 现象。可以很清楚地看到,当服务器数量超过 4 时,TCP 性能出现急剧下降,结果见图 2。

从图中可以看到,当参与数据传输服务器数量较

少时,网络吞吐量高,没有出现吞吐量大幅下降甚至崩溃现象。但是,随着服务器数量逐渐增加,吞吐量开始出现明显下滑。当服务器数量达到 5 时,吞吐量下降为原先的近 50%。当服务器数量进一步增加,吞吐量继续下降,最终出现吞吐量崩溃现象,这也是难以接受的。由此可见,当参与传输的服务器数量较少时,网络传输正常。但是随着服务器数量增加,多条数据流在同一时间并发传输造成交换机缓冲区溢出,发生延时,最终导致网络传输吞吐量崩溃。

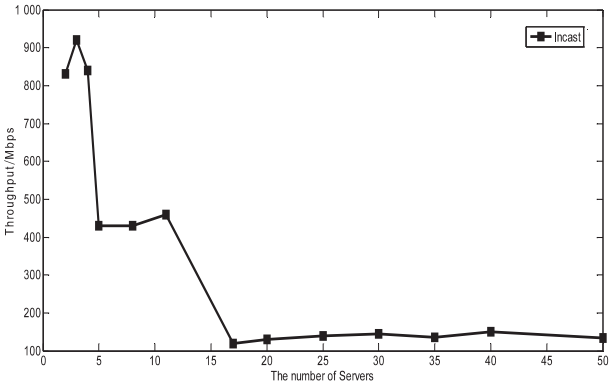

图 2 Incast 问题发生时吞吐量模拟图

3.3 改进后的传输性能

将文中提出的基于紧迫度分组调度算法应用到传输过程中。将待传输数据按照紧迫度进行分组,在每一个分组之间插入一个固定时延 2 ms。在图 2 中可以看到,默认情况下,在服务器数量为 2、3、4 时吞吐量比较高。因此按照紧迫度进行分组,分别对 4 条数据流分为一组,3 条数据流分为一组,2 条数据流分为一组进行实验测试,通过观察吞吐量来确定最合适的 l 值,进而确定将几条数据流分为一组。从图 3 可以看到,通过引入紧迫度分组调度算法可以有效提高吞吐量。不管是将 2 条、3 条还是 4 条数据流分为一组,都没有出现吞吐量大幅下降的现象。在这次实验中,随着服务器数量的增加,吞吐量没有出现大幅下降甚至崩溃现象。

另外,通过比较可以发现,当将 3 条数据流分为一组时,吞吐量性能达到最佳,因此 l 值取 3。在 l 取 3 以后,进一步实验,观察当服务器数量增加时,引入 BPGS 下是否还会发生 Incast 现象。通过实验可得,引入 BPGS 后,基本没有出现 Incast 现象。当服务器数量较少时,Incast 发生率接近 0。当服务器数量逐渐增加时,TCP Incast 发生率呈增长趋势,但是非常缓慢。相对的,在传统传输模式里,随着服务器数量的增加,当数量达到 7 时,Incast 发生率接近 100%。并且,随着服务器数量进一步增加,TCP Incast 发生率始终停留在 100%,没有下滑趋势。这说明,在传统传输过程中,当服务器数量超过 7 时,极有可能会发生 TCP Incast,而它也会导致吞吐量崩溃。在引入 BPGS 后,基本可以避免在网络传输过程中发生 TCP Incast,结果如图 4 所示。

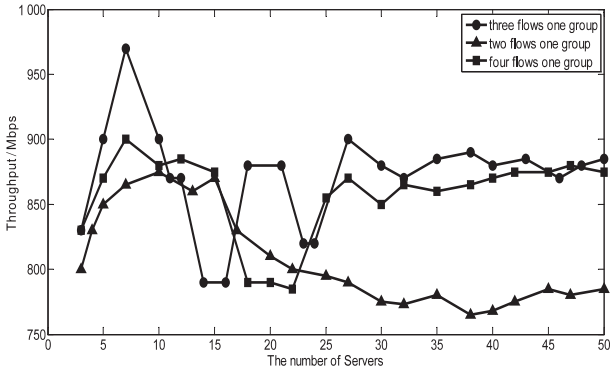

图 3 引入基于紧迫度分组调度算法后吞吐量模拟图

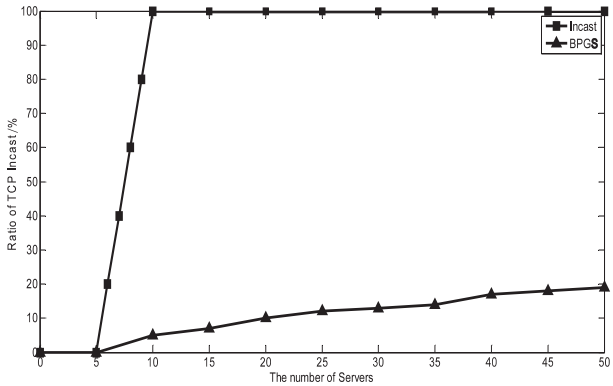

图 4 服务器数量与 Incast 发生率关系

4 结束语

针对在数据中心网络中经常发生的 TCP Incast 现象,文中提出 BPGS 算法。首先引入网络传输中紧迫度概念,接着按照紧迫度对待传输数据流进行分组。传输过程中,在每一组待传输数据流之间插入一个固定时延后再进行传输。通过实验可得,该方法能有效避免 Incast 现象发生,并且可以保证最紧迫的数据可以最先被传输到客户端,提高网络传输性能。

未来的工作是对组与组之间插入的延时进行分析。在这些实例中,首先是按照经验值指定好了延时值,但是随着分组算法的引入,或许需要对延时值进行修改。另外,没有将现实传输时一些其他因素考虑进去。因此,这两方面将是未来研究的方向。

参考文献:

[1] Yang Y, Abe H, Baba K, et al. Staggered flows: an application layer's way to avoid Incast problem[C]//Proc of IEEE Asia Pacific conference on cloud computing congress. [s. l.]: IEEE, 2012: 64-67.

[2] Hwang J, Yoo J, Choi N. IA-TCP: a rate based Incast-avoidance algorithm for TCP in data center networks[C]//Proc of

(下转第 106 页)

参考文献：

[1] Donoho D L. Compressed sensing[J]. IEEE Transactions on Information Theory,2006,52(5):1289-1306.

[2] Candès E J,Romberg J,Tao T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information[J]. IEEE Transactions on Information Theory, 2006,52(2):489-509.

[3] Candès E. Compressive sampling[C]//Proceedings of international congress of mathematicians. Zürich, Switzerland: European Mathematical Society Publishing House,2006:1433-1452.

[4] Duarte M,Davenport M,Takhar D,et al. Single-pixel imaging via compressive sampling[J]. IEEE Signal Processing Magazine,2008,25(2):83-91.

[5] Zheng J,Jacobs E. Video compressive sensing using spatial domain sparsity[J]. Optical Engineering,2009,48(8):1-10.

[6] Scholkopf B,Smola A,Muller K R. Nonlinear component analysis as a kernel eigenvalue problem[J]. Neural Computation, 1998,10(6):1299-1319.

[7] Varma M,Babu B R. More generality in efficient multiple kernel learning[C]//Proceeding of 26th annual international conference on machine learning. Montreal:ACM,2009.

[8] 黄海燕,柳桂国,顾幸生. 基于文化算法的 KPCA 特征提取方法[J]. 华东理工大学学报:自然科学版,2008,34(2):256-260.

[9] Reynolds R G. An introduction to cultural algorithms[C]//Proceedings of the third annual conference on evolutionary programming. San Diego,California:[s. n.],1994:131-139.

[10] Yuan X H,Yuan Y B. Application of culture algorithm to generation scheduling of hydrothermal systems[J]. Energy Conversion and Management,2006,47:2192-2201.

[11] Coello C A,Becerra R I. Evolutionary multi-objective optimization using a cultural algorithm[C]//Proc of 2003 IEEE swarm intelligence symposium. Indianapolis: IEEE,2003:6-13.

[12] Bin Peng. Knowledge and population swarms in cultural algorithms for dynamic environments[D]. Detroit:Wayne State University,2005.

[13] Candès E,Tao T. Near-optimal signal recovery from random projections:universal encoding strategies[J]. IEEE Transactions on Information Theory,2006,52(12):5406-5425.

[14] Mallat S,Zhang Z. Matching pursuit with time-frequency dictionaries[J]. IEEE Transactions on Signal Processing,1993, 41(12):3397-3415.

[15] Tropp J,Gilbert A. Signal recovery from random measurements via orthogonal matching pursuit[J]. IEEE Transactions on Information Theory,2007,53(12):4655-4666.

[16] Needell D,Tropp J A. CoSaMP:iterative signal recovery from incomplete and inaccurate samples[J]. Applied and Computational Harmonic Analysis,2008,26(3):301-321.

[17] Shtok J,Elad M. Analysis of the basis pursuit via the capacity sets[J]. Journal of Fourier Analysis and Applications,2008, 14(5-6):688-711.

[18] Reynolds R G,Zhu Shinin. Knowledge-based function optimization using fuzzy cultural algorithms with evolutionary programming[J]. IEEE Transactions on Systems, Man, and Cybernetics Part B:Cybernetics,2001,31(1):1-18.

[19] Liu Sheng,Gu Mingming. K-L transform in video compressed sensing[C]//Proceeding of the 32nd Chinese control conference. Xi'an,China:IEEE,2013:4528-4532.

(上接第 100 页)

ICC.[s. l.]:IEEE,2012:1292-1296.

[3] Kulkarni S,Agrawal P. A probabilistic approach to address TCP Incast in data center networks[C]//Proc of 31st international conference on distributed computing systems workshops. [s. l.]:IEEE,2011:26-33.

[4] 郝淑贤. 数据中心网络 TCP Incast 研究[D]. 桂林:广西师范大学,2013.

[5] Wang G,Ren Y,Dou K,et al. IDTCP:an effective approach to mitigating the TCP Incast problem in data center networks [J]. Information Systems Frontiers,2014,16(1):35-44.

[6] Zhang P,Wang H,Cheng S. Shrinking MTU to mitigate TCP Incast throughput collapse in data center networks[C]//Proc of third international conference on communications and mobile computing. Qingdao:IEEE,2011:126-129.

[7] Tahiliani R P,Tahiliani M P,Sekaran K C. TCP Variants for data center networks:a comparative study[C]//Proc of international symposium on cloud and services computing. Mangalore:IEEE,2012:57-62.

[8] Floyd S,Jacobson V. Random early detection gateways for congestion avoidance[J]. IEEE Transactions on Networking, 1993,1(4):397-413.

[9] Yang Y,Abe H,Baba K,et al. A scalable approach to avoid Incast problem from application layer[C]//Proc of IEEE 37th annual computer software and applications conference workshops. Japan:IEEE,2013:713-718.

[10] Zhang J,Ren F,Lin C. Modeling and understanding TCP Incast in data center networks[C]//Proc of INFOCOM. Shanghai:IEEE,2011:1377-1385.

[11] 王增福. 高速串行传输关键技术的研究与设计[D]. 西安:西安电子科技大学,2012.

[12] Mukhopadhyay A,Ranjan P. Nonlinear instabilities of D2TCP-II[C]//Proc of international conference on technology,informatics,management,engineering,and environment. [s. l.]: IEEE,2013:99-104.

[13] Zhang Y,Ansari N. On mitigating TCP Incast in data center networks[C]//Proc of INFOCOM. Shanghai:IEEE,2011:51-55.

[14] Kliazovich D,Bouvry P,Khan S U. Optical interconnects for future data center networks[M]. New York:Springer,2013.

基于紧迫度分组调度算法

作者： 刘桐羽， 陈翰雄， 陈春玲， LIU Tong-yu， CHEN Han-xiong， CHEN Chun-ling

作者单位： 刘桐羽,陈春玲,LIU Tong-yu,CHEN Chun-ling(南京邮电大学 计算机学院 软件学院,江苏 南京,210003)， 陈翰雄,CHEN Han-xiong(南京邮电大学 通信与信息工程学院,江苏 南京,210023)

刊名： 计算机技术与发展 ISTIC

英文刊名： Computer Technology and Development

年，卷(期)： 2015(10)