

基于 SCTP 多路径并行传输的性能评估

鄢 欢,高德云,宋 飞

(北京交通大学 电子信息工程学院,北京 100044)

摘 要: SCTP-CMT 是一种基于 SCTP 多路径并行传输的实现方法。阐述了在 SCTP-CMT 实现过程中不能尽快启用所有可用路径同时传输数据的不足,并在此基础上进行优化,同时提出了一种 IP 网络环境下数据传输性能的评估方法。使用该方法对优化后的 SCTP-CMT 和 TCP 的传输性能进行比较,分析了带宽、往返传输时延及接收缓存区大小等因素对传输性能的影响。测试结果显示在接收缓存区大小合适的情况下,使用优化后的 SCTP-CMT 在聚合带宽、提高传输吞吐量、增加可靠性等方面的具有明显优势。

关键词: 流控制传输协议;多路径并行传输;带宽;往返传播时延;接收缓存区

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2010)11-0029-04

Performance Evaluation of SCTP - Based Concurrent Multipath Transfer

YAN Huan, GAO De-yun, SONG Fei

(School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China)

Abstract: SCTP-CMT is an implementation method of the SCTP-based concurrent multipath transfer. Analyze the deficiencies of SCTP-CMT which can't use all the available paths as soon as possible to transmit data simultaneously, and on this basis optimize it. An evaluation procedure of transmission performance in IP network is proposed as well. The sending speed of optimized SCTP-CMT and traditional transmission control protocol (TCP) is compared according to our procedure. Also give the effect factor of transmission performance between SCTP-CMT and TCP by analyzing bandwidth, round-trip-time and the size of receive buffer. The results of testing show that using SCTP-CMT can obviously benefit bandwidth aggregation and increase the throughput of transmission when appropriate receive buffer size is applied.

Key words: stream control transmission protocol; concurrent multipath transfer; bandwidth; round-trip-time; receive buffer

0 引 言

随着互联网及相关技术的飞速发展,接入设备的性能不断提高。越来越多的终端存在多个网络接口。与此同时,实时多媒体业务对带宽的需求也越来越大。多路径并行传输具有充分利用网络资源、聚合带宽、增强安全性等诸多优势。支持多宿的传输层协议 SCTP^[1]的提出,为在传输层实现多路径并行传输提供了便利条件。然而传统的 SCTP 协议是选择同一关联中的一条路径作为数据传输的主要路径,其他作为备用路径。只有当重传数据或者路径故障时,才进行主备路径的切换,而不支持多条路径同时传输数据。

AI 等研究人员提出了名为 LS-SCTP^[2]的多路径

并行传输协议。其主要思想是将基于关联的拥塞控制扩展为基于每条路径的拥塞控制,并为每条路径上传输的数据块添加了序号。这种实现方式需要改变 SCTP 数据块格式,并修改发送端和接收端。

为了避免大量的修改 SCTP 协议,Ye 等研究人员提出了设计思想与 LS-SCTP 相似的 IPCC-SCTP 协议^[3],但其实现只需修改发送端,并且保留了 SCTP 数据块格式。Iyengar 等研究人员提出了一种多路径并行传输的实现方法 CMT_{sd}^[4]。它通过改变发送端的发送策略解决了不必要的重传及拥塞窗口不能正确增加的问题,并且通过研究发送重传数据的路径选择问题^[5],提出了 5 种选择策略,并分析了不同的策略对 CMT_{sd}传输性能的影响。

文献[6]介绍了一种实现 SCTP-CMT 的方法,文中在该方法的基础上进行优化。并且在实际的网络环境中,通过修改路径带宽、延时、接收缓存区大小等相关参数,测试不同情况下 SCTP-CMT 与传统 TCP 的吞吐量,并对性能进行比较和分析,从而得到最终的结

收稿日期:2010-03-08;修回日期:2010-06-27

基金项目:国家 863 高技术研究发展计划(2007AA01Z241)

作者简介:鄢 欢(1986-),女,新疆奎屯人,硕士,研究方向为计算机网络;高德云,副教授,博士生导师,研究方向为下一代网络体系结构。

论。

1 SCTP 协议简介

与传统 TCP 协议相比, SCTP 协议具有“四次握手”的关联建立机制、多流、多宿等特性^[7]。在 SCTP 协议的握手过程中, 服务器端收到客户端发来的 INIT 后, 需要创建一个状态 COOKIE 并将其放入 INIT ACK 中作为响应, 此时不用为该关联分配任何资源, 也不用保留任何状态。服务器端只有在收到 COOKIE ECHO 消息后才进入到关联建立状态, 这样可保护服务器免受 DoS 攻击^[8]。

多流^[9]特性是指在一个关联中多个流可在两个终端间并行传输。若某数据块丢失, 则该数据块所在流的其它数据将被接收方缓存直至丢失数据块重传到达后才能递交, 而其余流中数据可以继续递交, 不受影响。

文中主要是基于 SCTP 协议的多宿特性进行的改进。多宿^[10]指的是给定终端的一个关联中可以绑定多对 IP 地址, 并且这多个 IP 地址使用相同的端口号。多宿的好处是增加路径冗余, 提高网络级容错能力。在通信过程中, 确定一条路径作为主路径, 其他为备用路径, 一旦出现网络故障或者通路条件改变都可切换到其他路径继续路由。SCTP 使用心跳机制检测地址是否可达。

因此利用 SCTP 本身具有的多宿特性进行多路径并行传输扩展较有优势。而文中介绍的 SCTP-CMT 的实现方法具有聚合带宽, 增加传输可靠性等特点。该方法可充分利用多网卡主机间的多条路径同时传输数据, 将传统网络下作为备份的路径也用来传输, 这时网络可用带宽约为多条路径带宽之和, 提高了网络吞吐量。该方法还可增强安全性, 在数据传输过程中, 若数据块通过多个不同的路径发送, 攻击方想截获终端用户之间的所有数据, 进行解密和还原的难度较大。

2 基于 SCTP 多路径并行传输协议的优化

分析了一种基于 SCTP 的多路径并行传输实现方法^[6]。该方法只需要修改数据发送方即可实现数据的多路径并行传输。在关联建立的握手协商阶段, 通信双方相互通告接收窗口, 最大入流、出流数及各自拥有的所有网卡地址等信息。在数据块传输过程中, 发送方综合考虑路径因素从路径列表中选择最优路径发送数据。若数据发送成功则该路径的错误计数器清零, 否则该路径的错误计数器加 1, 重新选路发送超时数据。接收方只需对收到的正确数据块进行 SACK 确认。发送方收到 SACK 后进行相应的处理, 包括重传、

更新拥塞窗口等。当接收方对所有数据块都验证之后, 发送关联关闭消息, 在通信双方释放所有为关联维护的资源后, 关联终止。

在 SCTP-CMT 的握手阶段, 多宿主机相互通告双方所有的网卡信息, 建立关联。但是在数据块发送阶段, 只有传输请求建立关联控制块的路径状态为 SCTP_ACTIVE, 因此只在该条路径上传输数据, 并且相当一段时间内不使用其它路径。而实际上, 其它路径也可能是活跃的。只有当发送方发送心跳包, 并收到心跳包应答后才能将该条路径列入传输列表中, 在发送后来的数据块时使用。但是由于最初阶段只有一条路径用于传输, 这样不仅导致多条路径之间传输数据块数量相差很大, 还会影响整体传输吞吐量。

文中针对该问题进行优化。首先为每条传输路径增加变量 state_change。在关联建立阶段, 将该变量初始化为 FALSE; 在数据发送阶段, 向路径状态不为 SCTP_ACTIVE 的所有路径各发送一个数据块, 用来探测路径状态, 并设置变量 state_change 为 TRUE。

若收到对该数据块的应答, 则更改状态为 SCTP_ACTIVE; 若没有收到应答, 则状态不变, 而在这种情况下, 当为下个数据块选择路径时, 由于变量 state_change 值为 TRUE, 因此将放弃使用当前路径, 而是从传输列表中选择其他路径。

此优化方法可以尽早的使用所有可用路径同时传输数据, 从而提高传输吞吐量。图 1 是路径状态优化的伪代码。

```

At beginning of an association [Sender side behavior]:
1)  ∀ transport  $t_i$  initialize  $t_i$ .state_change = FALSE;
2)  for each address in the transport_addr_list do
    ∀ transport  $t_i$ ;
    if (state != SCTP_ACTIVE) then
        if (  $t_i$ .state_change = FALSE ) then
             $t_i$ .state_change = TRUE;
            set tag = 0; /*then can choose the min_tag as
                           the new transport to transmit next data*/
        else
            choose other transport in the transport_addr_list;

On receipt of a SACK [Sender side behavior]:
1)  ∀ transport  $t_i$ ;
2)  if the SACK carries tsn smaller than cum then
    set state as SCTP_ACTIVE;
3)  for each chunk in the control_chunk_list do
    if state is other statement without ACTIVE then
        choose the active path;
  
```

图 1 路径状态优化伪代码

3 测试结果

3.1 实验环境设置

利用搭建的网络环境对优化后的 SCTP-CMT 和传统 TCP 进行测试。具体的网络拓扑如图 2 所示。

整个试验环境由以下六个部分组成:1) SCTP-CMT 客户端:操作系统为 Fedora Core 5,内核版本为 2.6.15,运行编写的支持 SCTP-CMT 的文件下载客户端程序;2) SCTP-CMT 服务器端:操作系统为 Fedora Core 9,内核版本为 2.6.28,运行编写的支持 SCTP-CMT 的文件下载服务器端程序^[11];3)广域网模拟器(2台):操作系统为 REDHAT 9,内核版本为 2.4.20,运行网络仿真程序 NISTNET^[12];4)运行路由协议的三个路由器节点;5)TCP 客户端:操作系统为 Fedora Core 5,内核版本为 2.6.15,运行编写的支持 TCP 的文件下载客户端程序;6)TCP 服务器端:操作系统为 Fedora Core 9,内核版本为 2.6.25,对内核不进行修改,运行编写的支持 TCP 的文件下载服务器端程序。

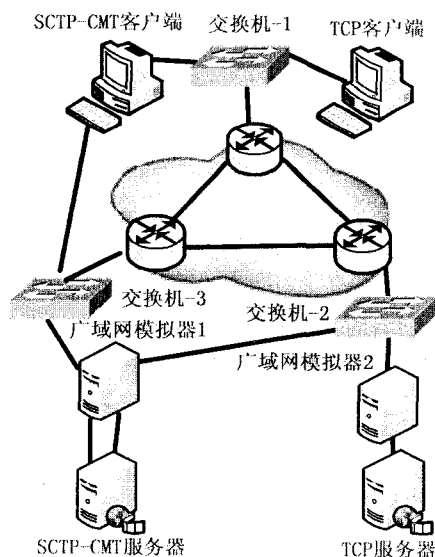


图 2 试验环境

为尽量全面模拟不同的场景,采用 NISTNET 作为广域网模拟器,设置客户端和服务端之间的带宽、延时等重要参数。

带宽(BW)的取值为[1Mb/s,2Mb/s,4Mb/s,6Mb/s,8Mb/s],该带宽是从客户端到服务器端和服务器端到客户端两个方向的取值。往返传输时延 RTT 的取值为[4ms,16ms,32ms,64ms,128ms]。服务器端的传输层发送缓存区为 107520 字节,客户端的传输层接收缓存区(RB)取值为[4kB,32kB,64kB,128kB,512kB],用来测试传输层接收缓存区大小对传输性能的影响。客户端和服务器的应用层发送缓存区及接收缓存区的大小都设为 8192 字节。

3.2 测试及性能分析

图 3 至图 6 是在不同试验场景下的基于 SCTP-CMT 和 TCP 的传输层协议,使用下载程序获取相同的文件时平均获取时间及吞吐量的对比。

图 3 是在设置每条路径的带宽为 4Mb/s,RTT 为

32ms,传输层接收缓存区大小为 128kB 字节的情况下,使用 SCTP-CMT 和 TCP 分别从服务器端获取相同文件所需要时间。由图 3 可得,与 TCP 相比,使用 SCTP-CMT 传输速率提高将近一倍。可见在相同网络环境下,SCTP-CMT 能更好的利用网络资源,提高传输速率。

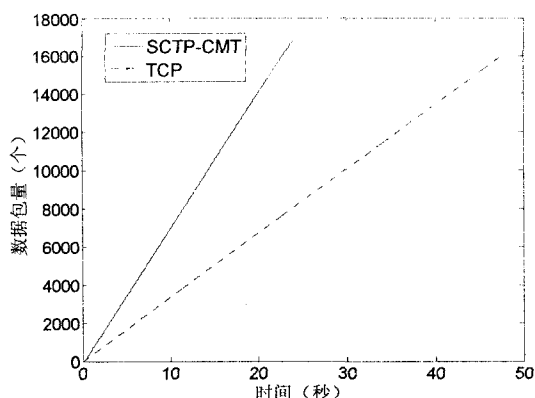


图 3 平均获取时间

图 4 是在固定 RTT 为 32ms,带宽分别为 1Mb/s 及 4Mb/s 的情况下获取同样的文件,以接收缓存区为变量的吞吐量变化曲线。从图中可以看出在带宽为 1Mb/s,接收缓存区为 4kB 时,SCTP-CMT 的吞吐量小于 TCP,但是随着接收缓存区的增加,SCTP-CMT 的吞吐量增加的速率明显大于 TCP。当接收缓存区的大小超过 64kB 时,SCTP-CMT 的吞吐量基本没有变化。这是因为当同时使用多条路径传输数据时,由于每条路径的参数不同,网络拥塞状况也不相同,很难保证数据块能够同时到达客户端,会导致严重的乱序问题。又因为接收缓存区很小,所以它很快会被乱序到达的数据块填满,出现接收缓存阻塞。在这种情况下,客户端既无法向上层递交,又无法再接收到达的数据块,只有等待缺失的数据块到达后,才能递交和继续接收数据。当接收缓存区增大到一定程度,从而能接收拥塞窗口允许发送的所有数据块时,接收缓存区不再限制吞吐量的增大,而带宽成为限制吞吐量的主要因素。

而对于 TCP 来说,接收缓存区的大小为 4kB 时,吞吐量大于 SCTP-CMT。但当接收缓存区大于等于 32kB 时,吞吐量变化不大。因为 TCP 使用单条路径传输,当接收缓存区很小时会限制吞吐量,而它增长到一定程度后,带宽将成为限制吞吐量增长的主要原因。

当带宽设置为 4Mb/s 时,SCTP-CMT 的吞吐量上升趋势与 1Mb/s 时类似,但该曲线是在接收缓存区为 64kB 时,吞吐量才会大于 TCP。原因是随着带宽的增加,服务器端能发送的数据块增多,而接收缓存区为 32kB 时已不能满足传输需要。接收缓存区大于等

于 128kB 时,可以看出,吞吐量约为带宽为 1Mb/s 的 4 倍。而对于此时 TCP 来说,接收缓存区大于等于 32kB 时,吞吐量不再变化,而其值约为 1Mb/s 时的 4 倍。

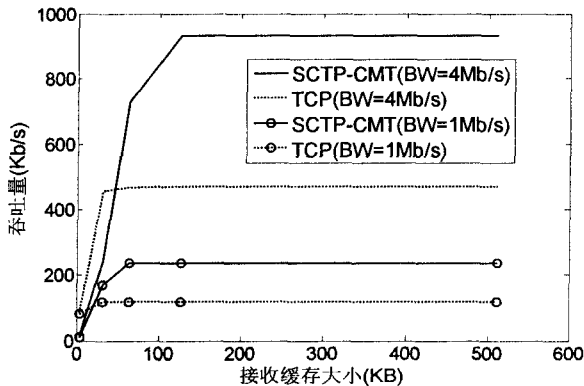


图4 以接收缓存区为变量的吞吐量

图5是在接收缓存区大小为 128kB, RTT 为 4ms 的情况下,使用 SCTP-CMT 和 TCP 分别获取同样的文件,以带宽为变量的吞吐量变化曲线。由图可得,在固定接收缓存区和 RTT 时,吞吐量随着带宽的增加而逐渐增大。当带宽小于等于 6Mb/s 时,使用 SCTP-CMT 及 TCP 时的吞吐量变化呈不同系数的线性增长;当带宽大于 6Mb/s 时,由于接收缓存区的限制, SCTP-CMT 的吞吐量增长速率下降,而对于 TCP 来说,接收缓存区的大小仍然能够满足缓存的需要,此时吞吐量变化只与带宽有关,因此其变化曲线仍为与之之前相同系数的线性增长。

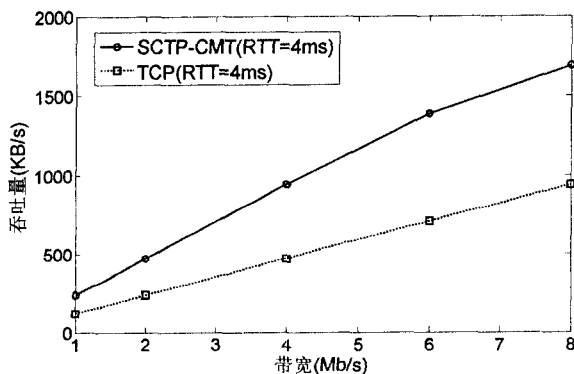


图5 以带宽为变量的吞吐量

图6是在固定带宽大小为 4Mb/s,接收缓存区大小分别为 32kB,64kB 的情况下,使用 SCTP-CMT 和 TCP 分别获取同样的文件,以 RTT 为变量的吞吐量变化曲线。由图可得,两条代表 SCTP-CMT 的实线都随着 RTT 的增大,吞吐量下降,这是因为传输延时的增大,降低了服务器端发送速率,从而降低了吞吐量。当 RTT 固定时,由于 SCTP-CMT 受接收缓存区的影响,因此吞吐量随接收缓存区的增大而增大。而对于 TCP 来说,吞吐量也随着 RTT 的增大而减小,增

大接收缓存区大小,同样也能增大吞吐量,但增加幅度在 RTT 较小时不明显。

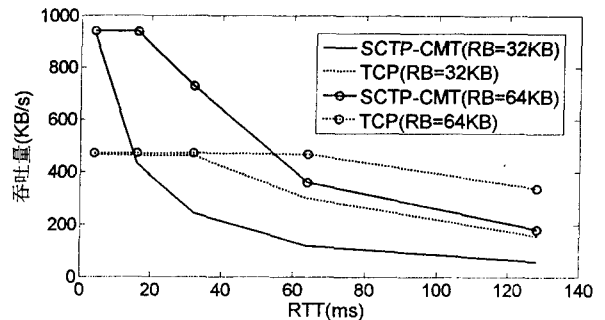


图6 以往返传输时延为变量的吞吐量

4 结束语

介绍了 SCTP 协议的相关特性,优化了一种 SCTP-CMT 的实现方法,并通过改变带宽、延时及接收缓存区大小的方式,在实际搭建的网络环境中将优化后的 SCTP-CMT 与传统 TCP 进行比较测试,最后就上述三个因素对 SCTP-CMT 及 TCP 的传输性能的影响作出了分析。从实验结果中可以看出,当接收缓存区较大时, SCTP-CMT 的传输性能明显优于传统 TCP,能够更充分利用网络资源。但是,由乱序导致的接收缓冲区阻塞问题会影响 SCTP-CMT 的传输吞吐量,该问题还有待进一步研究。

参考文献:

- [1] Stewart R, Ed. Stream Control Transmission Protocol[S]. RFC4960,2007.
- [2] Abd El Al A, Saadawi T, Lee M. LS-SCTP: a Bandwidth Aggregation Technique for Stream Control Transmission Protocol[J]. Computer Communications, 2004 (10): 1012 - 1024.
- [3] Ye G, Saadawi T N, Lee M. IPCC-SCTP: an enhancement to the standard SCTP to support multi-homing efficiently [C]//ICPCC2004. Phoenix,AZ,USA:[s. n.], 2004:523 - 530.
- [4] Iyengar J R, Shah K C, Amer P D, et al. Concurrent Multipath Transfer Using SCTP Multihoming [C] // SPECTS 2004. San Jose,USA:[s. n.],2004:265 - 273.
- [5] Iyengar J R, Amer P D, Stewart R. Retransmission policies for concurrent multipath transfer using SCTP multihoming [C]//Proceedings of the IEEE International Conference on Networks. Singapore:[s. n.], 2004:713 - 719.
- [6] Casetti C, Gaiotto W. Westwood sctp: load balancing over multipaths using bandwidth-aware source scheduling[C]// Proceedings of the IEEE International Conference on Vehicular Technology. Los Angeles:[s. n.], 2004:3028 - 3029.

(下转第 41 页)

4 结束语

Contourlet 变换相比于小波变换,具有更好的方向特性,能够更好地显示图像的几何特征,所以能为图像拼接提供更多的信息。但是由于在分解和重构过程中同时具有上、下采样过程,所以拼接过程不具有平移不变性。针对这一缺点,NSCT 方法应运而生,它利用 NSPFB 和 NSDFB 对图像进行多尺度、多方向分解,只具有上采样过程,具有平移不变性,得出的实验结果相比于非下采样小波方法,得到的结果图更加自然,细节信息更加清晰,过渡更加平滑。采用基于二维直方图的 B 样条曲面拟合的颜色校正方法在处理拼接图像明暗差异效果较好,相比于传统的颜色校正方法具有多个优点:不需要目标图;B 样条曲面具有良好的灵活性;二维直方图所描述的是不同量在整幅图像中所占的比例,跟空间位置无关,应用在全景图像的颜色校正时,此方法更加的简洁。

参考文献:

- [1] 蔡丽欢,廖英豪,郭东辉. 图像拼接方法及其关键技术研究[J]. 计算机技术与发展,2008,18(3):1-4.
- [2] Lucas B D, Kanade T. An iterative image registration technique with an application in stereo vision[C]//In: Seventh International Joint Conference on Artificial Intelligence (IJCAI-81). Vancouver: [s. n.], 1981: 674-679.

- [3] 钟 力,胡晓峰. 重叠图像拼接算法[J]. 中国图像图形学报,1998,3(5):367-370.
- [4] 肖 甫,吴慧中,肖 亮,等. 基于静态小波分解和能量函数优化的图像拼接[J]. 光子学报,2007,36(4):763-767.
- [5] Burt P J, Adelson E H. The Laplacian pyramid as a compact image code[J]. IEEE Trans. Common, 1983, 4(31): 532-540.
- [6] 郁 梅,易文娟,蒋刚毅. 基于 Contourlet 变换尺度间相关的图像去噪[J]. 光电工程,2006,33(6):73-77.
- [7] 李剑峰,高志荣,熊承义. 基于非下采样小波与方向上下文滤波的图像边缘检测[J]. 中南民族大学学报,2008,27(1):68-70.
- [8] 张 强,郭宝龙. 基于非采样 Contourlet 变换多传感器图像融合算法[J]. 自动化学报,2008,34(2):135-140.
- [9] 梁 栋,殷 兵,于 梅,等. 基于非抽样 Contourlet 变换的自适应阈值图像增强算法[J]. 电子学报,2008,36(3):527-530.
- [10] Zhou Jianping, Cunha A L, Do M N. Nonsubsampled contourlet transform: construction and application in enhancement [C]//Proc. of IEEE International Conference on Image Processing. Genoa, Italy: [s. n.], 2005: 469-472.
- [11] 张 强,郭宝龙. 一种基于非采样 Contourlet 变换红外图像与可见光图像融合算法[J]. 红外与毫米波学报,2007,26(6):476-480.
- [12] 戚世贵,戚素娟. 一种基于图像特征点的图像匹配算法[J]. 理论与方法,2008,27(1):3-4.

(上接第 32 页)

- [7] 白 刚,王重钢,隆克平,等. 流控制传输协议 SCTP 及其性能分析与应用[J]. 北京邮电大学学报,2001,24(4):63-66.
- [8] Stewart R R. 流控制传输协议 SCTP 参考指南(影印版)[M]. 北京:清华大学出版社,2003.
- [9] 夏 云,孙力娟,叶晓国,等. SCTP 协议分析与仿真研究[J]. 计算机技术与发展,2009,19(11):27-30.

- [10] 李 健,陶 洋. 基于 SCTP 多归属主机特性的多路径传输算法研究[J]. 重庆邮电学院学报,2005,17(4):1-4.
- [11] 冉春玉,张广军,屈 力,等. 基于 SCTP 的客户/服务器的设计与实现[J]. 计算机技术与发展,2007,17(6):127-130.
- [12] NIST NET Emulator[EB/OL]. 2005-07. <http://snad.ncsl.nist.gov/nistnet/>.

(上接第 36 页)

- [5] 王文义,梁青云,王若雨,等. Linux 集群系统并行应用程序监测技术的研究[J]. 郑州大学学报,2005(2):98-101.
- [6] 王若雨. Linux 集群可视化监测技术方法研究[J]. 中原工学院学报,2006(6):9-11.
- [7] 王勇超,张 景,王新卫,等. 基于 MPICH2 的高性能计算集群系统研究[J]. 计算机技术与发展,2008,18(9):101-104.
- [8] Talia D, Trunflo P. Toward a synergy between P2P and grids [J]. IEEE Internet Computing, 2003,7(4):94-95.
- [9] 潘志华,张 涛. Linux 预取算法分析与研究[J]. 计算机技术与发展,2009,19(12):93-96.

- [10] 魏 晓,胡金初. 基于 Linux 系统的分布式网络管理系统[J]. 计算机技术与发展,2007,17(6):60-63.
- [11] 蔡志明,卢传富,李立夏. 精通 Qt4 编程[M]. 北京:电子工业出版社,2008.
- [12] Devine, Boman K D, Erik G, et al. New Challenges in Dynamic Load Balancing[J]. Applied Numerical Mathematics, 2005,52(2):133-152.
- [13] Buyya R, Cortes T, Jin Hai. Single System Image[J]. The International Journal of High Performance Computing Applications, 2001,15(2):124-135.