

文章编号:1001-9081(2005)12-2916-03

Internet 自治系统间负载均衡机制及其性能分析

葛敬国¹, 马宏伟², 钱华林¹

(1. 中国科学院 计算机网络信息中心, 北京 100080;

2. 山东建筑工程学院 计算机科学与技术系, 山东 济南 250101)

(gejingguo@cnic.cn)

摘 要:提出了一种分布式动态调节的负载均衡机制,采用 Hash 表的负载均衡策略,具有保证流顺序、按链路带宽比例分配流量以及基于流量历史记录实时动态调整的特点。模拟实验评价了 Hash 表调整、调整时间间隔、判定阈值对算法性能的影响。

关键词:自治系统;负载均衡;Hashing;性能评价

中图分类号: TP393.02 **文献标识码:** A

Inter-AS load balancing mechanism and performance analysis

GE Jing-guo¹, MA Hong-wei², QIAN Hua-lin¹

(1. Computer Network Information Center, Chinese Academy of Sciences, Beijing 100080, China;

2. Department of Computer Science and Engineering, Shandong Institute of Architecture and Engineering, Jinan Shandong 250101, China)

Abstract: A distributed and dynamic load balancing mechanism was proposed. To preserve per-flow packet ordering and support unequal weighted distribution, hashing-based algorithm with dynamic adapting according to history records on traffic was used. Simulation experiments evaluate the effects of hash-tuning with varied time window and threshold.

Key words: autonomous system; load balancing; Hashing; performance evaluation

0 引言

为改善网络的可扩展性和可靠性,一般通过添加冗余的网络设备和并行链路方法提高网络的性能。大的企业或商务网络可以与多个 ISP(Internet Service Provider)之间存在多宿连接,自治系统(Autonomous System, AS)间一般存在多条链路,并且链路出口分布在不同地理位置的设备上。ISP 内 POP 点(Point Of Presence)之间存在多条长距离链路,形成 mesh 互连的骨干网络。负载均衡机制建立在现有网络结构之上,它提供了一种廉价有效的方法提高资源的利用效率和高可用性,减轻网络拥塞,提高服务响应速度。

一些路由协议在自治系统内部支持多路径路由,如 OPSF(Open Shortest Path First)^[1]、IS-IS(Intermediate System to Intermediate System)^[2]的等价多路径和 IGRP(Interior Gateway Routing Protocol)^[3]的非等价多路径,路由器转发引擎在转发包时必须依据负载均衡机制作出决策,选择其中的一条路径。当前自治系统之间的路由协议 BGP(Border Gateway Protocol)^[4]基于策略的配置来控制流量穿过自治系统到达相邻的自治系统,难以根据实时的网络流量和资源的状态分配流量。BGP 在流量负载均衡方面的能力十分有限,因为标准的 BGP 协议通常为目标网络选择唯一一条最优的路由,而且 BGP 很少交换关于网络拥塞和网络流量方面的信息,从而使得 BGP 协议不可能获得真正意义的动态负载均衡,只能通过复杂的策略配置实现基于子网粒度的流量的分担。

本文针对自治系统间外部链路的流量分配特点提出了一种分布式动态调节的负载均衡机制(Distributed and Dynamic Load Balancing Mechanism, DDLBM)。DDLBM 的特点在于:

1)以自治系统的外部链路的负载均衡为设计目标;2)实现按链路带宽比例分配负载,均衡策略保证流的顺序;3)每个包在进入自治系统的入口点独立地进行负载均衡决策,在穿过本自治系统的过程中不再进行负载均衡决策;4)不依赖于实时网络状态信息的交换,根据本地历史信息记录实现实时的流量动态调整。

1 相关工作

常用的负载均衡技术有两大类,一类包括随机数分配法、加权循环(round-robin)分配法,适合于输入的负载是离散的、相互独立的任务,任务之间没有特殊关系;另外一类有 Hash 分配法、状态记录分配法,适合于输入的部分任务之间存在相互依赖关系,算法需要保持这种依赖关系。绝大部分的 Internet 流量是基于 TCP 连接的, TCP 流包的顺序对 TCP 连接的性能影响很大,包的乱序将错误地触发拥塞控制机制导致 TCP 连接吞吐量的下降和大量不必要的重传而浪费网络带宽资源。网络负载均衡系统中流量的划分需要尽量避免 TCP 流内包的乱序,应采用第二类的负载均衡技术。由于状态记录分配法需要维护大量的流状态,存在扩展性问题,而基于 Hash 分配法仅维护有限的状态信息,实现简单,在网络负载均衡系统中大量应用。

文献[5]给出支持 OSPF 等价多路径的转发引擎实现负载均衡的三种算法:Modulo-N Hash、Hash-Threshold 和 Highest Random Weight(HRW)。文献[6]分析了上述算法的性能,主要讨论算法的复杂性和分裂因子(定义为由于等价路径变化时而发生路由变化的流量占总流量的比例)。Modulo-N Hash 和 Hash-Threshold 的算法复杂度相当, HRW 复杂度最

收稿日期:2005-06-09;修订日期:2005-09-05

作者简介:葛敬国(1973-),男,安徽肥东人,博士,主要研究方向:计算机网络体系结构; 马宏伟(1968-),男,山东济宁人,副教授,博士,主要研究方向:计算机网络体系结构; 钱华林(1940-),男,江苏人,研究员,博士生导师,主要研究方向:计算机网络体系结构。

大。假设路由器当前共有 N 个等价路径, 如果增加或删除一个等价路径, Modulo- N Hash 算法的分裂因子最大, 为 $(N-1)/N$, Hash-Threshold 算法的分裂因子在 $1/4$ 和 $1/2$ 之间, HRW 算法的分裂因子最小, 一般为 $1/N$ 。上述算法都需要进行 Hash 计算, 算法的均衡特性与 Hash 函数特性直接有关。文献[7]定义了一些评价负载均衡算法性能的参数, 并集中评价了各种用于网络负载均衡的 Hash 算法的性能。具有流量调节能力的基于 Hash 表的负载均衡算法能够获得相似的均衡效果, 且分裂因子最小, 具有按比例分配负载的能力, 计算复杂性也比 CRC 算法小, 但需要实时监控链路的负载和存储维护一张映射表。

2 域间负载均衡系统模型

Internet 由许多自治系统组成, 自治系统之间存在多条链路, 其使用决定于 BGP 协议, 如图 1。ISP 的主干网络由许多 POP 点通过长距离链路互相连接, 如图 2。自治系统之间或者 POP 点之间负载均衡算法类似, 本文统称为域间负载均衡。将域间的链路称为外部链路, 其关联的端口为外部端口。域间可能存在多条外部链路, 这些外部链路集合称为一个逻辑链路, 其相关联的端口集合称为逻辑端口。域间负载均衡机制的目标是将流量均匀地分布到逻辑链路所包含的所有外部链路上。

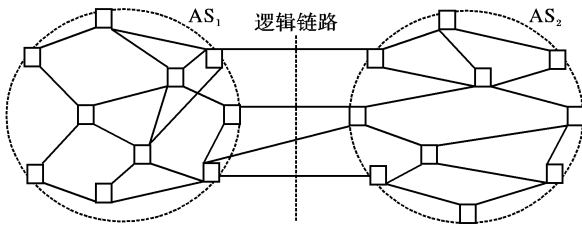


图1 AS之间的逻辑链路

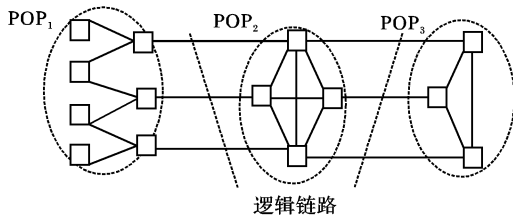


图2 POP点之间的逻辑链路

图3表示域间负载均衡模型。针对于逻辑链路 L^i , 域间负载均衡系统可以抽象为域所有外部入口端口通过域内部网络与域的逻辑端口 P^i 所包含的外部端口 $\rho_j^i (j=1, 2, \dots, k)$ 相连, 并将来自这些外部入口端口、目标指向逻辑端口 P^i 的流量均匀分布到 k 个外部端口 $\rho_j^i (j=1, 2, \dots, k)$ 上。流量自域外到达域后, 在外部入口端口, 首先确定退出域的目标逻辑链路, 然后根据域间负载均衡算法从目标虚拟链路中选择一条目标外部出口链路。逻辑链路以及目标外部出口链路的选择都在域的外部入口端口处执行一次, 在域内部不再执行。



图3 域负载均衡模型

3 一种分布式动态调节的负载均衡机制

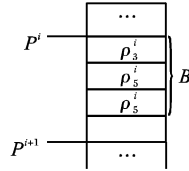
本节给出了一种基于 Hash 表的域间负载均衡机制——

分布式动态调节的负载均衡机制 DDLBM。参考图3描述的负载均衡模型, DDLBM 的基本思想是在每个外部入口端口处, 将进入网络并指向逻辑端口 P^i 的数据流按照某种方法分布到所有属于逻辑链路 L^i 的外部链路上, 如果所有的外部入口端口都能够将流量按各外部链路带宽比例均匀分布在逻辑链路 L^i 包含的外部链路, 则这些流量在外部链路的叠加也是按各外部链路带宽的比例分布的。

DDLBM 由两部分组成: 基于 Hash 计算的流聚类及目标外部链路的选择, Hash 表调整过程。

3.1 基于 Hash 计算的流聚类及目标外部链路的选择

在核心骨干网络, 流的数量庞大, 记录、查找和管理每流信息非常复杂, 可扩展性差, 而且负载均衡需要将一些流在外部输出链路间迁移, 由于流的粒度太细, 操作效率低。因此, 为保证节点域的可扩展性和操作效率, 需要将数量庞大的细流聚集为合适粒度的宏流, 以此宏流为基本单位执行负载分配算法。采用 Hash 聚类方法可以将大量细流映射到有限的 Hash 空间上, 映射到相同 Hash 空间的所有数据流聚为一个宏流, 以宏流作为负载分配的基本单位。



在域的每个外部入口端口, 对应于每个逻辑端口 P^i 有一张 Hash 表。图4是外部入口端口 Hash 表示意图, 每个逻辑端口 P^i 对应的 Hash 表长为 B , Hash 表

图4 外部入口端口 Hash 表 内容为属于逻辑端口 P^i 的某一个外部端口号, 即记录 Hash 值与外部出口端口 $\rho_j^i \in P^i (j=1, \dots, k)$ 的映射关系。

当一个包到达域的某个外部入口端口, 首先从包中抽取识别流特征的五元组, 确定其目标逻辑端口号 (这里假定其逻辑端口号为 i), 然后, 以该五元组为参数调用 Hash 函数计算 Hash 值, 以 Hash 值为索引查找与逻辑端口 P^i 相应的 Hash 表, 得到相应的外部输出端口 ρ_j^i , 完成包的目标外部端口选择。

3.2 Hash 表调整方法

由于不同流的流量特征相差很大, 基于 Hash 的流分布没有保证流量在逻辑链路内均匀分布。负载均衡的关键是如何确定 Hash 表内容, 即如何将这宏流指派到相应的外部出口链路上, 使得各外部出口链路的利用率尽可能的接近。根据逻辑链路表描述的域所有的外部链路的状态以及它们与逻辑链路的从属关系, 计算各外部端口 ρ_j^i 对应的配载因子 α_j^i :

$$\alpha_j^i = \frac{C(\varepsilon_j^i)}{\sum_{\varepsilon_m^i \in L^i} C(\varepsilon_m^i)} \quad (1)$$

其中 $C(\varepsilon_j^i)$ 表示外部链路 ε_j^i 的带宽。配载因子 α_j^i 表示按绝对负载均衡需求, 所有进入该域指向逻辑链路 L^i 流量的 α_j^i 应分配给外部链路 ε_j^i 。

在各个外部入口端口使得流量分布满足相应外部输出链路的配载因子, 这些来自所有外部入口端口、满足配载因子的流量在外部输出端口叠加, 最终保证节点域流量分布满足外部输出链路配载因子。

确定 Hash 表内容的步骤如下 (以在某个外部入口端口对逻辑链路 L^i 执行负载均衡操作为例):

1) Hash 表的初始化。在外部输入端口处, 假设流的强度相同且 Hash 函数将指向逻辑链路 L^i 的流均匀散列在 Hash 表空间, 则流量将在 Hash 空间均匀分布。此时, Hash 表初始化为有 m_j^i 个表项内容被指派为 ρ_j^i , 其中:

$$m_j^i = \lfloor \alpha_j^i \times B \rfloor \quad (2)$$

2) 流量累计。在外部入口端口处,采用时间窗口方法累计分配给逻辑链路 L^i 内 k 个外部输出链路的流量 $\varphi_j^i (1 \leq j \leq k)$, 变量 Φ^i 记录当前时间窗口逻辑链路 L^i 所有外部输出链路的流量总和。时间窗口大小 τ 固定,以此为基本时间片累计流量。考虑流量分配历史,对不同时间槽历史记录给予不同的权值 $\omega(h) (1 \leq h \leq H)$, $\sum \omega(h) = 1$,记录越新,权值越大。变量 $\psi_j^i(h) (1 \leq j \leq k, 1 \leq h \leq H)$ 记录前 h 个时间窗口内的实际负载分配因子(分配给外部出口端口 ρ_j^i 的流量占指向逻辑链路 L^i 流量的比例,即 $\psi_j^i(h) = \varphi_j^i(h)/\Phi^i(h)$)。各外部入口端口实时地独立计算分配给逻辑链路 L^i 内 k 个外部输出端口 $\rho_j^i (1 \leq j \leq k)$ 的实际负载分配因子 $\psi_j^i (1 \leq j \leq k)$:

$$\psi_j^i = \sum_{h=1}^H \omega(h) \times \psi_j^i(h), 1 \leq j \leq k \quad (3)$$

3) 基于流量记录的负载均衡判定方法。在外部出口端口,对应于逻辑链路 L^i 内所有的外部链路,计算分配给各外部输出端口的实际负载因子与理论配载因子的差异 $\delta_j^i = \psi_j^i - \alpha_j^i (1 \leq j \leq k)$ 。设预定的阈值 $\Delta > 0$,如果, $\exists m$, 且 $\delta_m^i > \Delta$,则判定逻辑链路负载不平衡,触发 Hash 表调整算法。

4) Hash 表调整算法。在链路利用率最大和最小的两个外部链路间进行负载均衡,使得这两个外部链路的链路利用率趋向一致。调整其对应的外部端口在 Hash 表中所占的表项数目,实现映射到这些表项的宏流从链路利用率高的外部链路迁移到链路利用率低的外部链路,完成一次负载均衡过程。调整表项的数量按如下方法计算:

设在时间窗口宽度 τ 内到达外部链路 ε_j^i 的流量 Φ_j^i , 链路 ε_j^i 的带宽为 $C(\varepsilon_j^i)$, ζ_j^i 表示外部链路 ε_j^i 的链路利用率,即到达速率与链路带宽的比值,如公式(4):

$$\zeta_j^i = \frac{\Phi_j^i / \tau}{C(\varepsilon_j^i)} \quad (4)$$

设 $\zeta_s^i = \max_{j=1, \dots, H}(\zeta_j^i)$, $\zeta_t^i = \min_{j=1, \dots, H}(\zeta_j^i)$ 分别为逻辑链路 L^i 中当前最大和最小的链路利用率,设 N_s, N_t 为当前 ρ_s^i, ρ_t^i 在 Hash 表中出现的次数。

基于 ρ_s^i, ρ_t^i 分配的流量历史,假设将内容为 ρ_s^i 的 z 个 Hash 表表项内容改写为 ρ_t^i 时,链路 ε_s^i 与 ε_t^i 的链路利用率相等,有:

$$\frac{(\phi_s^i - \phi_s^i \times \frac{z}{N_s}) / \tau}{C(\varepsilon_s^i)} = \frac{(\phi_t^i + \phi_t^i \times \frac{z}{N_t}) / \tau}{C(\varepsilon_t^i)} \quad (5)$$

根据链路利用率定义,化简方程式(5),有:

$$\zeta_s^i (1 - \frac{z}{N_s}) = \zeta_t^i (1 + \frac{z}{N_t}) \quad (6)$$

解方程式(6),得:

$$z = \frac{(\zeta_s^i - \zeta_t^i) \times N_s \times N_t}{(\zeta_s^i \times N_t + \zeta_t^i \times N_s)} \quad (7)$$

相对互补调整过程是将内容为 ρ_s^i 的 z 个 Hash 表表项内容改写为 ρ_t^i ,使得链路 ε_s^i 与 ε_t^i 的链路利用率相等。

3.3 算法性能

DDLBM 采用分布式的负载分配方法,在节点域的各个外部入口端口进行流量的合理分配以保证全局的负载平衡,适合应用于域间大范围高速核心域的负载均衡。外部入口端口之间没有相互的信息交换和全局的协同,各外部端口依据本节点记录的局部的流量信息动态调整 Hash 表的内容,独立进行流量的负载均衡操作。

此外,算法稳定性、适应性、避免大量的流量迁移震荡,也

是域间负载均衡算法考虑的重要因素。影响 DDLBM 性能的因素有:

1) 选择适当的 Hash 函数,提高流的均匀散列能力。Hash 函数的优劣直接影响到负载平衡的质量和算法的稳定性。

2) 确定适当的 Hash 表空间大小 B, B 越大,流汇集粒度越细,流量调节越准确,但存储、管理和流量调整计算的开销增加。 B 太小,流汇集粒度增大,难以精细地调节流量,负载均衡不易实现,甚至导致流量在不同链路路上的震荡,此外,调整一个 Hash 表项可能影响多个流的路径。

3) 确定适当的时间窗口宽度,时间窗口小,则计算的负载因子实时性强,反映端口瞬时负载分配情况,但可能造成流量调节频繁,使系统不稳定。

4) 确定适当的历史记录权值 $\omega(h) (1 \leq h \leq H)$,以反映不同阶段历史记录的重要程度,即负载均衡的判断依据是侧重于平均流量,还是侧重于突发流量。

5) 确定适当的阈值 Δ ,需要在减少流的切换频度(切换可能带来乱序)与提高流量调整精度之间折中。

4 域间负载均衡模拟实验及评价

用 NS-2^[8] 搭建了一个域间负载均衡模拟实验平台,图 5 是域间负载均衡实验的拓扑图。考察一个逻辑链路 L^i 内 k 个外部链路 $\varepsilon_j^i (j = 1, \dots, k)$ 的负载均衡性能,可以通过观察各链路的利用率 ζ_j^i 值是否接近来评价逻辑链路的负载均衡性能。 ζ 值越接近,并且随时间、负载等变化而保持一致趋势,反映负载均衡机制效果越明显、性能越稳定。反之,反映负载均衡机制的效果不明显,或者性能不稳定。用三个参数 ζ_{\max}^i 、 $\Delta_{\max} \zeta^i$ 、 σ_{ζ}^i 来描述同一逻辑链路的各外部链路利用率 ζ 之间的关系。

ζ_{\max}^i 表示在模拟实验的所有时刻, k 个外部链路利用率的最大值。当 $\zeta_{\max}^i > 1$ 时,逻辑链路将出现过载现象,即某个时刻、某个链路的到达速率大于其带宽。

$\Delta_{\max} \zeta^i$ 表示在模拟实验的所有时刻 $\{1, 2, \dots, T\}$, 任意两个外部链路利用率 ζ 差的最大值,反映 ζ 最大的差异。

$$\Delta_{\max} \zeta^i = \max_{t=1, \dots, T} (\zeta_{u,t}^i - \zeta_{v,t}^i), \varepsilon_u^i, \varepsilon_v^i \in L^i, u \neq v \quad (8)$$

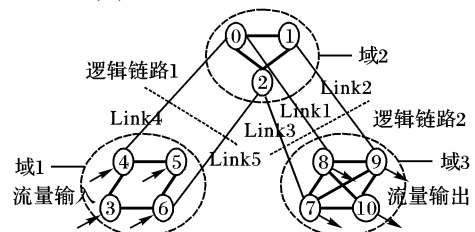


图5 域间负载均衡实验拓扑结构

$\bar{\sigma}_{\zeta}^i$ 表示在模拟实验的所有时刻 k 个外部链路利用率的方差的均值,综合反映各个外部链路利用率的相互偏离大小。

$$\bar{\sigma}_{\zeta}^i = \sqrt{\frac{\sum_{t=1}^T Var_t^i}{T}}, Var_t^i = \frac{\sum_{j=1}^k (\zeta_{j,t}^i - \bar{\zeta}_t^i)^2}{k-1} \quad (9)$$

实验考察在各种实验情景下,DDLBM 的域间负载均衡能力,分析 Hash 表调整与否、调整时间间隔、调整阈值、流量分布等因素对负载均衡性能的影响。表 1 集中给出了各实验的结果。

(下转第 2927 页)

集成的性能要优于最大值集成。

表 1 不同分类器及按不同规则集成的实验结果

分类方法	识别性能	
	训练集	测试集
归一化小波	94.61	92.11
笔划密度	95.45	93.88
黑像素百分比	91.73	88.08
最大值集成	96.67	95.69
平均值集成	98.88	98.06
加权平均集成	98.89	98.08

5 结语

模式识别中的多分类器集成方法得到了越来越多的关注和应用。多分类器集成的关键是根据应用的背景,寻找一种合适的组合准则。

本文基于 HMM 模型,在构建基于归一化小波特征、笔划密度特征以及黑像素百分比特征的独立分类器基础上,通过多分类器集成技术,将三种分类器进行有效集成。本文所采

用的多种集成方法,都保持了比较高的识别率。集成后的系统显著改善了系统整体的识别能力。

参考文献:

- [1] 朱小燕,史一凡,马少平. 手写体字符识别研究[J]. 模式识别与人工智能, 2000, 13(2): 174-180.
- [2] RABINER LR. A Tutorial on Hidden Markov Models and Select Applications in Speech Recognition[J]. IEEE, 1989, 77(2): 257-286.
- [3] NISHIMURA H, KOBAYASHI M, MARUYAMA M, *et al.* Off-line character recognition using HMM by multiple directional feature extraction and voting with bagging algorithm[A]. Fifth International Conference on Document Analysis and Recognition[C]. Bangalore, 1999. 49-52.
- [4] WANG WW, BRAKENSIEK A, RIGOLL G. Combining HMM-based two-pass classifiers for off-line word recognition[A]. Proceedings 16th International Conference on Pattern Recognition[C]. Quebec, 2002. 151-154.
- [5] FAN GL, XIA XG. Wavelet-Based Texture Analysis and Synthesis Using Hidden Markov Models[J]. IEEE Transaction on Circuits and System, 2003, 50(1): 106-119.

(上接第 2918 页)

模拟实验(1)、(2)比较了当输入流量是指数分布的 on/off 流时,在没有 Hash 调整的情况下, $\zeta_{\max} = 1.41 > 1$, 表明部分时刻已经出现外部链路过载现象,节点域的负载均衡性能很差,进行 Hash 调整时则没有出现链路过载现象。模拟实验(3)、(4)比较了当输入流量是常数速率流时,进行 Hash 表调整的负载均衡机制效果明显,在同一时刻,不同外部链路的 ζ 值更接近。

表 1 实验结果

实验序号	实验情景	ζ_{\max}	$\Delta_{\max}\zeta$	$\overline{\sigma_{\zeta}^2}$
实验(1)	指数流量分布,没有 Hash 表调整	1.41	0.85	0.19
实验(2)	指数流量分布,进行 Hash 表调整	0.91	0.18	0.049
实验(3)	常数速率流量分布,没有 Hash 表调整	0.575	0.23	0.05
实验(5)	指数流量分布,调整间隔 $\tau = 0.1s$	0.91	0.18	0.04919
实验(6)	指数流量分布,调整间隔 $\tau = 0.5s$	0.96	0.415	0.079
实验(7)	指数流量分布,调整间隔 $\tau = 1.0s$	1.22	0.46	0.11
实验(8)	指数流量分布,判定阈值 $\Delta = 0.01$	0.72	0.24	0.052
实验(9)	指数流量分布,判定阈值 $\Delta = 0.05$	0.80	0.45	0.088
实验(10)	指数流量分布,判定阈值 $\Delta = 0.1$	0.80	0.67	0.18

可以看出,执行 Hash 表调整的负载均衡机制可以有效地改善网络性能。由于 Internet 流量的突发性,DDLBM 中 Hash 表调整是必要的。

模拟实验(5)、(6)、(7)比较了 Hash 表调整时间间隔 τ 不同对负载均衡性能的影响。随着调整时间间隔的增加,同一时刻链路的 ζ 差异增加,表明负载均衡效果减弱。调整时间间隔太小,负载均衡依赖负载的即时速率进行负载均衡操作,当流量具有较高的突发特征时,即时速度变化幅度大、变化频率快可能使得 Hash 表调整频繁,容易造成流在逻辑链路内不同外部链路之间频繁切换,最终导致流内包的乱序。调整时间间隔太长,负载均衡依赖负载的平均速率进行负载均衡操作,负载均衡依赖的流量速率估计过于乐观,可能导致链路过载。确定 Hash 表调整时间间隔需要在负载均衡效果与调整的频率之间折中。

模拟实验(8)、(9)、(10)比较了负载均衡算法中判定是否需要 Hash 表调整的阈值发生变化时对负载均衡性能的影响。随着判定阈值 Δ 的增大,同一时刻链路的 ζ 差异增加,表明负载均衡效果减弱。判定阈值 Δ 的大小影响 Hash 表调整的频次,进而影响负载均衡算法的效果。可以根据负载均衡的需要,灵活地设置阈值 Δ 。

5 结语

给出了一个通用的域间负载均衡模型,并以此为基础提出一种基于 Hash 表的分布式动态调整的域间负载均衡机制 DDLBM,算法详细描述了 Hash 表调整的方法,计算需要调整的链路以及调整的力度。算法在域的各个入口处独立地进行负载的分配,依赖局部的流量历史信息调整分配到各外部链路的流量,没有消息的传递和全局的协同,适合应用于大型的高速网络。模拟实验揭示了采用 Hash 表调整的算法负载均衡效果明显优于没有 Hash 表调整的算法。

参考文献:

- [1] MOY J. OSPF Version 2, RFC 2328[S]. 1998.
- [2] ISO 10589. Intermediate System to Intermediate System Intra-Domain Routing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473) (Also republished as RFC 1142)[S].
- [3] HEDRICK CL. An introduction to igmp[R]. Technical report, Rutgers University, August 1991.
- [4] REKHTER Y, LI T. A Border Gateway Protocol 4(BGP-4), RFC 1771[S]. 1995.
- [5] THALER D, HOPPS C. Multipath Issues in Unicast and Multicast, RFC 2991[S]. 2000.
- [6] HOPPS C. Analysis of an Equal-Cost Multi-Path Algorithm, RFC 2992[S]. 2000.
- [7] CAO ZR, WANG Z, ZEGURA E. Performance of Hashing-Based Schemes for Internet Load Balancing[A]. Proceedings of IEEE Infocom[C]. 2000. 332-341.
- [8] UCB/LBNL/VINT Network Simulator-NS (version 2)[EB/OL]. <http://www-mash.cs.berkeley.edu/ns>, 2005-05.