

# 基于消息的加权负载均衡算法

王德民<sup>1</sup>, 何立东<sup>2</sup>, 刘菲菲<sup>2</sup>, 苏男<sup>2</sup>, 刘昕<sup>1</sup>

(1. 吉林大学网络中心, 长春 130012; 2. 吉林大学计算机科学与技术学院, 长春 130012)

**摘要:**在研究 EAP 协议与 Diameter 协议的基础上,为解决分布式认证系统节点间会话一致性的问题,以及尽可能减少服务节点数量改变时产生的会话迁移数量,提出了一种基于消息的加权负载均衡算法(MOLB)。该算法使用散列法、虚拟节点和红黑树相结合的技术,实现了客户端请求在服务节点间的合理分布。Diameter 网络环境中的实验结果表明:与其他常用的负载均衡算法相比,本文算法具有较小的负载均衡度和会话破坏度以及较低的会话破坏分布度。

**关键词:**计算机应用; Diameter 协议; 负载均衡; 虚拟节点

**中图分类号:** TP393 **文献标志码:** A **文章编号:** 1671-5497(2012)01-0140-05

## Message-oriented load balancing algorithm

WANG De-min<sup>1</sup>, HE Li-dong<sup>2</sup>, LIU Fei-fei<sup>2</sup>, SU Nan<sup>2</sup>, LIU Xin<sup>1</sup>

(1. Network Center, Jilin University, Changchun 130012, China; 2. College of Computer Science and Technology, Jilin University, Changchun 130012, China)

**Abstract:** Based on the study of extensible authentication protocol (EAP) and Diameter protocol, a message-oriented load balancing (MOLB) algorithm was proposed. This algorithm can maintain the consistency of session between nodes in distributed authentication system and reduce the number of migration sessions when service nodes are varied. In this algorithm hash method, virtual nodes and red-black tree are combined to achieve a rational distribution of requests between service nodes. In the Diameter network environment, the experiment results show that compared with other commonly load balancing algorithms, MOLB algorithm has smaller load balancing degree, smaller session destruction degree and lower session destruction distribution degree.

**Key words:** computer application; Diameter protocol; load balancing; virtual node

随着接入网络的用户数量不断增加,网络规模不断扩大,网络中待处理的数据量与系统处理能力之间的矛盾越来越突出。部署分布式系统,用其并行处理大规模数据量已成为当下主流的解决方案。其中的一个难点便是负载分配算法的设计,目前已有多种算法实现负载均衡。

Paul Vixie 在 BIND 中使用了 Marshall Rose 提出的轮询调度算法(Round-Robin Scheduling)来解决 DNS 集群的负载均衡问题<sup>[1-2]</sup>; Kargerl 等<sup>[3]</sup>提出了一致性哈希算法,解决了分布式缓存系统中的热点问题; Godfrey 等<sup>[4]</sup>研究了结构化的 peer-to-peer 网络,并基于 DHT 提出了一种负

**收稿日期:** 2010-11-04.

**基金项目:** 国家发展改革委员会项目(CNGI2008-023); 国家科技支撑计划项目(2008BAH37B05023); 吉林省科技发展计划项目(20080702).

**作者简介:** 王德民(1958-),男,教授. 研究方向: 分布式与网络. E-mail: wdm@jlu.edu.cn

**通信作者:** 刘昕(1976-),女,工程师. 研究方向: 智能交通网络. E-mail: lx@jlu.edu.cn

载均衡算法。

在 Diameter 网络中,EAP-TLS 协议规定认证的双方需要经过多轮信息交互才能完成身份认证、证书交换和密钥协商<sup>[5]</sup>。这些特点使得上述负载均衡算法在应用到基于 Diameter 协议<sup>[6]</sup>的分布式认证系统时存在如下问题:不能保证会话具有一致性;当网络结构发生变化时,会产生大量会话迁移的情况,从而导致某些服务节点存在过载或轻载问题,使得网络的可扩展性不强。

针对 EAP-TLS 协议具有多轮交互的特性,本文提出了一种基于消息的负载均衡(Message-oriented load balance,MOLB)算法。该算法不仅使得请求能够均匀地分布在服务器端,较好地实现负载均衡,而且解决了 Diameter 网络中的会话保持问题,减少了网络结构变化时节点增减所带来的会话迁移量。

## 1 MOLB 算法

### 1.1 基于消息的负载均衡模型

基于消息的负载均衡模型如图 1 所示,汇聚的消息发送至 Diameter proxy 节点,proxy 节点利用 MOLB 算法从  $n$  个服务节点中选定消息的目的地址,将消息发送至该地址。Diameter 服务节点将到达的请求缓存至本地的未决消息队列中,由守护进程处理具体的应用请求。

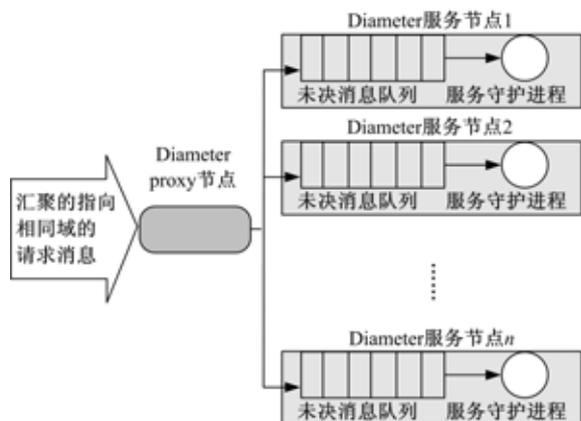


图 1 基于消息的负载均衡模型

Fig. 1 Message-oriented load balance module

Diameter proxy 作为系统的负载均衡器,必须能够将消息按照配置的权值,均衡分配至各个服务节点,同时,尽可能地将属于同一个会话的消息发送至同一个 Diameter 服务节点进行处理。Diameter proxy 节点是基于消息的负载均衡模型中最重要的部件,其性能的优劣直接决定整个系

统性能的好坏。

### 1.2 MOLB 算法数据结构

为了既满足维持会话一致性的要求又兼顾效率,MOLB 算法使用了散列函数,虚拟节点和红黑树结构。

#### 1.2.1 散列函数

散列法<sup>[7]</sup>是通过计算出单向不可逆的散列值来为消息选择一个目的地址,因此可以利用该算法保持会话的一致性。

如图 2(a) 所示,MOLB 算法使用 MD5 算法<sup>[8]</sup>将节点映射到一个  $0 \sim 2^{32}$  的整数范围内,当消息到达负载分配器时,负载分配器查找比当前消息的 session-id 散列值大的第一个散列值,该散列值代表的节点即为命中的服务节点。

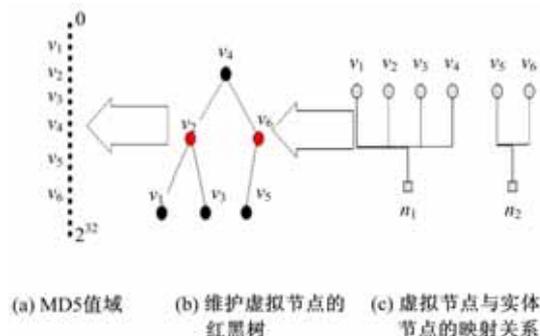


图 2 MOLB 算法中服务节点映射示意图

Fig. 2 Node mapping of MOLB algorithm

#### 1.2.2 虚拟节点

由于散列值的分布具有不规则性,并且实体服务节点的数量通常很少,因此节点在散列值域上常常会分布不均。为了使节点的散列值在散列域中获得渐进意义上的分布均衡,可采用将一个实体节点映射成多个虚拟节点的方法,增加节点所映射的散列值的数量。负载均衡器负责维护它们之间的映射。

如图 2(c) 所示,  $n_1$  和  $n_2$  表示两个实体节点,  $v_1$  至  $v_4$ ,  $v_5$  和  $v_6$  分别代表它们所对应的虚拟节点。

#### 1.2.3 红黑树

为了提高查询效率,MOLB 算法采用红黑树<sup>[7]</sup>来维护虚拟节点之间的一种全序关系(见图 2(b))。

红黑树以虚拟节点所对应的散列值作为关键字组织起来,提供增加虚拟节点、删除虚拟节点和查找给定关键字后继节点(即查找第一个比所给关键字大的虚拟节点)等操作。红黑树能够在插

入节点或删除节点的同时自动调整树的高度,保证了在最坏的情况下,基本的动态集合操作时间为  $O(\lg n)$ 。

### 1.3 MOLB 算法流程

MOLB 算法的主要流程可分为三部分:初始化映射关系,更新节点信息和查询消息目的地址。

#### 1.3.1 初始化映射关系

初始化过程获取各个节点的信息,建立实体节点、虚拟节点和散列域之间的映射关系,其过程如下:

Step1 初始化存放操纵 MOLB 算法数据结构的句柄,即初始化指向红黑树的指针和设置所使用的散列函数。

Step2 获取一条实体节点信息,包括实体节点标识和对应的虚拟节点数目。

Step3 设置节点所对应的实体节点信息:添加节点标识,设置对应的虚拟节点数目并将节点状态设置为 NODE\_INIT。

Step4 将实体节点状态设置为 NODE\_IN,利用实体节点标识与虚拟节点编号求得虚拟节点在散列域中的值。

Step5 查询存储虚拟节点信息的红黑树中是否存在与所求得的散列值相等的值。若存在,跳转至 Step7,否则继续。

Step6 根据虚拟节点信息创建一个红黑树节点,并将其插入至红黑树中,虚拟节点总计数加 1,修复红黑树性质。

Step7 若存在未加入红黑树的虚拟节点,跳转至 Step3,否则继续。

Step8 若存在未处理的候选节点信息,跳转至 Step2,否则继续。

Step9 初始化过程结束。

#### 1.3.2 更新节点信息

当分布式系统改变拓扑结构时,即增加或减少实体服务节点时,执行更新节点信息算法,更新映射关系,算法流程如图 3 所示。

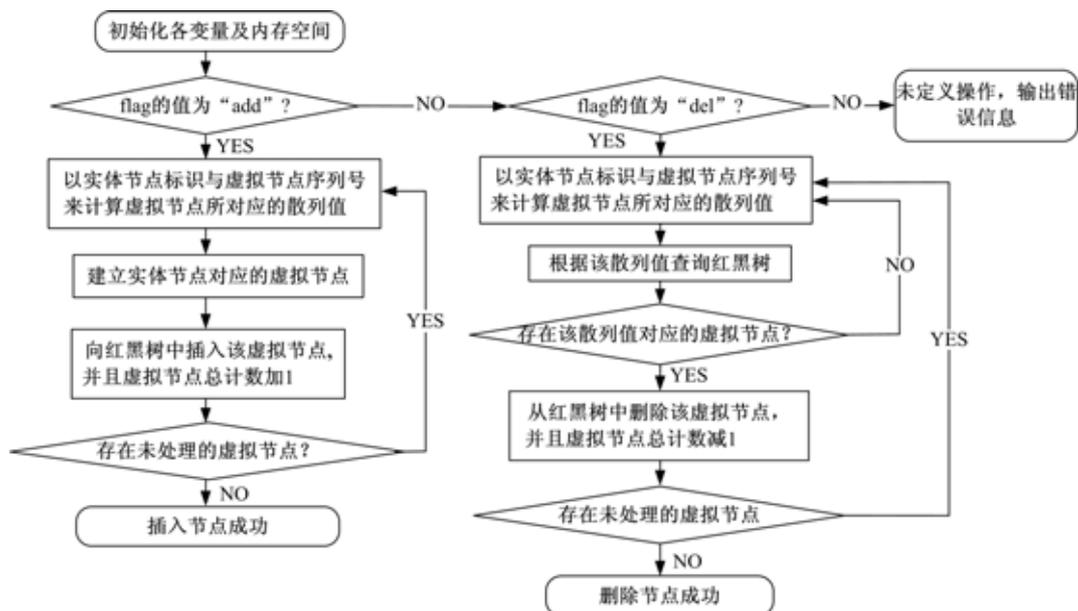


图 3 更新节点算法流程图

Fig. 3 Flow chart of updating node

MOLB 算法通过操纵红黑树来维护虚拟节点与散列函数域之间的映射关系。在添加节点阶段,红黑树以虚拟节点对应的散列值为关键字创建虚拟节点;在删除节点阶段,红黑树也是根据散列值来查询是否有符合删除条件的虚拟节点。

#### 1.3.3 查询消息目的地址

当 Diameter proxy 节点收到一条服务请求消息时,执行查询消息目的地址过程。步骤为:

Step1 初始化各变量及内存空间。

Step2 从 Diameter 请求消息中 Diameter-EAP-Request 提取 Session-id avp 中的值。

Step3 使用散列函数计算 Session-id 的散列值。

Step4 在红黑树中查询该散列值的后继。若该后继不存在,输出红黑树中最小节点对应的实体节点标识;若后继存在,则输出后继所对应的节

点标识。

## 2 算法评价参数

在 Diameter 网络中,负载分配器需要将属于同一个会话的消息分配到同一个服务节点进行处理,即会话保持。在面向会话的处理调度中,文献[9-10]中提出的性能评价虽然能够在微观的角度较准确地表现出系统的负载情况,但无法准确地衡量会话的一致性,也无法表现出节点数目变化时带来的负载分布的变化,所以不能评价出 MOLB 算法的真实性能。为了进行真实、客观地评价,现对上述文献的负载均衡评价标准进行修改和扩充,完善基于消息的负载均衡算法的评价标准。该标准主要由负载均衡度、会话破坏度和会话破坏分布度组成。

**定义 1** 负载均衡度:设  $m$  为实体服务节点的总个数,  $req$  表示客户端在  $\Delta t$  时间内的请求次数,  $req_i$  表示在  $\Delta t$  时间内请求命中第  $i$  个服务节点的个数,  $n$  表示与  $m$  个实体服务节点对应的虚拟节点的总个数,  $n_i$  表示与第  $i$  个实体服务节点对应的虚拟节点的个数。

当  $n_1 = n_2 = \dots = n_i = \dots = n_m$  时,  $r_i$  的自协方差为

$$CV[req_i] = \frac{D(req_i)}{(E(req_i))^2} = \frac{E(req_i^2) - (E(req_i))^2}{(E(req_i))^2} \quad (1)$$

将  $CV[req_i]$  归一至  $[0, 1]$  区间,得到负载均衡度  $mLBM$ :

$$mLBM = \frac{CV(req_i)}{CV(req_i) + 1} = \frac{1 - \frac{(E(req_i))^2}{E(req_i^2)}}{1 + \frac{(E(req_i))^2}{E(req_i^2)}} \quad (2)$$

使用  $req_i, req_i^2$  在  $\Delta t$  内的观察值表示它们的数学期望,其观察值计算如下:

$$E(r_i) = \sum_{i=1}^m req_i / m \quad (3)$$

$$E(r_i^2) = \sum_{i=1}^m req_i^2 / m \quad (4)$$

将式(3)(4)代入式(2),得

$$mLBM = 1 - \frac{(\sum_{i=1}^m req_i)^2}{m \sum_{i=1}^m req_i^2} \quad (5)$$

根据式(5)可知:①  $0 \leq mLBM \leq 1$ ; ②负载

均衡时的  $mLBM \rightarrow 0$ 。

会话负载均衡度以消息为单位计算得到会话的均衡度,体现了负载的宏观均衡度。在基于消息的处理系统中,好的宏观均衡度表示负载能够均匀地分配到各个节点。

**定义 2** 会话破坏度:实体服务节点在一定时间内增加或减少时,所造成的会话未命中的数量与会话总数的比值即称为会话破坏度。

会话破坏度  $desSession$  的公式如下:

$$desSession = \frac{reqUpdate}{reqSum} \quad (6)$$

式中:  $reqUpdate$  表示增加或减少服务节点时发生的会话迁移数量;  $reqSum$  表示更新节点前会话总量。

使用会话破坏度评价负载分配过程中会话的未命中行为,能够直接反映出会话迁移的频度,会话破坏度越小,会话迁移量就越少。

**定义 3** 会话破坏分布度:会话破坏分布度  $desDistribute$  的定义如下:

$$desDistribute = \sum_{i=1}^m \frac{(req_i + sessIn_i - sessOut_i)^2}{req_i} \quad (7)$$

式中:  $sessIn_i$  与  $sessOut_i$  分别表示实体节点数目变化后,在节点  $i$  上迁入会话与迁出会话的数量。

会话破坏分布度可以反映出更新实体节点数量后,未命中的请求在实体节点间的分布情况,会话破坏分布度越小,说明未命中的请求消息分布越均匀。

## 3 实验结果与分析

实验的硬件环境:CPU 为 Pentium(R) Dual-Core CPU E5500 2.80 GHz;内存为 2 G;操作系统为 Ubuntu10.04。

实验的软件环境:选择实现了 RFC3588 的开源软件 FreeDiameter 作为实验软件。FreeDiameter 的扩展模块对应着 Diameter 协议中的应用部分,具有处理应用请求、定义高级的寻路方法和管理对等节点等功能。本文编写了 FreeDiameter 的扩展模块,实现了 MOLB 算法。

表 1 为在  $m = 4$  和  $m = 3$  时,轮询法(RRS)、散列法(HM)和 MOLB 算法所获得的各项评价参数。其中,  $S_i$  表示  $m = 4$  时请求命中实体节点  $i$  的数量值,  $S'_i$  表示  $m = 3$  时请求命中实体节点  $i$  的数量值。

表1 会话破坏实验数据  
Table 1 Data of session damage test

参数	RRS	HM	MOLB
$S_1$	250000	201008	240073
$S_2$	250000	270951	247507
$S_3$	250000	138171	257873
$S_4$	250000	389870	254547
$S'_1$	333334	240668	321191
$S'_2$	333333	279614	332994
$S'_3$	333333	479718	345815
$sessIn_1$	250000	39660	81118
$sessIn_2$	249999	8663	85487
$sessIn_3$	249999	341547	87942
$sessOut_1$	166666	0	0
$sessOut_2$	166666	0	0
$sessOut_3$	166666	0	0
$mLBM$	$2 \times 10^{-12}$	0.089	0.026
$DesSession$	0.749	0.390	0.256
$DesDistribute$	$8.333 \times 10^4$	$8.528 \times 10^5$	$8.692 \times 10^4$

实验结果表明:RRS算法和MOLB算法的负载均衡度都较小,两种算法都可以将消息均衡地分配到各个实体服务节点,但RRS算法会话破坏度明显大于MOLB算法,即存在大量的会话消息未命中,所以并不具有实际使用意义。虽然HM算法具有较低的会话破坏度(会话迁出数量为0),但由于HM算法中散列值的分布具有不规律性,导致其会话破坏分布度较高。

MOLB算法拥有三种算法中较小的会话破坏度,同时其会话破坏分布度也维持在一个较低的水平,因此在基于消息面向会话的Diameter网络中,MOLB算法能够保证在结构频繁变化的情况下各实体服务节点间的负载均衡,并能够较好地维持已有的会话通信。

## 4 结束语

提出了一种基于消息的负载均衡(MOLB)方法。MOLB算法通过建立实体节点与虚拟节点间的映射,使得节点的分布较为均衡;利用散列函数将虚拟节点映射到散列域中,保证了会话的一致性;采用红黑树来维护虚拟节点与其散列值间的映射,以提高查询目的地址和更新实体节点等

功能的效率。为了真实客观地评价出MOLB算法的性能,定义了三个评价标准:负载均衡度、会话破坏度和会话破坏分布度。通过分析在Diameter网络中测得的实验数据得知:与其他常用的负载均衡算法相比,该算法具有较小的负载均衡度和会话破坏度以及较低的会话破坏分布度。

## 参考文献:

- [1] Edith Cohen. Proactive caching of DNS records: addressing a performance bottleneck[C]// Applications and the Internet, 2001: 85-94.
- [2] IETF. DNS Support for load balancing[S/OL]. [2010-10-09]. <http://tools.ietf.org/html/rfc1794>.
- [3] Karger D, Lehman E, Leighton F, et al. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web[C]// The 29th Annual ACM Symp on Theory of Computing, El Paso, 1997: 654-663.
- [4] Godfrey B, Lakshminarayanan K, Surana S. Load balancing in dynamic structured P2P systems[C]// The Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, 2004: 2253-2262.
- [5] Latze C. Roaming, accounting and seamless handover in EAP-TLS authenticated networks[C]// Software, Telecommunications and Computer Networks, 2008.
- [6] Wu W T. Design and implementation of WIRE Diameter[C]// The 3rd International Conference on Information Technology: Research and Education, 2005: 428-433.
- [7] Knuth Donald E. Sorting and Searching trees[M]// The Art of Computer Programming, New Tersey: Addison-wesley Professional, 1973.
- [8] Xie Nan-bin, Huang Xiang-dan. The mixed encryption algorithm based on MD5 and XOR transformation[C]// Education Technology and Computer Science (ETCS), 2010: 394-396.
- [9] Hi W G, MacGregor M H, Gburzynski P. Load balancing for parallel forwarding[J]. IEEE/ACM Trans on Networking, 2005, 13(4): 790-801.
- [10] 张晓明. 网络处理器设计的关键技术研究[D]. 长沙:国防科学技术大学, 2006.