

融合通信中 SIP 服务器研究与优化^①

王雨辰^{1,2}, 胡顺境³, 吕立², 贾正锋²

¹(中国科学院大学, 北京 100049)

²(中国科学院 沈阳计算技术研究所, 沈阳 110168)

³(东港人民广播电台, 东港 118399)

摘要: 随着即时通信类应用, 诸如微信, 阿里钉钉, 易信等类似应用在当下移动互联网时代的不断涌现, 如何在融合通信的现有体系中高效的支持该类应用, 是需要考虑的问题. SIP 作为应用层的信令协议, 在处理会话类业务上具有优势, 但在处理消息类业务上却显得不足. 本文在研究和分析 SIP 的基础上, 设计并实现了一种新的 SIP 服务器系统, 可以更好的支持消息类业务的处理. 并可通过集群的方式完成多个 SIP Broker 服务器的分布式部署, 具有良好的开放性, 为未来业务的发展提供了支撑.

关键词: SIP; 即时通信; 融合通信技术

引用格式: 王雨辰, 胡顺境, 吕立, 贾正锋. 融合通信中 SIP 服务器研究与优化. 计算机系统应用, 2017, 26(11): 89-94. <http://www.c-s-a.org.cn/1003-3254/6074.html>

Research and Optimization of SIP Server in Information Communications Technology

WANG Yu-Chen^{1,2}, HU Shun-Jing³, LV Li², JIA Zheng-Feng²

¹(University of Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

³(Donggang National Radio, Donggang 118399, China)

Abstract: In the Mobile Internet era, many Instant Messaging applications such as WeChat, Ali DingTalk, YiXin and other similar applications have emerged. How to serve for Instant Messaging applications effectively in the current system is a problem to be considered. Session Initiation Protocol is a signaling protocol in application layer. It has advantages in dealing with session, but inadequate in dealing with instant message. This paper describes the design and implementation of a new SIP server system based on the research and analysis of SIP which can deal with instant message better. In addition, SIP Broker can be deployed through the cluster, which can provide supports for future services expansion.

Key words: session initiation protocol; instant messaging; information communications technology

SIP(Session Initiation Protocol) 会话初始协议是一个应用层控制协议, 可以用来建立, 修改和终止多媒体会话 (或者会议)^[1]. SIP 协议具有只定义如何管理会话而不定义如何建立会话的灵活性, 以及易于扩充而不损坏网络的重要特点^[2]. 随着移动互联网的逐渐成熟, 通信方式已经从过去单纯的语音通话逐渐转变为多媒体通信^[3]. 基于 SIP 协议的传统服务器架构已经难以适用于今天的业务需求, 特别是在 IM(Instant Messenge)

即时消息的处理上显得尤为突出. 并且随着用户数量的大规模提升, 融合通信对并发处理 SIP 请求能力的要求越来越高. 因此对 SIP 服务器系统的研究与优化, 以及研究 SIP 服务器的分布式部署和集群搭建具有意义.

本文第一部分简述了 SIP 协议的特点, 着重分析了 SIP 协议现有处理 IM 消息机制. 此外还对 SIP 开源项目 OpenSIPS 做了简单介绍. 第二部分指出了 SIP 协议在处理即时消息问题上的不足, 设计了优化后的

① 基金项目: 教育部-中国移动科研基金 (MCM20150103)

收稿时间: 2017-02-17; 修改时间: 2017-03-23; 采用时间: 2017-03-27

SIP 服务器系统, 根据发布/订阅模型对当前 SIP 协议进行扩展, 对 IM Broker 服务器进行设计. 并对 SIP 集群的设计思路进行阐述. 第三部根据以上设计思路对 OpenSIPS 开源服务器进行二次开发, 实现了 IM Broker, 并通过集群的方式部署. 第四部分对系统的功能和性能进行验证. 最后对所做工作进行总结.

1 协议分析及开源项目

1.1 SIP 协议简介

通信都可以简单的分为两部分: 信令部分和媒体部分. 其中信令部分负责呼叫的建立和控制, 而媒体部分负责将通信的具体内容通过合适的方式携带. SIP (Session Initiation Protocol) 会话初始协议是互联网工程任务组 (IETF) 于 2001 年推出的 RFC (Request For Comments) 标准. 它是一个应用层协议, 被用来建立、修改和终止多媒体会话, 也就是负责信令部分.

SIP 协议采用客户端/服务端模型, 一个 SIP 逻辑实体包括 UAC (User Agent Client) 用户代理客户端和 UAS (User Agent Server) 用户代理服务器. 前者创建并发送一个请求, 后者对 SIP 请求做出响应, 包括应答接受、拒绝或者转发对应的请求.

1.2 SIP 协议现有处理即时消息机制

2007 年 9 月, IETF RFC4975 文档定义了消息会话中继协议^[4](MSRP 协议), 用于在已经建立的 IM 会话中传输即时消息的内容, 完成信息交互过程. 基于 SIP/IP 网络的即时消息技术可划分为三种通信模式^[4]:

1) 页面模式 (Page Mode)

页面模式下的即时消息通过 SIP MESSAGE 方法完成消息的递送过程. 该模式直接在端点之间进行 MESSAGE 消息交互, 因此又称为寻呼模式. 页面模式下的即时消息受限于 SIP 消息体的大小限制 (1300 字节), 只适用于小型消息的发送和接收.

2) 大消息模式 (Large Message Mode)

大消息模式下的即时消息通过 SIP/MSRP 结合的方式完成消息的递送过程. 首先通过 SIP INVITE 方法建立 SIP 会话, 协商 MSRP 底层参数. 然后通过 MSRP 协议在媒体平面完成消息的传输过程. 当消息传输完成后即通过 SIP BYE 方法释放 SIP 会话. 大消息模式适用于包含多媒体数据或文件的大型即时消息的传输过程, 不受即时消息长度的限制.

3) 会话模式 (Session Mode)

会话模式下的即时消息交互也是通过 SIP/MSRP

结合的方式完成消息的交互过程, 即通过 SIP INVITE 方法建立会话, 协商 MSRP 底层参数, 通过 MSRP 传输消息内容, 通过 SIP BYE 方法释放会话. 与大消息模式的区别在于, 会话模式下通过 MSRP 协议进行多次即时消息交互过程, 直到通信方决定结束即时消息通信再完成会话的释放过程. 会话模式适用于聊天室、会议等持续一段时间的即时消息交互方式.

MSRP 协议的典型应用流程如图 1 所示.

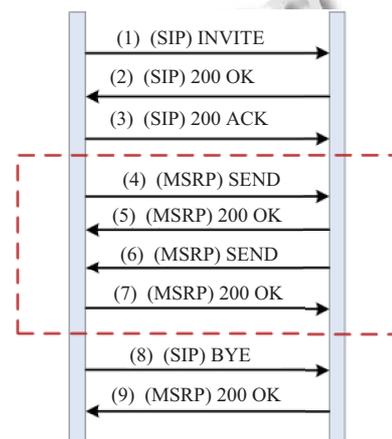


图 1 MSRP 协议典型应用流程

1.3 开源服务器 OpenSIPS

OpenSIPS 是一个成熟的开源 SIP 服务器, 除了提供基本的 SIP 代理及 SIP 路由功能外, 还提供了一些应用级的功能. OpenSIPS 的结构非常灵活, 其核心路由功能完全通过脚本来实现, 可灵活定制各种路由策略, 可灵活应用于语音、视频通信、IM 以及 Presence 等多种应用^[12]. 同时 OpenSIPS 性能上是目前最快的 SIP 服务器之一, 可用于电信级产品构建. 凭借其可扩展、模块化的系统架构, OpenSIPS 提供了一个高度灵活的、用户可配置的路由引擎, 可以为 voice、video、IM 和 presence 等服务提供强大高效的路由、鉴权、NAT、网关协议转化等功能. 其架构如图 2 所示.

2 系统设计

2.1 架构设计

为满足当今用户对丰富多媒体通信的需求, 融合通信技术整合了电信网和互联网两大网络. 通信方式不仅仅是传统的语音, 短信. 还有视频通话, 视频会议, 即时消息, 企业通讯录, 公共账号, 轻应用等一系列服务. 其整体架构设计如图 3 所示.

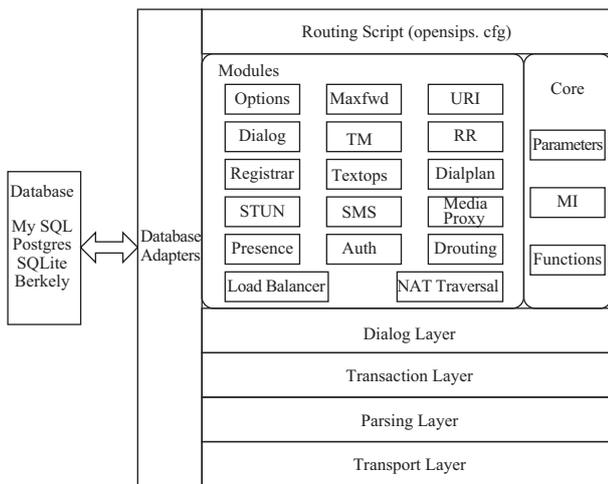


图2 OpenSIPS 架构

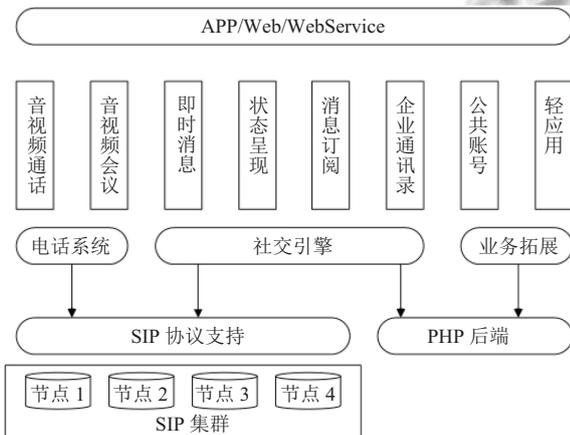


图3 融合通信系统架构

从图中可以看出, 在融合通信系统中, SIP 服务器系统主要负责两部分内容: 一部分是传统的音视频业务, 其中包括普通点对点的音视频业务以及会议中发起的音视频业务. 另一部分是社交引擎的一部分, 包括即时消息, 状态呈现, 消息订阅. 当前 SIP 服务器系统在处理音视频业务上已经非常成熟, 但在社交引擎上, 特别是对即时消息的处理上显现出不足. 从上一节 SIP 协议对即时消息处理的方式可以看出, SIP 的处理方式依然采用建立会话的思想: 即需要进行通信的双方必须建立连接, 并准确知道对方地址. MSRP 定义了两个方法: SEND 和 REPORT. SEND 方法用来发送 MSRP 消息, REPORT 用来报告消息传输过程中的状态信息. 这样的处理方式不仅会导致消息的收发可能出现延迟 (特别是在群聊的时候), 而且对即时消息的大小也有限制. 这是当前 SIP 服务器系统存在的一个

问题, 解决问题的方法是提出一种新的基于话题的发布/订阅模型的处理即时消息机制.

另一个问题是, 单一 SIP 服务器节点提供服务的方式虽然操作和执行功能简单, 但当负载量超过单一的设备所能处理的极限时, 系统将无法再提供更多的服务. 即使增加硬件性能 (如增加内存, 提升网络带宽) 也只能在一定范围内满足需求, 无法从根本上解决问题. 这个问题的处理方法是分离 SIP 服务器逻辑功能, 通过 Cluster (集群) 的方式完成多个 SIP Broker 服务器的分布式部署.

2.2 拓展 SIP 发布/订阅模型

传统 SIP 消息处理方式依然采用建立会话的思想, 即一会话一连接, n 会话 n 连接的方式. 这样处理的缺点在上一小结中已经进行阐明. 为了避免频繁的建立连接, 减少建立连接造成的时延和流量, 提出了一种基于发布订阅/订阅模型的处理即时消息机制. 基本思路是: 在现有的 SIP 代理服务器, 重定向服务器, 注册服务器, 定位服务器的基础之上, 拓展一种 IM Broker 服务器. 该服务器的主要功能是: 通过注册服务器获取在线客户端 (此处实际上为 User Agent Client), 并与之建立长连接 (当用户下线后释放). 每个用户拥有一个属于自己的 IM 主题, 并且每个用户的每个终端均订阅该 IM 主题. 在发送即时消息的时候, 发送方向 IM Broker 服务器发送一条包含接收方 IM 主题的消息. 这样 IM Broker 一旦收到有发向某个用户的相应消息, 可以立刻将消息发送给该用户, 由于在用户上线时已与 IM 服务器建立连接, 故避免了建立连接的代价, 其设计思路如图 4 所示.

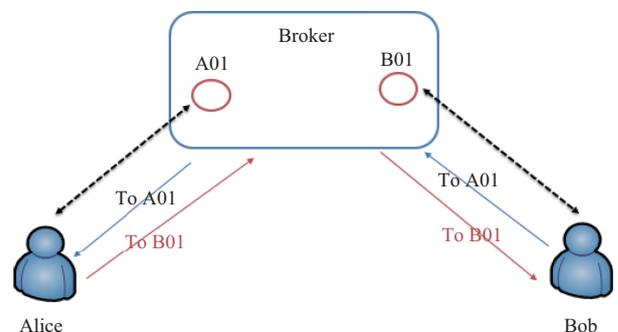


图4 SIP 发布/订阅模型

图中 Alice 和 Bob 所代表的两个 User Agent Client 分别提前与 IM Broker 建立好连接, 并订阅各自相应话

题. 当 Alice 想给 Bob 发送信息时, 只需发送给 IM Broker, 并标明发送给主题 B01(即 B 订阅的主题), IM Broker 服务器接收到后会转发给 Bob. Bob 给 Alice 发消息的流程同理. 这样就避免了 Alice 和 Bob 之间重复多次连接的代价.

2.3 集群设计

随着融合通信的广泛应用, 用户数量的大规模增加, 构建高性能的 SIP 服务器不仅需要改进对即时消息类业务的处理方式, 还需要构建高并发高可用性的集群. 原始的单一 SIP 服务器节点提供的 SIP 服务操作固然简单, 但当负载量超过单一的设备所能处理极限时, 系统服务质量将大大降低.

SIP 服务器在逻辑上由四种基本实体构成: 代理服务, 重定向服务器, 注册服务器, 定位服务器. 目前的融合通信系统中, 往往一台节点包含上述所有逻辑功能, 这在实现上固然简单, 但降低了系统的扩展性和开放性, 可以考虑采用集群架构, 如图 5.

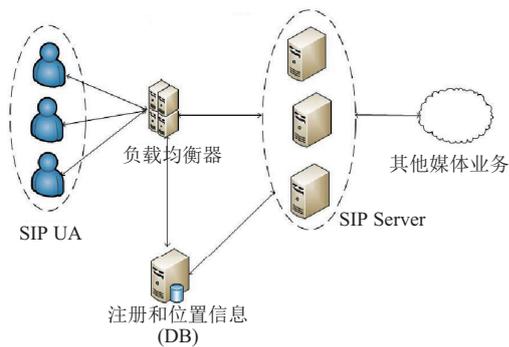


图 5 SIP 服务器集群

设计集群的基本思路是分离其逻辑功能, 将 SIP 服务器系统中的几个基本实体以及上一小结拓展的 IM Broker 分离, 使其做到各司其职. 这样做会降低单 SIP 服务器系统的耦合度, 并减少单个节点压力. 在设计上的难点是对负载均衡器的设计. 负载均衡器是设计集群的核心. 当其收到初始请求时, 需要根据请求类型和当前系统的各个服务器节点负载量做出判断, 将请求转发给系统中合适的节点. 其设计要点有:

(1) 对注册请求转发: 由于集群中已经对注册服务器进行了分离, 所以当负载均衡器收到 REGISTER 请求时要将注册请求转发给分离出来的注册服务器. (2) 节点信息统计: 负载均衡器需要智能的将请求转发给各个节点, 故需要对每个节点的基本信息进行统计, 其

中包括节点是否在运行, 节点最大承载量, 节点当前负载量. 负载均衡器根据以上相关信息进行会话的转发. (3) 心跳检测: 各个 SIP 服务器节点需要与负载均衡器保持心跳连接. 这样做一方面可以检测当前各个节点是否在正常运行, 另一方面可以感知新加入的节点.

3 系统实现

3.1 IM Broker 实现

为了实现上一节中的基于发布/订阅模型的 IM Broker. 需要对现有 SIP 协议进行拓展. SIP 设计者在保持其核心协议简洁的同时, 为其建立了强大的扩展机制. SIP 的扩充方法主要是在消息的 3 个基本部分 (消息基本类型、消息头、消息体). 在 OpenSIPS-2.2.2 中的 Presence 模块由于已经支持 SUBSCRIBE、NOTIFY、PUBLISH 方法, 所以选择消息头扩展方式即可满足要求. 所谓消息头的扩展是指用户可以根据自己的需要增加新的消息头来支持新的特性. 例如, 可以为支持 SUBSCRIBE/NOTIFY 类型消息, 增加 event 头域, 用以注明要订阅的事件类型. 这里以 SUBSCRIBE 为例说明如何扩展以支持 IM 类业务.

在原有 SUBSCRIBE 订阅方法中, event 头域在 Presence 模块中的值是 presence, 为了满足 IM 类业务需求, 现扩展一种新的 event 头域 im. 即当解析到一条 SIP 消息的 event 头域为 im 时, SIP Broker 服务器转向处理 im 的逻辑. 在 SUBSCRIBE 方法中处理该方法的函数为 handle_subscribe, 该函数接收一个 struct sip_msg 类型的形参作为要接收待处理的一条 sip 消息. 函数处理逻辑为:

(1) 调用 parse_header() 函数解析接收到的 SIP 消息.

(2) 调用 parse_to_init() 函数解析 SIP 消息的 to 头域, 获取被叫用户的 URI. 事实上这里的 to 头域一般就是处理 IM 消息的 SIP Broker 服务器地址.

(3) 调用 parse_from_uri() 函数解析 SIP 消息的 from 头域, 获取主叫用户的 URI.

(4) 检查 event 头域中, 用 search_event 函数检查当前消息的 event 是否是已知类型.

(5) line-height: 15.7pt 解析到的 event 若为 im, 调用 parse_body 函数解析 SIP 消息体, 以便查询主叫用户订阅的是哪个主题.

```

event= search_event(parsed_event);
if(event== NULL)
{
    LM_ERR("un-registered support for known event [%.*s]\n",
        parsed_event->text.len, parsed_event->text.s);
    goto bad_event;
}
subs.event= event;
/* extract the id if any*/
ev_param= parsed_event->params;
while(ev_param)
{
    if(ev_param->name.len== 2 && strncasecmp(ev_param->name.s, "id", 2)== 0)
    {
        subs.event_id= ev_param->body;
        break;
    }
    ev_param= ev_param->next;
}
    
```

图6 handle_subscribe 部分逻辑

3.2 构建高性能集群

3.2.1 请求转发

本文在 2.3 节讲到, SIP 服务器在逻辑上由四种基本实体构成: 代理服务器, 重定向服务器, 注册服务器, 定位服务器. 传统的 SIP 服务器各个逻辑功能间耦合度较高, 而构建集群的思路是将 SIP 服务器的逻辑功能分离, 降低单个节点的负载, 按照逻辑功能的不同进行协调工作.

下面以分离注册请求来分析在 OpenSIPS 中如何实现将 SIP 服务器的逻辑功能分离. 分离注册请求是通过注册服务器的分离实现的. 当负载均衡服务器收到客户端发送的 REGISTER 请求时, 需要将其转发给注册服务器进行处理. 首先在 OpenSIPS 的 is_method 模块中判断接受到的请求方法, 然后将在 OpenSIPS 的 tm 模块中利用 t_replay 方法将本条请求进行转发, 转发给已经分离的注册服务器, 最后由注册服务器进行处理.

```

event= search_event(parsed_event);
if(event== NULL)
{
    LM_ERR("un-registered support for known event [%.*s]\n",
        parsed_event->text.len, parsed_event->text.s);
    goto bad_event;
}
subs.event= event;
/* extract the id if any*/
ev_param= parsed_event->params;
while(ev_param)
{
    if(ev_param->name.len== 2 && strncasecmp(ev_param->name.s, "id", 2)== 0)
    {
        subs.event_id= ev_param->body;
        break;
    }
    ev_param= ev_param->next;
}
    
```

图7 分离注册请求核心逻辑

3.2.2 负载均衡

本文在 2.3 节部分讲到, 负载均衡器需要智能的将请求转发给各个节点, 这就需要对每个节点的基本信息进行统计, 来构成负载均衡策略. 在 SIP 服务器系统中, 一般用 SIP 事务来作为衡量一个节点负载量的粒

度. 为了衡量一个节点的负载量, 在数据库库中设计负载量表来作为参考标准, 如表 1.

表 1 负载量表

name	Type	Size	Null	Key
nid	unsigned int	10	no	Primary
name	string	20	no	
adress	string	64	no	
status	unsigned int	1	no	
waiting	unsigned int	1	no	
work_num	unsigned int	5	no	
max_num	unsigned int	5	no	
moniter	unsigned int		no	

表 1 中 nid 唯一标识一个节点, 一个节点启动后, status 置为 1, waiting 置为 0. 对于等待使用的节点, 将 status 置为 0, waiting 置为 1. 通过 work_num 来记录当前该节点所负载的事务数, max_num 来记录该节点所能承受的最大事务数. 通过这样一个表, 负载均衡器分析各个节点情况后将请求转发给不同节点.

综合上述实现, 负载均衡器接受到一个请求后处理流程如图 8 所示.

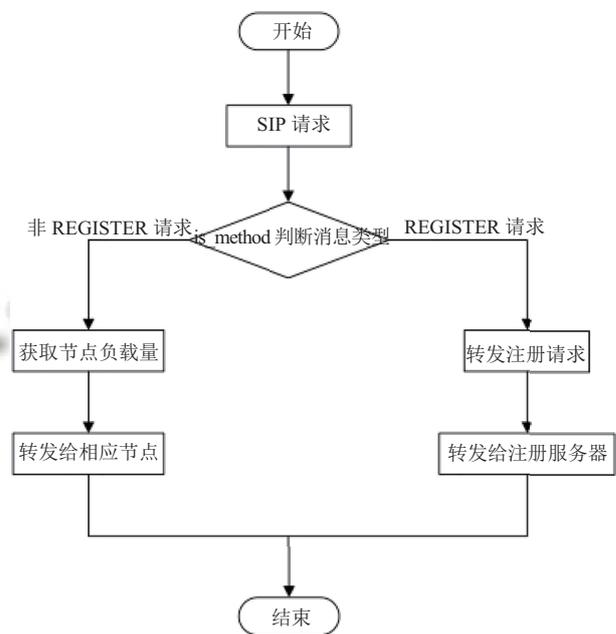


图8 负载均衡器处理流程

4 系统验证

4.1 IM 功能验证

为了验证 IM Broker 功能, 使用中国科学院沈阳计算技术研究所产品企发号作为客户端作为测试用例, 在 ubuntu 14.04 下使用 wireshark 作为抓包工具, 当客

户端之间进行消息收发时, 截获数据包如图9所示。

从捕获的数据包中可以看出, 在 Method 为 NOTIFY 方法的数据包中, 其消息头中的 event 头域类型为 im 类型, 即拓展的类型。由此证明客户端之间在收发消息时使用了基于发布/订阅模型的 IM Broker。

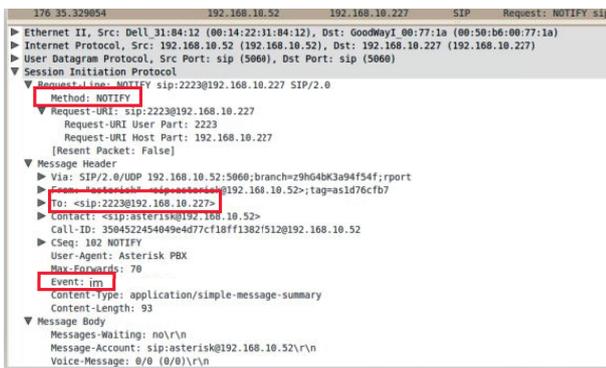


图9 im 抓包

4.2 集群性能验证

为了对 SIP 服务器集群负载能力进行测试, 需要在客户端模拟大规模用户对服务器发起连接请求。SIPp 是用于 SIP 协议的免费开源测试工具, 可以使用 INVITE 和 BYE 方法建立和释放多个呼叫。针对 SIP 服务器集群, 采用系统极限吞吐率作为衡量标准, 所谓极限吞吐率指系统能成功处理 99% 以上的 SIP 请求。分别测试单一节点, 集群不同数量节点系统极限吞吐率, 验证结果如图10所示。

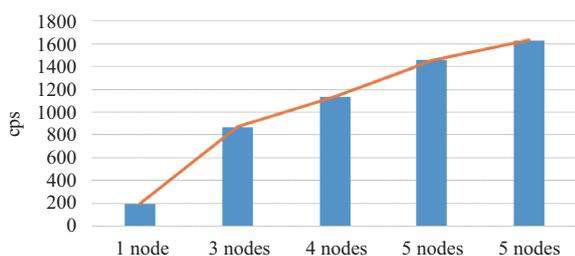


图10 不同节点吞吐率

从图10可以得出两个结论: 一是集群吞吐率比单一 SIP 节点吞吐率大幅提升; 二是在一定范围内, 集群吞吐率随着集群内节点的增加而呈线性增长。可以验证集群达到了对原先 SIP 服务器系统的优化与提升。

5 结语

本文对 SIP 协议及 SIP 协议现有处理 IM 消息机制进行研究, 分析 SIP 协议中已有的 SUBSCRIBE、NOTIFY、PUBLISH 方法方法, 对现有 IM 消息机制进行创新, 提出了一种基于发布/订阅模型的 IM 方法。通过扩展 SIP 协议头域的方式, 利用开源项目 OpenSIPS, 对 presence 模块进行扩展, 实现了一种 IM Broker 服务器以对即时消息类业务支持。对集群构建中的消息转发, 负载量衡量, 心跳检测等问题进行研究与设计。并通过 OpenSIPS 中的 tm 模块完成了集群的搭建。对系统的功能和性能进行测试后的结果表明实现了预期设计目标。未来将对该 SIP 服务器系统进行继续优化, 并逐步应用到现有融合通信系统中。

参考文献

- Rosenberg J, Schulzrinne H, Camarillo G, et al. SIP: Session initiation protocol. RFC 3261, Internet Engineering Task Force, 2002.
- 马荣荣. 移动统一通信中 SIP 协议节流传输机制的设计与实现[硕士学位论文]. 北京: 中国科学院大学, 2016.
- 王久远. 高性能 SIP 服务器系统的设计与实现[硕士学位论文]. 北京: 中国科学院研究生院 (沈阳计算技术研究所), 2016.
- Campbell B, Mahy R, Jennings C. The message session relay protocol (MSRP). RFC 4975, IETF, 2007.
- 关琳, 杨维忠, 张琳峰. 即时消息——网络融合中的亮点业务. 移动通信, 2008, 32(14): 37-41. [doi: 10.3969/j.issn.1006-1010.2008.14.008]
- IBM. IBM MQ telemetry transport format and protocol. <http://msqq.org>. [2013-06-05].
- Roach AB. SIP-specific event notification. RFC 6665, IETF, 2012.
- Rosenberg J. A presence event package for the session initiation protocol (SIP). RFC 3856, IETF, 2004.
- Roach AB, Campbell B, Rosenberg J. A session initiation protocol (SIP) event notification extension for resource lists. RFC 4662, IETF, 2006.
- Bitar N, Zhang R, Kumaki K. Inter-AS requirements for the path computation element communication protocol (PCECP). RFC 5376, IETF, 2008.
- Niemi A. Session initiation protocol (SIP) extension for event state publication. RFC 3903, IETF, 2004.
- Goncalves FE. Building telephony systems with openSIPS 1.6. Packt Publishing, 2010.