

SIP 服务器系统^①

林 浒¹, 王久远^{1,2}, 杨海波¹, 贾正峰¹

¹(中国科学院 沈阳计算技术研究所, 沈阳 110168)

²(中国科学院大学, 北京 100049)

摘 要: 随着会话初始协议 SIP(Session Initiation Protocol)的完善和发展, 基于 SIP 协议的服务器广泛应用于通信系统中. 然而在硬件不变的前提下, 单一 SIP 服务器的负载能力有限, 无法满足大规模用户数量的增长. 本文在充分研究和分析 SIP 协议的基础上, 设计实现了一个 SIP 服务器系统. 该系统中, 使用多个服务器实例协调工作, 并发的增加了处理请求的数量, 从而提升对 SIP 服务的处理能力. 该设计具有易于实现和扩展性强的特点. 实践证明, 本文设计的方案能够解决当前的不足, 为未来的业务的扩展提供了支撑.

关键词: 会话初始协议 SIP; 服务器; 性能

SIP Server Clusters

LIN Hu¹, WANG Jiu-Yuan^{1,2}, YANG Hai-Bo¹, JIA Zheng-Feng¹

¹(Shenyang Institute of Computing Technology, Chinese Academy of Sciences Shenyang 110168, China)

²(University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: With the development and improvement of Session Initiation Protocol (SIP), servers based on SIP have been widely applied in Communication Systems. However, the performance of single SIP server to handle SIP messages is limited on the condition of the hardware does not upgrade. The performance won't meet the demand of large-scale increasing of the number of users. Through of research and analysis on SIP, the paper designed and implemented SIP Server Clusters. By using multiple servers working together, more requests can be dealt with concurrently, which is an improvement of performance. The design is simple to implement and has high scalability. These clusters Practice has proved that the proposed solution can adequately address the current shortage, providing supports for future services expansion.

Key words: session initiation protocol(SIP); server; performance

SIP(Session Initiation Protocol)会话初始协议, 它是一个通用目标的信令协议, 用于控制多种类型的多媒体会话^[1]. 它采用基于文本的编码方式, 具有简单灵活、实现成本低、易于扩展等特点. 目前, 越来越多的支持 SIP 协议的软交换设备和其他 SIP 应用服务器, 被广泛应用于 VoIP(Voice over IP), 即时通信, 音视频会议等应用中, 处于核心地位, 其功能和性能表现影响着整个应用的工作. 然而, 单一的 SIP 服务器, 可以支持数百, 甚至数千的用户, 但是通信应用提供商需要上万或者更多的用户支持量. 因此, 提高对 SIP 服务的处理能力至关重要. 一个普遍的想法就是利用多

个实例协同工作. 本文就是基于这种思想, 通过对 SIP 协议的研究, 设计实现了一个 SIP 服务器系统, 协调多个 SIP 服务器同时工作, 从而提升了对 SIP 服务的处理能力, 提升用户支持量.

本文第一部分简短介绍了 SIP 协议的背景及本文实现系统所使用的开源项目. 第二部分根据使用多个实例协调工作的思想, 对系统的结构进行设计. 第三部分根据设计的结构进行了实现. 第四部分对系统性能进行了评测. 最后对全文进行了总结, 并对下一步工作进行了展望.

① 收稿时间:2015-09-01;收到修改稿时间:2015-10-14

1 背景

1.1 SIP 协议和 SIP 实体

SIP是下一代网络(NGN)中的核心协议之一,是由互联网工程任务组(IETF)提出,用来解决IP网上多媒体会话的控制。SIP是应用层协议,独立于底层的传输协议和会话的类型,可以用来建立、修改和终止有多方参与的多媒体会话进程^[2]。

SIP网络有两种类型的逻辑SIP实体组成:用户代理和SIP服务器。用户代理是SIP的终端系统,它具有很多种存在方式,如软电话(softphone)、实体SIP话机等。用户代理可以分为用户代理客户机(UAC)和用户代理服务器(UAS)。SIP服务器逻辑上主要分为三种形式存在,分别是代理服务器,重定向服务器和注册服务器,但一个物理设备可以具有多种SIP服务器的逻辑功能。代理服务器用于处理SIP请求和响应,并可以按照需求解释、修改和重写请求。重定向服务器查询SIP请求中被叫方的可替代位置,将其映射为新的地址。注册服务器接受用户代理的注册并保存,为后续处理呼叫提供依据。

1.2 使用的开源项目

OpenSIPS是一个成熟的开源SIP服务器,遵从IETF RFC3261中描述的SIP协议。它不仅具有基本的SIP服务器功能,还提供了一些应用级的功能^[3]。OpenSIPS提供了用户可以自行按需配置的路由引擎,同时是目前性能上最快的SIP服务器之一,已经被诸多通信运营商广泛应用。

OpenSIPS的架构十分灵活,其核心功能都可以通过脚本控制,以模块的形式加载,配置和使用^[9]。它采用lex和yacc工具设计了自己的语法规则,使其配置文件的执行速度达到了C语言级别。OpenSIPS的架构如图1^[4]。

SIPp是一个开源的SIP协议性能的测试工具。它内置了一些基本的用户代理工作场景,通过使用INVITE和BYE请求建立和释放多个会话。它也能读取用户的XML文件来配置工作场景。SIPp可以用来测试不同的SIP设备,例如SIP代理,SIP媒体服务器等^[10]。

2 系统结构的设计

2.1 分离逻辑功能

如上文所述,虽然SIP服务器逻辑上主要分为三种形式,但是通常在同一物理设备上集成多种逻辑功

能。对于单一的SIP服务器,该服务器将实现所有所需的逻辑功能,这样做的优点是各逻辑功能之间通信简单,快速,缺点是耦合度高,各逻辑功能对硬件资源划分不明确,扩展性差。

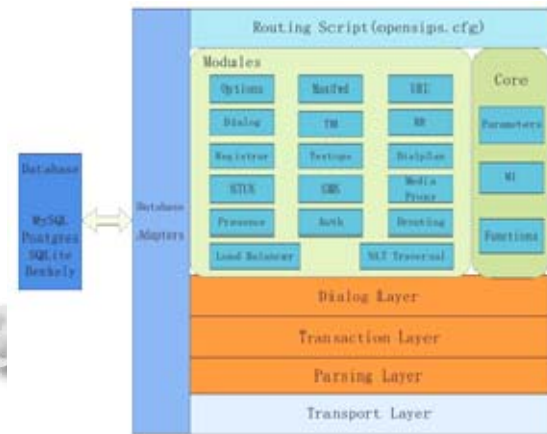


图1 OpenSIPS架构图

对于SIP服务器系统,如果依旧保持旧的实现方式的话,将会进一步增加实例间的耦合度,各个实例间无法协调工作,同时实例间的通信会十分的复杂,从而导致系统的性能低下,无法达到预期的性能要求。因此,将SIP服务器的逻辑功能相互分离,减低单一实例中对逻辑功能的支持能力,将各个实例按照提供逻辑功能的不同进行划分并协调工作,构成一个服务器系统。从外部看来,系统如同单一的SIP服务器一样提供SIP服务。从内部看,各个实例“各司其职”,对于自身所支持的逻辑功能,可以充分利用硬件资源。所以将SIP系统中各个实例按照功能进行分离,可以松散耦合,充分利用硬件资源和提高可扩展性。

2.2 SIP 协议的特性

在SIP协议中,为了保证SIP会话的完整性,会话的建立、修改与删除都必须经过同一服务器处理,即INVITE和BYE等SIP事务请求必须转发到同一SIP服务器实例中。因此系统中需要额外利用一个实例,用于决定选择将SIP请求和响应发送给适当的实例,保证上述SIP事务的粘性。根据SIP协议的特点,可以将SIP消息分为三类,即SIP对话内请求消息,SIP对话外请求消息和SIP响应消息。对于SIP对话外请求消息,也就是对话初始请求,可以直接选择某个实例进行处理并记录,这样对于后续的该对话内的请求,均发送给同一个实例处理该请求。SIP对话内请求和SIP

响应消息直接查询相应对话的记录, 选择实例节点进行转发. 如果没有找到相关记录的话, 返回失败.

2.3 负载均衡

为了使系统的性能得到最大的利用, 每个实例都能发挥充分发挥自己的性能, 负载均衡是必不可少的. 否则, 有可能某个实例处理的 SIP 消息过多造成拥堵, 无法处理更多的消息, 而此时其他的实例只有少数的消息正在处理, 甚至闲置. 显然, 上述情况无法充分利用系统的性能, 如果为了满足更多的用户会提高了系统的成本, 同时降低用户的体验性.

结合上述 SIP 事务的粘性, 为了平衡负载, 必须在处理 SIP 对话外请求时, 选择合适的实例进行转发并记录. 目前比较常用的几种负载均衡的调度算法有: 哈希调度算法, 轮转调度算法, 负载最低算法^[4,5]. 哈希调度算法, 是利用特定的哈希函数将负载请求映射到某个服务实例. 该算法可以不需要额外的操作就可以 SIP 事务的粘性, 但是负载是否平均受哈希函数选取的影响. 轮转调度算法, 是指将负载请求依次发给系统中的各个服务实例, 各个服务节点的地位完全平等, 未考虑其他因素如响应时间的影响. 负载最低算法, 是将新的负载请求发给系统中负载最低的实例, 是一种动态负载均衡的调度算法, 但是实现较为复杂. 为简化系统, 本文使用轮转调度算法实现负载均衡的功能.

2.4 结构设计

结合上述三个小节的讨论, 对 SIP 服务器系统进行了设计, 如图 2.

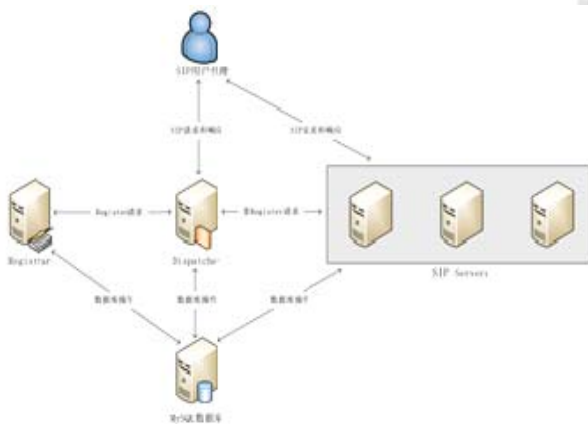


图 2 SIP 服务器系统结构图

增加 Dispatcher 实例, 用于将 SIP 请求消息转发给

相应的实例, 保证 SIP 事务的粘性, 同时为系统提够负载均衡的策略. 系统共享的数据库选用 MySQL, 用于保存用户的注册信息, 对于 SIP Server 实现了 location service 的功能, 同时为保证 SIP 事务的粘性提供支持. 使用一个单独的实例 Registrar 实现了注册服务器的功能, 对于系统的所有注册请求消息由它处理. SIP Server 为系统的核心组件, 实现了代理服务器的功能. 每个实例用于负责部分对话的所有请求和应答, 不同的实例处理的对话没有交集.

3 系统的分析与实现

3.1 消息处理的分析

对于 SIP 服务器系统, Dispatcher 有至关重要的地位. 当用户代理打算发起一个会话时, 将初始请求发送给 Dispatcher 服务器, 由它负责将不同的 SIP 请求消息转发给合适的实例.

在收到注册请求消息, Dispatcher 将其转发给 Registrar, 并通过在消息中添加 Via 头域保证了 SIP 响应消息也可以经过 Dispatcher, 这样可以对外隐藏了注册服务器的存在, 提高了安全性. Registrar 服务器收到注册请求消息, 将所需要的信息保存到数据库中. 处理注册请求消息的流程图如图 3.

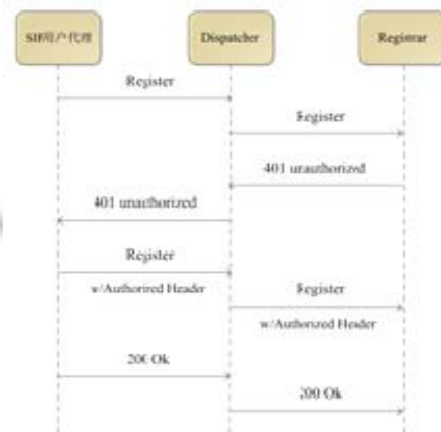


图 3 注册消息流程

Dispatcher 利用消息中的 Call-ID 的哈希值在数据库中维护一个处于正在活动的对话表, 用于记录活动的对话和相应的 SIP Server 实例的地址. 在收到非注册请求消息时, 首先会利用该消息中的 Call-ID 的哈希值判断是否是活动中的对话. 如果是活动中对话的请求或响应消息, 则直接读取记录表中的实例地址转发该消息. 如果不是活动中的对话且是请求消息, 即 SIP

对话外请求消息, 则 Dispatcher 服务器根据设定的负载均衡策略选择一个 SIP Server 实例转发该请求, 并更新记录. 否则返回失败. 详细处理流程如图 4.



图 4 非注册消息流程

3.2 系统的实现

系统中所有实例均选用开源项目 OpenSIPS, 其功能十分丰富, 例如系统中使用 db_mysql 模块实现与 MySQL 的连接与读写, 使用 dispatcher 实现轮转调度的算法等. 为降低系统的复杂度, 每个服务器实例使用相同的硬件配置, 如表 1. 并假设每个 SIP Server 具有相同的初始负载能力. 这样可以消除一些复杂因素的影响, 更加适合轮转调度算法的使用.

表 1 实例硬件配置

CPU 型号	Intel64 Family 6 Model 60
CPU 主频	3.5GHz
内存	2GB
硬盘	200G
网卡	RealTek PCI-E 千兆网卡
操作系统	Ubuntu server 14.04
软件	openSIPS/SIPp

4 系统的性能评测

对于 SIP 的性能测试主要是测试 SIP 实体相关的性能, 一般采用大量的 SIP 请求来模拟真实的呼叫场景^[6]. 测试服务器在进行多路呼叫的情况下, 记录对比消息接收及转发过程中的成功率、每秒处理呼叫个

数等^[7]. 本文使用经典的 SIPp 作为测试工具. 它能使用 INVITE 和 BYE 建议和释放 SIP 会话. SIPp 能动态显示测试中的统计数据, 包括呼叫速率、统计消息等. 本系统目标是提高对 SIP 服务的处理能力, 因此本文只针对系统的吞吐量进行测试. 系统的吞吐量用系统每秒完成的 SIP 请求数来衡量. 其中, 峰值吞吐量是指系统在能完成 99.9% 的 SIP 请求的情况下所能维持的最大吞吐量^[8].

使用 SIP 的基本 UAC 模型. 首先 UAC 发起 INVITE 请求, UAS 收到请求后依次响应 100 Trying, 180 Ringing 和 200 OK 消息, 然后 UAC 发送 ACK 确认建立会话. 在停顿一分钟后, 由 UAC 发送 BYE 消息结束会话, UAS 发送 200 OK 响应要求. 在对系统进行测试前, 先对 SIP Server 单个实例的性能进行测试, 测试结果为单个实例的最大吞吐量约为 300cps. 然后对系统中, 对使用不同实例节点数量进行测试. 此时结果如图 5.

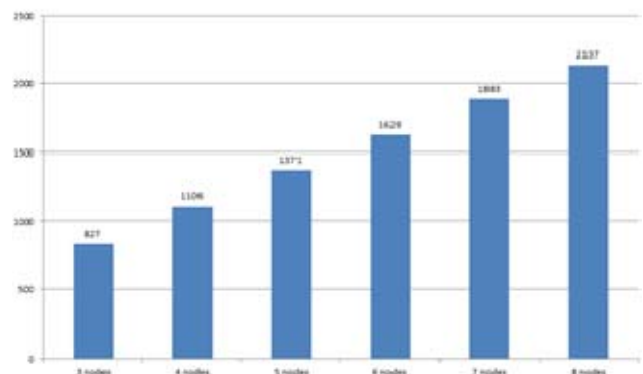


图 5 不同实例数的系统最大峰值吞吐量

5 结束语

本文通过对 SIP 协议的研究和分析, 根据实际应用情况, 指出了单一 SIP 服务器的性能有限的缺点. 根据此缺点, 利用现有的开源软件 OpenSIPS, 设计和实现使用多个实例协调工作的 SIP 服务器系统, 该系统具有结构简单, 扩展性强等特点. 并利用开源测试工具 SIPp 对系统性能进行了测试, 表明该系统对 SIP 服务的处理能力具有提升. 不过该结构的不足之处在于实例失效时, 该实例正在处理的对话将全部失效, 未来还需要对系统进行优化和改进.

参考文献

1 Jiang HB, Iyengar A. Design, implementation, and

- performance of a load balancer of SIP server clusters. IEEE/ACM Trans. on Networking, 2012, 20(4): 1190-1201.
- 2 李伟,赵保华,周颢.SIP 服务器性能测试系统的研究与实现. 计算机应用与软件,2007,24(5):12-14.
- 3 Goncalves FE. Building telephony systems with open SIPS 1.6. 1th ed. Packt Publishing, 2010: 63-90
- 4 夏三波.SIP 服务器集群系统的负载均衡技术研究[硕士学位论文].武汉:华中科技大学,2012.
- 5 Teng SB, Liao JX, Zhu XM. Dynamic weighted random load balancing algorithm for SIP application server. Dependable and Adaptable Networks and Services Lecture Notes in Computer Science, 2007, 4606: 103-110.
- 6 Subramanian SV, Dutta R. A study of performance and scalability metrics of a SIP proxy server-a practical approach. Journal of Computer and System Sciences, 2011, 77: 884-897
- 7 詹艳.SIP 服务器测试方法研究.光电系统,2011,6(2):23-26
- 8 Krishnamurthy R, Rouskas GN. Evaluation of SIP proxy server performance: Packet-level measurements and queuing model. IEEE ICC Communication Qos, Reliability and Modeling Symposium. 2013. 2326-2330.
- 9 Opensips Project. <http://www.opensips.org/>. [2015-6].
- 10 SIPp Project. <http://sipp.sourceforge.net/>. [2015-6].