

SIP 协议栈滚动升级策略与研究

付 玮¹,袁鑫昌²

(1. 武汉邮电科学研究院,湖北 武汉 430074; 2. 武汉中光通信公司,湖北 武汉 430074)

摘要:由于会话初始协议(SIP)已逐渐成为下一代网络的核心协议,并广泛应用于电信等关键部门,因此,对 SIP 服务器的可用性要求也越来越高。文章根据对 SIP 协议栈分层结构重新认识,提出了一套完整的高可用 SIP 协议栈解决方案,在此方案的基础上,详细介绍了协议栈滚动升级策略,并研究了其在部署过程中遇到的一些问题和应对措施。
关键词:会话初始协议;高可用;滚动升级;分布式
中图分类号:TN915 **文献标志码:**A **文章编号:**1005-8788(2006)04-0028-03

SIP stack rolling upgrade strategy and related studies

Fu Wei¹, Yuan Xingchang²

(1. Wuhan Research Institute of Post and Telecommunications, Wuhan 430074, China;
2. Wuhan Zhongguang Telecommunications Co., Wuhan 430074, China)

Abstract: As the Session Initiation Protocol (SIP) has gradually become the core protocol for the next generation networks and has been widely applied in telecom and other key sectors, increasingly higher availability of SIP servers is required. A complete high availability SIP stack solution has been presented on the basis of the re-analysis of the layered structure of the SIP stack, the protocol stack rolling upgrade strategy explained in detail and the problems to be encountered in its applications and the relevant resolutions studied.
Key words: SIP; high availability; rolling upgrade; distributed

随着会话初始协议(Session Initiation Protocol, SIP)在下一代网络(NGN)中应用越来越广泛, SIP 协议栈的重要性也越来越显得突出,特别是在电信等重要部门,对 SIP 服务器的可靠性要求更高:要求其具备“5 个 9”的高可用性(即可用时间概率>99.999%)。本文所介绍的滚动升级技术允许 SIP 协议栈在维护升级期间仍能保持服务器不停机、服务不中断的高可用性。滚动升级能力已成为评价 SIP 服务器是否具备高可用性的主要依据之一。

1 高可用 SIP 协议栈

1.1 基本分层结构

SIP 是一个典型的分层结构协议^[1]。消息传输层负责接收请求和发送响应,提供中间件功能。该层在逻辑上位于由用户和应用程序组成的高层协议栈与由操作系统组成的低层之间,使协议栈代码独立于操作系统^[2]。实践中可选用自适应通信环境 ACE(Adaptive Communication Environment)^[3]等开源项目进行构建。

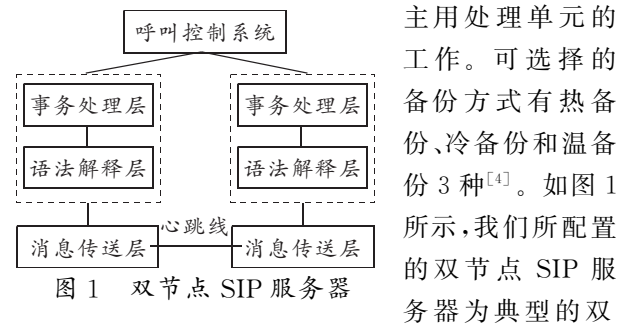
消息传输层之上是 SIP 协议栈层,我们根据 SIP 的特性将其划分为上下两层,即语法解释层(下层)和事务处理层(上层)。SIP 是采用 UTF-8 字符集来进行编码的文本协议。SIP 消息分请求和响应两类,其中请求消息由客户机发往服务器,响应消息

由服务器发往客户机。下层负责对 SIP 请求消息进行解码,并根据消息类型由特定接口原语发送消息到上层进行事务处理。反之,上层(事务处理层)也会由接口原语发送消息到下层,下层将其编码为 SIP 响应消息。

事务是 SIP 的基本元素。一个事务是由客户机事务发送给服务器事务的请求,以及对应该请求的从服务器事务发送回客户机的所有响应组成。事务处理层负责处理消息重传、匹配响应到请求、以及应用层超时等状态事件。

1.2 服务器架构分析

要使 SIP 协议栈具备滚动升级高可用性,协议栈就要至少分配在两个处理单元中,这里的处理单元可以是节点服务器,也可以是进程池中的某个进程,二者必须拥有完全相同的结构和事务状态信息。在任何时候,任何一个备用处理单元都能随时接管



机备份分布式容错架构,备份节点保存着主节点实时变化的状态信息。当主节点服务器出现故障时,备份节点可及时接管主节点工作^[4]。

图 1 中虚线框内的部分是 SIP 协议栈实现的抽象表示,分为上下两层。层与层之间采用松耦合结构,通过特定接口原语进行通信。采用这种松耦合结构的好处是上下层可分别升级,当其中一层需要升级时,不会影响到另一层。在实践中,采用这种松耦合结构,也更容易使各层的功能件模块化。

呼叫控制系统位于其他节点服务器上,仅对整个通信系统的呼叫进行管理,并不参与对 SIP 协议栈的具体控制,因此,这里不做详细讨论。

2 协议栈滚动升级

2.1 滚动升级的概念

软件滚动升级是指在一个多处理单元的系统
中,某一时刻对其中单个处理单元进行升级且不会
使整个系统服务中断的升级过程^[5]。

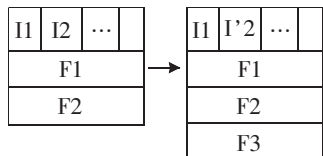
2.2 协议栈升级的缘由

对协议栈进行定期维护和增强都属于软件开发生命周期的一部分,具体说来,SIP 协议栈升级的原因在于以下几个因素:(1) 引入了新头字段,需要语法解析层进行解释;(2) 市场需要开发新的业务,协议栈应新增事务状态处理模块;(3) 对现有协议栈的某些 bug(错误)进行修正;(4) 新的 SIP 扩展协议被制定,要求实现其功能。

根据对服务器的架构分析可知,前两个因素只需对一层进行升级,而后两个因素则要对两层同时进行升级。

2.3 软件升级模型

软件升级实际上是将程序中的接口(Interface)



版本 1 版本 2
图 2 软件升级模型

和特性 (Feature) 进行必要的增加、删除或修改的过程。图 2 演示了一个软件升级过程。从图中可知, 升级后的版本 2 程序有一个新特性 F3, 并通现新版本特性。

作为一个单一程序,这样的升级很平常,也不会出现什么问题。但是,在分层结构的协议栈中,这样的升级设计则需要特别小心,新接口 I'2 是否能继续和其他未升级的接口通信,以及如何对新特性 F3 进行控制,都成为滚动升级要考虑的关键问题。

2.4 滚动升级的关键问题

(1) 接口兼容问题

滚动升级后的新版本处理单元不仅要能同旧版本的其他处理单元进行通信,还要能同比它更新的处理单元通信。我们称这种能力为协议栈前后兼容能力。

例如,协议栈上层某事务模块升级后,事务处理层接口发生变化,语法解释层接口不变。升级后的事务处理层必须能够兼容旧版本的语法解释层,即语法解释层发送的任何消息都能够由事务处理层正确处理。

(2) 接口特性控制

当新版本处理单元的新特性接口还不能与其他处理单元正常通信时,要求该处理单元具有灵活控制其接口的能力,有时也称其为版本接口同步化处理。即在其他旧版本程序未升级之前,应当关闭新版本处理单元中的新特性功能,只有当检测到有与之相符的新接口时,才会自动打开新特性功能。

还是引用上面的例子,当升级后的事务处理层检测到语法解释层还不能解释新特性所传输的消息时,就不应该发送新消息到下层。当语法解释层也升级后,事务处理层才可以发送新特性消息到下层。

这两个问题将会始终贯穿于整个滚动升级流程。

2.5 滚动升级流程阶段分析

下面重点介绍滚动升级实现的各个阶段,如图 3 所示。

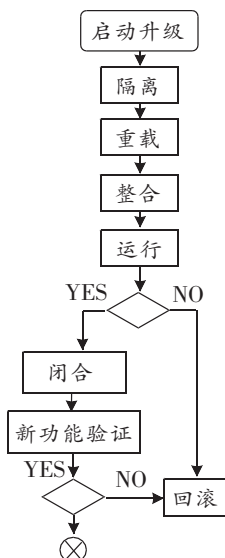


图 3 滚动升级流程图

(1) 隔离阶段

隔离阶段要求系统首先将待升级的处理单元(备用)进行逻辑分离,所有业务数据将由主处理单元处理,关闭待升级协议栈和呼叫控制系统之间的通信,中断“状态备份”连接。

(2) 重载阶段

一旦系统完成备用处理单元的隔离后,被隔离的处理单元程序被终止,根据实际升级要求对协议栈上下层选择升级。新版

本程序将启动,替代旧版本程序。

替换程序应该满足最小替换原则,也就是使系

统更新所涉及或影响到的范围尽可能小。

(3) 整合阶段

当被隔离的处理单元完成新版本协议栈启动,并完成必要的程序内部配置后,应尽快解除其隔离状态,将其重新整合到系统中。此时,栈管理程序将会在第一时间发现新版本协议栈,并对其进行初始化配置。

(4) 运行阶段

升级后的协议栈重新整合到新系统中后,新协议栈会告知旧协议栈将所有会话状态信息进行复制迁移。呼叫控制系统将接收到新协议栈发送的消息,并和主处理单元发送信息进行比较分析。验证新协议栈发送数据是否和旧版本一致,其目的就是要验证新版本是否能够完全取代旧版本。

此时还要求监控新版本协议栈的运行情况,若发现新版本协议栈导致系统性能降低,或者有大量非法消息产生,表明新版本程序工作有问题,必须能使其回滚到旧版本协议栈。

如果新版本系统性能或服务可接受,则可以进行到下面的闭合阶段。

(5) 闭合阶段

在此阶段,已升级并验证其可用的新版本协议栈将完全接管当前的所有通信业务,其余所有未升级的协议栈处理单元将被隔离,并再次进行重载、整合,整个过程同前面所述的升级方式一致。只不过在此阶段可省略对于系统服务可用性的监视。

(6) 新功能验证阶段

当所有协议栈都升级完成,系统已完全运行新版本协议栈时,系统开始检测新版本协议栈带来的新特性是否能正常工作,同时还必须验证这些新功能对系统性能的影响情况。如果系统可用性被接受,则表明滚动升级已经成功。如果系统对新特性功能验证的结果为不可接受,可采用以下两种方案

来解决:① 禁止新特性功能实施,即关闭相应配置,等待后续版本解决该问题。这个方案的前提条件是该系统在禁止新特性后仍能够稳定工作。② 将旧版本协议栈进行回滚操作,完全恢复到协议栈未升级前的状态。

(7) 回滚阶段

当运行或者功能验证失败后,系统将进入回滚阶段,将旧版本协议栈重新载入到系统中。其操作同上述升级操作差不多,但需要以最快速度恢复,很多需要验证功能的地方可以根据情况取舍。

3 结束语

滚动升级是提高服务器可用性的关键技术之一。特别是在通信领域里,电信级的 5 个 9 的高可用性是每个通信设备制造商追求的目标。在滚动升级技术的实践中,我们认为设计阶段是成功部署滚动升级的基础,只有在设计之初充分理解滚动升级原理,日后的升级程序才会非常轻松。

参考文献:

[1] IETF RFC 3261-2002, SIP: Session Initiation Protocol [S].
[2] Tanenbaum Andrew S, Steen Maarten van(著), 杨剑峰, 李敏(译)。分布式系统原理与范型 [M]。北京:清华大学出版社,2004。
[3] Douglas C. Schmidt the ADAPTIVE communication environment [EB/OL]. <http://www.cs.wustl.edu/~schmidt/ACE.html>, 2005.
[4] 构造 TD-SCDMA 系统上的高可用性平台 [EB/OL]. <http://it.sohu.com/20050914/n240387434.shtml>, 2005.
[5] The Linux-HA crowd. Rolling Upgrade [EB/OL]. <http://linux-ha.org/RollingUpgrade>, 1999.



(上接第 6 页)

[2] Takahashi H, Matsuyama A. An approximate solution for the Steiner problem in graphs [J]. Math. Japonica, 1980, 24 (6): 573-577.
[3] Sahin G, Azizoglu M. An efficient wavelength assignment algorithm for service and restoration in WDM rings [A]. Proceedings of OFC 2001 [C]. Anaheim: OFC, 2001. TuO4-1 - TuO4-3.
[4] 周明. 遗传算法原理及应用 [M]。北京:国防工业出版社, 1999。
[5] 焦悦光. 波分复用光网络中的业务梳理 [D]。北京:清华大学电子工程系, 2004。
[6] Baroni S, Bayvel P. Wavelength requirements in arbitrarily connected wavelength-routed optical networks [J]. Lightwave Technology, 1997, 15 (2): 242-251.