第34卷 第19期 计 算 机 工 程 2008年10月
Vol.34 No.19 Computer Engineering October 2008
·网络与通信· 文章编号：1000—3428(2008)19—0094—04 文献标识码：A 中图分类号：N945

FTP服务漂移技术的研究及实现

刘红军，黄遵国
(国防科学技术大学计算机学院，长沙 410073)

摘 要：基于服务漂移必须包含的3类状态——初始状态、活动状态和完成状态，提出服务漂移的体系结构，分析服务漂移的实现过程。面临威胁事件时，服务漂移技术能主动或被动地漂移到备份节点，以适应威胁，决定是否漂移。服务漂移对用户是透明的，用户感觉不到服务器的变化。服务飘移只对服务状态进行漂移，可以减少传输数据量、缩短漂移时间。以FTP服务为例实现了服务漂移的原型。
关键词：可生存性；服务漂移；服务状态；FTP服务

Research and Realization on FTP Service Migration Technology

LIU Hong-jun, HUANG Zun-guo
(School of Computer Science, National University of Defense Technology, Changsha 410073)

【Abstract】Service migration should include three types of state——initial state, active state and complete state. Based on these, this paper proposes the architecture of service migration and analyzes the realization of service migration. When facing security threat, service migration determines whether to migrate, either actively or passively. It is transparent to the users so that they will not feel the change of service. It just migrates the service states with the purpose of shorting migration time by reducing transmitted data. The migration of FTP service is realized.
【Key words】survivability; service migration; service state; FTP service

1 概述

随着我国计算机网络和信息系统建设的发展，各个职能部门越来越依赖于网络信息系统，依赖程度的加深使得网络信息系统的安全问题更加突出。同时系统自身的规模日益庞大，并朝着高度的分布式方向发展，在这种情况下，传统的安全技术越来越无法满足系统的安全要求。因此，可生存性研究作为新的安全理念成为当前网络安全研究的新方向。

在APPDRR(Assessment, Policy, Protection, Detection, Reaction, Restoration)体系中，生存性研究属于Restoration代表的灾难恢复研究范畴。文献[1]通过对与生存性相关的论文进行统计分析，表明绝大多数论文仅认识到可生存性的重要性，缺乏对可生存性的统一认识和标准，对可生存性的实现也都基于非正式应用，没有经过实际的应用检验。

目前可生存性研究大多围绕可生存性是什么、如何测定可生存性以及如何实现可生存性3个问题进行的，尚未形成统一的严格定义。文献[1]认为，在定义可生存性时必须考虑以下5个因素：

(1)系统：要明确提出定义可生存性的分布式网络环境、关键服务类型以及系统是有边界还是无边界等。

(2)威胁：可能影响到系统提供服务的威胁因素，主要分为意外威胁、恶意威胁和灾难威胁。

(3)自适应性：面临威胁事件，系统有能力适应威胁并且继续向用户提供正常服务。

(4)服务的持续性：服务的可用性应该作为系统需求进行定义，网络性能的下降不应该被用户觉察到。

(5)响应时间(及时性)：服务应该在系统要求或者用户所期望的时间内可用。

为了满足在遭受攻击、故障或意外事故时，系统能够及时完成其关键任务，本文提出的服务漂移技术能很好地满足自适应性、服务的持续性和及时性要求，并且以FTP服务为例进行了实现。

2 相关研究

目前专门针对服务漂移的研究还不多，文献[2]针对普适网络环境中的实时通信服务提出了服务初始化和飘移系统，但它没有考虑保存服务的状态，需要在新节点上重启服务。文献[3]为可变的网格计算提出了基于分布式虚拟机的服务漂移方法，它不仅需要迁移进程或线程的执行上下文，还要重构服务在目的节点的执行环境，但是没有阐明如何获取执行环境状态和让服务重新运行所需的应用程序状态。The Stanford's Collective project[4]将机器的完全状态(包括操作系统、应用程序和运行进程)进行压缩封装。它通过虚拟机监视器捕获系统状态，并将封装的状态传送和绑定到目的节点继续执行。但是由于它在操作系统层设计，这种迁移方法带来的负担过重。文献[5]提出一种基于经典虚拟机的结构来提供计算网格中的服务。它允许在任何节点上创建虚拟机，虚拟服务器通过将全部计算环境迁移到远程虚拟服务器上实现迁移。但是文中并没有讨论迁移的细节和相关的开销。

为了支持适应性计算，还对进程和线程的迁移方法进行了大量的研究。但是很难将它们扩展到异构网格中，因为它们依赖的系统软件和编程语言面临可移植性问题。JESSICA2[6]是一个Java虚拟机，它支持异构节点上多线程Java应用程序，通过在节点间传送线程状态来进行线程迁移。它的成功基于能为分布节点提供共享内存的全局对象空间。

基金项目：国家“863”计划基金资助项目“基于全息机制的网络可生存性系统模型及关键技术研究”(2006AA01Z401)
作者简介：刘红军(1982－)，男，硕士研究生，主研方向：网络信息安全；黄遵国，副研究员、博士
收稿日期：2007-11-18 **E-mail：**seeker_lhj@163.com

但是当一个节点失效时，整个系统都会停止运行。HPCM[7]是一个支持在异构环境中进行进程迁移的中间件，但它是通过将预处理代码移动到预运行的实时系统中实现的，因此，仅仅依靠进程和线程迁移不能使服务重新配置和适应计算中的变化环境。

通过以上分析发现，服务漂移面临的主要问题有：没有将服务的状态分类处理并进行详细解析；仅仅将虚拟机上进程/线程的执行状态进行获取和恢复，没有针对应用服务获取其恢复执行所需的最小状态集；对服务迁移机制的研究没有统一的认识。

3 相关定义

3.1 服务类型的定义

要进行服务漂移，首先要理解什么是服务。本文将服务定义为可通过网络被公共访问的、具有良好接口的功能单元，它可有效地封装和操作各类共享资源，因此，只要从网络上得到所需的数据，就可以认为是得到了服务。另一方面，网络上的服务是类型各异、数量庞大的。

从功能上可以将服务大致分为 3 类：(1)路标型服务：此类服务就像提供了一个公告栏，用户可以在上面查询所需要的信息，但不能对其提供的服务信息进行修改，如添加、删除。它主要应用于公用查询、检索等领域，如 DNS 服务、域名解析。(2)仓库型服务：此类服务给各个用户分配了一个私有空间，或者多个用户共享一个公共空间，用户可以向该空间中添加信息、资料等，并能根据其被授予的权限进行相应的修改。这类服务就像提供了一个中转站，大家将事物先存放在一个共享的空间中，需要相关信息的用户可以从该共享空间得到需要的东西。(3)作坊型服务：此类服务需要根据用户的请求对一些基本信息进行加工再处理才能满足用户的需要，即很多情况下用户不能马上得到其需要的信息，需要获取相关的基本信息并进行组合计算才能得到需要的结果。

将这 3 种服务进行比较可知：路标型服务从用户的角度来说只有只读权，不能对服务主体进行修改，所有的服务信息全部由服务主体提供，并且这些服务信息不会经常发生变化。仓库型服务允许用户作为信息提供者之一，并对自己空间内的信息有一定的修改权限，这个权限由服务主体授予，因此，服务的信息变化速度较快，维护的信息量较大。作坊型服务需要根据用户的请求进行再加工才能产生用户需要的结果，这个过程会产生许多中间结果，信息的变化量也较大。

3.2 服务状态的定义

服务主体要为用户提供服务，就必须维护接受服务的所有用户信息，比如用户的登录认证信息等私有信息，否则，用户的服务请求得不到服务主体的允许。服务主体还必须维护其自身能够正常服务的相关信息(比如服务的数据)、服务正常运行的相关配置信息、单个用户服务的临时中间操作信息(比如进行的操作、该操作所需要的对象)以及服务操作的结果信息(比如计算出的结果、上传的文件)。因此，一个服务要能在漂移后继续正常地提供服务，就必须维护这 4 方面的状态信息：用户信息，服务主体信息，中间操作信息和操作结果信息。其中，用户信息和服务主体信息都是初始信息，一般不会经常变动；而中间操作信息是随不同用户、不同时间、不同操作不断变化的，这些信息都是活动的；操作结果信息也是不断变化的，不同的操作、不同的用户、不同的需求所得到的操作结果信息也是不同的，因此，这些信息在操作产生前是不可预见的，而一旦产生，操作结果一般不会改变。上述状态信息可以划分为 3 类状态：初始状态，活动状态，完成状态。

将 3.1 节中划分的 3 类服务结合起来分析可知：(1)路标型服务只需要维护初始状态和活动状态。这类服务是面向大众的，在大多数情况下只须维护服务主体信息，不需要维护用户信息，而活动状态记录用户操作的内容，以便在服务漂移后继续完成该操作。因为用户不能对此类服务进行修改，所以无须维护完成状态。(2)仓库型服务需要维护初始状态、活动状态和完成状态。因为需要授权才能对该用户的私有空间进行操作，所以需要维护用户信息。服务主体在初始情况下会存放一定的信息在仓库中，这些信息需要进行维护，同时服务主体的配置信息(如对访问用户的过滤)也需要保存。由于用户可以向仓库存取信息，因此必须记录用户的操作信息，如上传的文件名称、上传文件的源目录、目的目录。还必须保存用户的操作结果，因为用户可以向仓库存取信息或删除信息，所以需要维护修改后仓库内的结果信息。(3)作坊型服务也需要维护初始状态、活动状态和完成状态。只有获得授权的用户才能得到服务，因此，必须维护用户信息以及服务主体提供的一些基本信息和配置信息。活动状态记录用户的请求，包含请求的内容、需要的相关信息、结果表示等，以便服务漂移后为用户提供的服务继续进行。对加工的结果信息可能因具有代表性而被很多用户需要，所以，需要将操作的结果进行保存。

3.3 服务漂移的定义

在可生存研究中，服务漂移是指在负载过重或遭受生存威胁时，系统能够主动或被动地将服务无缝漂移到工作情况良好的节点上继续提供服务的技术。它要求在漂移过程中对用户透明，这就要求漂移过程是快速的。在完整性方面，服务应该包含数据、代码和状态 3 部分，但实际应用中，备份节点上安装有与主服务节点相同或相似的服务软件。所以，从速度和应用上考虑，只需要将保证服务继续正常运行的状态漂移到备份节点并恢复运行。

4 服务漂移模型

4.1 三层结构的选定

在传统的 C/S 模式中，客户和服务器的直接连接增加了系统的危险性，因此，可以在中间增加 1 层形成 3 层结构：服务层，服务代理层和客户层。服务层和服务代理层的分层设计能够使服务器从繁重的安全任务中解放，专注于高性能和可靠地提供服务，服务代理节点专注于安全任务，对其作全面优化，以有效防御多数攻击。服务层是针对传统生存性研究中较少考虑客户端的安全而设计的。

服务代理层位于服务端和客户端之间，能够有效抵御各种攻击，尤其是分布拒绝服务攻击；隐藏服务器的位置，极大地减少服务器面临的安全威胁；监控服务器的状态，引导用户透明的服务漂移；为服务器提供负载均衡服务等。服务代理节点是针对网络连接管理做优化的精简节点，能够实现大容量的连接管理。

服务层由多个服务节点组成的服务池构成。服务节点可协作完成服务，自适应地执行负载均衡。服务节点在故障状态下可执行服务漂移、服务恢复、服务重配置等动作，确保服务的不间断性。资源分配可动态调整，最大程度地确保关键服务的可持续。

3 层结构分为基于安全性的方式和基于性能的方式，如图 1、图 2 所示。

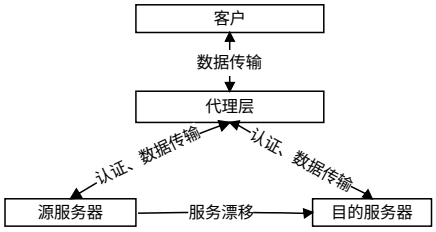

图 1　基于安全性的 3 层结构

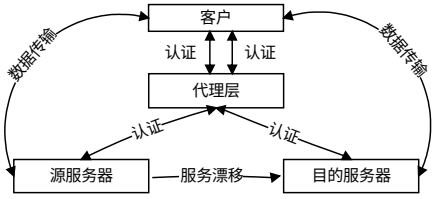

图 2　基于性能的 3 层结构

在图 1 中，所有的交互数据都要经过代理层，这就加重了代理层的负载。从服务器角度来看，代理层就是它的客户；从客户角度来看，代理层就是服务器。代理层起着服务中转的作用，这样 3.2 节中讨论的活动状态就可以保存在代理层中，这从实现和应用来讲都是可行的。当服务漂移后，代理层根据活动状态重新连接服务器，继续完成用户请求，对用户来说，整个过程是透明的。在图 2 中，用户经过代理层向服务器认证后，由服务器和客户直接进行数据传输，提高了用户的响应时间，降低了代理层的负载，但提高性能的同时降低了安全性。同时，由于客户和服务器直接交互，活动状态只能保存在客户端或服务器。如果保存在客户端就必须修改客户端，这在实用性上难以推广，用户也难以接受，可行性不高。如果将活动状态保存在服务器，就会增加服务器的开销，影响服务质量，更为重要的是，当服务漂移后，用户需要重新连接服务器，保存的活动状态对用户没有意义，因为对用户来说，服务的漂移是不透明的。综上所述，本文选用图 1 的 3 层结构。

4.2　服务漂移体系结构

在基于安全性的 3 层结构的基础上，图 3 给出了基于漂移技术的服务体系结构。

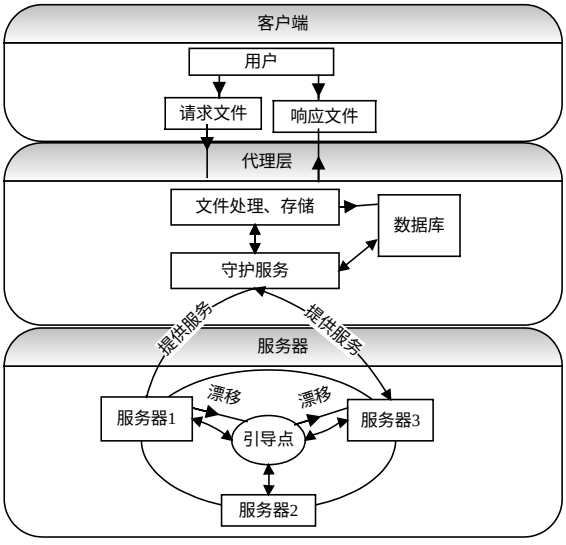

图 3　服务漂移体系结构

用户和服务间的通信大多是使用消息机制实现的，因此，保证消息的可靠性非常重要。消息机制本身具有突发性、不可预知性等特点，要处理这些问题需要加入其他的处理逻辑，增加了开发难度，并且没有考虑消息的持久性问题，一旦发生连接中断或者系统崩溃，其间的消息也就随之丢失，不可再恢复。由于目前大多数协议是面向连接的(例如 HTTP)，因此很难有效地为消息传递提供可靠保障。为此，采用存储/转发方式把消息转换成文件，并将其保存至一个相对独立的文件系统中，然后由该文件系统负责转发请求、接收响应，从而提高消息传递的可靠性。

代理层以文件为处理对象，用户的多个请求能够被集中到同一个文件中以批处理的方式调用服务，能有效减少网络负载和降低对网络连接情况的依赖。代理层负责接收客户端的传输请求，将请求存入数据库，控制服务器进行数据传输，保存传输状态信息，向用户返回传输状态。数据库用来存储客户端发来的传输请求和文件传输状态，当发生网络故障或由其他原因导致传输失败或进行服务漂移后，代理层可以根据数据库中存储的信息进行恢复，提高传输的可靠性。守护服务负责代表客户与服务器进行交互，将用户的请求发送给服务器，接收服务器返回的结果，将请求的中间结果及状态保存到数据库中。如果发生服务漂移或服务器故障，并且用户请求并未完成，守护服务根据用户请求和保存的中间状态自动重新与服务器交互。如果服务器完成了用户的请求，守护服务负责删除数据库中与该请求相关的文件和状态。

服务层由多个服务节点组成的服务池构成，其中一个服务节点作为主服务节点提供服务，其他节点作为备份服务节点。当主节点出现故障或需要服务漂移时，服务就从主服务节点漂移到一个备份节点上。各服务节点间并不直接相连，减少了一个服务节点被攻破后其他服务节点立即被攻击的风险，提高了安全性。它们之间通过引导点进行连接，每隔一段时间就将自身节点当前状态信息(如负载、安全威胁、服务软件信息)传送到引导点进行保存。同时，主服务节点每隔一段时间就将初始状态和完成状态传送到引导点上。当需要漂移时，引导点根据各个服务节点的状态信息选取新的主服务节点，并将原主服务节点传来的初始状态和完成状态传送到新的主服务节点上恢复服务。服务漂移的具体过程如图 4 所示。

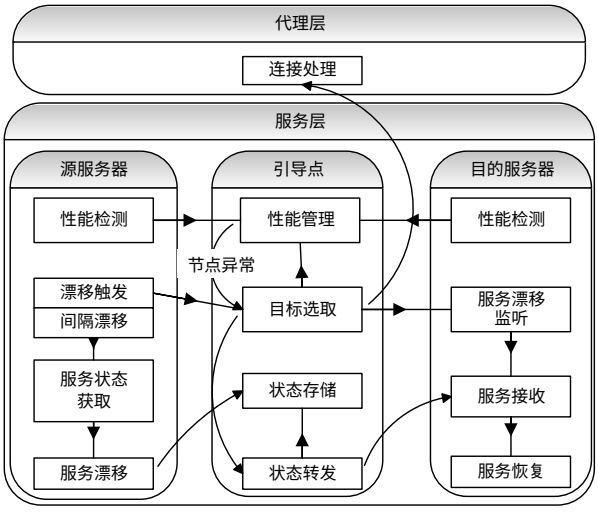

图 4　服务漂移模型

服务层的服务节点都需要间隔一段时间检测自身的性能，并将它发送到引导点的性能管理模块进行综合、分析，作为选取新的主服务节点的依据。当它发现当前主服务节点

在设定的时间间隔内没有将其性能状态发送到引导点，表明主服务节点发生了故障，它就会触发节点异常到目标选取模块选取新的主服务节点，并将选取的目标节点通知代理层，以便将用户请求转发到新的服务节点上，同时通知选取的服务节点监听服务漂移，为接收状态信息做好准备。选取目标节点后，通知状态转发模块将主服务节点的状态信息发送到目的服务器。这个过程防止了源服务节点在遭受突然打击后立即崩溃，在来不及发送漂移信号的情况下，系统能选取新的服务节点继续提供服务，保证服务的连续性。

在源服务节点上每隔一段时间就会触发间隔漂移，使得服务器将当前的完成状态信息和初始状态信息发送到引导点的状态存储模块进行保存。这就防止了源服务节点在需要立即漂移而状态信息又不能在短时间内传送完毕时，丢失大量的当前状态信息，从而无法保存当前计算结果，造成巨大损失。当源服务节点触发漂移时，会发送漂移信号到引导点的目标选取模块，其后的操作过程和引导点发现主服务节点失效后的处理过程一样。

5 FTP 服务漂移实现

FTP 是 Internet 上应用最为广泛的协议之一，因为 FTP 的命令数量非常多，功能很强大，协议底层也很复杂，再加上 FTP 使用的是稳定的 TCP 协议，有很高的传输质量，所以 FTP 是大文件传输、大吞吐量的数据交换的首选途径。因此，以 FTP 服务为例实现服务的漂移具有代表意义和实用价值。

根据服务类型的划分，FTP 服务属于仓库型服务。根据服务漂移体系结构，活动状态存放在代理层，初始状态和完成状态存放在服务层。因此，实现 FTP 服务漂移主要是实现活动状态在代理层的保存与删除及初始状态和完成状态在服务节点间的传送，并且新节点利用这些状态实现服务的恢复。

FTP 服务实现节点间的文件传送，因此，其主要操作是文件的上传和下载。上传和下载的基本原理相似，在具体实现上略有不同。以上传为例，用户首先需要连接服务器进行认证，然后用户执行上传操作，这需要选定上传的文件、在服务器上保存的目录，在上传过程中还需要记录已经传输的数据量以便进行断点续传。活动状态包含的内容如图 5 所示。

服务器信息		认证信息		操作信息				过程信息	
服务器名	服务端口	用户名	密码	操作名称	文件源路径	文件存放路径	选取文件信息	已传输文件大小	缓存文件

图 5　活动状态信息

当正在上传文件并且需要进行服务漂移时，需要在代理层新建一个线程以继续接收用户上传的数据，并进行缓存，当服务漂移后，代理层利用服务器信息和认证信息重新登录到新的服务器上，根据操作信息继续执行未完成的操作，根据过程信息将缓存的数据继续上传到新的服务器上，实现文件传输的断点续传。

当正在下载文件并且需要进行服务漂移时，为保证服务漂移的透明性，需要在服务状态漂移的时间段内向用户发送空数据报文，保持连接的持续性。连接时间是有限的，而服务漂移的时间受状态的数据量大小和网络传输时延的限制，不能保证在连接时限内一定完成服务漂移。在服务漂移后，代理层利用服务器信息和认证信息重新登录到新的服务器上，根据操作信息继续执行未完成的操作，根据过程信息实现文件传输的断点续传。

在服务层需要保存和传送初始状态及完成状态。初始状态包括服务为用户提供的登录信息和操作权限，还包括服务本身的配置信息，具体见图 6。

用户信息				服务配置信息			
用户名	密码	为该用户分配的目录	允许的操作(如下载、上传)	FTP 端口	连接最大用户数	保持连接最大时长	安全过滤模式

图 6　初始状态信息

为了保证传输的可靠性，将用户信息保存在文件 users.dat 中，将服务配置文件保存在文件 configure.dat 中。进行服务漂移时，只需要将这 2 个文件传送到服务器上，并将其中的内容读取出并更新到服务中。

完成状态包括在某一时刻用户对文件操作后在服务器上留下的结果。它可能包括对文件进行了删除、上传、重命名和新建目录等操作后所产生的结果，也可能包括没上传完的临时文件等。由于完成状态的内容是大量的文件，因此可以直接利用 FTP 将完成状态传送到目的节点上。但 FTP 传输文件首先需要认证信息，因此，需要先利用 Socket 将初始状态信息传送到目的节点上，更新服务后才能利用 FTP 传送完成状态。

6 结束语

本文从服务应用层出发，先将服务划分为路标型、仓库型和作坊型 3 种类型，然后分析了服务漂移必须包含的 3 类状态，在此基础上提出了服务漂移的体系结构，最后以 FTP 服务为例实现了服务漂移。由于 FTP 服务具有一定的特殊性，因此下一步工作是研究一般服务进行漂移所必需的标准。

参考文献

[1] 张鸿志, 张玉清, 李学干. 网络可生存性研究进展[J]. 计算机工程, 2005, 31(20): 12-14.

[2] Imai N, Isomura M, Horiuchi H, et al. Flexible and Seamless Service Migration for Real-time Communication with Ubiquitous and Heterogeneous Networked Resources[C]//Proc. of GLOBECOM'04. [S. l.]: IEEE Communications Society, 2004.

[3] Fu Song, Xu Chengzhong. Service Migration in Distributed Virtual Machines for Adaptive Grid Computing[C]//Proc. of ICPP'05. [S. l.]: IEEE Press, 2005.

[4] Sapuntzakis C P, Chandra R, Pfaff B, et al. Optimizing the Migration of Virtual Computers[C]//Proc. of the 5th Conf. on Operating Systems Design and Implementation. Boston, USA: [s. n.], 2002.

[5] Figueiredo R, Dinda P, Fortes J. A Case for Grid Computing on Virtual Machines[C]//Proc. of the International Conference on Distributed Computing Systems. Rhode Island, USA: [s. n.], 2003.

[6] Zhu Wenzhang, Wang Choli, Francis C M. JESSICA2: A Distributed Java Virtual Machine with Transparent Thread Migration Support[C]//Proc. of CLUSTER'02. Chicago, Illinois, USA: IEEE Press, 2002.

[7] Du Cong, Sun Xiaohe, Chanchio K. HPCM: A Pre-compiler Aided Middleware for the Mobility of Legacy Code[C]//Proc. of the Int'l Conf. on Cluster Computing. Hong Kong, China: [s. n.], 2003.