

# 面向 SWIM 系统改进的服务调度算法

吴志军 刘 中 胡涛涛

(中国民航大学电子信息与自动化学院 天津 300300)

**摘 要** 广域信息管理(System Wide Information Management, SWIM)是“航空云(Aeronautical Cloud)”的基础设施,用于航空交通运输相关信息的传输与共享。SWIM 系统的可靠性和生存能力对航空交通运输的安全运行具有重大影响。设计了面向 SWIM 系统生存能力的弹性灾难恢复方案。该方案采用 Linux 虚拟服务的组织架构,改进了 Linux 虚拟服务中的加权最小连接 WLC(Weighted Least-Connection)调度算法,提高了 SWIM 服务的连续性。实验结果表明,改进后的加权最小连接算法可以有效提升 SWIM 系统的灾难恢复能力,满足 SWIM 系统的弹性灾难恢复需求。

**关键词** 广域信息管理,加权最小连接,弹性灾难恢复,业务连续性,生存力

**中图分类号** TP393.2 **文献标识码** A

## Improved Service Scheduling Algorithm for SWIM System

WU Zhi-jun LIU Zhong HU Tao-tao

(Department of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China)

**Abstract** System wide information management is the infrastructure of aeronautical cloud, and it is used for transmission and sharing of information related to air transportation system (ATS). The function of SWIM is to achieve air cloud storage. The reliability and survivability of SWIM have a significant impact on the safe operation of ATS. This paper presented a resilience disaster recovery scheme for the purpose of enhancing the survivability of SWIM system. This scheme adopts the organizational structure of virtual service in Linux and improves weighted least-connection (WLC) algorithm being applied to Linux virtual service. Related experiments have been carried out in the simulation environment, and experimental results show that this proposed scheme meets the requirements of resilient disaster recovery for SWIM system and has better performance in guaranteeing reliability and survivability of SWIM network.

**Keywords** System wide information management, Weighted least-connection, Resilient disaster recovery, Service continuity, Survivability

## 1 研究背景

广域信息管理系统(System Wide Information Management, SWIM)是信息化整合下一代空中交通管理的核心,是一个高度集成的大规模网络系统<sup>[1]</sup>。SWIM 可以将全球航空成员国的空中交通管理空管(Air Traffic Management, ATM)系统进行信息化集成,并将其连接在一起,也可以将本国内的空中交通管理相关单位系统连接在一起。利用 SWIM 网络框架可以实现机场、空管运行中心、航空公司等多方多部门之间的数据安全交换和信息共享<sup>[2]</sup>。随着我国民航业的快速发展,民航信息化网络系统的建设基本完善,然而民航运输需求的迅速增长与空管处理能力和航空运输总体容量不足的矛盾日益突出,民航业在迎接前所未有的机遇的同时也必须面对这种矛盾带来的挑战。因此,加强对现有民航信息网络的升级改造和建设我国的 SWIM 系统迫在眉睫。航空交通信息管理对国家安全十分重要,对保障 SWIM 系统在恶意攻击和灾难下的生存能力,以及保证其服务的完整性和可用性意义非凡。

“9.11 事件”后,特别是“棱镜门”事件主角斯诺登爆料美国攻击包括中国清华大学骨干网络在内的世界各国主要网络的内幕,世界各国将重要信息系统和网络的应急响应及灾难恢复上升到了前所未有的高度。SWIM 信息共享和信息安全是空管科技领域的一个核心问题,针对 SWIM 信息共享的研究,美欧航空运输发达国家在安全体系结构及关键技术方面取得了重要成果,已经考虑在 SWIM 中实现协同式应急响应与弹性灾难恢复的技术<sup>[3]</sup>。针对这项技术,文献[4]提出了多路由路径的方法,研究了最小失效概率下的多路由算法,以确保网络的快速恢复。文献[5]研究了基于多云灾难恢复的模型 DR-Cloud,使得云计算服务供应商与灾难恢复供应商可以协同应对可能出现的灾难情况。目前,在我国 SWIM 大力发展之际,保证 SWIM 网络的可靠性和生存能力十分必要。因此,研究具有应急响应和灾难备份信息安全保障机制的广域信息管理系统将对我国新一代国家空中交通管理系统具有深远的影响。

本文受国家自然科学基金委员会与中国民航局联合基金项目(U1533107),天津市自然科学基金资助项目(17JJCZDJ30900),中央高校基础研究成果基金项目(3122016D003),中国民航大学研究生课程开发项目资助。

吴志军(1965—),男,博士,教授,博士生导师,主要研究方向为网络与信息安全, E-mail: zjwu@cauc.edu.cn; 刘 中(1990—),女,硕士生,主要研究方向为网络与信息安全; 胡涛涛(1990—),男,硕士,主要研究方向为网络与信息安全。

## 2 SWIM 系统灾难恢复总体框架设计

SWIM 是一个介于 IP 网络层和应用层之间的大规模分布式网络<sup>[6]</sup>。SWIM 系统可以实现机场、航空公司、塔台等多方面的信息共享和数据交换,并为不同子系统的统一调度和协同共筹提供网络保证。SWIM 可以在不同的应用中通过中间件实现数据的转换和共享,现有的民航空管网络以数据管理为核心,而 SWIM 提出了一个“虚拟信息池”的概念,“虚拟信息池”作为不同应用子系统的数据转换中心,实现了数据交换和资源共享,简化了网络框架,使资源得到合理利用,降低了网络的运营成本和维护成本,为不同部门实现协同决策和统一调度奠定了基础。SWIM 有大型的控制调度中心,可对相关各端进行调度控制,因此提出面向 SWIM 系统灾难恢复的体系框架,如图 1<sup>[7-8]</sup>所示。

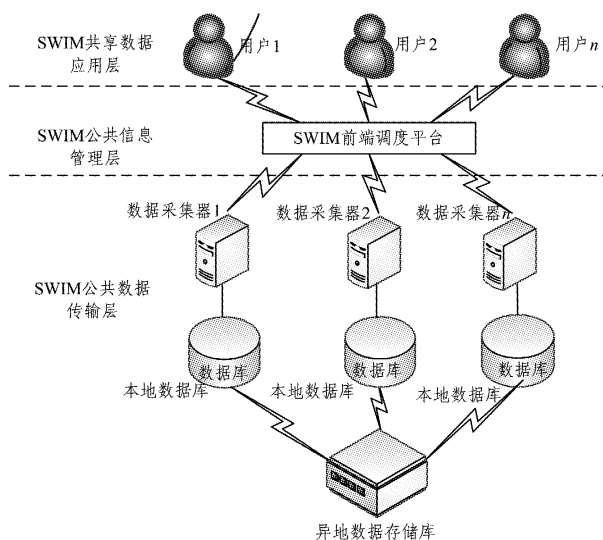


图 1 SWIM 系统灾难恢复总体框架

SWIM 网络系统按照功能的分层结构可分为公共数据传输层、公共信息管理层和共享数据应用层 3 个层次<sup>[7]</sup>,SWIM 公共数据传输层负责实现 SWIM 平台对共享航空飞行资源数据的采集;SWIM 公共信息管理层负责实现对 SWIM 承载业务的负载均衡、配置和数据转换等;SWIM 共享数据应用层负责实现 Web 客户端发出订阅请求。

在不同的地理位置(如机场前端、空管数据采集中心、管控中心等)分别存放不同的服务器,它们共同采集数据,并将数据传递给后端的数据存储库,每个节点之间均是通过网络连接的,它们属于 SWIM 网络系统的公共数据传输层,实现 SWIM 平台对共享航空飞行资源数据的采集;在前端存在一个 SWIM 前端调度平台,当用户需要与后端的服务节点建立连接并访问节点数据时,首先要访问 SWIM 网络前端调度器,它属于 SWIM 网络的公共信息管理层,实现对 SWIM 承载业务的负载均衡、配置和数据转换,以及应急响应预案等。SWIM 调度器通过调度算法将用户的服务请求发送到后端服务节点中,使其与用户建立连接,当调度的服务器不能提供服务时,SWIM 调度器则通过一定的调度算法将服务请求调度到其他服务节点中,以保证正常提供服务。当后端服务器的数据库系统发生异常宕机而无法提供服务时,系统切换到远端的异地数据库中,继续为用户提供服务,从而保证整个系统的可生存性。

## 3 SWIM 系统服务调度算法

SWIM 网络需要提供  $24 \times 7$  的实时服务,任何服务的中断和关键性数据的丢失都会造成不可估计的损失,因此保证 SWIM 网络的可用性和鲁棒性非常重要。

Linux 虚拟服务器(Linux Virtual Sever, LVS)技术是优秀的集群性能解决方案<sup>[9]</sup>,它既可以解决服务集群中的节点负载失衡的问题,保证服务器集群服务的质量,又可以在服务节点发生宕机、瘫痪等情况下使其自动恢复服务,大大增加 SWIM 网络的可靠性和生存能力,因此在 SWIM 网络的负载均衡和灾难恢复建设中可采用 LVS 工作方式。

### 3.1 调度算法

在 LVS 集群系统下,每个用户与集群系统建立连接后,必须与 WEB 服务器节点建立一个 TCP 连接,而每一个用户的请求都会被前端调度器调度到集群中任意真实的服务节点中,使得单个服务器由于突然访问量过大而发生宕机、瘫痪等情况从而系统响应时间的问题得到有效解决。而在 IPVS 内核中存在轮叫调度(Round Robin, RR)、加权轮叫调度(Weighted Round-Robin, WRR)、最小连接调度(Least-Connection, LC)、加权最小连接(Weighted Least-Connection, WLC)等调度算法。

#### (1) 轮叫调度 RR 算法

轮叫调度 RR 算法通过轮转方法来与服务节点建立连接。该算法的优点在于其简洁性,然而当服务器处于过载状态时,轮叫算法依旧向服务器调度连接,这会对集群的生存能力造成严重后果<sup>[10]</sup>。

#### (2) 加权轮叫 WRR 调度算法

加权轮叫 WRR 调度算法是在轮叫调度 RR 算法的基础上,依据服务节点本身的差别和每台服务器性能的不同设定相对应的权值而得到的<sup>[11]</sup>。在客户端发送请求连接时,WRR 算法会依据服务器权值的大小将连接调度到不同的服务器上,权值系数越大的节点在一个周期内被连接的次数越多<sup>[12]</sup>。但是在具体实现中,由于集群中服务器的性能不同,容易造成资源调度不均,并且由于算法本身未考虑节点已经建立连接,当权值系数较大的节点已经建立了通信时,调度算法依然会向权值系数大的服务器继续调度连接,从而影响整个系统的可靠性。

RR 和 WRR 算法属于静态调度算法,特点是无状态和无记忆,不能反映服务节点之前的状态。

#### (3) 最小连接 LC 调度算法

最小连接 LC 调度算法是在客户端向集群发送连接时,把新的连接请求分配到当前连接数最小的后端节点上。这种算法存在很大的缺陷,因为客户端的请求是不同的,每一次请求服务在服务器停留的时间可能会有较大差异,有的请求很短,服务器直接就能返回所需数据,有的请求却需要服务器进行数据计算,这时 TCP 连接就会进入等待状态并占据着服务器资源,那么在实际操作中可能出现两种情况:处理效率好的集群系统节点会分析数据报,TCP 连接进入等待状态;而处理效率差的集群系统节点会在分析所收到的数据报的同时持续收到新的连接,这会导致整体集群系统的处理能力下降,进而影响其可靠性。

#### (4) 加权最小连接调度 WLC 算法

加权最小连接调度 WLC 算法是对 LC 的改进,在考虑到集群系统节点的连接数的同时兼顾节点的性能,为不同性能的节点配置不同的权值,以实现在调度连接时系统集群整体的负载均衡。

加权最小连接调度 WLC 算法的描述如下<sup>[11,13]</sup>。

假设有一组 SWIM 服务器,  $W(S_i)$  表示服务器  $S_i$  的权值,  $C(S_i)$  表示服务器  $S_i$  的当前连接数。所有服务器当前连接数的总和为  $CSUM = \sum C(S_i) (i=0, 1, \dots, n-1)$ 。

当前的新连接请求被发送至服务器  $S_m$  上,  $S_m$  需要满足如下条件:

$$\frac{C(S_m)}{CSUM} / W(S_m) = \min\{ \frac{C(S_i)}{CSUM} / W(S_i) \} (i=0, 1, \dots, n-1) \quad (1)$$

其中,由于  $CSUM$  为常量,因此  $W(S_i)$  是非零值。  $S_m$  的判定条件可以简化为:

$$C(S_m) / W(S_m) = \min\{ C(S_i) / W(S_i) \} (i=0, 1, \dots, n-1) \quad (2)$$

在服务器中,除法运算所消耗的资源比乘法运算大得多,因此,其判定条件可进一步优化为:

$$C(S_m) * W(S_i) > C(S_i) * W(S_m) (i=0, 1, \dots, n-1) \quad (3)$$

加权最小连接调度 WLC 算法的流程如图 2 所示。

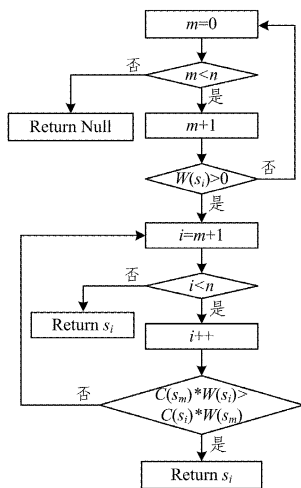


图 2 WLC 算法的流程

WLC 以连接数为参考条件并兼顾节点性能的差异性而进行连接。因此,对于整体的集群系统而言,能够较好地达到均衡状态,增强整个集群的生存能力。WLC 是一种综合性能较好的调度算法。

#### 3.2 WLC 算法的不足

WLC 算法的不足之处在于:1) 服务器权值的设定是管理员凭借自身的经验进行预先手动设定完成的,这显然缺乏合理性和科学性;2) 服务器的权值是固定不变的,服务器自身性能和对数据报的运算能力会随时发生改变,若未能及时相应地更正服务器权值,则无法与当前服务器的负载能力相匹配,影响服务器集群的健康性;3) 权值设定连接数是唯一的调度衡量标准,然而对于整个服务器集群而言,这是远远不够的,服务器的 CPU 利用率、内存利用率、网络带宽利用率、进程数、磁盘利用率等性能指标均能够反映出当前服务器的负载情况,而现有的 WLC 未考虑这些因素。

结合 SWIM 本身的建设需求,本文提出了一种基于 WLC 算法的改进的动态权值算法,该算法会科学地选取负载

参照因子,实时更新服务器的权值,更加科学严谨并真实地体现出服务器的负载情况:

(1) 依照真实服务节点的负载情况,将服务器的权值进行实时更新,由 WLC 算法可知,当有新的用户请求时,首先将请求转发到负载相对较低的节点上,这样可以保证整个系统运行的平稳均衡,最大程度地避免了服务器瘫痪等情况引起的服务中断。

(2) 在体现当前服务节点的负载指标上,改进的动态权值因子算法的提出需要考虑节点的 CPU 利用率、内存利用率和网络带宽利用率来综合反映当前节点的负载状态,并且依据 3 个指标的不同影响将 3 个负载指标进行量化,选取适当的影响系数来计算实时的节点权值。

#### 3.3 改进的动态 WLC 算法

SWIM 提供了丰富的网络服务,不同的网络服务对服务器的资源消耗是不一样的,但是在现有的 WLC 算法中将连接数作为调度细度,不能体现网络服务对真实服务器的不同资源消耗。结合当前实际情况,对于集群系统中的节点,对其负载状态会产生作用的因素包括:处理器使用效率、网络带宽使用效率、系统存活时间、系统响应时间、内存的使用效率等<sup>[14]</sup>。通常考虑的作用因子越多,越能精确反映活动的集群节点负载的情况,但对节点造成的负担就越重。因此,本文在选取负载因子时,只考虑最重要的 3 个影响因子:处理器使用效率、网络带宽使用效率和内存使用效率,这 3 个作用因子的取值范围为  $[0, 1]$ 。在真实节点中,这 3 个作用因子对集群系统的作用效果不同,按照 3 个作用因子在服务器中的作用大小,设置 3 个系数  $L_{cpu}, L_{mem}, L_{net}$ , 这 3 个值分别代表其在真实节点负载中的作用系数。本文依据实验平台的具体参数配置,设置了这些值的大小,按照作用的大小分别设为 0.5, 0.3, 0.2。依据上述理论基础,确定服务器的实时负载系数  $K$  的计算公式为<sup>[15-16]</sup>:

$$K = L_{cpu} * P_{CPU} + L_{mem} * P_{MEM} + L_{net} * P_{NET} \quad (4)$$

处理器使用效率、网络带宽使用效率、内存使用效率的取值范围都在  $[0, 1]$  之间,其 3 个影响系数也都为定值 0.5, 0.3, 0.2, 那么  $K$  的取值范围也在  $[0, 1]$  之间,此公式能够很好地实时反映出服务器所承受的压力。

当负载权值较大导致服务器的承载过大,可能影响服务器的响应时间以及用户体验时,原权值需要适当减小,反之亦然。只有这样动态地增大和减小服务器的权值才能更加科学地反映当前服务器的负载情况。因此,设计一个新的权值计算模型:

$$W_{new} = W + A(0.7 - L) \quad (5)$$

其中,  $W_{new}$  为新的权值, 0.7 表示期望到达的 SWIM 网络设计的理想状态。

此时服务器的负载合适,整个服务器集群的响应时间良好,既能够解决服务器的资源配置问题,又能够很好地提供网络服务。若实时负载  $K$  大于 0.7,则表示服务器当前处于资源紧张、负载过重的状态,可能引起服务响应时间变慢、服务器宕机等问题,需要下调其服务器权值。式中的  $A$  是一个调节参数,是反映权值更改的敏感系数,这里根据实验平台的负载情况和资源配置,将  $A$  的值设定为 10,能够很好地体现出负载因子的改变对新的权值的影响,并且不会过于敏感。且此时新的权值的范围只会  $[-3, 7]$  之间,但是由于初设的服

务器权值不会小于 2,因此经过计算出来的新的权值取值在 [0,7]之间。

当现实中处理器的使用效率、网络带宽的使用效率、内存的使用效率中的任意一个值大于 0.9 时,不论其他值是否在合理范围,此时服务器均处于十分忙碌的状态,濒临瘫痪,因此,为了保证服务质量,我们在处理器的使用效率、网络带宽的使用效率、内存的使用效率中的任意一个值大于 0.9 时,将其服务器的权值均改为 0 并进行提示,需要管理员手动检测服务器的健康状况,做出修复,其整个自适应算法流程如图 3 所示。

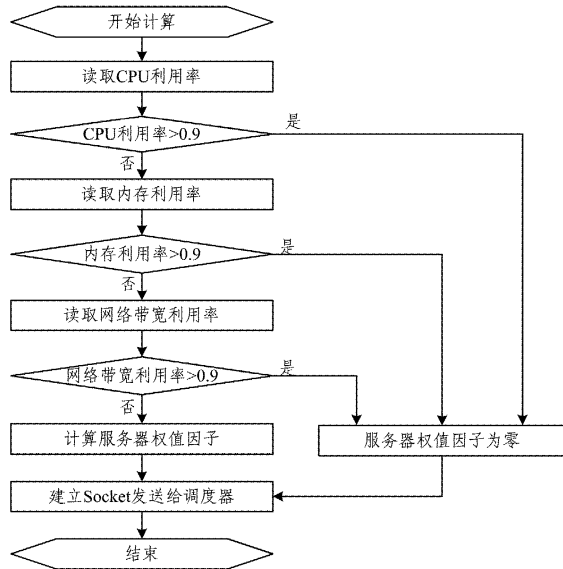


图 3 服务节点权值因子获取流程

当在真实节点端获取了处理器的使用效率、网络带宽的使用效率、内存的使用效率并计算得到了新负载权值后,需要将新获取的实时权值传递给前端的调度器,前端调度器获取了新的权值因子后需要将原权值更改为新的权值,并将新的权值传递到系统内核的 IPVS 程序中去,等待下一次 WLC 调用权值时能够使用已经更新过的节点权值。Linux 虚拟服务技术提供了 IPVSADM 等相关的管理软件,因此,只需要利用 IPVSADM 重新设定新的权值即可。在 Linux 操作系统中,system 函数会将字符串以指令的形式传递给系统内核,从而实现用户层和系统内核之间对权值的传递,其具体流程如图 4 所示。

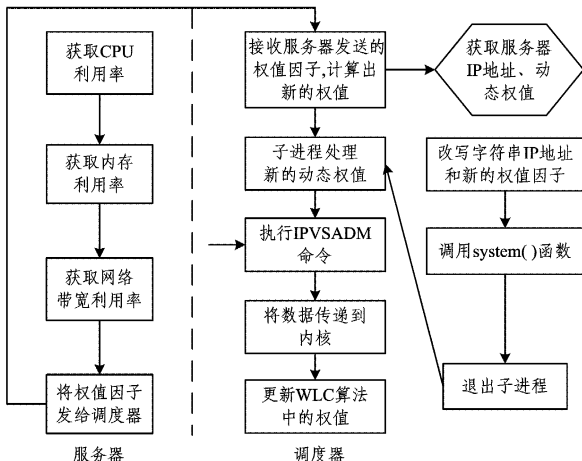


图 4 动态权值自适应调整流程

### 4 实验及结果分析

为了检验改进的加权最小连接调度 WLC 算法的有效性,搭建了 SWIM 模拟实验平台,并配置安装了 Linux 虚拟服务软件工具。实验环境的网络结构拓扑如图 5 所示。

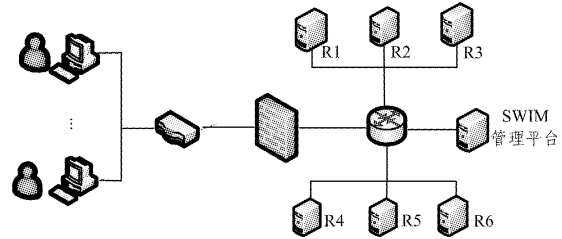


图 5 测试平台网络拓扑结构

实验中,需要使用 PC 机、路由器、防火墙和交换机,其中一台 PC 机作为模拟的 SWIM 管理平台,配置虚拟 IP,并设置加权最小连接算法,对客户机的访问提供调度连接,并验证此改进的加权最小连接调度算法对集群系统的可靠性和灾难恢复是否有效。PC 机的硬件配置如表 1 所列。

表 1 实验 PC 机的硬件配置

名称	操作系统	CPU	内存	IP 地址
测试机	Red Hat 5	Core2 2.66GHz	4GB	10.1.20.30
SWIM 管理平台	Red Hat 5	P4 2.2GHz	1GB	10.1.20.10 VIP 10.1.20.99
服务器	Red Hat 5	Core2 2.66GHz	2GB	10.1.20.11- 10.1.20.20

在 Linux 操作系统中,proc 文件是一个为查看许多系统参数提供条件的虚拟文件。若要监视一台服务器的运行情况,如查看它的处理器的使用效率、网络带宽的使用效率、内存的使用效率等基本使用情况,可以从 /proc 目录读取。因此,实验中,在服务节点均部署数据采集程序,提取参量,计算出实时的参数,然后同 SWIM 管理平台建立 Socket 连接,并将信息传递给前端 SWIM 管理平台。

在环境搭建服务节点时,使用 Tomcat 作为服务器,采用 Mysql 作为数据库,服务节点安装完 Tomcat 后将模拟的 SWIM 访问信息页面放置在 Tomcat 中,然后在 SWIM 管理平台下安装 LVS 管理软件 IPVSADM,在管理平台和服务器群之间部署 LVS,并在每一个服务节点中运行采集处理器使用效率、网络带宽使用效率、内存使用效率的程序,在 SWIM 管理平台运行接受程序,实验环境部署完毕。

Siege 是 Linux 操作系统下常用的开源压力测试软件,可以根据需求模拟仿真多个客户端同时对某一服务器端发起大量访问,并记录访问的平均响应时间、传输的字节数、访问成功率等详细的访问信息。在测试环境平台搭建完成后,可以使用 Siege 模拟不同的客户端对系统集群发起连接请求,通过比较 LVS 自带的 WLC 算法与改进后的算法的区别,判定出面向 SWIM 的 WLC 算法的改进是否满足现阶段对 SWIM 网络的灾难恢复的要求。

#### 4.1 测试指标

对于灾难恢复技术,可以使用 Siege 模拟不同的客户端对系统集群发起连接请求,通过比较 LVS 自带的 WLC 算法与改进后的算法的区别,判定面向 SWIM 的 WLC 算法的改进是否满足现阶段对 SWIM 网络的灾难恢复的要求。对系统的测试主要考察以下几个参量<sup>[17]</sup>。

(1)平均响应时间:即客户端在发出连接请求时,收到的第一个回复所用的时间。平均响应时间反映了整个集群系统的响应速度,平均响应时间越小,系统的响应越迅速,集群系统的可靠性越好;平均响应时间越长,系统的响应越慢,集群系统的可靠性越差。

(2)吞吐量:是指客户端平均每秒收到的字节数的多少,反映了整个系统的处理能力的大小。吞吐量越大,系统的处理能力越强;吞吐量越小,系统的处理能力越差。

(3)灾难恢复时间:是指当服务器集群系统里的某台服务器发生宕机后停止工作时,客户端发出请求后得到经调度器调度产生回应的的时间。这个时间越长,整个系统在遇到突发情况时进行恢复的能力越差。

### 4.2 结果分析

针对不同情况的实验,取得的实验结果及分析如下。

(1)测试系统在采用原有的 WLC 算法和经过改进后的 WLC 算法的情况下,相同的访问人数和相同的访问时间,在相同压力下系统的响应时间。测试时,将真实服务器 1 和服务器 2 的初始权值分别设定为 10 和 8,然后在客户端使用 Siege 设定并发访问时间为 5s,设定不同访问人数进行测试,得到系统响应时间比较曲线,如图 6 所示。

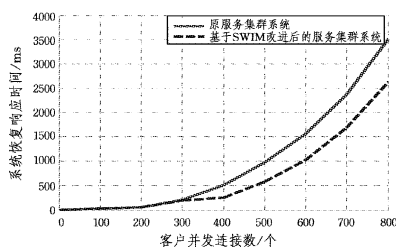


图 6 系统响应时间的比较

根据图 6 得到的结果,整理得到的测试数据如表 2 所列。

表 2 系统响应时间的比较

客户并发连接数	原服务集群系统的平均响应时间/ms	改进后的服务集群平均响应时间/ms
100	23.41	29.66
200	49.32	52.37
300	201.53	197.22
400	507.16	250.17
500	980.35	577.23
600	1560.44	1022.39
700	2366.32	1693.58
800	3511.80	2630.30

由此可知,当连接数较少、整体服务器系统处于空闲状态时,改进后的算法与原有的 WLC 算法的系统响应时间差别不大,此时,整体服务器集群均处于空闲状态,响应时间很短,系统健康性良好。但是,随着连接数的增加,原有的 WLC 算法的响应时间急剧增大,系统的负载急剧增加,而改进后的算法的系统响应时间变化较为平缓,整个系统的负载水平较低,能够更好地保证服务质量,因此改进后的 WLC 算法能够适应 SWIM 大量的数据传输,并增强服务质量和系统可用性。

(2)测试系统在采用经过改进后的 WLC 算法时,服务器中性能差别不大的服务器能否根据性能变化自动调节服务器的权值,以达到整个系统的稳定状态,并保证资源合理分配。测试时,将真实服务器 1 和服务器 2 的初始权值分别设定为 10 和 4,然后在客户端使用 Siege 设定并发访问时间为 80s,同时访问人数为 800,测试得到了服务器权值比较曲线,如图 7 所示。

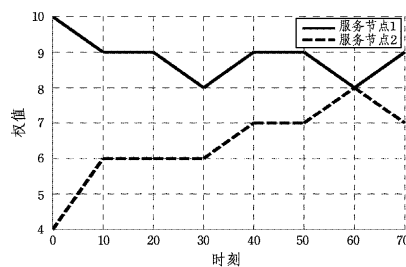


图 7 服务器权值的比较

根据图 7 测试得到的结果,整理的测试数据如表 3 所列。

表 3 服务器权值的比较

时刻	真实服务器 1 的权值	真实服务器 2 的权值
0	10	4
10	9	6
20	9	6
30	8	6
40	9	7
60	8	8
70	9	7

由图 7 和表 3 可以得出:原有的 WLC 算法在设定权值后是固定的,因此,服务器权值会分别固定为 10 和 4,造成在集群系统提供服务时,大量的连接请求指向权值较大的,而权值较小的则处于空闲状态,从而造成资源浪费。相比之下,改进算法能够自动调节后端服务节点的权值大小,根据服务器本身性能自动改变其节点权值,使得整个系统更加健康稳固,满足 SWIM 网络需要不间断服务的要求。

(3)测试提供的服务器在突然中断的情况下,集群系统按照自适应调度策略切换到冗余的服务器上所需的时间。分别配置软件自带最小加权算法和基于 SWIM 改进后的 WLC 进行测试,测试服务器集群的规模与恢复时间的关系如图 8 所示。

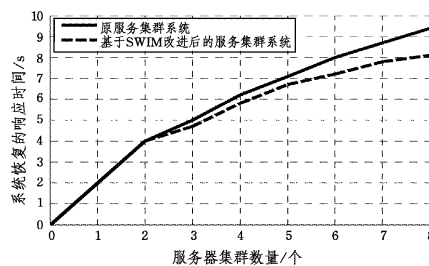


图 8 灾难恢复时间与服务器节点数量的关系

如图 8 所示,当系统内的服务节点数量增加时,整个系统的灾难恢复的响应时间在逐渐增加,这是由于集群内提供服务的节点中断后,WLC 算法需要选定其他的节点替代原节点,服务节点的规模增大,所需要的时间相应增加。但是改进后的 WLC 算法切换到冗余服务节点的时间平均节约了 1.2s,性能得到了提升,具体如表 4 所列。

表 4 集群恢复响应与服务节点数量的关系

服务器集群数量	原服务集群系统/s	基于 SWIM 改进后的服务集群系统/s
2	4	4
3	5	4.7
4	6.2	5.8
5	7.1	6.7
6	8	7.2
7	8.7	7.8
8	9.4	8.1

(4)测试服务器突然中断后,集群系统按照自适应调度策略切换到冗余的服务器上所需的时间。分别配置软件自带最小加权算法和基于 SWIM 改进后的 WLC 进行测试,Siege 设定的客户端访问持续为 5min,节点权值分别设为 10 和 8,得到的测试结果如图 9 所示。

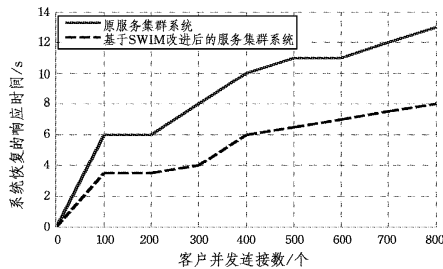


图 9 灾难恢复时间与客户并发数的关系

根据图 9 得到的实验结果,整理得到的数据如表 5 所列。

表 5 集群恢复响应与客户并发连接数的关系

客户并发连接数	原服务集群系统/s	基于 SWIM 改进后的服务集群系统/s
100	6	3.5
200	6	3.5
300	8	4.0
400	10	6.0
500	11	6.5
600	11	6.5
800	13	8.0

通过对以上数据和直观图的分析可知,随着连接数的增加,集群服务器受到的压力增加,基于 SWIM 改进后的服务集群系统在服务器突然中断后,集群系统按照自适应调度策略,切换到冗余的服务器上所需的时间平均减少了 23%,因此,对系统自带的加权最小连接算法进行基于 SWIM 的改进有助于系统的灾难恢复。

**结束语** SWIM 是一个高度集成的大规模网络系统,由于 SWIM 服务的重要性,搭建 SWIM 应急响应和灾难恢复网络架构十分重要,关系整个 SWIM 服务集群的可靠性和实用性。鉴于我国现阶段的空管信息建设基础,借鉴大型网络中常用的 Linux 虚拟服务技术,提出一种面向 SWIM 的应急响应和灾难恢复的网络框架,并结合 SWIM 网络的具体需求,在加权最小连接算法的基础上,针对 SWIM 网络提出改进,以适应我国的 SWIM 网络建设。

## 参考文献

- [1] CRESCENZO D D, STRANO A, TRAUSMUTH G. SWIM: A Next Generation ATM Information Bus-The SWIM-SUIT Prototype[C]//Proceedings of the 14th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW). Brazil; IEEE, 2010: 41-46.
- [2] DELOSIERES L, NADJM-TEHRANI S. Batman Store and Forward: The Best of the Two Worlds[C]// Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops'12). Lugano; IEEE, 2012: 721-727.
- [3] 郑黎明, 邹鹏, 张建锋, 等. 面向大规模网络的安全态势实时量化感知模型[J]. 计算机科学, 2011, 38(Z10): 30-35.
- [4] KIEN N, QUANG T M, YAMADA S. A Software-Defined Networking Approach for Disaster-Resilient[C]// Proceedings of the 22nd International Conference on Computer Communications and Networks (ICCCN). Nassau; IEEE, 2013: 1-5.
- [5] GU Y, WANG C Y. DR-Cloud: Multi-cloud Based Disaster Recovery Service[J]. Tsinghua Science and Technology, 2014, 19(1): 13-23.
- [6] OLIVEIRA S, OLIVEIRA T R, NOGUEIRA J M S. A Policy based Security Management Architecture for Sensor Networks[C]// Proceedings of IFIP/IEEE International Symposium on Integrated Network Management. Long Island; IEEE, 2009: 315-318.
- [7] 李明磊, 王红卫, 祁超, 等. 非常规突发事件应急决策方法研究[J]. 中国安全科学学报, 2012, 22(3): 158-163.
- [8] AGARWAL P, et al. The Resilience of WDM Networks to Probabilistic Geographical Failures[C]// Proceedings of 2011 IEEE INFOCOM. Shanghai; IEEE, 2011: 1521-1529.
- [9] RAHNAMAY-NAEINI M, et al. Modeling Stochastic Correlated Failures and Their Effects on Network Reliability[C]// Proceedings of the 20th International Conference on Computer Communications and Networks (ICCCN). Maui; IEEE, 2011: 1-6.
- [10] RUAN L, FENG T. A Hybrid Protection/Restoration Scheme for Two Link Failure in WDM Mesh Networks[C]// Proceedings of Global Telecommunications Conference (GLOBECOM 2010). Miami; IEEE, 2010: 1-5.
- [11] CULLY B, LEFEBVRE G, MEYER D, et al. Remus: High Availability via Asynchronous Virtual Machine Replication[C]// Proceedings of USENIX Symposium on Networked Systems Design and Implementation (NSDI'08). San Francisco; IEEE, 2008.
- [12] 杨昌武. 基于 LVS 集群负载均衡技术研究与应用[D]. 重庆: 重庆邮电大学, 2013.
- [13] RAFFELSBERGER C, HELLWAGNER H. Evaluation of MANET Routing Protocols in A Realistic Emergency Response Scenario[C]// Proceedings of the 10th Workshop on Intelligent Solutions in Embedded Systems (WISES). Klagenfurt; IEEE, 2012: 88-92.
- [14] 谭浩宇. 多虚拟机管理平台中的监控系统[D]. 武汉: 华中科技大学, 2008.
- [15] ZHANG Y, BEHESHTI N, TATIPAMULA M. On Resilience of Split-architecture Networks[C]// Proceedings of Global Telecommunications Conference (GLOBECOM 2011). Kathmandu; IEEE, 2011: 1-6.
- [16] STERBENZ J P G, HUTCHISON D, C E K, et al. Resilience and Survivability in Communication Networks: Strategies, Principles, and Survey of Disciplines[J]. Computer Network, 2010, 54(8): 1245-1265.
- [17] ZHANG J, ZHANG N. Cloud Computing-based Data Storage and Disaster Recovery[C]// Proceedings of International Conference on Future Computer Science and Education (ICFCSE). Xi'an; IEEE, 2011: 629-632.