



应用服务器群服务安全监控机制

鞠洪尧

(浙江纺织服装职业技术学院, 浙江 宁波 315211)

摘要: 应用服务器关键硬件的承载状态及应用软件自身的工作状态是网络应用系统服务安全的重要因素, 通过对应用服务器中央处理器、内存、网络流量、应用软件目录、文件等监控技术的研究, 将动态综合监控、安全审查、应用软件智能修复技术进行集成, 构建了网络应用系统的智能监控体系架构, 并阐明了实现原理和方法, 解决了网络应用系统软硬件工作状态综合监控问题, 实现了应用系统硬件和软件工作状态异常时自动报警、定位和应用软件遭受攻击时自动修复的目标。

关键词: 应用服务器群; 动态监控; 安全审查; 异常定位; 智能恢复

中图分类号: TP393.2

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2016173

Service security monitoring mechanism for application server cluster

JU Hongyao

Zhejiang Textile & Fashion College, Ningbo 315211, China

Abstract: For an application system, the workloads on the application servers' key hardware and the working state of the application software are important factors affecting the system's safety. Based on studies of the monitoring techniques for CPU, RAM, network traffic, software directories and software files of application servers, dynamic comprehensive monitoring, safety review and application software intelligent recovery were integrated, and an intelligent monitoring system was constructed for the application systems. The mechanisms and methodology of the monitoring system were illustrated. The system realized intelligent monitoring the working state of hardware and software, achieved the goal of automatic alarming, fault locating upon abnormal working states of the hardware or software and automatic recovery upon attacks to the software.

Key words: application server cluster, dynamic monitoring, security review, anomaly locating, intelligent recovery

1 引言

承载大数据服务的网络应用系统主要由应用服务器硬件和应用软件构成, 硬件是应用软件的运行载体, 应用

软件为网络用户提供应用服务。在过去的几年中, 大数据支撑网络的系统硬件工作安全和应用服务安全问题日益突出, 全球范围多次发生机场空管中心应用系统、护照/签证系统瘫痪的状况, 这些承载大数据服务的网络应

收稿日期: 2016-04-20; 修回日期: 2016-06-15

基金项目: 科技部创新基金资助项目 (No.14C26213311599); 宁波市智团创业基金资助项目 (No.2013B70071)

Foundation Items: The Innovation Fund for Technology Based Firms of the Chinese Ministry of Science and Technology (No.14C26213311599), Ningbo Intelligence Group Entrepreneurship Project (No.2013B70071)



用系统在发生故障时没有给出预警信息,发生故障后系统不能进行自我修复及故障定位,使得系统故障的解决非常困难。

目前,应用服务器硬件负载监测的研究大多通过 CPU 使用率、内存使用率及网络带宽占用率等指标来实现,其缺陷是 CPU 的占用率在短时间出现峰值时会造成假过载问题。而对应用服务器中应用软件的监测研究大多通过通信协议分析来实现攻击和病毒代码的判定,其缺陷是定位遭受攻击的具体位置及自动修复应用软件比较困难,且当应用服务器群中成员服务器数量较多时会额外占用大量的网络带宽,导致通信效率大大降低。

在参阅参考文献[1-5]及总结实验验证数据的基础上,对大数据支撑网络应用服务器的 CPU 平均工作负载、内存的使用率、网卡接口带宽占用率 3 个关键硬件指标值进行了监测,并分别设定了上述 3 个监测指标的工作负载阈值。当某个监测指标工作负载值超过阈值时,系统自动发出预警提示,并输出过载硬件的信息。同时,对应用服务器上的应用软件的工作目录、目录数量、目录中文件的数量及每个文件的长度进行了动态监测,并将实际监测值与应用软件原始的目录名称、目录数量、目录中文件的数量及每个文件的长度值进行实时比对。当对比值出现不一致时,系统自动发出预警提示,并输出应用软件代码发生改变的位置信息,同时,利用应用软件的原始备份自动覆盖应用软件的所有代码。

依据上述原则,构建了大数据支撑网络应用系统智能监控模型,该模型重点解决了如下问题。

- 应用服务器关键硬件过载及故障时给出预警和定位信息。
- 对 CPU 平均工作负载进行监测,即监测 CPU 队列中排队任务的个数,避免了由于 CPU 占用率在短时间出现峰值而造成的假过载问题。
- 应用软件遭受攻击发生改变时,给出预警和定位信息,并实现应用软件的自动修复。
- 应用软件状态监测采用目录扫描及文本比较的方法实现,产生的额外网络流量非常小,特别适用于应用服务器群中成员数量较多的网络。

2 智能监控模型架构及实现机制

2.1 模型设计原则与架构

模型包含应用服务器集群、智能监控系统 and 数据交换

系统 3 个部分。其中,应用服务器集群中每个应用服务器都安装相同的应用软件,并配置了应用软件原始代码池 and 软件整体更新触发器。智能监控系统包含应用服务器集群成员特征信息数据库、应用服务器集群硬件工作状态监控器、应用软件状态监控器、应用软件修复控制器和负载调度控制器。数据交换系统负责智能监控系统与应用服务器集群间的通信。

根据上述设计原则,大数据支撑网络的智能监控系统模型架构如图 1 所示。

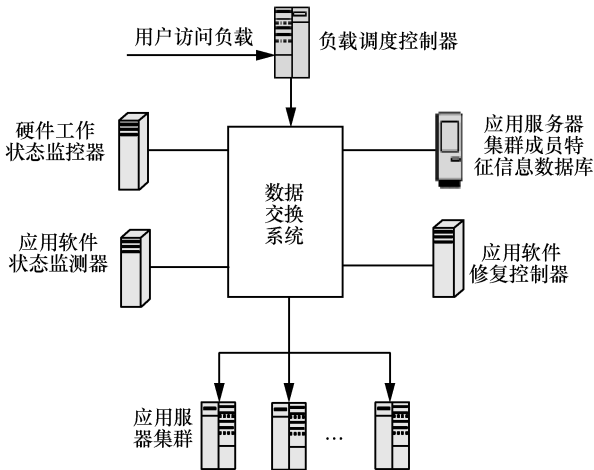


图 1 大数据支撑网络的智能监控系统模型架构

2.2 两层监控结构及实现机制

第一层监测与安全结构设置在应用软件运行环境层面,由硬件工作状态监测器和负载调度控制器来实现。硬件工作状态监测器监测对象包括应用服务器的 CPU 平均工作负载、内存使用率和网络带宽占用率 3 个指标,将监测结果与每个监测值的阈值进行比对,根据比对结果确定硬件是否过载及过载点位置,输出预警信息。负载调度控制器依据监测指标的工作负载情况动态调整分配给集群中每个成员服务器的工作负载。第一层监测与安全结构关系如图 2 所示。

第二层监控与安全结构设置在应用软件层面,由应用软件状态监测器及应用软件修复控制器来实现。应用软件状态监测器的监控对象包括应用软件目录名称、目录数量、目录中的文件名称、文件数量及文件长度 5 个指标,并根据这 5 个指标的变化信息确定发生变化对象的位置,然后输出预警信息,并通过应用软件修复控制器将应用软件恢复到原始状态。第二层监控与安全结构关系如图 3 所示。

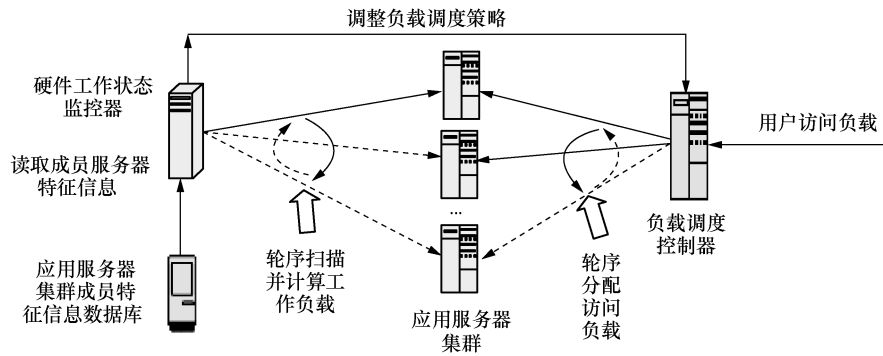


图 2 应用服务器硬件状态监测与负载调度机制

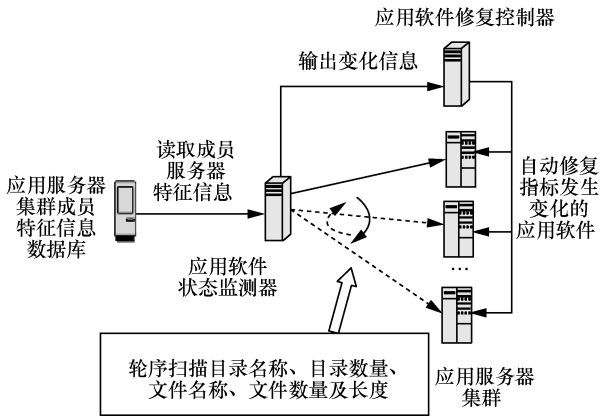


图 3 应用软件状态监测与修复机制

3 硬件工作状态监控系统工作过程及关键技术

硬件工作状态监控系统从应用服务器特征信息数据库中依次读取每个应用服务器的名称 $NAME_{computer}$ 、IP 地址信息 $IP_{computer}$ 和 CPU 核心数量值 N_{CPU} ，并将这些值传递给硬件工作状态监测器，为它们提供工作参数，具体工作过程如图 4 所示。

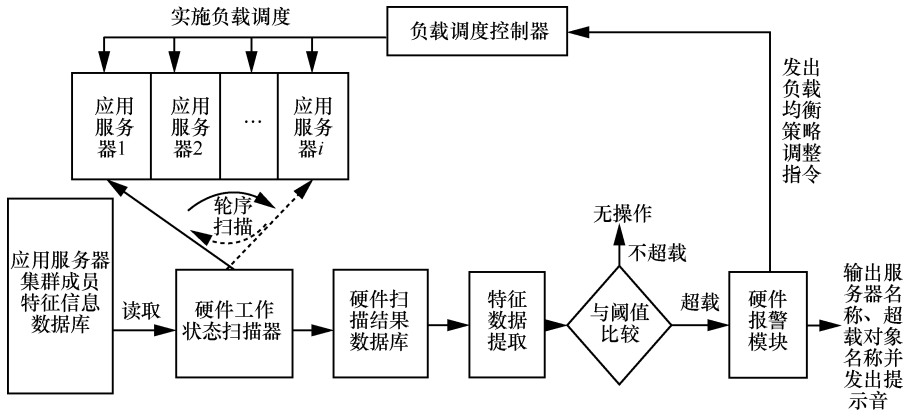


图 4 硬件工作状态监控系统工作过程

3.1 应用服务器特征信息数据库

应用服务器特征信息数据库包含被监测的大数据支撑网络系统中每个应用服务器使用的 IP 地址信息和 CPU 核心数量信息，其中，IP 地址为硬件工作状态扫描器和软件状态扫描器提供的目标信息。CPU 核心数量值是每个 CPU 核心的负载值。应用服务器特征信息数据库结构见表 1。

表 1 服务器特征信息数据库结构字段及含义

服务器名称	绑定的 IP 地址	CPU 核心数量值
$NAME_{computer}$	$IP_{computer}$	N_{CPU}

3.2 硬件工作状态监测器

硬件工作状态监测器^[6-11]负责应用系统运行环境关键硬件负载状况的实时监测，该系统包含硬件工作状态扫描器、硬件扫描结果数据库、特征数据提取模块、数据分析模块和硬件报警模块 5 个部分。

3.2.1 硬件工作状态扫描器

硬件工作状态扫描器包含 CPU 状态扫描模块、内存状态扫描模块、网络接口状态扫描模块 3 个部分。



(1) CPU 状态扫描模块

该模块根据智能监测系统提供的 IP 地址信息 IP_{computer} , 实时监测 IP_{computer} 指定应用服务器 CPU 任务队列的长度, 获得应用服务器 1 min、5 min、15 min 内 CPU 的平均负载值, 将其写入硬件扫描结果数据库中。

(2) 内存状态扫描模块

该模块根据智能监测系统提供的 IP 地址信息 IP_{computer} , 实时监测 IP_{computer} 指定应用服务器内存的使用量和总量值, 并将这两个值追加写入硬件扫描结果数据库中。

(3) 网络接口状态扫描模块

该模块根据智能监测系统提供的 IP 地址信息 IP_{computer} , 实时监测 IP_{computer} 指定应用服务器网卡接口的数据流量、传输速率值和总带宽值, 并将这 3 个值追加写入硬件扫描结果数据库中。

3.2.2 硬件扫描结果数据库

该数据库中记录了 IP_{computer} 指定应用服务器的信息如下:

- CPU 在 1 min、5 min 和 15 min 内的平均负载总值;
- 内存使用量值和总量值;
- 网卡接口数据流量值、数据传输速率值和带宽值。

3.2.3 特征数据提取模块

该模块从硬件扫描结果数据库中读取 CPU 在指定监测周期内的平均负载值、内存使用量值、内存总量值、网卡接口数据流量值、网络传输速率值和网卡接口带宽值, 传递给数据分析模块进行分析处理。

3.2.4 数据分析模块

该模块根据特征数据提取模块传递过来的特征数据, 计算应用服务器 CPU 每个核心的工作负载、内存使用率和网络带宽占用率, 并判断是否过载, 同时输出是否过载的结果信息, 具体计算及判定方法如下。

(1) CPU 工作负载计算及过载判定方法

$$F_{\text{CPU}} = F_{\text{CPU-average}} / N_{\text{CPU}} \quad (1)$$

其中, F_{CPU} 表示 1 个 CPU 核心的工作负载值, $F_{\text{CPU-average}}$ 表示实时监测获得的应用服务器 CPU 平均工作负载总值, N_{CPU} 表示被监测的应用服务器 CPU 的核心数量值。

CPU 每个核心的工作负载值与其临界值的判定关系见表 2。

其中, M_{CPU1} 表示 CPU 正常工作负载临界值, 在实际工作中, M_{CPU1} 通常取 3。 M_{CPU2} 表示 CPU 超载临界值, 在实际

表 2 CPU 工作负载状态判定关系

CPU 负载判定关系	判定结果	输出值 R_{CPU}
$F_{\text{CPU}} < M_{\text{CPU1}}$	CPU 工作正常	0
$M_{\text{CPU1}} \leq F_{\text{CPU}} < M_{\text{CPU2}}$	CPU 工作繁忙	1
$F_{\text{CPU}} \geq M_{\text{CPU2}}$	CPU 工作超载	2

工作中, M_{CPU2} 通常取 5。

(2) 内存使用率计算及过载判定方法

$$F_{\text{RAM}} = G_{\text{RAM}} / G_0 \quad (2)$$

其中, F_{RAM} 表示应用服务器内存的使用率, G_{RAM} 表示内存使用量, G_0 表示被监测的应用服务器的内存总量值。

内存工作负载值与其临界值的判定关系见表 3。

表 3 内存工作负载状态判定关系

内存负载判定关系	判定结果	输出值 R_{RAM}
$F_{\text{RAM}} < M_{\text{RAM1}}$	内存负载正常	0
$M_{\text{RAM1}} \leq F_{\text{RAM}} < M_{\text{RAM2}}$	内存工作繁忙	1
$F_{\text{RAM}} \geq M_{\text{RAM2}}$	内存超载	2

其中, M_{RAM1} 表示内存正常工作负载的临界值, 在实际工作中, M_{RAM1} 通常取 60%。 M_{RAM2} 表示内存超载临界值, 在实际工作中, M_{RAM2} 通常取 85%。

(3) 网络带宽占用率计算及过载判定方法

$$F_{\text{net}} = L_{\text{net}} / L_0 \quad (3)$$

其中, F_{net} 表示应用服务器网络带宽使用率, L_{net} 表示网络的实际数据传输速率值, L_0 表示被监测应用服务器的网络带宽值。

网络负载值与其临界值的判定关系见表 4。

表 4 网络负载状态判定关系

网络负载判定关系	判定结果	输出值 R_{net}
$F_{\text{net}} < M_{\text{net1}}$	网络负载正常	0
$M_{\text{net1}} \leq F_{\text{net}} < M_{\text{net2}}$	网络工作繁忙	1
$F_{\text{net}} \geq M_{\text{net2}}$	网络超载	2

其中, M_{net1} 表示网络正常工作流量负载临界值, 在实际工作中, M_{net1} 通常取 70%。 M_{net2} 表示网络流量超载临界值, 在实际工作中, M_{net2} 通常取 90%。

3.2.5 硬件报警模块

该模块接收来自数据分析模块的 CPU、内存及网络负载状态结果输出值 R_{CPU} 、 R_{RAM} 和 R_{net} , 并根据接收到的 R_{CPU} 、 R_{RAM} 和 R_{net} 值将应用服务器名称 $NAME_{\text{computer}}$ 、 R_{CPU} 、 R_{RAM} 和 R_{net} 值显示在智能监控服务器屏幕上。当 R_{CPU} 、 R_{RAM} 或 R_{net} 等于 1 或 2 时, 硬件报警模块将发出不同的报警提示音;

当 R_{CPU} 、 R_{RAM} 和 R_{net} 等于 0 时,硬件报警模块不发出报警提示音。

4 应用软件工作状态监控系统工作过程及关键技术

应用软件工作状态监控系统主要包含应用服务器特征信息数据库、软件工作状态监测器^[12-16]、应用软件修复控制器和应用软件原始代码池等组件。其中,软件工作状态监测器负责服务器中应用软件运行状态的实时监测,该组件包含软件工作状态扫描器、软件扫描结果数据库、与原始库对比处理模块、比对结果数据库和智能模块 5 个部分。具体工作过程如图 5 所示。

4.1 应用软件状态扫描器

软件工作状态扫描器通过目录操作指令实时扫描应用服务器中处于工作目录中应用软件的子目录和文件的名称及长度信息,将扫描结果以覆盖方式写入软件扫描结果数据库中。具体过程如图 6 所示。

其中,1,2,⋯,k 代表某一个应用服务器应用软件主目录下的第 1,2,⋯,k 个子目录。

4.2 软件扫描结果数据库

该数据库是一个文本文件,其中保存有使用指定 IP 地址 IP_{computer} 的应用服务器中应用软件的子目录名称、目录下的文件名称以及它们的长度值,具体信息见表 5。

表 5 扫描结果数据库记录信息

对象名称	扫描记录信息
应用软件主目录	主目录名称、子目录名称、数量及容量值,文件的名称、数量及长度值
子目录 1 及下级目录和文件	子目录名称、数量及容量值,文件的名称、数量及长度值
子目录 2 及下级目录和文件	子目录名称、数量及容量值,文件的名称、数量及长度值
⋯	子目录名称、数量及容量值,文件的名称、数量及长度值
子目录 k 及下级目录和文件	子目录名称、数量及容量值,文件的名称、数量及长度值

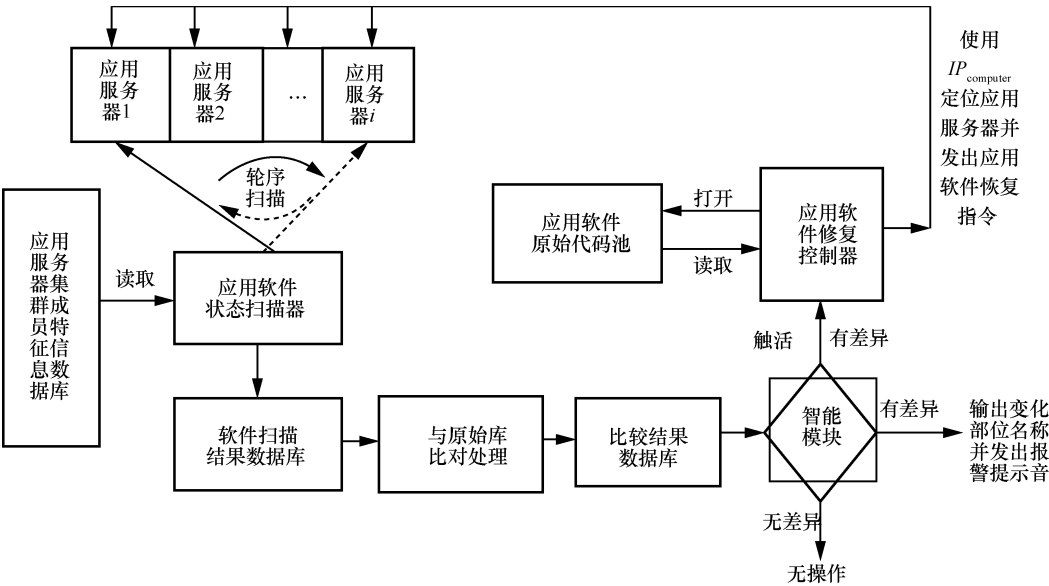


图 5 应用软件工作状态监控与修复工作过程

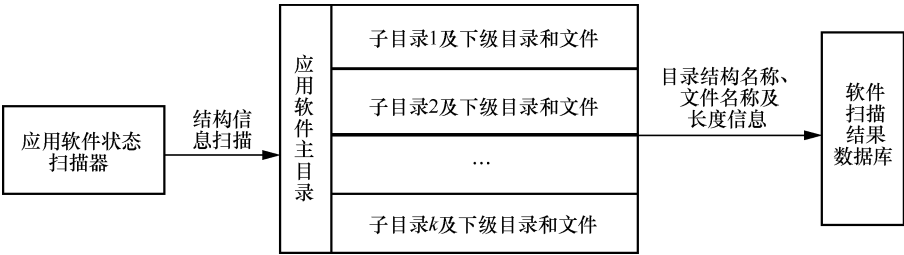


图 6 应用软件状态扫描器工作过程



4.3 与原始库对比处理模块

该模块通过文本文件比较的方式将软件扫描结果数据库与应用软件原始代码池中应用软件的扫描文件进行对比分析,若比较的记录信息存在差异,则将存在差异的应用服务器的IP地址、差异目录或文件的名称以覆盖方式写入比对结果数据库。若比较的记录信息不存在差异,则不向比对结果数据库中写入数据。数据对比过程如图7所示。

4.4 比对结果数据库

该数据库保存了软件扫描结果数据库与应用软件原始代码池中应用软件的扫描文件进行对比的结果。如果两个比对文件不存在差异则该数据库长度为0,如果两个比对文件存在差异则该数据库中保存着差异部分的名称信息。具体信息见表6。

表6 比对结果数据库记录信息

数据库记录	说明
长度值为0	应用软件代码与原始代码一致,应用软件状态正常
长度值不为0,且包含发生变化的目录名称、文件名称和所在应用服务器的IP地址	应用软件代码与原始代码不一致,应用软件状态不正常

4.5 智能模块

该模块首先打开比对结果数据库,若比对结果数据库长度是0,表明软件扫描结果数据库与应用软件原始代码

池中应用软件的扫描文件一致,应用软件工作状态正常;若比对结果数据库长度不是0,表明软件扫描结果数据库与应用软件原始代码池中应用软件的扫描文件不一致,应用软件代码已被篡改或已遭受攻击,读取比对结果数据库中的第一个记录,并将该记录信息与 $IP_{computer}$ 值对应的应用服务器的名称一起输出到监控服务器屏幕上,并发出报警提示音。同时激活应用软件修复控制器。此处只读取第一个记录的原因在于:由于采用文本文件整体比对,当应用软件工作目录中子目录或文件信息发生变化,后续的信息记录比对会连续出现不匹配的情况,因此,只读取首个出现不同对象的信息即可。

4.6 应用软件修复控制器

该控制器接收来自智能模块的激活信息和 $IP_{computer}$ 值,使用 $IP_{computer}$ 值定位应用服务器集群中的服务器,启动应用软件整体更新触发器。首先强制删除应用服务器应用软件工作目录中的所有子目录和文件,然后将应用软件原始代码池中子目录和文件复制到应用服务器应用软件工作目录中,使得网络用户在后续的访问中及时使用正确的应用软件代码,获得正确的访问数据,实现应用软件代码的自动复原过程,具体工作过程如图8所示。

4.7 应用软件原始代码池

应用软件原始代码池位于应用服务器中,其中保存着应用软件完整的原始代码备份。当工作目录中的代码遭受攻击,被修改时,软件整体更新触发器使用这个备份恢复

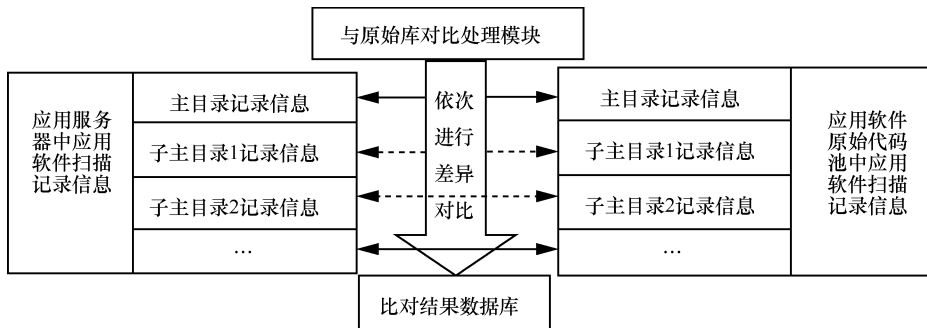


图7 数据对比过程

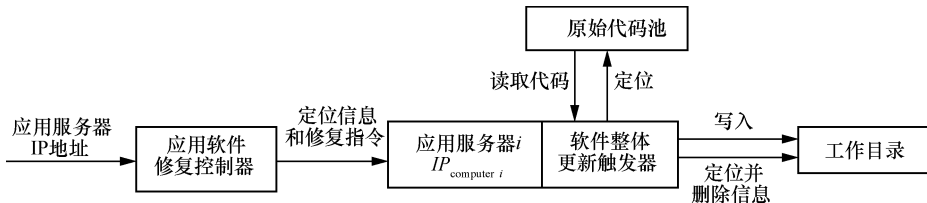


图8 应用软件自动修复过程

应用软件到原始状态。

5 测试环境、方法及结果

5.1 测试环境及阈值参数设置

在大数据支撑网络应用系统智能监控模型测试过程中,选择了 2 台 Linux 6 服务器构建了一个小规模的应用服务器集群,每台应用服务器都安装 Web 应用软件,其中,服务器 1 含有 1 颗 4 核心的 CPU、16 GB 内存和 100 Mbit/s 网络连接,服务器 2 含有 1 颗 2 核心的 CPU、4 GB 内存和 100 Mbit/s 网络连接。智能监控服务器采用 Linux 6 系统。系统中配置了 1 台安装 Windows Server 2012 操作系统的 DNS 服务器,并配置了负载均衡策略。测试终端使用了 5 台安装 Windows 7 操作系统的计算机。监测周期为 5 min。测试过程中使用的阈值见表 7。

5.2 测试方法及结果

5.2.1 测试硬件工作状态监测器

每个终端持续打开多个浏览器副本,访问应用服务器 1

表 7 系统测试时使用的阈值参数

应用服务器 1		应用服务器 2	
对象名称	参数值	对象名称	参数值
M_{CPU1}	0.07	M_{CPU1}	0.14
M_{CPU2}	0.14	M_{CPU2}	0.28
M_{RAM1}	13%	M_{RAM1}	44%
M_{RAM2}	15%	M_{RAM2}	52%
M_{net1}	0.15%	M_{net1}	0.6%
M_{net2}	1%	M_{net2}	1.2%
N_{CPU}	4	N_{CPU}	2
G_0	16	G_0	4
N_0	100	N_0	100

和服务器 2,同时监测并获取应用服务器 1 和应用服务器 2 的 CPU、内存及网络带宽占用率数值的变化情况。监控服务器实测获得应用服务器 1 和应用服务器 2 的 CPU、内存及网络带宽占用率数值的变化及报警点出现情况如图 9 所示。

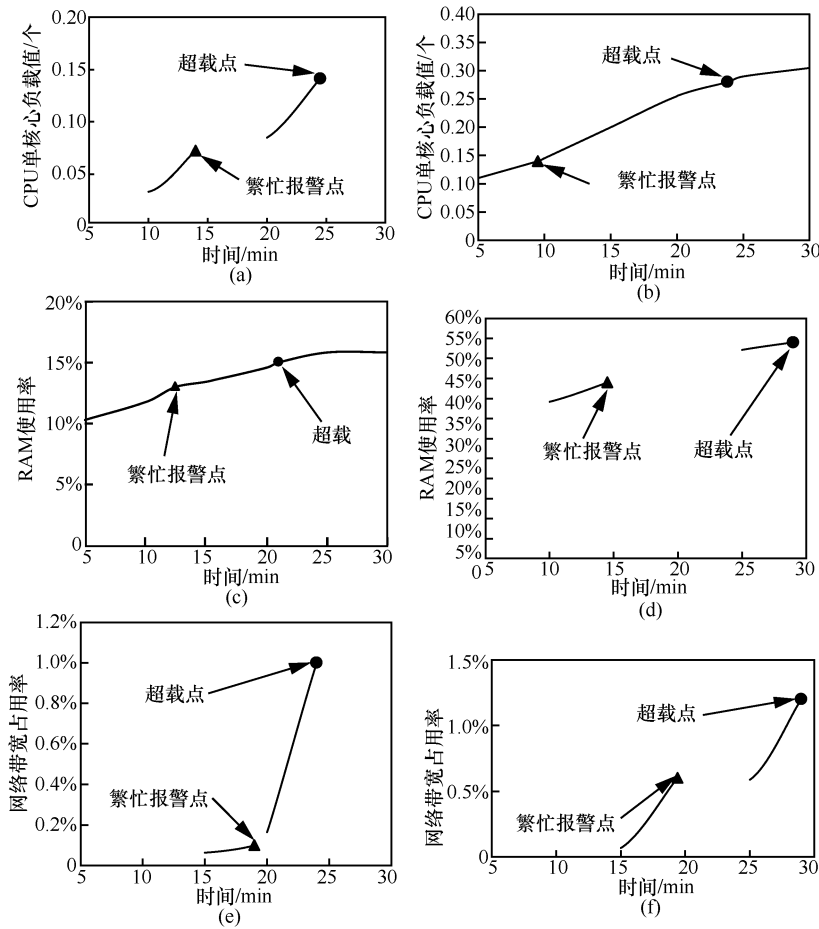


图 9 服务器的 CPU、内存及网络带宽占用率数值的变化及报警点



其中,图 9(a)、图 9(c)、图 9(e)为应用服务器 1 硬件监测指标的变化关系,图 9(b)、图 9(d)、图 9(f)为应用服务器 2 硬件监测指标的变化情况。

5.2.2 测试软件工作状态监测器及修复控制器

通过任意终端登录应用服务器 1 和应用服务器 2,修改应用软件子目录或文件,同时使用终端访问应用服务器上的应用软件,观察访问终端浏览器显示信息、智能监控服务器屏幕状态信息及应用服务器目录内容变化情况。测试结果见表 8。

6 测试结果分析

6.1 应用软件工作支撑环境测试结果分析

从图 9 中可以看出:在图 9(a)中接近 15 min 和在图 9(b)中接近 10 min 时,应用服务器 1、2 的 CPU 负载进入测试设定的工作繁忙状态区间,同时,智能监控服务器提示了 CPU 工作繁忙预警信息。在图 9(a)中接近 25 min 时和在图 9(b)中接近 24 min 时,应用服务器 1、2 的 CPU 负载进入测试设定的工作超载状态区间,同时,智能监控服务器同步提示了 CPU 超载报警信息。

在图 9(c)和图 9(d)中接近 15 min 时,内存负载进入测试设定的工作繁忙状态区间,同时,智能监控服务器提示了内存繁忙预警信息。在图 9(c)接近 23 min 和图 9(d)接近 29 min 时,内存负载进入测试设定的工作超载状态区间,同时,智能监控服务器同步提示了内存超载报警信息。

在图 9(e)和图 9(f)中接近 20 min 时,网络负载进入测试设定的工作繁忙状态区间,同时,智能监控服务器提示了网络繁忙预警信息。在图 9(e)接近 24 min 和图 9(f)接近 29 min 时,网络负载进入测试设定的工作超载状态区间,同时,智能监控服务器同步提示了网络超载报警信息。

上述测试结果表明:本文构建的智能监控模型,硬件工作状态监控部分已达到预期的目标,在应用服务器关键硬件负载过重时实现自动预警和故障定位。

6.2 应用软件工作状态测试结果分析

从表 8 中可以看出:当应用软件主目录或子目录中增加目录或文件时,在实施增加操作的同一个监测周期内,软件工作状态监测器无响应,终端的相关访问受到影响。在超过同一个监测周期后,软件工作状态监测器立即响应,并激发修复控制器对应用软件主目录中的所有内容进行恢复,同时将新增加的目录或文件名称输出到智能监控服务器屏幕上,终端的相关访问正常。

上述测试结果表明,本文构建的智能监控模型、软件工作状态监测器及修复控制器实现了在应用服务器的应用软件遭受攻击时进行自动预警、故障定位和自我修复。

7 结束语

本文从大数据网络应用软件的硬件支撑环境工作状态监测和应用软件工作状态监测两个方面探讨了应用服务系统的故障监测、故障定位及自动修复机制。在现有研究成果的基础上对运营和维护的实用性、便捷性做了改进,降低了智能监控系统在监测过程中对应用服务器 CPU、内存及网络造成的额外工作负载,解决了大数据网络应用服务器群关键硬件及应用软件工作状态的并行监控、故障预警、故障定位及自动修复问题。

参考文献:

- [1] 王建民,尹建伟,窦万春.业务过程管理技术专题前言[J].软件学报,2015,26(3):447-448.
WANG J M, YIN J W, DOU W C. Business process management technology thematic introduction [J]. Journal of Software, 2015, 26(3): 447-448.
- [2] 孟小峰,慈祥.大数据管理:概念、技术与挑战[J].计算机研究与发展,2013,50(1):146-169.
MENG X F, CI X. Big data management: concepts, techniques and challenges [J]. Journal of Computer Research and Development, 2013, 50(1): 146-169.

表 8 软件工作状态监测及修复控制器测试结果

测试操作项目	1 个监测周期内			1 个监测周期后		
	智能监控服务器显示信息	终端浏览器显示信息	服务器应用软件目录内容	智能监控服务器显示信息	终端浏览器显示信息	服务器应用软件目录内容
新建 1 个目录	无	正常	保持修改后状态	新建目录名称	正常	新建目录消失
修改 1 个已有目录名称	无	相关访问出错	保持修改后状态	修改目录名称	正常	修改目录被恢复原状
新建 1 个文件	无	正常	保持修改后状态	新建文件名称	正常	新建文件消失
修改 1 个已有文件	无	相关访问出错	保持修改后状态	修改文件名称	正常	修改文件被恢复原状

- [3] 王宇. 自主可控系统安全防护技术的现状与对策[J]. 保密科学技术, 2014(12): 24-30.
WANG Y. Current situation and countermeasures for independent and controllable system security technology [J]. Secrecy Science and Technology, 2014(12): 24-30.
- [4] 袁健, 陈冬露. 云环境中基于动态监测周期的动态资源管理模型[J]. 小型微型计算机系统, 2016, 37(3): 474-478.
YUAN J, CHEN D L. Resource management strategic model based on dynamic monitoring period in cloud environment [J]. Journal of Chinese Computer Systems, 2016, 37(3): 474-478.
- [5] SHI J, LI H, ZHOU L D. The technical security issues in cloud computing. International Journal of Information and Communication Technology [J]. 2013, 5(3-4): 272-282.
- [6] NWOBODO I, JAHANKHANI H, EDOH A. Security challenges in the distributed cloud computing [J]. International Journal of Electronic Security and Digital Forensics, 2014, 6(1): 38-51.
- [7] CHEN G, WU S, WANG Y. The evolvement of big data systems: from the perspective of an information security application [J]. Big Data Research, 2015, 2(2): 65-73.
- [8] 王帅, 汪来富, 金华敏, 等. 网络安全分析中的大数据技术应用[J]. 电信科学, 2015, 31(7): 139-144.
WANG S, WANG L F, JIN H M, et al. Big data application in network security analysis [J]. Telecommunications Science, 2015, 31(7): 139-144.
- [9] 胡小梅, 刘嘉勇. 基于 DNS 劫持的流量监测系统设计与实现[J]. 网络安全技术与应用, 2016(1): 110-112.
HU X M, LIU J Y. Design and implementation of traffic monitoring system based on DNS [J]. Network Security Technology & Application, 2016(1): 110-112.
- [10] MOHAIMENUZZAMAN M, RAHMAN S M M. A design pattern for real time progress monitoring of a time-intensive process in a stateless software application [J]. International Journal of Software Engineering, Technology and Applications, 2015, 1(1): 53-63.
- [11] CASOLA V, BENEDICTIS A D, DRAGO A, et al. SeNsiM-SEC: secure sensor networks integration to monitor rail freight transport [J]. International Journal of System of Systems Engineering, 2013, 4(4): 291-316.
- [12] 王一川, 马建峰, 卢笛, 等. 面向云环境内部 DDoS 攻击检测的博弈论优化[J]. 计算机研究与发展, 2015, 52(8): 1873-1882.
WANG Y C, MA J F, LU D, et al. Game optimization for internal DDoS attack detection in cloud computing [J]. Journal of Computer Research and Development, 2015, 52(8): 1873-1882.
- [13] KIM S K. JenniferSoft: APM software development leader in Korea [J]. International Journal of Teaching and Case Studies, 2015, 6(2): 95-107.
- [14] JAIN M, PREETI. Availability analysis of software rejuvenation in active/standby cluster system [J]. International Journal of Industrial and Systems Engineering, 2015, 19(1): 75-93.
- [15] GAUR V, SONI A. A knowledge-driven approach for specifying the requirements of multi-agent system [J]. International Journal of Business Information Systems, 2015, 19(3): 300-323.
- [16] CAROL S S, SHI Y. Predicting residual software fault content and their location during multi-phase functional testing using test coverage [J]. International Journal of Reliability and Safety, 2013, 7(1): 32-57.

[作者简介]



鞠洪尧 (1966-), 男, 浙江纺织服装职业技术学院教授, 主要研究方向为网络性能优化、云计算、智能制造等。