

文章编号:1674-5949(2016)03-0072-07

数据中心自动化运维平台的设计与实现

王骏翔

(中海网络科技股份有限公司,上海 200135)

摘 要:针对传统人工运维方式已无法满足业务发展需求的问题,借助自动化运维技术管理海量服务器,并建设系统运维云,提升数据中心的响应速度和管理效率。分析了数据中心自动化运维平台的层次结构,阐述了自动化运维平台的建设方法。

关键词: 自动化运维;Python 语言;Shell 语言

中图分类号: TP311.52 **文献标志码:** A

Design and Implementation of Data Center Automatic Operation and Maintenance Platform

WANG Junxiang

(China Shipping Network Technology Co., Ltd., Shanghai 200135, China)

Abstract: To address the growing workload of the staff of the data center an automatic operation and maintenance platform for the data center is developed. The platform also improves the efficiency and availability of the data services. This paper introduces the hierarchical structure of the platform and the process of the development.

Key words: automatic operation and maintain; Python; Shell

0 引言

随着大型集团企业信息系统建设不断发展,数据中心内的服务器、存储设备、网络设备和数据库服务器等设备的种类和数量与日俱增,IT 软、硬件承担的责任越来越重。这对企业信息化部门的系统安全运营和维护管理工作提出了更高的要求,因此迫切需要建设综合性、自动化的综合运维管理系统对其进行支撑。

自动化运维管理平台是一个一体化、综合性的日常运维、监管、监控和调度平台,其功能涉及数据中心日常巡检、企业设备资产管理、服务器生命周期管理、应用服务可用性监控、数据库服务可用性监控、本地/异地数据库同步监控和信息化部门 IT 知识库管理等,为企业各信息系统正常运行和日常监管提供有力支撑,提高运行效率和服务质量,降低运营成本。

利用面向对象的设计思想,结合跨平台、多语言的研发手段,提出一种适用于多行业信息部门的综合自动化运维管理平台。对系统设计目标进行阐述,并对架构和功能进行详细设计。将该平台应用于某大型航运企业,可较好地解决信息调度管理分散、信息化设备资产管理缺失和数据库本地/异地实时同步专业化监控繁琐等问题,为类似工程提供建设思路。

1 设计目标及原则

1.1 设计目标

自动化运维即在最少的人工干预下,结合运用脚本和第三方工具,保证业务系统 7×24 h 高效、稳定地

收稿日期:2016-06-02

作者简介:王骏翔(1987—),男,上海人,从事系统运维工作。

运行。系统设计目标可分解为以下 3 个层次。

1.1.1 化繁为简

随着企业信息系统建设的需求增多,信息化部门分管的 IT 设备、软件种类和业务场景均呈现出以下几个特点:

- (1) 设备种类繁多、管理分散;
- (2) 运维涉及专业技术性强,难普及;
- (3) 场景多、重复性强;
- (4) 手工运维易遗漏。

1.1.2 数据的实时性

- (1) 实时掌握信息设备和信息系统的运行情况;
- (2) 及时发现故障及异常,并迅速定位问题原因。

1.1.3 数据的可靠性

- (1) 通过变更流程管理,保证设备生命周期管理的数据准确性;
- (2) 通过日常巡检单管理,保证数据中心设备健康状况的准确性;
- (3) 通过实时数据抓取和校验底层监控对象,保证监控对象数据的准确性。

1.2 设计原则

1) 面向对象设计思路^[1]。面向对象是目前普遍采用的软件开发设计思路,目的是清晰、精确地定义问题领域,以类和继承为构造机制,充分利用接口和多态提供灵活性来认识、理解、刻画客观世界,设计和构建相应的软件系统。

对于该系统而言,面向对象的设计思路主要体现在模块的功能设计上。传统的运维相对呈现出多样化、零散化和独立化的特征,系统运维人员常常会尝试通过编写一些零散、简易的脚本实现日常的工作,但其很难具有普及性和实际批量推广的效用。该系统在整体架构和功能设计部分充分结合实际运维中普遍存在的难点和重复性进行突破,并以面向对象的设计原则逐步实现和调试,以满足大型企业运维团队的需求。

2) 充分利用现有的产品及技术,在多平台、多产品种类的场景下,以简化繁琐、重复、复杂的日常运维工作为核心,实现企业级一体化的自动化运维管理,从管理域的角度可分为信息中心(部门新闻、公告)、日常巡检(数据中心巡检、系统运维巡检、备份自动化巡检)、运维监控(应用可用性、数据库可用性、本地/异地数据库同步)、设备管理(物理服务器、虚拟化设备、网络设备)、知识库管理和第三方平台接入。

系统设计不仅要汲取现有的技术和运维经验,还要兼顾已有系统的使用情况,在现有产品和技术的基础上创建自动化综合运维管理平台。

3) 充分考虑到跨平台系统的兼容性,利用日常运维管理的工作经验,将繁琐、专业化的操作转化为自动化脚本;同时,通过消息通信技术保证其抓取数据的实时性和准确性。

2 系统设计

从企业数据中心的日常运维角度出发,围绕日常巡检、备份检查、应用/数据库服务器的可用性和日常繁琐的重复性运维工作等进行统一设计,采用面向对象的设计思想,利用 PHP,JavaScript 和 HTML 等 Web 技术,结合 Unix Shell 和 Python 技术进行统一无缝调度。该架构适合跨服务器平台的系统自动化运维管理,具有高灵活的动态扩展性。

2.1 顶层架构设计

根据企业已有运维系统的使用情况,在不影响现有架构的前提下架设一套自动化的综合管理平台,以图形化的界面灵活管理所有核心业务系统及日常运维工作。平台分层的架构见图 1。

2.2 底层数据抓取设计

系统基础信息采集模块作为监控模块的重要组成部分,不仅能帮助运维人员了解系统当前的健康程度,同时还是衡量业务服务质量的依据。例如,系统资源吃紧会直接影响业务的服务质量和用户体验。此外,获取设备的流量信息也可让运维人员更好地评估带宽和设备指标数据,包括 Linux 基本信息、块设备、系统信息和网络地址库等信息。在采集到这些数据后,即可全方位了解系统服务的状态,再结合告警机制,可在第

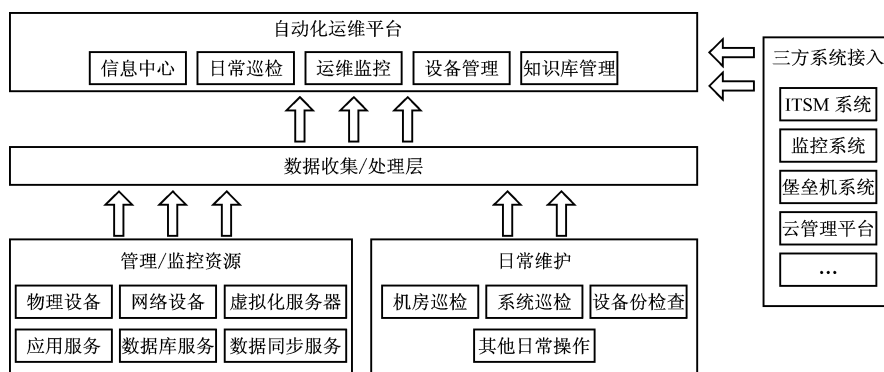


图1 平台分层架构

一时间响应,使得异常现象及时得以处理^[2]。

鉴于企业级服务器基本以 Unix 小型机和 Linux X86 服务器系统为主,计划在底层数据抓取层利用 Unix Shell & Python 实现对小型机 Unix 和 Linux 系统健康状况及服务进程状态的抓取。

2.3 自动化运维功能设计

根据企业级信息化系统日常运维的特点,自动化运维管理平台将实现以下 6 类的数据统一监管和综合展现,实现即时的准确性数据捕获及告警。在运用面向对象的设计思想对 IT 运维资源进行对象化、功能化划分后,结合平台设计建设目标,从信息化部门运维人员角度出发,把平台应用细分为以下 6 个功能区。

1) 日常巡检管理。以工作表单的形式记录汇总所有数据中心值班人员和系统运维人员对基础设备、系统环境及数据库系统的常规健康检查工作,若遇到故障事件,通过单点登入相应的事件管理系统(IT Service Management, ITSM)进行后续事件跟踪。

2) 运维监控管理。以图形化的方式实时监控核心业务系统及数据库的可用性;异地/本地灾备环境的数据库同步也以简化繁琐的命令行代码为前提,实时展现其同步情况,并细化到其进程状态,以便非专业的运维值班人员一目了然地发现问题。

3) 设备资产管理。通过集中化的平台和规范的变更流程细化并归档所有硬件物理设备、网络设备及虚拟化服务器的信息,并通过对各业务系统进行归类划分使信息化部门的运维管理人员可即时查到所需的服务器信息。

4) 知识库管理。统一地管理和归档所有的技术类运维文档,以供后续维护人员查阅。

5) 第三方平台接入。根据已有的 IT 运维管理工具(如监控系统、ITSM 流程管理系统、堡垒机服务等),通过配置简易的账户信息,实现从自动化运维平台到其他三方系统的单点登入。

6) 信息中心管理。企业内部的运维管理条例、运维体系规定和信息化动态的实时发布。

3 系统实现及应用

某大型集团航运企业拥有异地双数据中心,两中心内共有 400 余台投产信息化设备,其中小型机 17 台,主要用于核心业务系统的数据库;共有 300 余台 X86 服务器,其中 64 台服务器提供虚拟化服务(约提供 435 台虚机);存储设备 37 台;网络设备 200 余台。

3.1 企业运维现状

目前,该企业运维现状的困惑主要表现在以下 5 个方面。

3.1.1 应用、数据库可用性、数据本地/异地同步等运维监控手段缺乏

根据该企业的日常运维反馈,监控和定期巡检所有操作系统、应用服务及数据库服务的运行情况和健康状况,对现有的系统工程师来说工作量较大。之前一些有运维经验的技术人员会手工编写一些 Shell 脚本部署在部分服务器上运维,但其在功能和时效性上有很大局限性,且时常会因人工遗漏或疏忽而造成不必要的系统异常(如磁盘空间满、数据库无法访问和数据同步异常等)。

3.1.2 数据中心日常巡检跟踪缺失

运维中心的机房值班人员未将每日常规的巡检工作形成工作单,也未统计工作量,在需要对过往操作进

行回溯时,往往仅凭记忆,难以确认。

3.1.3 物理硬件设备和虚拟化服务器的生命周期管理缺失

对于数据中心内的硬件设备和虚拟化服务器的管理,依靠传统的 Excel 文档进行人工记录和变更,缺少一个系统的平台进行管理;对于其生命周期的管理,从资源开设到配置和后续回收都依靠往来邮件进行归档,缺少体系化的变更管理。

3.1.4 日常运维的知识库管理缺失

对于日常系统的运维,记录和技术文档的归档、发布及管理仅凭邮件往来,缺少系统的平台进行管理维护,有碍于企业知识储备的积累和回溯。

3.1.5 已有的第三方平台繁多,系统入口不统一

大型企业在运维管理过程中往往已引入第三方的管理工具和平台,如企业内部办公自动化(Office Automation,OA)平台、硬件设备监控平台和 ITSM 平台等。功能性系统一旦陆续上线,对于企业运维人员而言,平台的入口往往很难记清,缺少统一的单点登录接口提供运维人员的日常操作。

在结合企业的实际运维需求后,该系统除了具有传统功能之外,还在运维监控模块的实现方式上进行细化,通过服务器端代理程序自动收集系统数据,准时地将采集到的数据同步至系统后台,并在自动化运维管理平台上集中展现。

3.2 系统功能实现

3.2.1 本地/异地数据库同步监控

本地 Oracle Dataguard 和异地 Shareplex 的监控对于企业运维而言往往是至关重要的,其数据复制的实时性和准确性保障着企业核心业务系统数据的安全。

在借助自动化运维平台之前,企业需依靠具备专业知识的数据库管理员(Database Administrator, DBA)去检查源端和目标端的系统及数据库状态、数据积压量和延迟时间,这对于数据中心值班人员或非专业人士而言较为困难。

系统通过 shell 代码实现对所有指标的本地数据抓取,同时通过消息传输机制,将实时数据准确地传输至运维平台后端数据库内,再由平台统一展现,做到及时告警,准确定位延迟队列和问题故障点^[3]。图 2 为运维平台异地同步监控界面。部分功能实现代码如下:

```
function check_splex(){
    n=1
    while ((n<= $(cat $oonfig|wc-1)))
    do
        source_ip[n], source_port[n], source_db[n], user[n], password[n], source_connect_string[n];
        # check the source sequence
        capture_seq_value=$(sqlplus -S $(user[n])/$ {password[n]}@ $(source_connect_string[n])<<eof@./sql/check_
_splex_capture.sql
        exit
        eof)
        # check the target sequence
        source_seq_value=$(sqlplus -S $(user[n])/$ {password[n]}@ $(source_connect_string[n])<<eof@./sql/check_
db_seq.sql
        exit
        eof)
        ((n+=1))
    done
}

function deliverToMysql(){
    # sync model
}
```

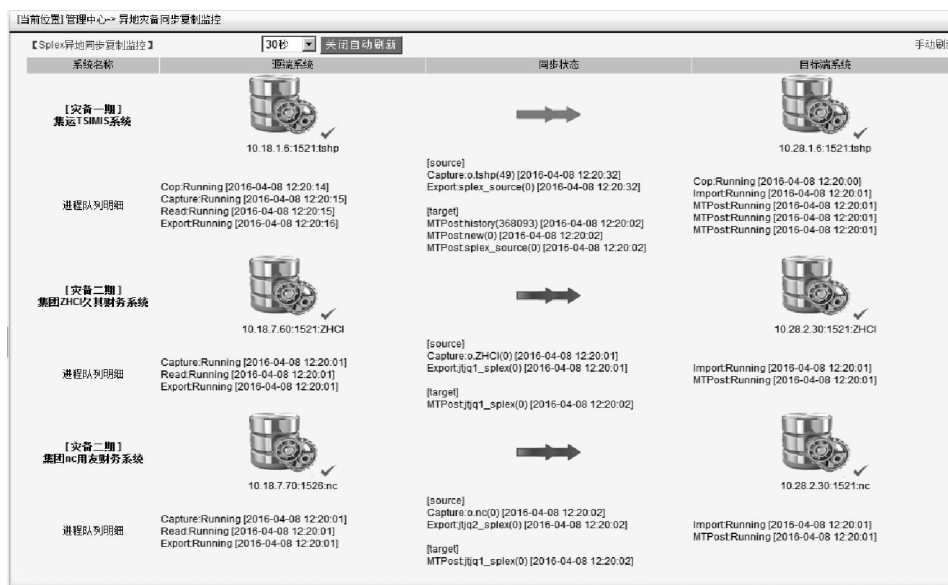


图2 运维平台:异地同步监控界面

3.2.2 应用/数据库服务可用性监控

应用服务的可用性健康与否是企业信息化部门日常运维工作中一项重要的任务指标,业务系统和企业公用系统的稳定运作保障着企业日常办公的顺利开展和业务的正常运营。

在借助自动化运维平台之前,往往是最终用户或研发团队无法使用时告知运维部门,这使企业运维团队无法第一时间发现故障。

通过实时抓取应用服务和数据库服务的状态,批量将信息发送至平台前端,供数据中心值班人员监控,解决人工发现故障时效性差的问题^[4]。图3和图4分别为运维平台应用可用性监控界面及数据库可用性监控界面。

【应用可用性监控】					
系统名称	系统IP地址	状态	返回时间(s)	操作	
集团OA系统	10.18.7.20	✓	0.004136 s	进入浏览	
集团HR系统	10.18.7.164	✓	0.003895 s	进入浏览	
集团HR系统	10.18.7.52	✓	0.007697 s	进入浏览	
集团NC系统	10.18.7.30	✓	0.006973 s	进入浏览	
集团欠其系统	10.18.7.7	✓	0.001346 s	进入浏览	
测试失败地址	222.222.222.222	✗	✗	进入浏览	
外网地址	www.baidu.com	✓	0.027519 s	进入浏览	

图3 运维平台:应用可用性监控界面

【数据库可用性监控】						
系统名称	系统IP地址	端口	SID	状态	操作	
集团OA系统	10.18.7.25	1521	jtoa	✓	查看DB报告	
集团HR系统	10.18.7.91	1521	orcl	✓	查看DB报告	
集团航标系统	10.18.7.50	1521	cbgl	✓	查看DB报告	
集团NC系统	10.18.7.70	1526	nc	✓	查看DB报告	
集团欠其系统	10.18.7.60	1521	zhci	✓	查看DB报告	
集团TSMIS系统	10.18.1.6	1521	tshp	✓	查看DB报告	
集团核算TSHS系统	10.18.1.8	1521	tshs	✓	查看DB报告	

图4 运维平台:数据库可用性监控界面

PHP 部分功能实现代码^[5]如下:

(1)应用服务可用性代码:

```
public function returnAppCheckStatus($web_address){
    try{
        $ch=curl_init();
```

```

curl_setopt( $ch, CURLOPT_URL, $web_address);
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt( $ch, CURLOPT_TIMEOUT, 10);
curl_setopt( $ch, CURLOPT_HEADER, true);
curl_setopt( $ch, CURLOPT_FOLLOWLOCATION, 1);
curl_exec( $ch);
$info=curl_getinfo( $ch);

if( ! curl_errno( $ch)){
    $info=curl_getinfo( $ch);
    if( $info['http_code']==" 200"){
        return $info['total_time'];
    }else{
        return false;
    }
}else{
    return false;
}
}
catch(Exception $e){
    //catch the error and poll the exception to main
    throw new Exception( $this->error_info);
    return null;
}
}

```

(2)数据库服务可用性代码:

```

$obj_db_connect_tmp=new ConnectOracle();
$obj_db_connect_tmp->setDBServerInfo( $db_ip, $db_port, $db_name);
$obj_db_connect_tmp->setDBLoginInfo( $db_user, $db_pwd, $db_charset);

set_time_limit(3);
if( $obj_db_connect_tmp->newConnect()){

    $sql=" select 1 from dual";
    $result= $obj_db_connect_tmp->query( $sql,1,array(" 1"));
    # db_status=(sizeof( $result)>0 ? OK : " FALSE");

}else{

    $db_status=" FALSE";

}

```

3.2.3 自动化备份检查

数据库、应用附件和文件的备份可靠性关系着企业数据的安全及故障发生的可恢复节点。备份有效性的检查和验证尤为重要。通常信息化运维部门会安排专人进行人工检查和复核,但往往容易出现疏忽或遗漏现象造成数据丢失。通过 Shell 脚本实现对不同种类的业务备份数据的统一自动化检查和结果推送,以保障系统备份的有效性。图 5 为运维系统自动化备份检查界面。

【当前位置】管理中心->备份检查			
【每日备份检查报告列表】			
ID号	检查日期	检查状态	操作
20160408_1460073606	2016-04-08 08:00:06	✓▲	浏览报告
20160407_1459987206	2016-04-07 08:00:06	✓▲	浏览报告
20160406_1459908006	2016-04-06 08:00:07	✓▲	浏览报告
20160405_1459814406	2016-04-05 08:00:06	✓▲	浏览报告
20160404_1459728006	2016-04-04 08:00:06	✓▲	浏览报告
20160403_1459641606	2016-04-03 08:00:06	✓▲	浏览报告
20160402_1459555208	2016-04-02 08:00:08	✓▲	浏览报告
20160401_1459468806	2016-04-01 08:00:06	⊕▲	浏览报告
20160331_1459382406	2016-03-31 08:00:06	⊕▲	浏览报告
20160330_1459296007	2016-03-30 08:00:07	✓▲	浏览报告
共 4 页 当前第 1 页 记录数 36 条 当前 1-10			
<<首页 <上一页 1 2 3 下一页> 末页>>			

图5 运维平台:自动化备份检查界面

4 结 语

自动化脚本是自动化运维实现的基石,源于运维人员长期的工作积累。通过构建自动化运维平台完成大型企业日常繁琐、重复的 IT 基础运维管理工作,是较为高效、稳妥和即时的解决方案。某大型集团企业在使用该系统前后的指标对比见表 1。

表1 使用该系统前后对比

事件	传统模式	自动化运维平台
设备资产管理	Excel、邮件	集成的设备及变更流程管理
日常巡检	无记录	巡检管理模块、工作单形式
应用/数据库可用性	人工报账、用户反馈	自动发现、故障告警
本地/异地 数据库同步	人工定期检查、用户反馈	自动发现、故障告警
备份检查	人工定期检查	自动发现、故障告警

由指标对比可知,自动化运维平台为企业日常信息化运维工作的模式带来了新的转变,未来可结合企业的新需求对该系统的功能进行扩展,以降低企业信息化人员的运维成本,保证高效、稳妥地运营。

参考文献:

- [1] 屈喜龙. UML 及面向对象的分析与设计的研究[J]. 计算机应用研究, 2005, 22(9):74-76.
- [2] 刘天斯. Python 自动化运维[M]. 北京:机械工业出版社, 2014.
- [3] 王军. LINUX 系统命令及 SHELL 脚本实践指南[M]. 北京:机械工业出版社, 2014.
- [4] 陈吉平. 构建 Oracle 高可用环境:企业级高可用数据库架构. 实战与经验总结[M]. 北京:电子工业出版社, 2008.
- [5] Macintyre P, Danchilla B, Gogala M. Pro PHP Programming[M]. New York: Apress, 2011.