

基于分布式及微服务的 IT 自动化工具开发运维实践

Practice in Development and Operation of IT Automation Tools Based on Distributed and Microservice

付永振 魏春来

Yongzhen Fu Chunlai Wei

中国移动通信集团河北有限公司, 中国·河北 石家庄 050035

China Mobile Group Hebei Co., Ltd. Shijiazhuang, Hebei, 050035, China

【摘要】随着私有云 IT 设备数量的日益增加和新技术的不断应用,以及资源变动的日益频繁,维护人员所面临的压力也越来越大。如若遵循原有的维护方式,凭借现有的维护人员数量、维护人员能力、维护工作经验,已难以应对公司向 ICT 转型过程中所出现的新挑战。本文秉承分布式及微服务的理念,开发了 IT 自动化运维工具,实现了对于 IT 基础设施的健康检查、配置管理、变更管理等功能。不仅有效地提高了运维人员的工作效率,也一定程度上提升了员工的自主研发水平,有效促进了公司 IT 运维向 DevOps 模式转型。

【Abstract】With the increasing of private cloud IT devices and the continuous application of new technologies, the pressure on maintenance is also increasing. If we follow the original maintenance methods, relying on the existing capabilities of maintenance, it has been difficult to cope with the new challenges that companies face in the ICT transition. Therefore, this article adheres to the concept of distributed and microservice, developed IT operation and maintenance of automation tools and implemented main functions, such as health check, configuration management, and change management for IT infrastructures. It effectively improves the work efficiency of operation and maintenance, but also promotes the level of independent development and the transformation of IT operation and maintenance to DevOps mode.

【关键词】IT 自动化运维;分布式;微服务;DevOps

【Keywords】IT operation and maintenance of automation; distributed; microservice; DevOps

【DOI】<http://dx.doi.org/10.26549/gejsygl.v2i6.826>

1 现网概况

随着河北移动私有云网管资源池 IT 设备规模的不断增长以及管理的深入,IT 集中化管理对于 IT 运维优化提出了更高的要求,一是公司明确要求扎实推进 IT 技术储备,用三年时间快速提升 IT 自主运维能力,追赶业界先进技术水平;二是预计到 2018 年 5 月份,河北网管资源池上线设备将增长 1 倍,加上非资源池设备主机达到 2400 台,日常维护工作量和难度将进一步增大;三是随着 SDN、SDS(软件定义存储)、NFV、容器等新技术的引入,现有的人员配置、能力水平、支撑工具等已越来越无法满足 IT 运维中故障及时发现、问题快速响应、环境稳定运行、性能不断调优等运维需求。

1.1 IT 运维之要求

为了能够保障各类业务系统的稳定运行,IT 运维通常会有以下要求:

1.1.1 事前预警

在故障出现之前,管理人员应该能在任何时间、任何地点以各种方式接收到预警信息,并且能够做到及时准确地处理问题,尽可能地把故障隐患扼杀在摇篮之中。

1.1.2 事中恢复

正所谓“天有不测风云”,纵然是再完美的方案也会有预料之外的故障。为保证业务在最短的时间内恢复,并且关键数据不因故障而丢失,我们需要有完整的备份方案来做到应对自如。

1.1.3 事后存档

此要求更加强调运维管理的方法。对于曾经处理过的故障,应该详尽地记录在案。对于故障处理的技术方案、故障处理的具体过程、故障处理的衍生问题,应该进行思考总结,提炼形成经验文档,以便日后能够快速准确地发现解决问题。

1.2 IT 运维之痛楚

在 IT 运维要求的指导下,IT 运维工程师竭尽全力地做好自己的工作。然而对于每个运维工程师来说,IT 运维工作却无疑是一种持久而痛苦的经历。目前 IT 运维工作中的痛楚有以下几点:

1.2.1 工作繁琐

运维人员面临的工作任务复杂繁重,诸如软硬件环境的部署,设备信息的统计核实,系统的调优,变更情况的复核等等。日复一日,年复一年,使运维人员感觉有始无终。

1.2.2 压力山大

运维人员的时间是有限的,运维人员的精力也是有限的,然而各种繁琐工作交织错杂,却显得无尽无穷。尤其是当各类故障猝不及防地出现时,在系统上线、调优、升级、恢复等特殊环境下,运维人员不仅仅要冷静快速地处理问题,同时还要面对业务侧的监察与盘问,使得运维人员倍感压力山大^[1]。

1.2.3 设备系统故障

设备系统,尤其是过保的硬件设备,很容易出故障。机房的温度湿度、业务的频繁读写、人员的野蛮使用等各种因素都可能导致设备系统意外故障的发生。意外就是意外,往往在不恰当的时间地点出现,令运维人员无比头疼。

1.2.4 IT 消防员

IT 运维人员就是 IT 的消防员,IT 运维人员的最高境界就是无我境界。大家使用系统很舒服时,都不会想起我;一旦想起来我,就是 IT 环境出了问题。此时,IT 运维人员只有硬着头皮去“擦屁股”,牺牲我一个,幸福一大家。

1.2.5 背黑锅

运维人员天生具有背黑锅的命。当找不出别人的问题时,那就只能由运维人员来背黑锅;即使能够找出问题,运维人员也可能会一同背黑锅。任何行业工作都有其委屈尴尬的一面,背黑锅是运维人员成熟历练的必经之路。

1.3 IT 运维之答案

为了解决 IT 运维中的痛楚,我们应该从人、事、物、流程四个方面将运维体系进行解构,使其互相作用,共同构建一个完整实用的运维体系,以达到解放运维人员的目的^[2]。

1.3.1 人

提高人员素质。完善岗位职责、规划职业发展、提升技术水平、促进知识共享、完善绩效考核、规范行为规范,以建立一支工作高效、技术高超、团结稳定、有职业素养的运维队伍。

1.3.2 事

明确做事目的。做好基础运维工作,保障业务正常运行,并且不断探索新的运维理念,优化系统架构。具体可以分为运维流程管理、资源架构规划、应急处理、监控与优化、安全与防护等。其目的是做到不仅要明白运维是做正确的事,而且要懂得如何正确地做事,如何有章法地做事,如何稳定高效地做事。

1.3.3 物

合理管控资源。管控好保障系统运行所需的所有软硬件资源,包括机房环境、系统服务器、网络设备、操作系统、应用软件等,并且依靠合理的管控流程,使资源合理配置,资源属性明确,资源变更清晰,以达到物尽其用的目的。

1.3.4 流程

厘清做事流程。运用合理的流程规范将上述要素(人、事、

物)有机地结合,使其有序科学地流转、稳定高效地运行。例如资源规划流程、资源采购流程、项目管理规范、软硬件部署规范、安全生产制度、工作交接流程等。

面对以上问题,实施自动化运维,便能够很好将人、事、物、流程四要素组织起来。运维体系的好坏可以影响自动化运维的实施执行,反过来说,自动化运维的部署也可以推动运维体系的建设。

1.4 自动化运维之所谓

自动化运维,即在最少的人工干预下,综合运用脚本与第三方工具,保证业务系统 7*24 小时高效稳定地运行,这是所有 IT 运维人员的终极梦想。按照运维发展的成熟度来看,运维大致可分为以下三个阶段:

①纯手工:依靠纯手工运维,需要重复地进行软件的部署与运维。

②编写脚本:通过编写脚本运维,可以方便地进行软件的部署与运维。

③第三方工具:借助第三方工具运维,能够高效地进行软件的部署与运维。

面对一些轻量级的业务系统,运维人员可以通过纯手工或编写脚本的方式完成运维管理。然而当面对那些重量级的业务系统时,运维人员则需要采用成熟度较高的开源工具或第三方管理工具。一般来说比较成熟的第三方工具都能实现以上运维需求,而且管理操作起来也更加方便。

运维方式	运维规模(x:设备台数)	运维效率	技能要求	劳动力损耗	规范管理
纯手工	小($x < 100$)	低下	一般	较大	难度较大
脚本	中($100 < x < 500$)	适中	较高	较低	相对容易
第三方工具	大($x > 500$)	高效	一般	较低	相对容易

表 1 运维方式对比

2 基于分布式及微服务的 IT 自动化运维工具

在对目前行业主流的开源及商用自动化运维解决方案的学习和比较之后,我们提出了基于分布式及微服务理念,且更加适合于我省 IT 系统现状及今后发展的 IT 自动化运维工具开发解决方案,以实现快速化、批量化、规范化、场景化、可视化的 IT 运维要求^[3]。

2.1 软件架构

整个 IT 自动化运维工具软件架构主要分为以下部分。

环境管理平台:通过环境管理平台,可以实现环境参数的采集、配置参数的采集以及配置文件的备份等功能[4]。

① 配置管理平台:通过配置管理平台,可以实现系统配置及应用配置等功能。

② 发布管理平台:通过发布管理平台,可以执行发布应用

表 2 自动化运维方案对比

对比项目	方案 1:基于 ansible 的方案	方案 2:基于 saltstack 的方案	方案 3:本项目 IT 自动化运维方案
系统效率	1. 通过 ssh 连接终端主机,并以指令方式或以 playbook 模块执行 yaml 脚本方式来完成。任务。 2.基于 python 开发,分布式,无需客户端。 3.服务端需要保留终端主机的账号密码信息,存在安全隐患。无权限控制管理。 4.官方 UI 界面不符合现场维护工作习惯,需要定制开发。	1.通过在管理对象中安装 agent 方式接入。 2.基于 python 开发,使用的 State 语法可读性较差。 3.通过 ZeroMQ 传输数据执行效率高。数据传输安全性较低。无权限控制管理。	1.支持 ssh 直连、telnet 直连、agent 代理等方式连接对象设备。 2.基于 command 及 shell-script 等方式构建原子操作,通过分布式、无状态或弱状态服务架构,实现高效服务并发,以及灵活的运维服务内容、规则等编排服务。 3.支持权限控制管理。
用户体验	1.作业执行均通过命令方式完成,自带的 TOWER 界面工具需收费,并且无法满足现场使用需求。 2.对执行对象和执行内容调整需要修改原始脚本内容,存在影响已有功能正确性的问题。 3.要求操作及使用人员熟悉 ansible 及 python。 4.执行结果不具备友好的界面呈现,可读性较差。	1.可通过 web 界面执行操作,但是操作过程较为复杂。 2.对执行对象和执行内容调整是需要修改原始脚本内容,存在影响已有功能正确性的问题。 3.要求操作及使用人员熟悉 saltstack 及 python。	1.定制化的 UI 界面,所有任务创建、作业执行均可通过界面操作完成。 2.界面符合现场维护及管理人员使用习惯,可直接界面呈现执行结果,以及结果导出。 3.支持所有执行内容主机原始记录查询,便于维护人员进行问题分类分析。 4.原子操作及服务编排,实现复杂操作解耦,增强服务灵活性。
技术难度	1.Ansible 轻量化系统架构,部署后一般只需对其中的被管理主机 IP 进行一些梳理即可。 2.无法进行执行权限控制。 3.设备环境复杂,系统类型多样,无法统一标准化处理。	1.master-minion 结构对服务器性能要求高,资源消耗较大,容易出现性能问题。 2.设备环境复杂,系统类型多样,无法统一标准化处理	1.设置环境管理模块,自动判断环境参数、匹配原子操作,实现作业自匹配操作。 2.可实现权限控制。 3.通过环境管理模块实现对复杂设备环境的作业管理。 4.具备灵活的编排能力和分布式架构,支持高效的任务并发能力。

的功能。

③巡检管理平台:通过巡检管理平台,可以实现数据采集及检测的功能。

④集成功能平台:通过集成功能平台,可以实现功能展示及数据展示的功能。

⑤流程管理平台:通过流程管理平台,可以实现整合操作项、形成服务、执行服务等功能。

⑥业务数据库:业务数据库是自动化业务数据存放的数据库。

⑦策略数据库:策略数据库是自动化业务配置策略存放的数据库。

①高并发:通过设计保证系统能够同时并行处理很多请求。

②可扩展性:通过不断扩张主机的数量以实现横向水平性能的扩展。

③高可用:通过多点、冗余设计,减少系统停工时间,保持其服务的高度可用性。



图 1 软件架构

2.2 分布式系统特点

分布式系统简单来说就是一群独立计算机集合共同对外提供服务,但是对于系统的用户来说,就像是一台计算机在提供服务一样。分布式意味着可以采用更多的普通计算机(相对于昂贵的大型机)组成分布式集群对外提供服务。计算机越多,CPU、内存、存储资源等也就越多,能够处理的并发访问量也就越大。

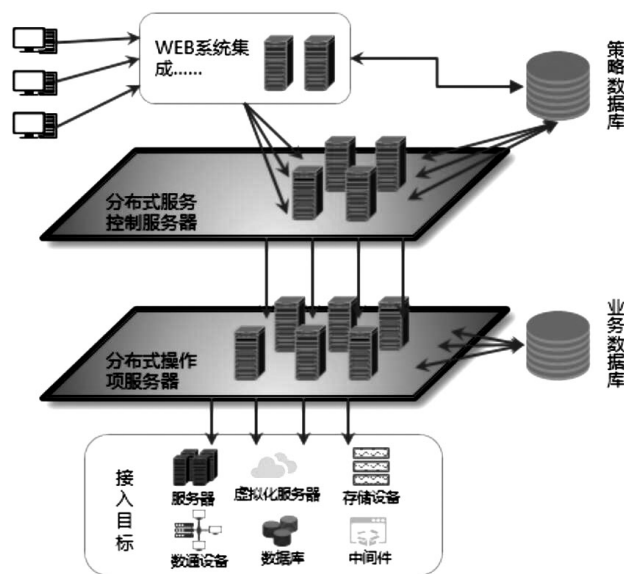


图 2 分布式特点

IT 自动化运维工具开发基于分布式理念进行应用架构设计,内置高效的作业调度引擎,对上万并行作业过程进行详尽记录,提供可视化跟踪和控制界面,让一切操作尽在掌握之中。

比如操作项管理平台,可依据所纳管的对象设备或软件数量进行水平扩展,以提高并发执行效率。

IT 自动化运维工具分布式技术应用举例:

1) 采用 EhCache 组件实现 JAVA 分布式缓存技术。

Ehcache 是一种广泛使用的开源 Java 分布式缓存。主要面向通用缓存,Java EE 和轻量级容器。它具有内存和磁盘存储、缓存加载器、缓存扩展、缓存异常处理程序,用于管理本地的控制服务任务,插入、取出、移除、销毁。目前控制服务平台有 3 个 cache,service 缓存队列、service 执行队列、调试信息 cache。

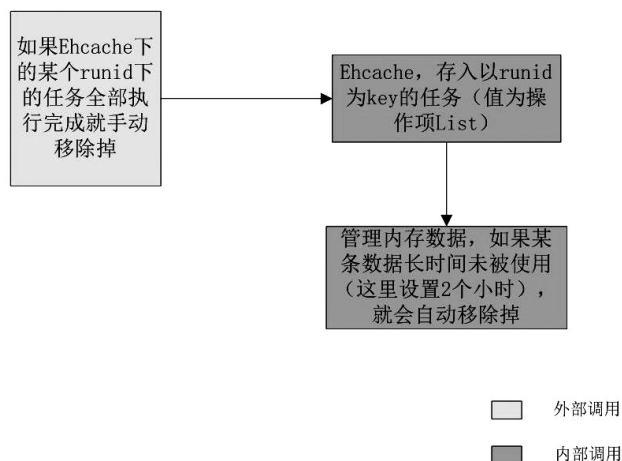


图3 EhCache 组件技术

2) 水平扩展、高并发操作执行

通过分布式服务技术设计,实现控制服务及执行服务的高并发及水平扩展等特性。如平台操作执行流程分为3部分:

- ①服务注册目录读取及下发
- ②操作项注册目录读取及下发
- ③操作结果入库

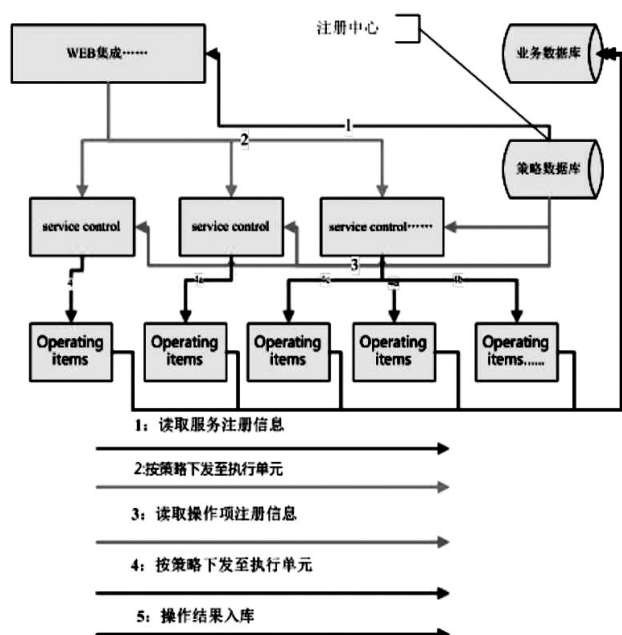


图4 操作执行流程

2.3 基于微服务理念的原子化思路

引入微服务架构理念。采用一组服务的方式来构建一个

应用,服务独立部署在不同的进程中,不同服务通过一些轻量级交互机制来通信,服务可独立扩展伸缩,每个服务定义了明确的边界,不同的服务甚至可以采用不同的编程语言来实现,由独立的团队来维护。每一个微服务都有自己的业务逻辑和适配器。微服务在互联网公司使用越来越多,更适合于标准化或同构的管理对象的服务。

在微服务基础上我们提出了更细操作维度、更适用于异构化管理对象的原子化操作思路。原子化操作可以是一个操作命令、一个简单脚本、一段复杂程序,通过对原子化操作的编排及编程方式,将一组原子化操作进行顺序、并行组合,提供具有灵活、敏捷、高效的服务开发、测试、发布过程。

将原有的复杂操作分解成更细操作维度、更适用于异构化管理对象的原子化基础操作思路,可以是一个操作命令、一个简单脚本、一段复杂程序,通过编排及编程方式,将一组原子化操作进行顺序、并行等组合,提供具有灵活、敏捷、高效的服务开发、测试、发布过程。

IT 自动化运维工具微服务技术架构应用举例:

把一个大型的复杂的应用,拆分成一个或者多个微服务。每一个微服务关注于完成一个任务,每一个任务代表着一个小的业务能力。通过再编排实现不同的应用。

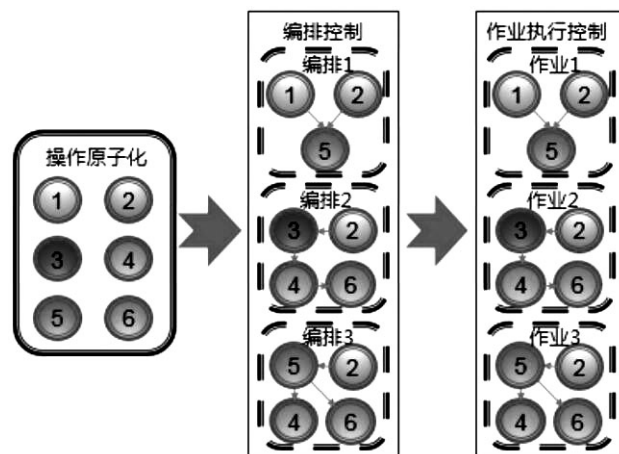


图5 微服务架构示意图

比如自动化巡检服务:将巡检作业目录拆分成小的可以单独完成的巡检小脚本或者单条命令,通过编排自定义重新编排组合成复杂的巡检项目,理论可以实现所有设备的所有品牌的巡检工作。实现最终使用者通过自定义选择所需的单个巡检内容,对不同的设备类型、设备等级组合成复杂的针对性巡检作业覆盖。

2.4 拟社会化微服务架构

经典微服务架构:一般包含两个部分:API 网关,一组微服务,API 网关是唯一的请求入口,它还要负责负载均衡,路由编排,失效切换等工作,存在以下缺陷:“笨重”的 API 网关,由于

它要负责各种核心功能,不能灵活扩展,比如负载均衡策略,也许每个微服务类型需求都不一样,它很难灵活变更;随着对接的微服务越来越多,每个 API 网关也集成大量的功能。

拟社会化设计思想:基于 springcloud 开源架构进行优化(注册、路由选择、负载均衡、熔断机制),通过自动发现、自我维护、自动适应,实现不同时间不同资源使用情况下服务等级阈值的动态变化,实现数百个服务实例依据上下文的环境以及历史状态的自主分析调节。

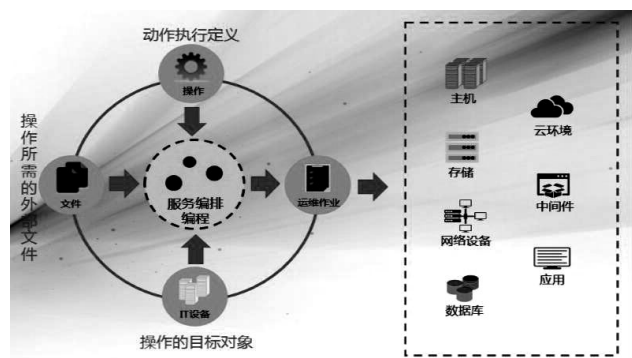


图 6 微服务架构

2.5 管理能力

IT 自动化运维工具,是基于 DevOps 敏捷运维理念打造,提供了业务应用和基础架构的自动化能力。本工具内置丰富的运维操作以及各种标准化操作流程,可通过简洁的可视化方式灵活应对各种运维场景。本工具可根据运维现状按需扩展,以实现最大限度地节约人力成本、降低管理风险、提升运维效率、提高服务满意度。

2.5.1 全栈运维场景

IT 自动化运维工具覆盖了全栈运维场景,纳管范围包括主机(刀片、服务器、虚拟机)、存储设备、操作系统、数据库、中间件、云环境等 IT 资源。

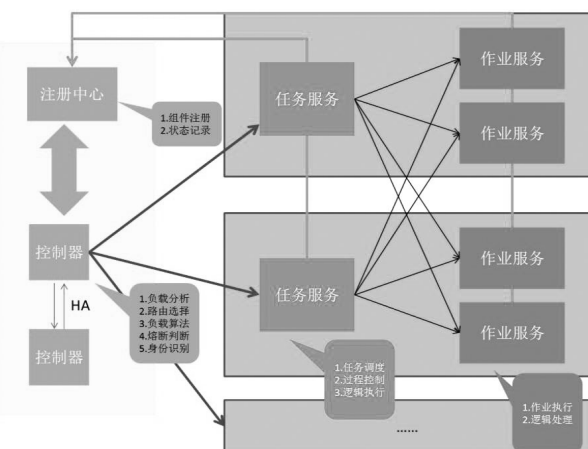


图 7 全栈运维场景

2.5.2 原子化操作

IT 自动化运维工具基于微服务架构的理念,提出了更细

操作维度、更适用于异构化管理对象的原子化操作思路。原子化操作可以是一个操作命令、一个简单脚本、一段复杂程序。通过编排/编程服务,按需扩展实现特定功能,更加符合 IT 运维自动化需求。

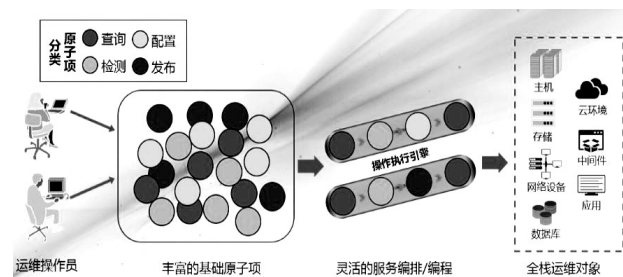


图 8 原子化操作

2.5.3 灵活的编排性与编程性

编排性:提供可视化的流程编辑器,根据指定的执行顺序、并发模式、逻辑依赖进行原子操作编排。支持任务的单次执行、循环执行与定时执行。

编程性:前一个原子操作输出参数值可被之后的操作引用,实现数据传递。原子操作可以基于脚本实现,且脚本具有较强的编程执行能力。针对不同的事件编程不同的响应动作,支撑自动化事件响应需求。

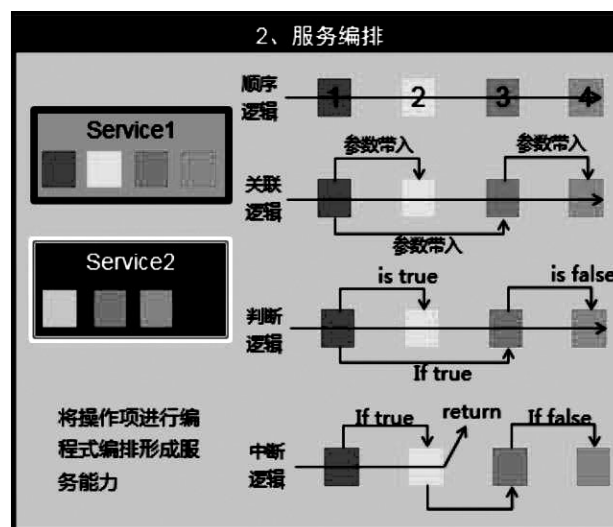


图 9 编排性与编程性

2.5.4 高效并发调度

系统内置高效的作业调度引擎,对上万个并行作业过程进行详尽记录,提供可视化跟踪和控制界面,让一切操作尽在掌握之中。(见图 10)

3 DevOps 理念

通过 DevOps 理念的渗透执行,重点突出开发人员和运维人员的沟通合作,通过自动化流程使得软件构建、测试、发布更加快捷、频繁、可靠。在实际工作中,运维人员通过积累的运维知识经验,利用 IT 自动化运维工具的自开发能力特点,针

对场景化要求输出运维操作项及服务编排内容。以此积极推动运维人员自主研发的积极性,与此同时也提升了员工的自主研发能力。目前员工已经能够自行完成脚本的编写、测试、下发、执行等工作,满足部分运维工作要求,如针对主机设备实现以天为粒度的性能指标采集、业务进程状态采集等任务。应用本工具以来,有效地减少了人机操作工作量,提高了员工运维效率。

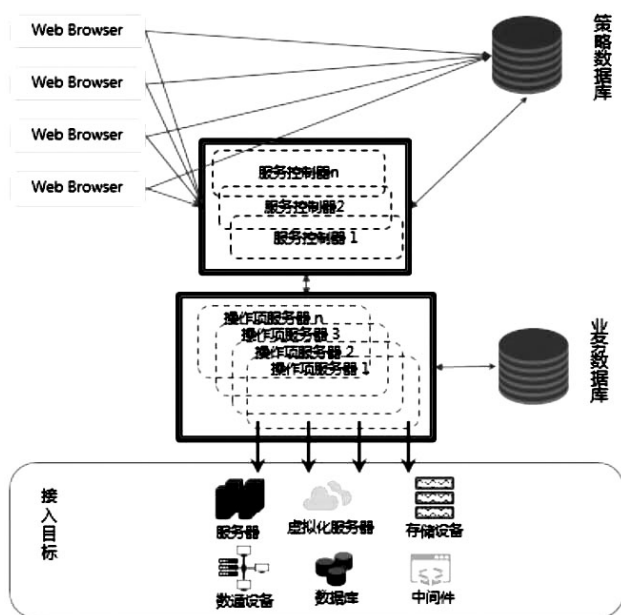


图 10 并发调度

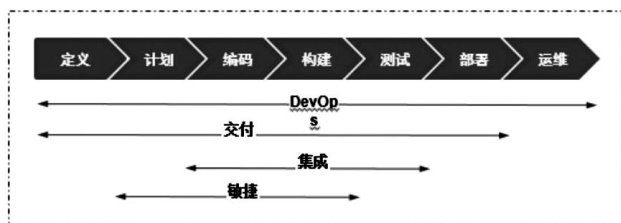


图 11 DevOps 理念

4 场景实践效果

在批量化、标准化的前提下,IT 自动化运维工具对于运维操作的过程及结果追溯提供了可视化。通过基于场景进行抽象和层次的整合,形成了统一的运维入口、交互过程、处理流程、数据格式,最后通过适配接口与本工具进行对接,实现基于场景的运维管理。同时还加入了可视化要素,把数据和关系的展现从平台扩展到立体,更清晰直观。

对于运维而言,不同的规模、现状、需求均各具差异,技术更是在快速地演进过程中,唯有运维的场景是不会变化的。通过平台化整合和场景化积累,构建新一代运维体系,最终实现可视化运维将是运维的演进方向。

目前通过 IT 自动化运维工具已纳管接入了 500 多台主

机设备,已定制并使用了健康检查、配置管理、变更管理等相关运维任务。

4.1 健康检查

以前的巡检方式,由于人员素质、支撑工具等条件的限制,难以满足全覆盖、高频次、关联性的要求。本工具通过对系统集成环境执行全面的周期性检查,实现了及时掌握系统的健康状态、发现系统隐患,保障了 IT 系统的稳定运行。

IT 自动化运维工具的健康检查功能包括检查内容自定义、周期自定义、范围自定义。检查内容自定义了丰富的原子化检查内容,通过服务编排界面操作灵活定制批量检查服务。检查作业执行结束输出检查结果报告,并主动对异常结果进行告警通知。(见图 12,图 13)

4.2 配置管理

通过对系统集成环境中的重点配置数据进行定时地备份,保证在配置数据丢失或配置误操作等情况下,可以快速查询备份记录,并完成配置数据恢复。

4.3 变更管理

当需要对批量对象执行标准化变更时,可以通过 IT 自动化运维工具执行变更内容的定义和编排,提供标准化的变更操作。变更编排服务必须包含变更前的 checklist、变更操作、变更后的 checklist、异常失败后的恢复等操作过程,这些过程均可通过灵活的编排及编程进行定义和控制。(见图 14)

4.4 建设收益

IT 自动化运维工具部署后,运维工作效率得到了有效地提升。

表 3 自动化运维方案对比

	人工方式	自动化方式	提升效率
补丁升级(openssh 补丁,共 100 台)	15 台/人天(8 小时)	10 台/批次, 每次 15 分钟。	2176%
健康检查(对象主机共 300 台)	50 台/人天(8 小时)	25 分钟完成巡检, 并产出结果。	17808%

5 结语

IT 自动化运维工具采用松耦合、模块化、层次化理念,广泛应用先进的分布式、微服务技术,支持服务能力对外开放共享和弹性伸缩,整体方案符合公司 ICT 技术路线发展规划,系统的上线改变了传统 IT 设备运维模式,通过原子化、标准化、灵活的任务编排更好地支撑了上层应用稳定高效运行,极大提升 IT 设备自动化运维水平和运维效率。

随着公司 IT 自动化运维的不断发展,运维工作由人工高密度型向技术高密度型转变,人才发展由基础系统运维向开发运维复合型转变,通过运用 DevOps 敏捷开发理念,不断迭代开发解决并积累经验,最大程度发挥了自动化运维工具效能,不断增强企业核心竞争力。

步骤	操作项名称	操作项别名	匹配强度	报告突出标识	是否导出	是否已发布	输入参数
1	ZJ-登录检测查询	登录检测	0	是	是	是	
2	ZJ-登录检测查询_C	登录检测	0	是	是	是	SQLSELECT step=3
3	SV-从文件服务器下载文	SV-从文件服务器下载文	0	是	是	是	http://188.102.7.10
4	ZJ-ispath检查	检查脚本下载是否成功	0	是	是	是	SQLSELECT step=3
5	ZJ-修改配置项值(key=\	ZJ-修改配置项值(key=\	0	是	是	是	taskNo:SQLSELEC
6	ZJ-布尔类型检测执行_C	检测修改脚本参数id	0	是	是	是	SQLSELECT step=3
7	ZJ-修改配置项值(key=\	ZJ-修改配置项值(key=\	0	是	是	是	Host_ip:SQLSELEC
8	ZJ-布尔类型检测执行_C	检测修改脚本参数执行	0	是	是	是	SQLSELECT step=3
9	ZJ-执行shell脚本(sourc	ZJ-执行shell脚本(sourc	0	是	是	是	SQLSELECT step=3
10	SV-等待指定时间(秒)-sl	SV-等待指定时间(秒)-sl	0	是	是	是	120
11	ZJ-搜索到指定文件并删	删除合规脚本	0	是	是	是	SQLSELECT step=3

图 12 健康检查服务编排

执行情况									
配置编号: BL50003561									
设备IP: 10.216 73.26									
ZJ-登录检测 查询	ZJ-登录检测 查询_C	ZJ-打印网卡 信息详情	ZJ-网卡检查 errors	ZJ-网卡检查 errors_C	ZJ-网卡检查 dropped	ZJ-网卡检查 dropped_C	ZJ-网卡检查 结果	ZJ-BMC进程 详情	ZJ-BMC监控 告警
loginSuccess ful	T	1: lox mtu 65536 qdisc noqueue...	0 0 0...	T	1398 0 0...	F	F	patrol 32594 1 0 2015 ? 1-05:47:47...	T
ZJ-boolean 类型检测规则	ZJ-僵尸进程 数	ZJ-ink类型检 测规则_C	ZJ-系统进程 数	ZJ-系统进程 数检查结果	ZJ-时钟同步 详情	ZJ-时钟同步 检查	ZJ-时钟同步 检查结果	ZJ-内存性能 详情	ZJ-内存性能 检查
T	0	T	1855	T	remote relid st l when poll reach...	T	T	procs...	87%
ZJ-内存性能	ZJ-svlogd	ZJ-svlogd	ZJ-svlogd	ZJ-FTP文件	ZJ-messages	ZJ-messages	ZJ-FTTCPU	ZJ-cpu性能	ZJ-cpu性能检

图 13 健康检查结果报告

步骤	操作项名称	操作项别名	匹配强度	报告突出标识	是否导出	是否已发布	输入参数
1	ZJ-登录检测查询	登录检测查询	0	是	是	是	
2	ZJ-登录检测查询_C	登录检测查询_C	0	是	是	是	SQLSELECT step=3
3	ZJ-搜索文件是否存在	搜索文件issue是否存在	0	是	是	是	/etc/issue
4	ZJ-搜索文件是否存在_C	文件存在打印T并继续	0	是	是	是	SQLSELECT step=3
5	ZJ-检查配置项是否存在	检查配置项是否存在	0	是	是	是	/etc/issue;Authoriz
6	ZJ-string类型检测执行	配置项不存在打印T并继	0	是	是	是	SQLselect step=5
7	ZJ-备份成tar.gz包	打包成tar.gz包	0	是	是	是	issue.bak;/etc/issue
8	ZJ-ispath检查	打包成功打印T并继续执	0	是	是	是	SQLSELECT step=3
9	ZJ-配置文件备份	配置文件备份	0	是	是	是	SQLSELECT step=3
10	ZJ-isUri检查	备份成功打印T并继续执	0	是	是	是	SQLselect step=9
11	ZJ-打印文件内容	打印文件内容	0	是	是	是	/etc/issue
12	ZJ-追加一行到文件	追加一行到文件	0	是	是	是	Authorized users o
13	ZJ-打印文件内容	打印文件内容	0	是	是	是	/etc/issue

图 14 健康检查结果报告

参考文献

- [1]刘俊.IT 自动化运维的研究[J].电脑迷,2016(6).
- [2]何倩.IT 的运维管理与实现[J].环球市场,2016(13).
- [3]曾德华.IT 运维的自动化探索[J].电子世界,2014(12).
- [4]张华兵,刘昕林, 张海涛.IT 运维管理的自动化的研究[J].信息与电脑,2017(9).
- [5]Spring Cloud Projects Quick Start [EB/OL].[2017-12-31].https://projects.spring.io/spring-cloud/.
- [6]Spring Cloud Dalston 中文文档参考手册[EB/OL].[2017-04-11].https://springcloud.cc/spring-cloud-dalston.html.
- [7]杨小晔.IT 运维管理平台的设计与实现[D].北京邮电大学,2011: 1-113.
- [8]Leader-us.架构解密:从分布式到微服务[M].电子工业出版社,2017.
- [9]程梦瑶.DevOps 成企业获胜驱动力[J].软件和信息服务业,2015 (3).
- [10]文太益.华数 DevOps 建设之路[J].广播电视信息,2017(10).