互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

第 1 篇

开门见山：运维发展史

中国互联网从 20 世纪 90 年代开始形成，随即进入了快速发展阶段。第一阶段是以新浪、搜狐和网易为代表的门户网站时期，解决了信息尤其是新闻信息的传播问题。第二阶段是以腾讯、百度和阿里巴巴为代表的科技型时代，解决了社交、信息获取及电商等需求问题。第三阶段进入一个新的时期，从直播、网红到短视频类公司，以及人工智能公司的兴起，足以说明要解决的是心理需求。短短二十年，中国互联网发生了翻天覆地的变化，网民数量已经增长到 9 亿，其中使用手机的用户量已经达到 7 亿，中国也被称为具有全球最大消费能力的国家，很多企业只要做好中国市场，就能与其他做全球市场的企业比肩。科技的进步需要技术的支撑，运维技术也在不断地发生着变化。

运维的技术体系伴随着互联网的发展而愈加完善，从最开始的纯人工运维到脚本和开源工具的使用，再到平台建设，运维也紧跟时代的步伐，逐渐向智能化方向发展，通过应用机器学习的一些算法和模型将极大简化运维对数据的理解，大幅度提升效率。与此同时，运维工作也面临着非常大的挑战，在大数据场景下信息过载、多维度和更加复杂的业务环境，以及故障的定位、检测等工作，都已经超过了单凭人工能够完成的极限，新的技术需要被引进和普及。

近年来，人工智能技术备受关注，开始将 AI 引入 IT 运维领域，AIOps 的概念由此应运而生。Gartner 的报告宣称，到 2020 年，将近 50%的企业将会在他们的业务和 IT 运维方面采用 AIOps，智能化已是大势所趋。

本篇主要展现运维的发展历史和经历的不同历史阶段，以及运维工作的现状。本篇主要分两个章节：

- 第 1 章　运维现状
- 第 2 章　智能运维

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

第1章 运维现状

1.1 运维工程

1.1.1 认识运维

运维，通常指 IT 运维（IT Operations），是指通过一系列步骤和方法，管理与维护线上服务（Online Service）或者产品（Product）的过程。运维有着非常广泛的定义，在不同的公司不同的阶段代表不同的职责与定位，没有一个统一的标准。尤其是随着互联网的发展，运维的含义也在逐渐互联网化。互联网运维通常属于技术部门，与研发、测试、系统管理同为互联网产品的技术支撑，这种划分在国内和国外以及大小公司之间都会多少有一些不同。

运维的重点在于系统运行的各种环境，从机房、网络、存储、物理机、虚拟机这些基础的架构，到数据库、中间件平台、云平台、大数据平台，偏重的也不是编程，而是对这类平台的使用和管理。运维的水平可以成为衡量一个公司（IT 公司）技术实力的标准。

运维工程师（Operation Engineer），是指从事运维工作的工程师。运维工程师的工作范围非常广泛，包括服务器购买、租用和上架等基本管理，调整网络设备的配置管理和部署，服务器操作系统安装调试，测试环境和生产环境的初始化与维护，代码部署和管理（Git 和 SVN 等），设计和部署线上服务的监控与报警，服务安全性检测（防止漏洞和攻击），数据库管理和调优等。在大型公司中，运维工程师根据工作内容被细化为网站运维、系统运维、网络运维、数据库运维（DBA）、IT 运维、运维开发（DevOps[1]）、运维安全等方向。

运维的工作内容决定了对运维工程师的要求会非常高，运维工程师需要对服务器资源（CPU、内存、磁盘、网络 IO 等）非常了解，对 Linux 系统和常见的开源框架、工具非常熟悉，因此在

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

运维工程师中更容易诞生架构师，因为他们知道如何优化服务、如何使得资源利用最大化。

运维工程在国内也被称作 SRE[2]（Site Reliability Engineering，来自 Google），直接翻译为网站可用性工程。SRE 工程师需要具备算法、数据结构、编程能力、网络编程、分布式系统、可扩展架构、故障排除等各方面技能，其核心工作包括容量规划与实施、服务集群维护、系统容错管理、负载均衡、监控系统以及值班等，最终为产品上线后服务的稳定性负责，但是不负责具体的机器运维。

SRE 工程师的首要工作任务是保障 SLA（Service-Level Agreement，服务等级协议），它定义了对服务有效性的保障，比如对故障解决时间、服务超时等的保障。根据这个协议标准可以定义系统的可用性（Availability），这里需要掌握如下几个衡量指标。

1. 平均故障间隔时间（MTBF）

平均故障间隔时间（MTBF，Mean Time Between Failure），指相邻两次故障之间的平均工作时间。MTBF 通常是衡量一个产品可靠性的指标，这个间隔时间越短说明系统可靠性越差。

2. 平均修复时间（MTTR）

平均修复时间（MTTR，Mean Time To Repair），指产品由故障状态转为工作状态时修复时间的平均值，即故障修复所需要的平均时间。MTTR 值越低说明故障修复越及时。

3. 可用性（Availability）

可用性是系统架构设计中很重要的衡量指标。根据 GB/T3187—97 对可用性的定义，可用性是指在要求的外部资源得到保证的前提下，产品在规定的条件下和规定的时刻或时间区间内处于可执行规定功能状态的能力。它是产品可靠性、维修性和维修保障性的综合反映，如公式（1-1）所示。

$$\text{Availability} = \frac{\text{MTBF}}{\text{MTBF+ MTTR}} \tag{1-1}$$

业界一般通过 N 个 9 来对可用性进行量化。如表 1-1 所示为对可用性的量化及描述。

表 1–1　系统可用性量化指标

可用性级别	通俗叫法	停机时间/年	描述
99%	2 个 9	87.6 小时	基本可用性
99.9%	3 个 9	8.8 小时	较高可用性
99.99%	4 个 9	53 分钟	具有故障自动恢复能力的可用性

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

续表

可用性级别	通俗叫法	停机时间/年	描述
99.999%	5 个 9	5 分钟	极高可用性
99.9999%	6 个 9	31 秒	

可以看出，当系统可用性超过 3 个 9 时，全年停机状态持续时间低于 8.8 小时，此时一般可以称作高可用性系统，可用性越高，对系统的设计要求就越高。需要注意的是，在分布式系统中，可用性和性能应该遵循 CAP 原则，即数据一致性（Consistency）、可用性（Availability）和分区容忍性（Partition Tolerance）三者通常只能满足其中两者，无法兼得。

1.1.2 主要职责

运维的主要职责是在产品生命周期的各个阶段，维护系统的稳定性。运维的职责覆盖了产品从设计到发布、运行维护、变更升级及至下线的生命周期，在产品生产环境各个阶段的职责也不同，如表 1-2 所示。

表 1-2 运维在产品研发的生命周期中的主要职责

产品阶段	运维职责
设计阶段	○ 稳定性评估：主要是针对系统架构设计的合理性进行评估，包括是否存在单点、是否可容错、是否有强耦合等，同时评估能够满足上线发布并稳定运行的基本要求 ○ 资源评估：包括对所需的服务器资源、网络资源以及资源的分布等进行评估，同时把握相关产品对资源预算申请的合理性，控制资源使用成本 ○ 资源申请和准备
开发阶段	环境部署、依赖库及包管理、操作系统维护、数据库准备等
测试阶段	测试环境部署，稳定性评估，从系统的稳定性和可运维性的角度提出开发需求
部署阶段	自动化部署、稳定性检验、可扩展部署等
线上运行阶段	保证线上服务的稳定运行： ○ 实时监控：对服务运行的状态进行实时监控，随时发现服务的运行异常和资源消耗情况；输出重要的日常服务运行报表，以评估服务/业务整体运行状况，发现服务隐患 ○ 故障处理：对服务出现的任何异常进行及时处理，尽可能避免问题扩大化甚至中止服务 ○ 容量管理：包括服务规模扩张后的资源评估、扩容、机房迁移、流量调度等规划和具体实施
下线/回滚阶段	由于产品效果不如预期或者其他原因，产品可能需要做下线或者回滚处理，在这个过程中运维工程师主要做好资源回收的工作，将服务终止，对机器/网络等资源进行回收

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

可以看出，运维工作贯穿于产品研发的各个环节。因此，运维工程师的日常工作主要包括：

- 产品技术方案评估。
- 资源预估、申请和管理。
- 环境部署、环境准备。
- 产品上线、下线及回滚，服务在线发布/升级产品。
- 监控线上的服务质量。
- 响应异常/处理突发故障。
- 和相应产品线的研发和测试团队协调处理产品问题。
- 与产品、技术、测试等团队沟通协作。
- 对系统实时数据进行分析。
- 建立工具或平台，保障系统的稳定性和可靠性。

1.1.3　运维技术

在产品的整个生命周期中运维工作重要而广泛，但运维工程师的职责不局限于这部分工作，还需要总结工作中遇到的问题，抽取出相关的技术方向，研发相关的工具和平台，以支持/优化业务的发展并提高运维的效率。

运维工程师所需的技术体系根据其专业方向而异，但对计算机系统架构、操作系统、网络技术的掌握是基本要求。例如，可能需要熟练掌握 Linux 操作系统的使用，熟练使用各种脚本工具来处理日常工作任务，精通 TCP/IP 协议栈以排查一个大规模网络系统中的流量异常问题等。更进一步的，需要形成一套软件可运维性方面的经验，以此作为后续工作的指导。

一个运维工程师在初期阶段的目的是掌握、维护一套系统所需的所有软硬件知识和经验。在进阶阶段需要能够设计开发一套基础的体系软件，以支撑业务系统的稳定可靠运行，即开发服务于软件的软件，以支持更大规模的业务系统，提高运维生产力。在最高阶段反要能作用于软件系统的构建和运行阶段，使得系统从诞生阶段起即具有天然的可运维性，以最大化系统的生产力，同时最小化对外部支撑资源的依赖。

运维以技术为基础，通过技术保障产品提供更高质量的服务。运维工作的职责及在业务中的位置决定了运维工程师需要具备更加广博的知识和深入的技术能力，需要掌握的技术非常广

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

泛，表 1-3 中列出了一些运维涉及的常见技术和框架。

表 1–3 运维涉及的常见技术和框架

功能描述	技术和框架
操作系统	Linux、Ubuntu、Windows、CentOS、Redhat 等
Web Server	以 Nginx 为典型代表，以及 Apache 等
网络工具	tcpcopy、curl 等
监控和报警系统	Grafana、Zabbix 等
自动部署	Ansible、sshpt、salt、Jenkins 等
配置管理及服务发现	Puppet、Consul、Zookeeper 等
负载均衡	LVS、HAProxy、Nginx 等
传输工具	Scribe、Flume 等
集群管理工具	Zookeeper 等
数据库	MySQL、Oracle、SQL Server 等
缓存技术（Cache）	Redis、Memcache 等
消息队列	Kafka、ZeroMQ 等
大数据平台	HDFS、MapReduce、Spark、Storm、Hive 等
大数据存储	HBase、Cassandra、MongoDB、LevelDB 等
时序数据（OLAP 平台）	Druid、OpenTSDB 等
容器	LXC、Docker、K8s 等
虚拟化	OpenStack、Xen、KVM 等

1.2 运维发展历程

伴随着互联网、移动互联网的发展，运维的发展大致经历了如下四个重要阶段。

1.2.1 人工阶段

早期运维处于人工阶段，这个时候的运维是一个通称，负责从机房、服务器选型，软硬件初始化，服务上下线，配置监控，盯监控等，运维和开发之间没有太明确的分工，基本是遇到什么问题解决什么问题。在这个阶段，因为服务器少、业务需求简单，只需要少数几个运维工程师就能完成运维工作，运维也基本不会成为企业发展的瓶颈。

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

1.2.2　工具和自动化阶段

通过人工维护和管理的方式，比较适合简单的业务，随着 IT 尤其是互联网的迅速发展，企业所承担的业务愈发复杂，运维也进入了第二阶段，即工具和自动化阶段。为了提升运维工作效率，简化操作流程，运维工程师开始将部分运维操作及重复性工作流程编写成脚本自动执行。

工具的产生是运维自动化的一个典型的标志，尤其是开源项目的逐渐兴起，大量工具被开源，很多项目都可以在 GitHub 上找到。随着容器技术的兴起，许多新的专门运行容器的 Linux 发行版本也出现了。Docker 及相关虚拟化技术的不断发展、K8s（Kubernetes，自动化容器管理工具，具备集群管理、任务调度等强大的功能）等技术的发展、助力云服务（Cloud，包括私有云、公有云）的快速发展，也进一步为运维自动化带来了极大的便利。

在工具和自动化阶段，运维工程师有一部分掌握了一定的开发能力，将日常的工作通过自动执行程序来完成，以替代人工，效率也逐渐比单纯人工运维更高，同时出错的概率逐渐降低。此时已经有掌握 Python、shell 等开发语言和工具的运维工程师逐渐转向运维开发的角色。

1.2.3　平台化阶段

在平台化阶段前，脚本和工具是分散的，不易管理，同时也需要人工干预，随着业务变得更加复杂，对大量脚本的管理是低效和复杂的。之后，将自动化脚本和工具进行整合，从系统层面构建更加易用和高效的运维管理工具，已经成为趋势，这也就是运维平台化。

围绕开源工具、开源平台，大中型企业开始结合业务构建自己的运维平台，包括监控平台、报警平台和自动化平台等，这些平台在一定程度上提高了产品开发效率和测试效率，降低了运维成本，并且降低了系统出风险的概率，提高了系统可用性。

在表 1-4 中简单描述了企业围绕开源框架构建运维体系的各类平台。

表 1–4　运维平台示例

平台类型	典型的基础开源平台	说明
监控系统	ELK、Grafana、OpenTSDB 等	围绕数据搜集、ETL、搜索、指标管理、可视化展示等，快速构建监控系统
报警系统	Zabbix	具有报警管理、报警聚合、提醒等功能，同时与监控系统配合，构成系统稳定性的保障体系
自动化平台	GitLab、Jenkins、Ansible、sshpt、salt、Docker	具有代码托管、编译、打包、环境部署、安装和回滚、灰度等基础功能，同时结合监控和报警系统构建动态扩缩容、自动化降级等系统

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

可以看出，具有平台化思想的开源工具几乎覆盖了运维的全部维度，从监控系统、报警系统到自动化平台都有非常优秀的开源技术框架。

1.2.4　智能运维阶段

这个阶段是智能化的，当基础设施固定下来后，运维模式最终也会固定下来，这些模式会把可用性、扩容等场景在内的诸多运维方案包含进来，把开发平滑地加入运维架构。AIOps（Algorithmic IT Operations）最早由 Gartner 定义为采用人工智能算法（AI 和机器学习），利用机器解决已知的问题和潜在的运维问题的一种技术解决方案。这里值得注意的是，根据 Gartner 定义，AIOps 中的 AI 并不是 Artificial Intelligence（人工智能），而是广义的算法。本书中也将遵从此定义，并且不严格区分智能运维与 AIOps。

AIOps 使用分析理论和机器学习等方法，分析和处理各种操作工具、服务和设备产生的大量数据。它能够自动发现问题，并且能够实时对问题做出反应。AIOps 建立在大数据与机器学习（Machine Learning）基础之上，如图 1-1 所示。

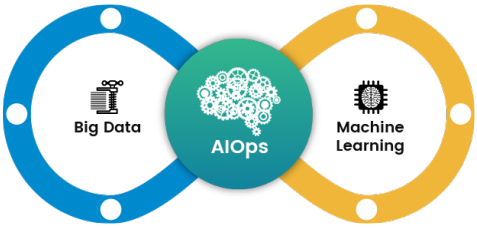

图1-1　基于大数据与机器学习的AIOps

举一个简单的例子。通常的报警策略是设置一个阈值范围（上、下界），当某个指标超出这个阈值范围时，则触发报警。这也是最直接、最简便的办法。然而，对于一些特定场景下的报警，设置可能没有这么简单。如图 1-2 所示是微博广告某产品某天的广告曝光次数走势图，从图中可以看出，每天早上 4 点到 6 点达到一天的最低，上午将近 11 点达到一天的最高。如果依然按照固定的阈值设置报警是非常不准确的，我们需要通过历史数据智能化地拟合出一条趋势线，以这条线的上、下界一定范围作为动态的报警阈值，以达到更加准确的报警。

另外，智能运维可以被用于故障分析，以快速定位问题，降低企业的损失。在 AI 中我们使用到的各类算法，比如基于指数平滑的二次平滑、三次平滑算法，基于 ARIMA 的算法，基于深度学习的前馈神经网络、循环神经网络（RNN）算法等，已经比较成熟，并大量使用在其他研究领域，比如图形图像处理、语音识别等领域。所以，在算法上，我们在很早之前就应该具

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

备了这方面的理论基础。

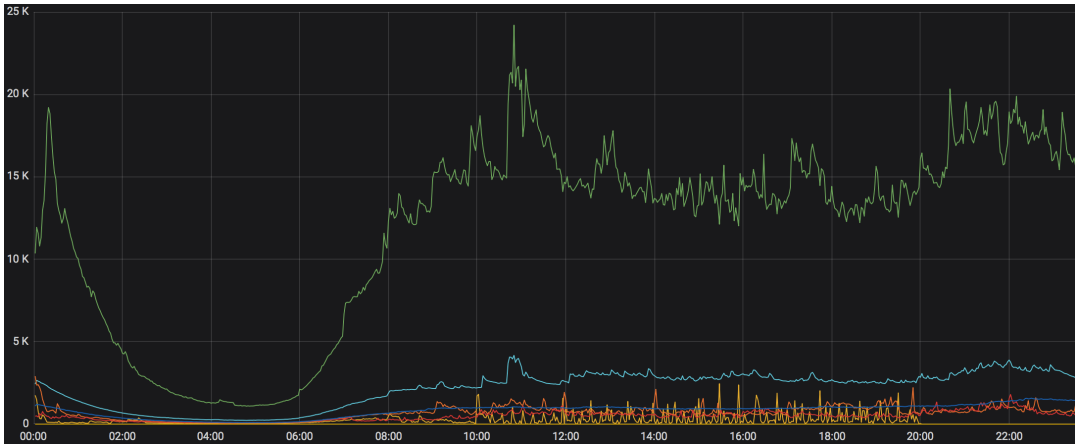

图1-2　微博广告某产品某天的广告曝光次数走势图

在计算能力上，目前我们看到，基于大数据技术的数据处理能力已经足够，Hadoop、Spark 等生态，包括时序数据的处理能力，都已经能够支撑智能化的复杂计算场景。

就目前来看，国内的百度、搜狗、宜信、阿里巴巴、微博等都已经探索尝试了 AIOps，并且取得了不错的收益。在 2017 年 InfoQ 举办的 QCon 全球架构大会上，已经有不少与 AIOps 相关的议题，感兴趣的读者可以关注。

1.3　运维现状

1.3.1　故障频发

快速解决故障、降低故障率、不断提高系统可用性是运维的非常关键的职责，然而在大数据场景下，系统越来越复杂，系统维护成本越来越高。Gartner Group 有一个调查显示，在 IT 项目经常出现的问题中，源自技术或产品（包括硬件、软件、网络、电力失常及天灾等）的问题只占 20%，但流程失误方面却占 40%，人员疏失方面也占到了 40%。这些年来，企业通过自动化运维平台，以及 DevOps 等协作理念，逐步解决了 Gartner Group 提到的与流程失误和人员疏失相关的 80%的问题，但企业系统故障带来的损失也是非常大的。

表 1-5 统计了在 2015 年到 2017 年间，国内外知名互联网企业出现的故障。可以看出，即使是顶级的企业，也面临着各类故障导致的用户流失、财产损失等巨大的风险。

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

表 1–5 国内外知名互联网企业故障列表（部分）

时间	企业/产品线	故障描述
2015 年	阿里巴巴/支付宝	支付宝因杭州机房网络光纤被挖，导致数小时部分用户业务不可用
	携程	携程网瘫痪事件，全网业务中断 12 小时
	知乎	知乎机房故障，影响系统使用近 2 小时
	阿里云	阿里云香港节点宕机，业务中断 13 小时
	七牛云	七牛云存储服务故障，业务中断 83 分钟
2016 年	GitHub	GitHub 全球服务中断，所有托管在上面的开源项目受到影响，影响时长超过 6 小时
	亚马逊	亚马逊电商网站中断访问，亚马逊电子商务主网站及云计算服务受到影响，持续 20 分钟
	阿里巴巴/支付宝	由于华南一处机房出现故障，支付宝出现故障，无法支付，部分用户无法在线上或线下通过支付宝进行支付购买
	腾讯微信	腾讯微信故障，朋友圈无法打开，微信图文也无法打开，持续 2 小时
	Google	谷歌云存储及文件备份服务器服务中断，部分云用户在访问服务器时会显示“服务器遇到错误，请稍后再试”的提示，持续 20 分钟
2017 年	IBM	2017 年年初，IBM 云的信用度受到影响，客户用于访问其 Bluemix 云基础架构（以前称为 SoftLayer）的一个管理网站服务中断了数小时
	GitLab	GitLab 极受欢迎的线上代码库——GitLab.com 遭遇了 18 小时的服务中断，最终无法完全修复。故障原因是员工在维护过程中从错误的数据库服务器中删除了数据库目录
	亚马逊	亚马逊 RDS 服务上的 MySQL 数据库文件大小限制引发了 Pinterest 服务器的长时间宕机
	亚马逊 AWS	一位 AWS 工程师试图调试亚马逊的弗吉尼亚数据中心 S3 存储系统，但输入了一个错误指令，导致许多互联网——包括 Slack、Quora 和 Trello 等众多企业平台宕机 4 小时
	微软 Azure	微软 Azure 公有云出现超过 8 小时的存储可用性问题，主要影响到美国东部的客户，导致有些用户无法配置新的存储空间或访问本地现有资源。之后，微软工程团队确认原因为断电导致的存储集群不可用
	今日头条	因服务器故障，今日头条全站及头条号后台全部无法访问
	百度	2017 年 2 月 28 日晚，百度出现大规模宕机事件
	新浪微博	新浪微博系统出现故障，网友通过 ping 命令测试 IP 地址的可用性时发现，新浪微博的服务器已经失去响应，此次新浪微博宕机时间接近 1 小时

1.3.2 系统复杂性

当前的 IT 项目基础设施环境与 5 年前已经无法同日而语，更不用说 10 年前了。近几年，

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

随着云计算、微服务等技术的流行，以及互联网业务的迅速发展，运维人员要关注的服务数量也呈现指数级增长，自动化运维虽然提升了效率，解决了一部分问题，但也遇到了新的难题，比如面对繁多的报警信息，运维人员应该如何处理；当故障发生时，又如何能够迅速定位问题。

系统的复杂性是业务复杂性的结果，我们可能无法直接评估当前互联网企业的系统有多复杂，但可以从客观数据反映出来。

从代码量角度看，Linus Torvalds 最初发布的 Linux 内核版本源码大概 1 万行，而随着操作系统的发展，当前 Linux 内核源码在千万行级别，完整的 Linux 操作系统的代码量过亿行。2015 年 Google 披露全部代码量大概为 20 亿行，按 Google 工程师每天编写 150 行代码估算，每天新增约百万行代码；国内一线企业的代码量也达到 10 亿行级别。这样庞大的代码量，从侧面反映了系统的复杂性。

从产品和业务角度看，仅仅微博广告系统中的业务监控指标就已经超过 10 万个，微博的监控指标量达百万级，这样量级的背后是一套非常庞大的业务系统支撑。BAT（百度、阿里巴巴、腾讯）这样的一线互联网企业，产品线非常多，系统就更加庞大了。如图 1-3 所示是百度首页展示的一部分产品列表，全部产品超过 105 个。

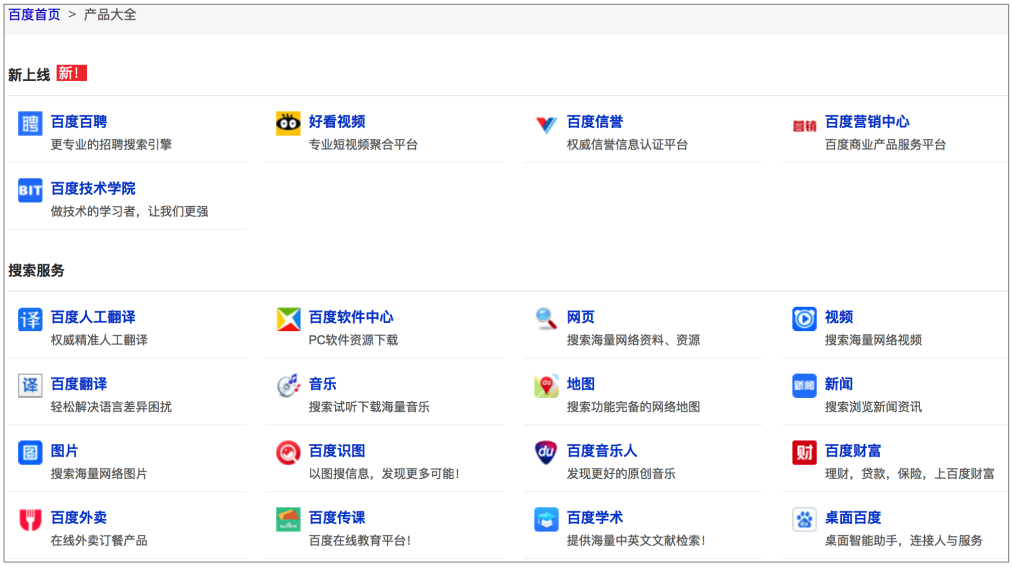

图1-3　百度产品大全（部分）

如图 1-4 所示是某云服务平台包括的产品列表，有超过 100 个复杂的产品服务。这样庞大

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

的产品系列背后的业务逻辑会非常复杂，对系统稳定性有非常苛刻的要求。

图1-4　某云服务平台包括的产品列表（部分）

1.3.3　大数据环境

随着大数据政策环境和技术手段的不断完善，大数据行业应用持续升温，中国企业级大数据市场进入了快速发展时期。互联网、电信、金融等开始实际部署大数据平台并付诸实践，带动了软件、硬件和服务市场的快速发展。

中国信息通信研究院公布的《中国大数据发展调查报告（2017）》显示，2016 年约 70%的企业拥有的数据资源总量在 50~500TB 之间，18.4%的企业数据量在 500TB 以上，与 2015 年相比，企业资源总量呈增长趋势。

数据规模的增大，一方面反映了系统的复杂程度；另一方面也反映了监控系统、自动化系统等运维平台的复杂程度。在大数据场景下，运维面临的主要挑战有以下几个方面。

1. 数据采集

数据采集是大数据分析处理的基石，其核心一是要保证数据的完整性；二是要保证数据的准确性；三是要保证数据的实效性。数据完整性要求采集系统能够尽可能搜集到足够多和完整

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

的信息，在采集过程及预处理过程中都不能丢数据。数据准确性要求在数据采集过程中，不能因为预处理而导致数据不一致，影响后续的分析和决策。数据实效性一方面要求数据采集要做到实时或者准实时，采集系统导致的延时率尽可能低，性能尽可能高；另一方面要求在数据预处理阶段，保留数据尤其是时序数据时间效应。这里的时序数据时间效应是指某个指标以某个固定时间间隔的波动变化情况，这个波动在一定程度上反映出系统的运行状态，数据采集器要在系统承载能力允许的前提下缩小时间间隔。比如对于请求量的走势，采集器可以 1 秒、5 秒、15 秒甚至 1 分钟采集一次数据，这样的时间间隔会带来不同的计算误差，最理想的情况是时间间隔越小越好，但带来的问题是数据规模的成倍增长，以及对后续数据分析的极大挑战。

2. 数据存储

目前大数据的原始数据及数据仓库存储介质一般在 HDFS 上，为了提升数据分析能力，部分数据也存储在 HBase、Hive、Redis 等集群上。

各个业务系统不断在产生和制造大量的数据，数据被分析和处理再加工再存储，在每个环节数据都会被复制，一般情况下，原始数据规模最大，越接近数据业务分析层，数据规模越小。

如表 1-6 所示是微博广告某产品线 2016 年数据仓库各层的存储规模。

表 1–6　微博广告某产品线 2016 年数据仓库各层的存储规模

仓库分层类型	每天增量（TB）
仓库分层——ODS（原始层）	4.053
仓库分层——DWD（明细层）	5
仓库分层——DWS（聚合层）	3
仓库分层——DM（集市层）	2
各层共计	14

如表 1-7 所示是微博广告某产品线存储 1.5 年总体数据规模。

表 1–7　微博广告某产品线存储 1.5 年总体数据规模

类型	数量
存储日增量（TB）	12.5643
增量系数	1.2
存储时长（年）	1.5
天数	365

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

续表

类型	数量
副本数	3
压缩比	0.33
存储量（TB）	8172

可以看出，数据存储规模是一个非常大的挑战。其中为了保证数据的可用性，一般都会有至少 3 个数据副本（设置 HDFS 副本数是为了保证数据可用性），同时为了节约存储资源，通常采用特定的数据压缩算法来降低存储量。

3. 分析和建模

数据分析和建模体现在数据集的大规模计算上，模型的训练是非常消耗资源的，数据需要在不同的计算节点之间进行复制和传播，都要耗用存储资源和网络带宽，而对数据的处理则需要耗用 CPU 和内存资源。

1.4 本章小结

在大规模、复杂架构的催生下，运维技术不断变化和发展。自动化运维被推到一个新的高度，给传统企业带来了福音，给基础运维带来了巨大的挑战与机遇，同时也给越来越多的企业带来了新的抉择；开源技术的飞跃发展、脚本语言的进化等，也给运维行业带来了革命性的影响。

1.5 参考文献

[1] DevOps 维基百科定义：https://en.wikipedia.org/wiki/DevOps

[2] SRE 维基百科定义：https://en.wikipedia.org/wiki/Site_reliability_engineering

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

第 2 章 智能运维

得益于 IT 外包服务的发达，现在的运维已经不包括搬机器上架、接网线、安装操作系统等基础工作，运维人员一般会从一台已安装好指定版本的操作系统、分配好 IP 地址和账号的服务器入手，工作范围大致包括：服务器管理（操作系统层面，比如重启、下线）、软件包管理、代码上下线、日志管理和分析、监控（区分系统、业务）和告警、流量管理（分发、转移、降级、限流等），以及一些日常的优化、故障排查等。

随着业务的发展、服务器规模的扩大，才及云化（公有云和混合云）、虚拟化的逐步落实，运维工作就扩展到了容量管理、弹性（自动化）扩缩容、安全管理，以及（引入各种容器、开源框架带来的复杂度提高而导致的）故障分析和定位等范围。

听上去每一类工作都不简单。不过，好在这些领域都有成熟的解决方案、开源软件和系统，运维工作的重点就是如何应用好这些工具来解决问题。

传统的运维工作经过不断发展（服务器规模的不断扩大），大致经历了人工、工具和自动化、平台化和智能运维（AIOps）几个阶段。如前文所述，这里的 AIOps 不是指 Artificial Intelligence for IT Operations，而是指 Algorithmic IT Operations（基于 Gartner[1]的定义标准）。

基于算法的 IT 运维，能利用数据和算法提高运维的自动化程度和效率，比如将其用于告警收敛和合并、Root 分析、关联分析、容量评估、自动扩缩容等运维工作中。

在 Monitoring（监控）、Service Desk（服务台）、Automation（自动化）之上，利用大数据和机器学习持续优化，用机器智能扩展人类的能力极限，这就是智能运维的实质含义。

智能运维具体的落地方式，各团队也都在摸索中，较早见效的是在异常检测、故障分析和定位（有赖于业务系统标准化的推进）等方面的应用，后面的章节会具体涉及。智能运维平台

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

逻辑架构如图 2-1 所示。

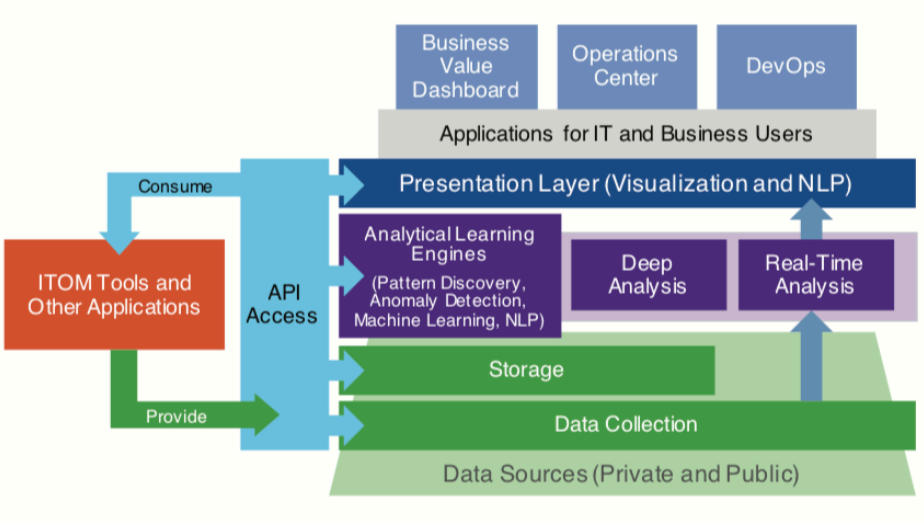

图2-1 智能运维平台逻辑架构图

智能运维决不是一个跳跃发展的过程，而是一个长期演进的系统，其根基还是运维自动化、监控、数据收集、分析和处理等具体的工程。人们很容易忽略智能运维在工程上的投入，认为只要有算法就可以了，其实工程能力和算法能力在这里同样重要。

那么，智能运维在工程方面会有哪些难题呢？这些难题是否会随着智能运维的深入应用而得到一定程度的解决呢？下面的章节会逐步展开这些问题，并提供一些解决方案。

2.1 海量数据的存储、分析和处理

运维人员必须随时掌握服务器的运行状况，除常规的服务器配置、资源占用情况等信息外，业务在运行时会产生大量的日志、异常、告警、状态报告等，我们统称为“事件”。通常每台服务器每个时刻都会产生大量这样的“事件”，在有数万台服务器的场合下，每天产生的“事件”数量是数亿级的，存储量可能是 TB 级别的。

在过去，我们通常采用的方法是将日志保留在本地，当发现问题时，会登录出问题的服务器查看日志、排查故障，通过 sar、dmesg 等工具查看历史状态；监控 Agent 或者脚本也会将部分状态数据汇报到类似于 Zabbix 这样的监控软件中，集中进行监控和告警。

当服务器规模越来越大时，如何统一、自动化处理这些“事件”的需求就越来越强烈，毕

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

竟登录服务器查看日志这种方式效率很低，而成熟的监控软件（比如 Zabbix、Zenoss 等）只能收集和处理众多“事件”当中的一部分，当服务器数量多了以后，其扩展能力、二次开发能力也非常有限。在具体实践中，当监控指标超过百万级别时，就很少再使用这种单一的解决方案了，而是组合不同的工具和软件，分类解决问题。

在通用设计方法中，有“大工具、小系统，小工具、大系统”的说法，这也符合 UNIX 的设计哲学，每个工具只做好一件事，一堆小工具组合起来可以完成很复杂的工作。如果使用的是一些大工具或者系统，表面上看功能很多，但是当你想处理更复杂的业务时，就会发现每一个功能都不够用，而且还很难扩展，它能做多“大”事取决于它的设计，而不是你的能力。

一个由典型的小工具组成的大系统，任何一个部分都可以被取代，你完全可以用自己更熟悉的工具来做，而且对工具或者组件的替换，对整体没有太大影响。

一提到海量数据的存储、分析和处理，大家就会想到各种各样的大数据平台。是的，大数据平台确实是用来处理海量数据的，但反过来不见得成立，对海量数据的分析和处理，并不总是或者只依赖大数据平台。

“分类”这个词听上去朴实无华，然而处理复杂问题最基本的方法就是分类，甚至“分类方法”也是机器学习非常重要的组成部分。“海量数据处理”这是一个宏大的命题，听上去让人一头雾水，但当你对“事件”或者需要处理的问题分类后，每一部分看上去就是一个可以解决的问题了。

后面的章节会详细介绍如何对海量“事件”进行分类和处理。

- 实时数据和非实时数据。
- 格式化数据和非格式化数据。
- 需要索引的数据和只需要运算的数据。
- 全量数据和抽样数据。
- 可视化数据和告警数据。

每一个分类都对应一种或多种数据处理、分析和存储方式。也可以说，当你对数据、需求完成分类后，基本的框架也就定了下来，剩下的工作就是集成这些工具。

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

2.2 多维度、多数据源

如图 2-2 所示，这是一个多维度模型示例。真实世界的情况是（至少按弦理论学家所说的是），除我们可以感知的 3 个“延展维”外，还有 6 个“蜷缩维”，它们的尺寸在普朗克长度的数量级，因此我们无法感知到。

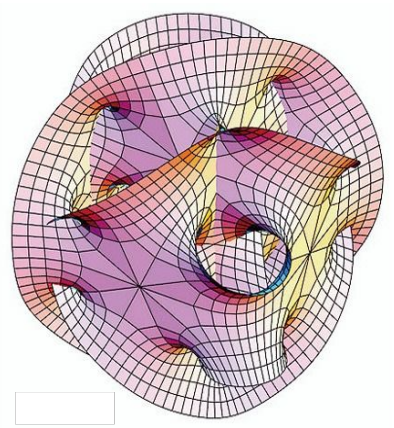

图2-2 多维度模型示例

当然，运维数据中的“多维度”，还没有复杂到这样难以理解。

在相对复杂的业务场景下，一个“事件”除包含我们常用的“时间”（何时发生）、“地点”（哪个服务器/组件）、“内容”（包括错误码、状态值等）外，还应当包含地区、机房、服务池、业务线、服务、接口等，这就是多维度数据。

很多时候，数据分析人员可能要使用各种维度、组合各种指标来生成报告、Dashboard、告警规则等，所以是否支持多维度的数据存储和查询分析，是衡量一个系统是否具有灵活性的重要指标。

对多维度数据的处理，很多时候是一个协议/模型设计问题，甚至都不会牵扯具体的分析和处理框架，设计良好的协议和存储模型，能够兼顾简洁性和多维度。

在后面的章节中，我们会介绍协议/模型设计中的多维度。

- 在单一 Key 中包含维度信息。
- 在 Tag 中标注不同维度。

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

不同的设计理念会对应不同的处理模型，没有优劣之分，只有哪个更合适的区别。

多数据源或者说异构数据源已经很普遍了，毕竟在复杂场景下并不总是只产生一种类型的数据，也不是所有数据都要用统一的方式处理和存储。

在具体的实践中，通常会混合使用多种存储介质和计算模型。

- 监控数据：时序数据库（RRD、Whisper、TSDB）。
- 告警事件：Redis。
- 分析报表：MySQL。
- 日志检索：Elasticsearch、Hadoop/Hive。

这里列出的只是一部分。

如何从异构的多数据源中获取数据，还要考虑当其中某个数据源失效、服务延迟时，能否不影响整个系统的稳定性。这考量的不仅仅是各种数据格式/API 的适配能力，而且在多依赖系统中快速失败和 SLA 也是要涉及的点。

多数据源还有一个关键问题就是如何做到数据和展现分离。如果展现和数据的契合度太高，那么随便一点变更都会导致前端界面展现部分的更改，带来的工作量可能会非常大，很多烂尾的系统都有这个因素存在的可能性。

2.3　信息过载

DDoS（分布式拒绝服务）攻击，指借助于客户/服务器技术，将多台计算机联合起来作为攻击平台，对一个或多个目标发动攻击。其特点是所有请求都是合法的，但请求量特别大，很快会消耗光计算资源和带宽，图 2-3 展示了一个 DDos 攻击示例。

当我们的大脑在短时间内接收到大量的信息，达到了无法及时处理的程度时，实际上就处于“拒绝服务”的状态，尤其是当重大故障发生，各种信息、蜂拥而至的警报同时到达时。

典型的信息过载的场景就是“告警”应用，管理员几乎给所有需要的地方都加上了告警，以为这样即可高枕无忧了

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

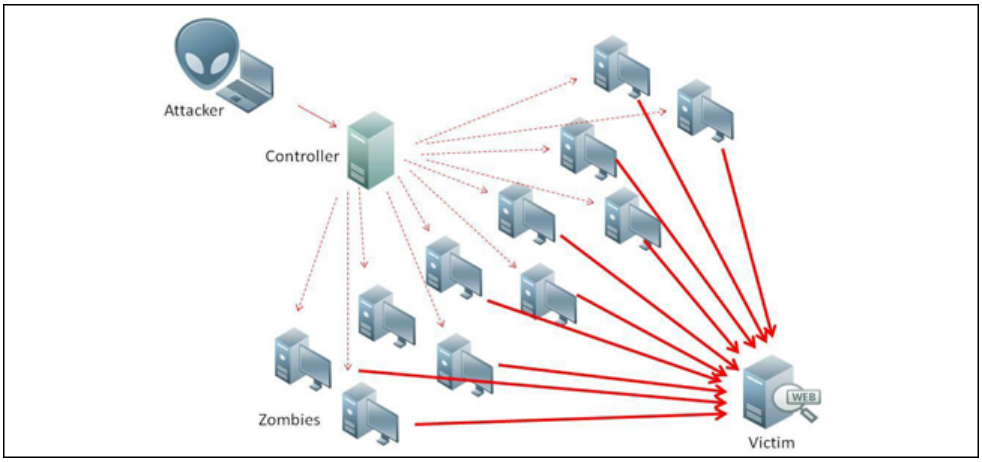

图2-3 DDoS攻击示例

然而，接触过告警的人都知道，邮件、短信、手机推送、不同声音和颜色提醒等各种来源的信息可以轻松挤满你的空间，很多人一天要收上万条告警短信，手机都无法正常使用，更别谈关注故障了。

怎样从成千上万条信息中发现有用的，过滤掉重复的、抖动性的信息，或者从中找出问题根源，从来都不是一件容易的事情，所以业界流传着“监控容易做，告警很难报”的说法。

还有一个场景就是监控，当指标较少、只有数十张 Dashboard 时，尚且可以让服务台 24 小时关注，但是当指标达到百万、千万，Dashboard 达到数万张时（你没看错，是数万张图，得益于 Grafana/Graphite 的灵活性，Dashboard 可以用程序自动产生，无须运维工程师手工配置），就已经无法用人力来解决 Dashboard 的巡检了。

历史的发展总是螺旋上升的，早期我们监控的指标少，对系统的了解不够全面，于是加大力度提高覆盖度，等实现了全面覆盖，又发现信息太多了，人工无法处理，又要想办法降噪、聚合、抽象，少→多→少这一过程看似简单，其实经过了多次迭代和长时间的演化。

在后续章节中，将为读者介绍这类问题在实践中的解决方法。

- 数据的聚合。
- 降低维度：聚类和分类。
- 标准化和归一化。

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

有些方法属于工程方法，有些方法属于人工智能或机器学习的范畴。

2.4　复杂业务模型下的故障定位

业务模型（或系统部署结构）复杂带来的最直接影响就是定位故障很困难，发现根源问题成本较高，需要多部门合作，开发、运维人员相互配合分析（现在的大规模系统很难找到一个能掌控全局的人），即使这样有时得出的结论也不见得各方都认可。

在开发层面，应对复杂业务的一般思路是采用 SOA、微服务化等，但从运维的角度讲，完成微服务化并没有降低业务的复杂度（当然结构肯定变清晰了）。

在这里，又不得不强调工程能力的重要性。在复杂、异构和各种技术栈混杂的业务系统中，如果想定位故障和发现问题，在各个系统中就必须有一个可追踪、共性的东西。然而，在现实中若想用某个“体系”来一统天下，则基本不可能，因为各种非技术因素可能会让这种努力一直停留在规划阶段，尤其是大公司，部门之间的鸿沟是技术人员无法跨越的。

所以，下面给出的几种简单方法和技术，既能在异构系统中建立某种关联，为智能化提供一定的支持，又不要求开发人员改变技术栈或开发框架。

- 日志标准化：日志包含所约定的内容、格式，能标识自己的业务线、服务层级等。
- 全链路追踪：TraceID 或者 RequestID 应该能从发起方透传到后端，标识唯一请求。
- SLA 规范化：采用统一的 SLA 约定，比如都用“响应时间”来约定性能指标，用“慢速比”来衡量系统健康度。

当这些工程（自动化、标准化）的水平达到一定高度后，我们才有望向智能化方向发展。

故障定位又称为告警关联（Alarm Correlation)、问题确定（Problem Determination）或根源故障分析（Root Cause Analysis），是指通过分析观测到的征兆（Symptom），找出产生这些征兆的真正原因。[2]

在实践中通常用于故障定位的机器学习算法有关联规则和决策树。

还有很多方法，但笔者也在探索中，所以无法推荐一个“最佳”方法。究竟什么算法更适合，只能取决于实践中的效果了。

需要注意的是，并不是用了人工智能或机器学习，故障定位的效果就一定很好，这取决于

互动出版网 www.china-pub.com 专业图书网上第一专营店 www.china-pub.com 独家提供样章

很多因素，比如特征工程、算法模型、参数调整、数据清洗等，需要不断地调整和学习。还是这句话：智能化的效果不仅仅取决于算法，工程能力也很重要，而且好的数据胜过好的算法。

2.5 本章小结

本章介绍了智能运维的定义和发展现状，智能运维需要解决的问题有：海量数据存储、分析、处理，多维度，多数据源，信息过载，复杂业务模型下的故障定位。在每一类问题后面，都给出了经过实践证明的解决方案和思路，同时也说明了为什么要这么做，以及在工程和算法上会遇到的问题。

2.6 参考文献

[1] Gartner.https://blogs.gartner.com/andrew-lerner/2017/08/09/aiops-platforms/

[2] 张成，廖建新，朱晓民. 基于贝叶斯疑似度的启发式故障定位算法