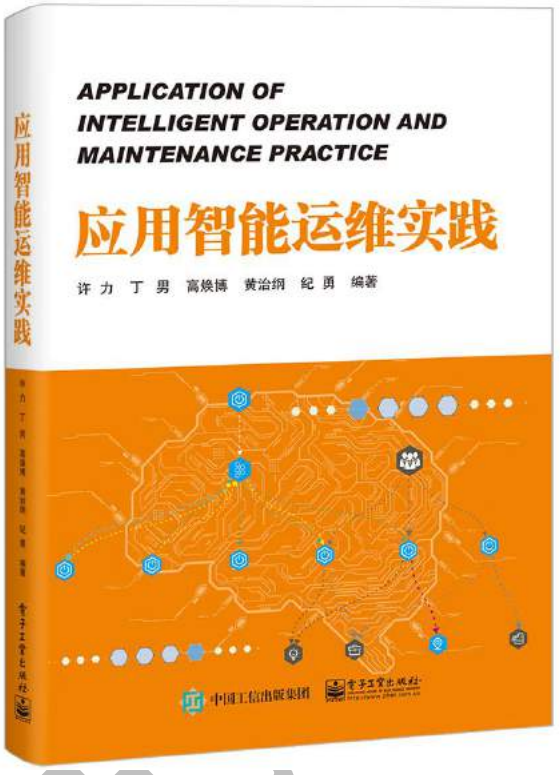
阿里云开发者社区
ALIBABA CLOUD DEVELOPER COMMUNITY

应用智能运维实践（试读版）

从应用运维和智能运维角度出发，详解未来应用运维场景新需求

许 力　丁 男　高焕博　黄治纲　纪 勇　编著

电子工业出版社



前言

我们正处在第三次信息技术浪潮的青萍之末，如今，几乎所有企业都面临如何利用新一代信息技术对外提升企业客户价值、对内优化生产流程的问题。虚拟化、云计算、大数据、物联网、人工智能、区块链等新技术如雨后春笋，新一代信息技术在金融、汽车、医疗等各行业落地应用的案例也层出不穷。

以智能、互联为主要特征的第三次信息技术浪潮将在提升生产力的同时，改变应用及其运维方式。物联网（Internet of Things，IoT）、车联网（Internet of Vehicle，IoV）等新一代信息技术已经开始改变产品或服务的设计、生产、营销、交付和售后支持过程。哈佛商学院院长迈克尔·波特教授预言，第三次信息技术浪潮将“有潜力成为目前影响最深远的技术浪潮，相比前两次会激发更多的创新，获取更大幅度的生产收益增长和经济增长”。

然而，新一代信息技术在赋能数字信息系统应用数据处理、智能决策支持和态势感知等能力以大幅度提升企业生产力的同时，系统自身复杂度急剧上升，应用运维难度和成本快速增加。更严重的是，很多企业在规划设计应用系统，或者在做互联网化系统升级改造的过程中，往往忽视对应用运行期的状态监视、风险管理、容量规划等运维保障系统和过程的建设，系统故障和宕机频率快速升高，人工运维成本飙升。

在数字时代，一切都依赖于应用系统稳定可靠的运行，缺少匹配新型信息系统应用的应用智能运维系统的支撑，新技术将很难发挥其应有的价值。要解决新技术演进带来的应用运维问题，则需要通过新技术来升级应用运维系统。目前，大多数企业都缺少能够应对未来来自应用运行期稳定性和性能方面的挑战的运维系统。为了帮助更多企业建设新一代应用智能运维平台，解决应用系统运维管理问题，将先进信息技术转化为生产力，本书从实际需求出发，总结分享了作者十余年来从事企业信息系统建设和运维的经验，介绍了如何利用算法运维、开发运维一体化、运维大数据分析等新一代智能化相关技术构建支撑未来企业信息化建设的应用智能运维系统。本书分别从技术发展演进路线、关键技术、系统建设实践、关键场景和行业应用案例等方面详细阐述了应用智能运维系统的建设思路、方法与策略。

本书第 1 章和第 2 章分别从应用运维和智能运维角度出发，梳理了运维技术发展的来龙去脉，简述了具有里程碑意义的运维工具和方法；第 3 章围绕信息技术发展趋势，详细分析了未来应用运维场景的新需求，以及建设智能化算法辅助运维系统的必要性；第 4 章和第 5 章相对全面地介绍了建设应用智能运维系统需要用到的关键技术和工具；第 6 章从企业实际需求出发，详述了系统规划建设需要做的前期准备、设计规划和概念验证的详细过程；第 7 章围绕一个具体案例，展开介绍了如何从零开始搭建完整的应用智能运维系统；第 8 章和第 9 章分别从典型场景和行业角度出发，分析了物联网、车联网、开发运维一体化等特定场景下运维需求的特点，总结了能源电力、广电传媒、数字医疗等行业面向具有超高复杂度的新一代应用系统的智能运维平台建设的要点和价值。

本书介绍的实战经验全部来自一线项目团队和产品研发团队的积累，每个项目建设过程都历经艰辛，这些经验来之不易，汇总梳理这些经验也耗费了大量的心血。在此，特别感谢东软集团 RealSight APM 应用智能运维产品研发团队的崔喜龙、王占、石子凡、邹康、刘长东等在关键技术和产品研发方面的贡献；感谢英特尔大数据技术全球 CTO、大数据和人工智能创新院院长戴金权（Jason Dai）与该院方案架构师乐鹏飞提供的技术支持及项目经验；感谢给我们提供宝贵需求、项目实施经验和验证环境的客户（包括中国航空、宝马中国、蒙牛集团、中国移动等），是你们赋予了技术社会价值，让产品研发团队和项目团队的工作更有意义。最后，感谢电子工业出版社的米俊萍编辑对本书认真负责的审阅，她帮助我们甄别了书中大量的错误和表述问题，让我们这些不善表达的技术人员写出的东西更通俗易懂。希望本书能帮助企业、政府在建设应用智能运维系统时少走一些弯路，同时为未来中国软件企业研发世界领先的国产应用运维软件提供一些参考。

本书的出版得到了国家重点研发计划项目“智能工厂工业互联网系统理论与技术”（2018 YFB1700100）及国家自然科学基金项目（61471084）的资助，在此表示感谢。

作　者

2020 年 2 月 25 日

目录

第 1 章　应用运维……013

1.1 初识应用运维……014
1.2 应用运维，保障企业应用稳定运行的关键……015
1.3 演进过程……018
1.3.1 软件性能工程……018
1.3.2 应用性能管理……021
1.3.3 网站可靠性工程……022
1.3.4 业务流程性能监控管理……023
1.3.5 用户数字体验监控……024

第 2 章　智能运维……027

2.1 初识智能运维……028
2.2 智能运维，赋予企业运维更强悍的大脑……029
2.3 演进过程……032
2.3.1 IT 运维分析……032
2.3.2 事件关联分析……034
2.3.3 自动化运维……034
2.3.4 人工智能运维……035
2.3.5 开发运维一体化……038

第 3 章　智能、互联时代的应用运维……040

3.1 应用演进趋势……041
3.2 技术演进趋势……051
3.3 应用智能运维系统：企业数字战略的关键支撑……055
3.4 商业价值评估（ROI 分析）……057

3.5 系统关键能力…… 071

第 4 章　应用运维智能化的关键技术…… 077

4.1 异常检测：筛选时间序列数据，发现潜在风险…… 079
4.1.1 技术简介…… 079
4.1.2 深入浅出应用实践…… 082
4.1.3 应用案例…… 093

4.2 关联分析：实现全景化应用监控的基础…… 097
4.2.1 技术简介…… 097
4.2.2 深入浅出应用实践…… 097

4.3 数据统计：敏捷高效的信息提取手段…… 101
4.3.1 技术简介…… 101
4.3.2 深入浅出应用实践…… 106

4.4 预测分析：使应用性能风险防患未然…… 111
4.4.1 技术简介…… 111
4.4.2 深入浅出应用实践…… 112

4.5 因果推理：专家经验辅助决策支持…… 115
4.5.1 技术简介…… 115
4.5.2 深入浅出应用实践…… 118

4.6 自治控制：应用运维过程的自动化管理…… 124
4.6.1 技术简介…… 124
4.6.2 深入浅出应用实践…… 126

第 5 章　应用智能运维工具图谱……079

5.1 开源工具……080
5.1.1 业务流程巡检拨测……080
5.1.2 应用请求链路追踪……084
5.1.3 存储海量监控数据……089
5.1.4 机器数据检索分析……093
5.1.5 人工智能算法支撑平台……094
5.1.6 应用监控数据可视化……102
5.1.7 告警及风险智能管理……111

5.2 商业化产品……114
5.2.1 Dynatrace：软件智能平台……114
5.2.2 AppDynamics：思科的战略新方向……115
5.2.3 NewRelic：让应用运维随需即取……116
5.2.4 RealSight APM：全景化应用智能管理……118
5.2.5 Datadog：深度分析应用性能……119
5.2.6 BigPanda：AIOps 算法驱动应用自动化运维……121
5.2.7 Numenta NuPIC：类脑计算践行异常检测……122

第 6 章　立足实际需求，规划系统落地方案……124

6.1 前期准备……125
6.1.1 需求准备：理解企业现有的应用运维过程……125
6.1.2 应用准备：为目标应用的运行状态准确画像……129
6.1.3 人员准备：组建技术和管理专家团队……132
6.1.4 技术准备：储备运维智能化的关键技术……133

6.2 规划设计……138
6.2.1 围绕运维现状，规划建设愿景……138
6.2.2 多部门协作，规划服务质量目标……141
6.2.3 制订监控策略，设计 SLO 计算算法……141
6.2.4 专注过程，规划有效的风险管理机制……142

6.3 概念验证……143
6.3.1 围绕核心业务，验证用户数字体验监控方案……144
6.3.2 验证应用全栈监控数据采集技术……145
6.3.3 验证业务流程监控的可行性……146
6.3.4 验证趋势预测算法的可行性……147
6.3.5 验证根源问题分析算法的可行性……148

第 7 章　从零开始搭建应用智能运维系统……152

7.1 目标应用场景的定义……152
7.1.1 目标应用介绍……153
7.1.2 建设愿景规划……153
7.1.3 应用运维现状……154

7.2 规划设计……157
7.2.1 逻辑架构……158
7.2.2 部署架构……159

7.3 应用全栈监控数据采集……160
7.3.1 用户侧用户数字体验数据采集……163
7.3.2 应用可用性数据采集……167

7.3.3 业务流程数据采集……174
7.3.4 应用运行环境状态数据采集……188

7.4 搭建数据湖，存储运维大数据……189
7.4.1 时间序列指标数据存储……191
7.4.2 应用代码链路数据存储……193
7.4.3 链路、拓扑图等关系数据存储……194
7.4.4 数据湖存储与检索能力融合……196

7.5 实现全景视图的监控数据可视化……199
7.5.1 业务优先的应用全景可视化仪表盘……200
7.5.2 定义级联可视化人机交互界面……202
7.5.3 选择监控指标，定义告警策略……204

7.6 算法驱动，实现应用风险态势感知……207
7.6.1 时间序列监控指标的趋势预测……207
7.6.2 建立实时智能的异常检测能力……208
7.6.3 通过因果推理分析定位风险根源……214

7.7 应用风险告警的智能化管理……219
7.7.1 搭建智能化的告警管理框架……221
7.7.2 遍在数据接入，随时回溯数据、解释告警……223
7.7.3 智能合并告警，有效管理风险……224
7.7.4 应用风险根源分析的智能化……228
7.7.5 手机端主动探伤检测，防患未然……236

第 8 章　典型应用场景实践……238

8.1 开发运维一体化场景……238
8.1.1 需求背景……238
8.1.2 解决方案……239

8.2 应用运行环境的稳定性性能保障……240
8.2.1 需求背景……240
8.2.2 解决方案……241

8.3 基于微服务架构的应用性能监控……243
8.3.1 需求背景……243
8.3.2 解决方案……245

8.4 基于大数据架构的应用运维智能化……249
8.4.1 需求背景……249
8.4.2 解决方案……250

8.5 遍在接入的云应用运维智能化……252
8.5.1 需求背景……252
8.5.2 解决方案……254

8.6 互联网应用的用户数字体验保障……255
8.6.1 需求背景……255
8.6.2 解决方案……256

8.7 物联网应用运维场景……260

8.7.1 需求背景……260
8.7.2 解决方案……261

8.8 车联网应用运维智能化……267
8.8.1 需求背景……267
8.8.2 解决方案……271
8.8.3 应用案例……274

8.9 应用运行环境的异常检测……275
8.9.1 需求背景……275
8.9.2 解决方案……276

8.10 应用网络质量的预测与分析……277
8.10.1 需求背景……277
8.10.2 解决方案……278

第 9 章 行业案例实践……280

9.1 网联汽车……280
9.1.1 建设背景……280
9.1.2 解决方案……280
9.1.3 建设效果……282

9.2 能源电力……283
9.2.1 建设背景……283
9.2.2 解决方案……284
9.2.3 建设效果……284

9.3 广电传媒……285
9.3.1 建设背景……285
9.3.2 解决方案……285
9.3.3 建设效果……286

9.4 数字医疗……287
9.4.1 建设背景……287
9.4.2 解决方案……288
9.4.3 建设效果……289

9.5 电子政务……290
9.5.1 建设背景……290
9.5.2 解决方案……291
9.5.3 建设效果……292

9.6 银行保险……293
9.6.1 建设背景……293
9.6.2 解决方案……294
9.6.3 建设效果……294

9.7 食品快消……295
9.7.1 建设背景……295
9.7.2 解决方案……296
9.7.3 建设效果……296

第 1 章　应用运维

本章内容简介：第三次信息技术浪潮推动应用运维技术和产品快速演进，使其在企业经营管理体系中的重要性快速提升。本章概要介绍应用运维的发展历史、核心价值和演进过程。本章从背景起源着手解释应用运维的概念，然后介绍应用运维在企业用户数字体验保障和企业运营等方面的价值，最后梳理应用运维技术伴随软件技术发展历程的演进脉络和其中具有里程碑意义的技术。

1.1 初识应用运维

应用运维保障是软件全生命周期管理过程中的关键环节。软件系统开发上线后，要达到预期的设计目标、稳定服务于目标场景，全靠运维保障支撑。应用运维系统和过程建设是企业建设完整的 IT 运维管理（IT Operations Management，ITOM）体系的核心。与网络运维、云环境运维、IT 基础设施运维相比，应用运维更贴近用户和业务目标。

从解决的目标问题域来看，通常理解的企业应用运维包含能够对应用系统运行期状态进行监控、风险发现和管理、根源问题分析的工具集及与之对应的运维过程。应用监控和运维支撑工具需要对应用服务的目标用户使用情况、应用业务流程执行过程、应用代码执行情况及应用运行依赖的运行环境进行监控、风险监测和告警通知；应用运维过程要能够与企业现有的 IT 运维体系对接，遵循 IT 服务管理的最佳实践 ITIL（Information Technology Infrastructure Library）[1]，能够与工单管理系统、配置管理数据库（Configuration Management Database，CMDB）[2]对接，组成从应用风险发现、上报、定位到应用恢复的完整闭环运维管理流程。

随着物联网、大数据、虚拟化、云计算等新一代信息技术的快速发展与应用，以及企业运营对应用系统的依赖增加，应用运维在企业内部的重要性在快速提升，但新技术也使应用复杂度快速增加，企业应用运维面临更严峻的挑战。

[1] https://en.wikipedia.org/wiki/ITIL.

[2] https://en.wikipedia.org/wiki/Configuration_management_database.

1.2 应用运维，保障企业应用稳定运行的关键

企业数据中心、云平台、网络存在的价值和意义体现为支撑应用系统为企业的内部、外部目标用户提供持续、稳定的数字服务。如果用户使用的应用系统连接缓慢、不稳定，那么即使数据中心计算能力强悍、云平台管理完善、网络架构优雅也无济于事；如果应用运行持续稳定，那么即使基础设施出现故障也不是大问题。持续提升应用运行期的稳定性和性能以保障用户数字体验流畅，是所有监控、运维管理工作的唯一关键目标。

在数字时代，一切都依赖于应用系统稳定可靠的运行。然而，智能、互联时代的数字信息系统日趋复杂化，应用之间的交互关系密如织网，随着企业经营对信息系统的依赖程度加剧，负载也急剧增加。互联网、物联网、车联网、体域网等网络结构的多样化也使应用系统越来越复杂。这些趋势给应用系统的稳定、可靠保障带来了挑战。系统故障和宕机频率快速升高，人工运维成本飙升。

著名管理咨询公司麦肯锡在名为 Measuring the Net’s Growth Dividend 的分析报告中指出，2013—2025 年，互联网将帮助中国的 GDP 增长率提升 0.3～1.0 个百分点，经济发展的需要势必推动企业对新型系统架构的需求快速增长。如今，几乎所有企业都面临如何利用新一代信息技术来对外提升企业用户价值、对内优化生产流程的问题。应用系统无疑是这些问题的解决方案的核心。

1. 稳定性决定企业数字战略的成败

如图 1-1 所示，专业评测网站 downdetector.com 统计，2018 年，Facebook 系统全年宕机 200 次，YouTube 宕机 140 次，Google 宕机 100 次。每次宕机损失至少 100 万美元。应用频繁宕机，用户数字体验糟糕，使得企业损失严重。

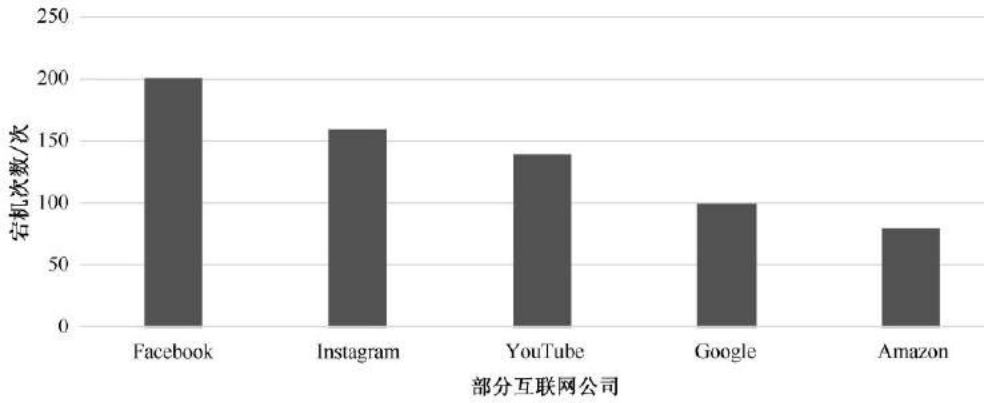


图 1-1 downdetector.com 统计的 2018 年部分互联网公司全年宕机情况

2. 应用性能决定企业的营收

对于今天更加依赖数字系统来实现、提升自身价值的企业来说，应用可用性、用户体验和响应时间等性能指标从未如此重要过。雅虎首席执行官玛丽莎·梅耶尔（Marissa Mayer）曾经做过一个实验：她把页面上的搜索结果从 10 个增加到 30 个，希望让用户一次性浏览更多的信息。但是，她发现，这样搜索结果的返回时间从 0.4s 增加到了 0.9s，广告收入下降了 20%。梅耶尔将提升在线业务的用户体验总结为：速度为王（Speed Wins）。

微软、亚马逊也做过类似的实验。2009 年，微软在必应搜索引擎上开展实验，发现当服务响应时间增加到 2s 时，每个用户带给企业的收益会下降 4.3%。由于该实验对公司产生了负面影响，最终不得不终止。亚马逊也发现其主页加载时间每增加 100ms，就会导致销售额下降 1%。对于年营收达数百亿美元的亚马逊而言，1%已是很大的损失。

在智能、互联场景下，在应用云端系统复杂度激增的同时，终端设备的代码量和系统复杂度同步快速增加。如图 1-2 所示，2014 年，大数据平台核心系统 Hadoop 的代码量为 140 万行；2015 年，Chrome 浏览器的代码量为 600 万行；2016 年，波音公司新型 787 客机的代码量激增到 1400 万行；2018 年，福特 F150 汽车的代码量达到 1.4 亿行。一般应用代码量和运维复杂度成正比，而且应用海量代码云、端协同的工作方式给运维带来了新的挑战。

无法抵消信息系统趋于复杂化带来的运维风险，企业数字化营销、数字化生产、数字化管理等战略就是空谈。建设具备全景监控、智能运维能力的应用性能管理系统，保障用户数字体验，提升应用可用性，已成为企业的必然选择。

图 1-2　软件系统代码量增长情况

1.3 演进过程

运维、运营在英语中对应同一个单词，即 Operation。一般地，运营指对企业经营过程的计划、组织、实施和控制；运维则指对生产依赖工具、设备的运行可用性保障、风险监控、故障排查、性能维护等。第一个提出运维管理概念的是现代经济学之父、英国哲学家亚当·斯密（Adam Smith）。1776 年，他在《国富论》中提到：“如果将产品生产工作划分为多个任务去组织，而不是让每个工人从头到尾完成所有任务，将更加高效。”之后，这个理念被亨利·福特用在汽车生产线上，获得了成功。信息技术发展使得企业对 IT 硬件设备、软件的依赖加剧，IT 运维的重要性提升，逐渐演变为独立的技术和管理体系，成为企业经营管理体系中不可缺少的组成部分。

软件发展过程中存在软件性能、稳定性优化等运行期维护问题。在初期，由于缺少运行期监控代码执行过程和性能方法的工具，最初的软件性能、稳定性优化工作主要在软件开发、测试阶段完成。在软件工程科学中，系统化的软件性能和稳定性优化、保障设计、开发、测试方法定义为软件性能工程（Software Performance Engineering，SPE）。

1.3.1 软件性能工程

麻省理工学院计算机科学与工程学院教授查尔斯·森特（Charles E. Leiserson）将软件性能工程定义为一门“让软件代码运行得更快的艺术”，而软件性能工程的概念在软件工程出现 13 年后，才由 L&S 计算机技术公司（L&S Computer Technology，Inc.）服务部的康妮·史密斯（Connie U. Smith）博士在 1981 年发表的论文《通过软件性能

工程提高信息系统的生产力》（Increasing Information Systems Productivity by Software Performance Engineering）中首次提出。软件性能工程要求在软件需求调研和设计规划阶段就充分引入工程化思想，考虑运维期可能产生的性能问题和稳定性风险，围绕业务实际需要定义服务质量目标，并量化分析应用上线后计划生命周期内的负载、持续稳定运行时间、运行环境变更等情况，以便指导软件架构设计和详细设计。但是，现实情况是，大多数项目经理和产品经理为了赶工期，忽视软件性能工程前期的需求调研和设计规划，在测试阶段才开始量化评估性能、稳定性指标，并相信在生产环境下通过硬件扩容和增加集群节点的方案可以解决性能问题，导致应用上线后运维成本不可控，应用系统复杂度和负载增加，从而使得情况快速恶化。

软件性能工程是在系统开发生命周期中保障非功能需求中的性能指标（如通量、时延、CPU 消耗等）达标的相关技术，在系统工程（System Engineering）中特指系统性能工程（System Performance Engineering）；在软件工程中对应软件性能工程（Software Performance Engineering）或应用性能工程（Application Performance Engineering）。随着企业业务成败与数字化应用系统建设成败的相关性逐渐增加，特别是在信息化成熟度较高的行业（如金融、ICT 运营商、汽车），应用性能工程在软件全生命周期管理过程中更加重要。

应用性能工程特指针对软件系统非功能需求进行设计、建模、定义、测试、维护，从而保障应用系统交付上线后定义的运行期日常运维服务质量目标（Service Level Agreement，SLA）达标。

应用性能工程涵盖范围涉及软件开发和运维支撑体系，与传统以 IT 服务管理（IT Service Management）、遵循 ITIL 方法学的 IT 运维相关。但是，由于应用与基础设施

的映射关系逐渐松耦合，且应用运维对于企业的重要程度提升，应用运维（或称为应用／服务性能保障）团队成为一个独立的部门成为趋势。其主要职责包含但不限于以下几方面。

（1）通过确保系统可以在必要的时间范围内处理交易来增加业务收入。
（2）消除由于应用性能目标不达标而需要优化甚至重构开发代码的系统故障。
（3）消除因性能问题导致的时延系统部署。
（4）消除因性能问题导致的可避免的系统重新工作。
（5）消除可避免的系统调优工作。
（6）避免额外的、不必要的硬件购置成本。
（7）降低因生产性能问题而增加的软件维护成本。
（8）降低因受临时性能修复影响而增加的软件维护成本。
（9）减少因性能问题而处理系统问题的额外操作开销。
（10）通过模拟原型识别未来的瓶颈。
（11）提高应用系统的最大负载能力。

完备的软件性能工程可以大幅度降低运行期的软件维护成本，但无法解决软件缺陷、需求变更或突发事件导致的生产上线后的所有运维问题。在 IT 运维发展的初期，监控管理的对象主要为服务器、网络设备等支撑应用运行环境的基础设施硬件。“不能监控，就无法管理”，由于缺少技术和工具，这个阶段对应用自身运行状态的监控管理发展相对滞后。直到硅谷的软件工程师 Lew Cirne 开发出第一款应用性能管理（Application Performance Management，APM）软件，人们才实现了对用 Java 语言编写的应用程序的代码执行过程等运行状态的监控。

1.3.2 应用性能管理

1998 年，Lew Cirne 在美国加州创建了第一家主营业务为企业级应用性能管理软件研发的软件公司——Wily Technology[1]，面向企业的、用 Java 语言编写的应用软件提供性能监控分析服务和工具。由此应用运维才真正进入了以工具支撑的企业应用运维时代。Wily Technology 在 2001 年时只有 50 名员工，之后营业额持续增长，2005 年的年营业额达到 5300 万美元，员工近 500 人，用户覆盖医疗、媒体、电信、零售、政府、金融等领域。2006 年，在 Wily Technology 被 CA Technology 以 3.75 亿美元收购后，Wily APM 被重新命名为 CA APM[2]。

APM 是建设企业应用性能管理平台，打通开发、运维，实现软件全生命周期管理的核心。诞生之初，APM 就已经显现了它在应用运行期发现、排查故障方面的价值和潜力。大数据、互联网、移动化、云计算等新兴信息技术的快速发展使应用系统本身的架构越来越复杂，系统间的交互关联增加。企业直接通过软件服务，对通过互联网面向用户交付服务的需求快速增加。Lew Cirne 看到商机，创建了 New Relic 公司。LewCirne 抛弃了 Wily Technology 以提供企业内部应用性能保障为主的经营模式，转而面向公有云、混合云环境下的互联网应用，以软件即服务（Software as a Service，SaaS）的方式提供网站、应用 Web 门户和移动应用终端的用户数字体验监控及后台支撑系统管理服务。有意思的是，公司名称“NewRelic”就是 Lew Cirne 本人名字字母的重组。

另一家具有相当影响力的 APM 企业是 2008 年创建于硅谷的 AppDynamics。其创始人 Jyoti Bansal 是 Wily Technology 的首席架构师。该公司共获得了 5 轮总计

[1] https://en.wikipedia.org/wiki/Wily_Technology.

[2] https://en.wikipedia.org/wiki/New_Relic.

2.06 亿美元的投资，2017 年被 Cisco 以 37 亿美元收购，这被认为是 Cisco 坚定发展其软件业务的策略之一。目前，AppDynamics 产品归并在 Cisco 物联网和应用业务线下[1]。

目前，市场占有率较高的 APM 企业是 Dynatrace，这家公司于 2005 年 2 月在奥地利林茨创建，2011 年 7 月被 Compuware 公司收购，更名为 Compuware APM。直到 2014 年，Thoma Bravo 将该产品私有化，并从 Compuware 剥离，将其重新命名为 Dynatrace[2]。近年来，Dynatrace 正在逐渐拓展 APM 产品的边界，提出超越 APM（Beyond APM）、为企业搭建软件智能平台（Software Intelligent Platform）的理念，重点进行基于人工智能算法的智能化运维研发，利用算法帮助运维人员发现、定位风险。

1.3.3 网站可靠性工程

网站可靠性工程（Site Reliability Engineering，SRE）这个名词来自谷歌员工 Ben Treynor Sloss 设立的岗位名称，他从 2003 年开始负责谷歌全球运维，到 2016 年，其团队规模超过 4000 人。Ben 给 SRE 岗位的定义是“软件工程师处理以往称为运维的事情”。由于起源于谷歌，SRE 过程和岗位规划比较适合注重用户数字体验保障、系统复杂度高的互联网企业。目前，几乎所有在线用户数上规模的互联网公司都已经规划了 SRE 岗位。传统行业的企业应用运维也正朝着互联网化演进，尤其是在已经具备直接面向用户提供数字服务能力的企业中。例如，汽车行业中有面向车主提供网联车云服务的车厂；金融行业中有建设了网上银行系统、手机银行系统的银行等。

[1] https://en.wikipedia.org/wiki/App Dynamics.

[2] https://en.wikipedia.org/wiki/Dynatrace.

SRE 工程师一半的工作内容是做运维的工作，如处理工单、告警、排查风险，但由于软件系统的复杂度高，SRE 工程师的运维目标是实现高度自动化和可自愈的运维体系；另一半的工作内容是规划设计、研发新特性，实现应用自动扩容或收缩适应负载变化、自动配置变更管理、故障自动发现与定位等功能。合格的 SRE 工程师不但要具备软件工程师的经验和技术基础，而且要具备系统工程师的运维管理经验，有很强的编码和自动控制能力。

区别于传统应用运维，SRE 运维的基本思路是通过软件实现所有的日常运维工作。因此，基于软件工程和性能工程解决问题是基本原则。SRE 运维的目标并不是要让所有系统和服务达到 100%可用的程度，这不太可能。同时，过高的可用性会带来运维工作量和成本的快速提升，而收益未必提升太多，会导致投入产出比降低。SRE 工程师需要基于场景与业务部门协商一系列切实可行的 KPI，其中对应的量化衡量目标称为服务质量保障目标（Service Level Objective，SLO）。为了达成既定的 SLO，SRE 工程师不但需要定义量化监控指标、告警策略和风险应对方案，而且需要与业务部门密切协作。尤其是当出现故障告警、SLO 对应指标不达标，并且找不到原因时，大家要坐在一起冷静、客观地分析和定位问题根源，商讨应对策略，不能相互抱怨。当然要做到这点并不容易。

1.3.4 业务流程性能监控管理

与 SRE 类似，应用业务流程性能监控需求起源于互联网公司对用户数字体验监控的场景。由于在互联网运维场景下，业务流程变化速度快，商品促销等数字营销推广的效果与应用平台指定业务流程的性能、稳定性直接相关。因此，监控指定业务流程的执行过程、点击量分布、用户访问数字轨迹就非常必要。最早提供业务流程性能监控功能的

是具备网络性能诊断分析（Network Performance Monitoring and Diagnostic）[1]产品的软件产品厂商，如 Riverbed、NetScout。这些厂商的主营业务是提供网络旁路方式，以侦听网络流量、监控网络状态和分析安全稳定性问题，并通过拆包拿到某些应用的性能、业务执行情况等信息，从而监控特定业务的笔数，如手机银行和网上银行的交易量、接口调用次数。这些数据可以帮助运维部门、运营部门监控和管理业务流程的执行状态。但是，这些厂商对 VPN 或 https 加密数据链路的拆包分析能力有限。

APM 产品通过提供开发期植入的 SDK 埋点，或者运行期链路追踪探针追踪指定请求链路的方式，可以更精确地、完整地识别监控业务流程，在系统节点故障告警时，能够快速找到影响用户的业务流程执行链路。这对关联运营 KPI 和运维服务质量目标、实现目标导向的精益运维非常有帮助。目前，能够提供这方面能力的 APM 产品包括东软 RealSight APM 应用智能运维平台[2]、Microfocus 业务流程监控[3]、Germain APM 业务流程监控与分析[4]。

1.3.5 用户数字体验监控

用户数字体验监控（Digital Experience Monitoring，DEM）是应用运维逐渐与基础设施、云平台等运行环境解耦合，紧密围绕用户提供应用服务质量保障能力的新阶段。DEM 概念最早出现在信息技术咨询服务公司 Gartner 关于 APM 产品魔力象限的报告中。根据 Gartner 的定义，DEM“是为了优化与应用服务交互的数字代理（人或机器）的操

[1] https://www.gartner.com/en/documents/ 3969863.

[2] http://www.rsapm.net.

[3] https://www.microfocus.com/ en-us/products/business-process-monitoring/overview.

[4] https://germainapm. com/features/Business-Process-Analytics/.

作体验和行为而制定的可用性及性能监控原则”[1]，“应用逐渐采用云计算与移动化技术的趋势，推动了企业 IT 部门转变应用性能监控的方式”。Gartner 给出了企业运维团队需要重点关注的驱动需求，具体如下。

（1）缺少 SaaS 平台监控运维经验，使得用户经常遭遇服务质量问题，对企业经营造成了影响。

（2）意识到用户数字体验不只是企业用户最关注的，其对考核企业运营效率、员工工作有效性和回应股东利益关切同样重要。DEM 技术能够提供独一无二的方式来提高员工工作效率和提升用户数字体验。

Gartner 在市场策略分析报告 Gartner’s Strategic Planning Assumption for Its Market Guide States 中指出，“截至 2023 年，60%的数字业务提案中都将要求运维部门汇报用户数字体验，相比现在 15%的比例有大幅度的提升。”

DEM 强调从用户角度量化监控操作性能、数字轨迹、统计使用习惯与性能指标的变化，通常以用户能够理解的服务质量目标自顶向下关联应用与基础设施指标，构建树形结构逐层细化的监控体系。对企业来说，其主要收益包括：提升从用户端量化监控、评估应用可用性及性能的能力，使得优化用户体验更有针对性；提升对应用 SaaS 及云服务的性能可见性；聚焦用户终端设备接入性能，更精确地理解和评估数字体验；结合用户情感变化数据和主观体验指标，提前处理风险，提高企业员工的工作效率并降低其工作负荷；对由于技术问题影响业务和企业经营的事故分析定位更准确；提升跨域全面监控能力，提供端到端的全景化应用监控视图。

[1] https://www.gartner.com/en/documents/3956998.

Gartner 将 DEM 定义为评估 APM 产品的三类功能象限中的关键评估项，其包括数字体验建模（Digital Experience Modeling），应用探查、链路追踪与诊断（Application Discovery，Tracing，Diagnostics，ADTD）和应用分析（Application Analytics，AA）[1]。随着数字空间和物理空间的加速融合，用户数字体验监控与保障对企业将越来越重要。

具备数字体验监控能力、提供相关工具产品的主要是传统 APM 厂商和新一代应用智能运维厂商，包括 Dynatrace、AppDynamics、NewRelic、Lakeside、RealSight APM 和听云等。

本章小结

应用运维是保障应用软件系统上线后发挥设计价值的过程。从企业实际需求出发，了解应用运维的详细过程和相关技术发展脉络是实践应用运维智能化的基础。本章重点对应用运维自诞生至今发展过程中出现的软件性能工程、应用性能管理、网站可靠性工程、业务流程性能监控管理和用户数字体验监控几个具有里程碑意义的技术及对应的工具、厂商进行了总结归纳。

[1] Smith C U.Increasing Information Systems Productivity by Software Performance Engineering[C]. Proc. CMG XII International Conference. December 1981.

第 2 章　智能运维

本章内容简介：智能化是未来企业 IT 运维的主要趋势。本章综述了通过算法替代人工发现、定位、处理风险，为 IT 运维提供决策支持的相关技术和产品的演进脉络，介绍了 IT 运维分析、事件关联分析、自动化运维、人工智能运维和开发运维一体化几个具有普遍认知的相关理念的发展历史及实际应用价值，总结了智能运维技术和产品的发展对企业应用运维管理的推动作用。

2.1 初识智能运维

近几年，人工智能技术发展很快，通常理解的智能运维是把人工智能技术应用在 IT 运维领域，替代人工进行风险管理决策。从通过机器实现自动化流程、替代人工并解放运维人员的根本需求出发，能替代人脑进行运维决策、人手管理配置的算法和工具都可以称为智能运维系统。2000 年，早期机器学习算法出现，用来代替人脑识别指标的变化模式，预测未来的趋势。IT 运维管理通过程序实现软件、硬件的自动管理，这已经是智能运维的初级阶段。未来，智能运维技术借助概率计算、深度神经网络、因果推理分析等高级人工智能算法，将进一步提升系统自主分析决策能力，实现自治程度更高的智能运维。

2.2 智能运维，赋予企业运维更强悍的大脑

数字化新术、新需求的涌现促使企业拥有的应用规模和应用复杂度快速膨胀，使得企业应用运维不堪重负。由于应用性能问题导致企业用户流失和经济损失的案例逐渐增加。传统 IT 运维的被动响应式风险处理机制已难以应对这些问题。实现主动预防的风险处理机制已逐渐成为构建面向未来的智能运维平台的关键。

为应对未来将面临的智能、互联时代的运维挑战，通过机器智能手段处理机器数据、解决机器系统的复杂度膨胀问题，是目前唯一可行的解决方案。搭建智能运维平台，构建高效、智能的应用性能风险主动防御体系，可以让企业变被动为主动，防患于未然。

《纽约时报》一篇文章曾报道，微软研究人员 Harry Shum 发现：当网站的响应时间比竞争对手慢 250ms 以上时，用户更倾向于关闭网站。这说明应用软件的用户体验下降或宕机将直接导致用户流失，当前企业经营运转比以往更依赖应用软件。除此之外，近年来新技术、新需求的涌现促使企业拥有的应用规模和复杂度快速膨胀，企业原有的 IT 运维逐渐无力招架，应用性能异常导致的用户流失和经济损失的问题更加突出。

目前，尽管已有很多企业认识到应用性能问题的严重性，并已加大投入来构建、完善应用性能管理平台，然而，传统应用性能管理主要以实时监控、被动告警方式通知运维人员处理风险。这种方式虽然能降低损失，但无论运维人员反应多么迅速，其仍需要耗费少则几小时，多则几天时间来排查解决故障，因此这种方式无法避免对企业运营造成的影响。阿里云、WhatsApp、Adobe Creative Cloud、Facebook 等频繁发生的事故

时刻提醒我们问题的严重性。因此，被动处理方式的 APM 已不能满足企业快速数字化转型的需要，主动分析定位潜在问题、预防应用性能风险已成为未来 APM 的趋势。如何做到主动防御，提前发现并规避风险呢?

红木神经科学研究院创始人、美国工程院院士杰夫·霍金斯认为：智能的本质是“预测”。只有能够预测未来趋势和可能发生的事件，才能争取提前规避问题的时间，这是变被动为主动的关键。因此，APM 只有具备了对未来应用性能变化趋势及风险的“预测能力”，才能主动发现并规避风险，将企业运维人员从繁冗的应用性能管理工作中真正解脱出来。

分析海量历史运维数据是在应用健康状态良好的情况下提前发现风险的主要途径。从数据中找到应用存在的潜在问题与风险，可主动预防应用性能风险。现阶段，APM 预测分析能力对用户的价值主要体现在以下几个方面：①预测未来应用性能的变化趋势；②实现更精准的容量规划；③预测、分析应用性能瓶颈；④预测、分析潜在的稳定性风险。

当前市场上具备运维数据分析能力的 APM 产品主要是面向企业应用的传统 APM 产品（如 CA APM）和面向互联网应用的新型 APM 产品（如 NewRelic、Dynatrace、Netuitive 等）。在新发布的产品中，CA APM 重点强调主动性能管理能力，通过预测应用未来的负载变化趋势，指导用户优化应用资源配置；NewRelic、Dynatrace 强调分析的实时性，提供围绕在线用户、应用事务、用户体验相关的数据统计分析功能，以易于理解的方式将当前围绕应用健康状态的分析结果展示给用户；Netuitive 则重点打造面向未来的预测分析能力，利用机器学习回归算法，通过分析历史监控指标数据来给出未来一段时间的指标曲线波动情况。除此以外，Netuitive 还能够通

过独特的行为学习技术，学习指定时间范围内的监控指标波动状态，发现指标之间的关联关系，预测未来可能发生的异常，并提前生成主动告警。

随着信息技术的快速发展，企业运营对数字信息系统的依赖加大，IT 运维的重要性和成本快速增加。同时，新一代信息技术和创新业务流程也在推动系统复杂化，人工运维已经难堪重负，智能运维被寄予厚望。近几年来，无论是学术界还是产业界，对智能运维领域技术和应用的关注度都在快速提升。ExtraHop 在 2016 年面向大中型企业的调查报告中指出，60%的企业有计划整合竖井式的分布异构运维数据源，实现统一运维数据存储分析平台[1]。Gartner 预测，到 2022 年，40%的企业将会部署智能运维平台，实现运维智能化。

[1] ExtraHop Inc.The State of the ITOA Today-How Organizations Are Building IT Operations Analytics (ITOA) Practices[C]. ExtraHop. 2016-6-21.

2.3 演进过程

在 IT 运维初级阶段，企业就有动力通过以算法和自动化流程驱动的“智能运维”来代替人工。当时，信息系统主要以企业内部自用的企业资源管理、计算机服务设计等系统为主，系统服务范围小，运维成本和压力相对较小。企业没有足够的动力来做 IT 运维智能化的事情。智能运维发展加速的一个重要的催化剂是，如 Google 这样的互联网公司迫于运维压力，开始尝试利用统计学方法分析运维数据中的模式，预测未来趋势。从 2010 年开始，云计算和大数据技术的快速发展也推动了企业利用大数据与算法提升 IT 运维能力的需求，智能运维发展真正进入了快车道。时至今日，在智能运维的演进过程中，主要的里程碑有 IT 运维分析、事件关联分析、自动化运维、人工智能运维、开发运维一体化。

2.3.1 IT 运维分析

IT 运维分析（IT Operations Analytics，ITOA）指实现基于海量 IT 运营数据的演绎、归纳推理，并支撑 IT 运营数据采集、存储、展现的相关技术及服务。其利用数学算法或创新方法，从海量 IT 监控管理系统采集的原始数据中挖掘有用的信息。ITOA 是通过分析海量、低价值密度的 IT 系统的可用性和性能数据，发现复杂的数据模式，从而辅助优化企业 IT 运营过程的系统，其需要具备的核心能力如下。

（1）风险根源定位分析：通过融合分析来自基础设施、应用、用户的监控数据，定位产生风险或对系统健康造成潜在威胁的根源所在。

（2）性能可用性预测分析：基于历史数据预测未来系统性能和可用性的变化趋势，以及关联分析对系统可能产生的影响。

（3）问题识别与派发：围绕当前问题，从历史记录中查找解决方案和适合解决问题的团队或人，提高处理问题的效率。

（4）影响范围推理分析：当发现多个风险可能对系统造成影响时，基于从数据中发现的模式推理找出可能影响更大、优先级更高的风险，指导相关人员及时、高效处理这些问题，降低损失。

（5）多源数据融合互补：对 IT 基础设施和应用采集的数据进行关联、融合，补全网络、应用、服务拓扑结构，完善探查管理类工具信息视图。

（6）动态风险告警阈值管理：自动发现监控指标的正常运行范围，在用户负载变化或系统配置变更后，能够自动从历史数据中发现规律，调整异常告警区间的限定阈值范围。

对于 ITOA 技术，Gartner 在 Data Growth Demands a Single, Architected IT Operations Analytics Platform 报告[1]中总结了六种：①日志分析技术；②非结构化文本数据索引、查询和推理技术；③拓扑分析技术；④多维数据库查询分析技术；⑤复杂运维事件处理技术；⑥数据统计分析、模式发现与识别技术。具备这些技术的 ITOA 才能满足基础设施和应用层的监控需求，实现由多源异构探针采集的时间序列指标、日志、代码链路、网络包和用户数字轨迹数据的聚合、关联和分析。目前，市场上的 ITOA 产

[1] https://www.gartner.com/en/documents/2599016.

品提供商主要有 Splunk、Elastic、Dynatrace 和 RealSight APM 等。

2.3.2 事件关联分析

在主动风险预测和预防性维护技术未成熟之前，企业运维风险管理工作主要以工单、风险告警等事件驱动工作方式为主。在运维过程中，事件关联分析（Event Correlation and Analysis，ECA）[1]则主要用来关联多种监控系统事件，协同不同团队角色人员的工作。具体地说，ECA 能够帮助 IT 运维人员消除重复上报工单事件或告警；根据不同人员角色和业务运维需要来过滤、查询相关事件；根据历史数据或预定义规则关联事件，找出告警事件的根源问题或查找事件间的相关性和影响关系。这种处理方式在一定程度上能减少人工过滤无效事件的工作量，并辅助查找对应事件最合适的处理角色，这也是通过算法实现指定类型风险处理的智能运维的一种简单、有效的方案。市场上主要的 ECA 产品提供商有 Argent Software、Augur Systems、BMC Software 和 CA。

2.3.3 自动化运维

如果说智能运维技术发展的主线是为了解放运维人员，ITOA、ECA 通过数据驱动辅助决策来解放 IT 运维人员的大脑，那么，自动化运维（Automated System Operations，ASO）[2]技术则主要是为了解放运维人员的手和脚。在日常运维中，当面临大量服务器、应用，需要有限的运维人员维护管理时，自动化运维工具和产品能够帮助运维人员设置自动化脚本，批量安装操作系统，部署中间件和应用，配置变更管理。Gartner 将 ASO 定义为“不需要人工干预，直接操控物理设备就能控制计算机安装配置

[1] https://www.gartner.com/en/documents/1492516/magic-quadrant-for-it-event-correlation-and-analysis.

[2] https://www.gartner.com/en/information-technology/glossary/aso-automated-system-operations.

硬件和软件的过程”。

借助ASO工具，IT运维人员可以在控制台通过定义自动化脚本准备应用的运行环境，安装部署应用，准备集群节点，控制弹性分组。结合脚本语言编程，运维人员可以将更复杂的控制流程自动化。结合ITOA和ECA的风险告警，以及根源定位分析事件触发，可以实现特定场景下对特定风险的自愈控制。比较常用的ASO工具包括Chef、Puppet、Ansible和Saltstack。

2.3.4 人工智能运维

第一个提出AIOps概念的是著名的IT咨询公司Gartner[1]，其给出的定义是算法运维（Algorithmic IT Operations），其中的AI并不是现在大家理解的人工智能。2017年4月，在印度孟买的新闻会上，Gartner将AIOps解释为“AIOps平台由可以完成数据采集、存储、分析和可视化的多层架构系统组成，具备与第三方应用通过不与厂商绑定的API接口对接数据的能力，能够和IT运维管理（ITOM）类工具进行数据交互和能力对接”。Gartner完全站在IT运维数据分析的角度给出了AIOps的基本能力边界，和人工智能没有一点儿关系。然而，由于人工智能技术是大热点，业界更愿意将AI理解为更时髦的人工智能算法，AIOps也就只能顺应潮流，被定义为人工智能运维。从目前机器学习、人工智能技术的应用现状和发展趋势来看，IT运维领域的目标数据以机器数据为主，机器行为相比于人的行为规律性较强，状态数据采集简单，质量相对可控。使用算法运维替代人工运维更容易落地，真正的人工智能运维已经不再遥不可及。

从需求和技术发展的趋势看，企业内多源数据融合和集中式运维与运营数据支撑是

[1] https://www.gartner.com/en/information-technology/glossary/aiops-artificial-intelligence-operations.

大势所趋，但由于采集方式和数据类型多样、数据存储分散、智能分析场景众多，实现难度较大，需要从核心场景出发，按需规划，分阶段递进实现。Gartner 给出的 AIOps 平台的核心能力包括以下几项。

（1）能够从多种数据源采集数据，不与厂商绑定。

（2）支持对接、处理实时数据和批量历史数据。

（3）提供对融合数据的检索、统计。

（4）提供海量实时、历史数据的存储。

（5）支持使用机器学习算法来分析、处理数据。

（6）能够基于分析结果规划下一步的处理动作。

总结企业应用的运维场景，可知常见的人工智能运维场景如下。

（1）基本和高级统计分析：单变量和多变量分析的组合，包括对跨 IT 实体捕获的指标使用相关性、聚类、分类和外推分析，以及从监控数据源中对数据进行整理。

（2）自动模式发现和预测：使用上述一种或多种类型的历史或流数据，得出数学或结构模式，描述可以从数据集本身推断但不会立即存在于数据集本身的新相关性；然后，这些模式可用于及时预测具有不同概率的事件。

（3）应用异常检测：使用前一个组件发现的模式，首先确定构成正常系统的行为，然后识别偏离该正常系统的行为。

（4）根本原因确定：向下修剪由自动模式发现和预测组件建立的相关网络，以隔离那些代表真正因果关系的依赖关系链接，从而提供有效干预的方法。

（5）规定性建议：对问题进行整理，将它们分类为已知类别；然后，挖掘以前解决方案的记录，分析这些解决方案是否适用，并优先提供这些解决方案，以便尽早使用补救措施；最终，使用闭环方法，并在使用后对其有效性进行表决。

（6）拓扑：对于 AIOps 检测到的具有相关性和可操作性的模式，必须围绕引入的数据放置上下文，该上下文就是拓扑；如果没有拓扑的上下文和事实上的约束，检测到的模式虽然有效，但可能毫无帮助且会分散注意力；拓扑中的数据派生模式将减少模式的数量，建立相关性并说明隐藏的依赖关系；使用拓扑作为因果关系确定的一部分可以大大提高其准确性和有效性；使用图形和瓶颈分析捕获事件发生的位置及其上下游的依赖关系，可以提供关于将补救工作重点集中到何处的见解。

一些企业，尤其是拥有庞大数据中心和复杂应用的互联网公司，已经将此技术应用于特定场景，比如用户异常行为检测、云端应用弹性控制、容量规划、入侵检测、数据中心 PUE 能效管理、硬盘损坏预测等。有些企业甚至开始尝试通过融合开发、运维、运营数据来打造一体化智能化平台，关联运营 KPI 和运维 SLO，同时为企业各部门提供全景数据视图和智能决策支持。非 IT 运维部门，如业务规划部、销售部、产品部和数字营销部都有自己的应用系统和数据，也希望借助其他途径获取更丰富的数据以了解目标用户、市场和使用场景。数据量的激增也使得大数据采集、存储和智能分析成为必备技术。因此，为了满足企业内更广泛的需求，AIOps 平台对接的数据源的种类在增加，能力边界也在扩大。例如，传统 APM 产品提供商 Dynatrace 已经在践行 AIOps 的基础上提出了软件智能（Software Intelligence）平台的概念，推出了数字业务分析（

Digital Business Analytics）服务，能够为企业数字运营部门提供实时的用户数字体验监控、转化率变化分析、企业营收与应用性能关联分析和用户画像分类等服务。

2.3.5 开发运维一体化

现在企业更加依赖数字信息系统与最终用户交互，企业应用互联网化已经是大势所趋。对于互联网应用的开发与运维，开发运维一体化（DevOps）是回避不了的一个话题。根据 Wikipedia[1]的解释，DevOps 这个说法第一次出现在 2009 年比利时 Ghent 举办的一次由敏捷实践者、项目经理和咨询顾问参与的称为 DevOpsDays 的会议上。虽然截至目前，学术界和产业界对 DevOps 的概念还未达成共识，但从企业信息化系统应用开发、运维的实际需求出发，DevOps 通常被理解为包含工具、过程和人的一系列最佳实践，融合了应用软件开发期管理（Dev）和运行期维护（Ops），旨在缩短应用全生命周期的开发过程，提升运行期应用的可靠性、可用性和性能。

业界将 DevOps 概念应用在软件系统运维过程中的实践最早可以追溯到 Google 提出 SRE 概念时。当互联网应用新功能上线周期越来越短、代码更新越来越频繁时，Google 不得不想办法在满足频繁发布代码需求的同时，保障上线代码的可靠性与性能能够支撑大规模用户同时在线访问，以及提供高质量的最终用户体验。践行 DevOps 与实现软件自动化发布或制定产品研发工作计划无关，DevOps 的初衷是通过提高软件开发与运维体系的衔接水平，将软件价值加速交付给企业的最终用户。要提供价值，企业必须在生产中运行应用程序以测试应用程序，并使用自动化流程管理工具来指导接下来交付的内容。

[1] https://zh.wikipedia.org/wiki/DevOps.

当企业践行 DevOps，建设基于 DevOps 的应用开发、运维全生命周期管理体系时，应用智能运维系统只是其中支撑应用运行期管理环节的工具。为了支撑 DevOps 落地，应用智能运维系统不仅需要支撑应用运维人员实现运行期的状态监控、风险管理、用户数字体验保障，而且需要对接开发人员，实现在开发期定义应用业务监控关键 KPI 指标、分析运行期代码质量和支撑性能工程等过程，并且在新功能上线、代码更新时，支持 A/B 测试、灰度发布、蓝绿发布等应用场景。在具备面向运维提供代码级白盒监控的能力和风险主动感知的能力的同时，DevOps 体系下的应用智能运维系统也需要无缝衔接开发，在代码有故障且运维人员无法处理时，需要快速找到责任人，向开发人员分享相关实时数据。如果应用上线后代码性能不达标，运行一段时间后，应用智能运维系统需要生成分析报告，指导后续性能优化和容量规划。

本章小结

智能运维是企业进一步提高运维效率、提升应用可用性和性能保障能力的关键。本章系统介绍了运维智能化过程中出现的 IT 运维分析、事件关联分析等一系列相关技术和产品，从背景起源、主要特点和应用场景方面概述了技术背景，相对完整地勾勒了智能运维的发展脉络，为后续介绍应用智能运维相关的技术和建设实践方法奠定了基础。

第 3 章　智能、互联时代的应用运维

本章内容简介：企业在规划建设面向未来的应用智能运维系统之前，首先要了解未来的应用系统和技术演进趋势。本章首先从历史和当前的发展路线总结应用与相关技术的发展趋势，通过对企业运维演进路线和现状的分析阐述为什么应对未来智能、互联时代信息化建设的挑战，需要应用智能运维系统的支撑，进而总结分析该系统能带来的商业价值，以及为了支撑企业建设面向未来的智能、互联数字信息系统，应用智能运维系统需要具备的关键能力。

时至今日，数字信息系统已经逐渐渗透，深刻改变了企业生产、经营、竞争、管理等活动格局和方式。2015 年，哈佛商学院教授迈克尔·波特在发表于《哈佛商业评论》上的《智能、互联产品如何变革竞争格局》[1]一文中指出：我们目前正处在以智能、互联为特征的第三次 IT 浪潮的边缘。在智能、互联时代，软件将渗透各行业，数字信息系统将成为各种产品不可分割的一部分。大量生产工具、生活用品将联网，成为数据链条的一环。在新场景下，大数据、物联网、人工智能、云计算等新型技术的普及应用会极大地提高生产效率，并提升生活品质。然而，对企业来说，新技术是一把双刃剑。复杂化的产品体系结构和无所不在的数字链路，必将导致企业拥有的应用数量快速增长，应用复杂度快速膨胀，使得企业 IT 运维不堪重负。

在信息技术过去五十多年的发展历程中，有两次信息技术快速发展的浪潮。如今，我们正处在第三次信息技术浪潮来临的前夕。新一代信息技术将再次深刻改变企业的经营方式，重塑竞争格局。在应用系统演进的同时，需要与之对应的应用运维系统，以便解决随之而来的稳定性和性能保障等运维问题与挑战，为新型数字信息技术应用落地保驾护航。

3.1 应用演进趋势

在使用信息技术之前，企业的生产经营依赖人工操作、文本记录和口头沟通。发生在 20 世纪六七十年代的第一次信息技术浪潮推动了企业经营价值链条中的关键活动的自动化。企业应用系统软件实现了从订单处理、财务管理和工程设计到生产资源计划管理的计算机辅助自动化。这是信息技术第一次在企业生产经营活动中发挥巨大的作用，

[1] https://hbr.org/2015/10/how-smart-connected-products-are-transforming-companies.

计算机将人工从海量数据采集、处理工作中解放出来，推动了生产力的提升。

在这个阶段，应用稳定性和性能保障等运维活动主要解决企业内部局域网内，面向生产、财务、销售等部门提供服务的软件和硬件系统的故障问题。应用特点是系统架构相对简单、接入用户数量固定、数据增长速度相对稳定。应用软件大多是标准化产品，有厂商提供运维支持，企业运维压力较小。

20 世纪八九十年代，互联网的快速发展带动了第二次信息技术浪潮。通过廉价、便捷的接入方式，互联网打通了用户、供应商和企业之间的信息通信交互通道。企业内部信息系统不再只联通、服务于企业内部。企业与供应商、合作伙伴、经销商、最终用户之间的信息交互成了可能。企业支撑经营管理活动的数字信息系统建设不再是购买标准化产品就能够完成了。应用系统开发、运维对企业，尤其是互联网公司的重要性快速提升。应用运维不再只解决标准化产品故障问题，而且要解决复杂多变的网络环境和系统间网状信息交互集成带来的新的问题，这也带动了系统监控类软件的快速发展，其中比较有代表性的是 Florian Forster 编写的 Collectd（UNIX 系统的软、硬件监控指标采集存储工具）[1]、UC Berkeley 开发的 Ganglia（用于分布式部署环境下的高性能计算平台的监控）[2]等。网络性能监控分析软件和应用性能管理软件也在这个阶段诞生了。

以智能、互联为主要特征的第三次信息技术浪潮将在提升生产力的同时，改变应用及其运维方式。物联网（Internet of Things，IoT）已经开始改变产品或服务的设计、生产、营销、交付和售后支持过程。迈克尔·波特教授预言，第三次信息技术浪潮将“有潜力成为目前为止影响最深远的，相比前两次会激发更多的创新，获取更大幅度的生产

[1] https:// collectd.org/.

[2] http://ganglia.info/.

收益增长和经济增长”。

企业规划建设面向未来的应用智能运维系统之前，首先要了解未来的应用系统和技术演进趋势。新需求、新技术激发的应用系统交互使用方式和开发运维方式的改变，首先体现在企业交付用户的产品形态上。互联网在已经建立的、面向人与人信息通信的网络的基础上，连接电器、汽车、家居、生产工具等产品，形成物联网。企业生产的产品将逐渐演进成为智能、互联产品。随之改变的人与产品之间的交互方式，以及随之生成的海量数据会推动企业运维方式演进。

如图 3-1 所示，智能、互联产品演进路线通常可以划分为四个阶段：在第一阶段，通过嵌入计算平台实现智能化控制能力，实现传统产品到智能产品（Smart Product）的升级；在第二阶段，通过植入联网能力，对接云平台服务和其他终端控制设备，产品演化为智能、互联产品（Smart，Connected Product），进一步优化用户体验，提升产品能力；在第三阶段，接入了更多第三方信息系统服务，为产品的智能化决策提供了更多信息，进一步扩展了产品的能力边界，这个阶段的产品称为产品系统（ProductSystem）；在第四阶段，产品系统进一步与其他产品系统能力对接，成为更庞大的系统联邦（System of Systems），不同产品系统的能力相互融合、放大，产品价值得以提升。例如，农用机械生产商 John Deere 和 AGCO 将拖拉机等农机联网信息化系统，不仅与智能终端设备对接，而且与灌溉、土壤检测施肥、天气预报、农作物价格管理、商品价格趋势预测等第三方产品系统和信息平台对接，以优化耕作流程，提高收益。在智能、互联产品升级之后，农机设备只是整个庞大系统的一部分。多系统通过协作，实现更大的价值。在这个场景下，应用运维不再只围绕独立应用系统解决一个点的问题，而是面向更大的场景，需要复杂的联邦系统协作，需要具备全景监控能力，也需要具备智能化态势感知和风险管理能力的智能应用运维系统的支撑。

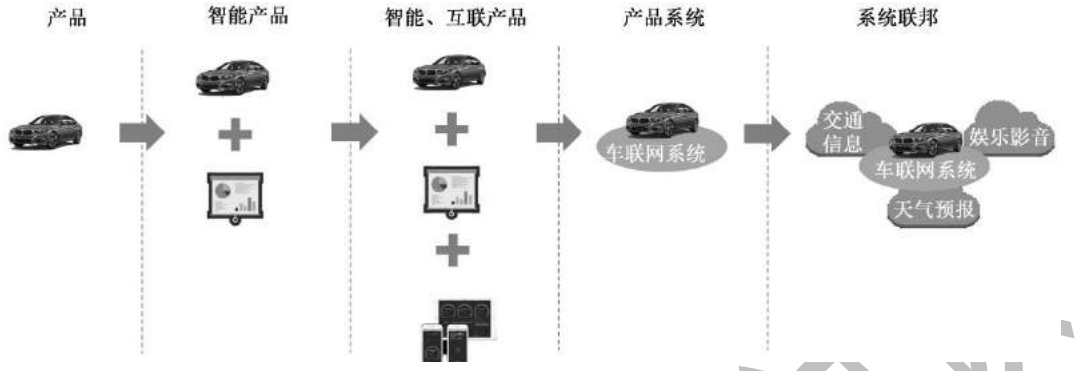


图 3-1　智能、互联产品演进路线

总的来说，智能、互联产品包含的三个关键组成部分是物理模块（Physical Components）、智能模块（Smart Components）和连接模块（Connectivity Components）。物理模块指产品物理实体存在的部分，如汽车引擎、轮胎，空调压缩机、电源等。智能模块包括状态数据采集传感器、微处理器、数据存储器、控制系统和软件，对应智能网联汽车就是引擎控制系统、下雨感知自动车窗控制系统、车载娱乐系统和汽车辅助驾驶系统。连接模块包括天线、接口、通信协议和信道等，其中，通信方式通常包含三种：一对一通信，即单个产品与用户、厂商和其他产品通信，如汽车通过OBD（OnBoard Diagnostics）接口与故障诊断系统连接；一对多通信，即集中控制系统实时或随需与多个产品连接，如新能源汽车与云端监控系统实时通信以上报电池状态数据；多对多通信，即多个产品之间或产品与多个独立系统之间进行通信，如车与车之间通信、车同时与路侧终端和云端服务通信等。

智能模块是对物理模块能力和价值的延伸。例如，空调、热水器系统通过采集的历史数据来分析判断什么时候需要将室温加热到适合的温度、什么时候需要准备好热水。连接模块通过连接云端能力和终端能力，将终端数据存储、计算任务负载卸载（Offload）到云端，通过云端随需即取的计算、存储能力来放大智能模块的价值。这样，

终端就不需要集成昂贵、复杂的数据处理分析系统。通过物理模块、智能模块和连接模块的配合协同，产品价值将循环放大，这同时意味着系统复杂度的提升和运维方式的改变。智能、互联技术与行业应用场景结合，衍生出了新一代数字信息系统（见图 3-2 中的数字化医院应用、数字银行应用等），也为应用运维的智能化建设带来了特殊的复杂性问题。

图 3-2　典型的智能、互联应用

从产品本身的功能和能力看，智能、互联产品区别于传统产品的能力主要体现在四个层级：状态监视、控制、优化和自治，如图 3-3 所示。每个层级的能力都能在目标场景中体现闭环的价值，并为下一层级能力奠定基础，如状态监视是产品控制、优化和自治的基础。企业在策划升级产品时，不仅要考虑提升产品的用户价值和自身竞争力，同时要为每一层级技术升级带来的运维问题准备解决方案。

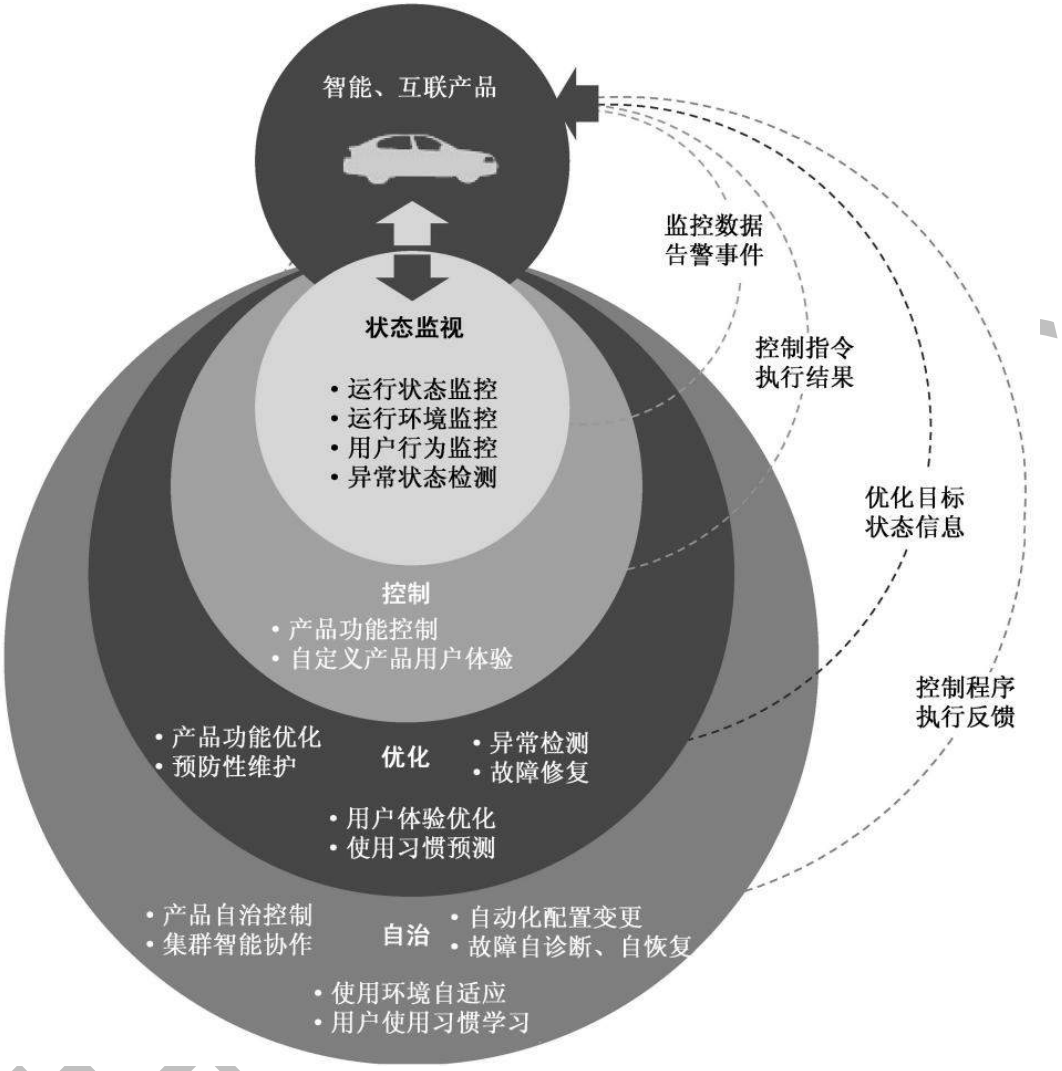


图 3-3　智能、互联产品的能力体系

1. 状态监视

监视智能、互联产品的运行状态，采集用户行为和外部环境变化等实时数据，是实现智能化管理、控制、优化、运营和运维的基础。不了解产品在用户目标场景中的使用情况和运行状态，就无法实现进一步的智能化改造。状态监视层级能够实现的能力：运

行状态监控、运行环境监控、用户行为监控和异常状态检测。应用系统与终端交互的数据主要是状态变化监控数据和告警事件。可以体现的产品价值有故障告警、发现产品缺陷、挖掘用户场景中的新需求以改进产品设计等。例如，对于新能源汽车，目前国家要求其每 10 秒给云端车厂和国家平台上报一次监控数据，一旦出现电池状态异常或车辆缺陷导致的驾驶安全风险，云端平台可以及时发现和告警；车厂通过对指定型号的汽车历史数据进行分析，可以挖掘目标用户群的使用习惯和驾驶行为特点，从而优化新款车的设计，或者指导充电桩建设地点的规划。

2. 控制

有了状态监视数据，下一阶段建设的能力目标是对智能、互联产品进行控制。通过实现控制能力，产品不但可以更好地适应目标场景用户的使用习惯，获得之前无法实现的可定制性，而且能够进一步简化用户的操控，提升用户体验。控制层级能够实现的能力：产品功能控制、自定义产品用户体验。应用系统与终端交互的数据除了监控数据，还包括控制指令和指令执行之后的结果反馈。例如，汽车的电子车身稳定装置、加速防滑控制系统、防抱死制动系统、刹车辅助系统等可实现车机端控制，简化用户的操控；某些高端车提供的通过手机 App 控制锁车、开关车窗和空调等的控制能力提升了用户体验；智能家居厂商在灯泡中加入远程控制能力，使用户能够用手机控制设备开关，甚至按需调节明暗和色彩。

对产品的控制可以通过嵌入终端的代码实现，也可以通过部署在云端的集中控制服务实现。终端代码控制响应迅速，实时性、运行可靠性高，因为程序在终端计算机以独占方式运行，不受外部网络连接和远端服务器性能的影响。但是，其计算能力有限，逻辑固定适应性差。云端控制需要产品终端设备与云端保持网络连接，由云端转发控制指

令。这种控制方式将终端控制程序运行卸载到云端，降低了终端的硬件成本，但网络时延导致其实时性差、运行可靠性低。采用哪种方式需要考虑具体的应用场景。例如，对于汽车自动辅助驾驶和自动泊车，用云端控制的话实时性不够，风险较大；而对于远程控制汽车空调，因为调节温度并不需要太高的实时性，没有占用终端计算能力实现的必要，所以，用云端转发手机指令到车端更合适。

3. 优化

状态监视和控制层级建设赋予了产品监视和控制的监控闭环能力，为建设更复杂的优化层级能力打下了基础。有了全面和丰富的监控数据，企业可以利用算法从数据中挖掘有用的信息，指导产品性能、稳定性、能效等的优化。优化层级能够实现的能力：产品功能优化、预防性维护、异常检测、故障修复、用户体验优化和使用习惯预测。对于实现了优化层级能力的智能、互联产品，数据交互包含更易于理解的优化目标和产品状态信息。我们只需要设置优化目标，调节相关参数，系统就能够自动生成优化方案，并向相关责任人反馈执行结果和状态信息。例如，在数据中心场景下，我们可以基于实时采集的基础设施温度、空气流动状态、负载、空调状态监控数据来设计数据中心能效优化系统，自动生成优化方案以控制空调的开启和关闭、调节制冷功率和冷风流向、优化数据中心 PUE（Power Utilization Effectiveness）指标；对于风力发电机，在实现了对获取电量效率监控和风叶角度控制的基础上，我们可以设计实现通过调整风叶角度来获取最大电能的优化系统。

4. 自治

自治层级的能力整合了状态监视、控制和优化层级的能力，通过智能化进一步解放人脑，从而形成无须人工干预即可应对某些场景特定任务的自治控制系统。自治系统无须人工运维干预。自治层级能够实现的能力：产品自治控制，集群智能协作，自动化配置变更，故障自诊断、自恢复，使用环境自适应和用户使用习惯学习。应用系统与终端的交互数据包含控制程序和执行反馈，如用于修改缺陷或升级自动控制策略的自动控制程序升级包、辅助诊断故障的执行日志等。在指定事件发生时，自治系统能够自动匹配解决方案以应对。例如，在云端部署面向全球提供服务的电商系统，其每天的访问热点地区会随时区变化而变化，当中国地区在白天访问量较大时，处于凌晨的美国地区的访问量就比较低；当美国地区白天的访问量增加时，中国地区的访问量则开始下降，这样就实现了自治控制的云应用，利用遍布全球的云数据中心自动控制热点跟随，当不同地区的访问量增加或降低时，启动或关闭对应地区本地数据中心的负载处理节点以提升用户体验。还有一个具备自治层级能力的智能、互联产品是扫地机器人。利用集成在终端的传感器和计算平台，扫地机器人能够自治地控制地面清洁操作，从而应对障碍物和地形变化。

具有自治能力的智能、互联系统能够应对部分已知故障或突发异常情况，并选择对应的处理策略，从而进一步降低人工参与运维工作的工作量。人工参与监控运维不再需要关注局部组件的具体指标，而只需要掌控全局运行状态和运行效果就行，只有在发生问题时才需要逐级排查问题根源，并调整处理策略。

产品智能、互联化的趋势不但影响着互联网公司和高科技公司的发展战略，而且影响着传统行业，如制造业、医疗、金融等。近几年，宝马、北汽、吉利等车厂信息化建设的速度加快，有些甚至组建了独立的公司和部门做数字化转型，自动驾驶、车与云端实时交互信息、从车端语音控制智能家电，这些已经不是概念了。A.O. Smith 公司在热

水器产品中植入了传感器，连接云端服务，采集用户的使用习惯数据，结合当地的水质、天气变化，实现了对水温的智能调节，优化了用户关怀服务。迅达电梯 PORT 科技（Schindler Elevator PORT Technology）公司通过预测电梯的需求模式，计算到达目的地的最快时间及分配合适的电梯，以使乘客快速移动，将等待电梯的时间减少了一半以上。在能源领域，ABB 的智能电网技术使其公用事业公司能够分析广泛的发电、变换和配电设备（由 ABB 及其他公司制造）的大量实时数据，如变压器温度的变化，从而提醒公用事业控制中心可能出现的过载情况，使其进行调整以防止停电。

在智能、互联产品使用场景下，驱动业务的应用系统不再是单纯连接用户和服务的 C/S（Client/Server，用户端/服务器）架构或 B/S（Brower/Server，浏览器/服务器）架构。数字信息系统应用与用户交互的终端从单纯的计算机、手机扩展到手表、电视、音响、汽车、门锁、空调、洗衣机等各种与我们日常生产、生活相关的物体。应用服务端也经历了从单体架构、垂直架构、SOA 架构、Lambda 架构、Kappa 架构、云原生架构到微服务架构的演进过程。应用运维的复杂度激增，性能和稳定性保障若只是单纯地采集服务节点状态、代码执行链路或网络运行状态，配置告警策略，已经不能满足实际生产场景的需要了。

3.2 技术演进趋势

从技术演进趋势看，虚拟化、云计算、容器、微服务等新技术正在逐渐将应用的业务逻辑与基础设施解耦。虚拟化技术将计算、存储、网络资源从物理硬件设备中剥离出来。云计算技术则将虚拟化资源形成资源池，并以自助的方式向租户交付随需即取的资源。容器技术将应用中间件从操作系统中解放出来。近些年逐渐兴起的微服务技术进一步将应用业务逻辑从中间件中剥离出来。应用本身运行对底层硬件环境的依赖逐渐减弱，映射关系的不确定性和动态性更加明显。这就导致应用运维与传统 IT 基础设施运维的目标大相径庭。由于更贴近用户和业务，应用运维的重要性和复杂度更高。

目前，金融、航空、汽车等行业都处在数字化转型的前沿，由于应用性能问题导致用户体验下降、企业用户流失和经济损失的案例在逐渐增加，而传统以应用性能管理工具和网络性能管理工具建设为主的应用运维系统的被动响应式风险处理机制已难以应对。实现主动预防的风险处理机制、建设智能化的应用性能管理平台已逐渐成为企业构建面向未来的运维体系的关键。

这些场景之所以能在今天成为现实，主要原因是，一系列新技术的发展和成熟，使得制约应用落地的障碍得以清除。其中，对数字信息系统应用的架构及开发、运维方式产生深远影响的技术如下。

1. 服务器虚拟化、云计算

近年来，首先掀起波澜的是服务器虚拟化、云计算技术的普及应用。创立于 1998 年的 VMware 公司推出的 VMware Workstation 服务器虚拟化软件将操作系统与硬件基

础设施解耦，使得软件系统不再与硬件平台绑定。2006 年，亚马逊以虚拟化技术为基础推出了首个云计算服务——AWS Elastic Compute Cloud（EC2），将数据中心剩余的计算、存储、网络资源以在线服务的方式出售。应用系统部署安装不再依赖特定的硬件和数据中心，软件定义基础设施成为可能。

2. 大数据

数据量的快速增长使得大数据存储分析技术成为研究热点。2006 年，基于 Google File System 论文[1]研发的 Hadoop 大数据存储分析平台成为行业焦点。有别于传统的结构化关系数据库，Hadoop 半结构海量的大数据存储能力和基于 MapReduce 算法的信息提取能力，为应对智能、互联场景下激增的数据量提供了解决方案。

3. 容器

出现于 2008 年的 Linux 操作系统层虚拟化 LXC（Linux Containers）技术在服务器虚拟化基础之上，通过将操作系统资源隔离，进一步将应用中间件与操作系统解耦，使得应用动态部署、更新、迁移和弹性伸缩控制更加灵活。LXC 对应的商业产品 Docker 的快速普及和应用已经证明了容器技术的商业价值。

4. 微服务

[1] Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung. The Google File System[C]. SOSP’03, Bolton Landing, New York, USA. 2003.

微服务（Microservices）技术进一步将业务逻辑和应用中间件解耦。2011 年 5 月，在威尼斯附近举行的软件架构师研讨会上，“微服务”一词被与会者用来特指业界正在普遍探索和实践的一种通用软件架构设计风格。2012 年，James Lewis 在克拉科夫的一次题为 Micro Services：Java, the Unix Way 的演讲中介绍了这些新想法。他描述了通过“分而治之”的方式使用康威定律（Conway’s law）来构建软件开发团队的一种更敏捷的软件开发方式，并把这种方式称为“微服务”。利用微服务架构和技术，应用业务模块被拆分成独立的微服务节点，以方便复杂系统的多团队协作开发、更新和测试；由于业务模块对应微服务节点的独立部署，其扩展性更高；每个微服务节点可以由不同语言、不同架构实现，支持对接遗留系统服务，业务需求变化导致的对应应用系统的架构重构不影响其他微服务节点。

如图 3-4 所示为某电子商务应用系统，在传统单体架构中，所有应用的业务代码部署在一个独立的服务节点上，运行在一台应用服务器上，代码耦合度高。一个独立服务对应的开发团队需要在同一个开发框架中使用一样的技术堆栈和开发语言。所有业务访问数据库，需要通过统一的数据访问层接入数据库。一旦需要升级功能、修改缺陷，所有代码需要重新编译发布。而微服务架构将电子商务系统中的服务配送、查询详单、接收订单、结账收款等业务功能解耦，使其成为可以独立开发部署的节点。各服务通过服务发现、注册方式进行管理，并通过接口交互。每个节点可以采用不同的架构、开发语言，可以有自己的数据库。这样，业务逻辑多样且多变、架构复杂的互联网和物联网应用系统，可以通过多团队协作开发来划清任务目标和功能边界，不再局限于一个统一的技术堆栈。虽然微服务架构优点很明显，但并不完美。在解决多团队协作问题的同时，微服务架构也加剧了系统的复杂程度，使系统的运行维护成本激增，数据量增加。

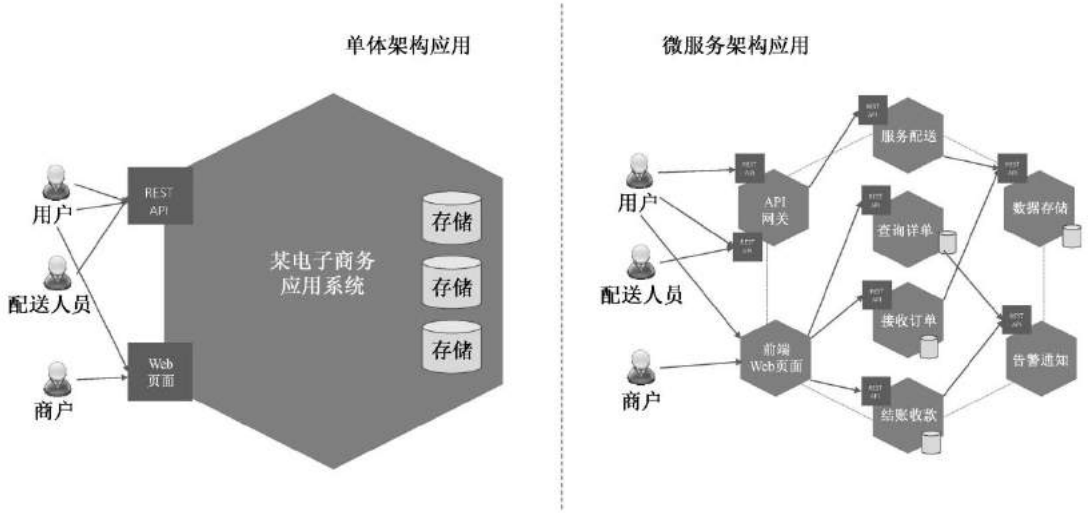


图 3-4　单体架构应用与微服务架构应用的结构对比

5. 人工智能

在计算机科学中，人工智能（也称为机器智能）是机器通过算法实现的智能。人工智能研究领域诞生于 1956 年达特茅斯学院的一个研讨会上，约翰·麦卡锡（John McCarthy）提出了“人工智能”一词[1]，以区分该领域与控制论，并摆脱了控制论专家诺伯特·维纳（Norbert Wiener）的影响。人工智能技术被认为是推动第三次信息技术浪潮的关键技术。近几年来，人工智能发展迅速，产业界和学术界对相关技术的研究、落地兴趣很浓。随着硬件平台计算能力的提升和算法的突破，人工智能的应用场景越来越多。例如，人脸识别应用于身份认证，图像识别应用于海量图片处理和搜索，异常检测和因果推理分析算法应用于海量机器数据的处理等。

[1] John McCarthy，Crevier，Russell Norvig. McCorduck [C].Dartmouth conference, 2004.

3.3 应用智能运维系统：企业数字战略的关键支撑

根据 Forrest 的统计数据，57%的企业用户 IT 运维部反馈，至少每周会发生一次影响应用性能和可用性的问题；每天都发生问题的比例占到了 28%。对于愈加依赖应用来面向用户以实现企业价值、提高工作效率的当今企业来说，这种问题越来越无法忍受。统计数据显示，超过一半的企业认为应用性能问题直接导致业务用户和 IT 部门效率降低；42%的企业认为应用性能问题直接影响了企业收入。当前，企业应用运维团队的压力主要来自以下两个方面。

（1）新需求推动应用数量激增。移动智能终端设备的普及使应用逐渐渗入我们工作、生活的方方面面，企业应用数量激增。企业面向用户、合作伙伴和内部员工建设的应用数量会随产品智能、互联化的深入持续增长。

（2）产品数字化导致应用结构愈加复杂，保障应用性能更困难。在技术方面，如混合云、数据分析、物联网、车联网、体域网等新技术的持续演进使得应用结构愈加复杂，保障应用性能更加困难。据统计，超过一半（52%）企业的 IT 运维部门在监控管理工具上的投入是被动、针对特定问题且分散的。这种投入方式虽然可以有效地解决当前的问题，但由于管理功能单一、分散、碎片化，难以应对未来以应用为核心的新需求和技术演进。随着时间的推移，现有应用运维问题会恶化。因此，采用被动处理方式的应用智能运维系统已不能满足企业快速数字化转型的需要，主动分析定位潜在问题、预防应用性能风险已成为未来应用智能运维系统的发展趋势。

自动化过程是将人手从简单重复的劳动中解脱出来的过程，而智能化过程则通过将经验和思维逻辑固化为算法，将人脑解放出来。对于智能、互联时代的应用运维场景，人工处理的速度已经远远跟不上运维工作量增加的速度，用机器智能解决机器复杂性问题是目前可行的解决方案。

随着信息系统的快速演进，企业对应用运维系统的期望也在上升。Gartner 于 2018 年 12 月发布的分析报告指出，企业对应用运维能力的需求核心正在从应用请求链路监控、用户数字体验保障向智能运维、业务流程监控、应用全景监控转移，如图 3-5 所示。

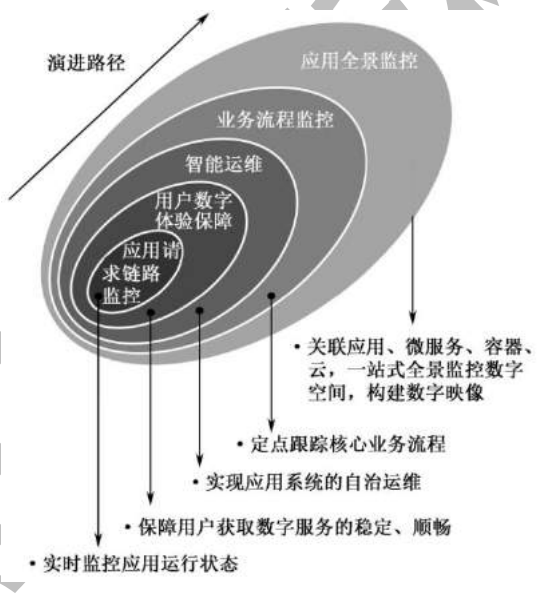


图 3-5　应用运维系统的演进路线

传统以 APM 平台提供的以应用代码监控分析能力为核心的应用性能监控运维体系，正向以用户数字体验保障为核心的方向演进。大数据、人工智能技术的发展使得监控系统有能力从海量数据中提取有用的信息，实现更符合应用运维需要的业务流程监控和全景监控。

3.4 商业价值评估（ROI 分析）

建设能够满足智能、互联时代应用运维需求的智能运维系统，意味着要对原有运维体系的监控数据采集、数据存储、数据分析的工具，以及运维流程和人机交互界面进行全面升级。我们在决策是否值得投入建设时，需要先判断 ROI（Return On Investment）是否能达到预期。

比较可行的计算办法是在系统目标场景下挖掘相比于现有方式能够改善、升级的价值点，选择可量化的指标计算 ROI。例如，常用的系统可靠性量化指标有故障平均修复时间（Mean Time To Repair，MTTR）、平均无故障工作时间（Mean Time Between Failure，MTBF）、平均失效前时间（Mean Time To Failure，MTTF）[1]和标准化的用户体验指标（Application Performance Index，APDEX）[2]。MTBF 即平均失效间隔，就是从新的系统在规定的工作环境下开始工作到出现第一个故障的时间的平均值。MTBF 越长，表示系统的可靠性越高，正确工作能力越强。MTTR 就是从出现故障到恢复之间的这段时间。MTTR 越短，表示系统的易恢复性越好。MTTF 就是系统平均能够正常运行的时间。系统的可靠性越高，MTTF 越长。APDEX 是从用户的角度评估系统使用体验的标准化指标。它提供了测量和报告用户体验的标准化方法，将用户体验量化成范围为 0～1 的满意度评价数值，把最终用户体验和应用性能联系在一起。

以某快销企业为例，其现有面向终端用户的营销平台、冷链管理等生产管理系统、用户关系管理系统和 ERP 等百余个系统。其应用运维团队有 40 人，负责日常的应用性能、可用性保障。根据历史数据统计，每年导致应用服务中断的严重故障次数平均为

[1] https://wiki.mbalib.com/wiki/MTTR.

[2] https://docs.newrelic.com/docs/apm/new-relic-apm/apdex/apdex-measure-user-satisfaction.

22 次。运维人员平均工作负荷为 120%，主要是由收到异常告警、需要处理突发事件、加班排查故障导致的。每次出现严重故障的平均故障恢复时间为 20 小时左右。应用持续稳定运行的时间为 219 小时。

该企业规划升级现有运维体系，实现应用系统的集中监管，打造具备监控指标集中存储、风险主动探查和根源定位分析能力的应用智能运维系统。通过技术可行性评估，结合历史运维场景，对现有运维流程进行优化估计，量化的 ROI 数据如表 3-1 所示。其中，在每年 22 次历史故障中，通过引入可用性主动拨测机制和全景监控能力，可以提前发现规避的故障有 10 次，按每次故障损失 12 万元计算，每年收益达 120 万元。采用自动化拨测应用关键业务流程的可用性，以及故障信息自动关联辅助根源问题分析，使得人工巡检和分析监控数据的工作量减少了约 1/4，应用团队规模可以缩减 10 人。按人均年成本 20 万元计算，年收益达 200 万元。

应用智能运维系统规划建设的主动探伤扫描功能可以每天自动分析应用的潜在风险，降低突发故障导致系统宕机的概率。运维人员加班处理突发问题的时间减少，工作负荷相应地可以降低到 90%，节约成本约 80 万元。由于意外故障导致的宕机次数减少，MTTF 可以从平均 219 小时提升到 438 小时，对应规避的运营损失（包括最终用户流失、代理经销商业务终止、故障恢复人力成本投入等）总计 76 万元。相比于现有系统，应用智能运维系统通过整合数据，深度分析和定位影响用户使用的性能瓶颈。应用性能的提升意味着用户体验的优化，APDEX 可以从目前的 0.75 提升到 0.92。对于运营部门来说，相关用户转化率有明显的提高，据运营部门估算，这对企业经营带来的可度量收益大概在 90 万元左右。经过整体评估，企业建设应用智能运维系统每年带来的可量化 ROI 为 7 36 万元。

表 3-1 应用智能运维系统建设 ROI 评估:

指标	运维现状	期望效果	收益/（万元/年）
每年应用严重故障次数	每年 22 次严重故障	每年 12 次严重故障	120
应用运维团队人员数量	40 人	30 人	200
应用运维人员平均工作负荷	120%	90%	80
MTTR	20 小时	6 小时	170
MTTF	219 小时	438 小时	76
APDEX	0.75	0.92	90
年收益总计			736

有了数据支撑，我们就可以进一步明确建设目标和愿景，并对规划建设的特性优先级和建设成本有相对准确的估计。ROI 估算只是第一步，接下来需要梳理目标场景，深入理解系统在实际场景中可以发挥的价值。对场景和实际需求的理解很大程度上决定了系统能否够达到期望效果。总的来说，应用智能运维系统的场景化价值主要有以下几点。

1. 实时感知风险态势，减少应用宕机损失

监控的目的是发现风险，在智能、互联时代，发现风险需要强大的监控系统的支持，著名的监控系统——宙斯盾和彭博终端的核心价值都是在复杂态势中找到风险点。应用运维也类似，在系统复杂度快速增加、接入用户终端设备多样化、系统间交互集成关系更紧密的背景下，应用智能运维系统的全景监控和智能化态势感知能力对企业更加必要，价值也更大。实现风险态势感知的前提是有全面、实时、丰富的监控数据。

信息化建设发展到今天，大、中型规模的企业几乎都会建设 IT 系统的监控系统来监控应用和应用运行环境状态。常用的监控系统基本上都是针对一个点进行数据采集和风险告警的。例如，网络性能监控工具 nTop[1]能够对网络中的网络包进行拆包分析，监控当前网络上信息交互应用的流量异常；开源网络及应用监控工具 ZABBIX[2]常用来对 IT 基础设施和中间件进行监控；应用性能管理工具 Pinpoint[3]擅长监控应用请求执行代码链路和追踪分布式事务执行过程异常；Logstash[4]、ElasticSearch[5]是用来对应用日志进行存储分析的常用工具。这些系统的数据采集、存储和风险告警相对独立。一个完整的智能、互联应用系统的部署架构和数据交互复杂，往往需要多种工具联合使用。对于运维人员来说，这些就像一个个数据孤岛，一旦发生异常，多套系统都有可能产生告警，形成告警风暴。要排查和定位问题根源，需要人工登录多个门户查询历史数据，因此系统易用性差，运维工作效率低。

应用智能运维系统首先能解决运维孤岛问题。如图 3-6 所示，通过搭建由不同类型的存储平台组成的运维大数据湖，将 ZABBIX、nTop 等的监控数据同步采集到一个集中的存储平台来做数据同构转换、清洗、聚合、统计等分析处理，为状态监控、异常检测、根源问题定位等应用场景提供一致的数据存储。一旦发生异常告警，风险点对应用整体运行态势产生影响，受影响的终端用户和业务流程能很快被定位出来。这样，人工介入处理数据、发现和定位风险的工作量减少，MTTR 会有一定幅度的减少。

[1] https://www.ntop.org/.

[2] https://www.ZABBIX.com/.

[3] http://naver.github.io/pinpoint/.

[4] https://www. elastic.co/cn/logstash.

[5] https://www. elastic.co/cn/elastic-stack.

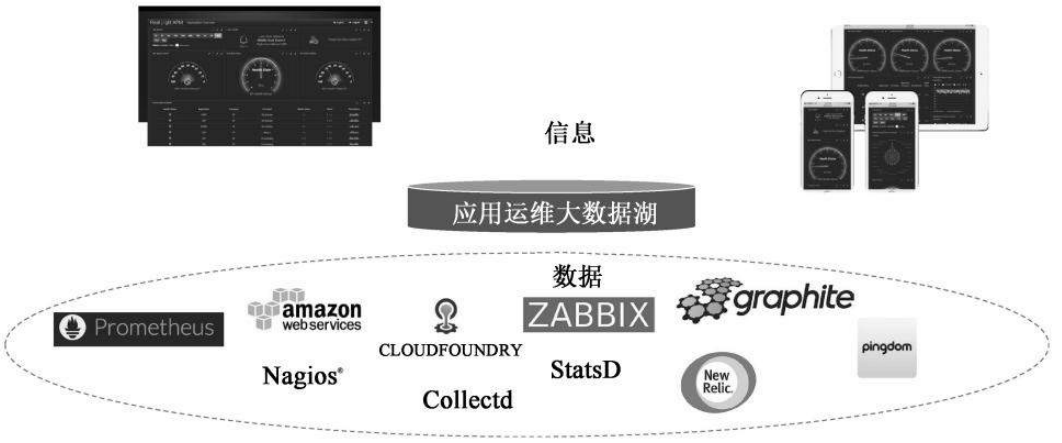


图 3-6　运维大数据湖打通运维数据孤岛

2. 提供专家经验指导，提高应用运维效率

智能化的关键支撑是经验和知识的积累，应用智能运维系统建设区别于其他监控运维系统的关键一点是，在发生异常或出现潜在问题的情况下，其能够通过算法和积累的专家经验来指导风险的发现、定位和处理，辅助决策支持。传统监控运维系统积累专家经验主要依靠告警策略、监控运维仪表盘和报表。告警策略针对时间序列指标数据配置自动探测异常的逻辑，出现问题自动生成告警；监控运维仪表盘和报表通过预定义模板的方式对指定类型的资源、监控场景或故障最常用的指标进行统计分析，并生成对应的可视化界面。开源监控数据可视化平台 Grafana[1]专注运维数据可视化，提供了大量根据经验定义的可视化仪表盘模板。利用类 SQL 查询语句，Grafana 将常用指标聚合、统计和展现策略固化为可下载的模板，并通过开源社区的方式让全球用户接入下载或分享自己的仪表盘。

[1] https://grafana.com.

除此之外，知识图谱与运维场景的结合也是解决运维专家经验积累和使用的可行途径。知识图谱（Knowledge Graph）[1]是实现人工智能落地的重要基础，它以结构化的形式描述客观世界中的概念、实体及其关系，将互联网的信息表达成更接近人类认知世界的形式，提供了一种更好地组织、管理和理解互联网海量信息的能力。知识图谱不是一种新的知识表示方法，而是知识表示在工业界的大规模知识应用，它将互联网上可以识别的客观对象进行关联，从而形成客观世界实体和实体关系的知识库，其本质上是一种语义网络。如图 3-7 所示，其中的节点代表实体（Entity）或概念（Concept），边代表实体/概念之间的各种语义关系，如用户（User）拥有某站点（Site）的管理员权限，在用户和站点两个实体之间，会有一条线标识拥有管理权限（has_administrator）。

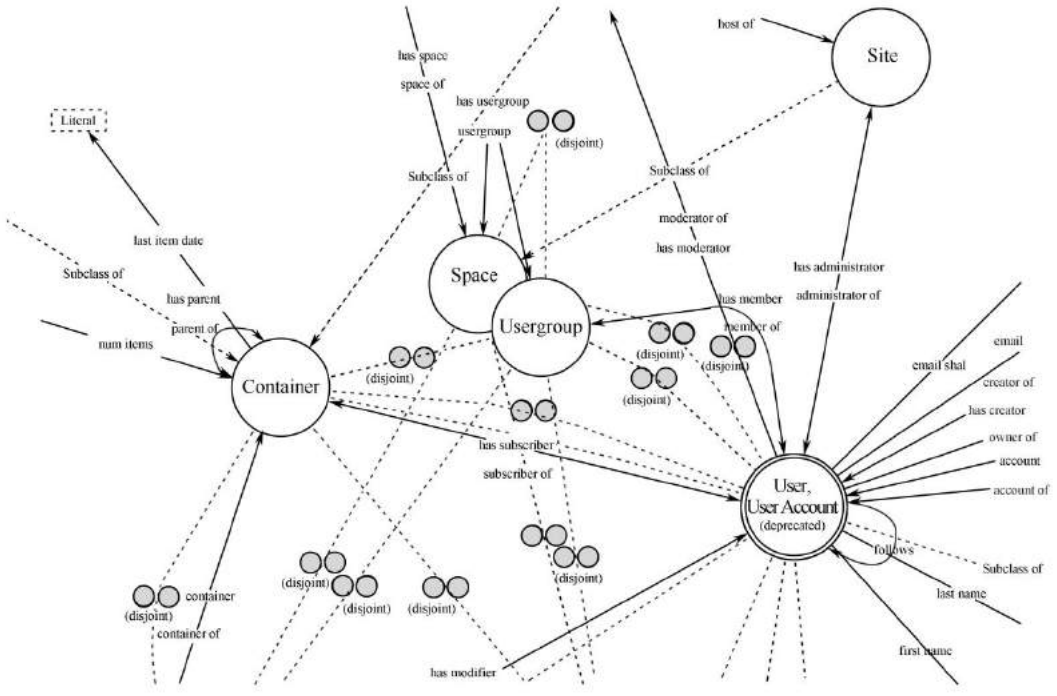


图 3-7　知识图谱语义网络模型示意（局部）

[1] https://google.fandom.com/wiki/Knowledge_Graph.

有了积累的专家知识和经验，我们就能够在发生异常且缺少专家指导的情况下，利用应用智能运维系统自动检索和匹配知识库信息，解决疑难问题，为企业降低人工成本。

3. 主动找出故障原因，提前预防和规避风险

有了积累的专家知识和经验，应用智能运维系统能够帮助我们利用这些知识和经验管理风险。具体场景：①在未发生风险时，通过设定先验条件来推理和判断系统是否可能出现性能瓶颈或故障，若可能，分析问题所在；②在已经发生了风险告警时，回溯数据到故障点，结合知识和经验推理及分析原因。

第一种场景重点是预防和规避风险，在故障出现之前就能解决问题，对企业的价值更大。例如，电商平台在既定时间进行线上营销活动，从历史数据可以预估确定时间点在线用户数量的大概范围。在线用户数估计值就是先验知识，利用从历史数据中学习得到的知识和经验模型推理分析就可以预判在此负载条件下，哪些指标会出现异常。从指标可以梳理出可能发生的性能瓶颈、潜在故障等，从而指导扩容或配置变更，以便减少应用宕机风险。图 3-8 为面向汽车故障诊断的概率图模型（Probabilistic Graphical Models）因果推理网络示意。其将每种影响稳定运行的状态指标的取值离散化，然后通过输入先验知识来推理其他指标的后验概率分布，从而判断最可能出故障的点。

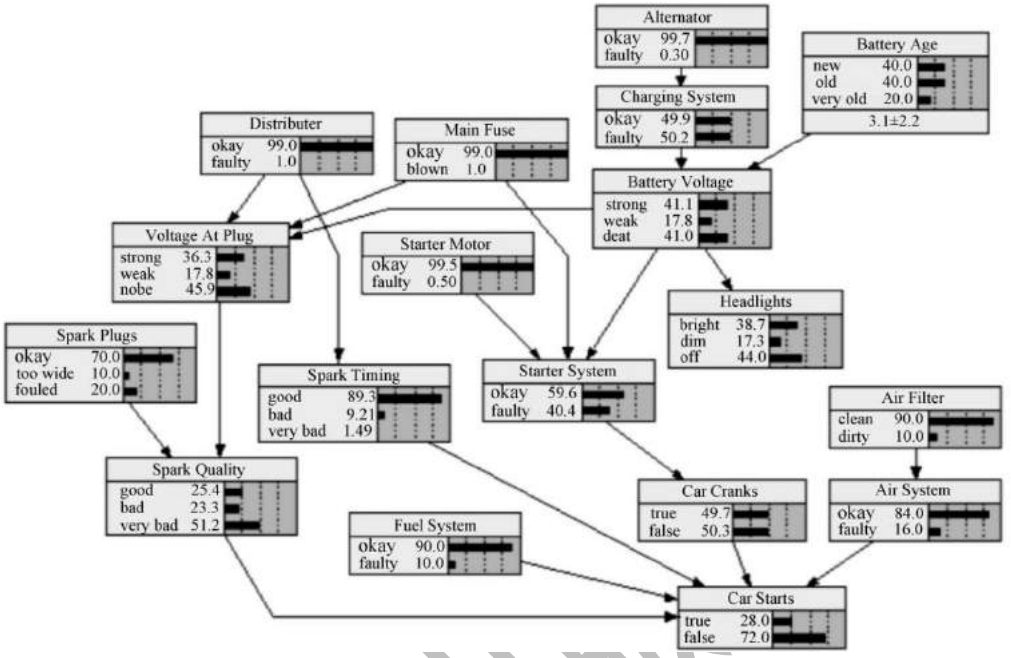


图 3-8　概率图模型因果推理网络示意

第二种场景是在故障发生时，利用提前学习生成的指定故障因果关系概率图模型，从高维海量监控数据中查找相关信息，辅助定位根源问题，从而缩短 MTTR。例如，利用知识库推理分析算法排查应用运行环境指标间的因果影响关系，定位出 HTTP 错误事件和 Java 内存使用率指标异常之间的相关性较强，从而可得出 Java 内存溢出导致应用宕机，进而导致用户 HTTP 请求错误。

4. 辅助容量规划决策，节约资源采购成本

大多数企业在新应用上线或扩容规划时，对需要准备多少计算、存储、网络资源，资源在应用系统中每个独立部署的节点之间如何分配，都缺少经验和有效的历史数据支撑。建设应用智能运维系统后，企业就可以通过算法分析全量采集的应用历史数据，从而进行决策。

区别于直接采集、分析应用性能管理监控数据和应用运行依赖的基础设施环境监控数据做容量规划分析，应用智能运维系统需要首先将业务流程请求处理链路、应用节点运行状态指标和对应的运行环境状态指标关联，从历史数据中筛选指标波动相关性。有了这些信息，我们能分析出各业务流程的历史峰值，以及在峰值发生时其对哪些服务节点和对应的运行环境状态指标有相关性影响。例如，计算密集型业务的并发量增加，对应节点的 CPU 利用率会显著升高，因此，我们需要判断对应节点的 CPU 利用率增加是否会使业务执行时间超时，以及使请求的数量超过服务质量目标的约束。如果通过算法计算发现有指标波动相关性，那么就意味着需要扩充服务节点的计算能力。

图 3-9 是应用容量规划决策支持样例。我们利用算法预测未来负载的变化趋势，通过历史数据推理分析什么时间段会导致哪种资源利用率增高。计算维度包含了 CPU 使用率、Java 内存使用率、交换空间使用率等常用相关资源的使用率。一旦发现未来某时刻可能负载会增加，则对应的某些资源使用率会不会超标，以及需要额外增加多少资源就都一目了然。

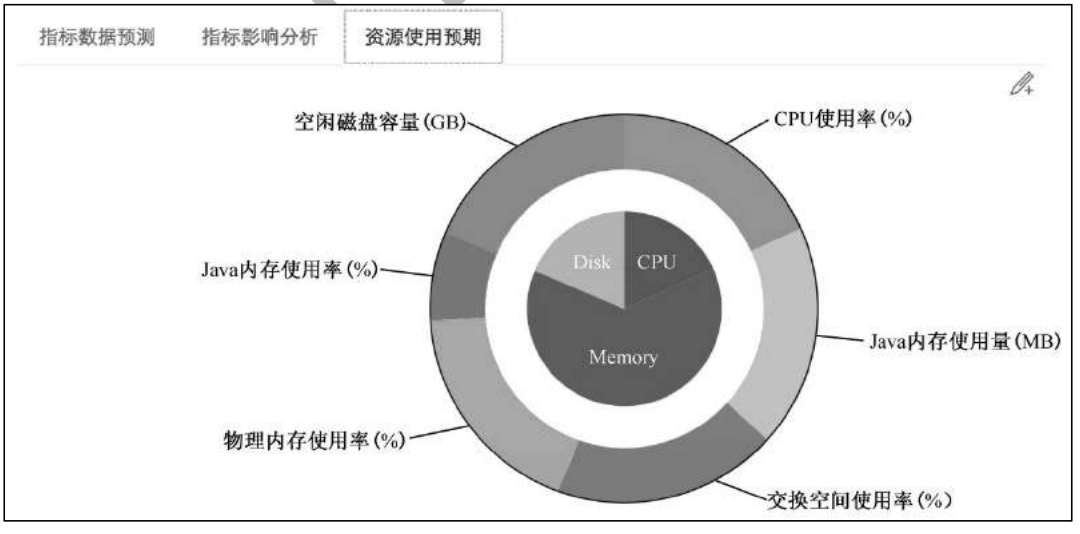


图 3-9　应用容量规划决策支持样例

5. 掌控全局业务状态，赋能业务数字化运营

应用智能运维系统通过整合多种运维产品监控数据，利用人工智能算法代替人工来挖掘数据中的信息。这种能力使得企业能够在未来智能、互联时代建设业务逻辑更加复杂的数字信息系统，支撑产品和服务能力升级。全景监控能力对企业的价值主要体现在用户数字体验保障和复杂应用系统的整体健康状态保障两方面。

如图 3-10 所示，对于运营场景，为了保障用户数字体验，运营人员关注用户侧使用情况和对应的应用侧业务流程的健康状态，对应用本身的服务节点状态和运行环境基础设施运行情况不太关注。因此，全景监控视图需要实时监控关键业务流程的运行状态，一旦出现问题，能够反映其对用户关注的业务的影响。

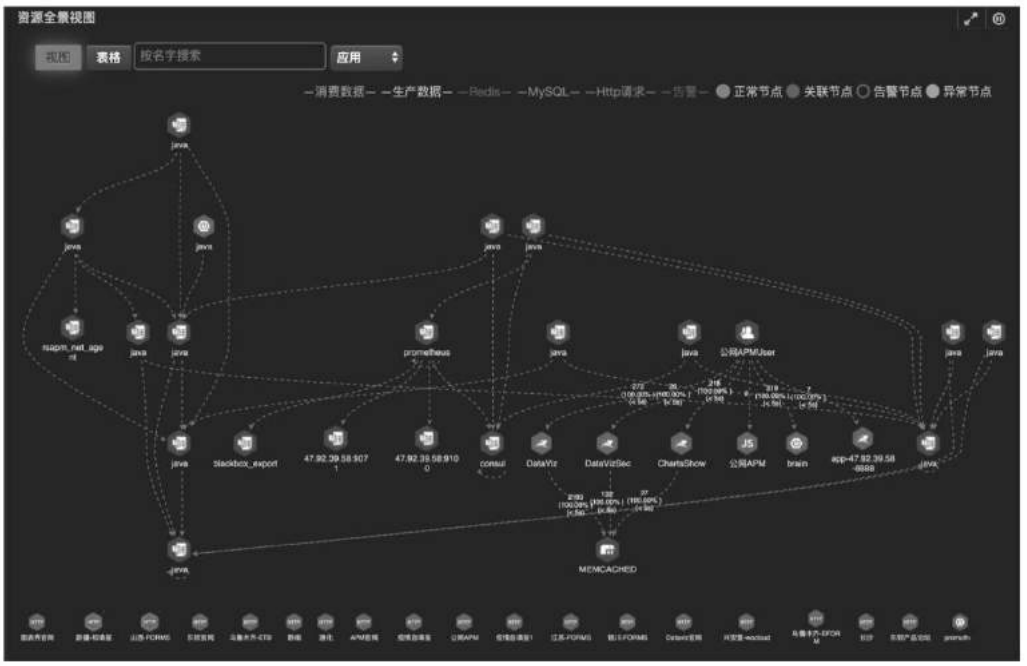


图 3-10　智能、互联应用全景监控视图样例

在运维对复杂应用系统的整体健康状态保障的场景下，监控重点也要从具体技术组件和运行环境向业务流程转移。微服务化、容器化使得应用本身的部署架构和数据交互关系更加复杂。逐个排查具体节点的运行状态，工作量会非常大。因此，运维思路需要从局部到整体，以业务流程为根节点逐级关联子业务流程和相关服务节点，如图 3-11 所示。一旦出现故障，运维可以快速评估影响范围，定位根源问题。

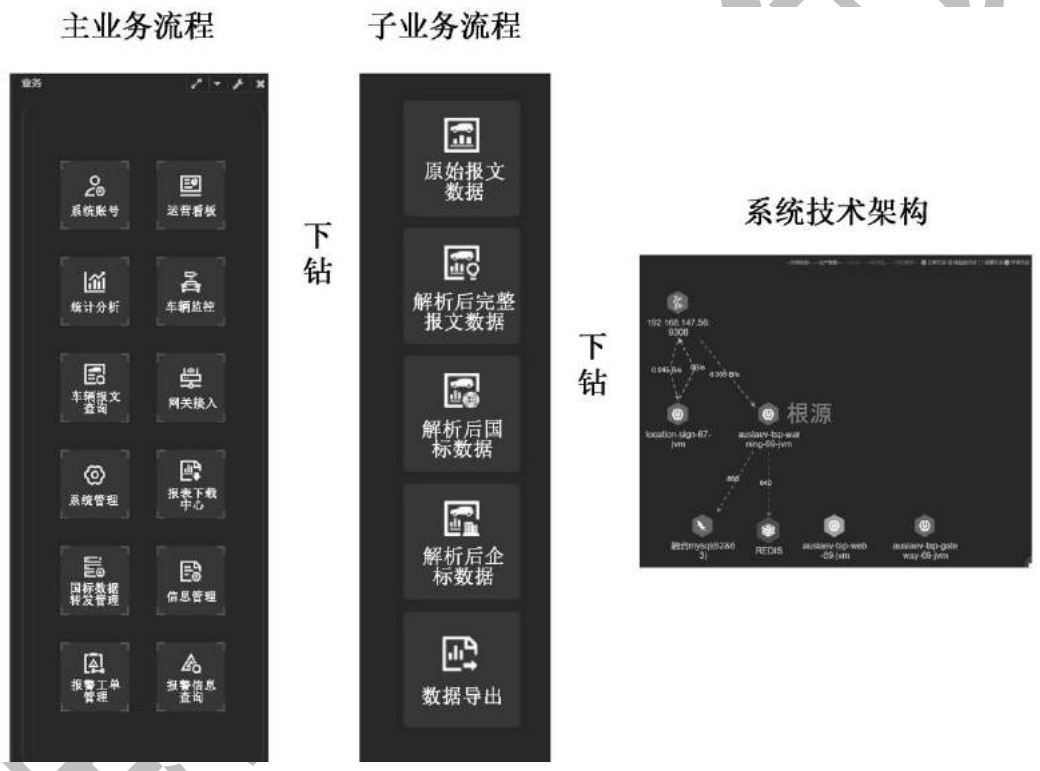


图 3-11　业务流程与系统技术架构的关联关系

案例：LinkedIn 应用智能运维建设方案

成立于 2003 年的 LinkedIn 自始至终以“为更好的工作机会连接用户人脉网络（to your network for better job opportunities）”为经营宗旨。公司信息系统复杂度

随业务增长快速增加。截至 2015 年年底，LinkedIn 拥有超过 3.5 亿用户，系统每秒处理的请求数量过万，触发后端系统查询量达百万级别。

公司工程部主管 Prachi Gupta 在 2011 年一份内部报告中强调了监控系统的重要性：“在 LinkedIn，我们一直在强调我们系统网站应用可用性保障的重要性，要保障我们的会员在任何时候都能够使用我们网站上的所有功能。为达到这个目标，我们要能够在问题发生时就探测到故障或性能瓶颈，并及时做出响应。因此我们使用具备时间序列数据展现能力的监控系统来实现分钟级的故障检测和响应。这些监控工具和技术已经被证明是必需的。它们为系统运维工程师检测、探伤、解决问题争取了宝贵的时间。”

2010 年，LinkedIn 建设了大量监控系统来覆盖应用运行期的方方面面，采集了大量监控指标数据，如图 3-12 所示。但是，开发工程师、运维工程师如何获取这些数据成了难题，更谈不上分析数据、获取信息了。因此，LinkedIn 启动了 Eric Wong 提出的夏季内部项目，这也促成了 InGraphs 系统的研发和投产。

写道，“仅仅是获取某些特殊服务的宿主机 CPU 使用率这种基本指标，都要填写工单，由某些人花费大约半小时时间来整理一份报告”。当时，LinkedIn 正在用 Zenoss（一款以应用基础设施为核心的监控软件）采集指标数据。Wong 解释说，“从 Zenoss 中获取数据需要逐级浏览响应缓慢的 Web 页面，所以我写了一些 Python 脚本来加速这个过程，虽然还得花时间手动配置所要采集的指标，但从 Zenoss 中抓取数据的过程已经大大简化了”。

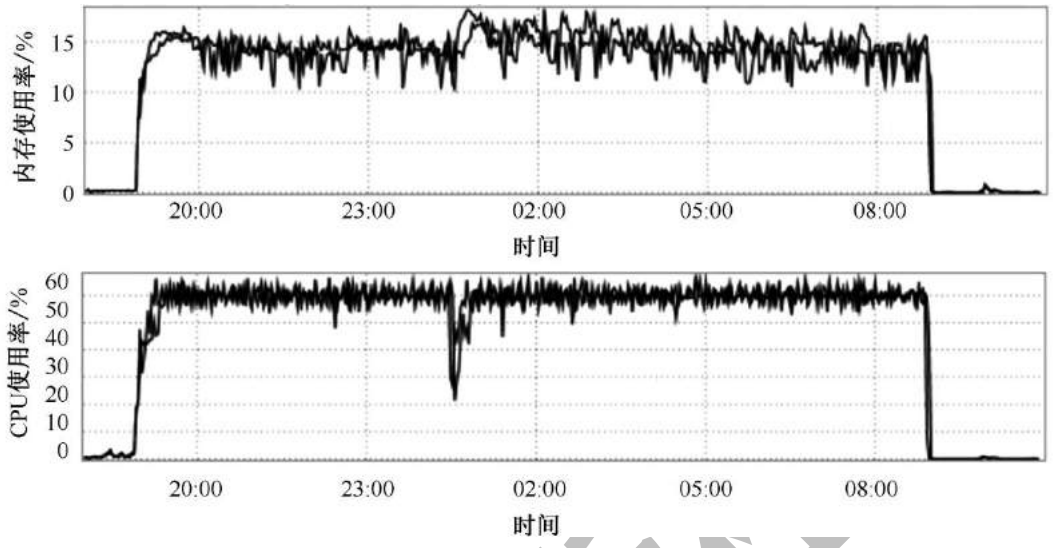


图 3-12　LinkedIn 采集的监控时间序列指标

在持续了一个夏天的研发之后，Wong 又陆续研发完善了 InGraphs 的功能，使得开发工程师、运维工程师可以从中获得需要的监控指标数据，并实现了跨多个时间序列指标数据集计算，每周变化趋势统计，历史数据环比、同比计算和监控指标自定义仪表盘自助选择等实用的功能，如图 3-13、图 3-14 所示。

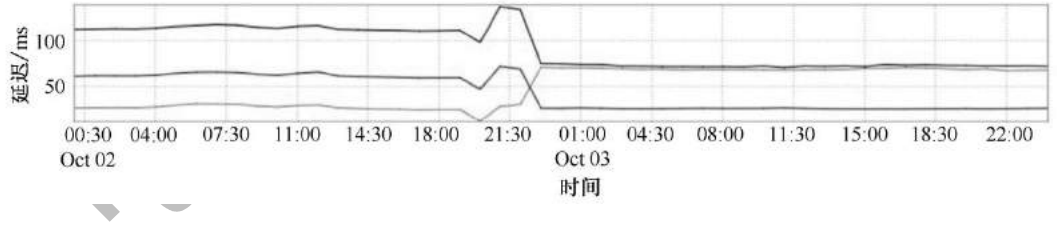


图 3-13　InGraphs 系统监控效果

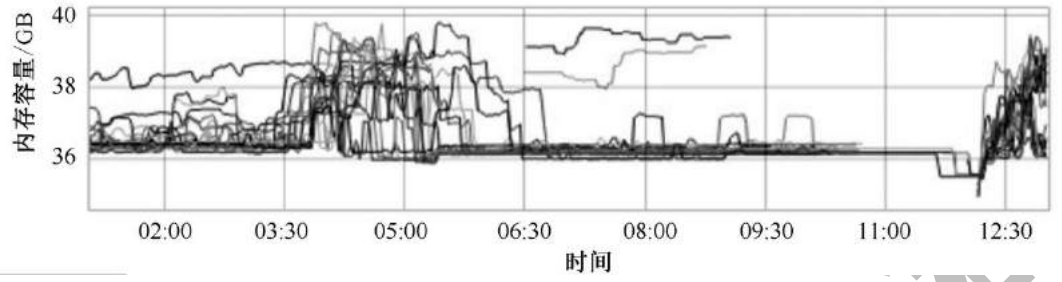


图 3-14　InGraphs 多指标历史数据对比

关于研发、完善 InGraphs 功能和它本身的价值，Gupta 表示，“在一个关键的 Web-mail 服务开始有趋势显现故障时，InGraphs 系统及时发现了，并在该应用维护团队意识到问题之前通知了相关责任人，这使得 InGraphs 监控系统的价值被公司认可”。

从一个初级项目孵化出来的 InGraphs 系统，目前已经成了 LinkedIn 运维体系中的关键组成，以至于 InGraphs 的时间序列数据监控图表遍布公司工程部门，成了最引人注目的部分。

3.5 系统关键能力

如果企业无法抵消信息系统趋于复杂化带来的运维风险，数字化营销、数字化生产、数字化管理等战略就是空谈。建设具备全景监控、智能运维能力的应用性能管理系统，保障用户数字体验，提升应用可用性，已成为企业必然的选择。

随着信息系统的快速演进，政府、企业对数字信息系统应用的依赖持续上升，对相应的应用性能、稳定性保障系统建设的关注同步升温。而传统以应用指标采集为主的 APM 系统已经难以满足云化、容器化、微服务化的复杂应用系统的监控运维需求。某知名 IT 咨询公司发布的最新分析报告指出，企业对 APM 能力的需求核心正在从应用请求链路监控、用户数字体验保障向智能运维、业务流程监控、应用全景监控转移。在此市场背景下，要保障政府、企业未来日趋复杂、多样、高负荷的数字信息系统建设，需要新一代以应用为核心的智能化全景运维平台的支撑。要打造用户体验优先的应用智能运维系统，其需要具备的核心能力如下。

1. 全景视图监控，实时掌控用户数字体验

应用智能运维系统能够自动探测和发现应用从用户端到服务端的端到端全栈拓扑结构、用户操作业务流程和代码执行链路，实时感知潜在风险并通知相关责任人，以全景化的应用监控视图展现用户请求触发的应用行为，监控范围涵盖从用户端到服务端的各环节。一旦出现风险，运维人员可以及时从全景监控视图观察到风险点，并能够下钻到原子指标或代码链路、日志等白盒监控数据，将其发送给开发人员解决处理，如图 3-15 所示。

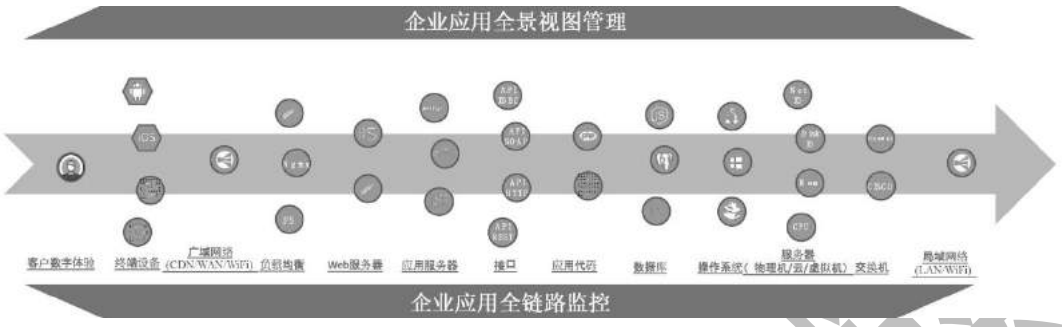


图 3-15　从用户端到服务端的应用监控全景

2. 运维大数据可视化，自助定义监控视图

应用智能运维系统能够支持自助、实时提取监控数据，定义可视化监控仪表盘视图，设置仪表盘间的跳转关系。监控视图可让海量运维数据更易理解，风险监控更及时、更直观。图 3-16 所示为可视化运维监控大数据仪表盘样例，只有通过全可视化界面实现信息的高效人机交互，才能满足未来应用运维的需要。

图 3-16　可视化运维监控大数据仪表盘样例

3. 应用全栈集中监管，全方位掌控应用的运行状态

应用智能运维系统能够提供对应用 360 度全方位、全栈的监管能力，不但能够对应用进行请求、事务、线程及代码级的深入分析，而且支持对应用依赖的应用服务器、数据库、虚拟化环境、云环境及主机、网络、存储等基础设施进行监管，帮助用户了解并掌控应用的性能、健康状态、风险及用户体验。

4. 聚合监控指标数据，简化日常应用性能管理工作

为简化对海量监控指标的监管工作，应用智能运维系统以聚合指标指示关键应用性能指标。通过指标聚合，应用智能运维系统将海量应用性能指标转换为容易理解、管理的应用健康状态、用户体验指标等指标，并通过仪表盘实时更新。这些指标反映了应用运行的全局状态，避免了人工筛查指标数据，定义了大量、复杂的告警策略，从而提高了管理效率。

5. 管理用户体验，追踪用户实时、历史在线状态

保障良好的用户体验是应用性能管理的最终目标。应用智能运维系统支持实时监控 APDEX，帮助用户掌控应用的用户体验变化情况。为实现更高效的敏捷管理，应用智能运维系统以用户体验保障为核心，提供能够追踪用户实时和历史在线状态、请求响应时间、请求异常状态等关键指标的驾驶舱式集中监管仪表盘。

6. 辅助性能优化，智能分析运行缓慢的业务流程

应用系统支撑企业运营的各环节，每个业务流程都对应众多的服务及功能调用，一旦某业务运行缓慢，会直接导致企业运转效率下降，甚至停滞。因此，在出现问题时，定位瓶颈所在并解决问题的及时性直接关系企业的营收指标。应用智能运维系统能够通过分析海量运维数据，查找指定时间段内运行缓慢的业务请求及对应的应用执行线程，快速定位应用性能瓶颈所在，从而提高解决业务响应缓慢问题的工作效率。

7. 应用白盒监控，深度分析应用性能风险的根源问题

在应用系统性能异常时，应用智能运维系统能够通过自上而下、逐层钻取应用堆栈的方式分析根源问题，生成指定时间段内的详细性能分析结果视图（见图 3-17）。分析结果视图涵盖应用行为、性能指标、异常日志、内存用量分析等几乎所有应用运行期的关键运维数据，这些数据可以帮助用户快速排查、分析应用性能异常的原因。

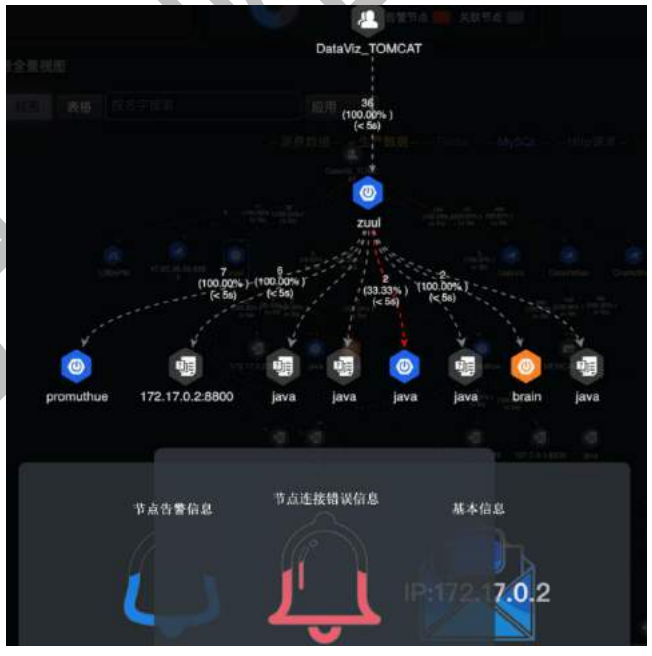


图 3-17　应用性能分析结果视图

8. 变被动处理为主动防御，提前规避应用性能风险

要从根本上扭转当前企业面临的应用性能管理被动，甚至有时近乎失控的局面，首先需要变被动解决风险告警为主动解决潜在问题及风险。有别于其他 APM 产品，应用智能运维系统致力于打造主动防御型应用性能管理体系，使企业能够提前发现风险，防患未然。基于概率图模型构建的指标间因果影响关系及推理分析模型，可使应用智能运维系统分析和处理海量数据，并通过自主研发的运维数据深度学习技术，从应用性能历史数据中分析最小粒度的指标，计算运维数据间的复杂概率分布，然后基于数据自动生成关联关系、影响程度等信息，从而生成可进行预测分析的数学模型。利用此模型，应用智能运维系统能够在给定时间范围或预期负载条件下发现潜在问题及风险，提升用户体验，减少由应用稳定性、性能问题带来的经济损失。

9. 预测应用性能变化趋势，优化应用资源配置

通过分析运维数据，生成对应用性能、负载及容量未来变化趋势进行预测的预测分析模型，应用智能运维系统能够帮助企业提前发现应用资源配置存在的问题，定位如 CPU、物理内存、Java 内存、物理磁盘、网络等资源存在的资源超配或资源配置不足问题，如图 3-18 所示。除此以外，应用智能运维系统能够借助预测分析模型计算提升或降低某种资源配置对关键应用性能指标（如请求响应时间、APDEX 等）的影响程度，从而帮助运维人员找到最优的资源配置方案，在保障应用性能的同时提高资源使用率，节约成本。

企业在规划、构建面向智能、互联时代的应用智能运维系统的过程中，需要摒弃传统以网络、资源、设备为核心的被动运维理念，实现以应用为核心的主动式、智能运维管理平台，实现对应用性能的全方位监控和预测分析。在此过程中，应用智能运维系统能够帮助应用运维人员应对未来的复杂应用系统运维挑战，构建更加简单、高效的智能运维平台，以适应未来数字化驱动的新型互联网企业发展的需要。

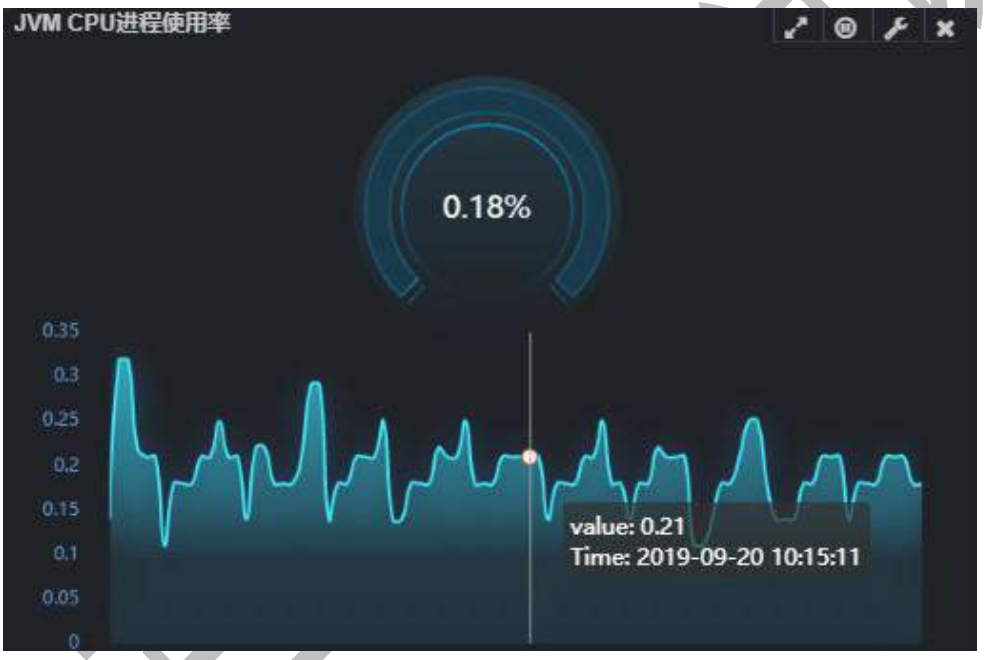


图 3-18　应用资源使用量容量规划截图

本章小结

应用是驱动企业对接“互联网+”、“工业互联网”和“工业 4.0”等国家战略的引擎，是否能有效解决应用性能管理问题关系成败。智能、互联场景下的应用智能运维系统以简单、智能、全可视化的理念重构了企业应用运维流程，相信其在提升企业用户数字体验、大幅度降低运维成本的同时，能为企业带来前所未有的数字化运维新体验。

第 4 章　应用运维智能化的关键技术

本章内容简介：前面介绍了应用智能运维发展演进的历史，回答了应用智能运维是什么、为什么、有什么价值、能干什么的问题。为了指导企业实践、落地，本章围绕应用场景，从技术角度总结归纳了相比于传统的监控运维技术，应用智能运维系统特有的几个关键技术特征，以及介绍了如何用这些技术来解决实际应用运维问题。

智能运维的核心思想是利用算法来处理海量运维数据，积累运维经验，从而代替人工思考判断，以自动化的过程实现风险的预防、发现、定位和处理。在应用运维场景下实现智能化，判断研究用哪些技术来解决实际问题，需要从具体应用场景出发，匹配现有可行的技术。图 4-1 中总结了当前常见的应用智能运维场景，其中包括用于主动发现

图 4-1 常见的应用智能运维场景

潜在风险的应用运行期风险主动探伤、用户数字体验保障与优化、风险定位与根源问题分析，以及应用运行期负载趋势预测与容量规划等。围绕这些场景，我们需要从当前可用的统计、机器学习、人工智能和自治控制技术堆栈中匹配相关的算法。总的来说，当前能够匹配企业应用运维场景、可以用来解决实际运维问题的技术有面向海量实时指标数据的异常检测、发现异常后的关联及根源问题定位、风险处理方案决策支持及预防性维护的探伤检测等。

4.1 异常检测：筛选时间序列数据，发现潜在风险

4.1.1 技术简介

随着互联网和大数据的发展，数据在现实生活中发挥着日益重要的作用。其中，大部分数据都是流式传输的时间序列数据（将同一统计指标的数值按其发生的时间先后顺序排列而成的数列）。针对时间序列数据的数据挖掘已经应用于许多领域，其旨在找到一些频繁出现的模式。当从这些模式中发现某种规律时，异常数据通常被作为噪声而忽略。但是，在庞大的数据量背后，难免会存在异常数据，从数据的异常中往往能够获得更有价值和参考性的信息[1]。快速准确地检测数据中的异常，既能及时减少损失，又方便在短时间内采取适当的应对策略。尤其是在企业应用中，如果能准确地发现系统中出现的异常，对于系统状态的检测及对系统错误的处理将起到积极的推动作用。特别是若能够在异常发生的短时间内检测且报告异常，然后根据以往的异常数据对异常进行分析，推断异常出现的位置及原因，并给予初步的建议解决方案，则将对系统状态稳定起到巨大的作用。

传统的时间序列数据异常检测方法通常聚焦在一维场景下，根据不同时间点数据样本间的关联来对异常进行判断。这个方面的工作经过多年的发展已经相对成熟，其中较为简单的方法包括自适应阈值法、聚类法和指数平滑法等。Smith 等人利用三次指数平滑法实现异常检测，利用历史数据中的不同特征来推测当前的数据值，这在商业领域十

[1] Subutai A, Scott P. Real-Time Anomaly Detection for Streaming Analytics[J]. Computer Science. AI, 2016, 1607:1-9.

分有效[1]；Stanway 等人提出了针对流数据异常检测的 Skyline 项目，其包含一组简单的检测器和一个投票方案，以输出最终的异常评分，该项目在监测高流量网站的实时异常方面卓有成效[2]；Bianco 等人提出的 ARIMA 算法是一种针对具有季节性的时间序列数据建模的通用技术，它对于检测有规律的数据效果较好，但无法动态地确定季节性数据中的异常[3]。另外，在一些特定领域，有许多基于模型的方法已经投入应用，但这些方法往往只针对它们建模的领域，如云数据中心的温度检测[4]、飞机发动机测量中的异常检测[5]和 ATM 欺诈检测[6]等。虽然这些方法在特定的异常检测系统中可能是成功的，但它们无法应用于通用领域。

循环神经网络（Recurrent Neural Network，RNN）等神经网络在时间序列数据异常检测方面具有一定的优势，是对于时间序列数据训练最常见的算法模型之一。然而，由于梯度消失问题的存在，传统的 RNN 在处理存在长期依赖问题的数据时会遇到巨大的困难[7]。近年来，长短期记忆网络（Long Short Term Memory Network，LSTM）由于其在处理时间序列数据方面的优势而受到广泛关注，LSTM 本身的特点使得

---

[1] Simon D L, Rinehart A W. A Model-Based Anomaly Detection Approach for Analyzing Streaming Aircraft Engine Measurement Data[J]. ASME Turbo Expo 2014: Turbine Technical Conference and Exposition, 2014,6(6):32-43.

[2] Hawkins J, Ahmad S, Dubinsky. HTM Cortical Learning Algorithms[R]. Redwood City: Numenta, Incorporation, 2011.

[3] Bao H, Wang Y. A C-SVM Based Anomaly Detection Method for Multi-Dimensional Sequence over Data Stream[C]. IEEE International Conference on Parallel and Distributed Systems,2017.

[4] Telangre K S. Anomaly Detection using multidimensional reduction Principal Component Analysis[J]. IOSR Journal of Computer Engineering,2014,16(1):86-90.

[5] Tan Z, Jamdagni A, He X, et al. Network Intrusion Detection based on LDA for payload feature selection[C]. IEEE GLOBECOM Workshops,2011.

[4] Dau H A, Ciesielski V, Song A. Anomaly Detection Using Replicator Neural Networks Trained on Examples of One Class[J]. Lecture Notes in Computer Science,2014,8886:311-322.

[7] Nanduri A, Sherry L. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)[C]. IEEE Integrated Communications Navigation and Surveillance, 2016.

其极适用于处理时间序列数据，同时 LSTM 克服了 RNN 无法处理长距离依赖的缺点，因此，许多学者提出了基于 LSTM 的异常检测方法。Numenta 公司提出了基于 RNN 的层级实时记忆 HTM 算法，并提出了公开数据集 NAB，在 NAB 数据集上验证了 HTM 算法的性能[1]；Pankaj Malhotra 等人利用基于 LSTM 的异常检测方法在四个不同领域的数据集上取得了极好的效果[2]；Sucheta Chauhan 等人定义了 5 种不同的异常类型，并修改获得了一种 LSTM 变体以对其进行区分[3]。类似的工作还有：Anvardh Nanduri 通过添加 GRU 来改造 LSTM，从而实现了对飞机航班的异常检测[4]；Jihyun Kim 等人实现了一种无监督的异常检测方法，并在真实的工业数据集上进行了验证[5]等。

近年来，随着数据样本量级与维度的迅速增长，多元时间序列数据异常检测的需求日益增加。诸多机构与学者在多元时间序列数据异常检测的研究方面取得了极大进展。Pavel Filonov 等人利用将多元向量合成一元向量的方法处理多元数据，再用常规一元数据异常检测方法进行检测[6]。这种将多维数据转化为一维数据再进行异常检测的方法在维数不多的情况下可行，且通常要求不同维度数据之间具有一定的关联性。类似的工作还

---

[1] Chen Y. Design and Implementation of Network Resource Management and Configuration System based on Container Cloud Platform[C]. International Conference on Frontiers of Manufacturing Science and Measuring Technology,2017:331-335.

[2] Kim J, Kim J, Thu H L T, et al. Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection[C]. IEEE International Conference on Platform Technology and Service,2016.

[3] Filonov P, Lavrentyev A, Vorontsov A. Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model[R]. NIPS Time Series Workshop,2016.

[9] Malhotra P, Vig L, Shroff G, et al. Long short term memory networks for anomaly detection in time series[J]. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning,2015(8):89-94.

[5] Lavin A, Ahmad S. Evaluating Real-Time Anomaly Detection Algorithms—The Numenta Anomaly Benchmark[C]. IEEE 14th International Conference on Machine Learning and Applications, Miami, FL, USA,2015.

[6] Lee E K, Viswanathan H, Pompili D. Model-Based Thermal Anomaly Detection in Cloud Datacenters[C]. IEEE International Conference on Distributed Computing in Sensor Systems,2013.

有 Han Bao 等人通过多维特征序列变换算法、增量特征选择算法以无损方式将时间序列数据转换为特征向量，再基于 C-SVM 的异常检测方法进行异常检测[1]。近年来，业内一些学者提出了一些具有创新性的方法，如 Jones 等人将 8 维数据扩展至 32 维，再结合一维异常检测方法，根据不同维度之间关联性的变化进行多维度的异常检测[2]。该方法既适用于一维时间序列数据，也适用于多维时间序列数据，是一种极具创造性的方法。

4.1.2 深入浅出应用实践

目前，异常检测方法很多。人们对一元时间序列数据的异常检测研究较多，对多元时间序列数据的异常检测研究较少，并多采用降维方法来处理多元时间序列数据。下面介绍几种常用的异常检测方法。

1. 基于曲线拟合的检测算法

对于时间序列数据来说，t 时刻的数值对于 t-1 时刻的数值有很强的依赖性。例如，某个游乐园的人在 8:00 这一时刻很多，在 8:01 时刻其人很多的概率是很大的；但如果其在 7:01 时刻的人较多，这对于其在 8:01 时刻人数的多少影响不是很大。

针对最近时间窗口内的数据遵循某种趋势的现象，可以使用一条曲线对该趋势进行拟合。如果新的数据打破了这种趋势，使曲线变得不平滑，则该点就出现了异常。

---

[1] Chauhan S, Vig L. Anomaly detection in ECG time signals via deep long short-term memory networks[C]. IEEE International Conference on Data Science and Advanced Analytics,2015.

[2] Song Q, Wu Y, Soh Y C. Robust adaptive gradient-descent training algorithm for recurrent neural networks in discrete time domain[J]. IEEE Transactions on Neural Networks, 2008,19(11):1841-1853.

曲线拟合的方法有很多，如回归、滑动平均等。本书用 EWMA，即指数权重移动平均方法来拟合曲线。EWMA 的递推公式：

$$EWMA(1)=p(1) \tag{4-1}$$

$$EWMA(i)=\alpha p(i)+(1-\alpha)EWMA(i-1) \tag{4-2}$$

其中，α是一个 0～1 的小数，称为平滑因子。EWMA(1)有时也会取前若干值的平均值。α越小，EWMA(1)的取值越重要。从式（4-2）可知，下一点的平均值是由上一点的平均值加上当前点的实际值修正而来的。对于每个 EWMA 值，每个数据的权重是不一样的，越近的数据拥有越大的权重。

有了平均值之后，就可以使用 3-σ理论来判断新的输入是否超过了容忍范围。根据实际的值是否超出了容忍范围就可以知道是否可以告警：若超出了上界，可能是流量突然增加了；若低于下界，可能是流量突然降低了，这两种情况都需要告警。可以使用 Pandas 库中的 ewma 函数来实现上面的计算过程。

EWMA 的优点如下。

（1）其可以检测到在一个异常发生较短时间后发生的另一个（不太高的突变型）异常。

（2）因为它更多地参考了突变之前的点，所以它能更快地对异常做出反应。

（3）其非常敏感，历史数据如果波动很小，那么方差就很小，容忍的波动范围也会非常小。

EWMA 的缺点如下。

（1）其对渐进型（而非突发型）的异常检测能力较弱。

（2）异常持续一段时间后可能被判定为正常。

（3）其业务曲线自身可能有规律性的陡增和陡降。

（4）其过于敏感，容易误报，因为方差会随着异常点的引入而变大，所以很难使用连续三点才告警这样的策略。

考虑到这些缺点，需要引入周期性的检测算法来针对性地处理具有周期性趋势的曲线。

2. 基于同期数据的检测算法

很多监控项都具有一定的周期性，其中以一天为周期的情况比较常见，如淘宝 VIP 流量在早晨 4 点最低，而在晚上 11 点最高。为了将监控项的周期性考虑进去，可以选取某个监控项过去 14 天的数据。对于某个时刻，将得到的 14 个点作为参考值，记为 x_i，其中 $i=1,2,\cdots,14$。

用静态阈值方法来判断输入是否异常（突增和突减）。如果输入比过去 14 天同一时刻的最小值乘以一个阈值还小，那么就认为该输入为异常点（突减）；如果输入比过去 14 天同一时刻的最大值乘以一个阈值还大，那么也认为该输入为异常点（突增）。

静态阈值方法中的阈值是根据历史经验得出的值，实际中如何给出 $\max_{\text{th}}$（最大阈值）和 $\min_{\text{th}}$（最小阈值）是一个需要讨论的问题。根据目前静态阈值方法的经验规则，

取平均值是一个比较好的思路。

静态阈值方法的优点如下。

（1）其反映了周期性。

（2）其可以确保发现大的故障，给出告警的一定是大问题。

静态阈值方法的缺点如下。

（1）其依赖周期性的历史数据，计算量大，而且无法对新接入的曲线告警。

（2）其非常不敏感，无法发现小波动。

3. 基于同期振幅的检测算法

基于同期数据的检测算法遇到如图 4-2 所示的现象就无法检测出异常。例如，今天是 11 月 11 日，过去 14 天淘宝 VIP 流量的历史曲线必然会比今天的曲线低很多，如果 11 月 11 日这天出了一个小故障，曲线下跌了，但相对于过去 14 天的曲线仍然是高很多的，这样的故障使用基于同期数据的检测算法就检测不出来，那么将如何改进呢？直观来看，两个曲线虽然不一样高，但“长得差不多”，那么，怎么利用这种“长得差不多”呢？此时就可以采用基于同期振幅的检测算法。

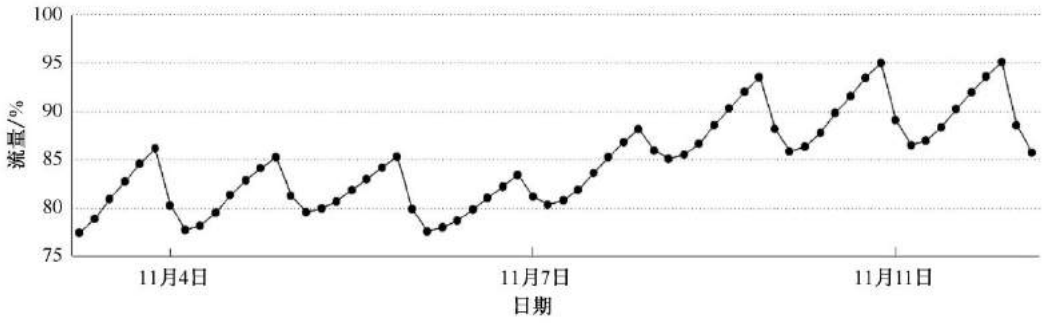


图 4-2　淘宝 VIP 流量示意

怎么计算 t 时刻的振幅呢？可以使用 $\frac{x_t - x_{t-1}}{x_{t-1}}$ 来表示振幅。例如，t 时刻有 900 人在线，t-1 时刻有 1000 人在线，那么，可以计算出掉线人数是 100。如果参考过去 14 天的数据，那么可得到 14 个振幅值。使用 14 个振幅的绝对值作为标准，如果 m 时刻的振幅 $\left(\frac{x_m - x_{m-1}}{x_{m-1}}\right)$ 大于振幅阈值 a_{th} 且 m 时刻的振幅大于 0，那么认为该时刻发生了突增；如果 m 时刻的振幅大于 a_{th} 且 m 时刻的振幅小于 0，那么认为该时刻发生了突减。

$$a_{\text{th}} = \max\left[\left|\frac{x_i(t) - x_i(t-1)}{x_i(t-1)}\right|\right],\ i = 1, 2, \cdots, n \tag{4-3}$$

基于同期振幅的检测算法的优点如下。

（1）振幅要比绝对值敏感。

（2）其利用了时间周期性，规避了业务曲线自身的周期性陡降。

基于同期振幅的检测算法的缺点如下。

（1）其要求原曲线是平滑的。

（2）周期性陡降的时间点必须重合，否则会发生误警。

（3）按百分比计算容易在低峰时期发生误警。

（4）陡降不一定代表故障，由上层服务波动引起的冲高再回落的情况时有发生。

4. 基于环比数据的检测算法

对于时间序列数据，可以利用最近时间窗口（T）内的数据遵循某种趋势的现象来进行检测。如将 T 设置为 7，取检测值（$\text{now}_{\text{value}}$）和过去 7 个点的值（记为 i）进行比较，如果结果大于阈值，将 count 加 1，若 count 超过了设置的 $\text{count}_{\text{num}}$，则认为该点是异常点。

$$\text{count}\left(\sum_{i=0}^{T}\text{Integer}(\left|(\text{now}_{\text{value}}-i)\right|)>\text{threshold}\right)>\text{count}_{\text{num}} \tag{4-4}$$

式（4-4）涉及 threshold（动态阈值）和 $\text{count}_{\text{num}}$ 两个参数，$\text{count}_{\text{num}}$ 可以根据需求进行设置，如果对异常比较敏感，可以将 $\text{count}_{\text{num}}$ 设置得小一些；如果对异常不敏感，可以将 $\text{count}_{\text{num}}$ 设置得大一些。业界关于 threshold 设置的方法有很多，下面介绍一种比较常用的阈值设置方法：通常阈值设置方法会参考过去一段时间内的均值、最大值及最小值，取过去一段时间（如窗口 T）的平均值（avg）、最大值（max）及最小值（min），然后取 max-avg 和 avg-min 的最小值作为阈值［见式（4-5）］。之所以取最小值，是要让筛选条件设置得宽松一些，让更多的值通过此条件，从而减少漏报。

$$\text{threshold}=\min(\text{max}-\text{avg},\ \text{avg}-\text{min}) \tag{4-5}$$

5. 基于 Ensemble 的检测算法

iForest 算法是南京大学的周志华于 2010 年设计的一种异常检测算法，该算法利用数据构建 iTree，进而构建 iForest，是一种无监督的检测算法，具有很好的效果，具体可参见 http://www.cnblogs.com/fengfenggirl/p/iForest.html。

iForest 是由 iTree 构建而成的。iTree 是一种随机二叉树，其每个节点要么有两个子节点，要么为叶子节点。对于给定的数据集 D，数据集中的所有的特征都是连续变量，iTree 的构造如下。

（1）在数据集 D 中随机选择一个特征 A。

（2）随机选择特征 A 的一个可能取值 v。

（3）根据特征 A 及值 v 将数据集 D 分为两个子集，将特征 A 的值小于 v 的样本归入左子节点，余下部分归入右子节点。

（4）递归构造左、右子树，直至满足以下的终止条件：

① 传入的数据集只有一条记录或多条相同的记录；

② 树的高度达到了限定高度。

iTree 建好以后，就可以对数据进行预测了，预测的过程就是将测试记录在 iTree 上走一遍。iTree 能有效地检测异常点是基于异常点都很稀有这一假设的，异常点应该在 iTree 中很快被划分到叶子节点，因此，可以利用检测点被分入的叶子节点到根的路径长度 h(x)来判断检测点 x 是否为异常点。

在构建好 iTree 后，就可以构建 iForest。在构造 iForest 中的每棵树时，并不是要将所有的数据都用上，而是随机采样，抽取一部分构造 iTree，并尽量保证每棵树都不相同。事实上，如果 iTree 在构造时运用了很多数据点，反而不能得到很好的效果，这主要是因为数据点会有重叠。因为由 iTree 变成了 iForest，所以 S(x,n)的计算公式也要改变，将 h(x)变为 E[h(x)]，它就是检测点 x 在每棵树上的平均高度。iForest 算法在 Python 中有现成的包可以调用。

利用 iForest 算法进行判断时，如果检测点的孤立森林分数为正数，那么，检测点为正常点；否则，检测点为异常点。

6. 基于神经网络的检测算法

人工神经网络（Artificial Neural Networks，ANN）是 20 世纪 40 年代后出现的。它是由众多的神经元可调的连接权值连接而成的，具有大规模并行处理、分布式信息存储、良好的自组织和自学习能力等特点。BP（Back Propagation）算法又称为误差反向传播算法，是人工神经网络中的一种监督式的学习算法。BP 算法在理论上可以逼近任意函数，其基本的结构由非线性变化单元组成，具有很强的非线性映射能力，而且其网络的中间层数、各层的处理单元数及网络的学习系数等参数可根据具体情况设定，灵活性很大，在优化、信号处理与模式识别、智能控制、故障诊断等许多领域都有广阔的应用前景。

当前用于异常检测的基于神经网络的检测算法有很多，其中比较常见的是卷积神经网络（CNN）算法、循环神经网络（RNN）算法、深度神经网络（DNN）算法等，下面介绍一种称为长短期记忆网络（LSTM）的算法。

LSTM 是一种改进后的 RNN，可以解决 RNN 无法处理长距离依赖的问题，目前比较流行。其思路：原始 RNN 的隐藏层只有一个状态，即 h，它对于短期的输入非常敏感，现在再增加一个状态，即 c，让它来保存长期的状态，称它为单元状态（Cell State），如图 4-3 所示。

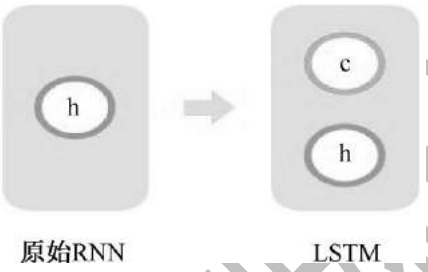


图 4-3　从 RNN 到 LSTM

把图 4-3 按照时间维度展开，如图 4-4 所示，在 t 时刻，LSTM 的输入有三个：当前时刻网络的输入值 $\boldsymbol{x}_t$、上一时刻 LSTM 的输出值 $\boldsymbol{h}_{t-1}$，以及上一时刻的单元状态 $\boldsymbol{c}_{t-1}$。LSTM 的输出有两个：当前时刻 LSTM 的输出值 $\boldsymbol{h}_t$ 和当前时刻的单元状态 $\boldsymbol{c}_t$。

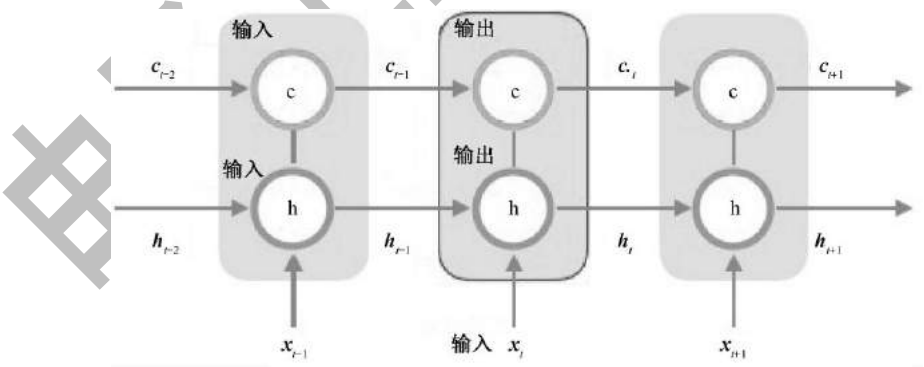


图 4-4　LSTM 示意

为了控制单元状态 c，LSTM 使用了三个“门”作为开关，如图 4-5 所示。

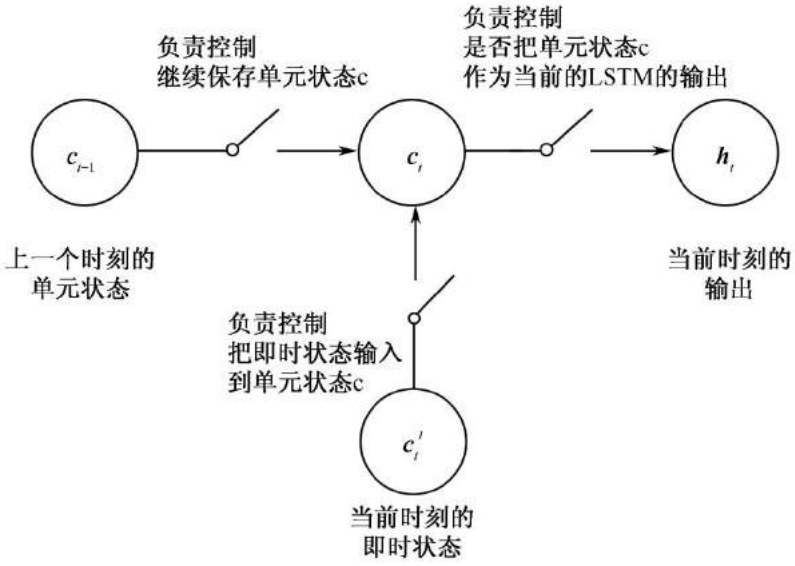


图 4-5 LSTM 的“门”开关

遗忘门（Forget Gate）：负责控制继续保存单元状态 c，它决定了上一时刻的单元状态 $\boldsymbol{c}_{t-1}$ 有多少保留到当前时刻的单元状态 $\boldsymbol{c}_t$。

输入门（Input Gate）：负责控制把即时状态输入到单元状态 c，它决定了当前时刻网络的输入 $\boldsymbol{x}_t$ 有多少保存到单元状态 $\boldsymbol{c}_t$。

输出门（Output Gate）：负责控制是否把单元状态 c 作为当前的 LSTM 的输出，它决定了单元状态 $\boldsymbol{c}_t$ 有多少输出到 LSTM 的当前输出值 $\boldsymbol{h}_t$。

遗忘门的计算如式（4-6）所示。

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \cdot \left[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t\right] + b_f\right) \tag{4-6}$$

其中，$\boldsymbol{W}_f$ 是遗忘门的权重矩阵；$[\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]$ 表示把两个向量连接成一个更长的向量；b_f 是遗忘门的偏置项；σ 是 sigmoid 函数。

输入门和一个 tanh 函数配合控制该加入哪些新信息。tanh 函数产生一个新的候选向量 $\tilde{\boldsymbol{C}}_t$，输入门为 $\tilde{\boldsymbol{C}}_t$ 中的每项产生一个 0～1 的值，用于控制新信息被加入的多少。至此，已经有了遗忘门的输出 $\boldsymbol{f}_t$（用来控制上一单元被遗忘的程度）和输入门的输出 $\boldsymbol{i}_t$（用来控制新信息被加入的多少），此时就可以更新本记忆单元的单元状态了：

$$\boldsymbol{C}_t = \boldsymbol{f}_t * \boldsymbol{C}_{t-1} + \boldsymbol{i}_t * \tilde{\boldsymbol{C}}_t \tag{4-7}$$

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + b_i\right) \tag{4-8}$$

$$\tilde{\boldsymbol{C}}_t = \tanh\left(\boldsymbol{W}_C \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + b_C\right) \tag{4-9}$$

输出门用来控制当前的单元状态有多少被过滤掉。先将单元状态激活，输出门为其中每项产生一个 0～1 的值，用来控制单元状态被过滤的程度。

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + b_o\right) \tag{4-10}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t * \tanh(\boldsymbol{C}_t) \tag{4-11}$$

上面描述的 LSTM 是一个标准版本，并不是所有 LSTM 都和上面描述的一模一样。事实上，每个人所使用的 LSTM 都有一些细微的不同，有人专门比较总结过 LSTM 的各种变体，并比较了其效果，结果显示，这些变体在多数公开数据集上的表现差异不大。[1]

上面介绍了六种检测算法，每种算法都有其优缺点，都有能检测和不能检测的范围。

[1] Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures[C]. International Conference on Machine Learning, 2015.

在应用时，应根据实际情况来选择具体的算法，也可以使用多种算法进行综合检测，更多的检测算法可以参考开源项目 Skyline 中的算法库。

4.1.3 应用案例

异常检测的应用场景极为广泛，其中针对时间序列数据的异常检测在工业、金融、军事、医疗、保险、关键系统安全、机器人、多智能体、网络安全和物联网等多个领域具有极其重要的地位与意义[1]。斯坦福大学的 Melvin Gauci 等人将 1000 个智能体组成系统，通过模拟实验证明不加限制的单个短时异常会在群体内快速传播，最终导致系统的崩溃[2]，从而说明异常检测效果是诸多场景安全交互的核心。鉴于异常检测在实际应用中的重要意义，开发性能更优、速度更快、检测更精准的异常检测算法急迫且意义重大。

1. 面向大数据应用的异常检测

随着计算机和互联网信息技术的迅猛发展与普及应用，各行各业的数据均呈现爆炸式增长，巨大的数据资源被很多国家和企业视为战略资源，大数据已经成为目前互联网领域的研究热点之一，这也标志着全球已经进入大数据时代[3]。

数据挖掘是从海量异构的数据中挖掘出未知的、潜在的信息和知识的过程。数据规模大、数据多样性是大数据的基本特点。海量复杂的数据中可能存在一些数据对象，这些数据对象与普通数据的期望行为模式并不一致，被称为异常值或离群点。随着数据规

[1] Subutai A, Scott P. Real-Time Anomaly Detection for Streaming Analytics[J]. Computer Science AI,2016,1607:1-9.

[2] Szmit M, Szmit A. Usage of Modified Holt-Winters Method in the Anomaly Detection of Network Traffic: Case Studies[J]. Journal of Computer Networks and Communications,2012(8):1-5.

[3] 王玉杰. 面向大数据应用的情境感知异常检测算法研究[D]. 兰州：兰州大学,2018.

模和数据多样性的不断增加，数据中异常值（或离群点）的个数也会不断增加，合理有效地处理和应用这些异常值对大数据挖掘具有重要的意义[1]。针对大数据中异常值的识别和挖掘称为异常检测。虽然数据中的异常值是不寻常的，但如果考虑的数据量多达数十亿，则可能性为“千分之一”的异常值也可能是百万量级，在大数据挖掘过程中，这些异常值是不能忽视的。因此，异常值检测在大数据挖掘中有着至关重要的作用[2]。

2. 面向车联网应用的异常检测

近年来，随着信息化时代的到来及社会经济的高速发展，人们对交通的需求日渐增长，致使车辆运输效率不断下降，能源消耗持续高涨，运输环境日益恶化，交通拥堵越发严重，交通事故愈发频繁，这些成为我国许多城市的普遍性问题。因此，智能交通系统（Intelligent Traffic System，ITS）应社会对交通发展的需求而产生。车联网（Internet of Vehicles，IoV）作为物联网（Internet of Things，IoT）在智能交通系统中的一个主要组成部分，其发展对于智能交通系统的发展具有推动作用。车联网将目前的新一代信息技术，如移动互联网、人工智能、物联网等相互融合，给传统汽车生产商带来了全新的变革，智能化和网络化已成为全球汽车与交通领域发展的主流趋势。预计 2020 年，全球将有超过 500 亿个智能设备接入物联网中，其中很大一部分便是车联网设备。车辆传感器的联网率将由现在的 10%增加到 90%，中国将有超过 35%的汽车实现网络互联[3]。

然而，由于车联网的特殊性，即开放的无线传输介质、车辆节点的高速移动性、网

[1] Hu Y, Peng Q, Hu X. A time-aware and data sparsity tolerant approach for web service recommendation[C]. 2014 IEEE International Conference on Web Services. IEEE, 2014.

[2] 孙大为，张广艳，郑纬民. 大数据流式计算：关键技术及系统实例[J]. 软件学报，2014, 25(4): 839-862.

[3] 张倩. 车联网异常检测及数据恢复技术研究[D]. 西安：西安电子科技大学，2018.

络拓扑结构的频繁变化、易受环境影响及人为的信息干扰，使得传感器或传输线路可能出现故障，从而引起数据被篡改、失真或丢失。如果一个突发交通事故的数据在传送过程中混入了其他虚假杂乱的数据，那么可能会造成交通堵塞，更有甚者会威胁司机的生命安全。此外，异常数据的存在会影响数据分析的完整性和准确性[1]。2016 年，腾讯科恩实验室通过车辆之间的无线连接和蜂窝连接漏洞两次成功破解了特斯拉 MODEL S，其向汽车网络中发送恶意软件并将破解程序渗透到 CAN 总线，从而获得了刹车系统的远程操控权。这些远程控制车辆的案例说明外设人员篡改车辆数据成为可能，进而导致车联网中数据的安全性和可靠性受到严重的威胁。

因此，对车联网实时数据的异常检测及恢复迫在眉睫，它可以有效地提高数据质量，确保交通分析模型的准确性和智能交通系统的实用性，进而有效协助司机做出适当的驾车行为，合理调度交通资源，实时监测车辆故障并在必要时发出警告，避免发生交通事故，对交通安全、环境保护及人员健康都有着极其重要的作用和意义。

3. 面向工业应用的异常检测

近几年，传统工业控制系统和互联网、云平台逐渐连接起来，构成了工业互联网平台。工业互联网平台将现场设备、生产物料、网络系统连接成一个整体的系统，实现了工业数据的动态采集和实时分析，用智能控制替代了原来的人为操作，提高了工厂生产效率，是工业生产布局的新方向。工业互联网平台集海量数据采集和分析于一体，能够精准高效地对数据进行实时处理，推进了制造业发展的新征程[2]。

---

[1] Zheng Y, Rajasegarar S, Leckie C, et al. Smart car parking: Temporal clustering and anomaly detection in urban car parking[C]. IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, 2014.

[2] 龚晓菲. 工业互联网平台数据的异常检测研究[D]. 北京：北京邮电大学,2019.

工业互联网在给工业控制系统带来便捷操作的同时，也引入了一系列的安全问题，各种入侵、攻击手段层出不穷，建设满足工业需求的安全体系是保障工业互联网平台正常运行的前提。各种网络入侵技术的发展已对工业互联网平台造成了严重的威胁，工业控制系统的现场设备、控制系统及网络设备都存在被攻击的风险，一旦这些设备出现异常，将会给工业带来不可估量的经济损失，影响生产进度，甚至危害人员的生命安全[1]。2010 年，在著名的“震网”病毒事件中，攻击者利用 4 个“0day”漏洞，致使伊朗核设施的离心机出现了故障，震惊了全球。因此，应对各种网络攻击已经成为保障国家关键基础设施安全的基本需求。

[1] Jairo G, David U, Alvaro C, et al. A Survey of Physics-Based Attack Detection in Cyber-Physical Systems[J].ACM Computing Surveys, 2018, 51(4):1-36.

4.2 关联分析：实现全景化应用监控的基础

4.2.1 技术简介

应用运维智能化技术和相关软件系统是伴随应用系统复杂度、运维工作量和技术难度激增而出现的，因此，通过人工智能算法来代替人工融合和分析数据、推理、决策、处理问题是建设应用智能运维系统需要考虑的关键问题之一。

传统应用运维过程中常用的监控运维系统一般是针对特定场景、特定资源建设的。例如，日志分析平台采集分析应用日志；APM 监控代码链路和对全量用户请求的处理情况；网络性能管理（NPM）平台追踪网络中的交易情况和网络异常；IT 资源监控系统监控服务器、网络设备、云环境和应用运行依赖的中间件等。要做到智能化，首先要有运维数据治理平台的支撑，将离散、竖井式的监控系统关联打通，构建同构的、一致的全景化应用监控视图，这样才能为运维人员过滤冗余信息，提供精准的风险态势监控和定位决策支持。

4.2.2 深入浅出应用实践

关联分析是整合应用运行期生成的各层级全栈数据、关联打通竖井式监控系统的关键。目前可以用来关联应用运维数据的方法主要有如下几种。

（1）读取配置管理数据库（CMDB）信息。CMDB 是一个数据库，其中包含有关组织 IT 服务中使用的硬件和软件组件，以及这些组件之间关系的所有相关信息。信息系统的组件称为配置项（CI）。CI 可以是任何可以想象的 IT 组件，包括软件、硬件、文档和人员，以及它们之间的任意组合或依赖关系。应用运行期依赖物理 IT 基础设施设备、虚拟 IT 基础设施设备与应用之间的部署关系，网络拓扑关联关系可以从 CMDB 中定义的 CI 关联读取出来。一旦设备出现故障，这些关系可以用来辅助找出影响范围。

（2）监控分析网络流量。NPM 工具可以通过旁路镜像网络流量来监控网络上应用中的服务接口之间、应用与用户之间的交互关系，获取网络层的关联关系。利用深度网络包检测（Deep Packet Inspection，DPI）技术，甚至可以将网络报文中的业务交互信息解析出来，补充业务层的调用关系。

（3）追踪应用代码链路。APM 工具提供了对应用程序性能深入分析的能力，当用户向应用程序发出请求时，APM 工具可以通过探针看到分布式部署的应用系统中的接口调用关系、代码链路执行过程和方法调用关系，并且可以显示有关此请求发生的系统数据、参数和与数据库交互的 SQL 语句。应用白盒监控能力提供的关联关系，可以在排查代码缺陷导致的故障时，快速定位根源问题。

（4）利用人工智能算法计算关联关系。以上三种方法利用传统运维监控工具提供的数据关联和检索能力构建了覆盖物理部署、网络交互、接口交互与代码交互的关系图结构（见图 4-6），基于此视图可以实现在异常情况下的信息关联。但是，一旦出现未能直接监控的问题导致的应用故障，就需要用算法来辅助分析海量历史监控数据，发现数据中隐含的关系，并根据发现的问题及已知事件推理进行决策。常用的技术是查找时间序列指标数据波动之间的相似性、相关性等关联关系（主要方法有 Pearson、

Granger Kendall、Spearman 等）。基于关联关系构建的因果推理分析模型，可以基于概率图模型（如 Bayesian Networks、Markov Random Fields 等）建模来实现因果关系发现和推理。

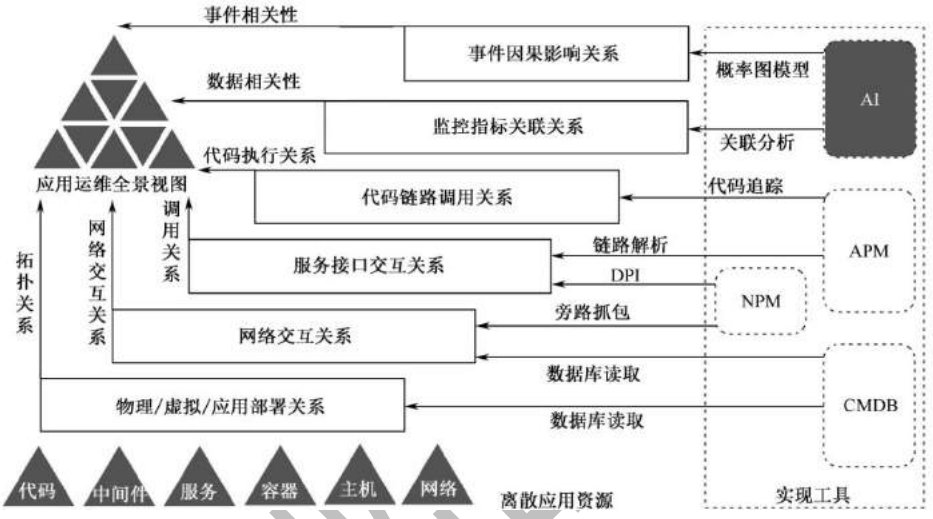


图 4-6 应用全景监控数据关联关系建模的策略

相关性是一种双变量分析，用于测量两个变量之间的关联强度和关系方向。就关联强度而言，相关系数的值在 +1 和 -1 之间变化，其值为 1 表示两个变量之间完全关联；值为 0 表示两个变量之间的关联较弱。关系方向由相关系数的符号指示：“+”表示正关系；“-”表示负关系。通常，利用统计学方法可以计算以下几种相关性：皮尔森相关性（Pearson Correlation）、斯皮尔曼相关性（Spearman Correlation）和格兰杰因果关系检验（Granger Causality）。

皮尔森相关性是使用最广泛的相关统计，用于测量持续变化的变量之间的线性相关程度。例如，在股票市场中，如果想要测量两只股票之间的关系，那么就可以使用皮尔森相关性。

斯皮尔曼相关性评估两个连续变量之间的单调关系。在单调关系中，变量往往一起变化，但不一定以恒定速率变化。斯皮尔曼相关系数是基于每个变量的排名值而不是原始数据的。斯皮尔曼相关性通常用于评估正数变量的关系。

格兰杰因果关系检验是经典方法，在计量经济学的时间序列分析中有较多的应用。除此之外，还有收敛交叉映射（Convergent Cross Mapping，CCM）方法。格兰杰因果模型的前提假设是事件是完全随机的，但现实情况是很多事件是非线性、动态且非随机的，格兰杰因果模型对这类情况不适用。CCM 方法则适用于这一类场景，其可在多组时间序列中构建因果网络。

4.3 数据统计：敏捷高效的信息提取手段

4.3.1 技术简介

虽然人工智能算法具有识别复杂模式、可替代人脑进行推理分析等优势，但目前由于缺少通用的人工智能平台，其计算复杂度和实施成本相对较高，在某些运维场景下并不适用，而某些统计学方法简单高效，与人工智能算法结合的效果很有可能出人意料。

Google 前 SRE 工程师 Tom Limoncelli 在编著的 The Practice of Cloud System Administration: Designing and Operating Large Distributed Systems 一书[1]中讲过一个故事：“当有人问我建议平时都要监控什么时，我会开玩笑地跟他们说，理想情况下，我会首先要求他们删掉监控系统里的所有数据采集和告警策略，当再次发生故障时，想想什么指标可以预测这次故障的发生，然后把这个指标监控和告警策略加回到监控系统中来。如此不断积累，现在监控系统中只存在能够预测各种不同故障的指标和告警，从而当故障发生时，监控系统不会被大量告警信息淹没。”

还有一种更加简便的方法也能达到预期效果，但要有完整的历史数据支撑，即查看历史 30 天或半年的故障恢复记录，看哪些指标对发现并解决特定问题和特定故障有用。例如，如果我们发现一台 Web 服务器停止响应了，首先要查找的是发现这一现象的相关指标数据，而且这些指标未必是从 Web 服务器本身采集的，例如：

[1] Thomas L, Strata R C, Christina J H. The Practice of Cloud System Administration: Designing and Operating Large Distributed Systems[M].New York: Addison-Wesley,2014.

（1）应用层：Web 页面加载时间持续增加；

（2）操作系统层：服务器内存使用率无波动，磁盘读写无波动；

（3）数据库层：数据库事务执行时间持续升高；

（4）网络层：负载均衡器挂载的活跃计算节点数量降低。

以上这些指标异常都有可能提前反映某些潜在的应用故障。对于之前发生过的每种故障类型，找到其对应的能提前反映异常的指标，定义告警策略。如果日常运维持续这个过程，积累经验数据，那么能提前发现的异常类型就会越来越多，由于应用故障导致直接影响用户的情况就会越来越少，运维体系的工作负荷就会越来越低。

使用平均值和标准差检测呈高斯分布的指标异常是行之有效的方法。但是，对于其他非高斯分布的指标，有可能达不到预期效果，一旦指标数据的概率分布不符合高斯钟形曲线，基于平均值和标准差来过滤异常数据的手段就不适用。例如，要监控某网站上每分钟下载特定文件次数这一用户行为，可定位下载次数异常增高的时间窗口，即过滤大于平均值 3 倍标准差下载量的时间段。在图 4-7 中，灰色柱形图展示了用户每分钟下载量的时间序列分布，上方黑色滑动窗口序列对应标识了下载量大于平均值 3 倍标准差的时间段，灰色未标识窗口对应的下载量则小于平均值 3 倍标准差。

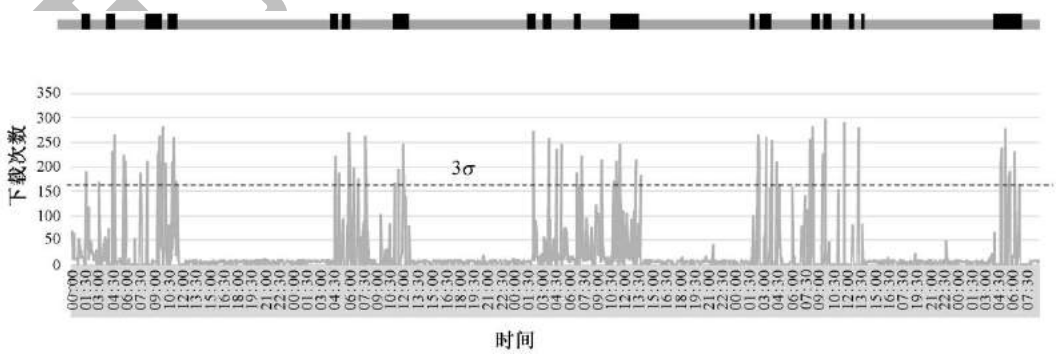


图 4-7　每分钟下载量指标使用平均值 3 倍标准差规则时过度警报的效果截图

从图中可以看出，如果用大于平均值 3 倍标准差策略生成告警，很明显的一个问题是，大多数时间段都需要产生告警。为了更明显地展示此问题，我们将该指标数据的概率分布可视化，如图 4-8 所示。图 4-8 中，横轴是指标可能出现的数值，纵轴是一段时间内该值出现的次数统计值。很明显，其并不是对称的高斯钟形曲线。通常情况下，文件下载频率都比较低，但高于平均值 3 倍标准差的下载任务，在时间轴上的分布规律性相对较强。

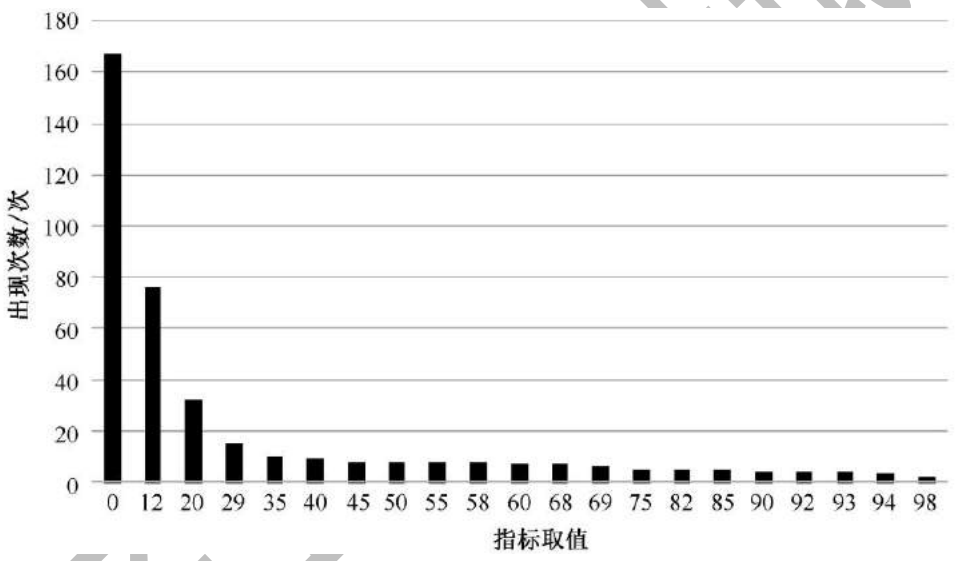


图 4-8　每分钟下载量指标数据直方图

像这种非高斯分布的指标，在生产环境中并不是少数。对这种现象，Simple Math for Anomaly Detection[30]一书的作者 Toufic Boubez 博士认为，“在运维过程中，我们采集的很多指标数据满足‘卡方（Chi Squared）’分布的概率分布。在这类指标上使用平均值 3 倍标准差做异常检测和告警，会导致告警风暴或干脆检测不出来”。她认为，“如果过滤小于平均值 3 倍标准差的数据，我们将得到负数，得到的结果很明显也没有什么意义” 。

告警风暴是运维人员不愿意遇到的情况，一些故障有时并不严重，或者根本没有必要深夜起床处理。而若出现风险迹象或已经发生故障未检测出来，后果则更为严重。假如我们要监控已经完成的用户提交事务指标，由于系统软件出现故障，该指标陡降 50%，如果我们使用平均值 3 倍标准差的统计学方法检测异常，则监控指标值在正常范围内，不会产生告警。后果就是，用户将先发现此问题，接近 50%的用户提交事务将返回执行失败的提示，该问题造成的损失会更大。我们需要用其他方法来发现这类问题。

案例

Scryer 是 Netflix 开发的用来解决 Amazon AWS 云平台 Amazon Auto Scaling（AAS）功能缺陷、提升应用服务质量的工具。AAS 可以探测 AWS 云上的应用负载，自动增加或减少应用云上弹性集群的计算能力。Netflix 开发的 Scryer 在 AAS 功能的基础上，可以通过分析历史数据的趋势和规律对应用未来的负载进行预测，预先弹性控制集群规模，分配或回收资源。总的来说，Scryer 弥补了 AAS 以下三个不足。一是 AAS 对用户并发访问量突发峰值处理方面的不足。由于并发量突然增加，持续时间较短，而 AAS 处理采用弹性控制策略，创建、启动新 AWS EC2 计算节点的速度要持续几分钟甚至几十分钟时间。等集群节点创建完毕，也错过了并发访问量激增的时段。二是由于 AAS 判断策略简单，用户访问量的突然减少会使 AAS 消减过多的集群节点，以至于其不足以处理即将发生的用户访问。三是 AAS 不能从历史用户访问数据中找到趋势和规律来指导未来的容量规划。

Netflix 用户访问数据的概率分布并不符合高斯分布，但数据规律性较明显，每天分时段访问量、节假日和工作日的访问量都有明显的规律可循，因此可预测性较强。可通过使用多种异常检测策略监测突发的访问量激增，结合快速傅里叶变换（Fast FourierTransform，FFT）、线性回归平滑处理数据，同时保留合理的有规律激增点。通

过这些处理技术，Netflix 能够在处理并发访问量激增时预测趋势，获得一些提前量来增加资源（见图 4-9），从而保障用户体验流畅。在 Scryer 系统上线第一个月，Netflix 就监测到了明显的用户体验和服务质量的提升，AWS EC2 计算资源的使用成本也有所降低。

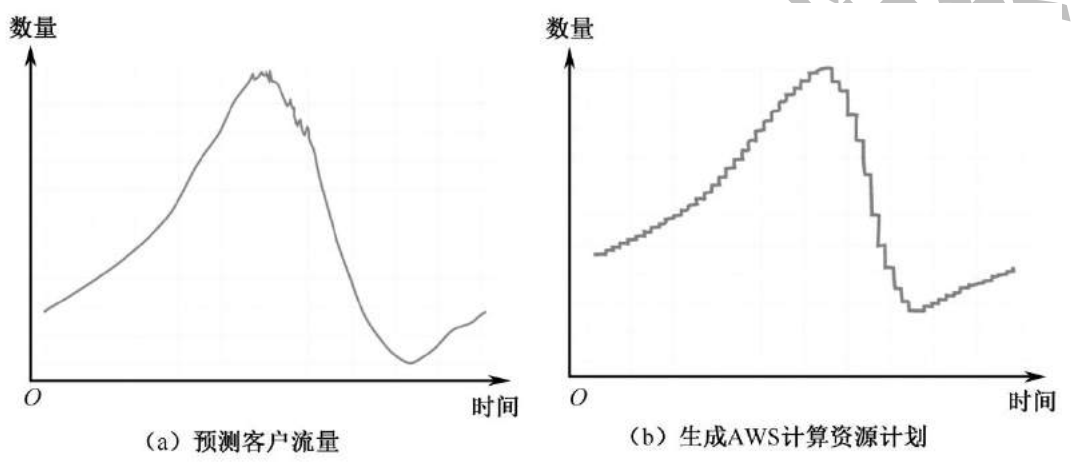


图 4-9 Netflix Scryer 预测用户流量和生成的 AWS 计算资源计划

对于规律性不是很明显的非高斯分布的时间序列指标数据的异常检测，常采用指定时间窗口平滑处理，即选定一个时间窗口，沿时间轴滑动，将每个点的监控值替换为时间窗口内所有点的平均值。这么处理可以将指标曲线波动剧烈的锯齿状波形平滑掉，突出曲线趋势和规律。图 4-10 所示为原始曲线和经过平滑处理的曲线的对比，灰色为原始数据曲线，黑色为经过 30 天时间窗口平滑处理后的曲线。

除了平滑处理，类似常用的处理方法还有 KS 检验（Kolmogorov-Smirnov Test，用于检测数据是否符合指定分布）、快速傅里叶变换（Fast Fourier Transforms）等。大多数和用户触发任务执行相关的指标都是存在规律性的，通过学习历史数据中每天、每周、每年的规律，就能够发现实时数据是否异常。例如，若周六晚上的用户成功交易

量相比于历史同期下降了 50%，则有可能存在系统异常，导致用户请求执行缓慢或失败。

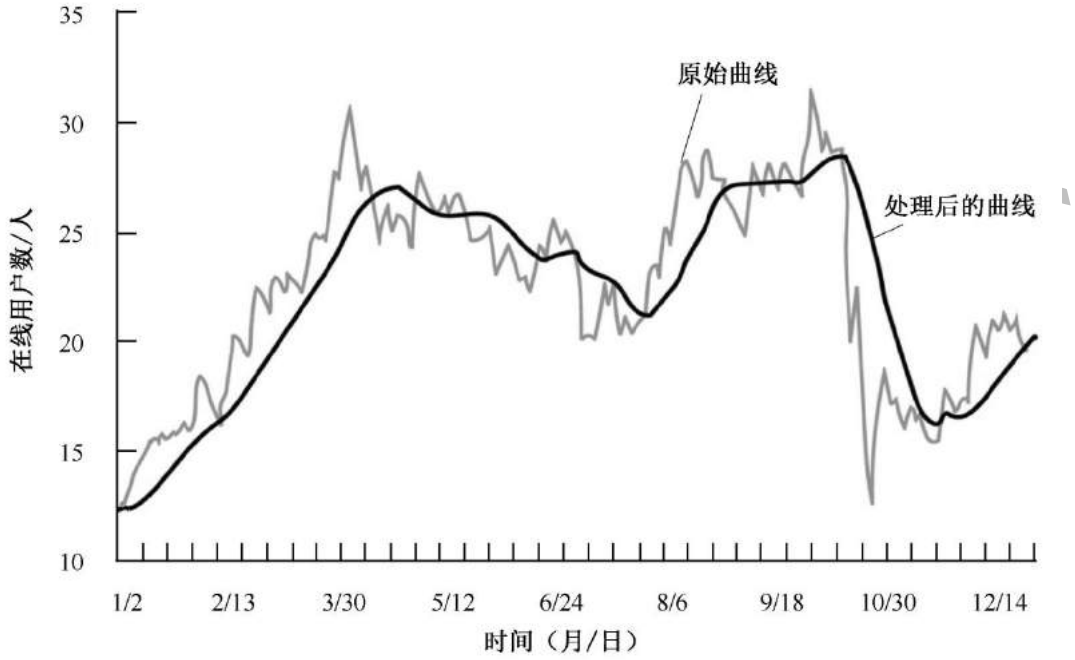


图 4-10　原始曲线和经过平滑处理的曲线对比

4.3.2 深入浅出应用实践

2014 年，在 Monitorama 公司，Toufic Boubez 博士介绍了使用 KS 检验方法实现异常检测的案例[1]。KS 检验方法在统计学中通常用于检验两个数据集的相似性，使用这种方法的常用运维监控工具有侧重数据采集与存储的 Graphite 和侧重分析展现的 Grafana。

图 4-11 所示为某电子商务网站的交易笔数指标的月交易量时间分布。从指标曲线变

[1] Toufic B. Simple math for anomaly detection[M]. Portland: Monitorama PDX,2014.

化趋势能直观看出，箭头所指处的交易量异常，并没有往常那么多。

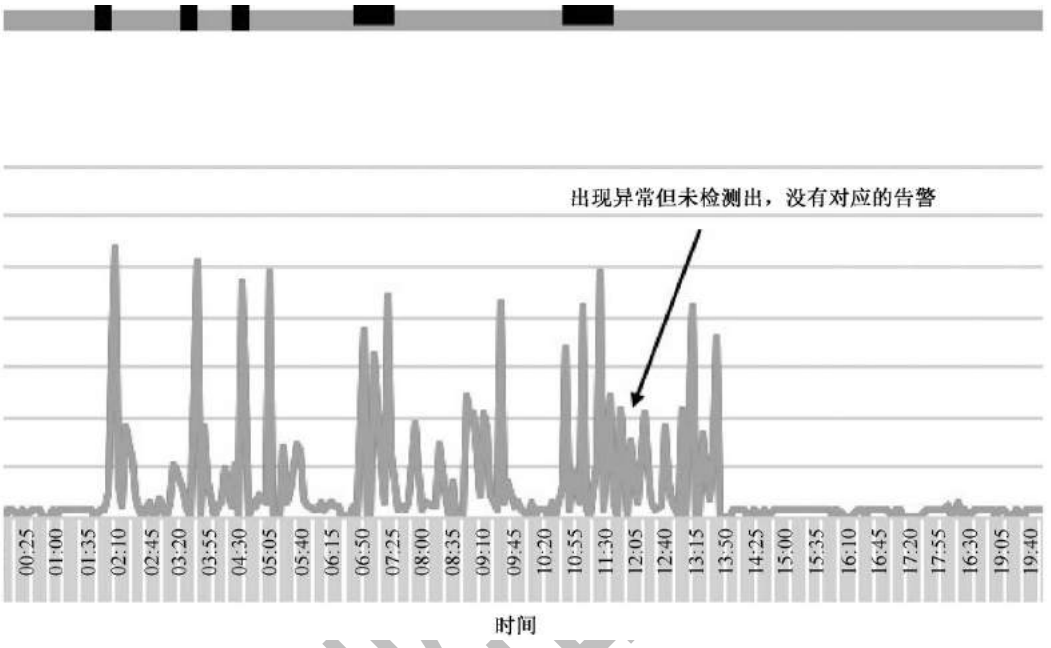


图 4-11　某电子商务网站的交易笔数指标的月交易量时间分布

如果使用平均值 3 倍标准差规则判断异常，将收到 2 次告警，周一交易量未恢复正常值的异常会被忽略掉。在理想情况下，若交易量与历史同期的平均值相差太大，我们也希望收到告警通知。Boubez 博士在 Simple Math for Anomaly Detection 一书中提到，“KS 检验方法对运维人员分析监控指标数据非常有帮助，因为其不需要预先判断分析的指标是否符合正态分布或其他概率分布，这对了解复杂系统的监控数据规律很关键，可以帮助运维人员找出周期性指标数据的周期波动变量。”

图 4-12 所示为通过 KS 检验方法处理交易笔数指标数据的效果。图中与时间轴平行的灰色时间序列代表处理后的正常状态，其中，黑色区域是检测出异常的时间窗口。目前有三个检出异常的黑色窗口，分别对应 1 次周二交易量增加、1 次周二交易量降低和 1

次周一交易量降低。这些异常是人眼观察不到的，用平均值 3 倍标准差规则也无法检测出来。如果这些异常是由系统运行异常导致的，接收到告警后，运维人员及时介入就有可能降低影响范围，保障未来交易量不会受到更大的影响，从而提升用户数字体验。

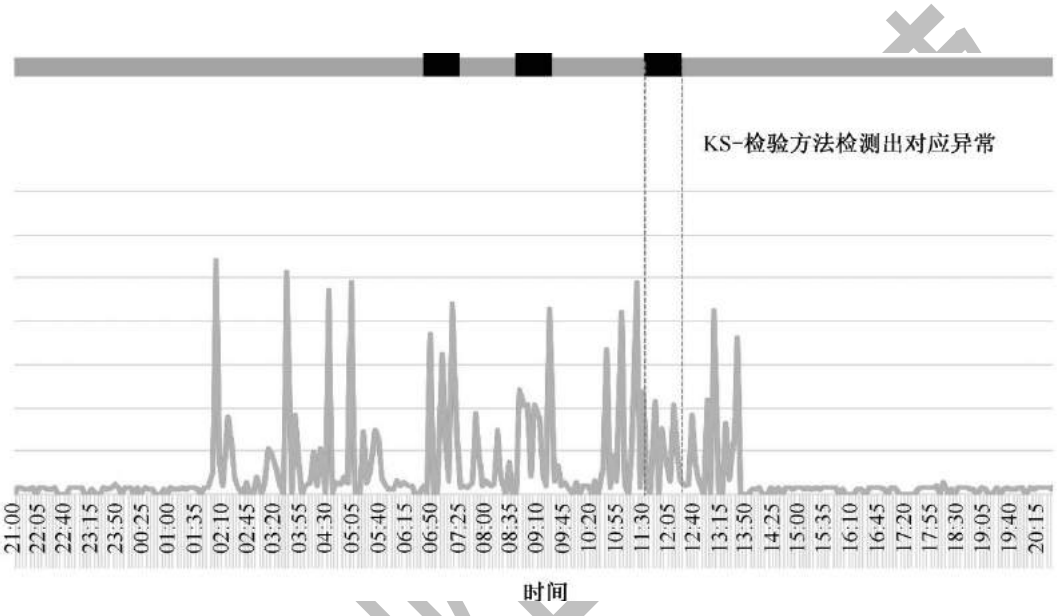


图 4-12　使用 KS 检验方法对交易笔数指标异常情况发出告警

时间序列数据异常检测中的一个重要问题是概念漂移，时间序列中的数据是流动的，有实时性且数据量庞大；随着时间变化，时间序列中数据的分布及标签可能发生变化，出现概念漂移及异常现象。针对概念漂移及异常问题，很多学者提出了解决方法。解决概念漂移及异常的方法大致可分为两类：①对概念漂移及异常进行检测，在检测到发生概念漂移及异常的位置调整学习策略，以适应新的数据；②实时动态调整学习器以适应新数据，不需要考虑是否发生了概念漂移及异常。

所谓时间序列概念漂移及异常是一种数据随时间而发生变化的现象。对于时间序列相关的挖掘来说，当发生概念漂移及异常时，已构建模型的性能指标会随时间而降低，

甚至导致模型完全失效。因此，准确检测和判断是否发生概念漂移及异常，对于时间序列相关的挖掘来说是至关重要的一环，对于概念漂移及异常的检测成为近年来学术界研究的热点问题之一。

由于发生概念漂移及异常时，模型只能通过自我更新的方式来适应新的数据环境，因此，对概念漂移及异常进行实时检测，从而控制模型进行更新，是克服概念漂移及异常最好的方法。传统的概念漂移及异常检测往往通过对模型内部参数的监控来达到检测概念漂移及异常的目的。例如，对于时间序列分类问题，通过监控模型的分类准确率来判断是否发生了概念漂移及异常，这种方式的优点是实现简单、直观，缺点是检测往往由于受噪声的影响而发生误判，且具有滞后性。

高维监控数据的概念漂移及异常时间序列学习是一个很重要的课题，高维数据带来的维数灾难使得传统的机器学习方法不再适用。一个比较经典的检测概念漂移及异常的方法是由 Dasu 等人提出的一种用于检测高维时间序列的概念变化问题的信息理论方法。该方法基于空间分割方案，运用 KL 散度度量两个经验分布之间的距离。但是，该方法在离散化划分之后才能求得概率密度，经过离散化后运用 Bootstrap，使算法需要花费较大的时间代价。另外，该方法只能判断两类情况下的概念漂移及异常，对多类情况只能两两判断或采用其他策略。

综上所述，对于高维时间序列的概念漂移及异常检测问题，关键是找到既能满足时间序列的实时性要求，又能准确判断概念漂移及异常是否发生的方法，从而为模型更新提供帮助。尽管目前有不少检测方法，但受时间序列自身特点的限制，其尚无法很好地解决概念漂移及异常问题。另外，大数据下的概念漂移及异常呈现多样化的特点，不同类型的概念漂移及异常所需方法也无法统一。

时间序列概念漂移的研究在机器学习和数据挖掘领域的重要性与日俱增，并在处理途径方面呈现多样化的趋势，从近年来机器学习与数据挖掘领域的一些国际权威期刊论文和国际顶级会议论文来看，时间序列概念漂移的挖掘和分类研究正日益成为学术界关注的焦点，对数据流概念漂移的研究已经开始与转移学习、进化计算、特征选择、聚类、时间复杂度分析、社会计算等结合。因此，从趋势上来说，已有各种模式分类的理论和算法都可与概念漂移相结合，从而引出更多新的研究问题。

4.4 预测分析：使应用性能风险防患未然

4.4.1 技术简介

面对智能、互联时代更广泛的用户群、更多类型的终端接入、更复杂的应用技术架构，采用数据驱动、机器分析决策代替人工运维方式建设运维系统、保障服务质量目标、提供应用运维服务已是大势所趋。下面介绍如何利用现有的开源、商业版工具分析监控数据，发现应用潜在风险，规避故障。分析手段涉及统计学方法、人工智能算法。

在用户受影响之前，预先发现问题并处理的能力对提供在线互联网服务的企业尤为重要。毕竟用户体验下降，甚至服务中断会直接影响企业营收。在互联网公司中，通过分析监控数据实现主动探伤、预防风险的公司已经非常普遍，Netflix（面向全球的在线视频服务提供商）就是其中之一。Netflix 依赖互联网在线平台向全球用户提供影音视频服务，非常重视这方面能力的建设，已经实现了一定范围的应用异常预警和预处理。由于需要面向海量终端用户提供在线服务，应用性能和用户数字体验直接决定企业营收，互联网行业对运维系统建设的重视程度相比于制造业、政府等要高。2015 年，Netflix 在线用户数达到 7500 万，年营业额达到 62 亿美元，其经营的首要目标是为用户提供极致用户体验的在线视频服务。为达到此目标，应用运维至关重要。Roy Rapoport 形象地将 Netflix 基于云平台的视频服务运维保障面临的挑战描述为“从一群长相和行为都相似的牛群中，找出最与众不同的那头牛”。如果应用系统包含上千个无状态的计算集群节点，所有节点都在运行同样的代码，分担相同规模的计算负载，那么从这些集群中选出异常节点并不容易。

4.4.2 深入浅出应用实践

为了解决上千节点运行期的海量监控数据筛选问题，Netflix 于 2012 年引入了异常检测算法。约克大学（University of York）的 Victoria J. Hodge 和 Jim Austin 将异常检测（Outlier Detection）定义为“检测导致明显性能下降的运行期异常事件，如自动检测飞机发动机运转过程中的异常的传感器指标数据”。Netflix 使用的方法类似，即首先利用算法识别节点反馈的监控指标正常运行的状态数据模式，然后过滤集群中监控指标状态数据模式类似的节点，将其识别为正常，剩下的就是疑似异常节点。

Netflix 已经能够在不需要人工定义什么是正常节点运行行为的情况下，自动找出异常状态节点。由于计算集群部署在云端，其实现了集群规模的可伸缩弹性控制功能。异常节点由系统自动清除处理，负载自动由新创建的或其他正常的节点接替。因此，不需要通知任何人干预处理。为了排查故障原因，自动处理过程和异常节点的状态数据会被保存并通知相关工程师。通过应用异常检测算法，计算集群中异常节点定位、故障排查和恢复的人工工作量大幅度降低了，服务质量也有了显著提升。Netflix 在利用异常检测算法检测监控数据方面的尝试验证了通过智能运维系统替代人工运维、实现大规模复杂应用运维管理的可行性。

分析指定时间范围内应用指标数据是否存在异常最简单的统计学方法是计算指标的平均值和标准差。通过这种方法，我们能很快发现持续采集的监控数据中指标波动异常的时间范围。例如，用户请求并发量的平均值环比显著升高，则应用有可能受到恶意攻击，需要定义告警策略并通知相关责任人。

当关键产品服务发生异常时，在凌晨或其他任何时间生成告警短信或电话以通知责任人都是有必要的。但是，当产生的告警并没有明确指出异常原因，或者根本就是错误告警时，就没有必要推送了。例如，某弹性组计算集群服务节点的 CPU 使用率升高，导致使用率升高的原因很多，且集群节点故障宕机并不影响应用业务正常运行，因此，基于此指标告警就意义不大。如开发运维一体化领导者、资深应用运维工程师 John Vincent 所说：“告警疲劳是我们现在唯一待解决的问题，我们需要让告警更智能，否则我们就得疯掉！”

从理论上说，好的告警需要有较高的信噪比，能指示关键 KPI 指标上实时产生的异常数据点，并能与明确有所指的告警信息匹配，引导责任人快速定位、修正问题。假如要监控某应用未授权用户尝试非法登录系统的行为，采集的指标数据的概率分布为高斯分布，概率密度函数如图 4-13 所示，其中，μ为指标平均值，σ为标准差。越靠近高斯分布钟形曲线边缘的取值，为异常数据的可能性越高。在运维数据分析场景下，这种方法最常用的场景是利用一段时间的监控数据，计算概率分布，并通过标准差设置告警阈值，然后计算实时采集的数据偏离平均值的程度来判断是否触发告警。例如，若监控的未授权登录量指标符合高斯分布，则可以设置告警策略为筛选未授权登录量比平均值 3 倍标准差大的时间。

对于统计学方法，要围绕实际场景，本着计算简单、结果有效的原则进行选择。因为我们面对的是几万甚至百万级别的指标，过于复杂的统计学方法会给监控系统带来巨大的负担，影响产生结果的时效性，我们也不可能对每个甚至每类指标都定义统计学方法。

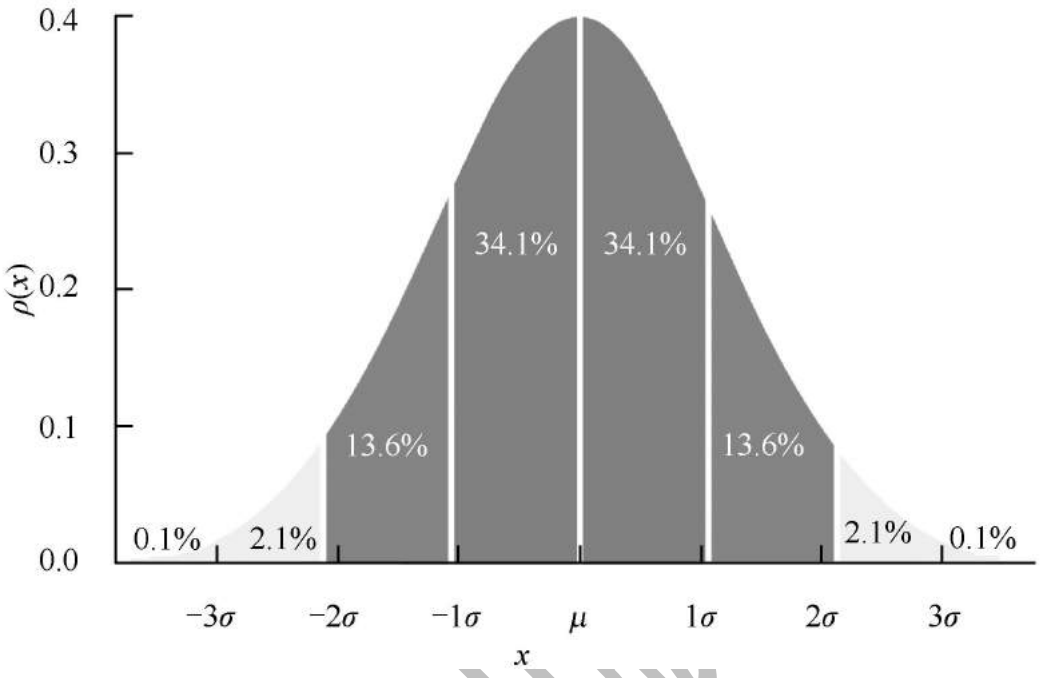


图 4-13　呈高斯分布的指标的概率密度函数

4.5 因果推理：专家经验辅助决策支持

4.5.1 技术简介

在实际运维场景下，很多应用故障的原因相当复杂，故障不能被直接监控到，或者不能靠设计确定性计算方法来分析现有监控数据，找到问题线索。在这种情况下，利用知识工程手段建设专家系统，利用非确定性计算方法积累专家经验，并基于经验推理来分析解决应用运维场景下的风险管理问题，是应对未来应用系统复杂度快速增长、运维成本增加的可行的技术手段。实现具备推理分析能力的应用智能运维系统，需要突破的技术难点主要有：①积累专家经验知识，形成专家系统知识库，为构建知识型人工智能运维系统提供基础支撑；②利用知识库中积累的知识，在出现异常时自动推理分析以找到最优解决方案。

寻找第一个难点的解决途径，需要首先从现有知识工程领域的研究成果下手。在人工智能领域，通过积累经验知识提升人工智能的水平已经不是一个新话题。早在 1977 年，美国斯坦福大学的计算机科学家、图灵奖获得者爱德华·费根鲍姆（Edward A. Feigenbaum）教授就提出，传统的人工智能忽略了具体的知识，人工智能必须引进知识。在第五届国际人工智能会议上，费根鲍姆教授第一次提出了知识工程的概念，并带领团队研发了第一代知识工程驱动的专家系统。如今盛行的知识图谱又将知识工程推向了一个新高度。

在 IT 运维领域，经验知识的积累主要体现在数据采集策略、指标告警策略、分析仪表盘、报表模板和 CMDB 方面。这些专家知识以结构化、半结构化的方式定义了针对不

同类型的应用中间件和运行环境支撑设备如何采集指标、如何判断异常状态，以及如何管理应用部署配置等相关知识。

这些知识固化在运维软件系统中，对辅助运维人员监控应用、发现风险发挥了重要作用。但是，应对技术架构、拓扑结构较为复杂的互联网应用，微服务架构已经力不从心。要找出包含几十种中间件和数据库、对接公有云服务和私有云服务、连接手机和汽车等多种智能终端的应用的潜在风险，定位故障原因，需要更加智能的专家系统。这些专家系统不但要能积累海量知识，而且要能基于条件自动关联知识进行因果推理分析，替代人脑在海量知识中找出答案。

1. 具有应用运维经验知识的专家系统

专家系统作为早期人工智能的重要分支，是一种在特定领域解决问题的能力达到专家水平的程序系统。专家系统一般由两部分组成：知识库与推理引擎。它根据一个或多个专家提供的知识和经验，通过模拟专家的思维过程进行主动推理和判断，从而解决问题。第一个成功的专家系统 DENDRAL 于 1968 年问世。1977 年，费根鲍姆将其正式命名为知识工程。目前获得广泛关注的知识图谱技术是在当前技术的发展背景下，知识工程演进到新阶段的产物。

对于应用运维场景，知识图谱提供了一种定义运维领域的经验知识，以及应用实体及其相互间部署、交互、网络拓扑等关系的结构化方法。知识图谱对应用本身及其相关的实体范围内可以识别的客观对象和关系进行规范化描述，形成运维智能化支撑的知识库。知识图谱本质上是一种语义网络，其中的节点代表实体或概念，边代表实体/概念之间的各种语义关系。

2. 使用知识对非确定性问题进行因果推理分析

因果推理分析是 UCLA 教授、图灵奖获得者 Judea Pearl 在 Probabilistic Reasoning in Intelligent Systems 一书中提出的，其将人工智能领域处理非确定性问题的方法划分为三个学派：逻辑主义学派（Logicist）、新计算学派（Neo-Calculist）和新概率论（Neo-Probabilist）[1]。

逻辑主义学派试图用非数字技术来处理不确定性，主要运用非单调逻辑。新计算学派使用不确定性的数值表示不确定性，但认为概率积分（Probabilistic Calculus）不足以完成这项任务，因此，其发明了全新的微积分，如 Dempster-Shafer 理论、模糊逻辑和确定性因素。新概率论仍然存在于概率理论的传统框架中，同时试图用执行人工智能任务时所需的计算工具来支持其理论。处理不确定性的延伸方法（也称为生产系统、基于规则的系统和基于程序的系统）将不确定性视为附加到公式的广义真理值，并将任何公式的不确定性计算转化为其子公式的不确定性计算。在有意的方法（也称为声明性或基于模型的方法）中，不确定性与可能的世界状态或子集相连。

对于运维场景，推理任何现实问题总需要对目标场景进行一些抽象、对高维数据进行一些降维以简化计算。准备知识来支持推理的行为要求我们留下许多未知、未说或粗略的总结事实。例如，如果我们定义“HTTP-500 错误代表服务器端页面 ASP、JSP 代码解析错误”或“应用可用性终端和服务端节点日志中同时出现 Out of Memory 异常，代表内存溢出导致的应用宕机”等规则来对知识和行为进行编码，那么将有许多我们无法列举的例外情况及规则适用条件。

[1] Judea P. Probabilistic Reasoning in Intelligent Systems[M].US LA: Cambridge University Press, 1988.

要实现人工智能驱动的运维，知识是不可缺少的。在设计运维系统的智能化处理过程时，只采用过程性方法来定义风险是不够的。应对复杂策略，还必须使用说明性方法及积累的历史经验和领域知识。当解决问题时，单纯设计算法来实现高效率的求解，而不考虑由于数据维度的增加、条件组合数的无限增加而导致的搜索量增加，也是不切实际的。

网络表示不是人工智能系统外在的。大多数推理系统使用复杂的指针系统（将事实分组为结构，如框架、脚本、因果链和继承层次结构）的索引网络来编码相关性。这些结构虽然被纯粹的逻辑学家所回避，但在实践中已经证明是不可或缺的，因为它们将执行推理任务所需的信息放在接近任务所涉及的命题的位置。事实上，人类推理的许多模式只能用人类遵循这种网络所制定的途径的倾向来解释。

本书讨论的网络的特点是它们具有明确的语义。换句话说，它们不是为使推理更有效率而设计的辅助设备，而是知识库语义中不可或缺的一部分，它们的大部分功能甚至可以从知识库派生出来。

4.5.2 深入浅出应用实践

在特定场景下面向具体问题的应用实践中，对于不确定性推理，可按照是否采用数值描述不确定性来选择不同的方法：一种是数值方法，它是一种用数值对不确定性进行定量表示和处理的方法；另一种是非数值方法，它代表除数值方法以外的其他各种对不确定性进行表示和处理的方法。

对于数值方法，其又可以根据所依据的理论分为两种不同的类型：一种是基于概率

论的有关理论发展起来的方法，如确定性理论、主观 Bayes 方法、证据理论和概率推理等；另一种是基于模糊逻辑理论发展起来的方法，如模糊推理，它可以用来对由于操作系统最大线程数限制，或者应用系统线程数过多导致的服务异常问题进行推理判断。如果采用传统的根据预定义条件判断的方法枚举这种操作系统配置导致的异常，那么前期需要配置的数据量将非常庞大。

考虑投入产出比，我们不可能为每种可能发生的异常情况和每个监控指标设计特定的数据采集策略与异常检测算法。因此，基于先验知识，使用由现象到本质的不确定性推理（Induction）来解决更为合适，这样虽然不能保证完全准确，但能在一定程度上替代运维专家辅助决策，给出解决问题的正确方向。

推理过程实质上是不断寻找和运用可用先验知识的过程。在应用运维场景下，可用先验知识是指根据经验积累的风险现象、应用配置前提条件，以及可与历史风险处理知识库匹配的知识。类似采用推理解决应用系统线程数过多的问题，针对运维过程中需要采用不确定性推理方法处理的场景，需要考虑的基本问题包括以下方面。

1. 不确定性的表示

不确定性的表示包括知识不确定性的表示和证据不确定性的表示。知识不确定性的表示通常需要考虑两方面的问题：如何能够比较准确地描述问题本身的不确定性，以及如何定义能便于推理过程中不确定性的计算。

知识的不确定性通常是用一个数值来描述的，该数值表示相应知识的确定性程度，也称为知识的静态强度。证据的不确定性表示推理中的证据有两种来源：第一种是应用

出现故障后在求解问题的原因的过程中所提供的初始证据，如系统内存溢出问题、内存使用率超阈值等先验知识；第二种是推理过程中得出的中间结果。通常，证据的不确定性应该与知识的不确定性表示保持一致，以便推理过程能对不确定性进行统一处理。

2. 不确定性的匹配

推理过程实质上是不断寻找和运用可用知识的过程。可用知识是指其前提条件与综合数据库中的已知事实相匹配的知识。那么如何匹配呢？目前常用的解决方案是，设计一个用来计算匹配双方相似程度的算法，并给出一个相似的限度，如果匹配双方的相似程度落在规定的限度内，那么称双方是可匹配的。

3. 组合证据不确定性的计算

在不确定性系统中，知识的前提条件既可以是简单的单个条件，也可以是复杂的组合条件。匹配时，一个简单条件只对应一个单一的证据，一个组合条件将对应一组证据，而结论的不确定性是通过对证据和知识的不确定性进行某种运算得到的。所以，当知识的前提条件为组合条件时，需要有合适的算法来计算组合证据的不确定性。

4. 不确定性的更新

由于证据和知识都是不确定的，那么就存在两个问题：如何利用证据和知识的不确定性更新结论的不确定性；在推理过程中，如何把初始证据的不确定性传递给最终结论。

对于第一个问题，一般的做法是按照某种算法，由证据和知识的不确定性计算结论

的不确定性。对于第二个问题，一般的做法是把当前推出的结论及其不确定性作为新的证据放入综合数据库。

目前已有一些解决此类问题的研究成果，如墨尔本大学 CLOUDS 实验室的 Rajkumar Buyya 等人[1]分析了以服务等级协议（SLA）感知方式解决计算资源分配问题的关键挑战，提出了基于计算风险管理的 SLA 感知推理资源分配策略。为利用云计算的资源动态分配能力，一些文献[2]分别介绍了集群系统及多层应用的资源的随需分配实现策略，详述了以 SLA 及 QoS 事件触发方式实现云计算资源动态调配的主要机制。

为实现服务质量感知的资源自适应配置，Amir Vahid 等人[3]通过语义查找、分析和匹配用户需求，实现了多云环境下的 QoS 感知云服务选择；针对私有云环境中资源交付与调度的高效实现问题，我们提出了一种基于分裂聚类的云应用的资源交付与配置方法，优化了虚拟机与虚拟设备的资源配置[4]；为了优化云应用在多种云环境下的部署策略，Grozev Nikolay 等人针对多云联邦环境下部署的三层 Web 应用性能进行了性能建模，并给出了跨多云部署的三层 Web 应用运行期基于负载的资源动态响应是资源优化策略

[1] Rajkumar B, Chee S Y, Srikumar V, et al. Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility [J]. Future Generation Computer Systems, 2009, 25(6): 599-616.

[2] Marcos D d A, Alexandre d C, Rajkumar B. A Cost-Benefit Analysis of Using Cloud Computing to Extend the Capacity of Clusters [J]. Journal of Cluster Computing, 2010, 13(3): 335-347.

Luis M V, Luis R M, Rajkumar B. Dynamically Scaling Applications in the Cloud [J]. Computer Communication Review, 2011, 41(1): 45-52.

Wu L, Garg S K, Versteeg S, et al. SLA-based Resource Provisioning for Hosted Software as a Service Applications in Cloud Computing Environments [J]. IEEE Transactions on Services Computing, 2014, 7(3): 465-485.

[3] Amir V D, Saurabh K G, Omer F R, et al. CloudPick: A Framework for QoS-aware and Ontology-based Service Deployment Across Clouds[J]. Software: Practice and Experience, 2015, 45(2).

[4] 许力，周进刚，张霞，等. 云应用资源交付与分裂聚类调度方法[J]. 计算机工程，2011, 37(11):52-55.

的结论[1]。现有文献研究成果[2]主要通过事件或人工触发被动调整资源配置的方式来实现资源与负载适配，存在响应不及时、调整效果不明显的问题。

所述目标应用场景如图 4-14 所示。由基础设施提供商（InP）提供的物理网络（SN）、虚拟网络（VN），以及由服务提供商（SP）通过部署云应用提供的服务之间的关系可抽象为相互依赖的三个层叠平面结构。其中，服务层实例由服务提供商以预定义模板的形式通过 IaaS 平台部署云应用来构建。在此过程中，云应用运行期所需的虚拟网络通过将模板中包含的依赖资源描述文件转换为虚拟网络来构建请求，然后由 IaaS 平台处理并创建对应的虚拟网络实例。运行期间，服务提供商可根据业务目标调整预期的服务质量目标，云管理系统则定期根据服务质量目标和历史监控指标数据生成虚拟网络重配置请求，修正虚拟网络配置以规避风险。

[1] Grozev N, Buyya R, Performance Modelling and Simulation of Three-Tier Applications in Cloud and Multi-Cloud Environments [J]. The Computer Journal, 2018, 58(1): 1-22.

[2] Mukaddim P, Rajkumar B. Resource Discovery and Request-Redirection for Dynamic Load Sharing in Multi-Provider Peering Content Delivery Networks [J]. Journal of Network and Computer Applications, 2009, 24(1): 976-990.

Amir V D, Saurabh K G, Rajkumar B. QoS-aware Deployment of Network of Virtual Appliances across Multiple Clouds [C]. IEEE CloudCom 2011, IEEE, Athens, Greece, 2011.

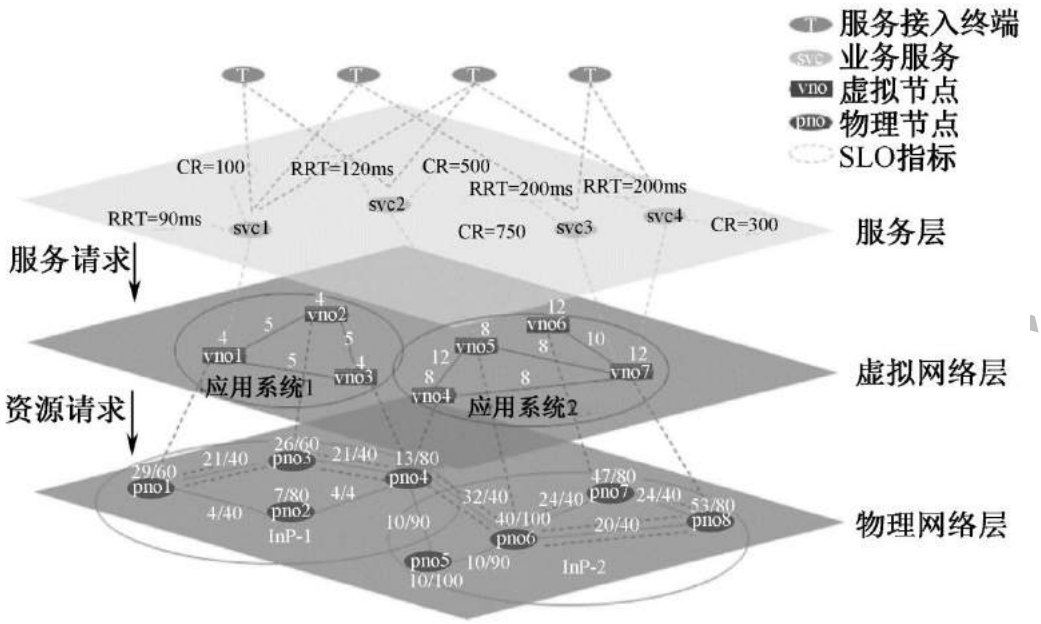


图 4-14　基于虚拟网络的服务部署逻辑层次映射

服务质量目标是服务提供商向其用户提供的请求响应时间、请求并发量等业务服务质量承诺。基础设施提供商以虚拟网络的形式向服务提供商交付所需资源。为了方便计算，所有网络资源量（网络带宽、计算资源）被抽象定义为整数。在图 4-14 中的物理网络层中，标注于物理网络节点和物理网络链路之上的以斜线分隔的数字分别代表可用资源和资源总量；在虚拟网络层中，标注于虚拟网络节点和虚拟网络链路之上的数字代表需要的资源量。

4.6 自治控制：应用运维过程的自动化管理

4.6.1 技术简介

基于自治控制的理念实现应用智能运维系统也是降低人工运维工作量的一种思路，对该课题的研究已存在一些行之有效的研究成果。其中，大多数成果集中在通过经典控制理论、自治计算和机器学习来实现集中式的资源管理方面。D.Ionescu 等人[1]提出了一种基于 IBM MAPE-K[2]自治控制框架的虚拟环境管理平台设计方案，其能够通过集中控制节点实现资源的自交付（Self-Provisioning）及自优化（Self-Optimization）。自治计算的处理被视为一种具有持续线性参变不确定性的非线性不确定系统。P.T.Endo 等人[3]设计了面向云计算基础设施的自治云管理系统，该系统能够对云基础设施的资源使用率做持续优化，并降低运维管理成本。为了研究云计算服务自治管理的可观测性和可控制性，L. Checiu 等人[4]提出了基于自治计算模型的输入－状态－输出（Input-State-Output）数学模型。通过预定义策略实现自治管理是一种简洁有效的手段，M. Sedaghat 等人[5]基于这

[1] Ionescu D, Solomon B, Litoiu M, et al. A Robust Autonomic Computing Architecture for Server Virtualization [C]. INES 2008, International Conference, 2008.

[2] http://www-03.ibm.com/autonomic/pdfs/ACBP2_2004-10-04.pdf.

[3] Endo P T, Sadok D, Kelner J. Autonomic Cloud Computing: giving intelligence to simpleton nodes[C]. Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference, 2011.

[4] Checiu L, Solomon B, Ionescu D, et al. Observability and Controllability of Autonomic Computing Systems for Composed Web Services[C]. Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium, 2011.

[5] Sedaghat M, Hernandez F, Elmroth E. Unifying Cloud Management towards Overall Governance of Business Level Objectives[C]. Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium, 2011.

种理念提出了基于策略的自治云环境管理方案。该方案利用更高层次的管理系统监控整个云计算环境与其中部署的服务的状态，当发现服务状态不能满足业务目标时，则调整底层资源控制策略来适应变化。Wenjie Liu[1]则利用不同类型的应用（如计算密集型应用、存储密集型应用）对不同类型的资源的消耗程度存在差异的特点，将不同类型的应用系统自动调配部署，从而提高了资源使用率和提升了应用可用性，达到了云计算环境下自治管理云计算资源的目标。

自治管理是为了降低日益复杂的数据中心环境中人工干预管理的复杂度，通过自动化的监控、管理和控制手段来代替传统人工手动处理异常、管理配置等的过程。当前常用的自治管理技术是 IBM 提出的 MAPE-K 自治计算框架（Autonomic Computing Framework）。MAPE-K 是 Monitor、Analysis、Plan、Execution and Knowledge Base 的缩写，其主要设计理念是以知识库为核心构建集监控、分析、计划和控制于一体的闭环自动控制系统，从而实现能够自配置（Self-Configuration）、自恢复（Self-Healing）、自优化（Self-Optimization）和自保护（Self-Protection）的自治控制系统。自治计算的层次可分为以下两种。

（1）自治元素（Autonomic Elements）：自治元素是组成自治系统的基本元素，如具有自治功能的服务器、路由器等设备。

（2）自治系统（Autonomic Systems）：自治系统给自治元素提供一个相互合作、通信的环境，并且提供人机界面。

[1] Liu W J, Li Z H. Research and Design of Autonomic Computing System Model in Cloud Computing Environment[C]. 2011 International Conference on Multimedia Technology (ICMT), 2011.

每个自治元素拥有一个 MAPE-K 闭环自动控制系统。在结构上，自治元素包含传感器、执行器及知识库（Knowledge Base）三部分。传感器负责感知周围环境，执行器通过推理分析得到需要执行的动作集合以调整和控制外部环境。推理和分析过程是依靠知识库中记录的数据来完成的。在一个自治计算环境中，新加入的元素将被自动设置，不同的自治元素间相互合作完成一个任务，这个任务可能由管理员通过所谓的政策来指定，但这个政策的定义是用高层语言编写的。各自治元素通过项目合作、谐调、优化来完成任务。

4.6.2 深入浅出应用实践

通常，部署在云环境下的互联网应用所面临的环境是开放的、动态的，云与云应用之间应能按多种静态链接和动态合作方式，在开放的网络环境下实现互连、互通、协作和联邦。这就要求建立能够支撑云环境下服务交互的云服务管理框架，采用面向服务的方式动态组织应用系统之间的服务交互，并且能够面向特定的应用场景，基于语义知识自主地做出交互服务决策。这种具有自适应服务交互特性的运行支撑框架涉及的研究内容包括以下几方面。

（1）自适应组合服务技术。对于公共云服务，用户的业务目标通常需要通过多个服务的组合来实现，这就需要建立一种被调用者和调用者一对多的关联关系，并提供相应的组合策略。这一过程涉及的需要突破的关键技术有面向服务的业务目标分解技术、聚合服务技术和多目标服务组合技术。

（2）云协同服务技术。公共云服务是一个复杂的需求形态，需要跨机构、跨领域的服务协同才能完成。因此，运行支撑框架，需要研究能支持服务需求特征提取，业务流

程编排，跨域服务交互，协同运行监控，实现跨组织、跨地域、多应用系统协同服务的技术，从而满足云计算复杂应用系统的业务敏捷化的需要。

（3）服务动态演化技术。服务的动态演化是支持云计算多变环境下自适应特征的一个良好体现。互联网的开放、动态和多变，以及用户使用方式的个性化要求，决定了云应用下的服务构件在发布之后，会在长时间内持续不断地演化；不同服务之间的运行、调整和演化并不是独立的，需要相互协作。服务的动态演化包含了服务构件本身的演化及聚合服务的动态演化，主要需要解决两方面的技术问题：服务调用的透明性技术和服务状态转换技术。

在典型的能够支撑多个云应用运行的多主机集群云计算环境下，基于集中部署的管理系统提供了对负载及资源使用率都在动态变化的云应用的性能和稳定性的保障能力。同时，提升主机、网络设备的资源使用率需要考虑很多动态变化的不确定因素，采用集中控制逻辑实现会非常困难。基于多智能体系统，利用多智能体协作方式解决此类问题，能够有效降低集中控制策略的复杂度和系统实现的工作量，被业界广泛采纳。

多智能体技术是实现软件系统自治的一个主流技术，也是解决复杂应用运维行之有效的技术方案。软件智能体能够系统化地开发可以适应随机的、动态的变化环境和情况的复杂应用。智能体的主要特征可归纳为封装、面向目标、反应性、自治性、主动性、交互性和持久性[1]。与对象相比，智能体间的交互更像请求服务而不是方法调用。关键是，与智能体交互相比，面向对象的方法缺少能够灵活控制方法双向调用通信的概念和机制。面向对象的构件的元素都是在设计期预定义好的、静态的；而在面向智能体的软件工程中，元素是动态创建的，因此，智能体比传统构件显示了更多的行为和灵活性。智能体

[1] 黎建兴，毛新军，束尧．软件 Agent 的一种面向对象设计模型[J]．软件学报，2007，18(3): 582-591.

间的交互需要一个智能体平台，智能体的通信通常采用智能体通信语言（ACL）。随着智能体理论和技术研究的不断深入，许多学者将智能体的概念、理论和技术引入软件工程领域，出现了面向智能体的软件工程。近年来，面向智能体的软件工程受到了学术界和工业界的高度关注与重视。至今，人们已经提出了许多面向智能体的开发方法学、程序设计语言，以及 CASE 工具和集成开发环境。研究人员正试图从更广的范围系统地开展面向智能体软件工程方面的研究，包括面向智能体的软件复用、项目管理、形式化规范、系统验证和模型检测。领域内也涌现出一批面向智能体软件系统的专业化公司及产品，如 Agent Oriented Software Ltd。IBM、Microsoft、Fujitsu 和 Toshiba 等著名软件产品生产商也纷纷加强在该领域的技术和产品研发。OMG 和 FIPA 等国际标准化组织也开始致力于智能体技术的标准化工作，并推出了一系列关键的智能体技术规范和标准，如 FIPA 提出了智能体通信语言。如今，面向智能体的软件工程正与其他计算机技术进行着更加紧密的结合，如面向服务的计算、语义 Web、对等计算、普适计算、网格计算和自治计算等[1]。一些开源的智能体平台产品，如 JADE、Tryllian 智能体软件开发包、KATO 也逐渐获得了业界的广泛应用。

案例

虚拟化技术的快速发展使得应用与基础设施环境解耦，为实现更灵活的资源配置交付奠定了基础。云计算使得将虚拟化的计算、存储、网络资源以随需即取的服务形式交付给目标用户成了可能。这种商业模式使企业用户能够以更低的成本灵活、快速地开发和部署应用系统。借助云服务，应用部署人员能以更加敏捷的方式配置和部署具有随需动态伸缩、迁移能力的云应用。当前，基础设施即服务（Infrastructure as a Service，IaaS）平台已被广泛应用，云应用开发平台即服务（Platform as a

[1] http://kato.sourceforge.net/kato.html.

Service，PaaS）服务平台及直接面向终端用户的软件即服务（Software as a Service，SaaS）也逐渐成熟。然而，由于不同云平台在相同配置下存在计算、存储、网络性能的差异，且云应用负载和资源使用率随时间动态变化，如何实现云应用运行期资源的自适应配置，在保障预定义应用服务质量目标的前提下支撑云应用资源分配策略随负载动态衍化以提升资源使用率，是目前该领域需要攻克的关键难题之一。

为了实现云应用资源的自适应配置，首先要能够实时探查云应用的运行期状态，通过分析历史指标数据推理判断云应用资源是否处于资源超配或资源配置不足的状态，之后生成对应的任务，自动调整资源配置。为实现目标效果，需要构建具备运行期监控指标采集、监控数据存储分析和管理控制能力的跨平台监管系统。

云计算环境将物理设备和应用解耦，应用运行直接依赖云计算环境，间接依赖物理设备。云计算环境为多应用提供运行期支撑服务，物理设备不再被一个应用独占。当物理设备、云计算环境中的虚拟设备或应用本身发生资源配置风险时，将影响应用的直接运行。为了有效监管云应用与云计算环境的运行状态，自动发现并处理资源配置风险，本书提出了如图 4-15 所示的云应用运行期管理系统模型。该模型分为应用层风险分析子系统和环境层管理子系统，两个子系统通过消息通信来交换监控指标数据集和风险处理任务。

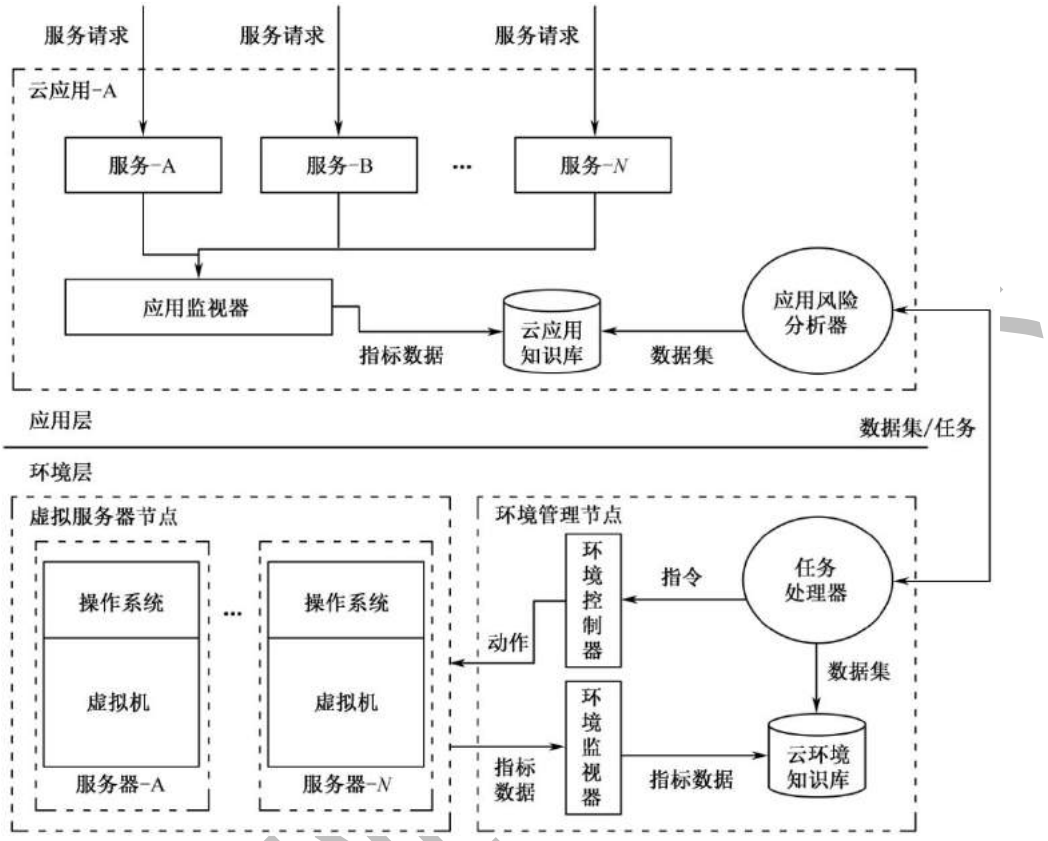


图 4-15　云应用运行期管理系统模型

环境层管理子系统包含的关键组件如下。

环境监视器：部署在云计算环境中的环境管理节点（可以是物理服务器或虚拟服务器），定期采集环境中各资源（物理主机、虚拟主机、网络、操作系统等）运行期的监控指标并将指标数据保存在云环境知识库中，以便供应用处理器查询。

环境控制器：部署在云计算环境中的环境管理节点，接收任务处理器发送的环境控制指令（如迁移虚拟机、提高虚拟机 CPU 的配额、重启虚拟机等），将指令转换成可直接执行的程序，并通过接口调用执行动作。

任务处理器：部署在云计算环境中的环境管理节点，与云应用中的应用风险分析器通信，向其发送指定资源运行期的指标数据集；接收应用风险分析器发送的环境控制任务。

应用层风险分析子系统包含的关键组件如下。

应用监视器：部署在云应用中，定期采集云应用中各服务运行期的监控指标，并将指标数据保存在云应用知识库中，供应用风险分析器查询。

应用知识库：部署在云应用中，存储应用监视器定期采集的指标数据。

应用风险分析器：部署在云应用中，负责维护云应用运行期监控指标的关联推理模型，定期读取云应用知识库中的监控指标数据集和环境层管理子系统从云应用运行支撑的云环境中采集的监控指标数据集，以便更新模型属性；定期执行告警判断策略，并在告警触发时执行风险自动处理，推理任务执行策略，然后向环境层管理子系统中的任务处理器发送任务。应用风险分析器由如图 4-16 所示的三个核心模块组成。

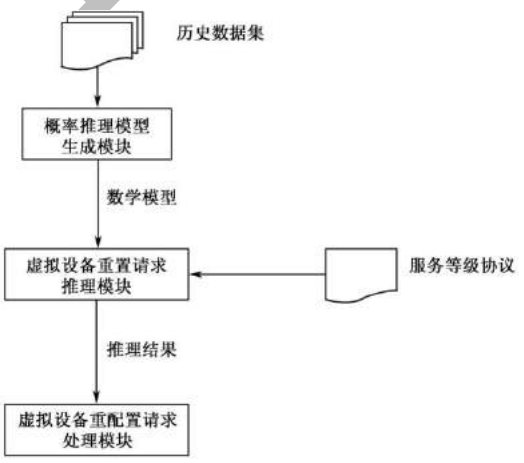


图 4-16　应用风险分析器的组成

概率推理模型生成模块：通过分析定期采集的包含虚拟设备指标历史数据及服务质量指标历史数据的历史数据集，生成基于贝叶斯网络的概率推理模型。

虚拟设备重置请求推理模块：部署在云应用中，存储应用监视器定期采集的指标数据；基于服务等级协议中的服务质量目标定义，利用随机本地搜索算法搜索给定的数学模型，查找资源超配与资源不足概率最大的虚拟节点及虚拟链路，同时通过计算找到各虚拟节点及虚拟链路需要追加或释放的资源量。

虚拟设备重配置请求处理模块：基于推理结果对指定虚拟设备进行重配置。

本章小结

应用智能运维系统能够在降低应用运维系统日常工作压力的同时，赋予运维人员、应用开发人员在发现、处理更复杂的应用系统故障或优化系统性能和稳定性过程中更准确的判断能力。本章深入浅出地介绍了一些常用的应用智能运维技术与实践案例，分别从技术原理和技术应用落地的角度做了介绍，希望能为企业规划应用智能运维系统的前期技术预研提供参考。