GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE

AIOps - 智能故障处理中的系统工程

王亚雷　云兴维智（北京）科技有限公司 CEO

TOP1
技术型企业案例研究智库

100

第七届 全球软件案例研究峰会

运营增长 | 组织发展 | 爆款架构 | 人工智能 | 测试实践 | 运维体系

产品创新 | 研发效能 | 数据平台 | AI/数据驱动 | 测试体系 | AIOps DevOps

体验设计 | 团队管理 | 工程实践 | AI实践 | 大前端 | 区块链

扫描二维码
查看最值得学习的100+创新案例

2018年11月30日-12月3日
北京丨国家会议中心

主办方 msup

上海精品公开课

管理3.0认证课程

时间：12月22-23日 | 地点：上海 | 讲师：林伟丹

大数据及AI挖掘技术

时间：12月22-23日 | 地点：上海 | 讲师：风清扬

高可用架构与设计

时间：01月12-13日 | 地点：上海 | 讲师：沈老师

K8S与service mesh

时间：01月12-13日 | 地点：上海 | 讲师：Jim

备注：扫码查看课程详情，两人以上报名有优惠，详情咨询：15802217295

msup®

AIOps 专业厂商面临的挑战

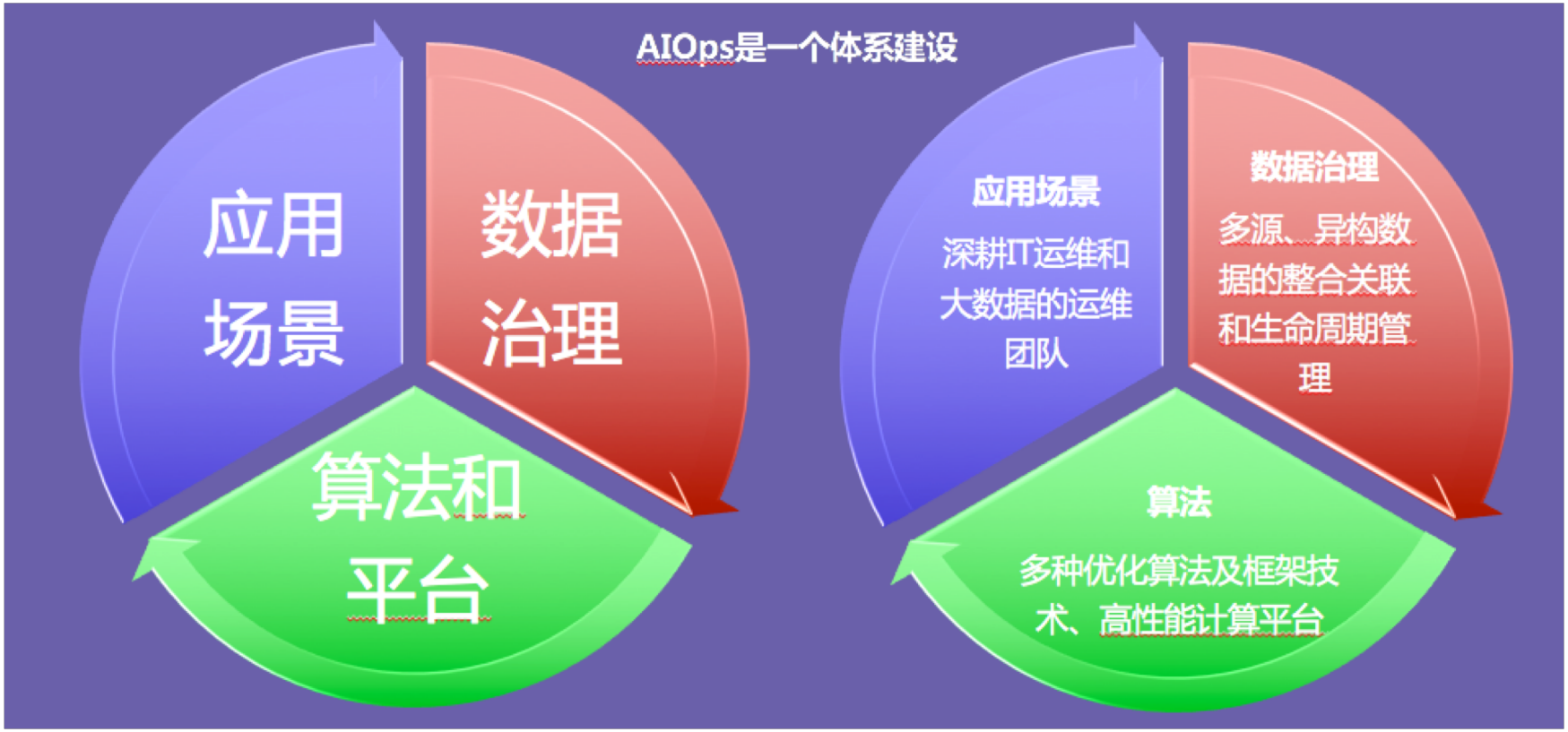
客户运维对象多样性、复杂性

客户运维数据的质量和不可控性

平台算法的可选性、可调性、可演进性 > 算法的先进性

完整的解决方案

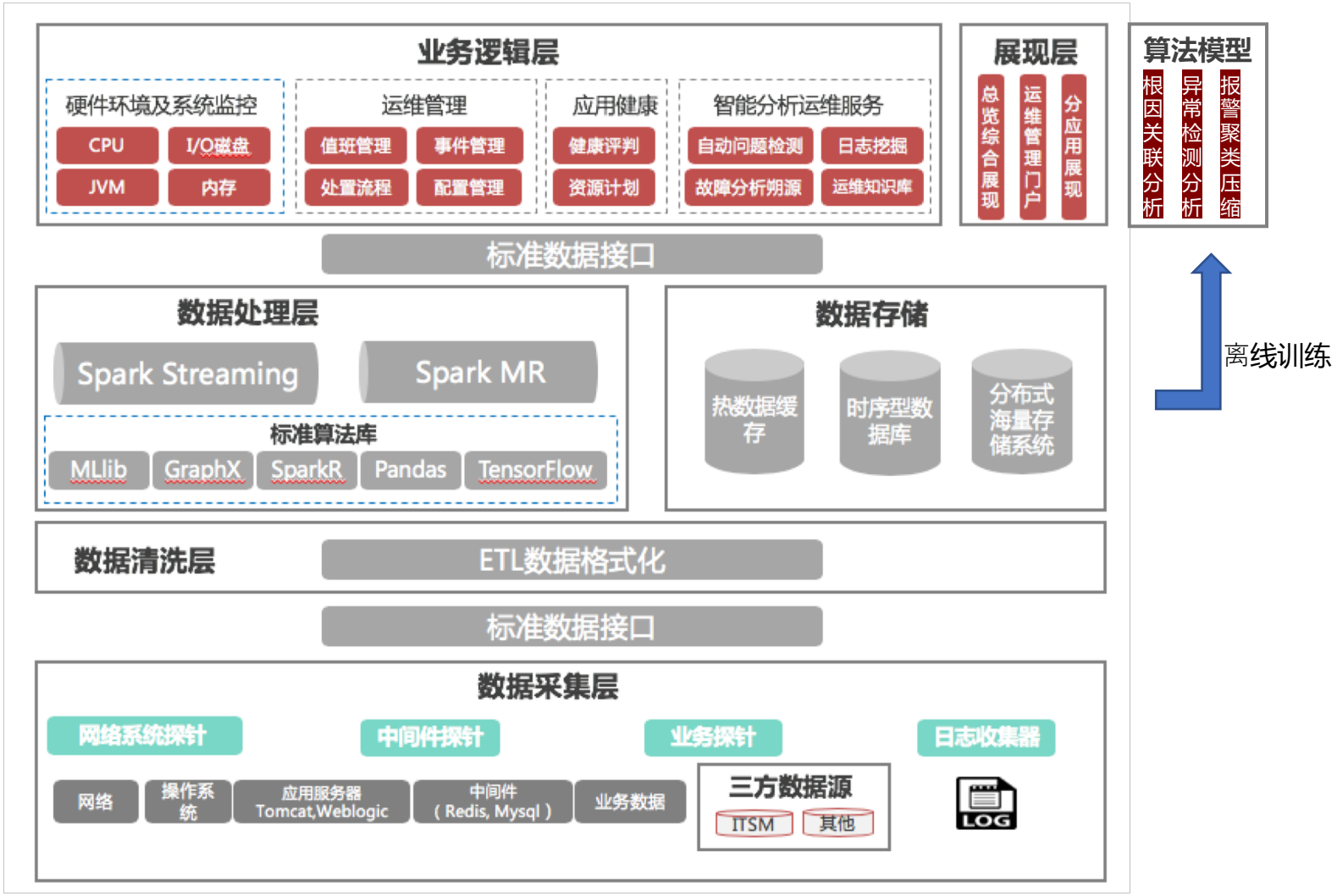
AIOps建设三要素


AIOps是一个体系建设
应用场景
数据治理
算法和平台
应用场景
深耕IT运维和大数据的运维团队
数据治理
多源、异构数据的整合关联和生命周期管理
算法
多种优化算法及框架技术、高性能计算平台

AIOps 建设路径

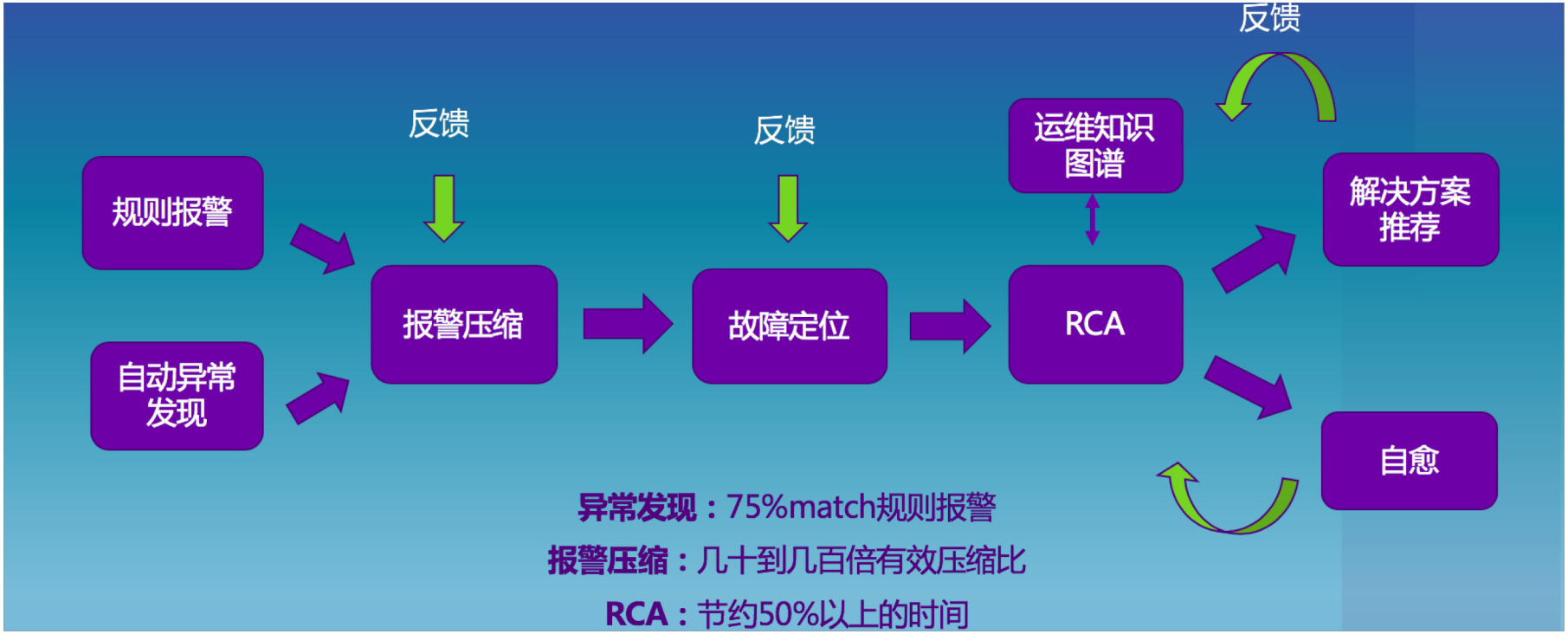

机器学习 / AI主导的
高效排障流程
算法驱动的成本
和性能优化
智能的资源预测和计划：
Capacity Forecasting
and Planning
全域运维数据服务：存、取、查询、更新
基于机器学习的运维知识图谱
统一的数据管理和治理：主题域、数据Schema
运维大数据平台: 存储、处理、计算

AIOps 建设路径


业务逻辑层
硬件环境及系统监控
CPU
I/O磁盘
JVM
内存
运维管理
值班管理
事件管理
处置流程
配置管理
应用健康
健康评判
资源计划
智能分析运维服务
自动问题检测
日志挖掘
故障分析朔源
运维知识库
展现层
总览综合展现
运维管理门户
分应用展现
算法模型
根因关联分析
异常检测分析
报警聚类压缩
标准数据接口
数据处理层
Spark Streaming
Spark MR
标准算法库
MLlib
GraphX
SparkR
Pandas
TensorFlow
数据存储
热数据缓存
时序型数据库
分布式海量存储系统
离线训练
数据清洗层
ETL数据格式化
标准数据接口
数据采集层
网络系统探针
中间件探针
业务探针
日志收集器
网络
操作系统
应用服务器 Tomcat,Weblogic
中间件 (Redis, Mysql)
业务数据
三方数据源
ITSM
其他
LOG

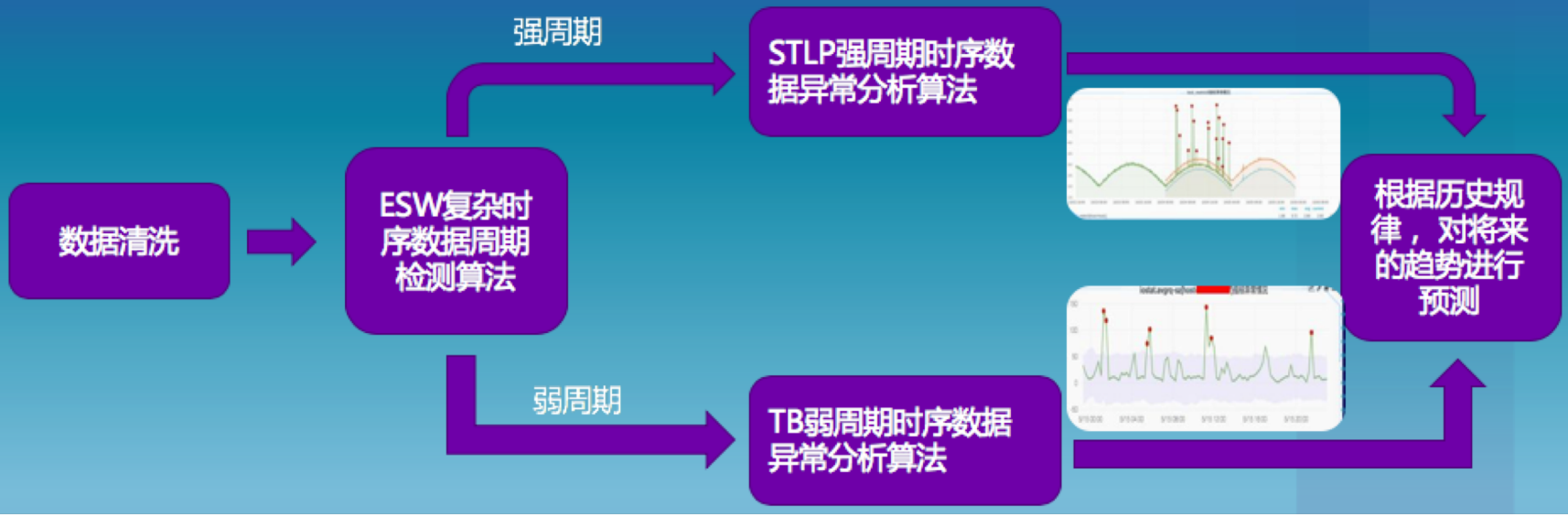
AIOps数据建设 – 运维数据主题模型

AIOps场景建设 - 高效排障流程

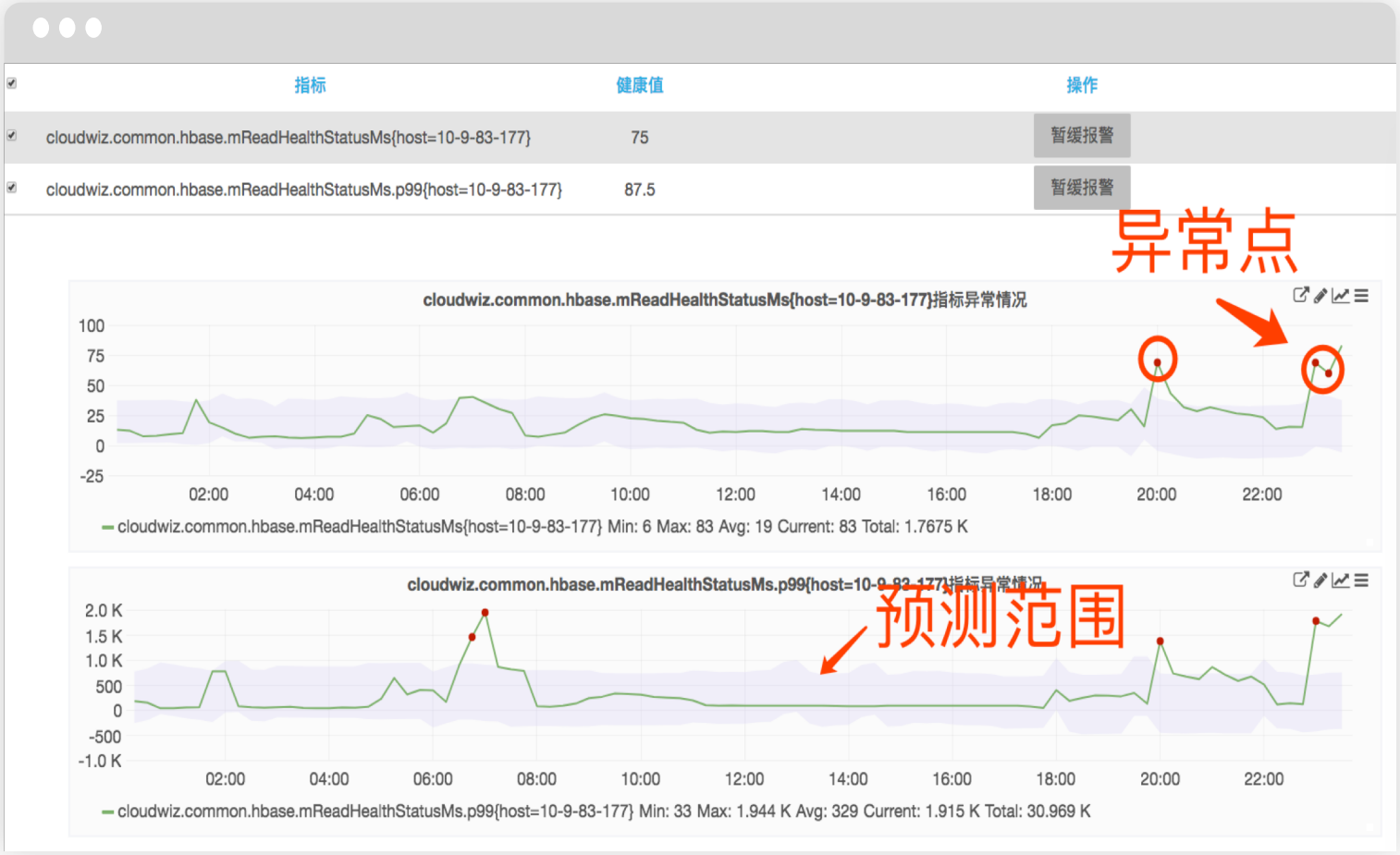


AIOps场景建设 - 异常检测

自动异常检测是通过查询历史数据计算未来时间的历史趋势并对比判断指标是否异常。
其中查询历史数据后依次需要进行数据清洗或补充，使用机器学习算法分解趋势和季节性及噪音，
利用趋势曲线定义规律，发现异常



AIOps场景建设 - 异常检测



AIOps场景建设 - 异常检测

	报警项	bppm(条)	Aiops(条)	备注
同时报警	OS:File system /xx/xx used is xxx%	20	18	
	OS:memory Page in over 293.19	4	3	
	OS:memory Page out over 1679.23	1	2	
	Sessionsused	0	6	人工反馈进行确认
	Phyreads	0	1	人工反馈进行确认
	Blkreads	0	2	人工反馈进行确认
仅bppm报警	OS:磁盘 hdisk219 平均响应时间为 35ms	3	0	Transit / 周期性

AIOps场景建设 - 异常检测

算法对周期性变化的处理

AIOps场景建设 - 报警压缩

某头部银行

1.数据中心IT环境复杂度高

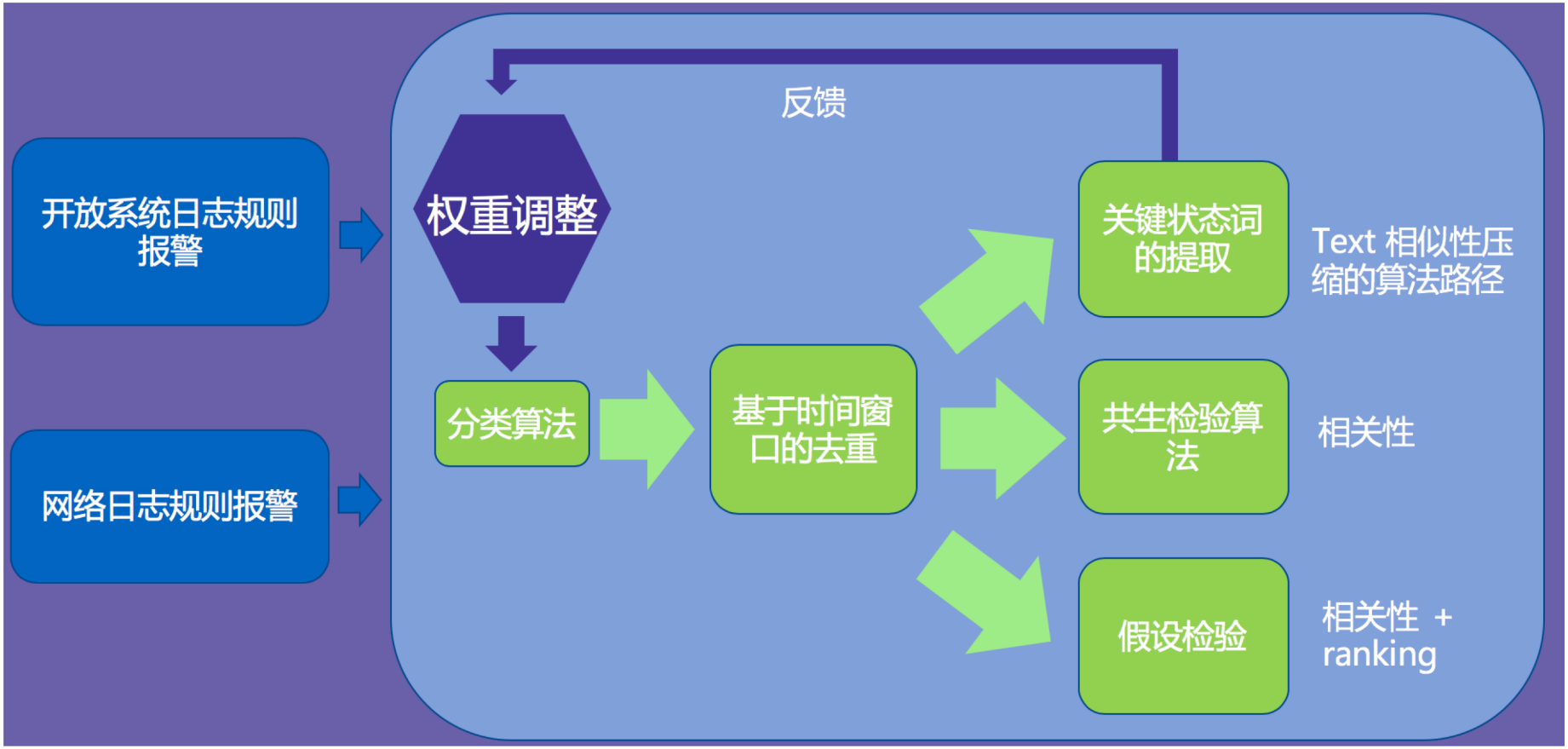
2.日志种类多（网络告警日志，开放系统日志）

3.网络告警日志20万条/天，开放系统日志3万条/天

4.传统告警处置流程繁琐，依靠人力成本居多，效率低下

5.无法发现关联告警

AIOps场景建设 - 报警压缩



AIOps场景建设 - 报警压缩

- 聚类算法对相似的报警进行压缩：编辑距离， 80% 压缩比
- 关键状态词的提取：编辑距离非常近，但不能进行压缩

  Line protocol on the interface Serial x/x/x/xx:x is DOWN
  Line protocol on the interface Serial x/x/x/xx:x is UP

  Member[raw=Caused by: java.net.ConnectException: Connection timed out]
  Member[raw=Caused by: java.net.ConnectException: Connection refused]

- Ngram + FT-Tree

  Normalize (归一化)

AIOps场景建设 - 报警压缩

提升准确率，依赖用户反馈

[CREATED, DELETED, OUT]
[DISCARDING, LEARNING]
[IN, OUT]
[存入, 消费]
[DOWN, UP, YQB_ECN_ACC_UNI]
[FLAP, UP]
[星期一, 星期二, 星期日]
[取款, 存入]
[INFO, WARN]

[CORRECT, WITHOUT]
[COMMAND-LOG, EXIT]
[32654, 39377, 49982]
[星期一, 星期二, 星期日]
[124, SERVER, 闪断]

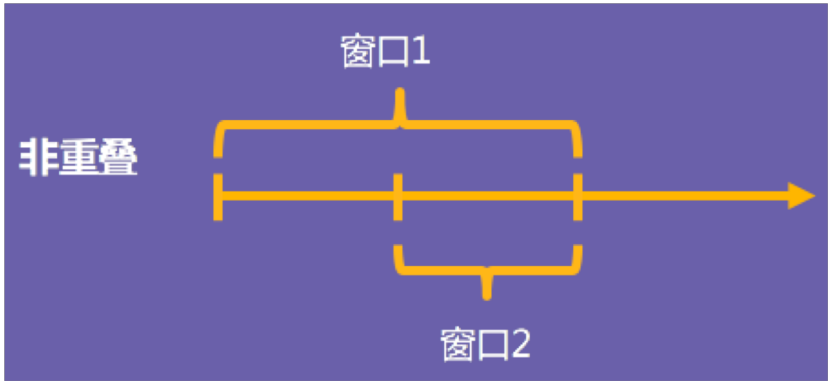
AIOps场景建设 - 报警压缩 - 相关性和根因

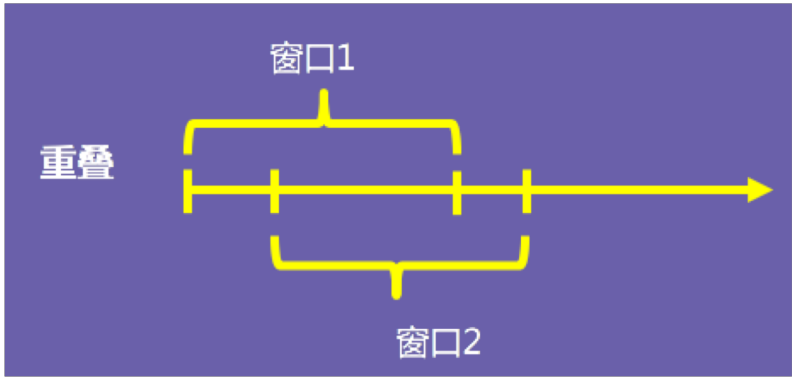
- 因为我们基于报警之间在同一时间窗口内的共生关系来进行报警压缩， 所以需要时间轴划分为若干窗口。
- 我们根据经验和数据观察暂时将窗口长度定为十分钟。
- 划分的方式有两种重叠和非重叠两种形式。

非重叠形式为窗口长度和窗口移动步长相等。

重叠形式为移动步长小于窗口长度。

重叠形式相当于对于非重叠形式的抽样。





AIOps场景建设 - 报警压缩 - 相关性和根因

假设检验推断相关性

- 采用假设检验进行相关性筛选。先假定两个独立报警之间不相关，然后依据不相关的假定，计算出它们之间实际共生发生的概率，如果概率过低说明假设不成立，得出两者相关的推断。
- 设定概率阈值， 筛选出相关的独立报警两两组合。通过报警之间的两两关联，我们可以构造出一个概率图。大部分独立报警都不在概率图中，也就是和任何其他报警都无时间上的相关关系。对于这样的报警我们不做压缩。在概率图中的独立报警种类数目虽然只占10%左右，但是在报警总数上占70%左右，所以在概率图中的报警有较大的压缩空间。
- 假定不相关的条件下的概率计算公式如下：

$$P_{N_c>n_c} = \sum_{k=n_c}^{k=\min(n_a,n_b)} \frac{\binom{N}{k}\binom{N-k}{n_a-k}\binom{N-n_a}{n_b-k}}{\binom{N}{n_a}\binom{N}{n_b}}$$

$$\binom{a}{b} = \frac{a!}{b!\,(a-b)!}$$

AIOps场景建设 - 报警压缩 - 相关性和根因

假设检验推断相关性

- 总共有1000个时间窗口的条件下，我们依照不同的逻辑假设计算了对应的不相关概率。
- 从下面的计算结果可以看出， 虽然我们的假设检验是针对两两报警组合的检验，但是对于更复杂的报警组合关系，依然有很好的效果。

1. $A \perp B$
总窗口数 N = 1000
报警A发生次数 $N_A = 100$
报警B发生次数 $N_B = 100$
A和B共生次数的期望值 $E(N_C) = 10$

$P_{N_c > n_c} \approx 0.555 > 0.01$ **无法拒绝A和B无关的假设**

2. $A \rightarrow B$
总窗口数 N = 1000
报警A发生次数 $N_A = 100$
报警B发生次数 $N_B = 100$
A和B共生次数 $N_C = 100$

$P_{N_c > n_c} \approx 1.57 * 10^{-140} < 0.01$ **拒绝A和B无关的假设**

AIOps场景建设 - 报警压缩 - 相关性和根因

假设检验推断相关性

3. $A_1 \wedge A_2 \rightarrow B$, $A_1 \perp A_2$

总窗口数 N = 1000

报警A_1发生次数 $N_{A_1} = 100$

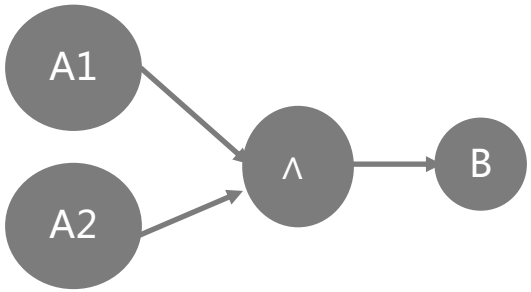
报警A_2发生次数 $N_{A_2} = 100$

A_1和A_2共同发生的次数的期望 $E(N_{A_1 \wedge A_2})=10$

报警B的发生次数 $N_B = 10$

A_1和B的共同发生次数 $E(N_C) = 10$

$P_{N_c > n_c} \approx 6.57 * 10^{-11} < 0.01$ 拒绝A_1和B无关的假设



4. $A_1 \wedge A_2 \rightarrow B$, $A_1 \perp A_2$

总窗口数 N = 1000

报警A_1发生次数 $N_{A_1} = 100$

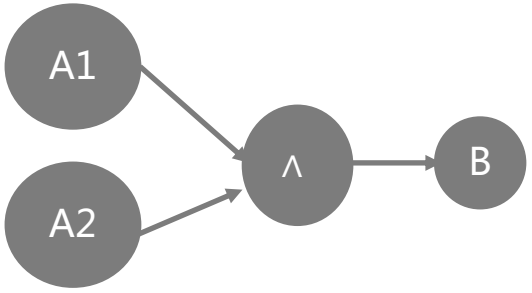
报警A_2发生次数 $N_{A_2} = 100$

A_1和A_2共同发生的次数的期望 $E(N_{A_1 \wedge A_2})=10$

报警B的发生次数 $N_B = 190$

A_1和B的共同发生次数 $E(N_C) = 100$

$P_{N_c > n_c} \approx 1.09 * 10^{-84} < 0.01$ 拒绝A_1和B无关的假设



AIOps场景建设 - 报警压缩 - 相关性和根因

报警ranking

- 在报警压缩的时候， 如果同窗口出现的一组独立报警之间存在很大的相关性，我们需要决定压缩哪一个报警，留下另一个呈现给用户。
- 我们首先采用对于独立报警进行因果排序。在压缩报警的时候，我们会压缩排名较低的报警（偏向于果的一侧），保留排名较高的报警（偏向于因的一侧）。
- 我们采用构造马尔科夫矩阵的方法来对于独立报警进行因果排序，具体如下：

1 定义报警两两之间的影响因子

在固定时间窗口下，两个报警之间相互影响因子相等。时间窗口总数为N, 报警A发生在n_a个窗口中，报警B发生在n_b个窗口中，两者共同发生的窗口个数为n_c。

AB影响因子$F_{AB} \stackrel{\text{def}}{=} \frac{A\text{和}B\text{共同发生的概率}}{\text{A发生的边缘概率}\times B\text{发生的边缘概率}} = \frac{P(AB)}{P(A)P(B)}$

2 构造影响因子矩阵

影响因子矩阵为一个N*N的对称正定矩阵，N为在前一部分的概率图中的独立报警数目。若报警A和B之间存在相关关系，则A行B列的元素为F_{ab}, B行A列的元素为F_{ba}。如果A和B之间无相关关系，则对应行列元素为0.

$$\begin{bmatrix} F_{11}F_{12}F_{13}\cdots\cdots \\ F_{21}F_{22}F_{23}\cdots\cdots \\ F_{31}F_{32}F_{33}\cdots\cdots \\ \cdots\ \cdots\ \cdots\ \cdots\cdots \\ \cdots\ \cdots\ \cdots\ \cdots\cdots \end{bmatrix}$$

AIOps场景建设 - 报警压缩 - 相关性和根因

报警ranking

3 从影响因子矩阵到马尔科夫矩阵

我们将第二部的影响因子矩阵的每一列除上每一列的和， 是矩阵的每一列的和都为1，就得到了一个马尔科夫矩阵。

4 利用稳定状态对于警报排序

因为我们构造的马尔科夫矩阵代表了警报之间的相互影响，所以稳定状态向量上的每一个元素代表了其对应警报在整个概率图中的因果影响力。在稳定状态数值越大的警报，我们认为其越接近果。。

5 马尔科夫矩阵的稳定状态

设上一步得到的马尔科夫矩阵为A。 马尔科夫矩阵存在一个性质， 对于任意一个非负初始向量u_0。马尔科夫矩阵不断与该向量相乘，会收敛于一个稳定状态向量u_s。且u_s和u_0无关。也就是存在u_s，对于任意u_0，都有

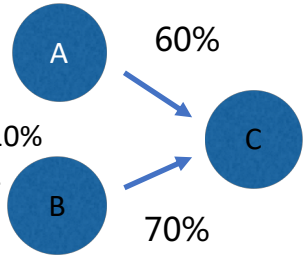
$$\lim_{k \to +\infty} A^k u_0 = u_s$$

AIOps场景建设 - 报警压缩 - 相关性和根因

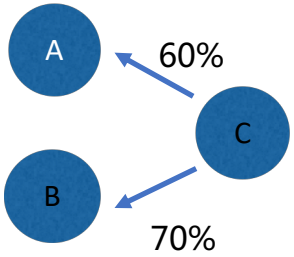
报警ranking

- 我们认为在我们的马尔科夫矩阵排序中，分数比较高的警报更可能是因果关系中的果，而分数比较低的警报更可能是因。
- 这里列出了两个迷你例子，在第一个例子中我们设定A和B是C发生的原因，在第二个例子中我们设定C是A和B的发生原因。
- 在第一个例子中，最为结果的C的分数最高，而在第二个结果中作为共因的C的分数最低，一定程度上支持了我们的推断。

- 假设A和B是C发生的原因
- A发生的时候在60%的情况下C发生
- B发生的时候在70%的情况下C发生
- A和B之间相互独立，各自发生的概率为10%
- 其对应的马尔科夫矩阵和稳态向量如下



- 假设C是A和B发生的原因， 在给定C发生的条件下，A和B的发生独立。
- C发生的时候在60%的情况下A发生
- C发生的时候在70%的情况下B发生
- C发生的概率为10%
- 其对应的马尔科夫矩阵和稳态向量如下



马尔科夫矩阵	A	B	C
A	0.634213	0.0603785	0.26087
B	0.0634213	0.603785	0.304348
C	0.302366	0.335836	0.434783

A	0.311629
B	0.327234
C	**0.361138**

马尔科夫矩阵	A	B	C
A	0.454545	0.291667	0.333333
B	0.272727	0.416667	0.333333
C	0.272727	0.291667	0.333333

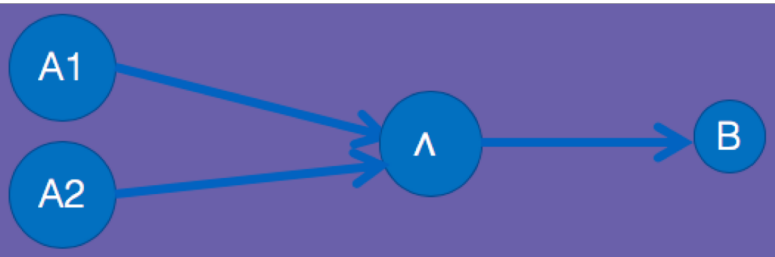
A	0.363216
B	0.339616
C	**0.297168**

AIOps场景建设 - 报警压缩 - 相关性和根因

模型训练和预测

- 在报警组合筛选以及报警排序中，我们发掘的都是根据两两报警之间的共生关系。但是仅仅依靠两两共生关系无法满足我们的压缩的功能。
- 在下面的例子中，报警A_1和报警A_2共同发生时会导致报警B的发生，这也反应在列出的迷你数据集中，$P(B=1|A_1=1,A_2=1)=1$。
- 但是如果我们只考虑两两的共生关系，那么$P(B=1|A_1=1)$= $P(B=1|A_2=1)=0.4<0.5$。两个条件概率都不足以让我们预测出报警B的发生进而进行压缩。
- 因此我们需要引入预测模型来对多个报警之间的共生关系进行学习。

时间窗口index	A_1	A_2	B
0	1	0	0
1	1	0	0
2	1	0	0
3	1	1	1
4	1	1	1
5	0	1	0
6	0	1	0
7	0	1	0
test	1	1	?



$P(B=1|A_1=1)=0.4$ $P(B=1|A_2=1)=0.4$

$P(B=1|A_1=1,A_2=1)=1$

AIOps场景建设 - 报警压缩 - 相关性和根因

模型训练和预测

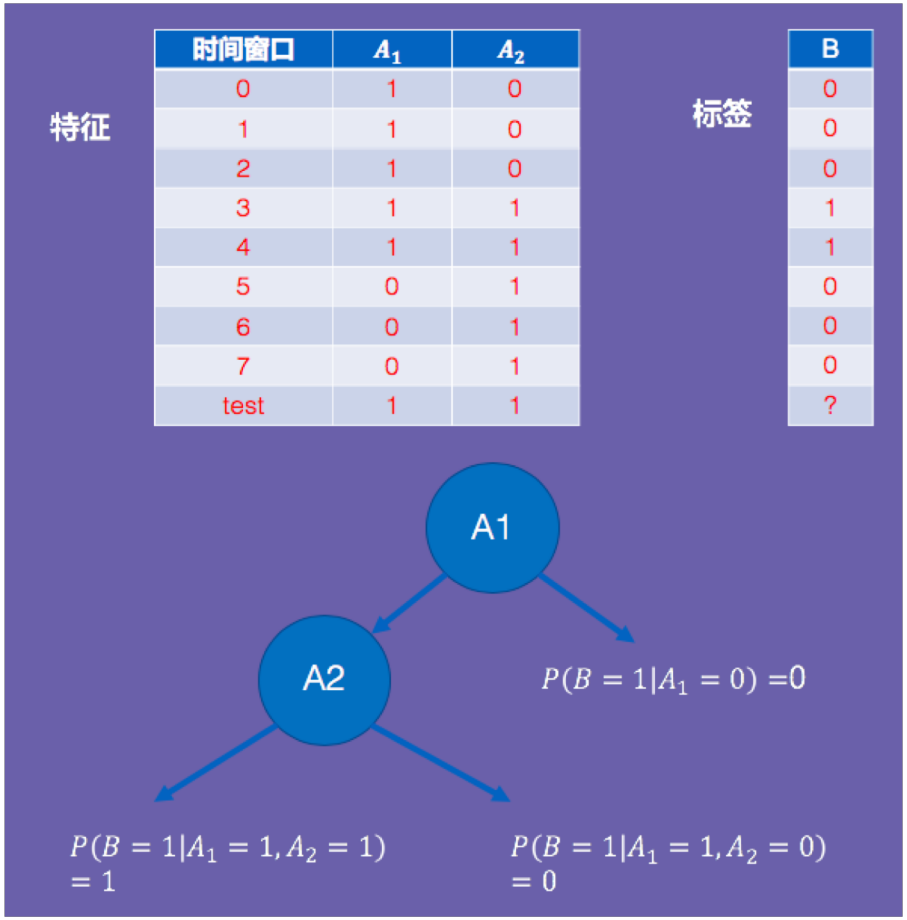
- 我们从因果排序中排名最低的警报开始， 注意训练判断该警报在某一时间窗口内会否发生的模型。模型的因变量为该警报的发生情况（1 为发生， 0 为没发生）。其他报警的发生情况作为特征。 只有符合以下两个条件的警报才有资格成为特征， 一是和因变量警报两两相关（在概率图中是因变量警报的邻居）， 二是在因果排序中排名比因变量警报高。
- 因为所有项目以及标签都是01变量，很适合决策边界具有阶梯型（锯齿形）的学习器，比如决策树。但是由于数据集噪音较大，而决策树又很容易产生过拟合。我们选择以决策树为自学习器的随机森林模型来做训练预测。
- 从因果排名中的低到高， 我们对于每一个警报都会训练一个预测模型。
- 在预测时， 我们在当前窗口中从排名最低的警报开始预测， 如果预测结果为不会发生，则不进行压缩，如果预测结果为会发生，则进行压缩，并将该警报归入其当前发生的邻居中排名最高的警报类下。

特征

时间窗口	A_1	A_2
0	1	0
1	1	0
2	1	0
3	1	1
4	1	1
5	0	1
6	0	1
7	0	1
test	1	1

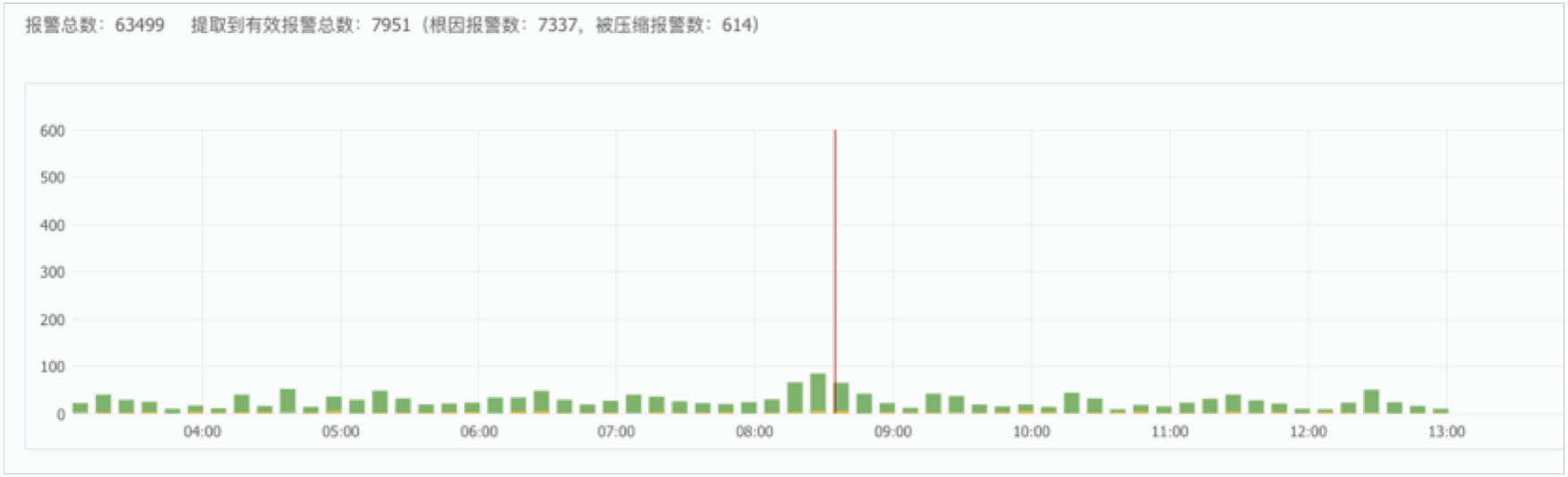
标签

B
0
0
0
1
1
0
0
0
?



AIOps场景建设 - 报警压缩 - 相关性和根因

模型测试样例



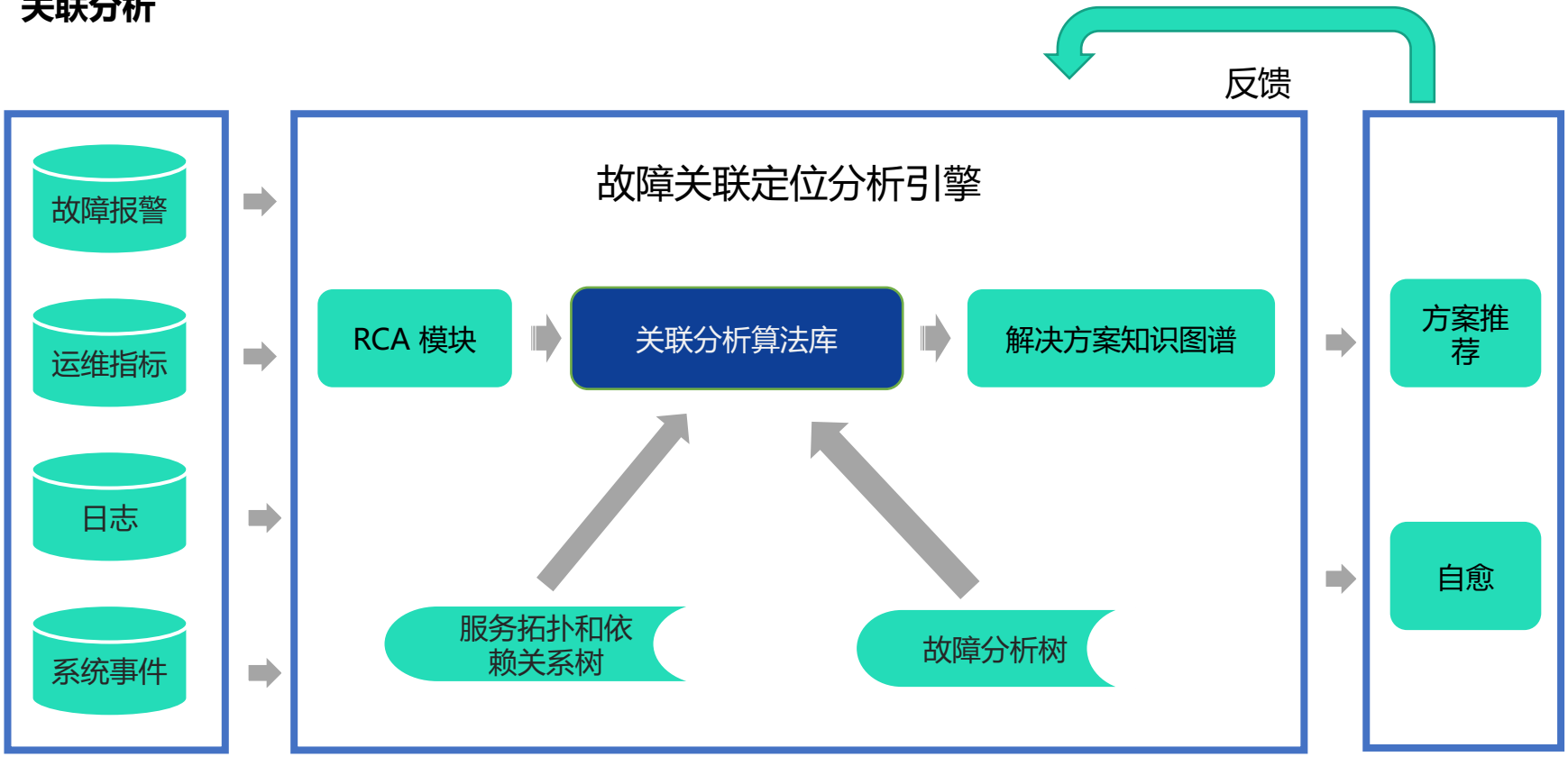
报警时间	报警内容
2018-08-02 03:03:36	"Interface GigabitEthernet1/0/23, changed state to down" ==> INTERFACE GIGABITETHERNET#/#/# CHANGED STATE TO DOWN
2018-08-02 03:03:53	"Outbound Speech Call Cleared or Failed, Disconnect Cause: 11 (Peer Address: 018518365075)" ==> OUTBOUND SPEECH CALL CLEARED OR FAILED DISCONNECT CAUSE PEER ADDRESS
2018-08-02 03:04:03	OSPF Link State Request Packet Retransmitted to 10.3.128.254 (10.197.132.229) ==> OSPF LINK STATE REQUEST PACKET RETRANSMITTED TO IP IP
2018-08-02 03:04:54	hh3cfanfailure: Fan failure ==> HH#CFANFAILURE FAN FAILURE

AIOps场景建设 - 故障根因分析和RCA

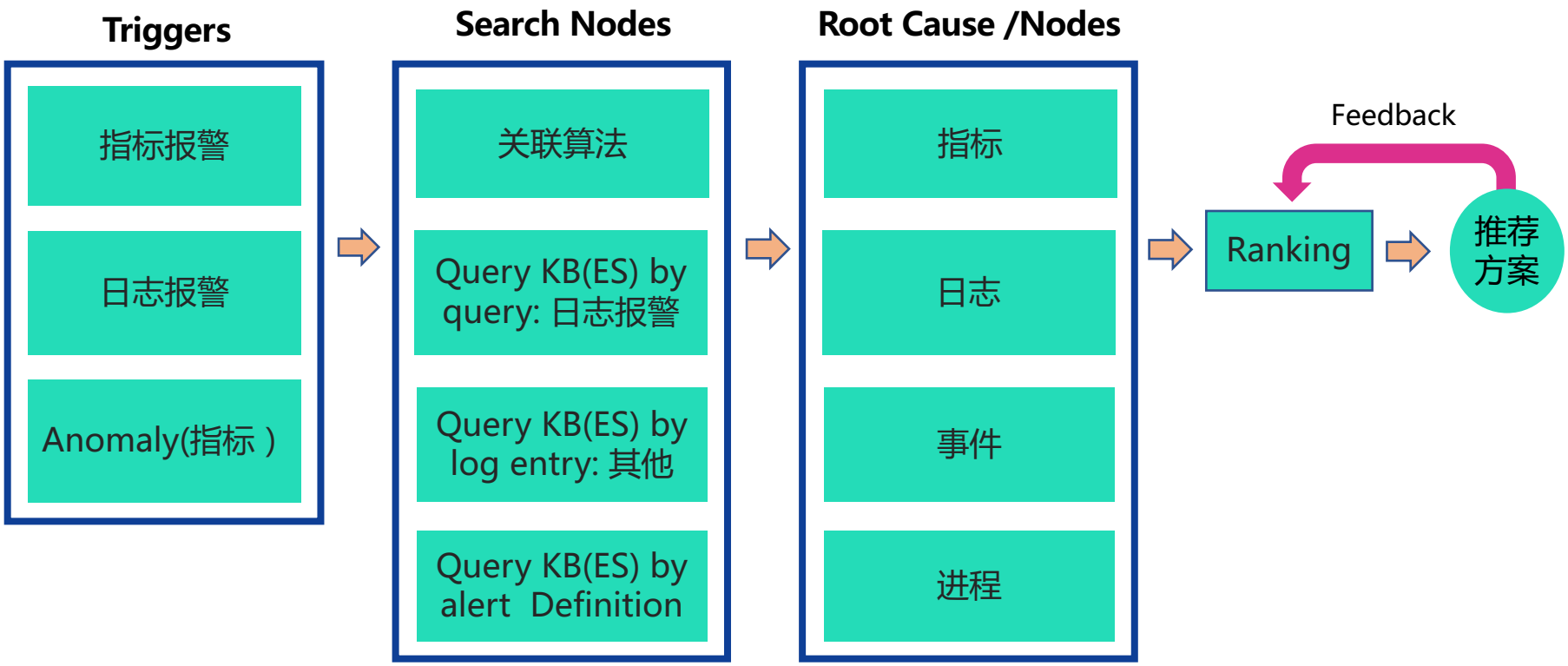
基于关联分析的根因定位

AIOps场景建设 - 故障根因分析和RCA

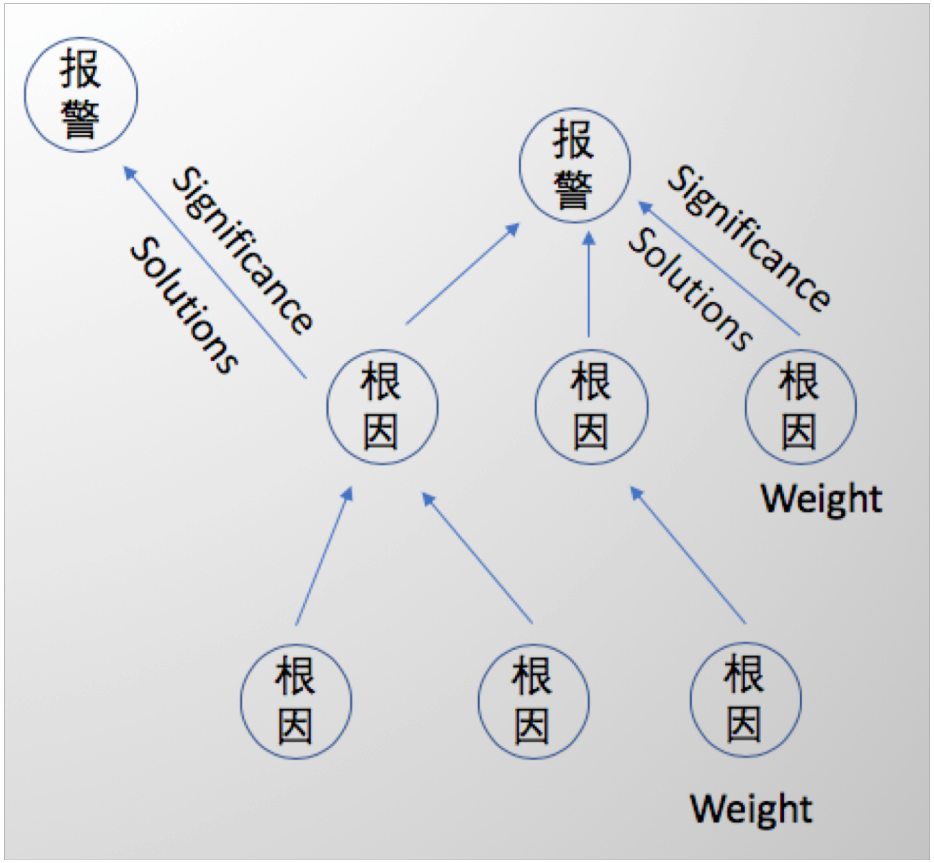
关联分析



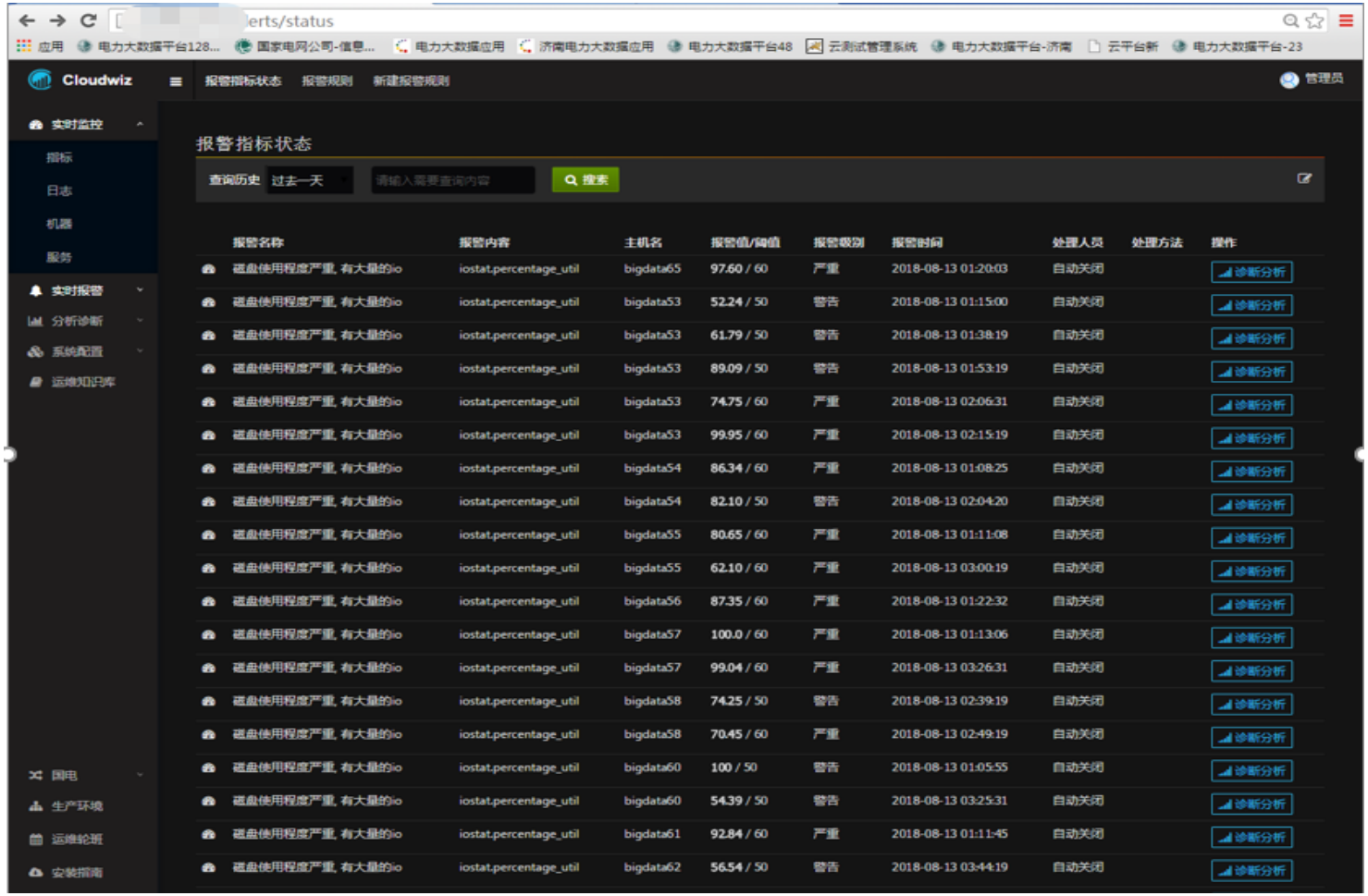
AIOps场景建设 - 故障根因分析和RCA



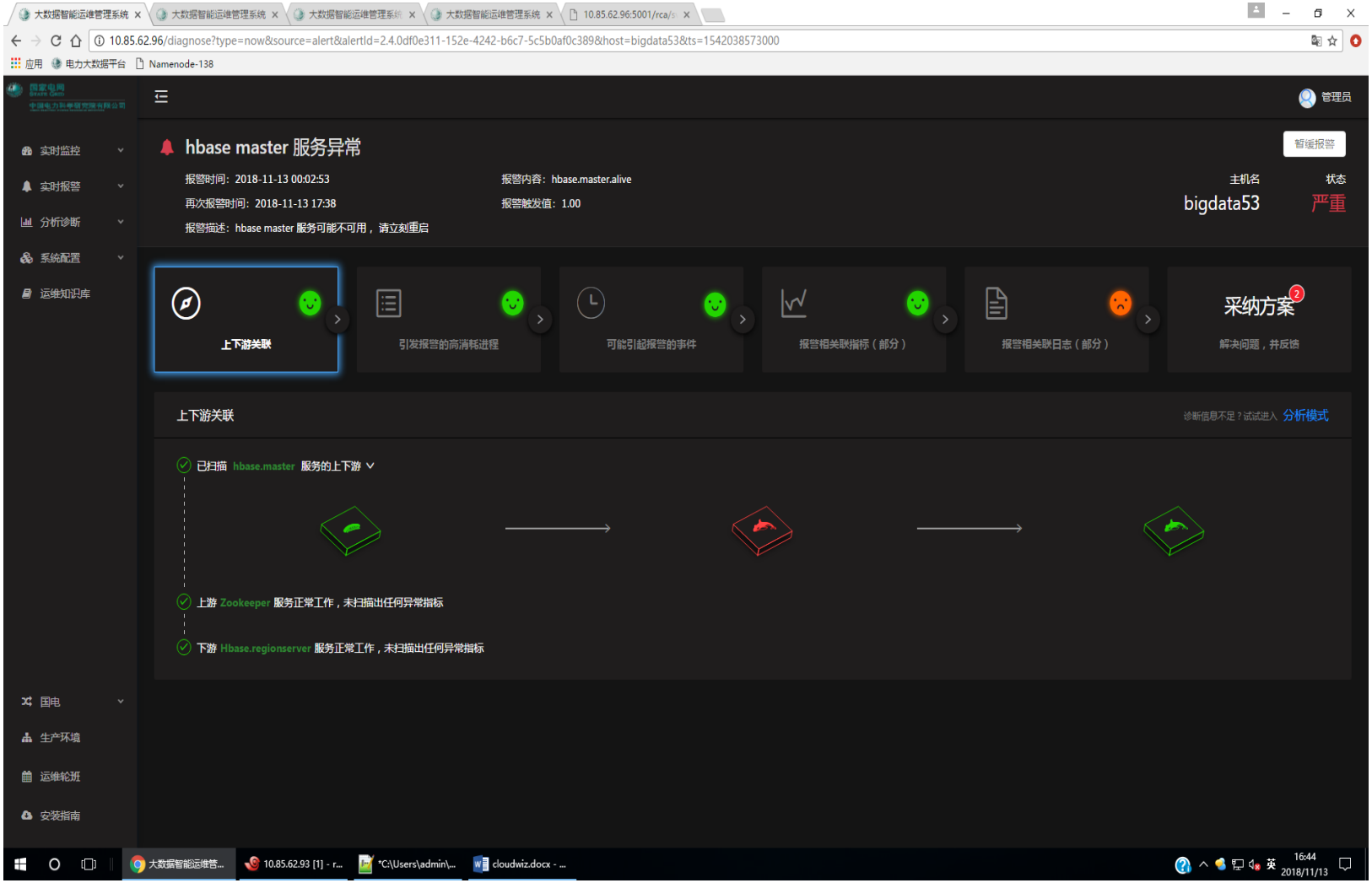
AIOps场景建设 - 故障根因分析和RCA



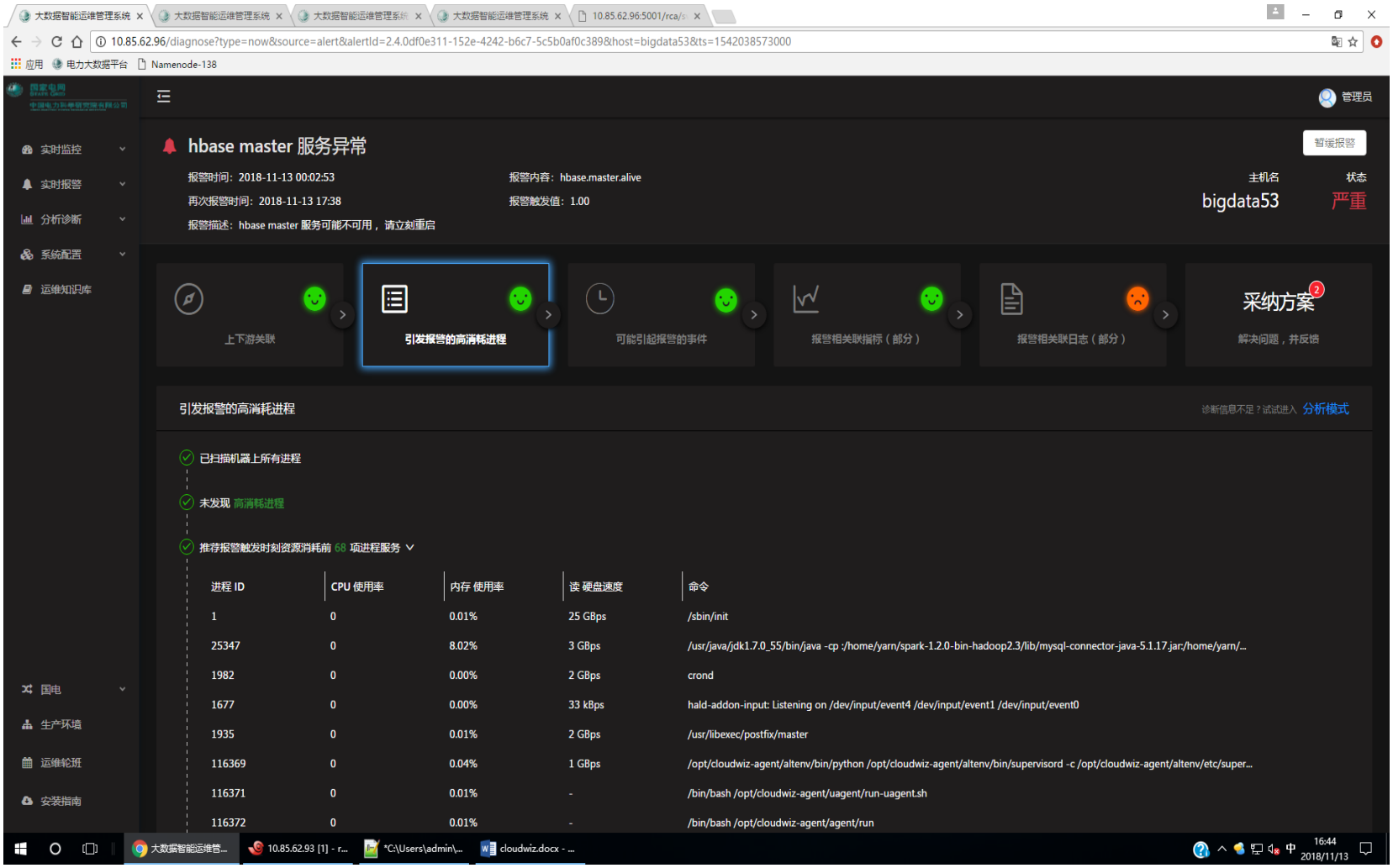
AIOps场景建设 - 故障根因分析和RCA



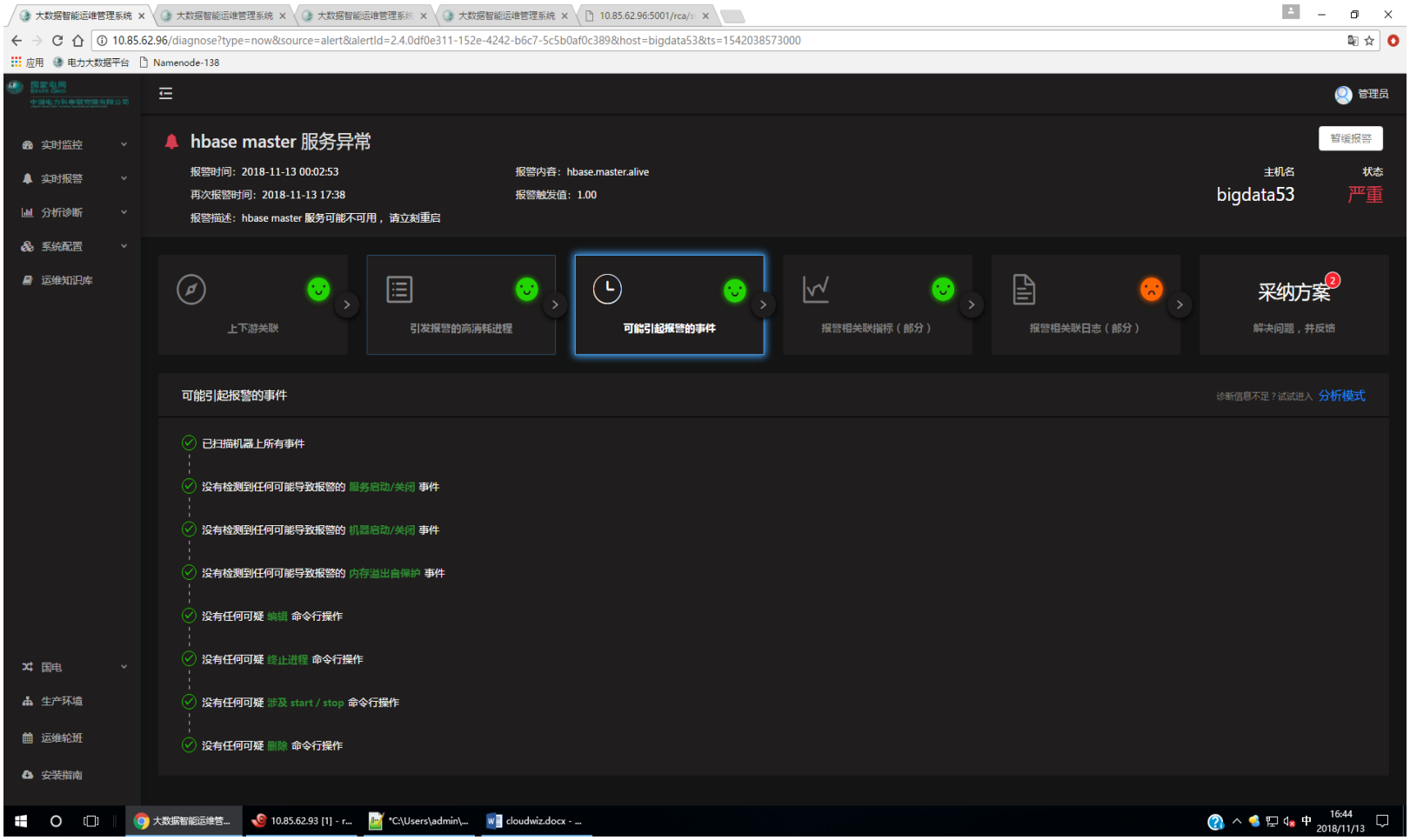
AIOps场景建设 - 故障根因分析和RCA



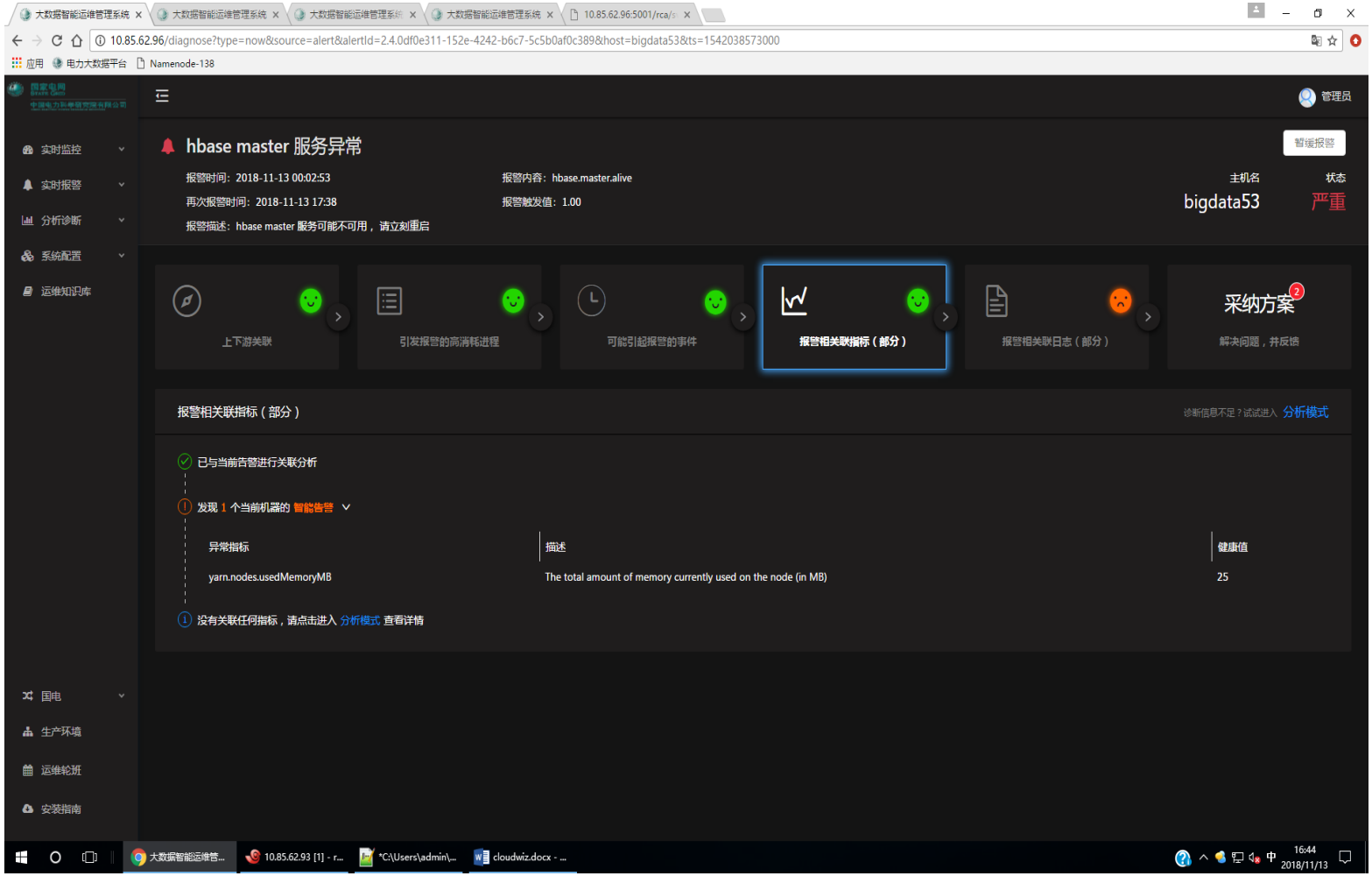
AIOps场景建设 - 故障根因分析和RCA



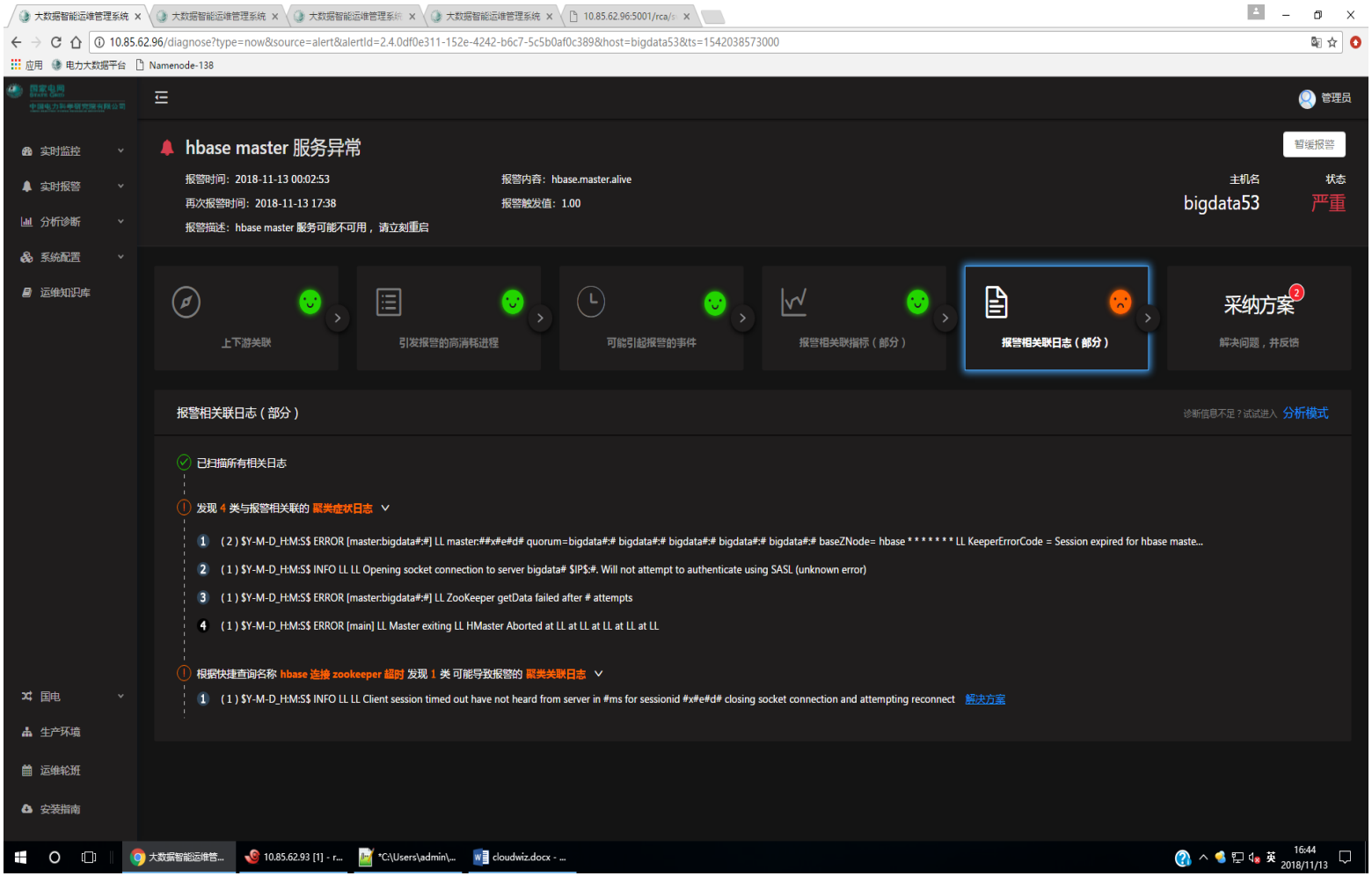
AIOps场景建设 - 故障根因分析和RCA



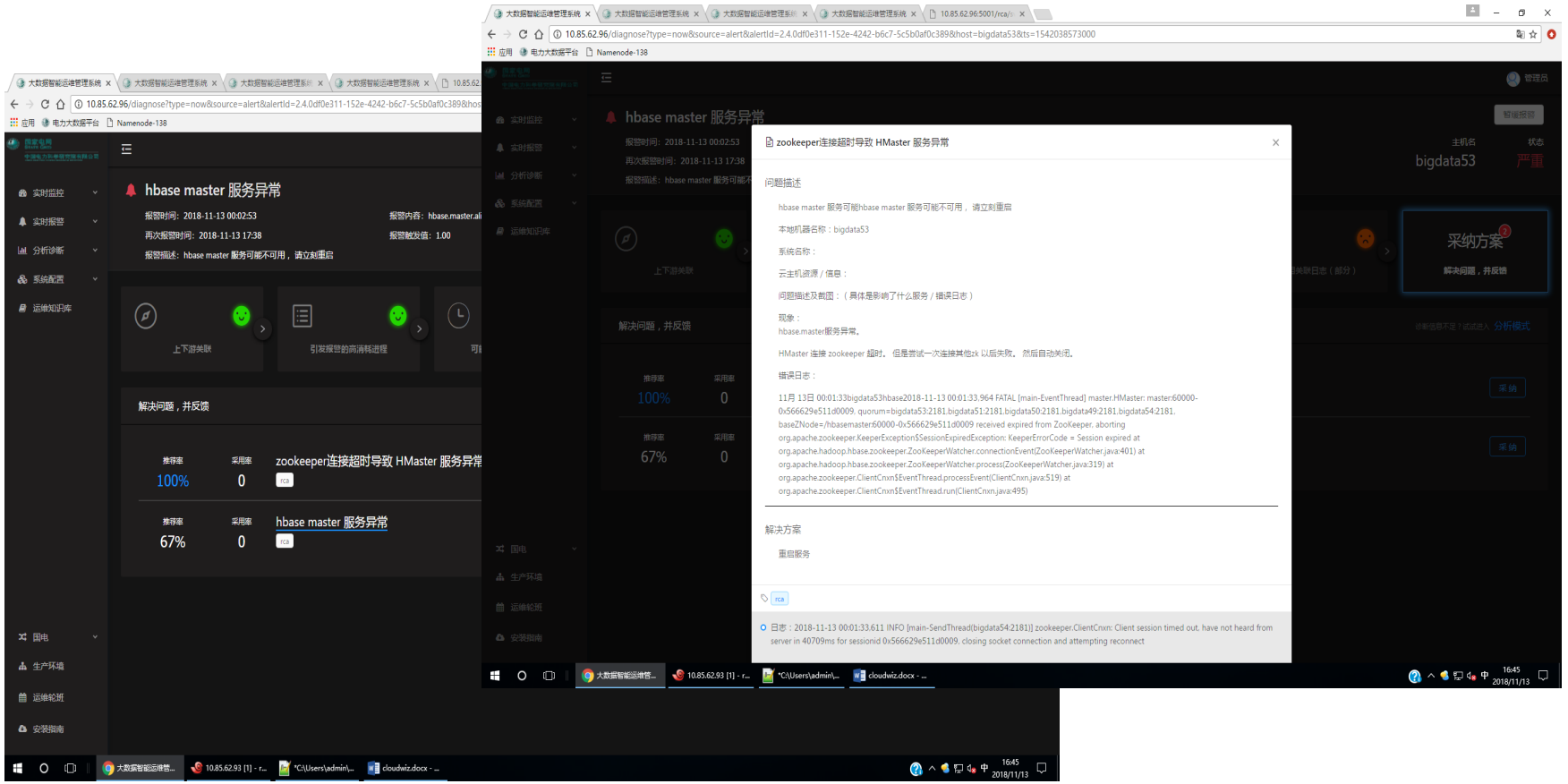
AIOps场景建设 - 故障根因分析和RCA



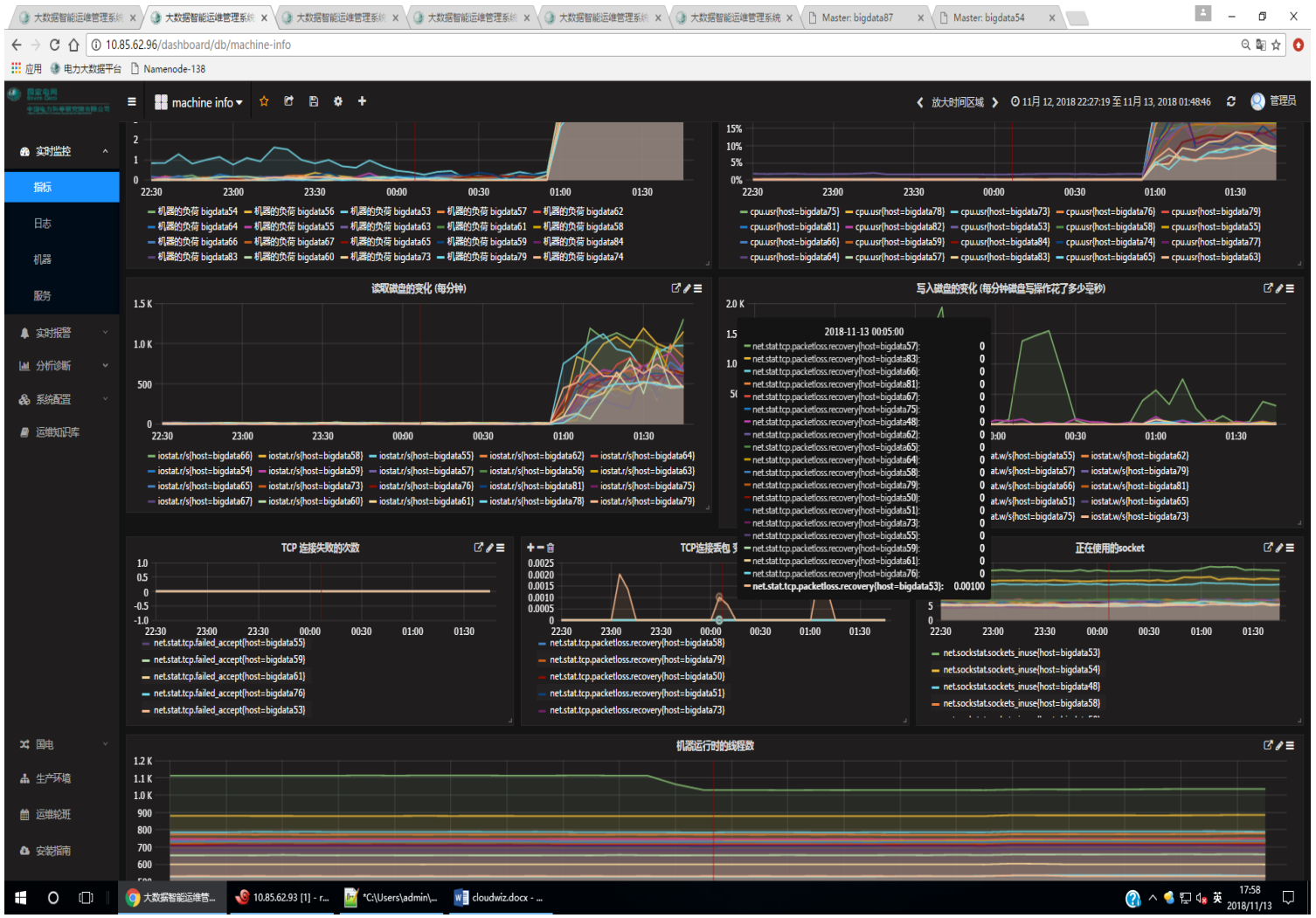
AIOps场景建设 - 故障根因分析和RCA



AIOps场景建设 - 故障根因分析和RCA



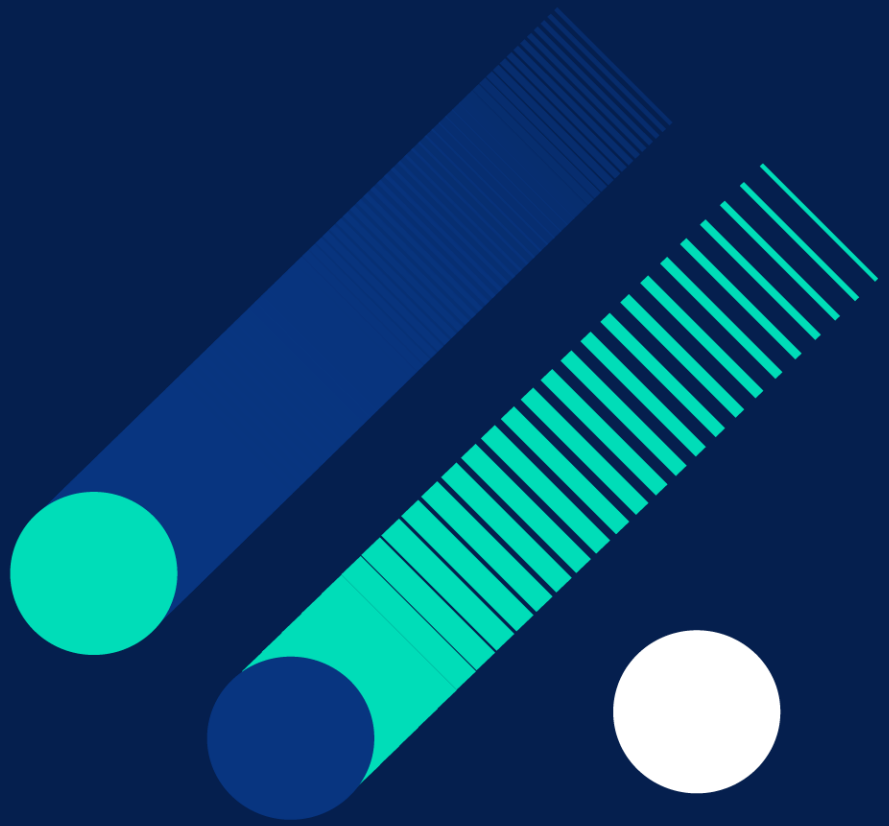
AIOps场景建设 - 故障根因分析和RCA



GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE



关注公众号获得
更多案例实践