DOIS

2018 | 中国·北京站

DevOps 落地，从这里开始

DevOps 国际峰会

暨 DevOps 金融峰会

指导单位：云计算开源产业联盟 OpenSource Cloud Alliance for industRy, OSCAR

主办单位：DevOps时代　高效运维社区 GreatOPS Community

2018年6月29日-30日

地址：北京悠唐皇冠假日酒店

DOIS

AIOPS在腾讯的探索和实践

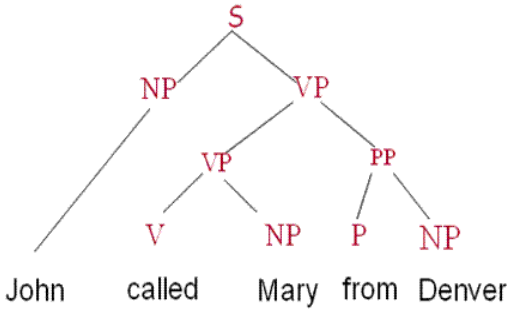
赵建春@腾讯社交网络运营部

目录

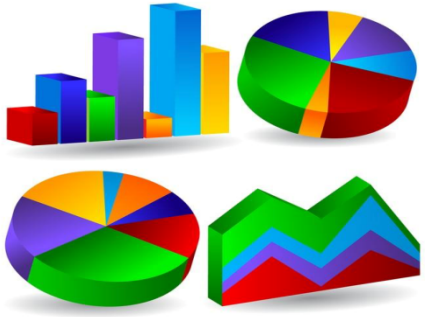
从自然语言处理的一个小故事开始

DOIS

- 二十世纪70年代，基于**规则**的句法分析逐渐走到了尽头。
- 1974年左右贾里尼克等人在IBM提出了统计语音识别的框架结构。
- 2005年， Google**基于统计方法**的翻译系统全面超过**基于规则方法**的SysTran翻译系统，基于规则方法固守的最后一个堡垒被拔掉了。



VS



二十世纪70年代前那些坚守基于规则句法分析的科研和学术人员，注定是失落的。

对这个故事的思考和借鉴

DOIS

- 语言的复杂与多样性和我们运维系统有相似性
- 我们的很多运维经验，其本质是一组组规则
- 我们的运维系统里，有大量的自动处理规则
- 规则好用，易理解，但往往有“场景”遗漏
- 当处理海量数据时，规则往往显得力不从心
- 规则，是我们的经验，也是我们的负担
- AIOPS，是DEVOPS的补充，是对运维规则的AI化

从大量输入中总结出准确预测的规律（模型）

DOIS

$$y = f(x) = ax + d$$

$$y = f(x) = ax^2 + bx + d$$

$$y = f(x_1, x_2, x_3) = ax_1 + bx_2 + cx_3 + d$$

$$a_1^{(2)} = f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)})$$
$$a_2^{(2)} = f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)})$$
$$a_3^{(2)} = f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)})$$

$$h_{w,b}(x) = a_1^{(3)} = f(W_{11}^{(2)}a_1^2 + W_{12}^{(2)}a_2^2 + W_{13}^{(2)}a_3^2 + b_1^{(2)})$$

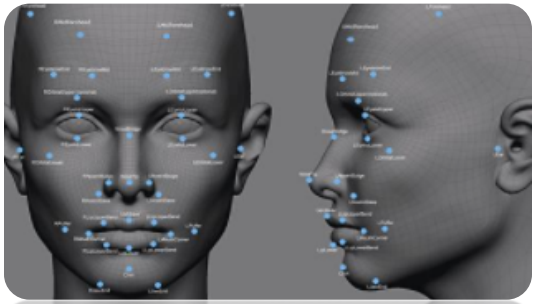
x, x_1, x_2, x_3是我们的输入

y是期望的输出

a,b,c,d（W，b）是我们要求解的参数

f是转换函数（公式、函数、算法、模型...）也就是我们要找的“规律”

数值型预测
0/1型预测
概率型预测等



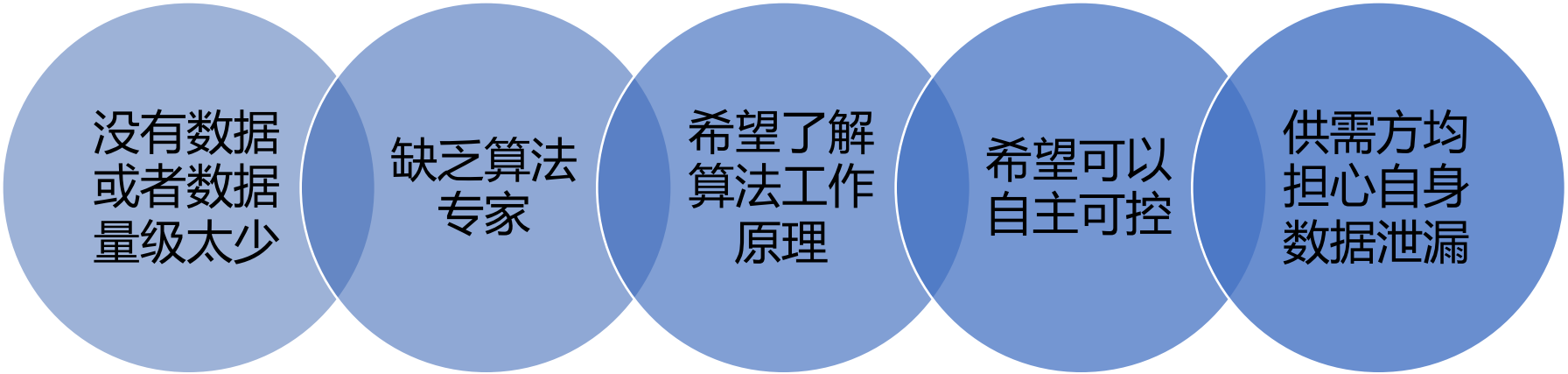
易于接入AI的运维场景的特点

DOIS

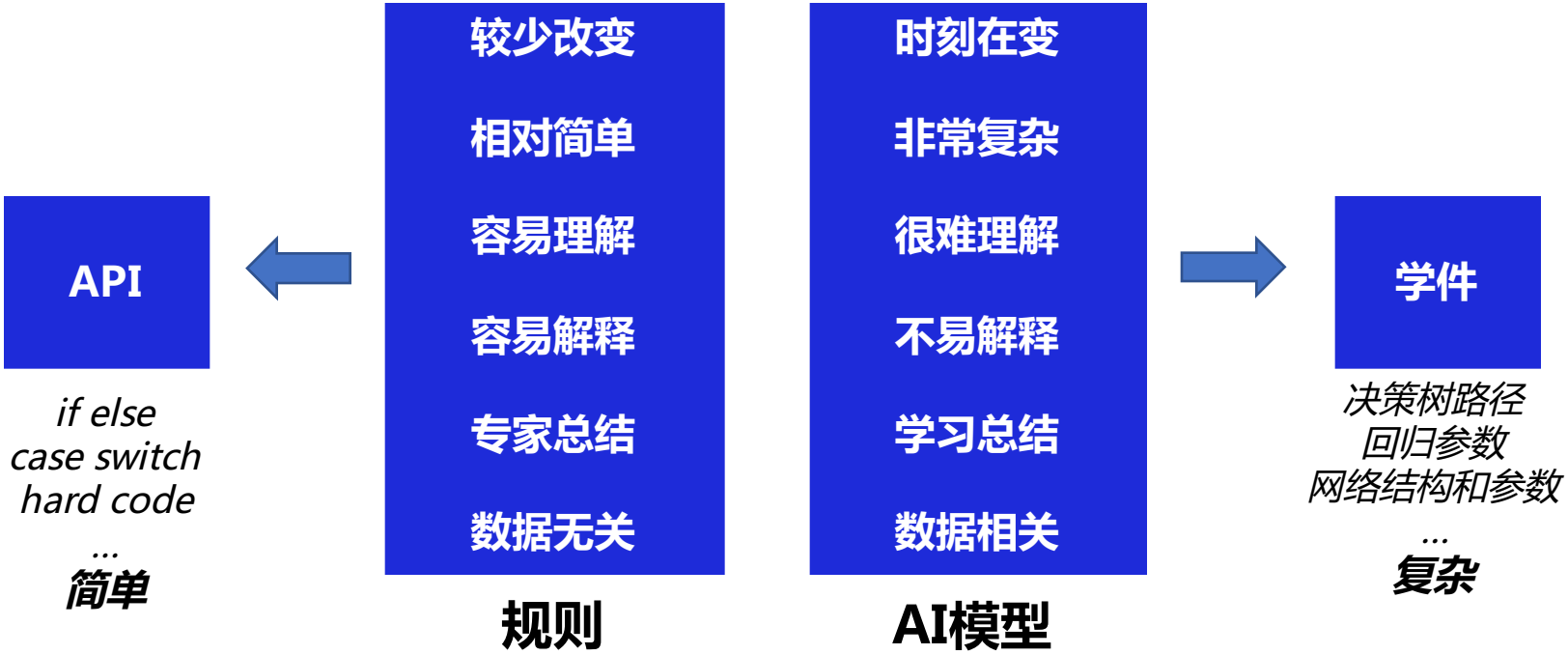
- 足够的**数据**量级，重复次数高
- **特征**（因素）较齐备
- **特征**质量高（提取、清洗容易等）
- **正负样本**易抽取
- 积累了**正负样本**库（标注）
- 有持续的**正负反馈……**

- 数据+算法+训练->模型
- 数据是基础

接入AIOPS的困难



AI模型与规则的差异


API
if else
case switch
hard code
...
简单
较少改变
相对简单
容易理解
容易解释
专家总结
数据无关
规则
时刻在变
非常复杂
很难理解
不易解释
学习总结
数据相关
AI模型
学件
决策树路径
回归参数
网络结构和参数
...
复杂

DOIS

目录

学件是什么？

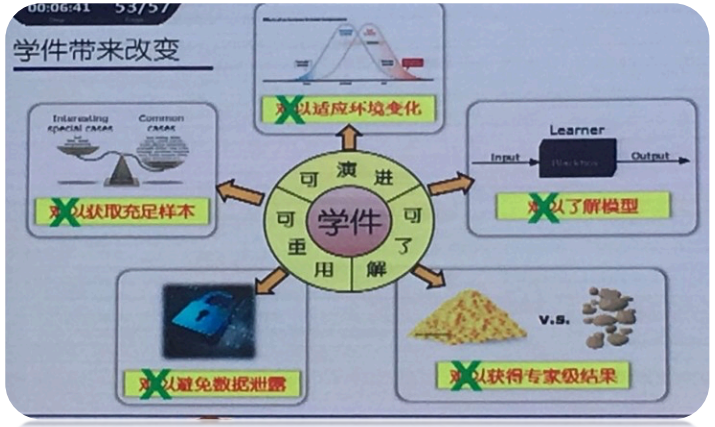
DOIS

学件（Learnware）= 模型（Model）+规约（Specification）

概念提出者：南京大学教授，国内机器学习领域领军人物周志华老师。

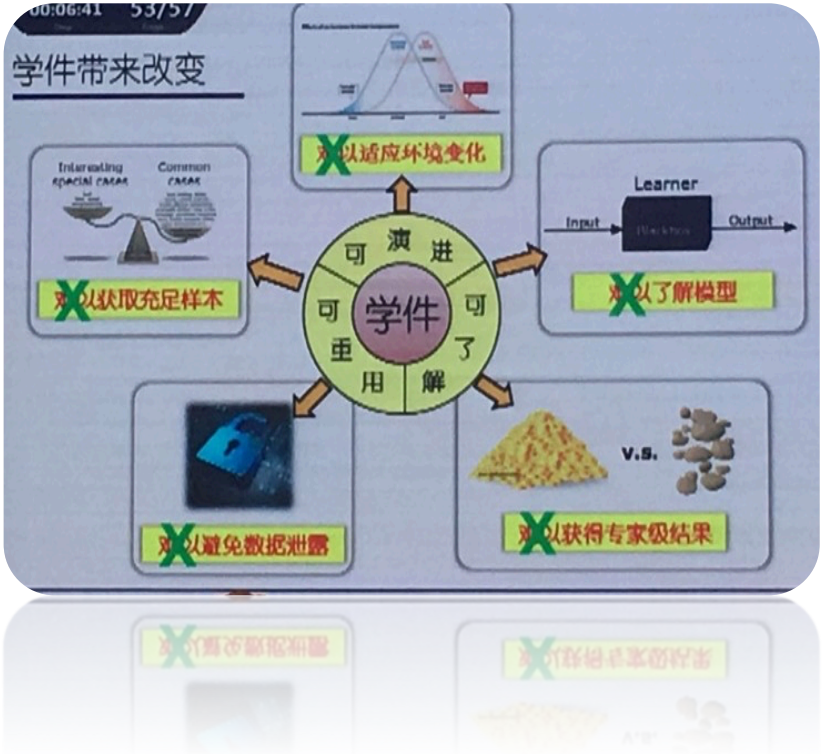
API / 规则

VS



学件

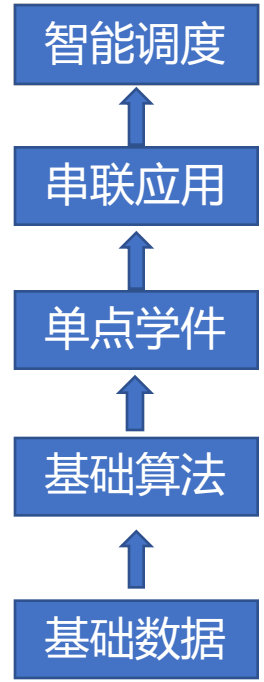
学件的特点和优势



- 可演进（修改、学习）
- 可了解
- 可重用
- 可不依赖数据
- 可不依赖专家

AIOPS能力框架

DOIS



多目标自调整AIOPS大脑

主场景均实现流程化免干预AI运维能力

串联AI运维应用如故障自愈

效率	质量	成本
AI运维组件(学件)库 智能预测 智能变更 问答机器人 变更机器人 建议机器人 ...	AI运维组件(学件)库 异常检测 异常预测 根因分析 舆情监控 ...	AI运维组件(学件)库 性能调优 成本分析 强化改进 碎片整理 容量管理 ...

运维开发框架及工具库

AI算法平台和算法库

离线计算 | 实时计算

元数据管理、数据仓库

数据清洗、ETL、特征工程

数据通道

数据采集如各类agent

数据上报标准，命名规范

AIOPS

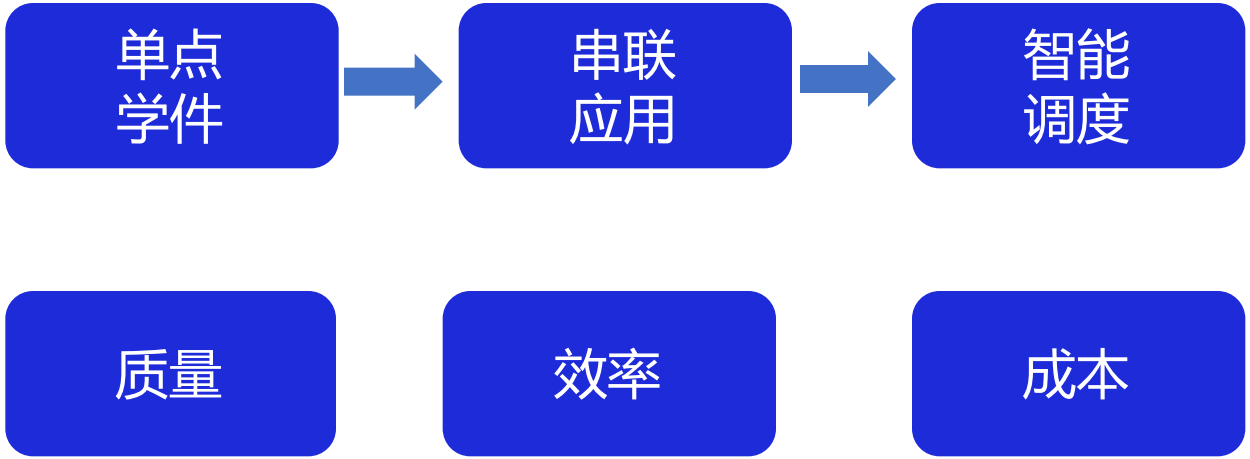
效率 ⟷ 质量 ⟷ 成本

AIOPS 五级分类

DOIS

- **一级**：• **尝试应用**：开始尝试应用AI能力，还无较成熟单点应用
- **二级**：• **单点应用**：具备单场景AI运维能力，初步形成供内部使用的学件
- **三级**：• **串联应用**：有由多个单场景AI运维模块串联起来的流程化AI运维能力
- **四级**：• **能力完备**：主要运维场景均已实现流程化免干预AI运维能力
- **五级**：• **终极AIOPS**：有中枢AI，可以在成本、质量、效率间从容调整，达到业务不同生命周期对三个方面不同的指标要求，实现多目标下的最优或按需最优

AIOPS是DEVOPS的补充，是对运维规则的AI化DOIS



DOIS

目录

单点案例
成本 — 内存存储智能降冷

内存存储智能降冷-背景

DOIS

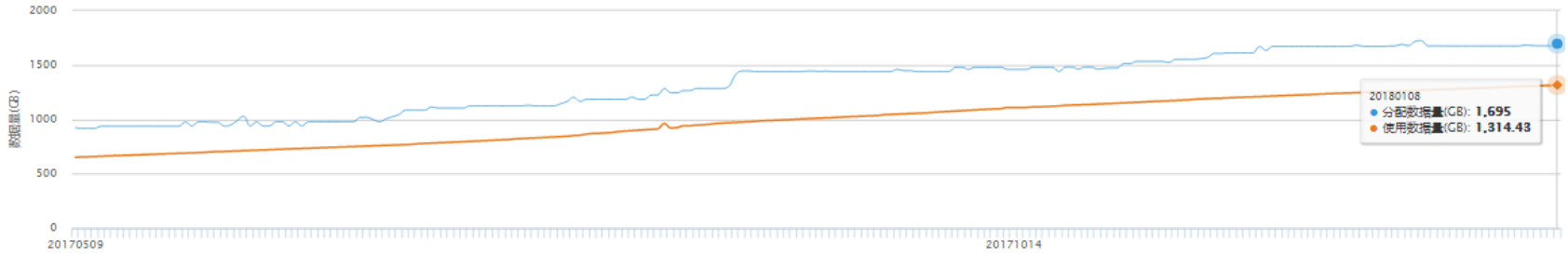
社交网络的业务特征

访问量大

延迟敏感

优先选择内存KV存储

随着业务发展，内存数据和成本迅速增加



存储内存智能降冷-方案

DOIS

- 生成数据特征--制作数据画像（抽样扫描业务数据）

KEY长度	记录大小	最后读时间	最后写时间	过期删除时间	周期热度变化	访问量	业务数据量	空查询率

- 专家经验打标分类

bid	over_3days	expire_in_7_rate	diff_rate	avg_denisty	used_g	empty_query_rate	class
[redacted]	0.24	1.00	0.00	82	172	1.00	0
[redacted]	0.62	1.00	0.29	11	168	0.57	0
[redacted]	0.00	0.94	0.00	154	198	0.84	0
[redacted]	0.83	0.94	0.53	27	2	0.87	0
[redacted]	0.02	0.93	0.05	866	221	0.75	0

- 生成分离策略--机器学习预测是否能够下沉

逻辑回归　随机森林

- 根据预测结果自动执行流程

流程更新时间	访问密度	流程状态	下沉时间(小时)	内存	SSD	下沉比	操作
2018-02-02 16:06:42	344.27	Sinking	72	50.6	2167	97.72	停止下沉 修改下沉时间 强制50%下沉

算法选择—LR和随机森林

DOIS

```
#使用逻辑回归预测
lr = LogisticRegression(C = 0.1,penalty = 'l1')
lr.fit(features_train,labels_train.values.ravel())
lr_y_pred = lr.predict(features_test.values)
cm = confusion_matrix(labels_test,lr_y_pred )
# 计算混淆矩阵的召回率
print("召回率: %.2f"%(cm[1,1]/(cm[1,0]+cm[1,1])))
# 计算混淆矩阵的准确率
print("准确率: %.2f"%((cm[0,0]+cm[1,1])/(cm[0,0]+cm[0,1]+cm[1,0]+cm[1,1])))
```

```
召回率: 0.82;
准确率: 0.81;
```

```
#使用随机森林预测
features_train, features_test, labels_train, labels_test =
train_test_split(features,labels,test_size=0.2,random_state=0) rf =
RandomForestClassifier(random_state=1, n_estimators=10,
min_samples_split=2, min_samples_leaf=1)
rf.fit(features_train,labels_train.values.ravel()) rf_y_pred =
rf.predict(features_test.values) cm =
confusion_matrix(labels_test,rf_y_pred) # 计算混淆矩阵的召回率 print("召回率:
%.2f"%(cm[1,1]/(cm[1,0]+cm[1,1]))) # 计算混淆矩阵的准确率 print("准确率:
%.2f"%((cm[0,0]+cm[1,1])/(cm[0,0]+cm[0,1]+cm[1,0]+cm[1,1])))
```

```
n_estimators=20, 召回
n_estimators=30, 召回
n_estimators=40, 召回
n_estimators=50, 召回
n_estimators=60, 召回
……
n_estimators=100, 召
```

采集数据特征

↓

特征选择

↓

特征数据转化（字符转数字，处理空字段）

↓

算法选择（LR,RF……）

↓

交叉验证，获取最佳参数（正则惩罚参数、树个数、节点数）

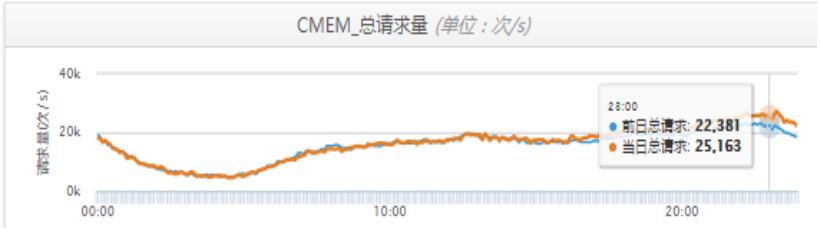
↓

混淆矩阵获取召回率、准确率

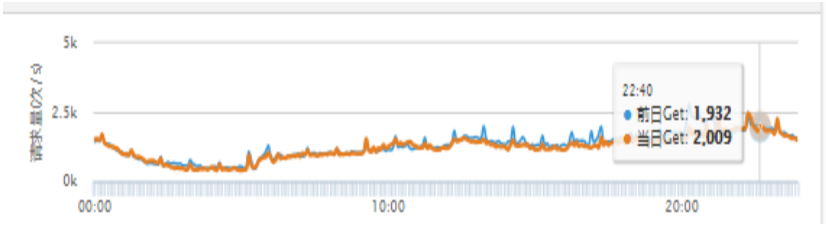
存储内存智能降冷-效果

DOIS

- 热数据和冷数据的访问比例10:1，下沉精准，对SSD没有造成访问压力

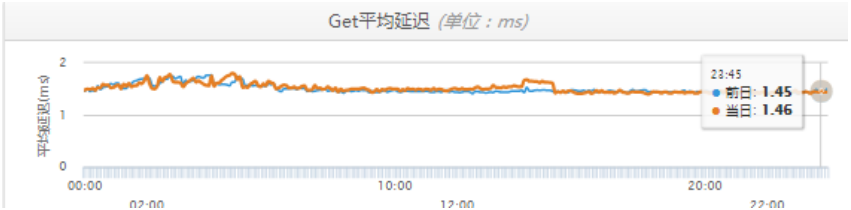


内存热数据访问量



SSD冷数据访问量

- 访问延迟没有明显变化



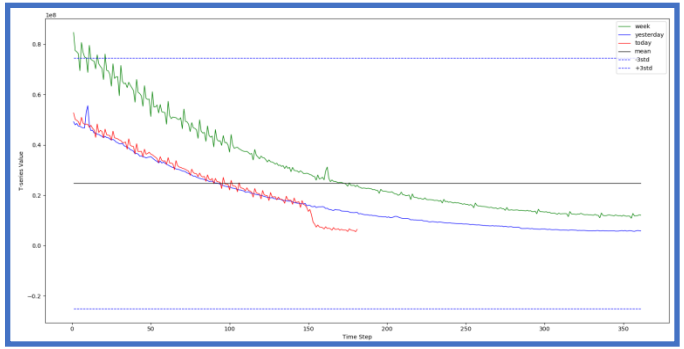
- 业务调用成功率正常，降冷对业务透明

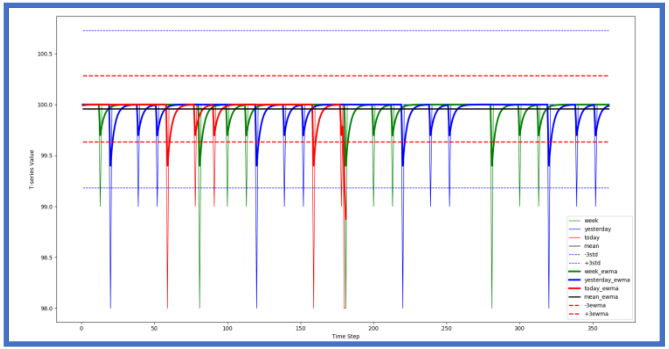
时间	调用成功数	调用失败数	调用成功率
2018-06-09 13:34:00	179500	0	100%
2018-06-09 13:33:00	177550	0	100%
2018-06-09 13:32:00	177600	2	100%

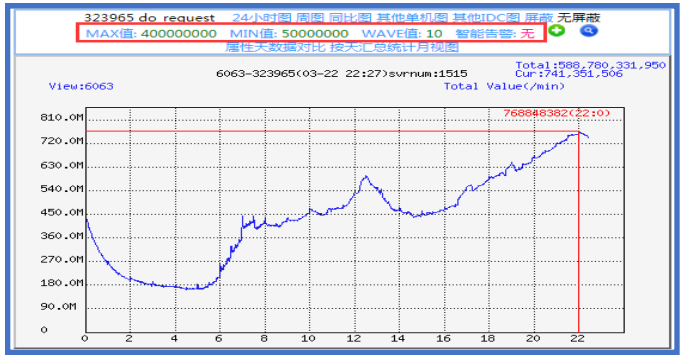
单点案例
质量 — 统一监控去阈值

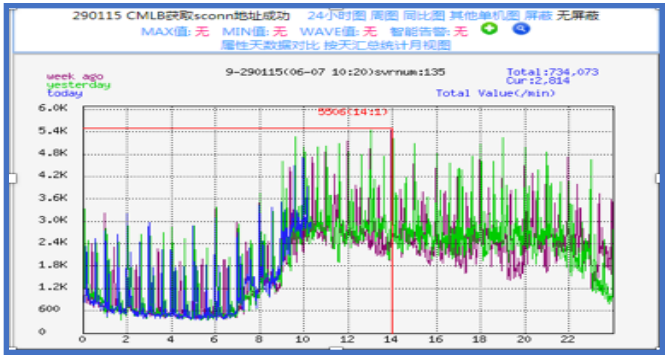
统一监控去阈值

DOIS





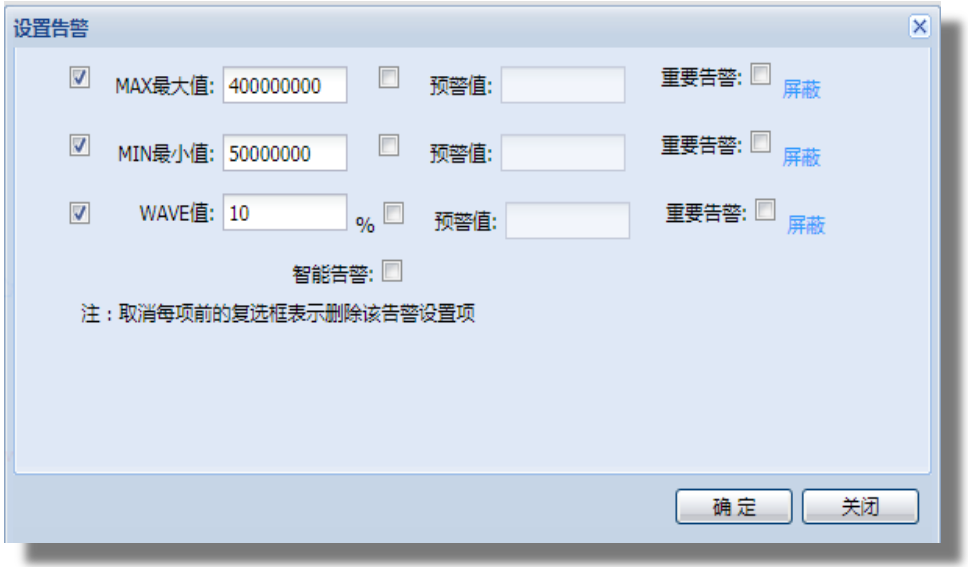


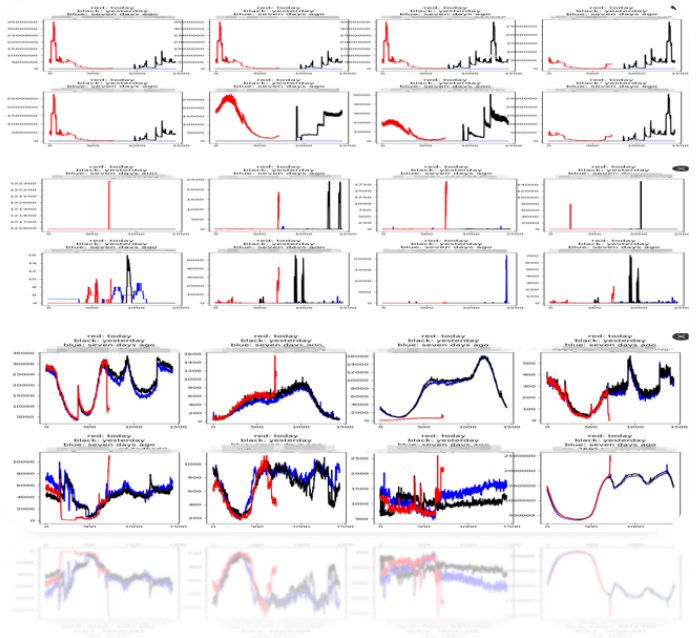


统一监控去阈值

DOIS

监控曲线千变万化，通过设置阈值方式费时费力，难以维护更新，容易误报漏报

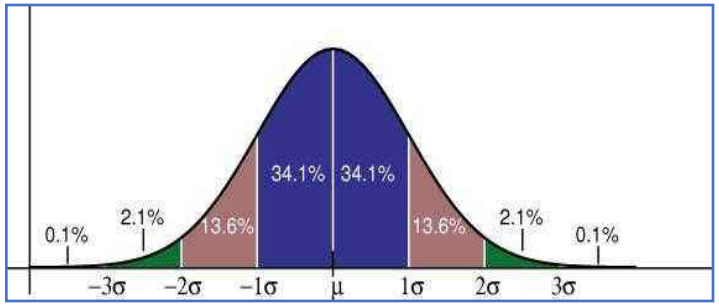


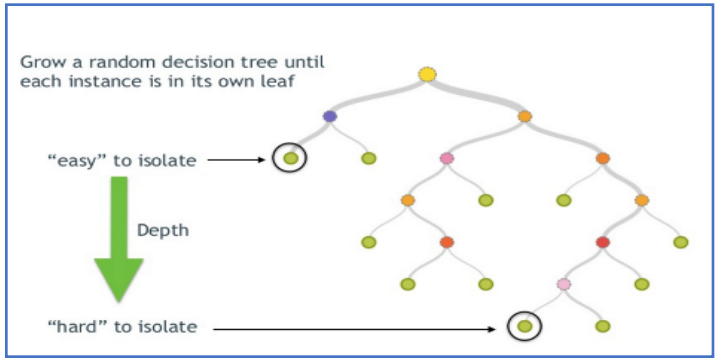


成功率监控

DOIS

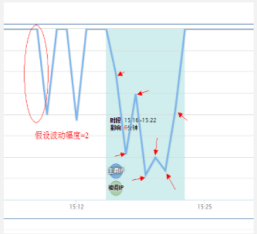
3sigma + 规则

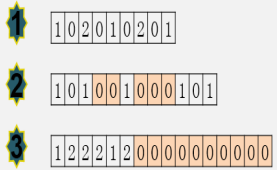




告警判断模型

DLP





1 折算值累计到达6触发告警

2 累计值小于3时，间隔2个正常点清空计数

3 累计值等于3时，间隔3个正常点清空计数

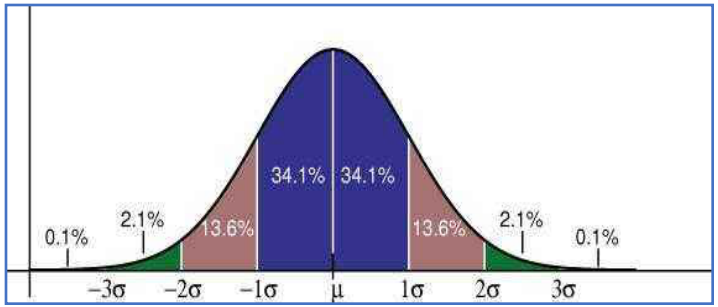
4 累计值大于等于4，小于6时，间隔3个正常点情况计数

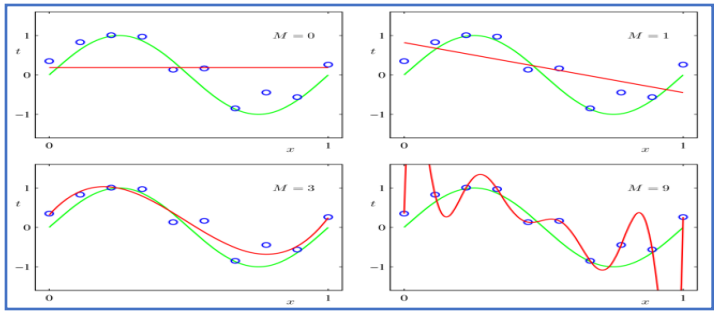
5 同时会记录告警的开始时间，结束时间，
告警时间点数，总异常时间点数，总记录数，告警次数，告警累计值

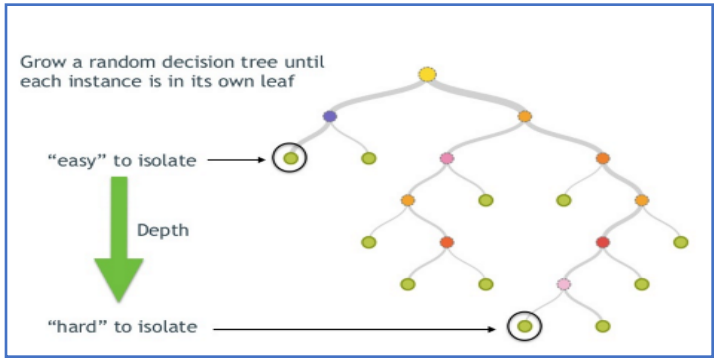
曲线监控

DOIS

（区间3sigma、统计判别、多项式拟合+阈值）+无监督+有监督







view_id	attr_id	beg_time	end_time	sender	alarm_continue	is_check
7335	2151648	2017-11-07 19:02:03	2017-11-07 19:10:02	0	0	1
1405	142637	2017-11-07 19:59:02	2017-11-07 20:06:02	0	0	1
15669	222132	2017-11-08 00:00:02	2017-11-08 10:39:02	0	0	1
6327	390161	2017-11-08 00:04:02	2017-11-08 00:11:02	0	0	1
6368	390161	2017-11-08 00:04:02	2017-11-08 00:12:03	0	0	1
6327	390161	2017-11-08 00:04:03	2017-11-08 00:12:02	0	0	1
6327	390163	2017-11-08 00:04:02	2017-11-08 00:11:02	0	0	1
6327	390163	2017-11-08 00:04:03	2017-11-08 00:12:02	0	0	1
6241	297212	2017-11-08 09:48:02	2017-11-08 09:53:02	0	0	1
6368	390163	2017-11-08 00:04:02	2017-11-08 00:12:03	0	0	1
16397	2389117	2017-11-08 09:03:03	2017-11-08 09:08:02	0	0	1
4392	20	2017-11-08 10:04:02	2017-11-08 10:09:02	0	0	1
4651	54039	2017-11-08 00:04:03	2017-11-08 00:09:03	0	0	1
11395	129854	2017-11-08 08:53:02	2017-11-08 08:58:02	0	0	1
15013	2389117	2017-11-08 09:04:03	2017-11-08 09:09:02	0	0	1
18018	629583	2017-11-08 09:20:02	2017-11-08 09:25:02	0	0	1
69141	10788	2017-10-25 11:04:03	2017-10-25 11:05:03	1	0	0
69141	10788	2017-10-25 11:04:03	2017-10-25 11:05:03	1	0	0
69141	10788	2017-10-25 11:04:03	2017-10-25 11:05:03	1	0	0
69141	10788	2017-10-25 11:04:03	2017-10-25 11:05:03	1	0	0
69141	10788	2017-10-25 11:04:03	2017-10-25 11:05:03	1	0	0
69141	10788	2017-10-25 11:04:03	2017-10-25 11:05:03	0	0	0
69141	10788	2017-10-25 11:04:03	2017-10-25 11:05:03	0	0	0

样本库构建——相关系数寻找相似样本

DOIS

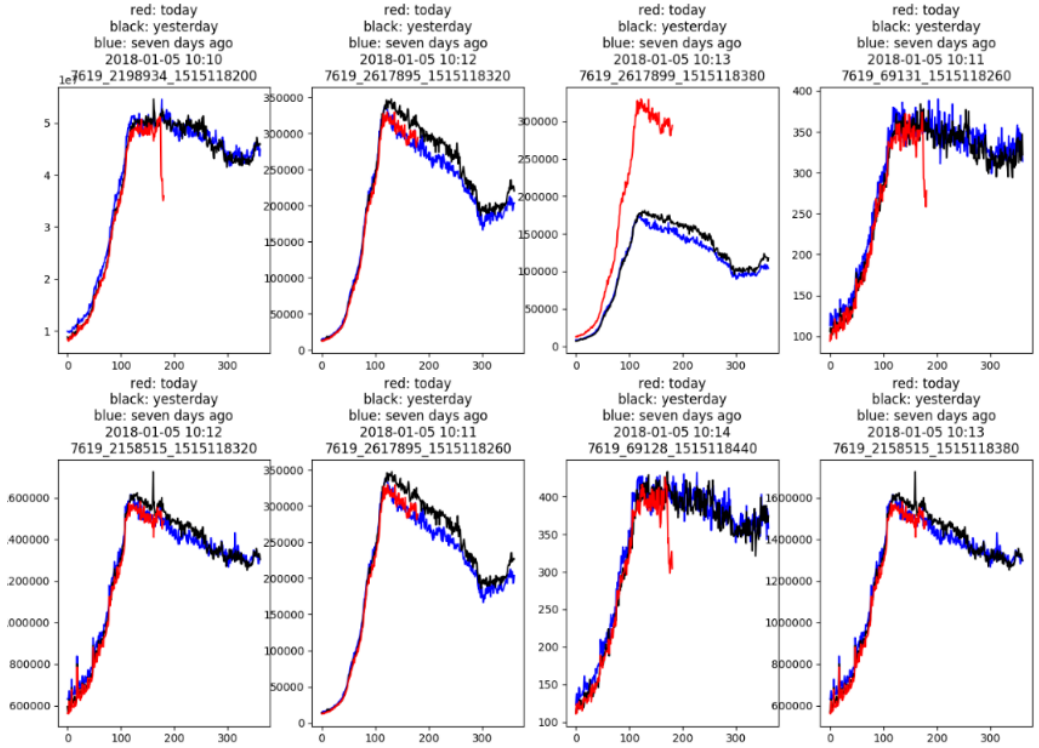
皮尔逊积矩相关系数：

在统计学中，Pearson相关性系数(Pearson Correlation)是衡量两个向量X和Y之间线性相关程度的一种方法。输出范围为[-1, +1]。

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}, \qquad \rho_{X,Y} \in [-1,+1]$$

$$\begin{cases} = +1, & \text{X和Y可以由直线可方程描述，且Y随着X的增加而增加} \\ = -1, & \text{X和Y可以由直线可方程描述，且Y随着X的增加而减少} \\ = 0, & \text{X和Y之间没有任何线性关系} \end{cases}$$

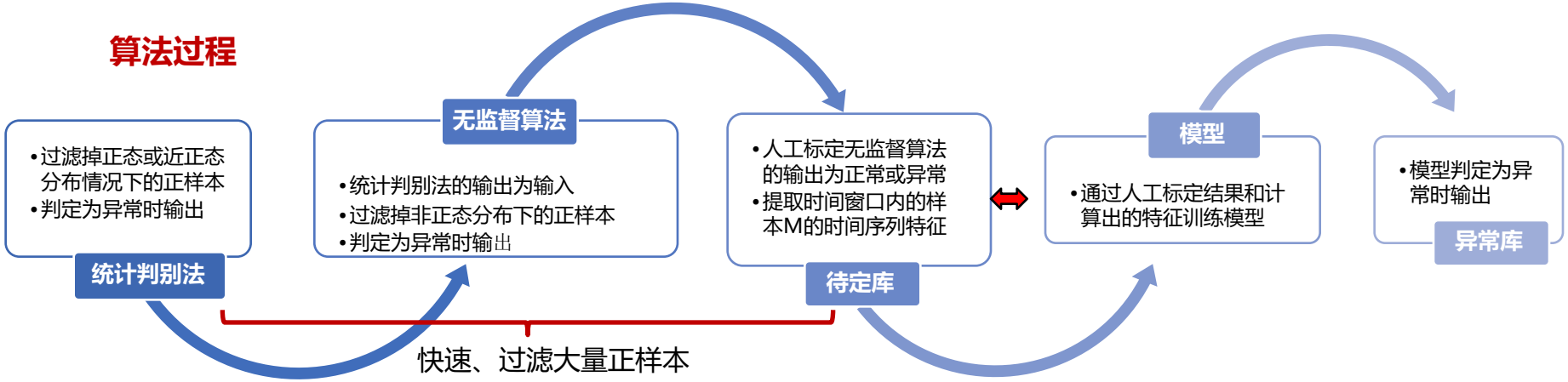
已实现基于Pearson相关性系数进行相似曲线判别与搜索能力(默认绝对值取0.9)。

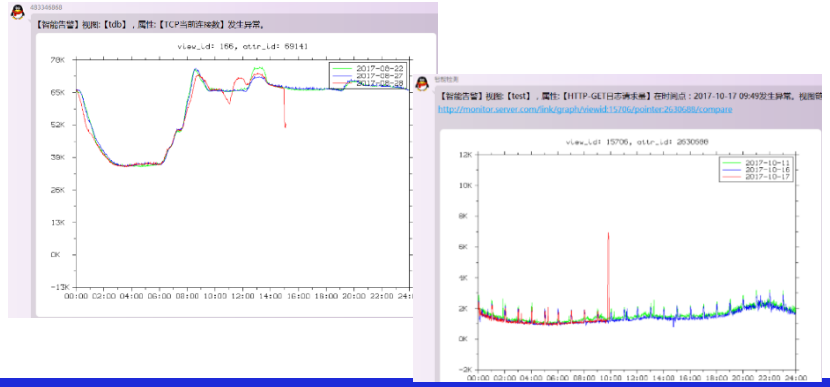


曲线监控的三级过滤

DOIS

算法过程





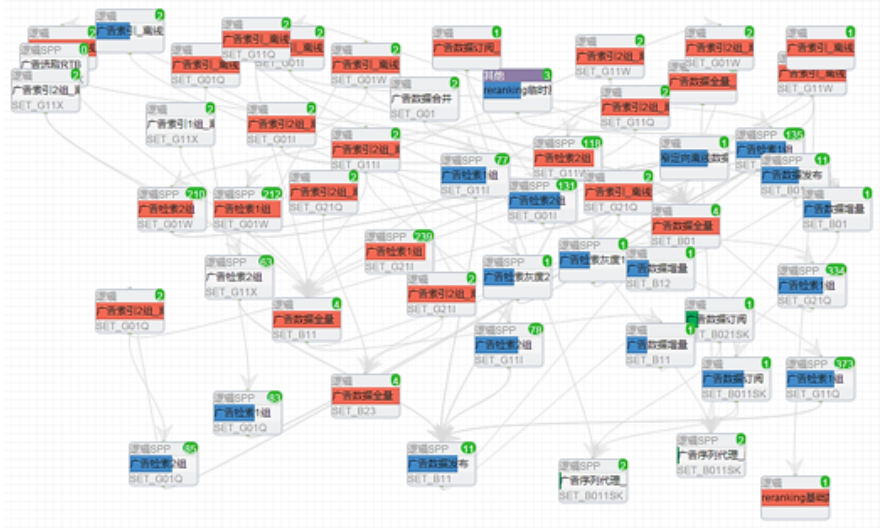
实际成果

- **秒级**监控
- 统计判别+无监督算法：**100%**覆盖、**125万**次/分

串联应用案例
质量 — ROOT智能根源异常分析

ROOT智能根源异常分析

DOIS



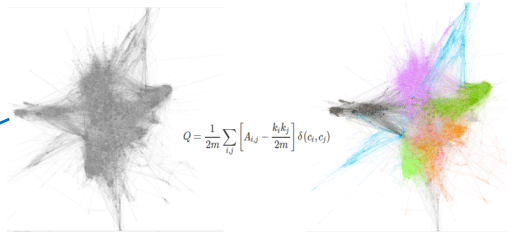
原始访问关系图

降纬访问关系的告警叠加图及面积算法

由原来的面积法更新为通过Apriori 频繁项集+皮尔逊相关系数定位根源告警

ROOT智能根源异常分析

DOIS



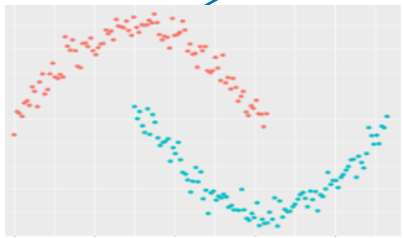
$$Q = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

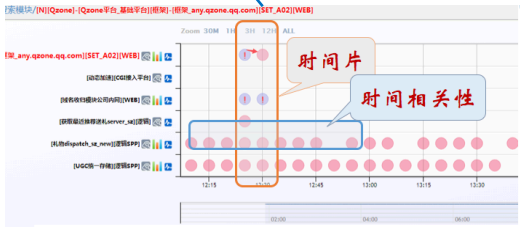
访问紧密度

访问关系采集 → 访问关系降纬 → Group社团划分 → 告警叠加 → DBSCAN异常度聚类 → 调用相关性 时间相关性 曲线相似性 → 根源结论

用户端监控　服务内监控

业务侧监控　基础监控





3sigma 统计判别 无监督 有监督

DBSCAN

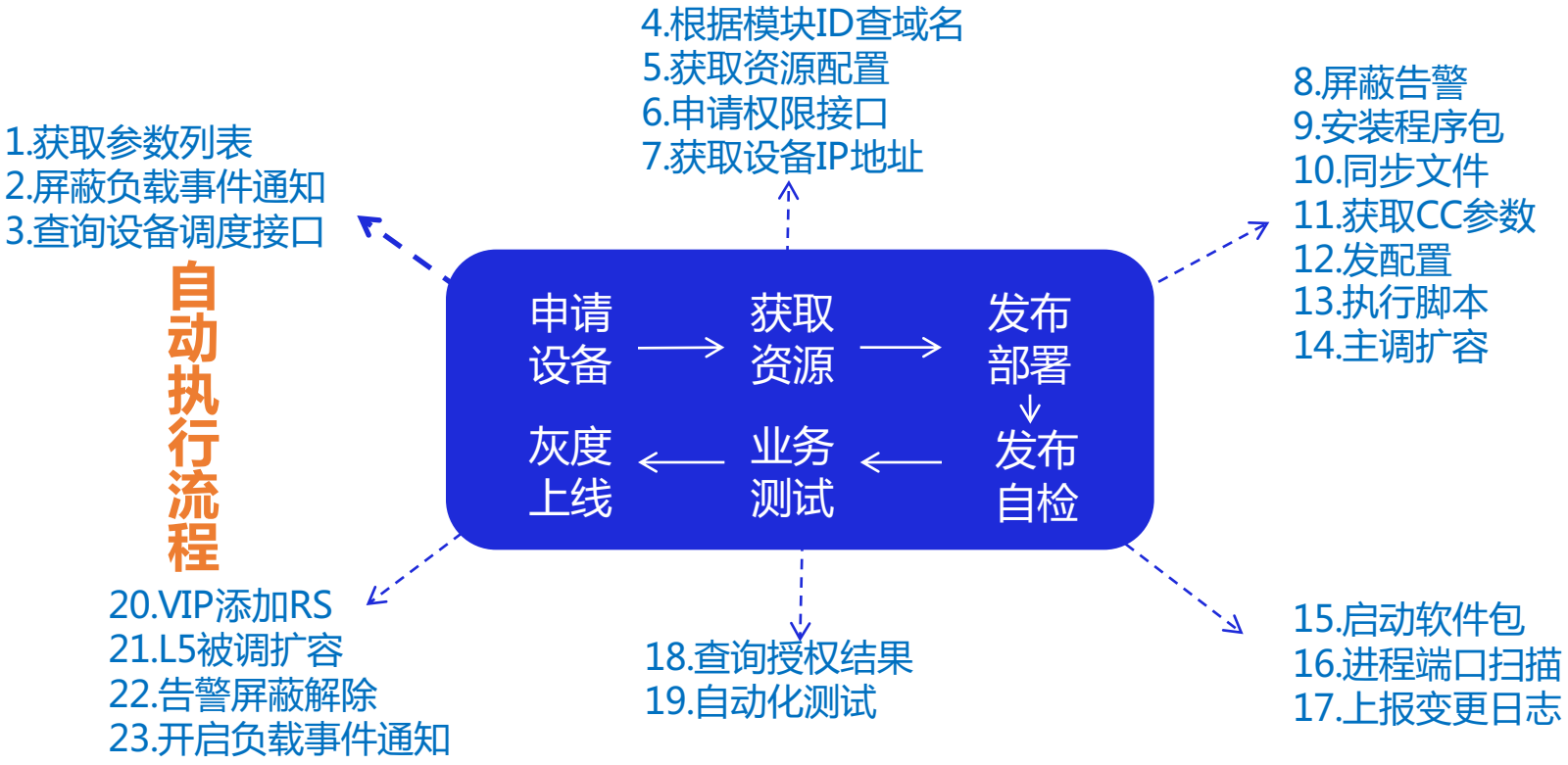
皮尔逊相关系数 +Apriori 频繁项集

DOIS

智能调度案例
效率 — 织云全自动扩容

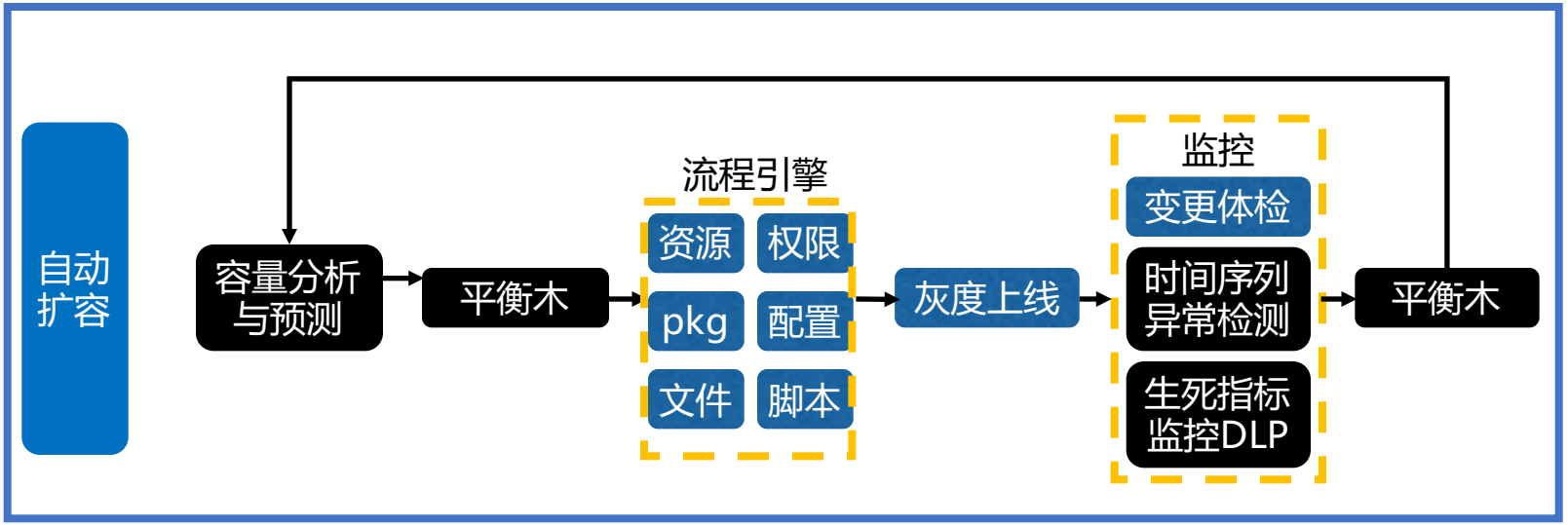
通过一个视频，重新认识织云全自动扩容流程

DOIS




织云
首页
设备列表
资源管理
包
配置
文件中转站
权限
脚本
流程
模调
调度策略
DLP
Docker
模块访问
平衡木
> [区服信息gc_area_info_svi][逻辑SPP]
? 图标说明
共计 0 个资源
进入旧版资源管理
暂停一致性 Off Docker模式 Off 进入Docker管理页
+ 添加资源
导入资源
一致性实时扫描
部署一致性
包 0 + 添加
配置 0 + 添加 导入CC下发配置
文件中转站 0 + 添加 目录拷贝
权限 0 + 添加 导入QZA 导入BitMap 导入OIDB 导入IpServer 权限全量导入 扫描权限一致性
测试工具 0 + 添加 执行测试工具
脚本 0 + 添加
Copyright©1998-2018 Tencent. Inc. All Rights Reserved
SNG社交网络运营部 版权所有
联系我们：zhyun-helper(织云助手)
织云接口文档 开发手册 访问统计 反馈 FAQ Changelog
本视频将展示腾讯织云无人运维的能力。

织云全自动扩容



灰色表示运维场景

蓝色表示自动化工具

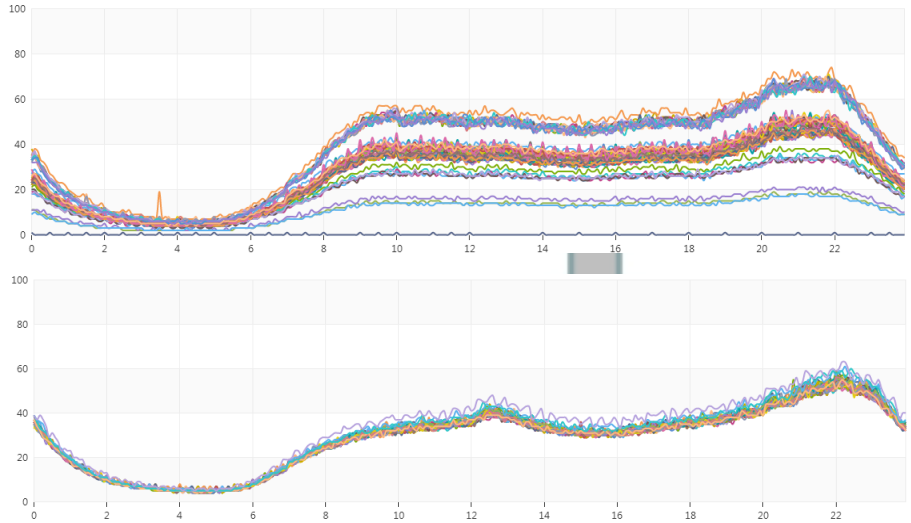
黑色表示智能化学件

流程管理 / 流程执行

- 获取参数列表
- 检查一致性上报
- 检查设备连通性
- 获取资源配置
- 申请权限
- 屏蔽告警
- L5主调扩容
- cmlb主调扩容
- 安装程序包
- 同步文件
- 获取cc参数
- 发配置
- 执行脚本
- 启动软件包
- 进程端口扫描
- 告警屏蔽解除
- 执行测试工具
- L5被调扩容
- CMLB被调扩容
- 上报变更日志

平衡木

DOIS



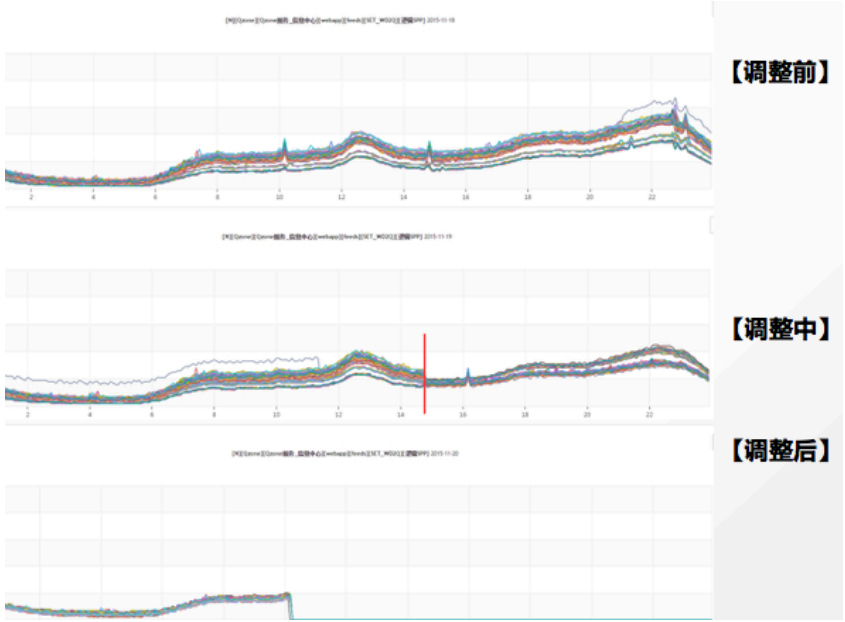
SET	当日负载值(%)
G21I	64.08%

支撑能力提升22%

SET	当日负载值(%)
G21I	42.27%

实际效果：通过几次迭代，模块整体支撑性能提升20%以上。

平衡木



$$L(w_1, \ldots, w_n) = \frac{1}{2n}\sum_{j=1}^{n}(f_j(w_j) - \mu)^2$$

通过梯度下降法，找到一组$w_1, \ldots w_n$，
使得所有机器的CPU使用率的方差最小

更多单点或串联应用

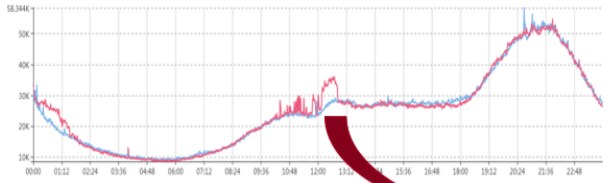
多维下钻智能分析

例如：视频服务业务的KPI指标总卡顿次数统计维度由以下多个维度组成

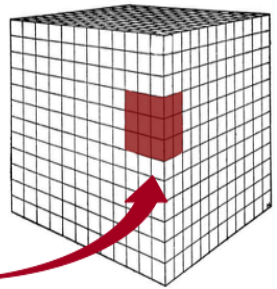
播放平台 | 客户端 | 运营商 | 省份 | 域名 | 版本 | IDC | ...

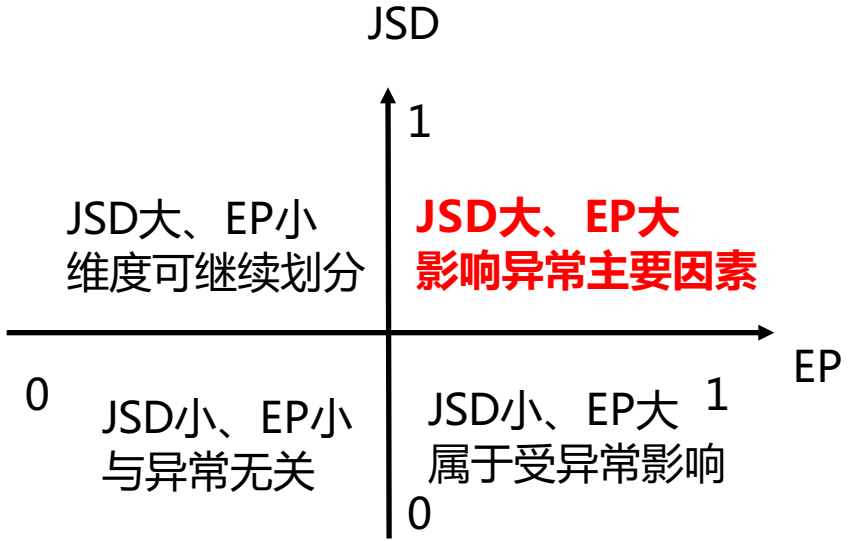
如何快速找到大量组合中最核心的影响因素？

异常KPI曲线



异常维度组合





目标：搜索出对异常波动和贡献度最大的维度组合，差异度：（ Jensen-Shannon (JS) divergence ）（找出维度）

$$D_{JS}(P,Q) = 0.5\left(\sum_i p_i \log\frac{2p_i}{p_i+q_i} + \sum_i q_i \log\frac{2q_i}{p_i+q_i}\right)$$（预测值和观测值）

贡献度：$EP_{ij} = (A_{ij}(m) - F_{ij}(m))/(A(m) - F(m))$　　（找出子项）

频繁项集关联告警智能分析

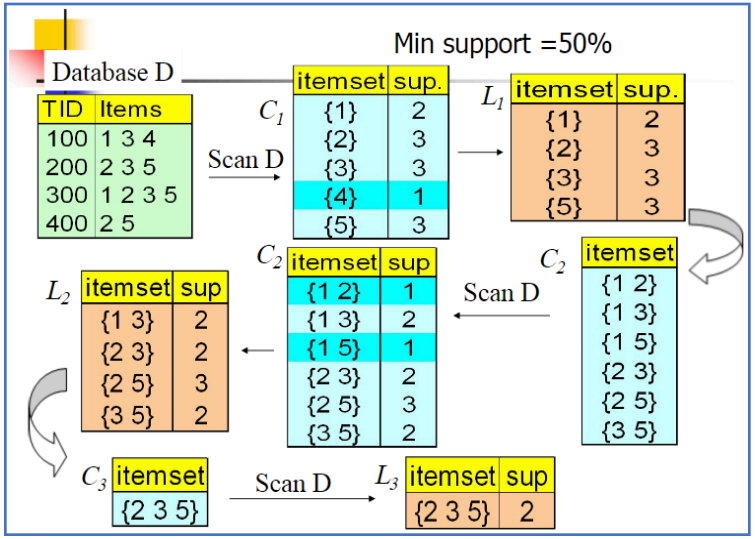
DOIS

【告警智能收敛】模调告警收敛详情列表

统计时间	发送Index	告警对象	告警实际发生时间	是否频繁发生	是否合并发送
2017-11-01 00:00:00	908	204968332#204500030#104420007	2017-10-31 04:32:00	N	Y
	920	206806286#211500228#111500298	2017-10-31 04:34:00	N	Y
		206806286#211500228#106000899	2017-10-31 04:34:00	N	Y
	1361	211800099#211900051#111900195	2017-10-31 06:42:00	N	Y
		211006383#211900051#111900195	2017-10-31 06:42:00	N	Y
		211800110#211900051#111900195	2017-10-31 06:42:00	N	Y
	1391	211006383#211900051#111900195	2017-10-31 06:52:00	N	Y
		211800099#211900051#111900195	2017-10-31 06:52:00	N	Y
	1794	211800099#211900051#111900195	2017-10-31 08:42:00	N	Y
		211006383#211900051#111900195	2017-10-31 08:42:00	N	Y
	1871	204968033#204971944#104974039	2017-10-31 09:01:00	N	Y
		204970539#204971944#104974039	2017-10-31 09:01:00	N	Y
	2005	211800099#211900051#111900195	2017-10-31 09:42:00	N	Y
		211006383#211900051#111900195	2017-10-31 09:42:00	N	Y
	2061	211800060#211100166#104971992	2017-10-31 09:56:00	N	Y
		211006089#211006088#104971992	2017-10-31 09:55:00	N	Y
		211800060#211100166#104971992	2017-10-31 09:55:00	N	Y
	2099	204972238#204972238#104974199	2017-10-31 10:05:00	N	Y
		204972238#204973775#104976557	2017-10-31 10:05:00	N	Y
	2294	204967896#204402154#104922867	2017-10-31 10:35:00	N	Y

893 2/45 1 2 3 4 5 6 7 8 9 10 2

主调 被调 接口



支持度就是**几个关联的数据**在数据集中出现的次数占**总数据集的比重。**

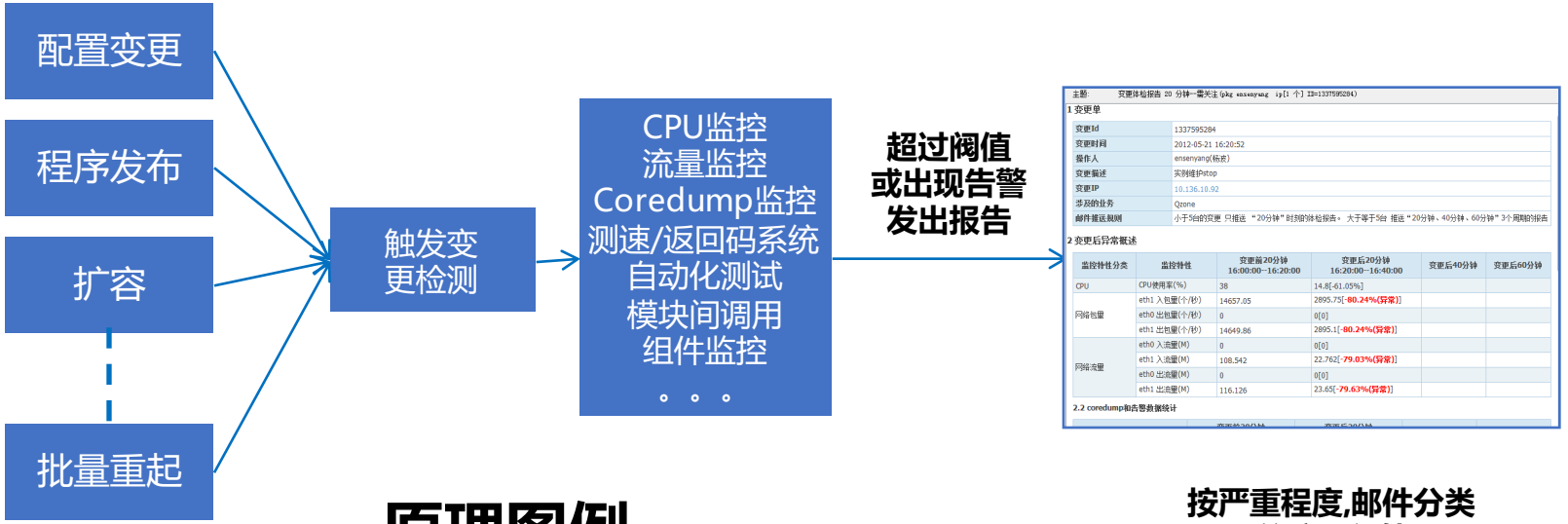
或者说几个数据关联出现的概率

置信度体现了**一个**数据出现后，**另一个**数据出现的概率

图片来源：
https://www.evget.com/article/2017/4/12/26086.html

Apriori 频繁项集

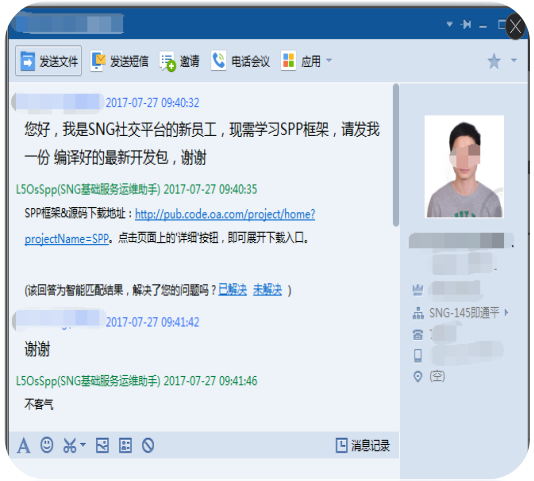
变更体检智能报告



原理图例

智能咨询客服 / 操作工具客服

DOIS



Demo1：
完全识别，直接给出答案

Demo2：
部分识别，给出相关问题

Demo3：
查询 / 操作模式

目录

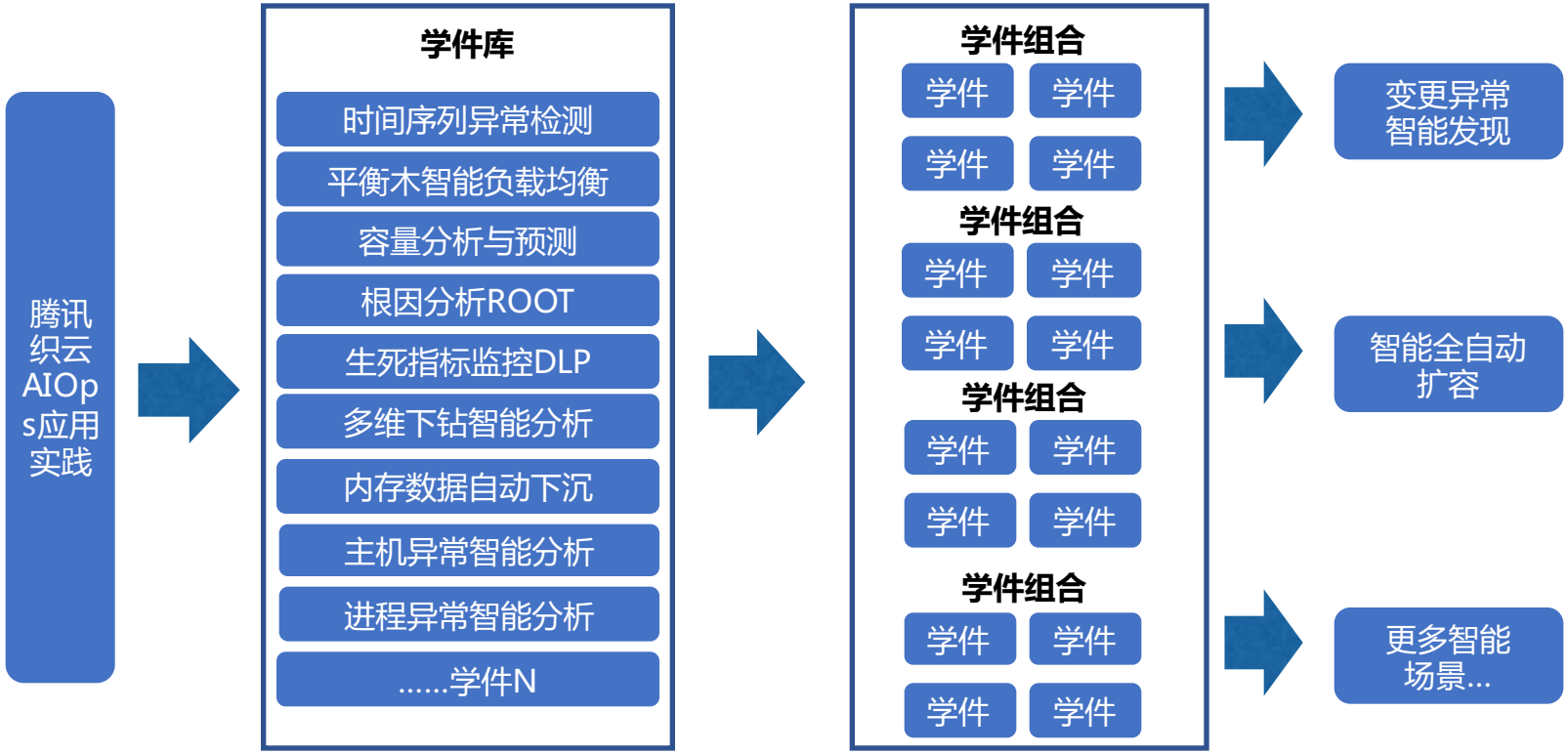
内部学件和公开学件

DOIS

- 内部学件
- 公开学件
- 学件仓库
- 学件接口标准

织云Metis平台

DOIS



DOIS

Thanks

DevOps 时代社区 荣誉出品

想第一时间看到高效运维社区的新动态吗？

高维在线