关于智能运维中
算法落地的一些思考

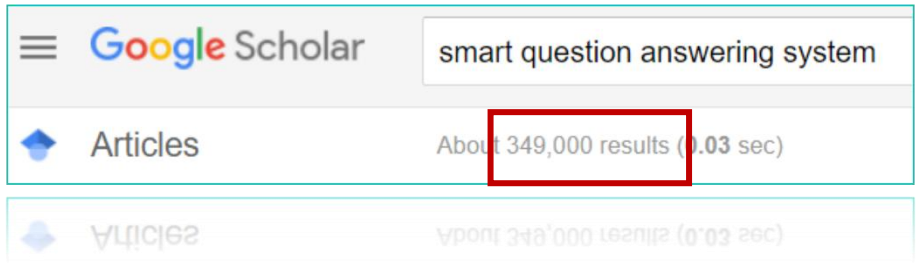
王鹏：复旦大学计算机科学技术学院 教授

目 录

智能运维的现状

智能问答系统

海量的算法和技术



VS

差强人意的效果

智能运维现状

算法日益丰富

- 指标
- 日志/告警
- CMDB、调用链

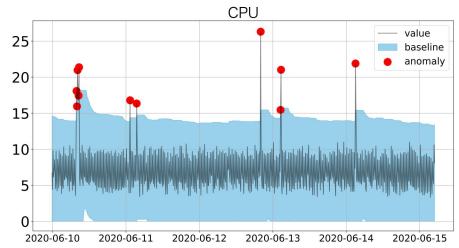
算法效果不断提升

- 指标异常检测
- 容量预测
- 日志聚类
- 日志日常检测
- 告警中的场景挖掘
- 根因定位

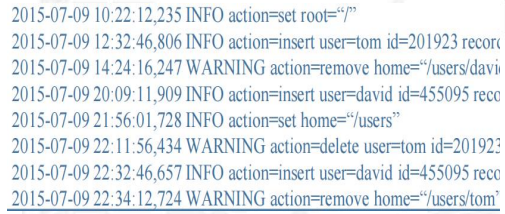
单指标异常检测
多指标异常检测
容量预测

日志聚类
告警压缩
场景挖掘

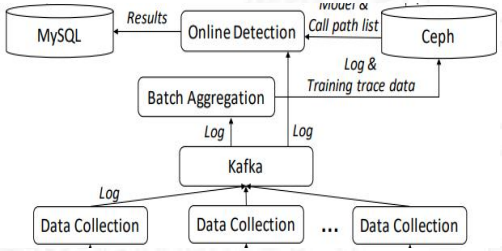
根因定位
图异常检测



性能指标



日志/告警



CMDB/调用链

智能运维现状：指标异常检测

落地最多的智能运维场景

- 数据容易准备、效果容易验证
- 对大规模指标进行异常检测（10000、100000、。。。）

研究者提出了大量的异常检测算法

- 单指标、多指标
- 基于统计模型、基于深度学习
- 无监督、有监督
- 多个公司和机构开源了异常检测数据集和算法

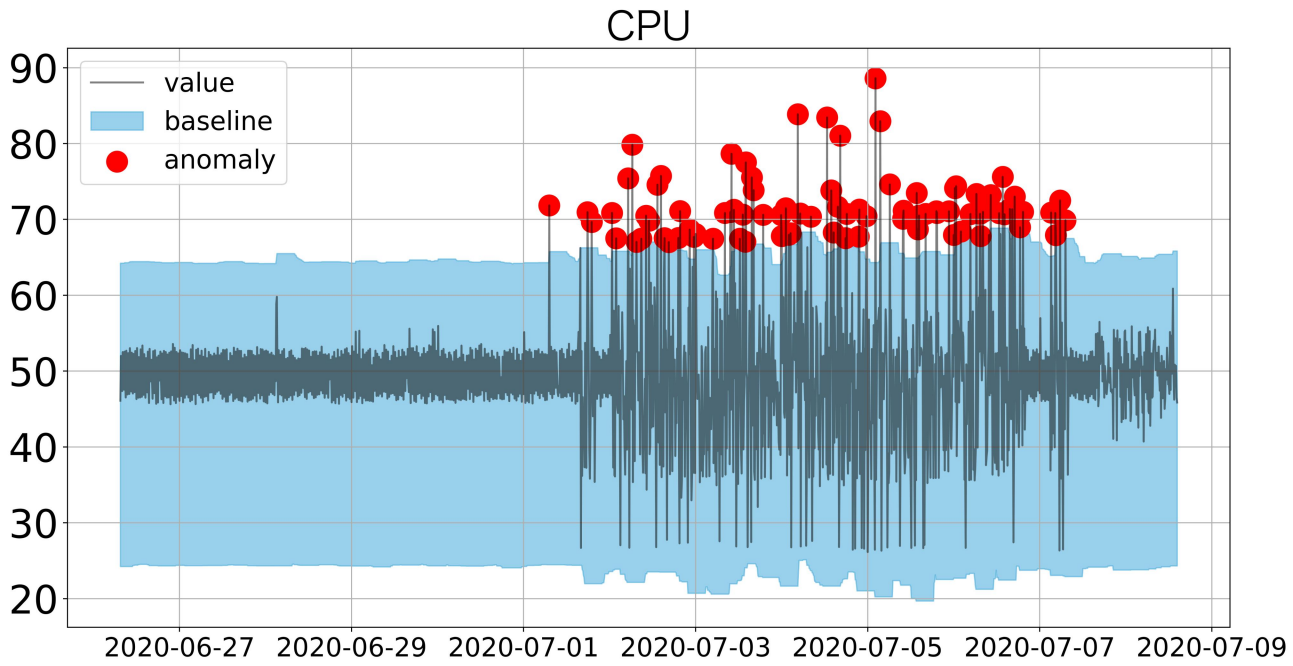
智能运维现状：指标异常检测

往往真实环境中应用的效果不尽如人意

01

误报太多

- 为了消除漏报，往往造成大量的误报
- 运维人员不得不忽略所有的指标异常告警

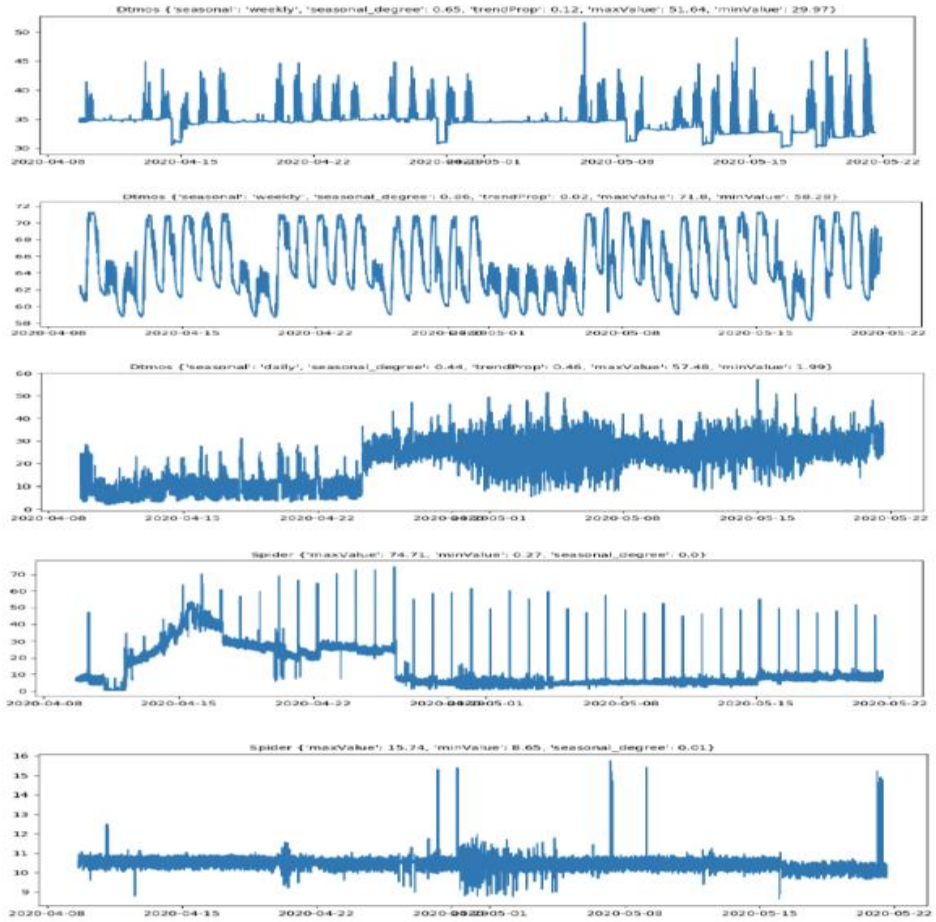


智能运维现状：指标异常检测

往往真实环境中应用的效果不尽如人意

02 模型/参数难以设置

- 不同类型的指标，往往适合不同类型的模型和参数

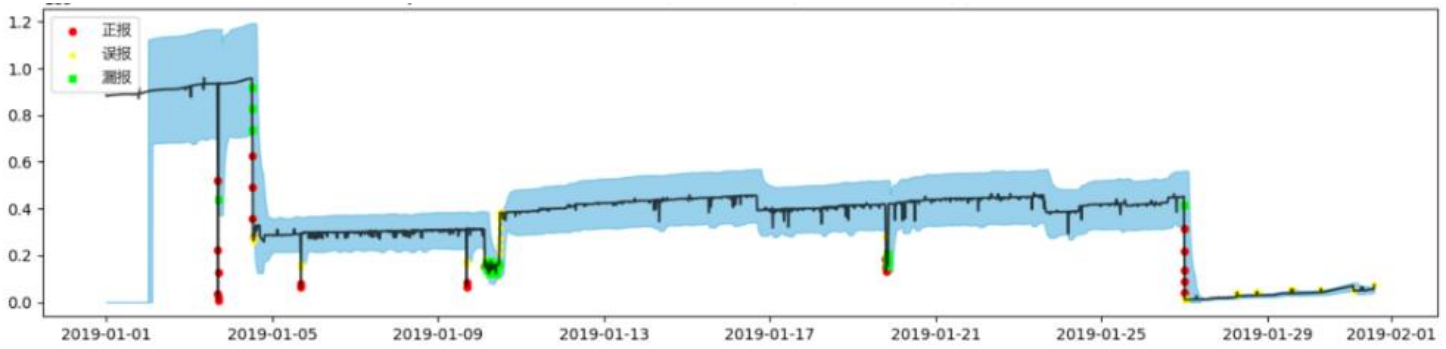


智能运维现状：指标异常检测

往往真实环境中应用的效果不尽如人意

03 缺乏有效的反馈和修正机制

- 缺乏问题发现能力：监测5万个指标，一天内报了2000个异常，难以对这些异常进行展示和分析，类型、主机、时间段、业务？
- 缺乏基于反馈的模型调整能力，难以应对“这个不是异常，后续检测中不要再报了”的个性化需求



智能运维现状：日志/告警智能分析

很多企业上线了日志/告警相关算法

- 人工难以处理，基于规则的方法维护性差
- 典型场景：模板提取，场景挖掘，基于日志的异常检测
- 变量取值异常、模板数量异常、语义异常等

研究者提出了大量的算法

- 模板提取：Drain、Spell、LogCluster
- 场景挖掘：OAS
- 日志异常检测：DeepLog、LogAnomaly
- 公开的数据集：Loghub

日志1： IP: 10.142.212.10 , Port: 80 - Connection open
日志2： IP: 10.142.212.11 , Port: 22 - Connection open

模板： IP: **$(IP地址)** , Port: **$(端口)** – Connection open

告警1： 通讯节点1(node1):在**2019-01-10 05:26:51**
时出现交易异常:超过**367**秒无交易上送
告警2： 通讯节点2(host2):在**2019-01-10 05:49:41**
时出现交易异常:超过**361**秒无交易上送

模板： 通讯节点**$NUM**(**主机名**):在**$datetime**
时出现交易异常:超过**$NUM**秒无交易上送

智能运维现状：日志/告警智能分析

日志智能分析实践存在若干问题

01 模板质量难以有效评估

- 模板数量大（几百上千），逐个人工判断耗时太长

Dataset	Description	Time Span	Data Size	#Messages	#Templates (total)	#Templates (2k)
Distributed system logs						
HDFS	Hadoop distributed file system log	38.7 hours	1.47 GB	11,175,629	30	14
Hadoop	Hadoop mapreduce job log	N.A.	48.61 MB	394,308	298	114
Spark	Spark job log	N.A.	2.75 GB	33,236,604	456	36
ZooKeeper	ZooKeeper service log	26.7 days	9.95 MB	74,380	95	50
OpenStack	OpenStack software log	N.A.	60.01 MB	207,820	51	43
Supercomputer logs						
BGL	Blue Gene/L supercomputer log	214.7 days	708.76 MB	4,747,963	619	120
HPC	High performance cluster log	N.A.	32.00 MB	433,489	104	46
Thunderbird	Thunderbird supercomputer log	244 days	29.60 GB	211,212,192	4,040	149
Operating system logs						
Windows	Windows event log	226.7 days	26.09 GB	114,608,388	4,833	50
Linux	Linux system log	263.9 days	2.25 MB	25,567	488	118
Mac	Mac OS log	7.0 days	16.09 MB	117,283	2,214	341
Mobile system logs						
Android	Android framework log	N.A.	3.38 GB	30,348,042	76,923	166
HealthApp	Health app log	10.5 days	22.44 MB	253,395	220	75

引自：Jieming Zhu, et al. Tools and Benchmarks for Automated Log Parsing. International Conference on Software Engineering (ICSE), 2019

智能运维现状：日志/告警智能分析

日志智能分析实践存在若干问题

02 **缺乏有效的反馈和修正机制**

- 缺乏客观评判
- 缺乏基于反馈的模板调整能力，难以应对“这种模板应该根据这个变量拆分“ 、 ”这个变量应该被泛化”之类的个性化需求

主机:$ip进程:<*CUST1_ExpFile*>进程消失

主机:$ip进程:<*ImportSWFMsg*>进程消失

例1 （某金融机构告警数据）

*** unregister callback for <*>@<*>

*** unregister callback for null

[HSM] stayAwake false uid: <*>, pid: <*>

[PhoneIntfMgr] getDataEnabled: subId=<*> phoneId=<*>

[PhoneIntfMgr] getDataEnabled: subId=<*> retVal=true

例2 （Andriod日志）

智能运维现状：日志/告警智能分析

告警智能处理存在若干问题

03 根因定位效果欠佳

- CMDB普遍质量不高
- 可能真正的故障原因不存在与告警数据中
- 标签数据缺失

问题分析

造成目前状况的主要问题

战术层面

- 问题1：难以对数据和算法结果进行深入分析
- 问题2：算法的自适应能力和反馈修正能力弱
- 问题3：全局算法能力弱

战略层面

- 问题4：算法和领域知识结合较差
- 问题5：对智能运维中算法作用的认识存在偏差

问题1：难以对数据和算法结果进行深入分析

数据探索是算法研究的前提，算法结果分析是算法优化的必要步骤

被监测对象规模庞大

- 数据：十万/百万个指标，每天上TB的日志
- 结果：每天上千个异常，上千/上万个日志模板
- 难以进行可视化
- 数据探索能力弱

可选算法多，参数搜索空间大

- 异常检测：不同指标适合的模型不同、参数不同
- 搜索空间巨大：指标规模 × 模型个数 × 参数取值个数

问题1：难以对数据和算法结果进行深入分析

数据探索是算法研究的前提，算法结果分析是算法优化的必要步骤

被监测对象规模庞大

- 数据：十万/百万个指标，每天上TB的日志
- 结果：每天上千个异常，上千/上万个日志模板
- 难以进行可视化
- 数据探索能力弱

可选算法多，参数搜索空间大

- 异常检测：不同指标适合的模型不同、参数不同
- 搜索空间巨大：指标规模 × 模型个数 × 参数取值个数

but rather, needs to be alerted to the abnormal case. These abnormal events are termed *anomalies*, and the algorithms that find them fall under the category of *anomaly detection* (AD) [6]. Time series AD systems are typically built based on machine learning (ML) models trained with previously collected datasets. This process is inherently an iterative and human-in-the-loop (HIL) process for various reasons:

Anomaly detection is a highly domain-specific problem. What constitutes an anomaly changes greatly from one application domain to another. This bears the need to develop, compare, and choose from multiple AD models and algorithms to find one that best suits a given application.

Time series anomalies can be complex. They typically fall under the category of collective and contextual anomalies [6], exposing themselves at different time granularities or aggregations. Anomalous patterns may vary over time due to the temporal and dynamic nature of the data. Multiple, potentially correlated attributes may require multi-variate analysis. Overall, interpreting and reasoning about time series anomalies is not a straightforward task.

Data or training meta-data can be noisy/unavailable. In most domains, anomalies are relatively rare or unique events, making it challenging to collect anomalous datasets. Often, train-

摘自：MIT团队，SIGMOD’ 2019

原因2：算法的自适应能力和反馈修正能力弱

算法普遍缺乏反馈修正能力

- 这个“异常”我不需要，后续检测中不要再报了
- 这两个“模板”应该合并掉，这个“变量”不能被泛化
- 目前的算法缺乏基于反馈的自动修正的能力

标签数据获取困难

- 算法团队和运维专家沟通成本高
- 异常/故障本身就是小样本事件

Existing log parsers **overlook the importance of user feedback**, which is imperative for parser fine tuning under the continuous evolution of log data.

摘自MSRA团队，FSE’2022

原因3：全局算法能力弱

缺乏异构数据的协同分析（横向）

- 指标、日志、调用链中的异常都可能反映故障相关征兆
- 缺乏统一的异常评估机制
- 缺乏不同数据的分析结果之间相互印证的能力

缺乏不同阶段的协同分析（纵向）

- 从**数据采集**到**根因定位**存在多个数据处理环节
- 异常/故障本身就是小样本事件

缺乏有效的异构数据处理工具

- 花很多时间进行数据查询和分析

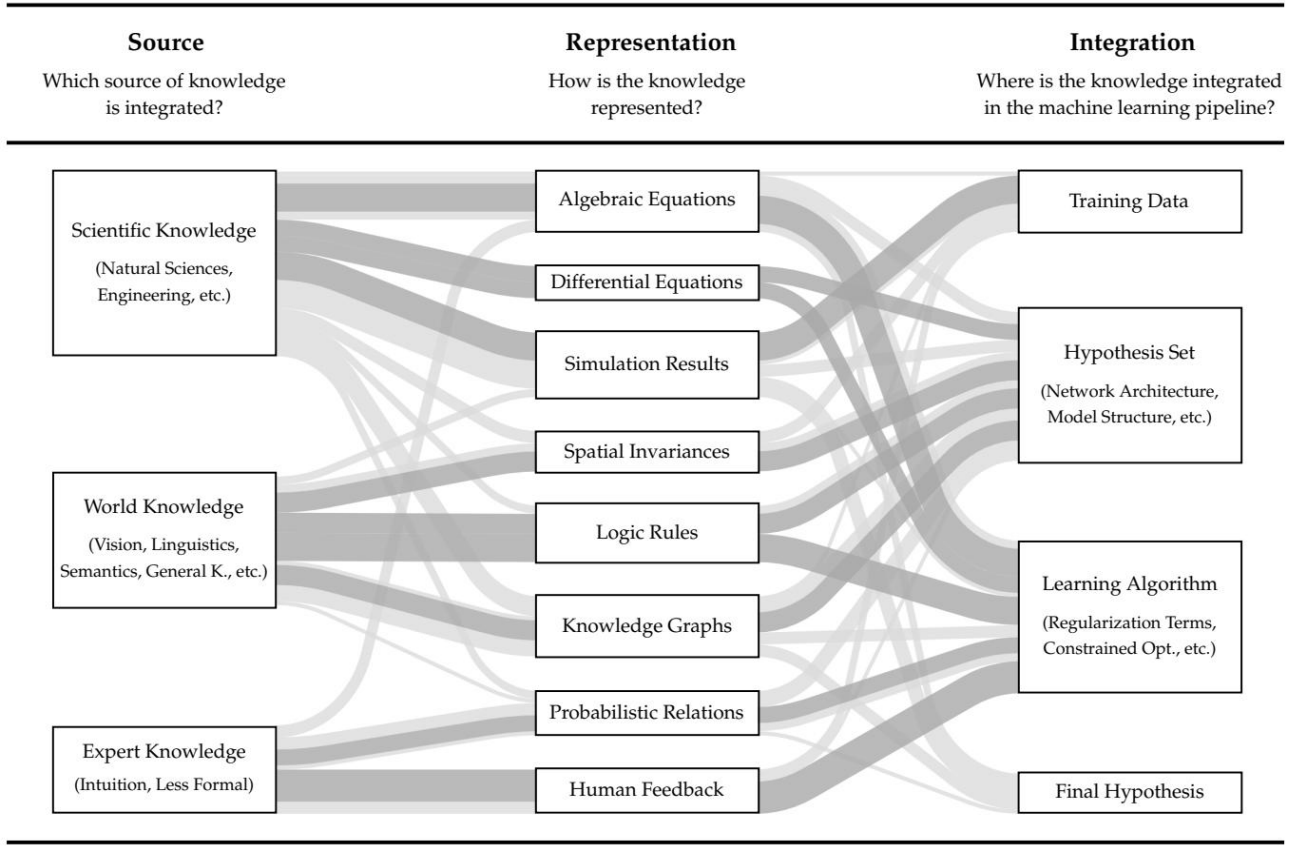
原因4：算法和领域知识结合较差

数据、算法和知识的融合是AI的研究热点

- AI 3.0
- 数据模型和机理的融合
- 可解释AI
- ...

运维本身是一个强知识领域

- 运维老兵的重要性体现在经验和知识的丰富性
- 现状：AIops算法对于运维知识的利用较少



源自：Informed Machine Learning--A Taxonomy and Survey of Integrating Knowledge into Learning Systems， TKDE 2021.

问题5：对智能运维中的算法的认识存在偏差

论文中的算法 vs. 真实场景需要的算法

以日志模板提取为例

- 大量的离线、在线日志模板提取算法
- 真实场景中的模板提取
- POC过程：在线 or 离线?
- 生产环境：在线 or 离线?
- 更为合理的日志提取算法
  - 小批量测试数据上的离线算法
  - 基于模板集合，流式数据上的在线算法

Log Parser	Year	Technique	Mode	Efficiency
SLCT	2003	Frequent pattern mining	Offline	High
AEL	2008	Heuristics	Offline	High
IPLoM	2012	Iterative partitioning	Offline	High
LKE	2009	Clustering	Offline	Low
LFA	2010	Frequent pattern mining	Offline	High
LogSig	2011	Clustering	Offline	Medium
SHISO	2013	Clustering	Online	High
LogCluster	2015	Frequent pattern mining	Offline	High
LenMa	2016	Clustering	Online	Medium
LogMine	2016	Clustering	Offline	Medium
Spell	2016	Longest common subsequence	Online	High
Drain	2017	Parsing tree	Online	High
MoLFI	2018	Evolutionary algorithms	Offline	Low

来源：Jieming Zhu, et al. Tools and Benchmarks for Automated Log Parsing. International Conference on Software Engineering (ICSE), 2019

问题5：对智能运维中的算法的认识存在偏差

论文中的算法 vs. 真实场景需要的算法

以日志模板提取为例

- 我们还需要别的算法吗
- 长日志模板提取算法
- 多行日志模板提取算法
- 特殊日志模板提取算法
- 反馈算法
- 参数自动设置算法
- ……

```
2017-07-01 19:46:26.133 GoogleSoftwareUpdateAgent[31702/0x7000002a0000]
[lvl=2] -[KSAgentApp(KeystoneThread) runKeystonesInThreadWithArg:] Checking
with local engine: <KSUpdateEngine:0x100259c60
ticketStore=<KSPersistentTicketStore:0x100253770
store=<KSKeyedPersistentStore:0x100254d10
path="/Users/xpc/Library/Google/GoogleSoftwareUpdate/TicketStore/Keystone.t
icketstore" lockFile=<KSLockFile:0x100254d80
path="/Users/xpc/Library/Google/GoogleSoftwareUpdate/TicketStore/Keystone.t
icketstore.lock" locked=NO > >> processor=<KSActionProcessor:0x100259e70
delegate=<KSUpdateEngine:0x100259c60> isProcessing=NO actionsCompleted=0
progress=0.00 errors=0 currentActionErrors=0 events=0 currentActionEvents=0
actionQueue=( ) > delegate=(null)
serverInfoStore=<KSServerPrivateInfoStore:0x1002594d0
path="/Users/xpc/Library/Google/GoogleSoftwareUpdate/Servers"> errors=0 >
```

Mac log example

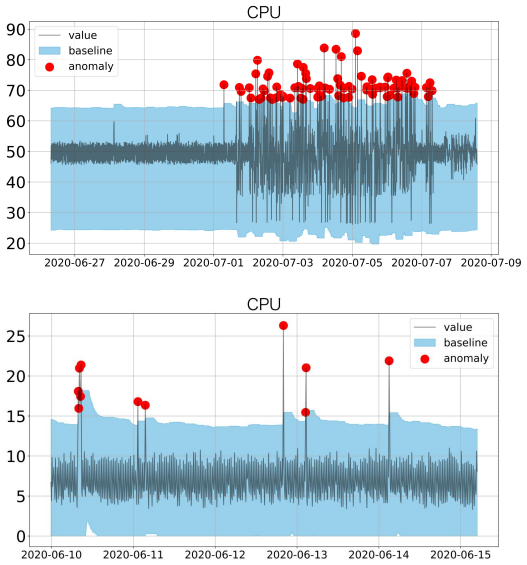
问题5：对智能运维中的算法的认识存在偏差

“完全依靠算法实现自动化运维”现实吗？

更现实的目标：算法做为一种辅助手段，让运维更高效

- 数据量太大，用算法来**提高效率**
  - 对每天几百TB的日志自动提取模板和变量
  - 对上万的指标自动进行异常检测
- **用算法来提高运维工具的易用性**
- 做为一种定位故障过程的辅助手段，灵活快速的**查询和探索数据**
- 算法做为一种**积累知识**的方式，构建知识图谱

智能运维三要素



- 算法只是手段
- 运维才是目标

算法的设计能力

- **需求个性化**：针对客户的需求，需要设计针对性的算法
- **数据个性化**：参数调整的复杂性和反复性

AIOps

运维知识的理解能力

数据平台的工程化能力

- 数据管理和探索能力是设计算法有效的前提
- 数据平台和算法的高效结合

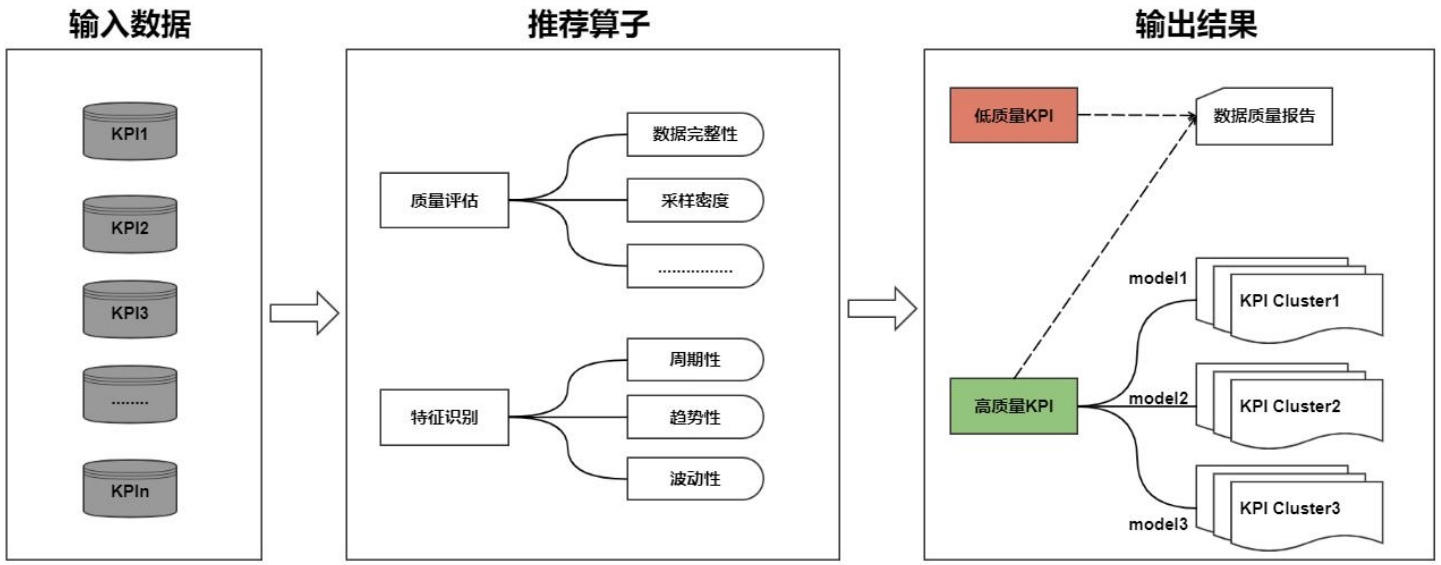
探索性工作

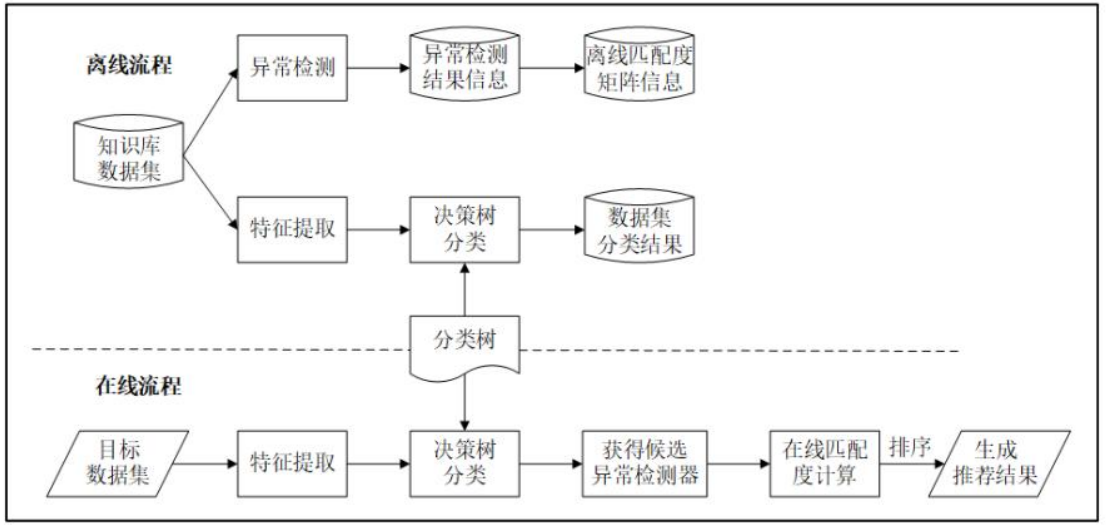
目标

- 提高算法应用效果
- 支持反馈，融合专家知识和经验
- 提高运维过程中的数据探索能力，从而提升运维工具的易用性

异常检测

- 算法和参数推荐
- 实现上万个指标的自动模型和参数推荐
- 构建指标和异常的知识库

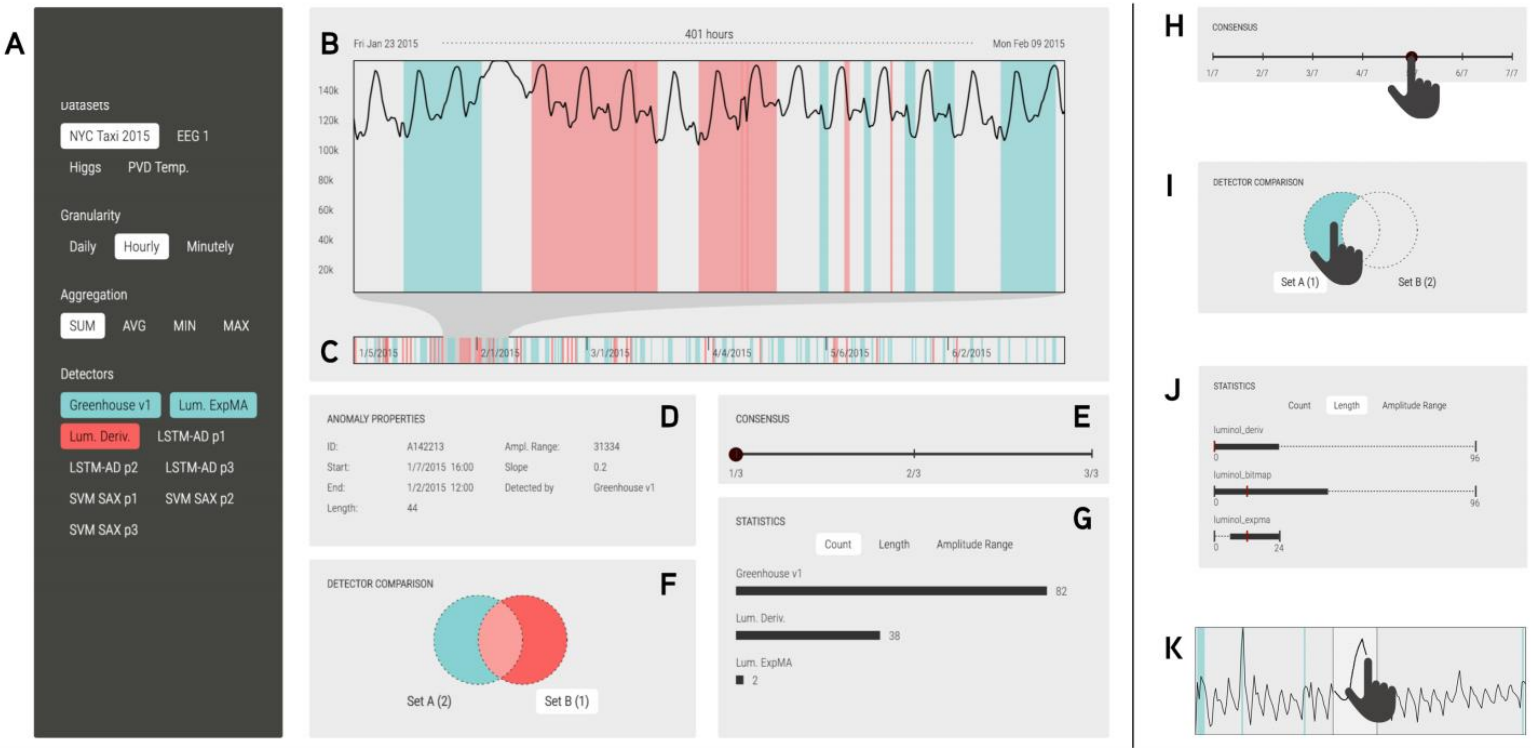


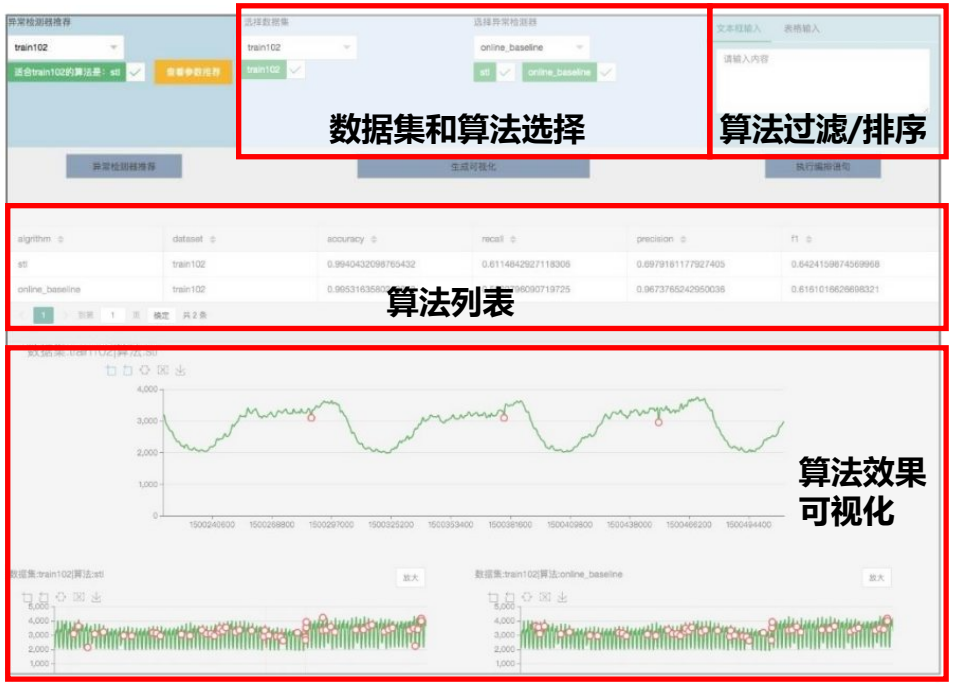


推荐算法

异常检测

- 基于可视化的算法比较、过滤、选择

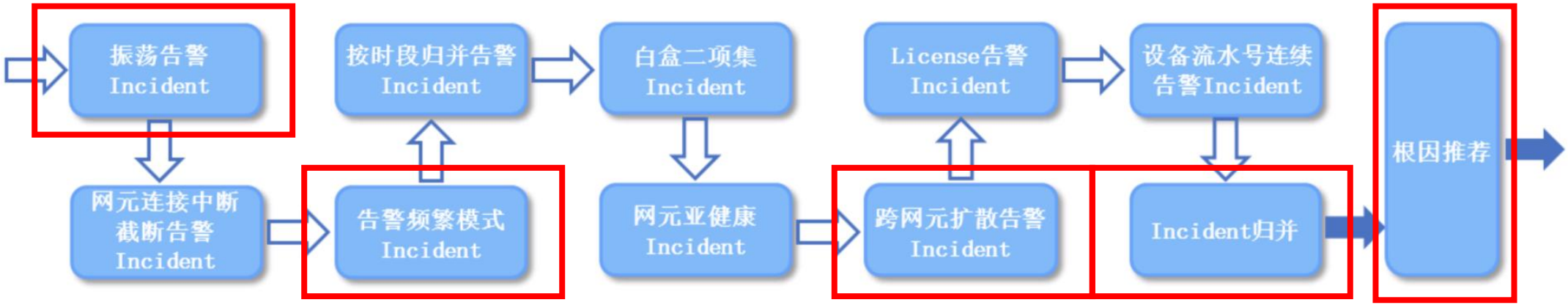




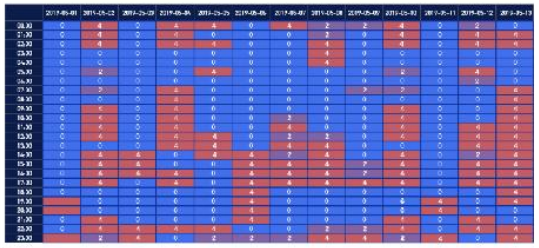
可视化算法选择

告警压缩

- 算法和专家经验的有效融合
- 将专家经验也做为一种算法，实现和真正算法同样的输入输出接口
- **专家经验的重要性**

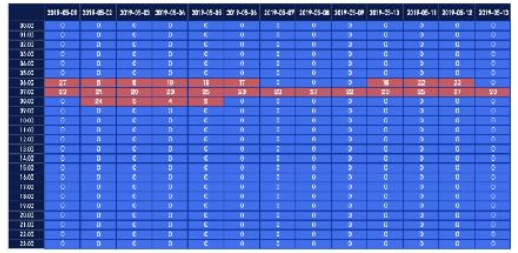


告警精细化管理



高频事件

历史出现次数较多，多天多次发生



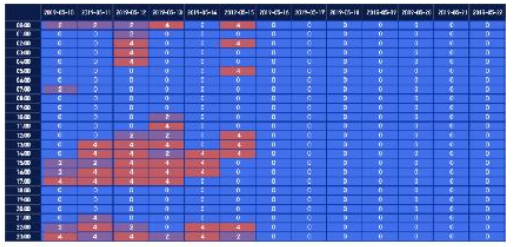
周期性事件

出现天数很多，并且集中在指定的时间段

场景化

新增事件

某时间点才开始出现的事件



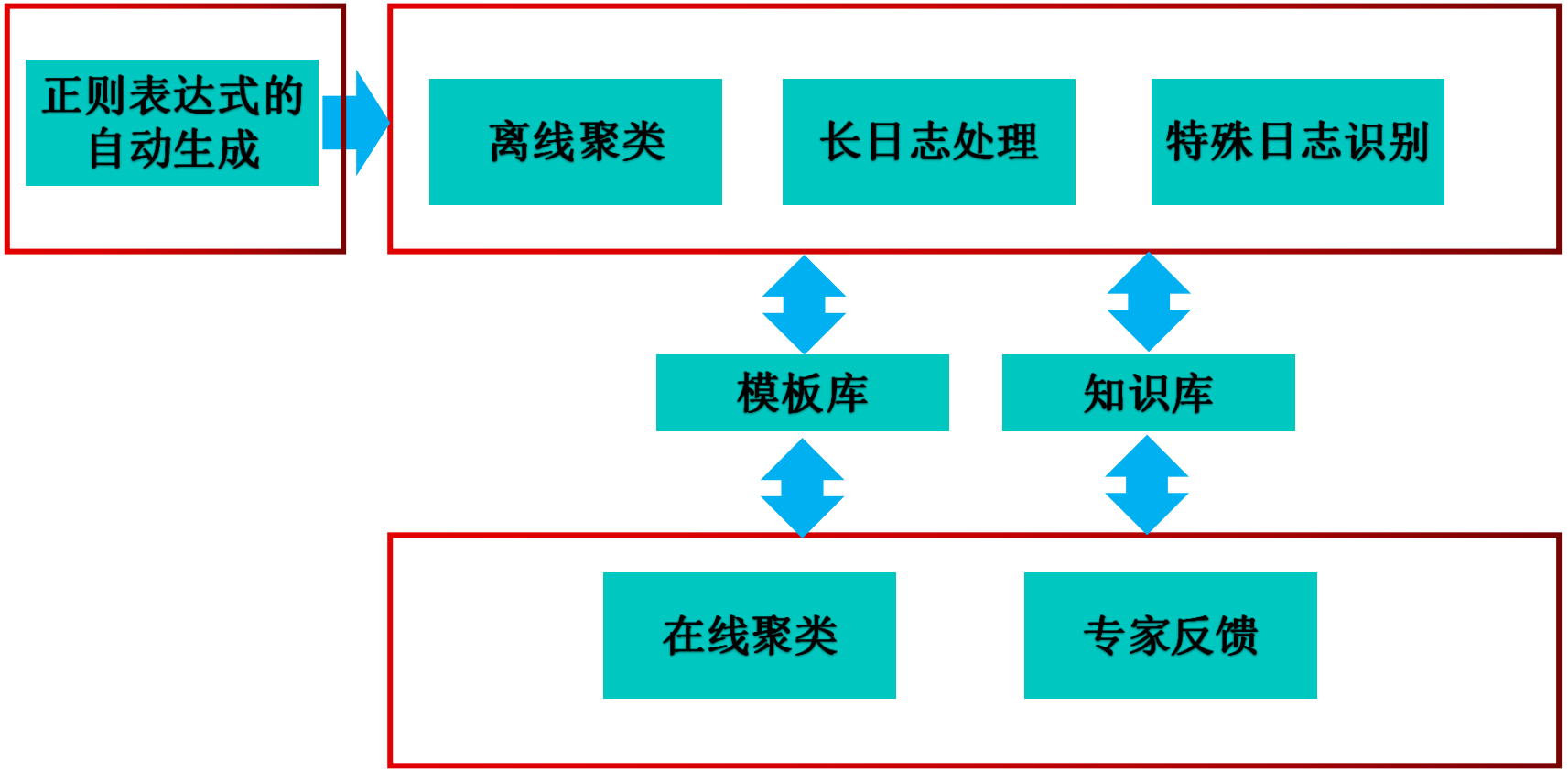
阶段性事件

某时间点后不再发生

业务理解和数据的深层次分析对于智能运维至关重要！

日志聚类

- 多算法融合

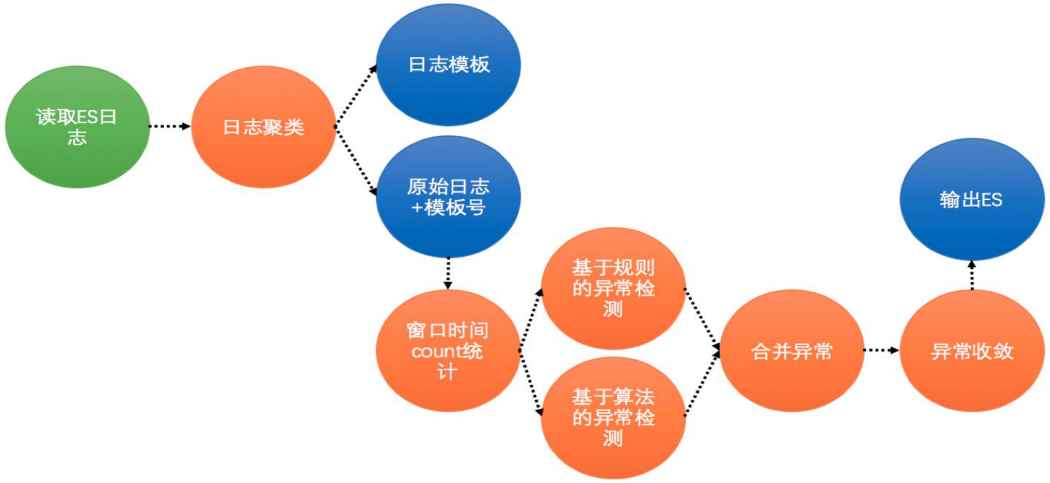


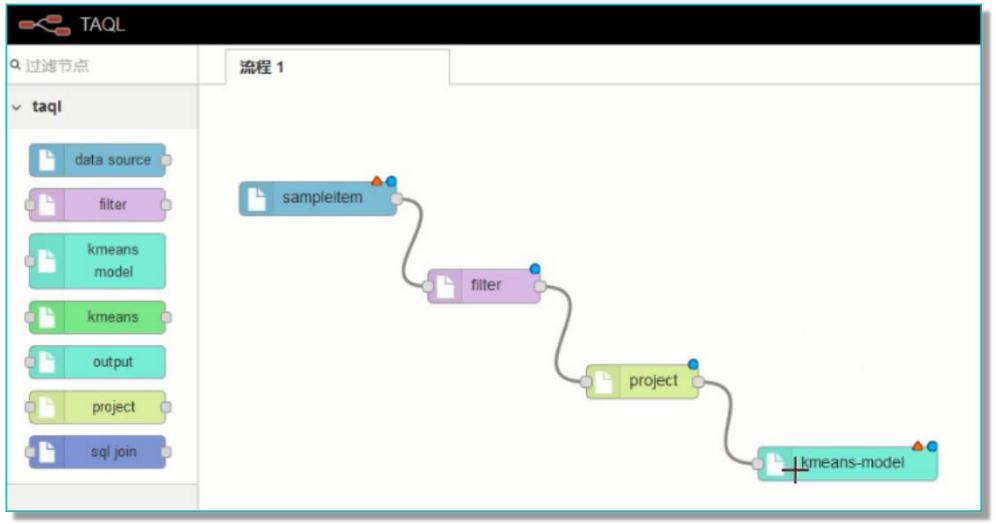
数据探索工具

做为辅助手段的数据探索技术 让运维人员也能灵活的分析数据

01 基于拖拽式的分析流程实现

- 便于领域专家结合不同分析算法搭建分析流程
- 融合了异常检测、聚类、场景挖掘等多种算法
- 支持不同语言开发的算法
- 支持输入数据格式的智能学习





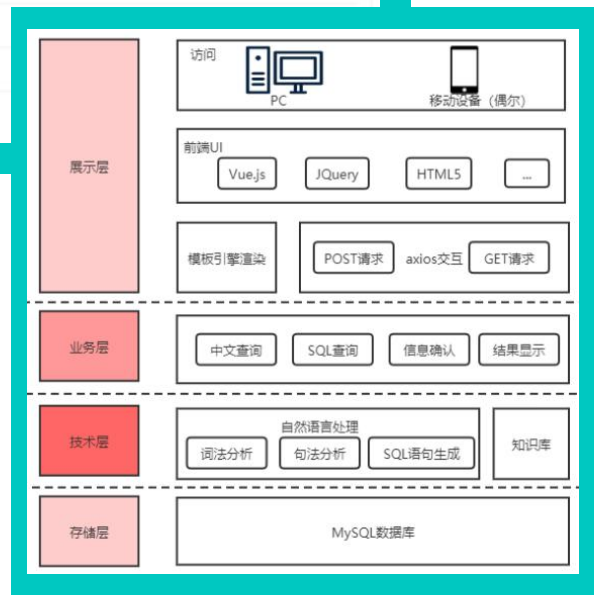
数据探索工具

做为辅助手段的数据探索技术 让运维人员也能灵活的分析数据

02

基于自然语言的问题系统

高易用性，便于运维人员进行个性化数据探索



问题示例

1. 在2019/11/28 11:25发生突增异常的指标有哪些?
2. A应用发生异常次数最多的主机是哪台?
3. B应用告警次数最多的告警种类是什么?
4. 最近一周内存使用率最高的十台主机是哪些?
5. 最近十天发生异常次数最多的应用是什么?
6. 最近一周内失败率最高的应用是哪个?

数据探索工具

做为辅助手段的数据探索技术 让运维人员也能灵活的分析数据

03 面向时间关联的复杂查询

HDFS日志

Log Type	Log Content
E_1	**2019/8/6 15:00 Adding a new node: /default-rack/192.168.0.231:50010**
E_1	2019/8/6 15:01 Adding a new node: /default-rack/192.168.0.232:50010
E_2	**2019/8/6 15:02 Adding new storage ID DS-efe44b9ea549 for DN 192.168.0.231:50010**
E_2	2019/8/6 15:02 Adding new storage ID DS-efe54b9sa352 for DN 192.168.0.232:50010
E_3	2019/8/6 15:03 Number of failed storage changes from 0 to 0
E_4	2019/8/6 15:04 BLOCK* fsync: /hbase/WALs/hadoop5
E_5	**2019/8/6 15:05 BLOCK* registerDatanode: from DatanodeRegistration(192.168.0.231:50010)**

```
SELECT A.*, B.*, C.*
FROM (SELECT * FROM HDFS WHERE LogType = E1) A
INNER JOIN (SELECT * FROM HDFS WHERE LogType = E2) B
ON 0 <= TIMESTAMPDIFF(MINUTE,A.Timestamp, B.Timestamp) <= 5
AND A.IP = B.IP
INNER JOIN (SELECT * FROM HDFS WHERE LogType = E5) C
ON 0 <= TIMESTAMPDIFF(MINUTE,A.Timestamp, C.Timestamp) <= 5
AND 0 <= TIMESTAMPDIFF(MINUTE,B.Timestamp, C.Timestamp) <= 5
AND A.IP = C.IP
```

SQL查询

$$\text{PATTERN } (E_1, E_2, E_5) \text{ WITHIN 5 minute}$$
$$\text{BETWEEN 2016/08/06 15:00 AND 2016/08/06 15:10}$$
$$\text{AND } E_1.IP = E_2.IP$$
$$\text{AND } E_1.IP = E_5.IP$$

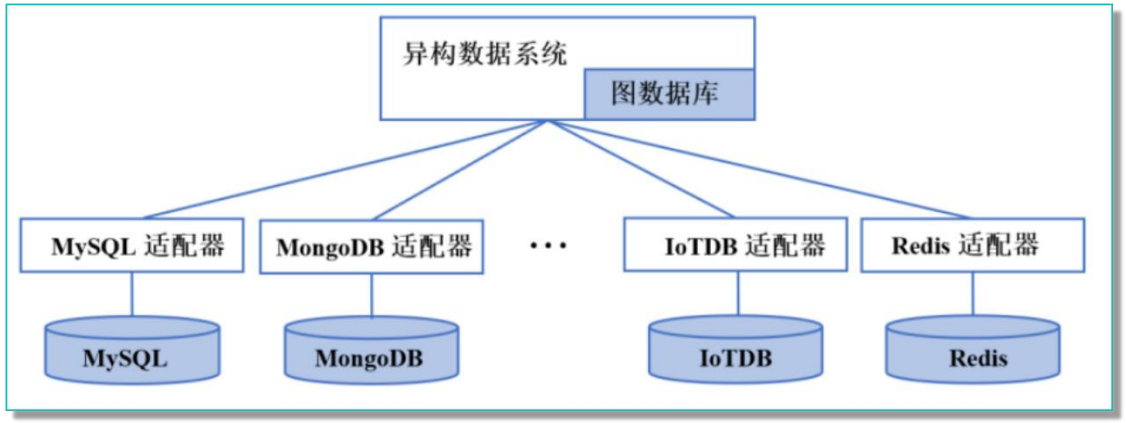
PLQ查询：更为简洁高效

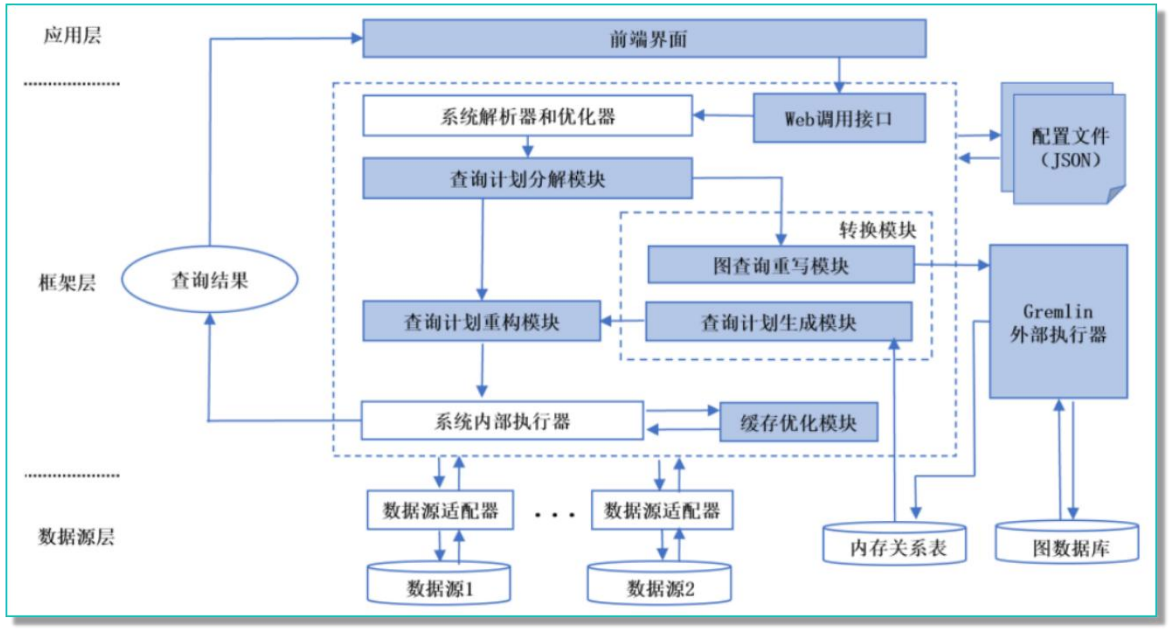
数据探索工具

做为辅助手段的数据探索技术 让运维人员也能灵活的分析数据

04 异构数据统一查询

- 指标数据（时间序列）
- 日志数据（文本序列）
- CMDB数据（图数据）
- 调用链数据（图数据）





总结

总结

01 智能运维中的算法发挥越来越大的作用

02 智能运维中的算法落地仍有大量问题需要解决

03 算法不能一蹴而就，需要有持续优化的能力

04 算法做为一种运维的辅助手段

谢谢！