AIOps Challenge

# 多模态协同的故障识别和分类

翼起飞/刘宽

天翼云科技有限公司 研发二部

2022 CCF国际AIOps挑战赛决赛暨AIOps研讨会

# 团队介绍

AIOps Challenge

## 公司介绍

天翼云科技有限公司（以下简称“天翼云”）是中国电信旗下一家科技型、平台型、服务型公司，以“云网融合、安全可信、专享定制”三大优势向客户提供公有云、私有云、专属云、混合云、边缘云、全栈云服务，满足政府机构、大中小企业数字化转型需求。

作为全球领先的云服务商及运营商云的领军者，天翼云拥有先进的云网基础设施和定制化解决方案，正在成为中国电信全方位建设能力体系的核心承载。天翼云依托自主研发的云平台、运营商央企底蕴与互联网创新机制，为用户提供安全云服务。

## 团队介绍

队伍名称：翼起飞

队伍成员：刘宽、成胜、邢航、向达、吴谦

团队简介：

来自天翼云研发二部云终端基础平台团队，负责包括云电脑、云手机等产品的底层云平台研发，产品已累计发展用户百万级别，服务器规模超3万台。

研发同时还积极践行DevOps及AIOps等理念与技术并参与日常云平台运维工作，在人工智能于云平台领域运维场景落地积累了较多经验。

天翼云

AIOps Challenge

2022 CCF国际AIOps挑战赛决赛
暨AIOps研讨会

# 目录
# CONTENTS



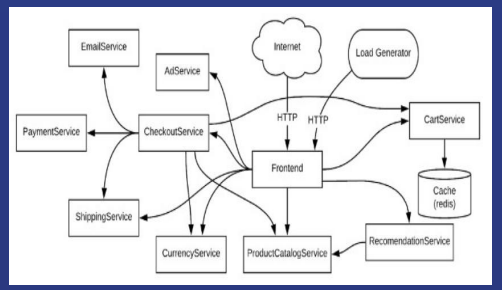
天翼云

第一章节

# 赛题解读

# 赛题解读

AIOps Challenge

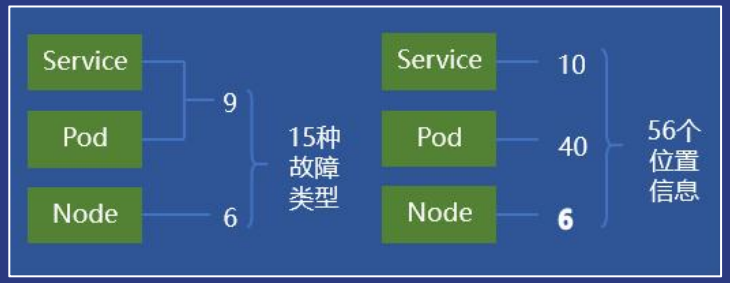



业务架构（动态部署）

动态部署架构

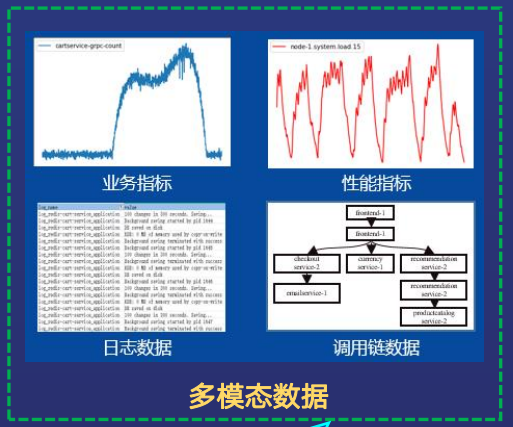
service
10个service
每个service有4个pod

node
6个node
pod动态部署在node上




故障类型 + 故障位置

[cmdb_id, failure_type]




多模态数据

数据挖掘和机器学习

- 提取算法模型
- 提取关键指标、调用链信息
- 提取日志异常模板
- 提取指标相关性、故障分类模型

测评规则

- 故障检测延迟
- 故障检测准确率
- 故障检测召回率
- 故障分类准确率

k*N
预算控制

KPI：可以解析出故障位置和类型信息，按特征分为平稳型、周期型、波动型等

Trace：可以解析出故障位置信息，通过节点间的调用延时做异常检测

Log：可以解析出位置信息，通过提取模板检测很稳定

Service：只部分故障造成业务异常，作为辅助指标

## ◆赛题总结

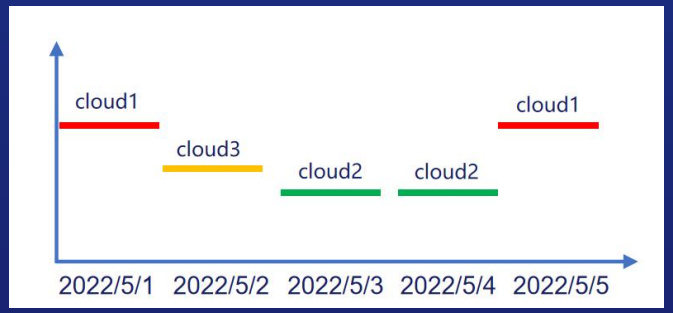
- 考验多模态数据的协同检测，算法通用性，故障检测实效性和准确度等
- 利用多模态数据，实现高效、鲁棒性时序异常检测和故障分类

天翼云

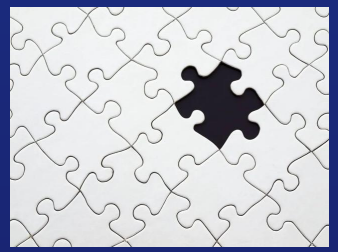
# 难点与挑战

AIOps Challenge

- 多模态数据种类多，数据/计算量大，数据结构存在差异
- 故障持续时间不确定，判断异常并进行训练有难度
- 三套系统，基线存在变化，同时数据存在延时，存在较多毛刺和噪声
- 正负样本不均，训练数据不足，数据缺失等
- 在所有指标数据中，采样周期存在不一致的情况。绝大多数都是1分钟一采样，个别指标是5分钟一采样。

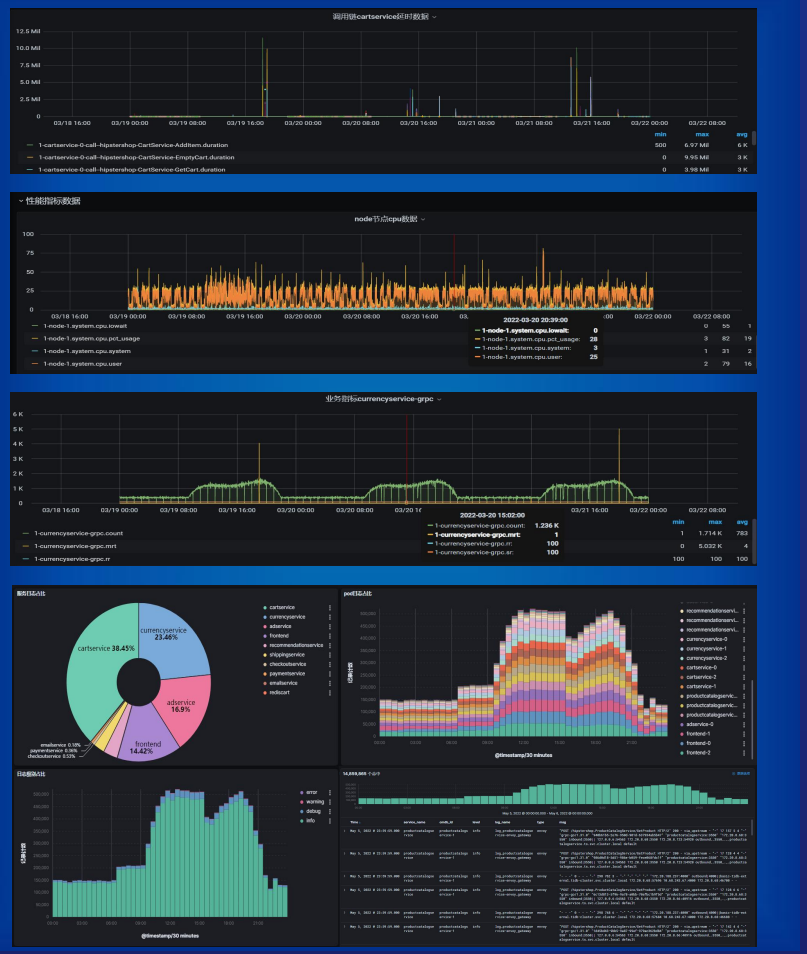



基线变化



数据缺失


正负样本极度不均衡



◆ **解题思路：**

- 尽量实践多模态数据，充分利用各种模态数据的特征，数据特征越丰富，可以给出更精确的结果
- 践行“知识+数据+算法+算力”

天翼云

第二章节

# 整体方案介绍

# 总体架构(多模态数据+多模态模型)

AIOps Challenge

离线数据

统计算法+无监督算法 → 迭代计算 → 输出指标与故障相关性数据

单指标检测训练

日志格式化 → 提取模板 → 模板集

日志检测训练

获取服务调用拓扑 → 计算调用延时 → 生成统计特征

调用链检测训练

指标降维 → 特征工程 → 数据训练 → 输出故障分类模型

多指标检测训练

在线数据

数据预处理 → 前置滤波 → 无监督算法+有监督算法 → 检测异常

单指标检测

日志格式化 → 匹配模板 → 检测异常

日志检测

计算完整调用链 → 计算z-score值 → 检测异常

调用链检测

数据预处理 → 故障检测

多指标检测

“检测模型——故障类型——权重”矩阵

| | 单指标检测 | 日志检测 | 调用链检测 | 多指标检测 |
|---|---|---|---|---|
| 故障类型1 | 准确率 | 准确率 | 准确率 | 准确率 |
| 故障类型2 | 准确率 | 准确率 | 准确率 | 准确率 |
| 故障类型3 | 准确率 | 准确率 | 准确率 | 准确率 |
| … | … | … | … | … |

“检测模型——故障位置——权重”矩阵

| | 单指标检测 | 日志检测 | 调用链检测 | 多指标检测 |
|---|---|---|---|---|
| node | 准确率 | 准确率 | 准确率 | 准确率 |
| service | 准确率 | 准确率 | 准确率 | 准确率 |
| pod | 准确率 | 准确率 | 准确率 | 准确率 |

缓存结果 → 计算评分 → 输出故障

故障位置和故障类型 [cmdb_id, failure_type]

天翼云

第三章节

# 详细方案与实现

# 单指标异常检测

AIOps Challenge

- 3-sigma $\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}}$ $P(|x - \overline{x}| > 3\sigma) \leq 0.003$
- polynomial
  $h_\theta(x) = \theta_0 x^0 + \theta_1 x^1 + \cdots + \theta_n x^n = X \cdot \theta$
- xgboost
  $Obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant$

离线学习

离线数据

kpi

mertic

数据格式化

数据分类

周期型

不稳定型

平稳型

异常检测

疑似异常

关键特性

关联关系

相关性

降维

相关性矩阵

专家经验

算法配置

指标库

示例

3-sigma + 多项式回归

3-sigma + xgboost

3-sigma

在线检测

在线流式数据

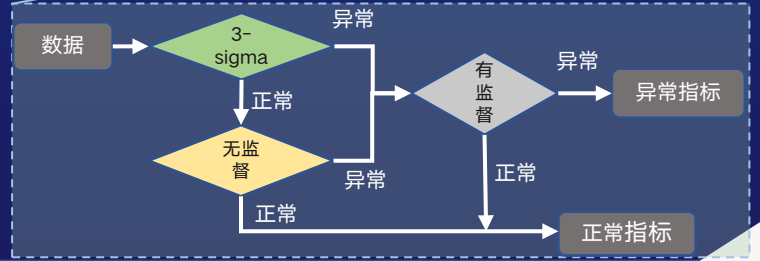
数据格式化

前置滤波器

异常检测

输出异常

检测算法逻辑




天翼云

# 单指标异常检测——筛选与故障相关指标

AIOps Challenge

**难点** 指标种类较多，全量指标检测的内存占用与耗时不可接受

**应对** 根据“指标-故障”相关性算法+指标间Pearson相关性系数+专家经验提取少量对故障定位具有代表性的指标进行检测，实现指标降维

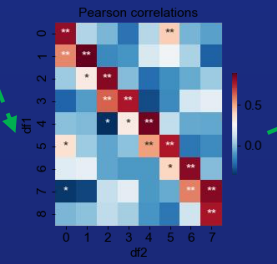
**效果** 降低内存占用和计算开销，极大缩短检测时间

离线训练，计算疑似异常指标与故障类型的相关性

故障标签

| 故障类型 | 故障总数 | 异常KPI名称 | 最小值 | 平均值 | 异常KPI次数 | 相关性 |
|---|---|---|---|---|---|---|
| node节点CPU故障 | 13 | system.cpu.user | 29.67 | 55.22 | 13 | 1 |
| node节点CPU故障 | 13 | system.cpu.pct_usage | 39.33 | 60.85 | 13 | 1 |
| node节点CPU故障 | 13 | system.load.1 | 11 | 11.5 | 4 | 0.31 |
| node 磁盘空间消耗 | 14 | system.io.await | 69.5 | 126.74 | 13 | 0.93 |
| node 磁盘空间消耗 | 14 | system.io.w_await | 85.75 | 132.05 | 13 | 0.93 |
| node 磁盘空间消耗 | 14 | system.io.avg_q_sz | 57 | 63.56 | 13 | 0.93 |
| node 磁盘空间消耗 | 14 | system.mem.free | 187.67 | 3158.84 | 11 | 0.79 |
| node 磁盘空间消耗 | 14 | system.io.w_s | 226.67 | 522.14 | 10 | 0.71 |
| k8s容器cpu负载 | 21 | container_memory_rss | 23848434.77 | 42785789.32 | 19 | 0.9 |
| k8s容器cpu负载 | 21 | container_cpu_cfs_periods | 478.92 | 592.43 | 19 | 0.9 |
| k8s容器cpu负载 | 21 | container_cpu_cfs_throttled_periods | 498.67 | 590.98 | 19 | 0.9 |
| k8s容器cpu负载 | 21 | container_memory_working_set_MB | 40.14 | 80.27 | 18 | 0.86 |
| k8s容器cpu负载 | 21 | container_cpu_user_seconds | 11 | 20.77 | 18 | 0.86 |
| k8s容器cpu负载 | 21 | container_cpu_cfs_throttled_seconds | 537.33 | 699.68 | 18 | 0.86 |
| k8s容器cpu负载 | 21 | container_memory_usage_MB | 54.67 | 87 | 17 | 0.81 |
| k8s容器cpu负载 | 21 | container_threads | 38.5 | 54.52 | 17 | 0.81 |
| k8s容器cpu负载 | 21 | container_cpu_usage_seconds | 11.5 | 19.6 | 16 | 0.76 |

计算指标间相关性




构造“指标-故障”评分矩阵

| 评分矩阵 | system.cpu.pct_usage | system.io.w_await | container_cpu_cfs_periods | …… |
|---|---|---|---|---|
| node节点CPU故障 | 100 | 0 | 0 | … … |
| node 磁盘空间消耗 | 0 | 93 | 0 | … … |
| k8s容器cpu负载 | 0 | 0 | 90 | … … |
| k8s容器写io负载 | 0 | 0 | 0 | … … |
| …… | | | | … … |

“指标-故障”相关性算法筛选出与故障类型最具关联的指标

皮尔逊相关系数用来衡量不同KPI指标间的线性相关程度，进一步减少指标检测数，降低计算开销，提升检测效率

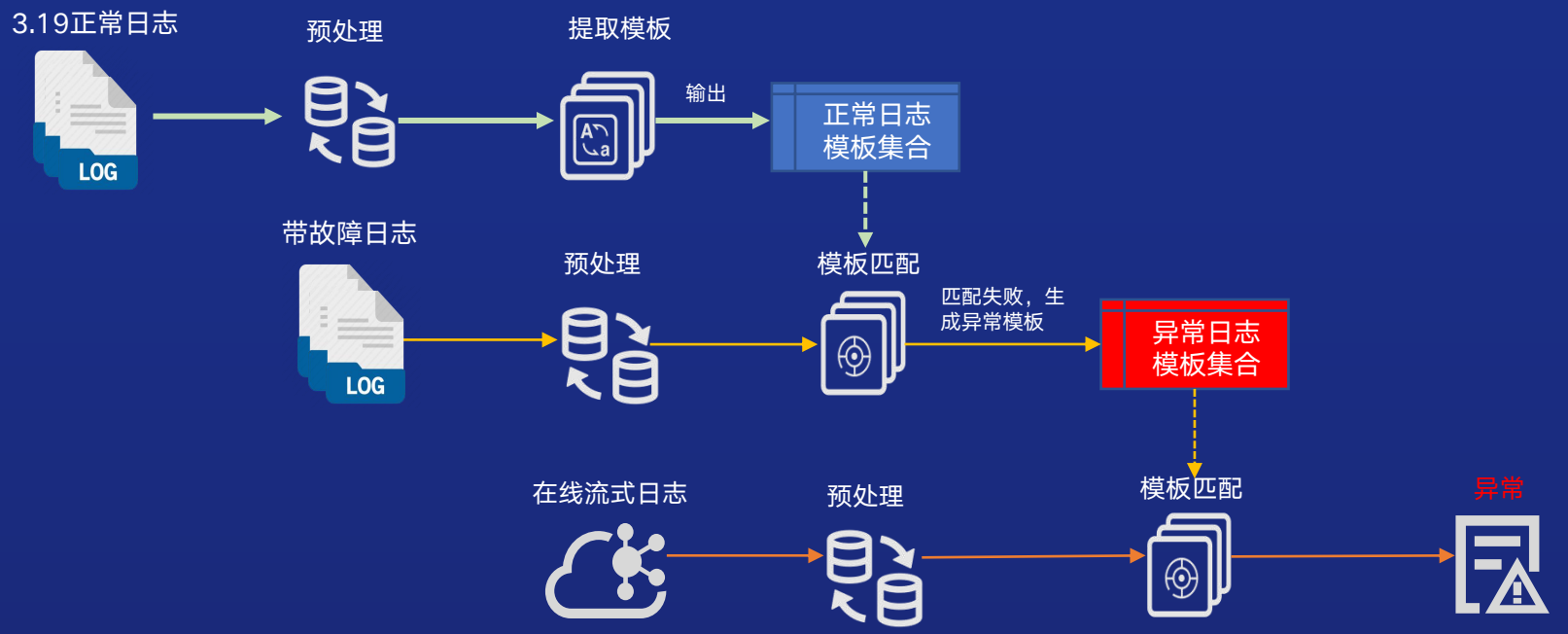
$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

可考虑结合专家经验进一步减少指标数量，降低计算量和内存占用……

天翼云

# 日志异常检测

AIOps Challenge




## ◆ 难点与应对

- 在线日志数据存在持续更新升级，是不断变化的，训练模型很难应对这些变化的数据和噪声。本算法采用正常模板+异常模板双重匹配检测。
- 日志检测结果并不包括完整的故障信息，需结合其它指标共同检测。

**原生日志**

""POST /api/v2/spans HTTP/1.1"" 202 - via_upstream - ""-"" 306 0 2 2 ""-"" ""okhttp/3.14.7"" ""01f97f5c-c3c7-9a69-87ec-fec2dc0bbf27"" ""jaeger-collector:9411"" ""172.20.88.16:9411"" outbound|9411||jaeger-collector.ts.svc.cluster.local 172.20.8.99:38914 10.68.243.50:9411 172.20.8.99:38030 - default

""POST /hipstershop.CartService/GetCart HTTP/2"" 200 - via_upstream - ""-"" 43 59 1 0 ""-"" ""grpc-go/1.31.0"" ""b619aff0-1c34-909e-8483-888895cb0b73"" ""cartservice:7070"" ""172.20.8.72:7070"" inbound|7070|| 127.0.0.6:36819 172.20.8.72:7070 172.20.8.123:52762 outbound_.7070_._.cartservice.ts.svc.cluster.local default

Request starting HTTP/2 POST http://cartservice:7070/hipstershop.CartService/AddItem application/grpc

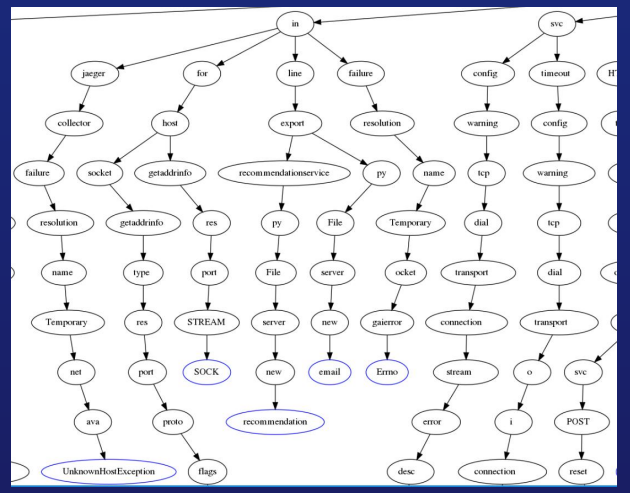
""POST /cart HTTP/1.1"" 302 - via_upstream - ""-"" 32 0 9 8 ""-"" ""k6/0.26.2 (https://k6.io/)"" ""16f0dda4-0c41-92e5-b288-17084e8c02e5"" ""frontend.ts:80"" ""172.20.8.66:8080"" inbound|8080|| 127.0.0.6:45296 172.20.8.66:8080 172.20.188.235:45362 - default

Request starting HTTP/2 POST http://cartservice:7070/hipstershop.CartService/EmptyCart application/grpc

.....

提取模板

| 日志模板 | ID |
|---|---|
| cluster svc outbound local HTTP POST upstream default ts via jaeger collector api spans okhttp | 1 |
| cluster svc outbound local HTTP POST upstream default ts via grpc hipstershop go | 2 |
| HTTP POST grpc hipstershop CartService cartservice http application Request starting AddItem | 3 |
| HTTP POST upstream default ts via inbound io https frontend cart | 4 |
| HTTP POST grpc hipstershop CartService cartservice http application Request starting EmptyCart | 5 |
| ..... | ..... |



ft_tree示例

- 使用ft_tree增量式地从系统日志消息中学习模板
- 通过模板可以将日志消息与一个特定消息模板匹配，并使用该消息模板的ID来表示该日志消息

天翼云

# 基于Trace的异常检测与定位

AIOps Challenge

## 离线训练阶段

离线Traces数据 → 获取服务调用静态拓扑图 → 计算调用链的时延作为时序指标 → 生成指标的统计学特征 → >_

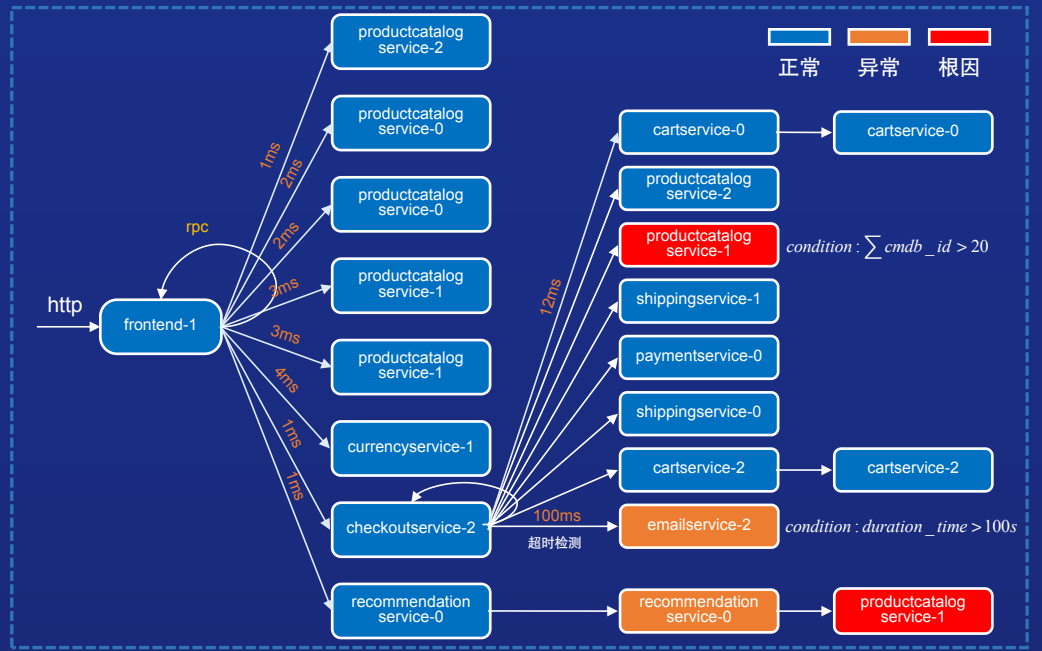
## 在线检测阶段

在线Trace数据 → 根据trace_id获取完整调用链 → 计算每条调用链指标的z-score值 → 判断所有调用超时的cmdb_id → 根据调用拓扑图定位故障cmdb_id

$$z_{score} = \frac{x-\mu}{\sigma}\left(u = \sum_{i=1}^{n} x_i / n, \sigma^2 = \sum_{i=1}^{n} (x_i - x)^2 / n\right)$$

| timestamp | cmdb_id | span_id | trace_id | duration | type | status_code | parent_span |
|---|---|---|---|---|---|---|---|
| 1651681102426 | frontend-1 | 4124d4e5ad727d4c | 00104d1aaac553c380 | 47249 | http | 0 | nan |
| 1651681102427 | productcatalogservice-2 | 2d933bbfc8d6c0c3 | 00104d1aaac553c380 | 4731 | rpc | 0 | 344f50aacb3fdde3 |
| 1651681102427 | frontend-1 | 344f50aacb3fdde3 | 00104d1aaac553c380 | 6420 | rpc | 0 | 4124d4e5ad727d4c |
| 1651681102433 | frontend-1 | 5e7288d5d6735b43 | 00104d1aaac553c380 | 1746 | rpc | 0 | 4124d4e5ad727d4c |
| 1651681102434 | currencyservice-1 | 49909919aa8bdc2e | 00104d1aaac553c380 | 107 | telemetry | 0 | 5e7288d5d6735b43 |
| 1651681102435 | frontend-1 | b08ca46460e041f7 | 00104d1aaac553c380 | 2097 | rpc | 0 | 4124d4e5ad727d4c |
| 1651681102436 | cartservice-2 | bc634ff35006e24f | 00104d1aaac553c380 | 0 | http | 200 | b08ca46460e041f7 |
| 1651681102436 | cartservice-2 | 6a94240f244abb4d | 00104d1aaac553c380 | 0 | db | ok | bc634ff35006e24f |
| 1651681102437 | frontend-1 | 2d4b24f1375b0a53 | 00104d1aaac553c380 | 2081 | rpc | 0 | 4124d4e5ad727d4c |
| 1651681102438 | currencyservice-2 | a82ba8c224da46dc | 00104d1aaac553c380 | 96 | telemetry | 0 | 2d4b24f1375b0a53 |
| 1651681102439 | frontend-1 | 8d6b463c1c22f5dc | 00104d1aaac553c380 | 3488 | rpc | 0 | 4124d4e5ad727d4c |
| 1651681102440 | recommendationservice-2 | c9bc61456d1bcb70 | 00104d1aaac553c380 | 1539 | | 0 | 9f64892a70ada82f |
| 1651681102440 | recommendationservice-2 | 9f64892a70ada82f | 00104d1aaac553c380 | 1847 | | 0 | 8d6b463c1c22f5dc |

父节点 → 子节点




## ◆ 难点与应对

- 难点1：调用链数据量大，平均每分钟有6000+条数据。
- ✓ 应对：采用独立的进程处理调用链数据，并选用计算量少、处理速度快、基于统计学的异常检测算法z-score。
- 难点2：干扰噪声数据多，存在系统正常运行过程中由于网络波动造成的偶发极端异常数据。
- ✓ 应对：对数据先进行去噪处理，通过阈值过滤掉极端异常的干扰噪声，并结合异常调用持续时长，过滤掉时长过短的噪声数据。
- 难点3：一次故障往往会造成多个节点异常，难以定位到根因。
- ✓ 应对：在一个时间窗口内统计每个cmdb_id出现异常调用的频数，根据频数的多少，并结合部署架构和调用拓扑定位根因。

天翼云

# 多指标检测

AIOps Challenge

**思路**：故障识别和分类转化为多分类问题处理

故障识别和分类

↓

多分类

↓

多标签多分类

↓

单标签多分类

Service — 9
Pod — 9
Node — 6
15种故障类型

Service — 10
Pod — 40
Node — 6
56个位置信息

**实践**：以结果为导向

尝试过的算法：
- Xgboost
- Lightgbm
- LSTM

基于xgboost的故障识别和分类

存在的问题：
1、时序变化信息
2、多套系统及动态拓扑

特征选取
兼容、高质数据

多系统特征兼容
istio-egressgateway
istio-ingressgateway
container_

特征降维
GBDT降维

时序统计特征工程
时间窗口统计计算

Xgboost模型训练
参数调优
离线训练

Xgboost故障识别和分类模型
在线故障识别和分类模型

**优点**：对已知故障检测准确；完全基于训练，不需人为分析指标与故障的对应关系，通用性很好

**缺点**：对未知、数量少或注入故障大小差距较大等情况的故障识别率低

# 工程实现

AIOps Challenge

**实现难点**

- 数据量大，检测耗时
- 单维度检测结果不够准确
- 不同指标检测要用不同算法，既不能互相干扰，又需要对各自结果做综合判断
- 同一异常窗口内不同检测算法可能检测到不同故障结果，需要合理定位根因
- 数据量大，保存在本地则读写性能差，保存在内存则内存消耗大

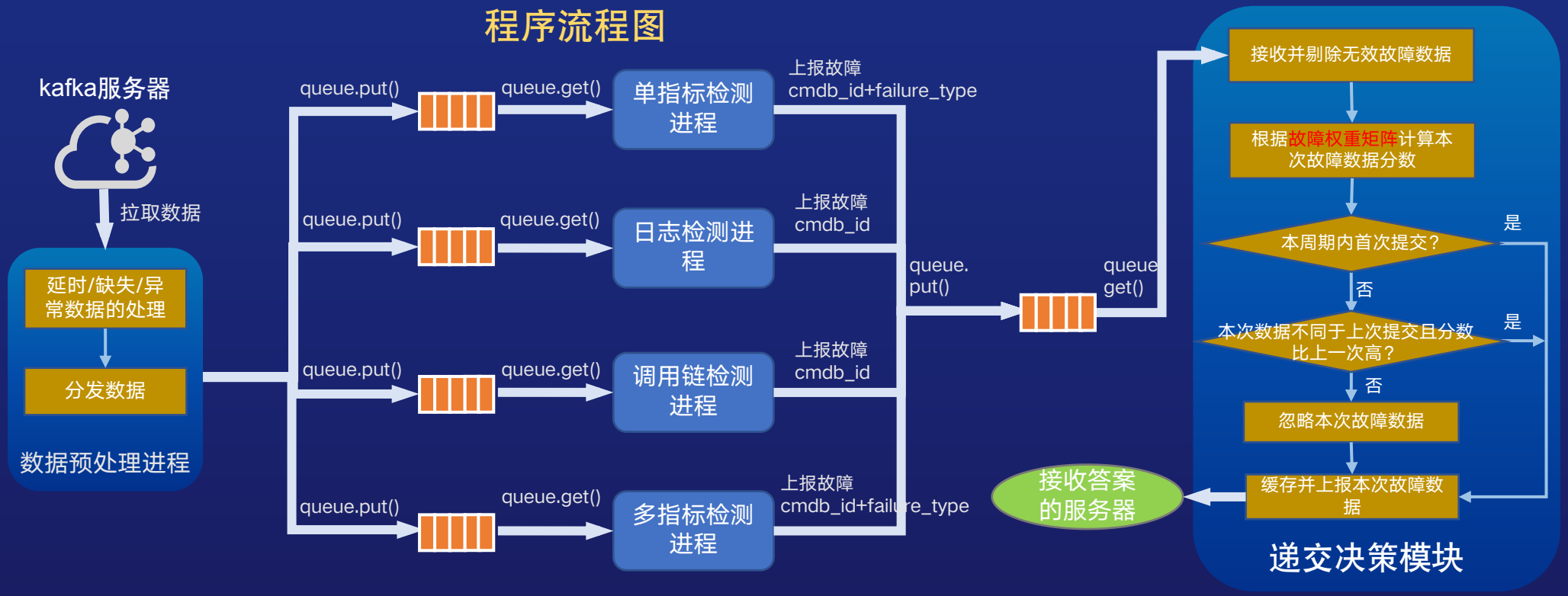
**工程应对**

- 采用多进程并行检测架构
- 分单指标/多指标/日志/调用链四维度四进程并行检测，互不影响
- 将不同检测结果结合权重矩阵等进行量化计分，提交得分最高的故障数据
- 数据保存在内存中并定期清理，既保证读写效率，又减少内存占用

**架构优点：**

- 有效解决数据量大处理耗时问题
- 多进程并行检测，提升检测效率
- 可插拔式架构，可以灵活扩展新的模态数据和检测场景及模型
- 多维度综合分析得到上报数据，提升准确率
- 进程间采用队列传输数据，确保数据连续性和完整性
- 实现进程异常处理及自动拉起机制，保证稳定运行
- 不同模块进程互不影响，一个异常不影响其他的运行
- 根据需要的时间窗口对缓存数据进行实时清理

## 程序流程图




天翼云

第四章节

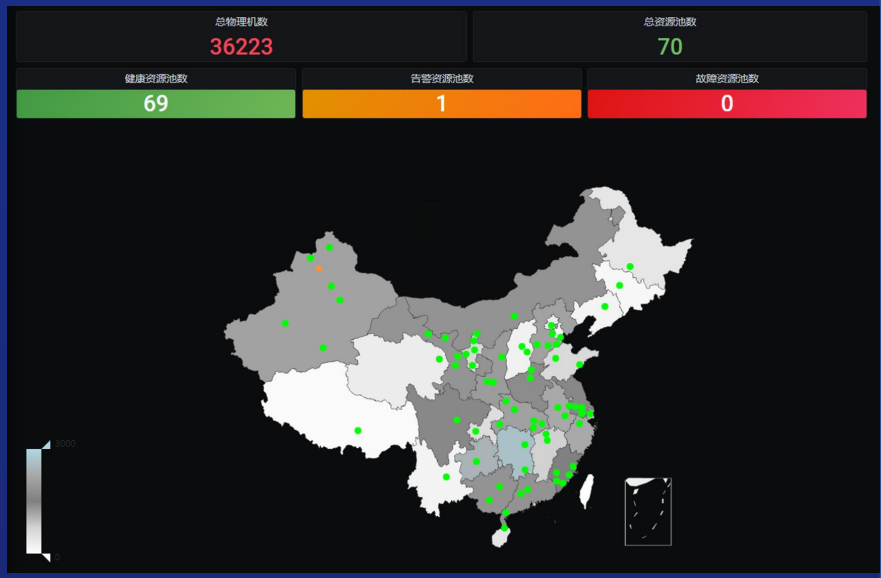
# 算法效果与展望

# 算法效果

AIOps Challenge

## 复赛阶段测评名次


## 超大规模生产实践




本检测算法的基础算法在部门业务线中已覆盖 3w+ 服务器，经过了大量监控数据打磨，该算法在异常检测和运维监控领域具有较广泛的应用性。内部一些应用的场景如：

- 告警聚合/抑制
- 磁盘故障预测
- 动态告警阈值
- 存储容量预测

## 算法通用性说明

- 单指标的故障定位，通过指标与故障相关性算法生成的评分矩阵进行综合评分得出，其自动计算相关性的方法，不需要人为干预，适用各种场景的指标与故障关联性计算
- 基于xgboost的故障识别和分类模型，只需输入指定的指标数据，即可输出对应的故障位置和类型，完全基于训练，不需要人为分析指标与故障的对应关系。
- 日志指标检测算法使用的是ft_tree，方案中，对日志内容进行了一些处理，使其降低对日志内容和格式的限制要求，适用所有具有模板性质的日志的异常检测。
- 调用链异常检测是基于统计学算法z-score计算时延指标的波动情况，通过分析时延偏离均线的程度判断异常出现的可能性。只需要通过离线学习各个时延指标的均值和方差即可进行在线异常检测。算法计算量小，检测速度快，比较适用于数据量大的调用链异常检测场景。

天翼云

# 总结与展望

AIOps Challenge

## 方案亮点总结：

- 实现对多种模态数据的充分利用，优于传统基于单模态数据的AIOps方案
- 优异的工程化能力，在使用多模态数据和模型的前提下兼具优秀的检测性能和稳定性
- 算法自动提取关键特征并进行降维，算法通用性强且兼具鲁棒性，部分算法已在内部超大规模集群中落地

## 改进思考与展望：

- 实际检测中，网络类故障准确性相对较低，需要设计更有效的算法
- 窗口期内可能存在多个故障告警，存在多次递交的情况，占用递交次数，需要研究进一步降低误报率的方法。
- 对调用链和日志的异常检测有待进一步深入研究和优化

天翼云

AIOps Challenge

2022 CCF国际AIOps挑战赛决赛暨AIOps研讨会

# THANKS