AIOps Challenge 智能运维挑战赛

2021国际AIOps挑战赛决赛
暨AIOps创新高峰论坛

# 云环境下商业银行应用系统的故障实时检测与根因定位

## 队伍名称： bocoiops

答辩选手：陈晓峰、崔世彬、何宁

# 团队介绍

## 亿阳信通

成立于1995年，是国家科技部首批认定的全国重点高新技术企业和全国创新型企业，主要从事OSS系统、企业IT运营支撑系统、信息安全等方面的行业应用软件开发、解决方案提供和技术服务。经过二十多年的自主创新和技术积累，已发展为中国最大的应用软件开发商和行业解决方案提供商之一，在OSS领域居于龙头地位，解决方案和市场占有率均排名第一

## 技术架构部

作为公司技术核心部门，主要有以下职责：
1. 研究行业最新技术发展趋势，规划公司技术发展路线
2. 制定和实施公司重大技术决策和技术方案
3. 负责新技术的预研、创新及在试点产品中的落地、推广
4. 对各事业部项目中的关键问题和技术难题提供技术保障

目前，主导的云化架构，大数据方案及多款AIOps产品：故障预测、告警关联、根因分析等均已在运营商多省成功实施

**陈晓峰** 数据分析、算法设计 亿阳信通CTO

**崔世彬** 数据分析、算法设计 数据架构师

**何　宁** 算法设计 AI算法工程师

**吴　琼** 算法设计 AI算法工程师

第一章节

# 挑战与方案

# 挑战1：故障定位

AIOps Challenge 智能运维挑战赛 | 科创会 NICON

## 一. 基于Metric性能指标的故障定位

### 1. 采用移动差分的故障检测定位方法

算法核心思想：

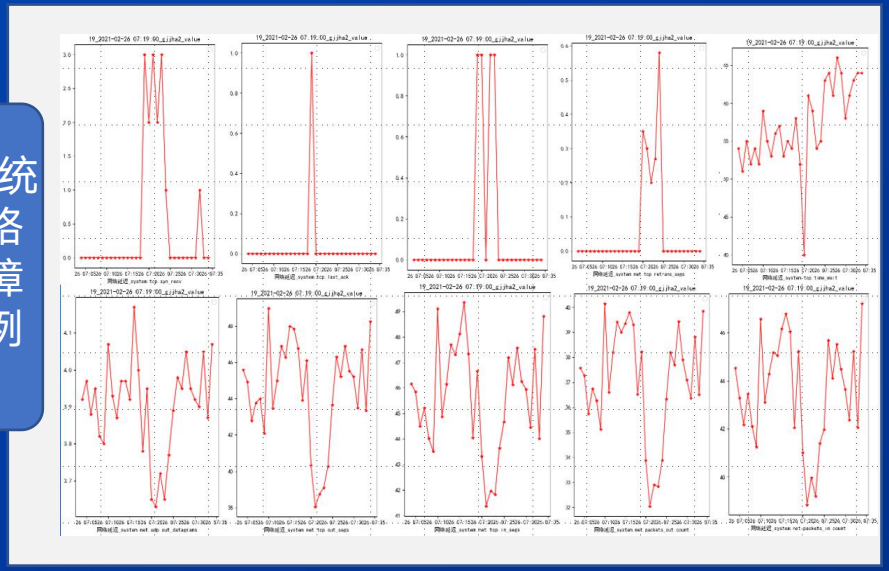
① 对移动指标按时序进行差分计算

② 对指标去量纲，研究指标波动。

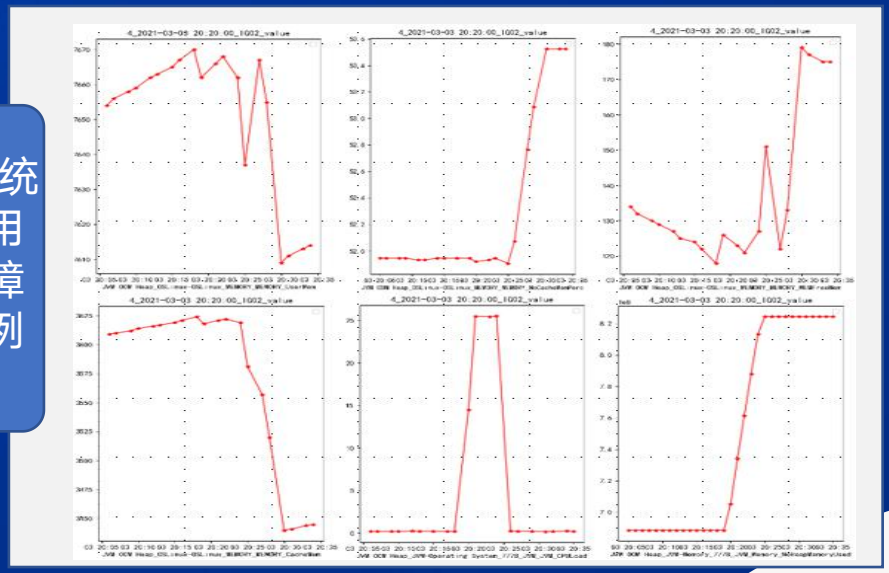
**差分数据**：1. 基于临近点　2. 基于移动均值
**波动数据**：1. 斜率值　2. 当前值与移动均值的比值
**判断依据**：根据波动的变化程度进行异常判断，若指标存在长时间异常波动则判别为干扰噪声。

**效果：**
**① 更容易发现特征弱故障**
**② 故障开始时，数据变化特别明显**
**③ 提高信噪比，更容易过滤背景干扰噪声，解决静态阈值判断能力的不足**

A系统网络故障示例



B系统应用故障示例



# 挑战1：故障定位

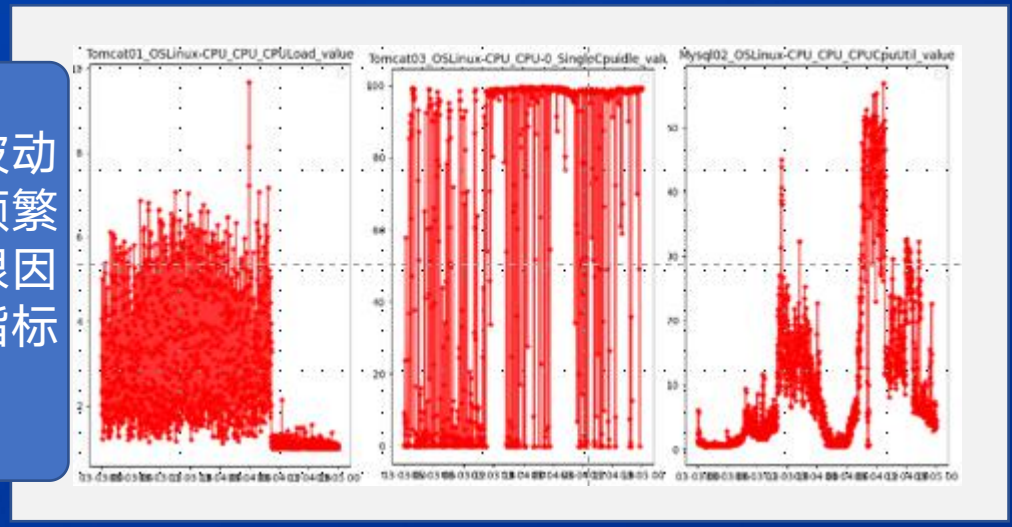
## 2. 采用根因的加权故障定位法

公式：

$$V = \Sigma Vi \cdot wi$$

移动差分定位检测出异常指标后，再将每个指标的波动程度乘以该指标的故障权重，最终加和得到该时点对应的故障得分。

当故障得分产生较大的波动时则判定为故障，如果同一时间产生多个故障对象，则选择得分最高的为该时点的故障对象。




**效果：**

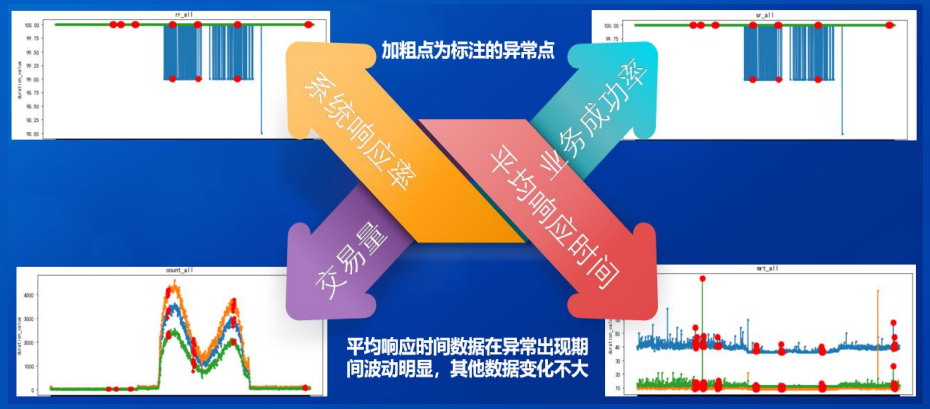
**有效地减少波动频繁根因、背景干扰噪声等因素引发的误报，提高查准率。**

# 挑战1：故障定位

## 二. 基于黄金业务指标的故障定位

### 采用移动差分的故障检测定位方法

A系统




| timestamp | mrt |
|---|---|
| 2021/2/26 2:47 | 95 |
| 2021/2/26 2:48 | 67 |
| 2021/2/26 5:59 | 142 |
| 2021/2/26 6:00 | 61 |
| 2021/2/26 10:09 | 96 |
| 2021/2/26 10:10 | 57 |
| 2021/2/26 11:11 | 6122 |
| 2021/2/26 11:12 | 57 |
| 2021/2/26 19:10 | 57241 |
| 2021/2/26 19:11 | 65 |
| 2021/2/26 22:03 | 145 |
| 2021/2/26 22:04 | 61 |
| 2021/2/26 23:28 | 95 |
| 2021/2/26 23:29 | 62 |

根据平均响应时间抓取应用故障。

### 黄金业务指标可以抓取部分应用故障以及网络故障

B系统


| timestamp | rr | sr | mrt |
|---|---|---|---|
| 2021/3/4 3:10 | 225 | 350 | 74764.25 |
| 2021/3/4 3:13 | 200 | 200 | 1549.33 |
| 2021/3/4 7:55 | 786.99 | 1078.667 | 111172 |
| 2021/3/4 7:56 | 867.58 | 1100 | 74491.69 |
| 2021/3/4 7:57 | 1016.7 | 1100 | 17518.07 |
| 2021/3/4 10:10 | 675.57 | 679.3922 | 253781.5 |
| 2021/3/4 10:11 | 0 | 0 | 660000 |
| 2021/3/4 10:21 | 73.71 | 53.92414 | 561462.3 |
| 2021/3/4 10:23 | 296.52 | 290.8782 | 437970.7 |
| 2021/3/4 10:24 | 452.22 | 475.0067 | 363808.3 |
| 2021/3/4 10:25 | 621.31 | 641.7401 | 238637.8 |
| 2021/3/4 10:26 | 853.92 | 816.9821 | 81418.02 |
| 2021/3/4 10:28 | 1090.32 | 1100 | 13138.19 |
| 2021/3/4 10:53 | 649.14 | 1100 | 92021.8 |

应用故障

数据库网络故障

# 挑战1：故障定位

## 三. 基于调用链指标的故障定位

A系统Trace数据只有单一调用链

```
调用链个数：3337369
存在父节点的调用链个数 0
```

B系统Trace数据调用链完整

```
2021-03-03 00:08:15.063 -- IG01 60000
2021-03-03 00:08:15.063   |-- IG01 60000
2021-03-03 00:08:15.541 -- IG02 533
2021-03-03 00:08:15.541   |-- IG02 532
2021-03-03 00:08:15.463     |-- Tomcat02 531
2021-03-03 00:08:15.464       |-- Tomcat02 0
2021-03-03 00:08:15.465       |-- Tomcat02 0
2021-03-03 00:08:15.466       |-- Tomcat02 0
2021-03-03 00:08:15.467       |-- Tomcat02 0
2021-03-03 00:08:15.467       |-- Tomcat02 525
2021-03-03 00:08:15.784         |-- MG01 523
2021-03-03 00:08:15.784           |-- MG01 523
2021-03-03 00:03:02.137             |-- dockerB1 516
2021-03-03 00:03:02.139               |-- dockerB1 1
2021-03-03 00:03:02.141               |-- dockerB1 1
2021-03-03 00:03:02.144               |-- dockerB1 1
2021-03-03 00:03:02.146               |-- dockerB1 0
2021-03-03 00:03:02.148               |-- dockerB1 0
2021-03-03 00:03:02.151               |-- dockerB1 1
2021-03-03 00:03:02.152               |-- dockerB1 10
2021-03-03 00:08:13.285                 |-- MG02 7
2021-03-03 00:08:13.285                   |-- MG02 6
2021-03-03 00:03:02.164               |-- dockerB1 1
2021-03-03 00:03:02.167               |-- dockerB1 1
2021-03-03 00:03:02.169               |-- dockerB1 1
2021-03-03 00:03:02.171               |-- dockerB1 1
2021-03-03 00:03:02.173               |-- dockerB1 1
2021-03-03 00:03:02.176               |-- dockerB1 1
2021-03-03 00:03:02.179               |-- dockerB1 1
2021-03-03 00:03:02.180               |-- dockerB1 10
2021-03-03 00:08:15.833                 |-- MG01 7
2021-03-03 00:08:15.833                   |-- MG01 7
2021-03-03 00:03:02.192               |-- dockerB1 1
2021-03-03 00:03:02.194               |-- dockerB1 0
2021-03-03 00:03:02.196               |-- dockerB1 0
```

### 1. 对Trace数据做N分位值处理

本次竞赛的调用链数据存在长短不一、分类不固定、数据质量不好等问题。针对这些问题，我们添加N分位数值进行处理。

通过耗时量的N分位数与最大最小值，平均值中位数方差等数据进行联合对比，从而避免调用链丢失的影响 。对比范围包括自身的耗时，自分位的耗时，节点及以下节点耗时，并形成相应的时序数据。

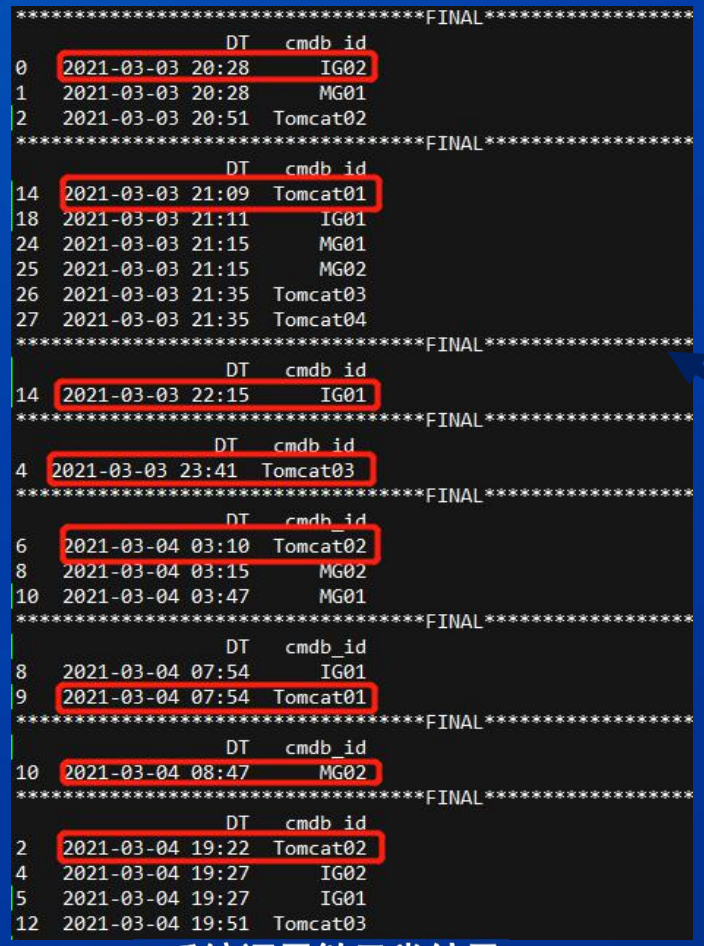
| dp65 | dp6rmpct | dp75 | dp7rmpct | dp85 | dp8rmpct | dp95 | dp9rmpct | dsum | dsmrmpct | csum | csmrmpct | dmean | dmermpct | cmean | cmermpct |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 44.15 | 1.05 | 53 | 0.95 | 103.3 | 0.56 | 1068.45 | 1.02 | 33958 | 0.67 | 289 | 0.68 | 146.3707 | 0.89 | 1.24569 | 0.91 |
| 46 | 1.08 | 66.75 | 1.18 | 550.35 | 2.99 | 1081.85 | 1.03 | 51341 | 1.05 | 300 | 0.72 | 213.9208 | 1.33 | 1.25 | 0.92 |
| 40 | 0.94 | 48 | 0.83 | 93.9 | 0.43 | 1022.65 | 0.98 | 38914 | 0.8 | 355 | 0.87 | 135.1181 | 0.82 | 1.232639 | 0.91 |
| 39 | 0.93 | 46 | 0.83 | 54.95 | 0.29 | 410.85 | 0.4 | 27100 | 0.59 | 389 | 1 | 87.98701 | 0.56 | 1.262987 | 0.95 |
| 38 | 0.91 | 45 | 0.83 | 61.05 | 0.36 | 807.75 | 0.84 | 29137 | 0.7 | 429 | 1.11 | 99.78425 | 0.67 | 1.469178 | 1.1 |
| 39.65 | 0.95 | 48 | 0.88 | 76 | 0.44 | 513.4 | 0.55 | 19701 | 0.49 | 279 | 0.72 | 88.74324 | 0.61 | 1.256757 | 0.93 |

N分位值处理

# 挑战1：故障定位

## 2.采用移动差分方法定位调用链异常

根据上一步骤中形成的时序数据进行异常检测，检测调用链上在不同百分位情况下的波动异常，从而定位故障时点。



```
**********FINAL**********
                 DT   cmdb_id
0   2021-03-03 20:28      IG02
1   2021-03-03 20:28      MG01
2   2021-03-03 20:51  Tomcat02
**********FINAL**********
                 DT   cmdb_id
14  2021-03-03 21:09  Tomcat01
18  2021-03-03 21:11      IG01
24  2021-03-03 21:15      MG01
25  2021-03-03 21:15      MG02
26  2021-03-03 21:35  Tomcat03
27  2021-03-03 21:35  Tomcat04
**********FINAL**********
                 DT   cmdb_id
14  2021-03-03 22:15      IG01
**********FINAL**********
                 DT   cmdb_id
4   2021-03-03 23:41  Tomcat03
**********FINAL**********
                 DT   cmdb_id
6   2021-03-04 03:10  Tomcat02
8   2021-03-04 03:15      MG02
10  2021-03-04 03:47      MG01
**********FINAL**********
                 DT   cmdb_id
8   2021-03-04 07:54      IG01
9   2021-03-04 07:54  Tomcat01
**********FINAL**********
                 DT   cmdb_id
10  2021-03-04 08:47      MG02
**********FINAL**********
                 DT   cmdb_id
2   2021-03-04 19:22  Tomcat02
4   2021-03-04 19:27      IG02
5   2021-03-04 19:27      IG01
12  2021-03-04 19:51  Tomcat03
```

B系统调用链异常结果

B系统调用链异常检测对于标签数据实现全覆盖

# 挑战2：根因定位

## 1．采用格兰杰因果关系检验判断哪些Metric性能指标为故障根因

### ① 互为因果关系的情况

- system.net.packets_out.count　接口发送的数据包的数目
- system.net.bytes_rcvd　网卡入流量

```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=77.5833 , p=0.0000  , df_denom=1435, df_num=1
ssr based chi2 test:   chi2=77.7455 , p=0.0000  , df=1
likelihood ratio test: chi2=75.7166 , p=0.0000  , df=1
parameter F test:         F=77.5833 , p=0.0000  , df_denom=1435, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=59.3070 , p=0.0000  , df_denom=1432, df_num=2
ssr based chi2 test:   chi2=119.0282, p=0.0000  , df=2
likelihood ratio test: chi2=114.3549, p=0.0000  , df=2
parameter F test:         F=59.3070 , p=0.0000  , df_denom=1432, df_num=2

Granger Causality
number of lags (no zero) 1
ssr based F test:         F=6.6033  , p=0.0103  , df_denom=1435, df_num=1
ssr based chi2 test:   chi2=6.6171  , p=0.0101  , df=1
likelihood ratio test: chi2=6.6019  , p=0.0102  , df=1
parameter F test:         F=6.6033  , p=0.0103  , df_denom=1435, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=5.5778  , p=0.0039  , df_denom=1432, df_num=2
ssr based chi2 test:   chi2=11.1946 , p=0.0037  , df=2
likelihood ratio test: chi2=11.1512 , p=0.0038  , df=2
parameter F test:         F=5.5778  , p=0.0039  , df_denom=1432, df_num=2
```

### ② 相互不为因果关系的情况

- system.tcp.last_ack　LAST_ACK状态套接字个数
- system.net.udp.in_datagrams　接收的UDP数据包

```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=0.0090  , p=0.9243  , df_denom=1435, df_num=1
ssr based chi2 test:   chi2=0.0090  , p=0.9242  , df=1
likelihood ratio test: chi2=0.0090  , p=0.9242  , df=1
parameter F test:         F=0.0090  , p=0.9243  , df_denom=1435, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=3.9427  , p=0.0196  , df_denom=1432, df_num=2
ssr based chi2 test:   chi2=7.9128  , p=0.0191  , df=2
likelihood ratio test: chi2=7.8911  , p=0.0193  , df=2
parameter F test:         F=3.9427  , p=0.0196  , df_denom=1432, df_num=2

Granger Causality
number of lags (no zero) 1
ssr based F test:         F=1.4685  , p=0.2258  , df_denom=1435, df_num=1
ssr based chi2 test:   chi2=1.4715  , p=0.2251  , df=1
likelihood ratio test: chi2=1.4708  , p=0.2252  , df=1
parameter F test:         F=1.4685  , p=0.2258  , df_denom=1435, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=0.8838  , p=0.4134  , df_denom=1432, df_num=2
ssr based chi2 test:   chi2=1.7738  , p=0.4119  , df=2
likelihood ratio test: chi2=1.7727  , p=0.4122  , df=2
parameter F test:         F=0.8838  , p=0.4134  , df_denom=1432, df_num=2
```

# 挑战2：根因定位

③ 在5%的显著水平上，A是引起B的Granger因，
B不是A的Granger因。

- system.tcp.last_ack　　LAST_ACK状态套接字个数
- system.tcp.fin_wait1　　FIN_WAIT1状态套接字个数

## 2．对故障根因中的metric性能指标采用移动差分方法定位根因

根据性能指标的波动判别异常。如果出现故障定位时点没有出现异常波动指标的情况，则降低异常波动的条件，直到出现根因指标为止。

## 3．根据日志的异常来定位根因

```
Granger Causality
number of lags (no zero) 1
ssr based F test:         F=67.2270 , p=0.0000  , df_denom=1436, df_num=1
ssr based chi2 test:   chi2=67.3674 , p=0.0000  , df=1
likelihood ratio test: chi2=65.8381 , p=0.0000  , df=1
parameter F test:         F=67.2270 , p=0.0000  , df_denom=1436, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=55.6898 , p=0.0000  , df_denom=1433, df_num=2
ssr based chi2 test:   chi2=111.7683, p=0.0000  , df=2
likelihood ratio test: chi2=107.6374, p=0.0000  , df=2
parameter F test:         F=55.6898 , p=0.0000  , df_denom=1433, df_num=2

Granger Causality
number of lags (no zero) 1
ssr based F test:         F=0.2385  , p=0.6254  , df_denom=1436, df_num=1
ssr based chi2 test:   chi2=0.2390  , p=0.6249  , df=1
likelihood ratio test: chi2=0.2390  , p=0.6250  , df=1
parameter F test:         F=0.2385  , p=0.6254  , df_denom=1436, df_num=1

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=0.0655  , p=0.9366  , df_denom=1433, df_num=2
ssr based chi2 test:   chi2=0.1314  , p=0.9364  , df=2
likelihood ratio test: chi2=0.1314  , p=0.9364  , df=2
parameter F test:         F=0.0655  , p=0.9366  , df_denom=1433, df_num=2
```

# 挑战3：日志异常检测

AIOps Challenge 智能运维挑战赛 | 科创会 NICON

## 采用DeepLog训练日志

DeepLog模型主要是把log信息当作NLP的自然语言序列来处理的，自动提炼正常运行的log模式信息，当log模式信息偏离训练的模型的时候，可以检测出异常。将检测出来的异常序列构建工作流，为用户提供诊断该异常的语义信息。最后对参数值构建异常检测模型，并将最终异常日志结果进行输出。

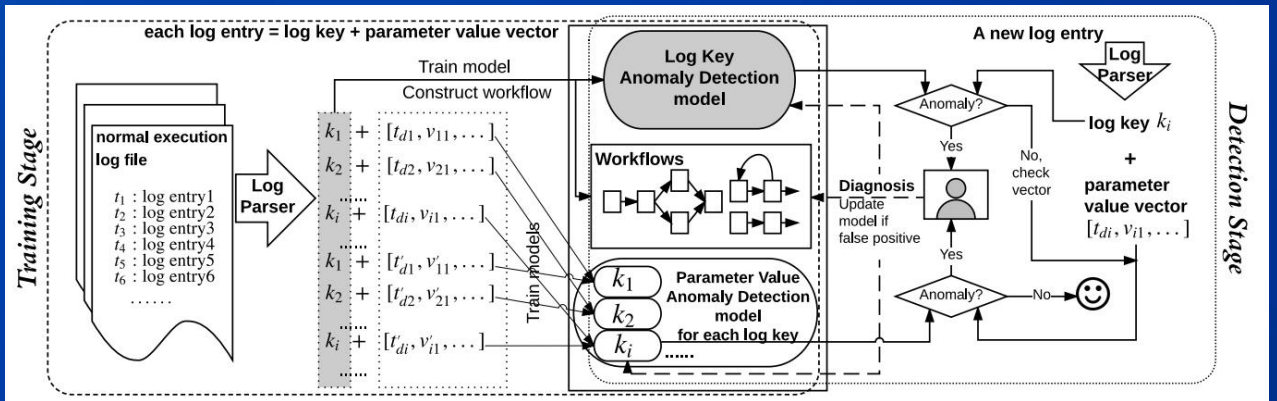



Figure 1: DeepLog architecture.

日志模式化 → 日志序列的异常检测 → 工作流进行根因诊断 → 对参数值做异常检测 → 输出结果

在比赛中，A系统的weblogic日志以及B系统的GC日志模板类型最多，参数变量类型较多，也是对判别日志异常最有价值的两种日志类型。

```
模板8:
*: [CMS-concurrent-reset: */* secs] [Times: user=* sys=*, real=* secs]
示例:
1825320.220: [CMS-concurrent-reset: 0.010/0.010 secs] [Times: user=0.01 sys=0.00, real=0.01 secs]
参数提取:
['0.010', '0.010', '0.01', '0.00', '0.01']
说明:
CMS-concurrent-reset - 这个阶段重新设置CMS算法内部的数据结构，为下一个收集阶段做准备;
0.012/0.012 secs - 展示该阶段持续的时间和时钟时间;
```

通过Drain进行日志模式化提取模板

# 挑战3：日志异常检测

AIOps Challenge 智能运维挑战赛 | NIICON 科创会

| LineId | EventId | EventTemplate | ParameterList |
|---|---|---|---|
| 1 | 1f421d15 | Created <*> resources for pool <*> out of which <*> are available and "0" are unavailable. | ['"1"', '"ds_2102001000"', '"1"'] |
| 2 | 1f421d15 | Created <*> resources for pool <*> out of which <*> are available and "0" are unavailable. | ['"1"', '"ds_2102001000"', '"1"'] |
| 3 | f0a09b42 | Size based data retirement operation completed on archive <*> Retired <*> records in <*> ms. | ['HarvestedDataArchive:', '2,400', '1,472'] |
| 4 | f0a09b42 | Size based data retirement operation completed on archive <*> Retired <*> records in <*> ms. | ['EventsDataArchive:', '0', '1'] |
| 6 | eceb749b | Size based data retirement operation started on archive <*> | ['HarvestedDataArchive:'] |
| 8 | eceb749b | Size based data retirement operation started on archive <*> | ['HarvestedDataArchive:'] |
| 9 | ece46bbe | Scheduled 1 data retirement tasks as per configuration. | [] |
| 10 | f0a09b42 | Size based data retirement operation completed on archive <*> Retired <*> records in <*> ms. | ['HarvestedDataArchive:', '2,400', '1,582'] |
| 11 | eceb749b | Size based data retirement operation started on archive <*> | ['EventsDataArchive:'] |
| 16 | f0a09b42 | Size based data retirement operation completed on archive <*> Retired <*> records in <*> ms. | ['HarvestedDataArchive:', '2,400', '1,381'] |
| 18 | f0a09b42 | Size based data retirement operation completed on archive <*> Retired <*> records in <*> ms. | ['EventsDataArchive:', '0', '1'] |
| 19 | c2da8937 | Self-tuning thread pool contains <*> running threads, <*> idle threads, and <*> standby thread | ['0', '40', '1'] |
| 21 | c2da8937 | Self-tuning thread pool contains <*> running threads, <*> idle threads, and <*> standby thread | ['0', '40', '3'] |

**A系统Weblogic日志Drain模式化与变量提取（处理前）**

| cmdb_id | log | value | parameter | logtype |
|---|---|---|---|---|
| Tomcat04 | gc | 1825318.852:[CMS-concurrent-abortable-preclean:2.085/5.002 secs] [Times:user=2.24 sys=0.07,real=5 | ['2.085', '5.002', '2.24', '0.07', '5.00'] | 6 |
| Tomcat04 | gc | 1825318.854:[GC (CMS Final Remark) [YG occupancy:248579 K (943744 K)]2021-01-30T00.00.04.211+08( | [('248579', '943744'), ('2382816', '3145728'), | 15 |
| Tomcat04 | gc | 1825320.210:[CMS-concurrent-sweep:1.096/1.096 secs] [Times:user=1.13 sys=0.03,real=1.10 secs] | ['1.096', '1.096', '1.13', '0.03', '1.10'] | 7 |
| Tomcat04 | gc | 1825320.220:[CMS-concurrent-reset:0.010/0.010 secs] [Times:user=0.01 sys=0.00,real=0.01 secs] | ['0.010', '0.010', '0.01', '0.00', '0.01'] | 8 |
| Tomcat04 | gc | 1825322.223:[GC (CMS Initial Mark) [1 CMS-initial-mark:2382816K(3145728K)] 2645893K(4089472K),0.0( | [('2382816', '3145728', '2645893', '4089472', | 11 |
| Tomcat04 | gc | 1825323.569:[CMS-concurrent-mark:1.274/1.284 secs] [Times:user=4.96 sys=0.02,real=1.29 secs] | ['1.274', '1.284', '4.96', '0.02', '1.29'] | 9 |
| Tomcat04 | gc | 1825323.585:[CMS-concurrent-preclean:0.016/0.016 secs] [Times:user=0.02 sys=0.01,real=0.01 secs] | ['0.016', '0.016', '0.02', '0.01', '0.01'] | 10 |
| Tomcat04 | gc | 1825328.625:[CMS-concurrent-abortable-preclean:2.133/5.040 secs] [Times:user=2.27 sys=0.08,real=5 | ['2.133', '5.040', '2.27', '0.08', '5.04'] | 6 |
| Tomcat04 | gc | 1825328.628:[GC (CMS Final Remark) [YG occupancy:296992 K (943744 K)]2021-01-30T00.00.13.984+08( | [('296992', '943744'), ('2382816', '3145728'), | 15 |
| Tomcat04 | gc | 1825329.897:[CMS-concurrent-sweep:1.049/1.049 secs] [Times:user=1.09 sys=0.01,real=1.05 secs] | ['1.049', '1.049', '1.09', '0.01', '1.05'] | 7 |
| Tomcat04 | gc | 1825329.907:[CMS-concurrent-reset:0.010/0.010 secs] [Times:user=0.01 sys=0.00,real=0.01 secs] | ['0.010', '0.010', '0.01', '0.00', '0.01'] | 8 |
| Tomcat04 | gc | 1825331.909:[GC (CMS Initial Mark) [1 CMS-initial-mark:2382790K(3145728K)] 2699070K(4089472K),0.0( | [('2382790', '3145728', '2699070', '4089472', | 11 |

**B系统gc日志Drain模式化与变量提取(处理后)**

# 挑战4：数据接收与结果提交

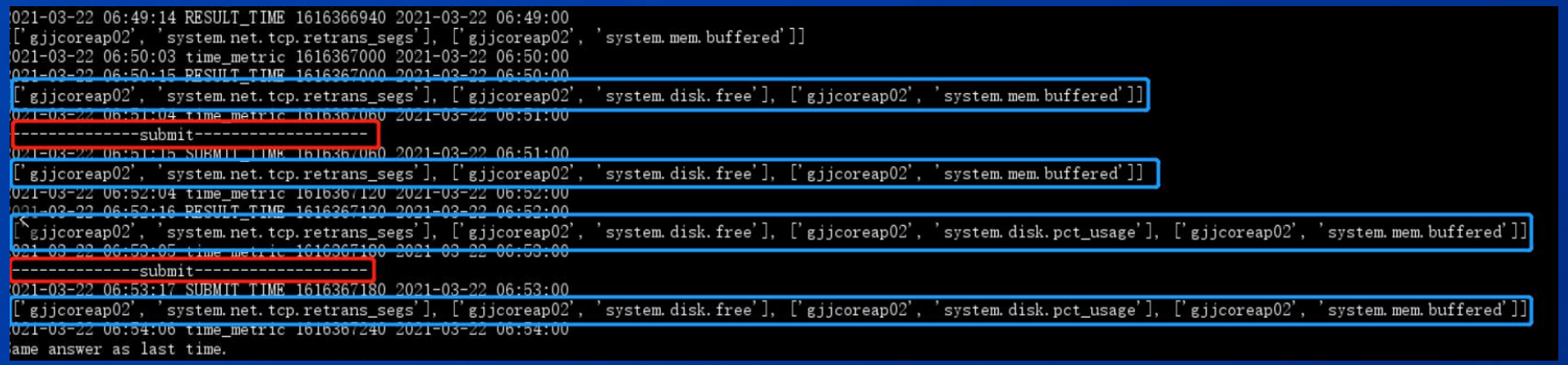
AIOps Challenge 智能运维挑战赛 | 科创会 NICON

## 1．数据接收

① 使用多进程并行计算，提升计算效率。
② 使用结果缓存，防止数据丢失。
③ 根据需要的时间窗口对缓存数据进行实时清理。

## 2．数据提交

添加故障抑制策略：
① 当且仅当前后两次出现相同的根因判断结果时进行上报。
② 当且仅当前后两次出现相同的根因增加时，才进行补充根因上报。
③ 设置最短时间间隔，不同对象之间的故障上报要大于10分钟，同一对象之间的故障上报要大于20分钟。




A系统根因增加重复两次后上报

第二章节

# 算法流程图

# 算法流程图


银行应用系统
接收Metric数据
接收调用链数据
接收日志数据
数据结构化预处理
调用链数据
识别调用链故障
Metric数据
识别Metric故障
识别业务故障
故障抑制与根因提交
业务指标数据
日志数据
日志异常检测

第三章节

# 通用性与展望

# 通用性与展望

## 通用性介绍：

本套算法核心采用移动差分的故障检测定位方法，并增加加权故障定位、N分位值处理，格兰杰因果关系检验等方法，具有较强的通用性，可以同时满足调用链数据、Metric数据以及业务指标的检测需求。对于特征弱的故障以及波动强的噪声，该算法都可以做有效的区分。通过判断调用链的拓扑关系以及N分位值特征，可以避免Trace丢失造成的影响，便于抓取故障时点，从而为根因定位提供强有力的保障。

## 展望：

I. 未来希望通过自动特征工程的方式去产生更多有效的特征

II. 在日志异常检测部分采用DeepLog算法，检测序列异常与根因等。

AIOps Challenge
智能运维挑战赛

2021国际AIOps挑战赛决赛暨AIOps创新高峰论坛

# THANKS

谢 谢 观 看

科创会 NIICON
第一届国际互联网产业科技创新大会暨互联网创新产品展览会
The First International Internet Industry Science And Technology Innovation Conference & Internet Innovation Product Exhibition