华为APM——智能化.

王琛
wangchen53@huawei.com

2018年11月22日

运维现状

DevOps —— Current Status

复杂度 Complexity

应用规模 Scalability

响应速度 Speed

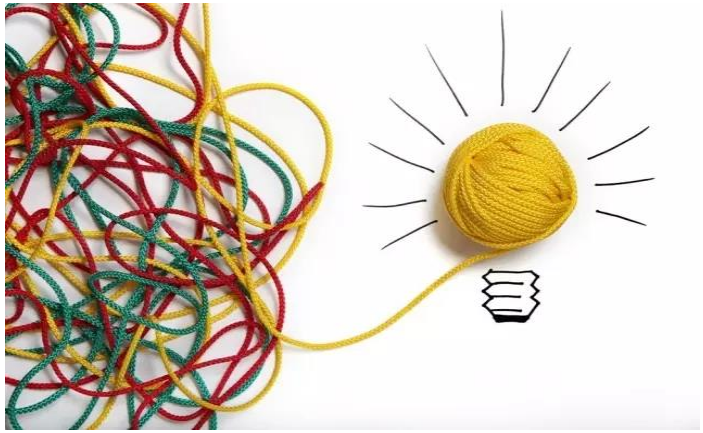
SRE = Site ~~Reliability~~ Engineer

Reboot

APM —— 提高响应速度

APM helps you to efficiently solve application performance problems

> Many companies see up to 60-70% reduction in **MTTR** and in business impact by using APM solutions[1].
> 许多公司在使用了APM后，系统问题的**平均修复时间**减少了60%-70%，从而也降低了对业务的影响[1]。



发现问题 **32%**
PROBLEM IDENTIFICATION

定位问题 **32%**
PROBLEM TROUBLESHOOTING

解决问题 **36%**
PROBLEM RESOLUTION

Vote for most challenging part / 您觉得最有挑战的部分[2] ?

[1] http://www.apmdigest.com/10-business-benefits-of-apm-application-performance-management
[2] Poll by Loom.com

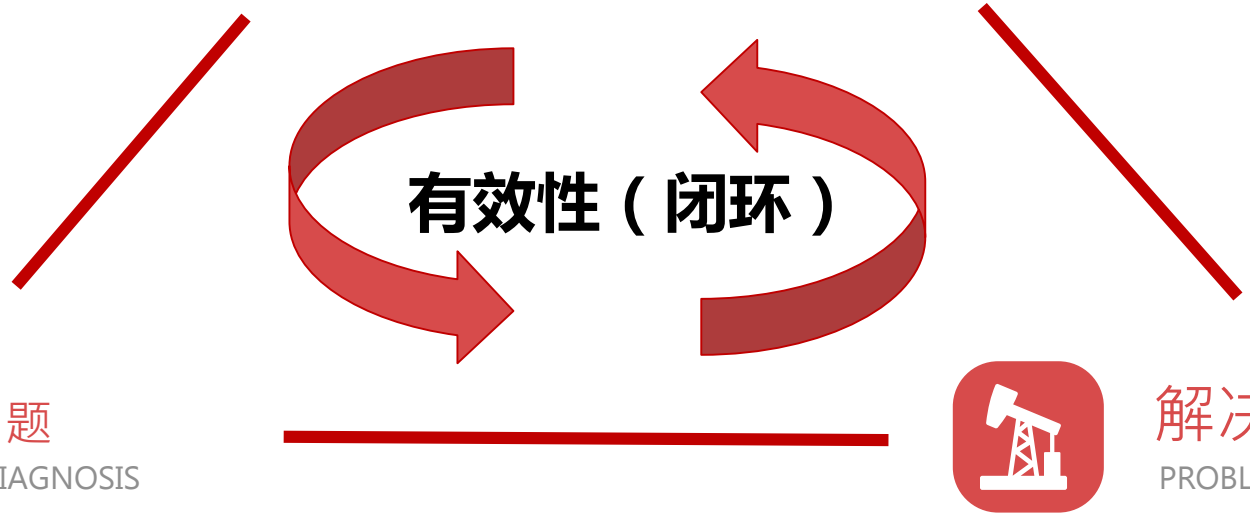
智能化APM解决方案

Artificial Intelligence Enabled APM Solution

发现问题

PROBLEM IDENTIFICATION

- 目标：**及时**发现问题
- 案例：VoltDB 挂了很久没有人发现，直到大规模影响业务才被感知



定位问题

PROBLEM DIAGNOSIS

- 目标：**准确**定位问题
- 案例：VM的内存被撑爆，无法定位是因为哪个应用实例的内存异常导致

解决问题

PROBLEM RESOLUTION

- 目标：**高效**解决问题
- 案例：问题很清晰，但是解决问题，需要一系列的调研工作，外加经历频繁试错，这一过程可能相当冗长

智能化APM解决方案

Artificial Intelligence Enabled APM Solution

智能复合事件处理引擎 Complex Event Process (CEP) Engine

基于动态阈值的方法

原理：利用时间序列过去及现在的值，学习指标随时间变化的模式，利用这个规律来预测未来值（即基线），并计算置信区间（基带）

- 差分整合移动平均自回归模型（ARIMA）
- 长短期记忆网络（LSTM）
- 在线处理（Online Processing）
- 对于有很强规律（周期/趋势）的数据效果比较好
- 对毛刺/突变效果比较好
- 基于变化点告警

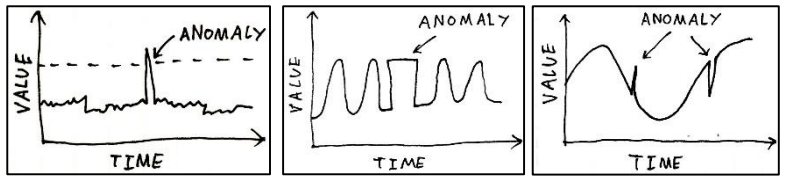
基于特征窗口的方法

原理：通过对时间窗口特征值提取，建立特征向量；基于特征向量，比较窗口间相似度，识别异常窗口

- K-均值算法（K-Means）
- 层次聚类（Hierarchical Clustering）
- 批处理（Batch Processing）
- 对于状态变化效果较好
- 对于严重的异常准确率较高，对小的异常不够敏感
- 基于异常窗口告警

流数据异常检测引擎 Streaming Data Anomaly Detection Engine

指标数据异常检测



挑战：

- 不同场景对异常的定义不统一
- 正常和异常行为界限不清
- 缺失对异常行为的标记数据
- 正常状态模式随实时计算环境动态变化
- 异常状态可能转变为正常状态

异常种类：

- 值异常（Value）
- 趋势异常（Trend Change）
- 周期异常（Seasonality Change）
- 状态变化（Level Shift）

调用链数据异常检测

1. 异常调用链结构发现

正常执行的应用，调用链结构应该呈现一定的规律性，异常调用链结构可能反映异常执行状态（比如错误），可用于异常发现

- 稀有链结构
- 断链

2. 慢链定位

通过指标数据异常功能，可以发现慢的调用链，慢链分析目的在于进一步定界问题，分析出是哪一个span慢导致的调用链慢

- 慢的定义：
  - 绝对值：不可能对每个span的耗时设定静态阈值
  - 相对值：对比历史数据和当前场景，相对较慢
- 对每个span历史执行分析，定位出导致当前慢链的相关性

日志数据异常检测

1. 日志解析

事件模板提取，支持用户通过关键字自定义事件

- 基于聚类（clustering）的算法
- 基于启发式（Heuristic）算法

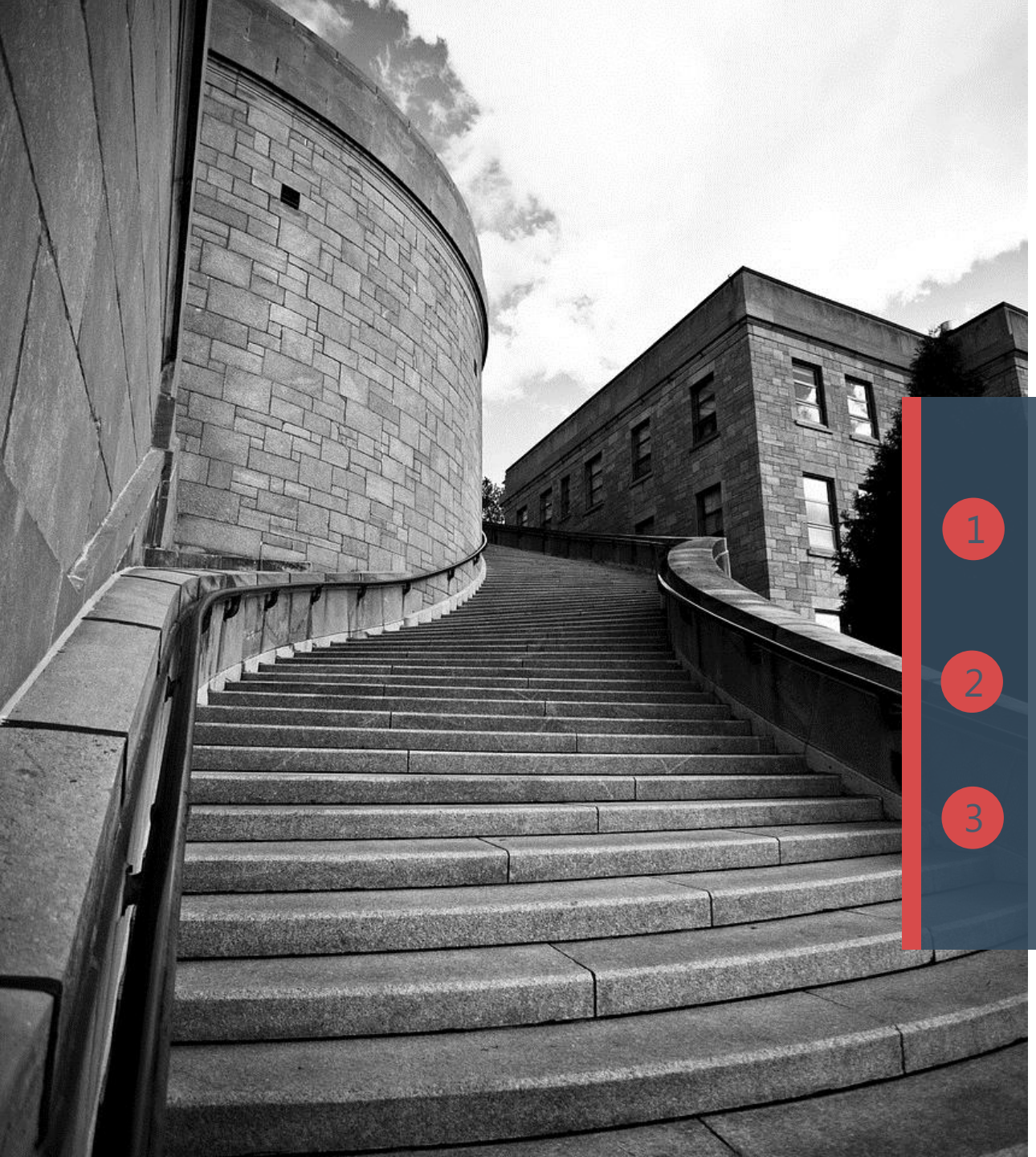
2. 事件时序向量提取

基于窗口的方法，生成事件统计矩阵（日志离散事件转化成连续型类指标数据）

- 固定窗口大小（fixed window）
- 滑动窗口（sliding window）
- 基于Session的窗口

3. 异常检测

复用流数据异常加测引擎，识别稀有事件，突发事件等异常



目录

第一部分
Part I

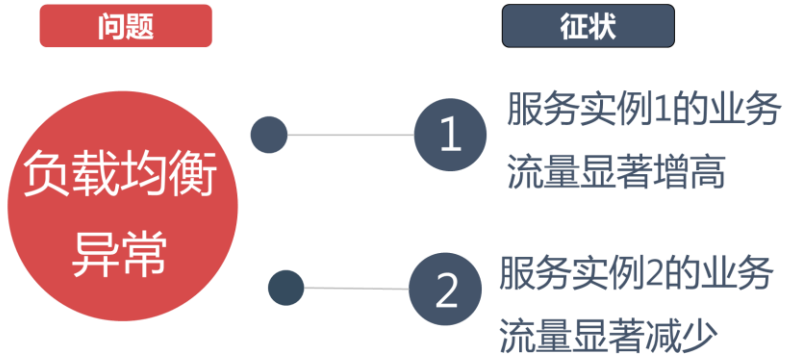
指标.

指标异常检测

Anomaly Detection

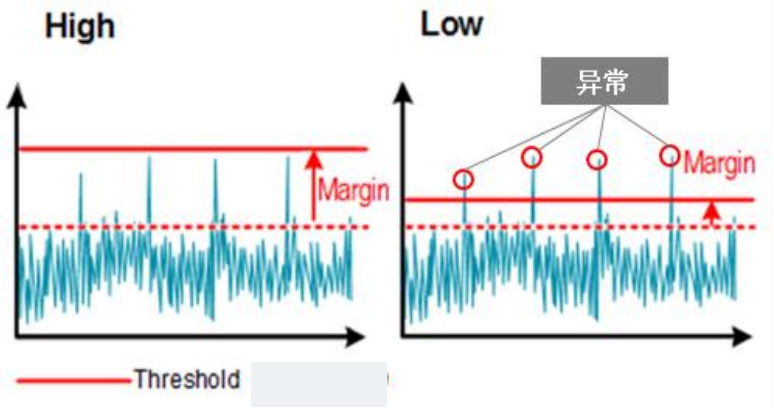
指标监控

- 一个问题，对应多个**征状**
- 多个征状映射潜在的问题（类似医生诊断）
- 因此，指标监控有助于发现异常，定位问题



传统方法：静态阈值

设定阈值和告警规则，例如，CPU使用率连续两个周期大于90% 报警



静态阈值的痛点

对于每个指标要手动设置阈值，麻烦！

云环境下，场景变化快，更新阈值麻烦！

阈值选择过高，敏感度低，错过很多异常！

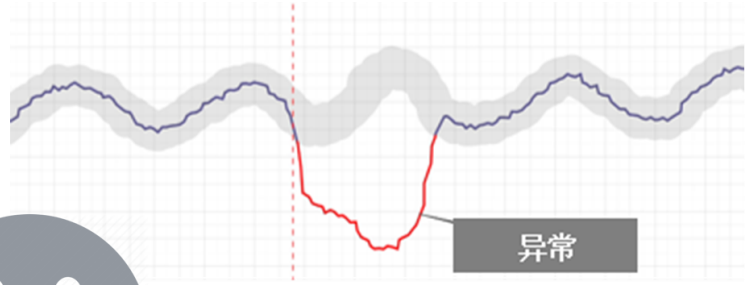
阈值选择过低，过于敏感，过多错误告警！

无法自适应于周期数据，例如闲忙时场景！

智能异常检测

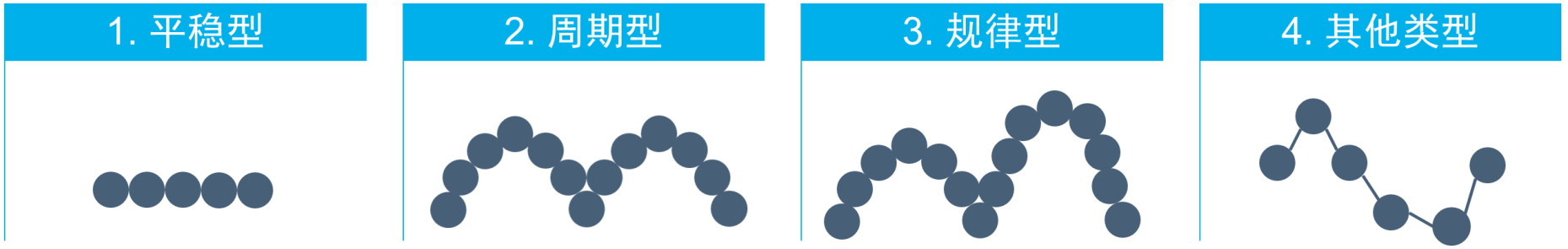
Autonomous Anomaly Detection

根据历史规律，自适应计算基线，判定异常



- 无需手动设置
- 自动调整
- 自适应场景

指标特性分类



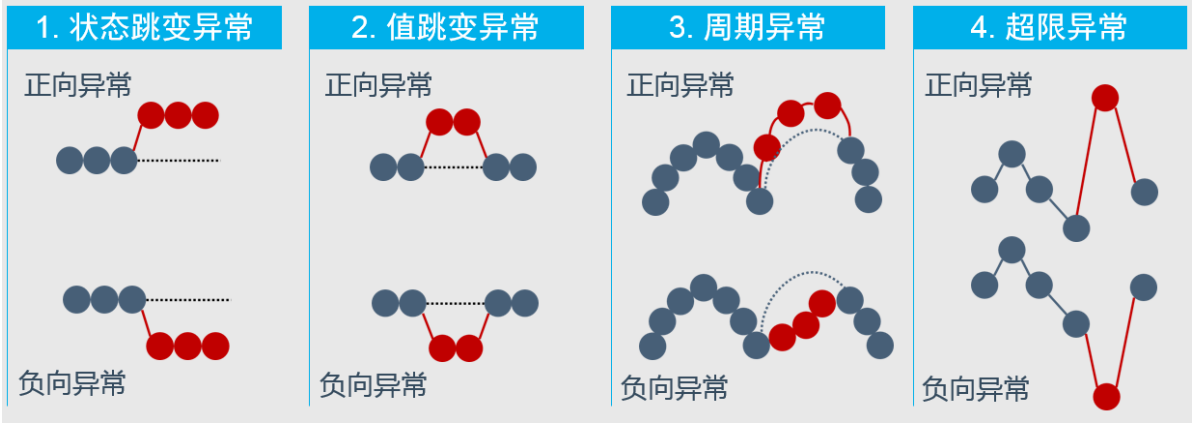
1. 平稳型
指标值在较长的时间窗口呈平稳状态

2. 周期型
默认日周期，即某一时间窗的指标值可以借鉴前几天同一时间窗的指标值

3. 规律型
每天有一个大体相似的规律（变化趋势），但是同样时间窗的值没有相互借鉴意义。

4. 其他类型
指标有较大的波动，规律不明显，不属于上述任意一种类型。

智能异常检测



1. 状态跳变异常
- 平稳型指标状态发生变化，则判定为异常（红圈）。
- 低敏感度可以容忍稍大的偏离，高敏感度则更加敏感，对较小的偏离也会判定为异常。

2. 值跳变异常
- 平稳型指标值突然发生变化，之后回归正常范围，则判定为值异常（红圈）。
- 低敏感度可以容忍稍大的偏离，高敏感度则更加敏感，对较小的偏离也会判定为异常。

3. 周期异常
- 基于历史规律计算出周期基线（蓝色虚线），如果指标值明显偏离基线，则判定为异常。
- 低敏感度可以容忍稍大的偏离，高敏感度则更加敏感，对较小的偏离也会判定为异常。

4. 超限异常
- 对于波动较大的指标，当指标值明显超出历史值范围，判定为异常（红圈）。
- 低敏感度可以容忍稍大的偏离，高敏感度则更加敏感，对较小的偏离也会判定为异常。

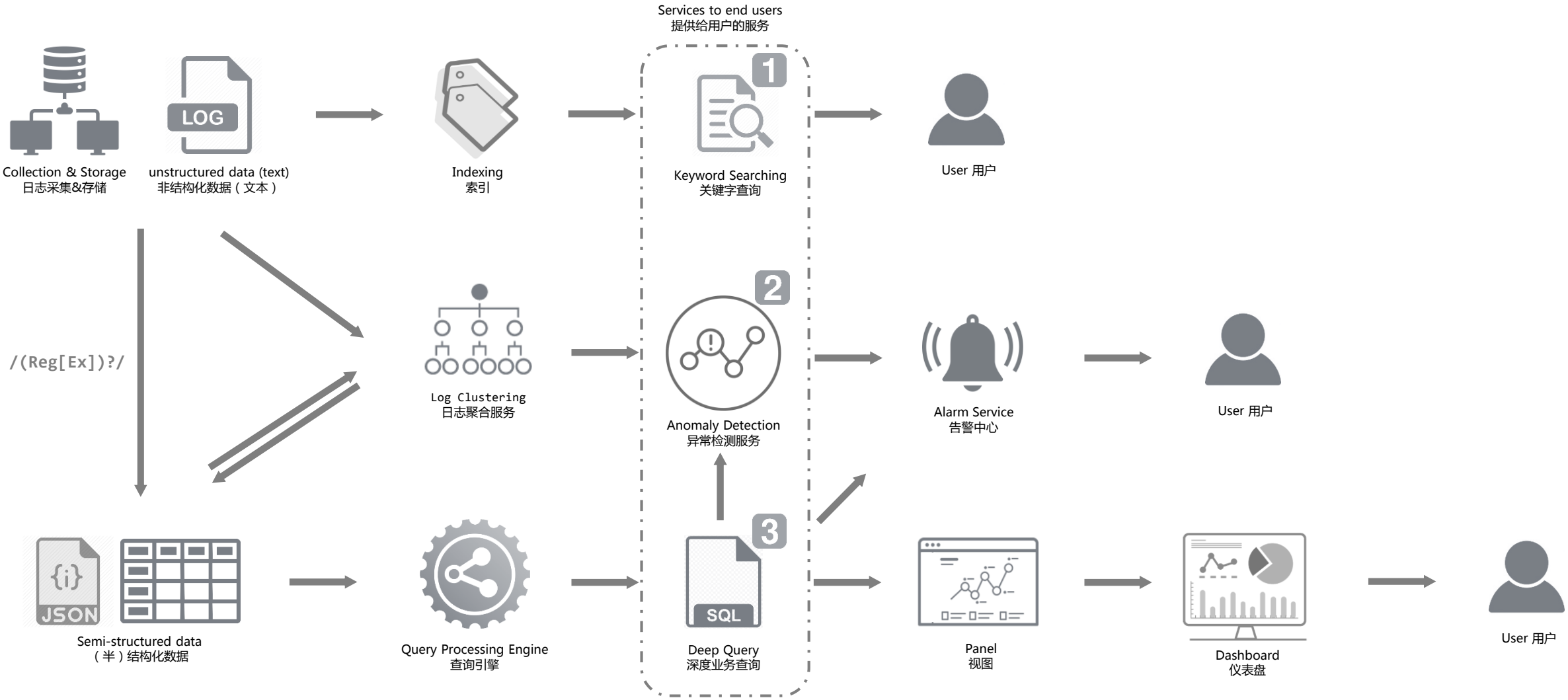
5. 状态变化异常
- 系统可以识别周期，规律，平稳，和无规律四种状态。每次在指标状态发生改变的时候，发送告警。

第二部分
Part II

日志.

日志管理服务

The Log Management Service —— Architecture



日志结构化

Structure Log Data

#	采集时间	日志内容
1	2018-06-07 10:33:21.268 +0800	161.71.8.142 - - [2018-06-07 02:33:21.268 +0000] "GET /aboutus/ HTTP/1.1" 200 2732
2	2018-06-07 10:33:21.261 +0800	70.69.152.165 - - [2018-06-07 02:33:21.261 +0000] "GET /_media/company_logo.png HTTP/1.1" 200 8336
3	2018-06-07 10:33:21.254 +0800	169.107.162.237 - - [2018-06-07 02:33:21.254 +0000] "GET /_js/master.js HTTP/1.1" 200 1419
4	2018-06-07 10:33:21.246 +0800	17.233.159.60 - - [2018-06-07 02:33:21.246 +0000] "GET /aboutus/ HTTP/1.1" 200 6009
5	2018-06-07 10:33:21.238 +0800	30.75.225.192 - - [2018-06-07 02:33:21.238 +0000] "GET /_media/customer_tab_selected_top.png HTTP/1.1" 200 2747
6	2018-06-07 10:33:21.227 +0800	19.174.45.8 - - [2018-06-07 02:33:21.227 +0000] "GET /_media/company_logo.png HTTP/1.1" 404 8421

需要用户参与，用户通过正则表达式来提取想要的字段（信息），并给予响应的名称（列名）。

#	采集时间	IP	标准时间	方法	资源URL	状态码	大小	原始日志
1	2018-06-07 10:33:21.268 +0800	161.71.8.142	2018-06-07 02:33:21.268 +0000	GET	/aboutus/ HTTP/1.1“	200	2732	****
2	2018-06-07 10:33:21.261 +0800	70.69.152.165	2018-06-07 02:33:21.261 +0000	GET	/_media/company_logo.png HTTP/1.1"	200	8336	****
3	2018-06-07 10:33:21.254 +0800	169.107.162.237	2018-06-07 02:33:21.254 +0000	GET	/_js/master.js HTTP/1.1"	200	1419	****
4	2018-06-07 10:33:21.246 +0800	17.233.159.60	2018-06-07 02:33:21.246 +0000	GET	/aboutus/ HTTP/1.1"	200	6009	****
5	2018-06-07 10:33:21.238 +0800	30.75.225.192	2018-06-07 02:33:21.238 +0000	GET	/_media/customer_tab_selected_top.png HTTP/1.1"	200	2747	****
6	2018-06-07 10:33:21.227 +0800	19.174.45.8	2018-06-07 02:33:21.227 +0000	GET	/_media/company_logo.png HTTP/1.1"	404	8421	****
7	……							

之后，用户可以使用查询语言（类SQL）对相应的字段进行查询。

```
select * from /apache/access
where status_code > 200
```

#	采集时间	IP	标准时间	方法	资源URL	状态码	大小	原始日志
1	2018-06-07 10:33:21.227 +0800	19.174.45.8	2018-06-07 02:33:21.227 +0000	GET	/_media/company_logo.png HTTP/1.1"	404	8421	****
2	2018-06-07 10:33:24.227 +0800	30.75.225.192	2018-06-07 02:33:21.227 +0000	GET	/_media/company_logo.png HTTP/1.1"	500	3621	****
3	……							

联合查询

```
select * from /apache/access
where status_code > 200 and ip=19.174.45.8
```

#	采集时间	IP	标准时间	方法	资源URL	状态码	大小	原始日志
1	2018-06-07 10:33:21.227 +0800	19.174.45.8	2018-06-07 02:33:21.227 +0000	GET	/_media/company_logo.png HTTP/1.1"	404	8421	****
2	……							

*样例日志 /apache/access.log

异质化日志

Heterogeneous Logs

#	采集时间	日志内容
1	2018-06-07 10:33:21.268 +0800	161.71.8.142 - - [2018-06-07 02:33:21.268 +0000] "GET /aboutus/ HTTP/1.1" 200 2732
2	2018-06-07 10:33:21.261 +0800	70.69.152.165 - - [2018-06-07 02:33:21.261 +0000] "GET /_media/company_logo.png HTTP/1.1" 200 8336
3	……	

日志本身必须是结构化的，我们称为同质化。
也就是说，全量日志按一个格式打印，例如 apache access.log

#	采集时间	IP	标准时间	方法	资源URL	状态码	大小	原始日志
1	2018-06-07 10:33:21.268 +0800	161.71.8.142	2018-06-07 02:33:21.268 +0000	GET	/aboutus/ HTTP/1.1“	200	2732	****
2	2018-06-07 10:33:21.261 +0800	70.69.152.165	2018-06-07 02:33:21.261 +0000	GET	/_media/company_logo.png HTTP/1.1"	200	8336	****
3	……							

反面示例：非结构化日志（异质化）

#	采集时间	日志内容
1	2018-06-07 10:33:21.268 +0800	[2018-06-07 02:33:21.268 +0000] [OrderMgmt.java-157] The Order #123456 has been updated.
2	2018-06-07 10:33:21.261 +0800	[2018-06-07 02:33:21.261 +0000] [UserMgmt.java-126] User 7854210 updated password.
3	2018-06-07 10:33:21.254 +0800	[2018-06-07 02:33:21.254 +0000] [InventoryMgmt.java-48] The quantity of product #abcde123 has been updated to 211.
4	2018-06-07 10:33:21.246 +0800	[2018-06-07 02:33:21.246 +0000] [UserMgmt.java-311] User 2589610 updated address.
5	2018-06-07 10:33:21.238 +0800	[2018-06-07 02:33:21.238 +0000] [OrderMgmt.java-532] The Order !543210 has been canceled.
6	2018-06-07 10:33:21.227 +0800	[2018-06-07 02:33:21.227 +0000] [IdMgmt.java-212] User 5423681 Login failed, wrong password.

无法通过查询语言查询关键信息
例如：查询登录失败超过3次的用户

#	采集时间	时间	类名	行号	日志内容
1	2018-06-07 10:33:21.268 +0800	2018-06-07 02:33:21.268 +0000	OrderMgmt.java	157	The Order #123456 has been updated.
2	2018-06-07 10:33:21.261 +0800	2018-06-07 02:33:21.261 +0000	UserMgmt.java	126	User 7854210 updated password.
3	2018-06-07 10:33:21.254 +0800	2018-06-07 02:33:21.254 +0000	Inventory.java	48	The quantity of product #abcde123 has been updated to 211.
4	2018-06-07 10:33:21.246 +0800	2018-06-07 02:33:21.246 +0000	UserMgmt.java	311	User 2589610 updated address.
5	2018-06-07 10:33:21.238 +0800	2018-06-07 02:33:21.238 +0000	OrderMgmt.java	532	The Order !543210 has been canceled.
6	2018-06-07 10:33:21.227 +0800	2018-06-07 02:33:21.227 +0000	IdMgmt.java	212	User 5423681 Login failed, wrong password.

只能对打日志的插件打印的信息进行结构化，无法对日志能容进行结构化，从而没有达到结构化的目的
多条regex， 等同于rule-based system（缺点）

智能日志分析

Smart Log Analysis

1. 日志收集/管理

Log Collection / Storage

1 **2008-11-09 20:55:54** PacketResponder 0 for block blk_321 terminating
2 **2008-11-09 20:55:54** Received block blk_321 of size 67108864 from /10.251.195.70
3 **2008-11-09 20:55:54** PacketResponder 2 for block blk_321 terminating
4 **2008-11-09 20:55:54** Received block blk_321 of size 67108864 from /10.251.126.5
5 **2008-11-09 21:56:50** 10.251.126.5:50010:Got exception while serving blk_321 to /10.251.127.243:
6 **2008-11-10 03:58:04** Verification succeeded for blk_321
7 **2008-11-10 10:36:37** Deleting block blk_321 file /mnt/hadoop/dfs/data/current/subdir1/blk_321
8 **2008-11-10 10:36:50** Deleting block blk_321 file /mnt/hadoop/dfs/data/current/subdir51/blk_321

- 采集/存储
- 日志查询/筛选（关键字，日期等）
- 日志可视化工具，例如日志dashboard（统计信息等）

2. 日志解析

Log Parsing

Event Templates:

Event 1: PacketResponder * for block * terminating
Event 2: Received block * of size * from *
Event 3: *:Got exception while serving * to *
Event 4: Verification succeeded for *
Event 5: Deleting block * file *

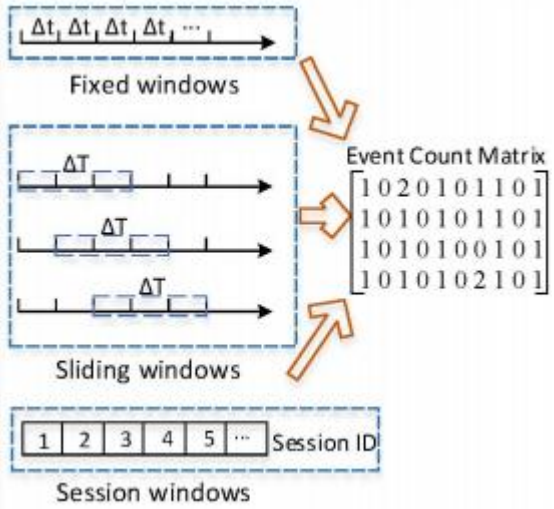
Log Events:

Log 1→ Event 1　Log 2→ Event 2
Log 3→ Event 1　Log 4→ Event 2
Log 5→ Event 3　Log 6→ Event 4
Log 7→ Event 5　Log 8→ Event 5

- 区分变量部分（variable）和常量部分（constant）
- 基于聚类（clustering）的算法
- 基于Heuristic 算法

3. 特征提取

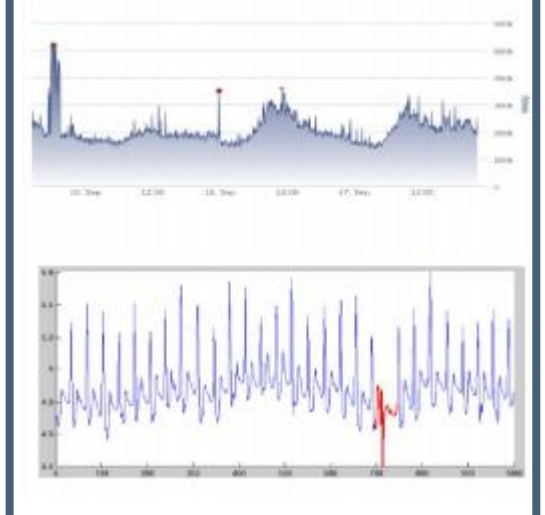
Feature Extraction



- 基于窗口的方法，生成事件统计矩阵
- 固定窗口大小（fixed window）
- 滑动窗口（sliding window）
- 基于Session的窗口

4. 异常检测

Anomaly Detection



- 稀有事件，突发事件检测
- 基于特征值的方法
- 基于时序数据处理

文本聚类

Text Clustering

原始日志

```
#1  The Order #123456 has been updated.
#2  User 7854210 updated password.
#3  The quantity of product #abcde123 has been updated to 211.
#4  User 2589610 updated address.
#5  The Order #543210 has been canceled.
#6  User 5423681 Login failed, wrong password.
#7  The Order #654321 has been updated.
#8  The quantity of product #heabc123 has been updated to 532.
#9  The Order #5462145 has been updated.
#10 User 5423681 Login failed, wrong password.
#11 The Order #123546 has been canceled.
#12 User 5423681 updated password.
#13 The quantity of product #1236541201 has been updated to 10.
#14 User 2589610 Login failed, wrong password.
#15 User 7854210 updated address.
```

日志聚类

```
The Order #123456 has been updated.
The Order #654321 has been updated.
The Order #5462145 has been updated.

User 5423681 updated password.
User 7854210 updated password.

The quantity of product #abcde123 has been updated to 211.
The quantity of product #heabc123 has been updated to 532.
The quantity of product #1236541201 has been updated to 10.

User 2589610 updated address.
User 7854210 updated address.

The Order #543210 has been canceled.
The Order #123546 has been canceled.

User 5423681 Login failed, wrong password.
User 5423681 Login failed, wrong password.
User 2589610 Login failed, wrong password.
```

日志模板提取

```
The Order [var1] has been updated.

User [var1] updated password.

The quantity of product [var1] has been updated to [var2].

User [var1] updated address.

The Order [var1] has been canceled.

User [var1] Login failed, wrong password.
```

生成日志模板

#	统计次数	日志模板
1.	251	The Order [var1] has been updated.
2.	220	User [var1] updated password.
3.	185	The quantity of product [var1] has been updated to [var2].
4.	132	User [var1] updated address.
5.	82	The Order [var1] has been canceled.
6.	10	User [var1] Login failed, wrong password.

第三部分
Part III

调用链.

事务根因分析

Root Cause Analysis

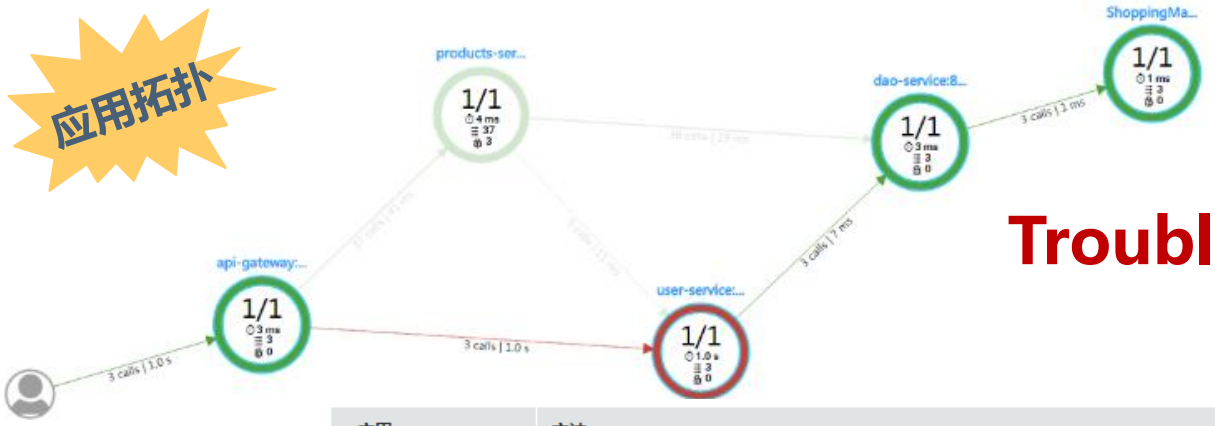
Where！ 可以告诉用户，**哪里**出问题了

- 通过埋点，获取调用链信息，做到函数级别问题定界

Why？ 不能告诉用户，**为什么**这里有问题

- 无法为用户查找根因，解决问题提供**有效信息**
- 调用链分析大多**基于人力**，缺少自动化，甚至智能化分析功能

应用拓扑



Troubleshooting=Guesswork！

调用链

应用	方法	参数	状态	时间线	
api-gateway	org.apache.catalina.core.standardhostvalve.invoke	/product/product/bu...	失败		0ms
products-service	org.apache.catalina.core.standardhostvalve.invoke	/product/buy/342112...	失败		0ms
products-service	org.springframework.web.servlet.frameworkservlet.dopost		成功		0ms
dao-service	org.apache.catalina.core.standardhostvalve.invoke	/persistence/payme...	失败		0ms
dao-service	org.springframework.web.servlet.frameworkservlet.doget		成功		0ms
dao-service	org.springframework.web.servlet.frameworkservlet.doget		失败		0ms
dao-service	com.huawei.cloud.api.persistancerestcontroller.addcart		失败		0ms
dao-service	com.huawei.cloud.mysql.dbmanager.addpayment		失败		0ms
dao-service	com.mysql.jdbc.connectionimpl.preparestatement		成功		0ms
dao-service	com.mysql.jdbc.nonregisteringdriver.connect	ShoppingMallDB	成功		4ms

问题定界

事务洞察——根因分析

Why We Need Insights?

目标

- 回答“**为什么**”的问题；
- 通过智能化分析，给用户一些**提示**（我们称为“洞察” Insights），帮助用户减少定位问题（troubleshooting）中的Guess Work，从而更高效的解决问题。

方法

- 输入：应用执行过程中从**调用链**抓取的实时数据：
  - 函数的入参出参（例如，REST 请求中URL里自带的参数）
  - 实时环境信息（例如，cpu_queue_length）
  - 其他参数（例如，操作系统版本，浏览器类型等）
- 输出：试图推导出一些问题出现时普遍存在的规律，帮助用户分析问题的根因；
  - 功能性问题。例如：当手机APP 版本为1.2.1时登录的错误率比较高；
  - 非功能性问题。例如，当浏览器为IE 时，查询加载结果的等待时间较长；

事务洞察原理

How Insight is Working?

异常请求 Bad Transaction

- 返回值为**错误**的事务，例如，返回码404（http 请求），返回值为1（内部调用）
- **慢**的调用，例如，超出某静态阈值（例如1秒）的事务请求，或者是最慢的5%请求

正常请求 Good Transaction

- 正常的调用，即，不是bad transaction 的事务调用

关键步骤：

1. 从调用链获取参数
   Attribute : value
2. 对每个调用链贴标签
   Good or Bad
3. 生成Insight
   分析每个参数对好坏的影响

调用链获取参数，以key-value对格式存储，可以是调用参数，也可以是环境变量。例如下图所示，可以记录用户浏览器类型，操作系统类型等。

根据调用链结果，对调用链上所有采集到的数据贴标签：正常请求为Good，异常请求为Bad。

例如browser= firefox 在正常事务中出现10次，在异常事务中出现132次，那么则说明用户使用火狐浏览器很有可能和导致失败（或者慢）相关。

Good

Key-Value	Count
browser= firefox	10
browser= chrome	200
OS = windows	253
OS = Mac OS X	18

Bad

Key-Value	Count
browser= firefox	132
browser= chrome	25
OS = windows	128
OS = Mac OS X	32

Thank You.

2018-11-22