



# 第十一届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2020

## 架构革新 高效可控

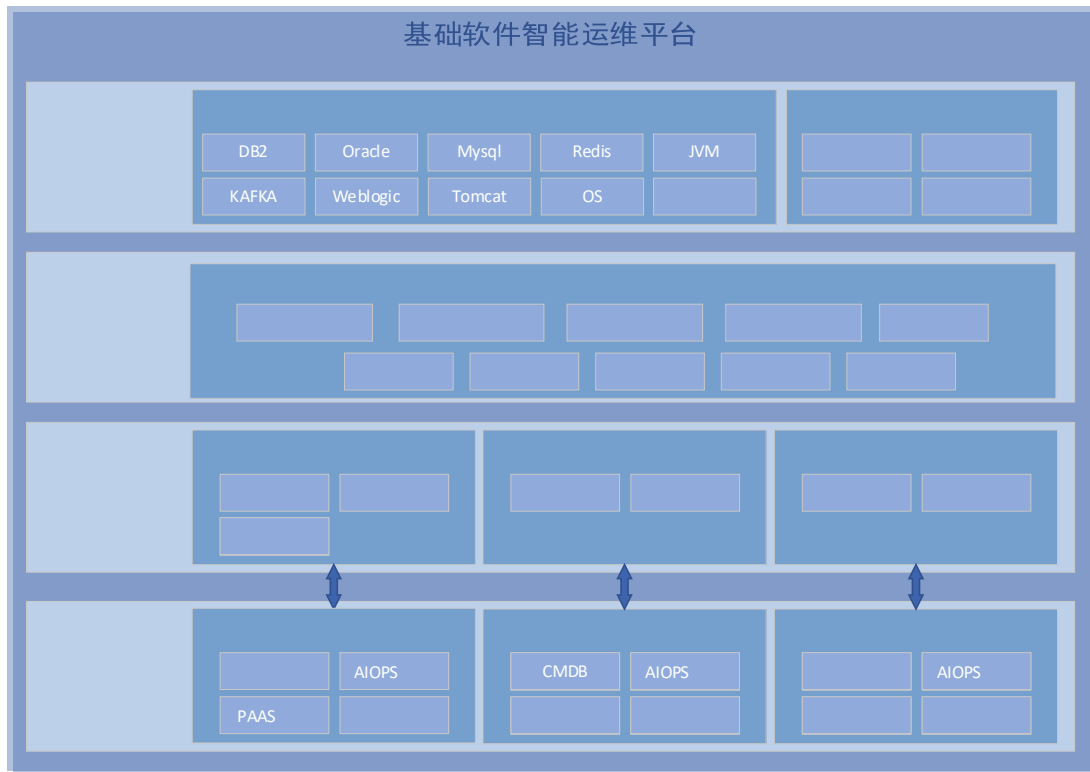


北京国际会议中心 | 2020/12/21-12/23

# 民生银行数据库智能运维实践

## 基础智能运维平台





## 智能运维服务

实现软硬件各类产品智能深度运维。同时对外提供自助服务，通过自定义指标，关系和服务的方式，用户可以实现个性化智能运维。

## 智能算法库

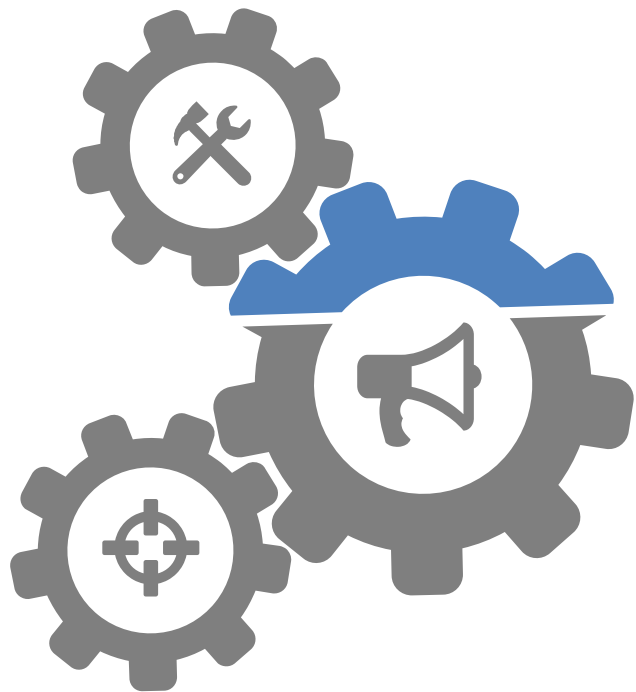
集成Sklearn、statsmodels、tensorflow、pytorch、prophet等AI算法库，利用pandas、numpy、sycipy、matplotlib等数据分析工具，自研智能运维相关算法并落地。

## 实时计算引擎

采用分布式计算框架，缓存数据库等实现实时数据处理，运用人工智能技术实现运维数据价值挖掘。

## 运维数据中台

运用ES、Hadoop、Flink、KAFKA、Redis、关系数据库，时序数据库等技术，整合各种监控数据，日志文档数据，配置数据，关系数据，知识库等数据来源，结合AIOPS分析资产，建立运维数据中台，为运维决策提供数据支撑。



### 产品深度智能运维

异常检测，根因分析，智能场景，故障预测



### 监控告警优化

使用动态阈值，提高告警准确性和相关意义



### 日志异常检测

通过比对日志和知识库来报告问题和  
分析解决问题



### 智能告警分析

实时分析相关告警，定位核心问题

# 民生银行数据库智能运维实践

产品深度运维



1

## 应急“双十”底线

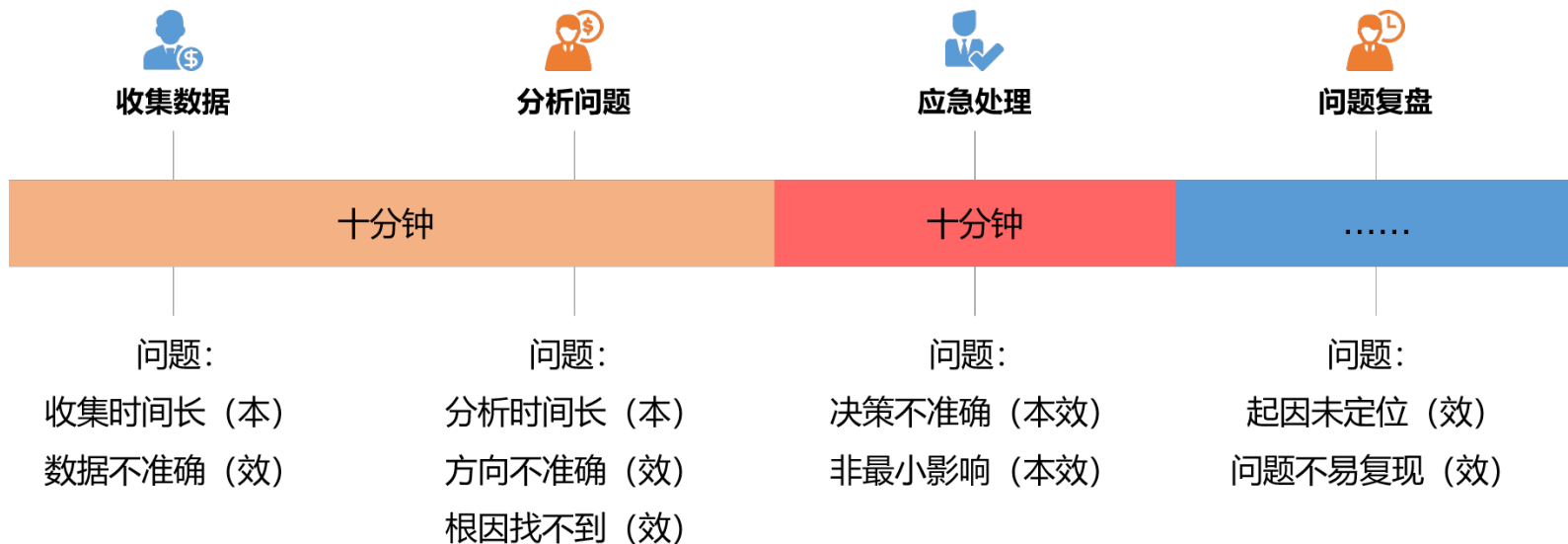
最长十分钟分析时间  
最长十分钟救急时间

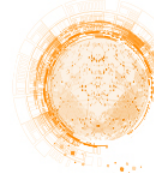


2

## 运维数据“价值”

海量运维数据价值如何挖掘





## 异常检测

- 支撑海量运维指标.
- 需要采用无监督学习方法, 减少人工标注成本



## 根因分析

- 一步定位到问题sql.
- 问题SQL执行详情分析, 定位问题原因.



## 智能场景

- 指标告警收敛.
- 便于普通用户使用, 提供告警场景解释和对应的解决方案.





# 民生银行数据库智能运维实践

数据库深度运维应用效果



架构革新 11<sup>th</sup>  
自主可控

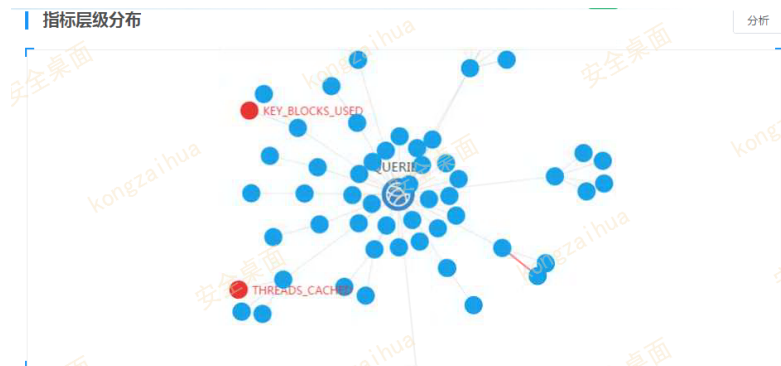




## 异常数量排行 (2019-12-04 14:50:00)

系统名称	数据库名称	异常量	命中场景
EMU系统	EMUEDU	24	日志写盘/门等待/数据传输/...
现金管理系统	CMPDB	15	QPS异常/扫描数据量
新网银系统	DBYDB_02	5	回滚异常
支付通道整合平台	PCIPDB2	4	无
计费系统	LAISDB	4	无
新网银系统	DBYDB_03	3	无
新网银系统	DBYDB_01	2	锁异常
分布式核心业务系统	VCDB	2	锁异常
新网银系统	DBYDB_00	1	无

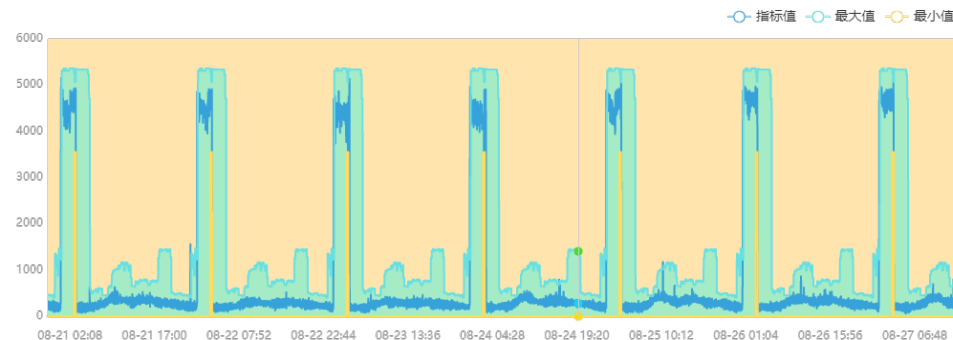
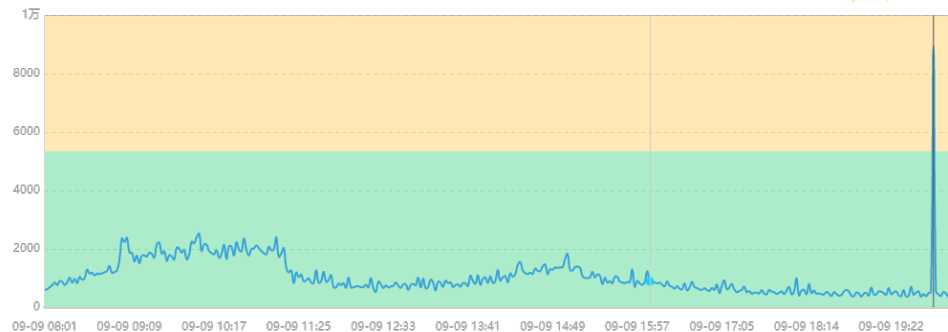
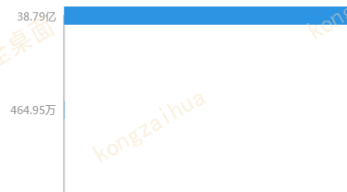


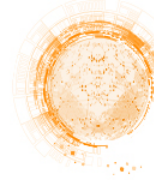


### 指标SQL排行 (SUM\_TIMER\_WAIT)

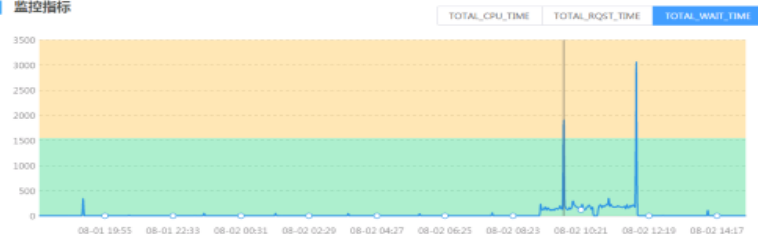


### 事件排行 (SUM\_TIMER\_WAIT)

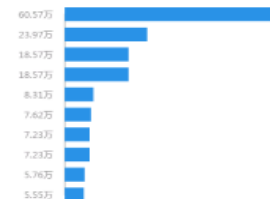




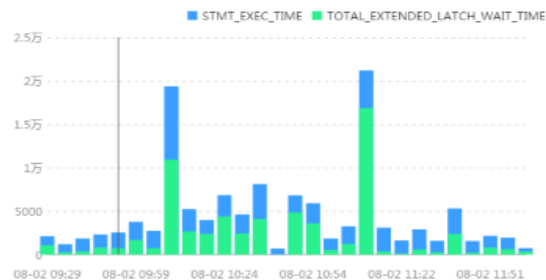
监控指标



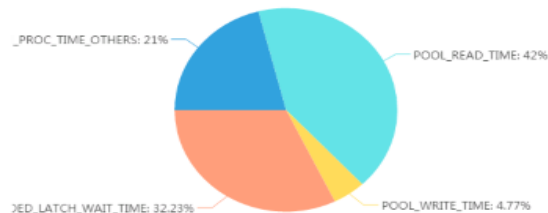
指标SQL排行 (TOTAL\_ACT\_WAIT\_TIME)

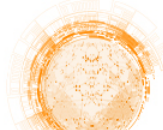


SQL执行时间



SQL时间分布图





## 场景分析

### 场景1:日志写盘异常 ▾

解析

当前异常表示数据库写活动日志到磁盘出现异常。

异常分析:

1. 一种原因是IO延时高,需要警惕。建议结合操作系统,存储信息来一起判断。

2. 另一种原因是出现临时大量INSERT|UPDATE|DELETE操作,写日志成为瓶颈。可以参考当前时间的ROWS\_MODIFIED, ROWS\_DELETED, ROWS\_INSERTED, ROWS\_UPDATED判断是否发生突变。

解决方案:

1. 找到写数据的原因,分析是否是正常业务。

2. 如果伴随着LOG\_BUFFER\_WAIT\_TIME, NUM\_LOG\_BUFFER\_FULL异常,可以考虑增加数据库参数Logbufsz的设置。

3. 排除是否是磁盘IO缓慢原因,如有问题请联系系统组分析解决。

异常DB指标

LOG\_DISK\_WAIT\_TIME TOTAL\_COMMIT\_TIME

关联SQL1 ▾

```
create index I_T_ORDER1_20191009
on t_order_20191009 ( PLAT_DATE,
SEQ_NO ) ALLOW REVERSE SCANS
```

SQL指标: LOG\_DISK\_WAIT\_TIME

关联SQL2 ▾

### 场景2:锁异常 ▾

解析

当前异常表示数据库锁等待出现异常。

异常分析:

1. 一般是业务锁冲突,可能有特殊情况发生,如特殊的跑批语句。

2. 如果是锁升级,可以看看是不是locklist设置不够。。

解决方案:

1. 分析锁等待事件。可以结合ipeas平台查看当时锁等待事件。找到的sql,分析是否是业务问题。

2. 如果是锁升级,并且是LOCK\_ESCALS\_LOCKLIST异常,可以考虑增大数据库参数locklist设置。但是前提是先判断业务sql是否合理。

异常DB指标

LOCK\_ESCALS LOCK\_ESCALS\_LOCKLIST

关联SQL1 ▾

```
insert
into
t_order_20191008
(select
a.*
from
t_order a,
tmp_order_mv b
where
a.order_id=b.order_id)
```

SQL指标: LOCK\_ESCALS LOCK\_ESCALS\_LOCKLIST

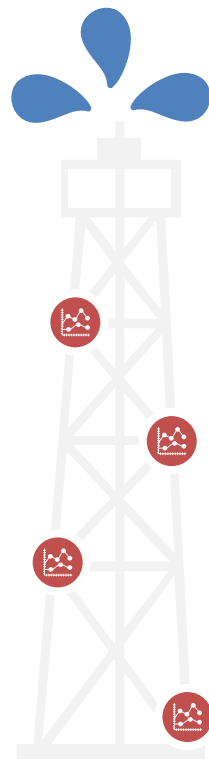
# 民生银行数据库智能运维实践

数据库深度运维实现方法



架构革新 11<sup>th</sup>  
自主可控





## 对象

选择数据库全局指标作为检测对象  
数据库的事务，SQL等作为根因定位对象

## 性能

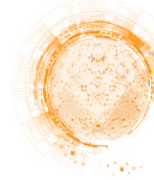
训练性能如何保证  
实时处理性能如何保证

## 算法

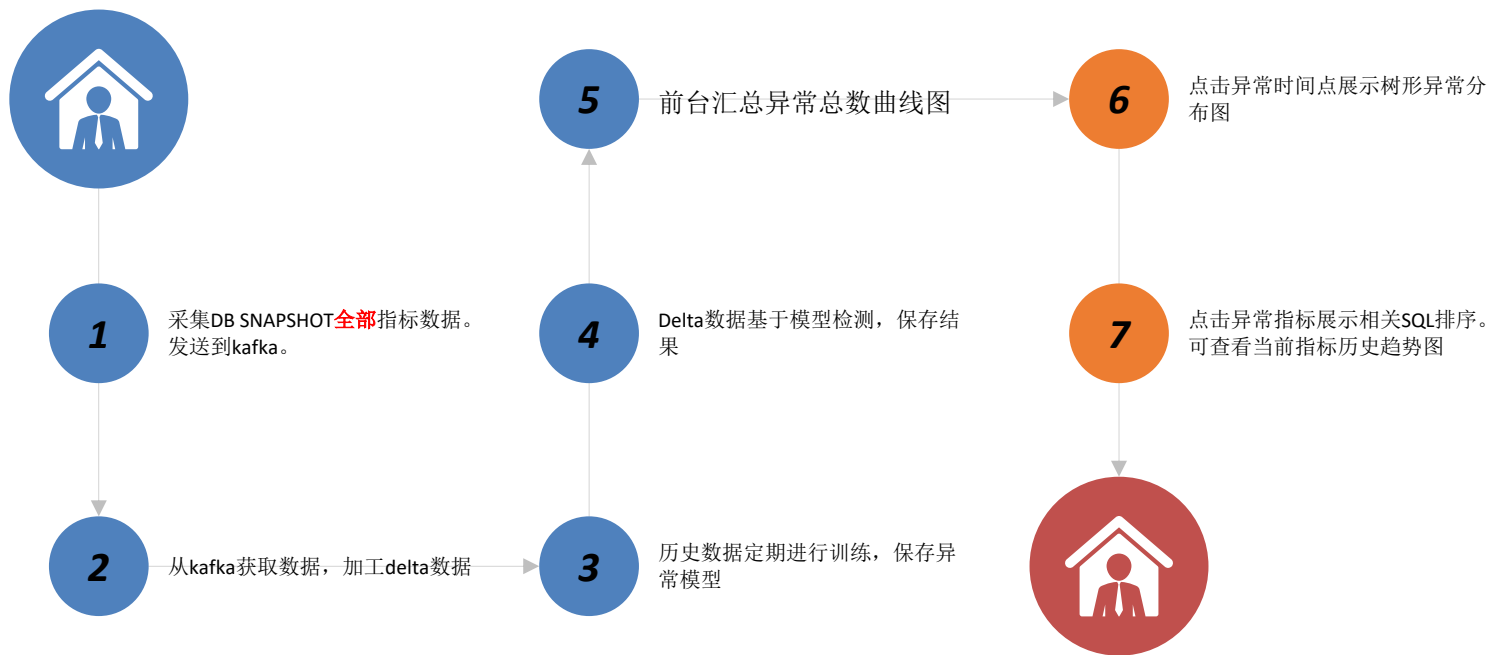
采用无监督学习算法，多种算法合成集成算法  
可选时间序列算法，提升周期性检测能力

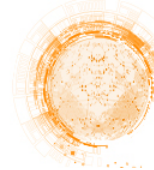
## 展示

指标聚合层次化展示，基于智能场景告警









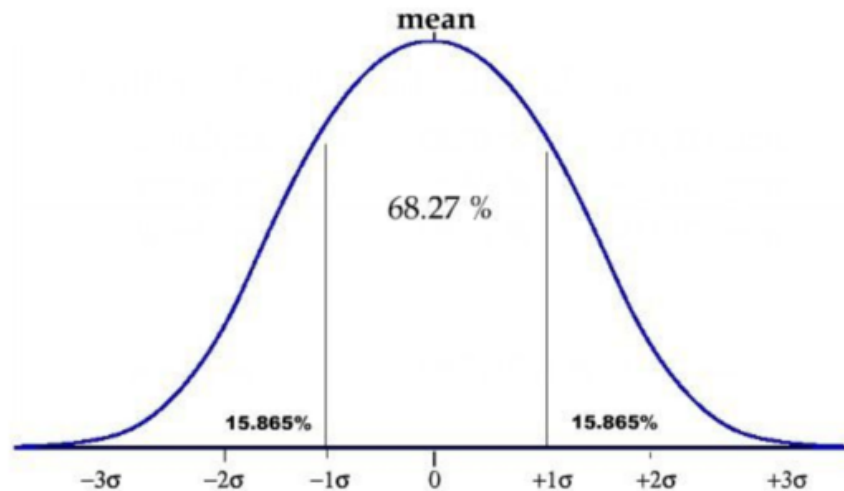
## Standard deviation

均值: 
$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

标准差: 
$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

在正态分布中 $\sigma$ 代表标准差, $\mu$ 代表均值  
 $x=\mu$ 即为图像的对称轴  
三 $\sigma$ 原则即为

数值分布在  $(\mu-\sigma, \mu+\sigma)$  中的概率为0.6526  
数值分布在  $(\mu-2\sigma, \mu+2\sigma)$  中的概率为0.9544  
数值分布在  $(\mu-3\sigma, \mu+3\sigma)$  中的概率为0.9974



利用一种名为孤立树*iTree*的二叉搜索树结构来孤立样本。

由于异常值的数量较少且与大部分样本的疏离性，因此，异常值会被更早的孤立出来。

异常值会距离*iTree*的根节点更近，而正常值则会距离根节点有更远的距离。

---

**Algorithm 1** : *iForest*( $X, t, \psi$ )

---

**Inputs:**  $X$  - input data,  $t$  - number of trees,  $\psi$  - sub-sampling size

**Output:** a set of  $t$  *iTrees*

- 1: **Initialize** *Forest*
  - 2: set height limit  $l = \text{ceiling}(\log_2 \psi)$
  - 3: **for**  $i = 1$  to  $t$  **do**
  - 4:    $X' \leftarrow \text{sample}(X, \psi)$
  - 5:    $\text{Forest} \leftarrow \text{Forest} \cup \text{iTree}(X', 0, l)$
  - 6: **end for**
  - 7: **return** *Forest*
-

$\epsilon$ 邻域( $\epsilon$  neighborhood, 也称为Eps): 给定对象在半径 $\epsilon$ 内的区域

$$N_{\epsilon}(x) = \{y \in X : \text{dist}(x, y) \leq \epsilon\}$$

密度(density):  $\epsilon$ 领域中 $x$ 的密度, 是一个整数值, 依赖于半径 $\epsilon$

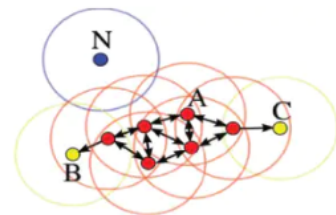
$$p(x) = |N_{\epsilon}(x)|$$

边界点(border point): 如果非核心点 $x$ 的 $\epsilon$ 邻域中存在核心点, 那么认为 $x$ 为 $X$ 的边界点。由 $X$ 中所有的边界点构成的集合为 $X_{bd}$ 。直白来讲, 边界点对应稠密区域边缘的点:

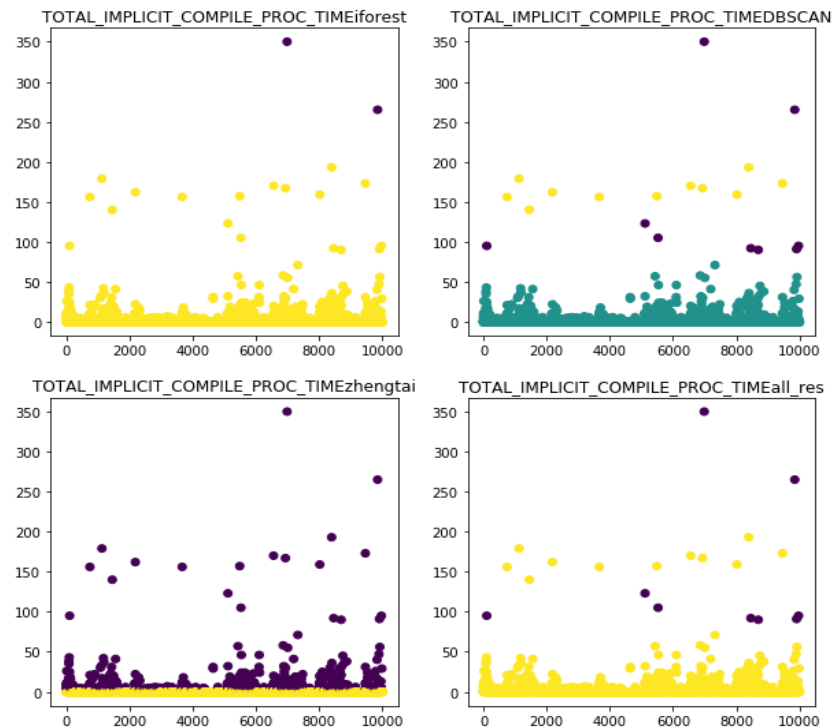
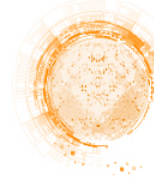
$$x \in X_{nc}; \exists y \in X; y \in N_{\epsilon}(x) \cap X_c$$

噪音点(noise point): 集合中除了边界点和核心点之外的点都是噪音点, 所有噪音点组成的集合叫做 $X_{noi}$ ; 直白来讲, 噪音点对应稀疏区域的点。

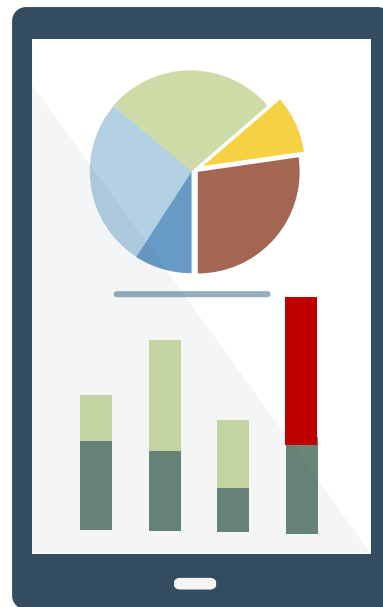
$$X_{noi} = X \setminus (X_c \cup X_{bd})$$



红色为核心点, 黄色为边界点, 蓝色为噪音点



场景名称	日志写盘异常
场景字段名	logdisk_abnormal
相关字段	LOG_DISK_WAITS_TOTALLOG_DISK_WAIT_TIMETOTAL_COMMIT_PROC_TIMETOTAL_COMMIT_TIME
解决方案	<p>当前异常表示数据库写活动日志到磁盘出现异常。异常分析：一种原因是IO延时高，需要警惕。建议结合操作系统，存储信息来一起判断。另一种原因是出现临时大量INSERT UPDATE DELETE操作，写日志成为瓶颈。可以参考当前时间的ROWS_MODIFIED，ROWS_DELETED，ROWS_INSERTED，ROWS_UPDATED判断是否发生突变。解决方案：找到写数据的原因，分析是否是正常业务。如果伴随着LOG_BUFFER_WAIT_TIME，NUM_LOG_BUFFER_FULL异常，可以考虑增加数据库参数Logbufsz的设置。排除是否是磁盘IO缓慢原因，如有问题请联系系统组分析解决。</p>



## 皮尔逊相关系数

皮尔逊积矩相关系数，用于度量两个变量 X 和 Y 之间的相关（线性相关），其值介于 -1 与 1 之间，其中，1 表示变量完全正相关，0 表示无关，-1 表示完全负相关。。两个变量之间的皮尔逊相关系数定义为两个变量之间的协方差和标准差的商。假设有两个变量 X、Y，那么两变量间的皮尔逊相关系数可通过以下公式计算：

$$\text{公式一: } \text{Corr}(X, Y) = \frac{n \sum xy - \sum x \sum y}{\sqrt{n \sum x^2 - (\sum x)^2} \sqrt{n \sum y^2 - (\sum y)^2}}$$

$$\text{公式二: } \text{Corr}(X, Y) = \frac{\sum (X - \bar{X})(Y - \bar{Y})}{\sqrt{\sum (X - \bar{X})^2} \sqrt{\sum (Y - \bar{Y})^2}}$$

## Apriori算法

**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

$$\text{Support}(X, Y) = P(XY) = \frac{\text{number}(XY)}{\text{num}(\text{AllSamples})}$$

**置信度**体现了一个数据出现后，另一个数据出现的概率，或者说数据的条件概率。

$$\text{Confidence}(X \Leftarrow Y) = P(X|Y) = P(XY)/P(Y)$$

**提升度**表示含有Y的条件下，同时含有X的概率，与X总体发生的概率之比

$$\text{Lift}(X \Leftarrow Y) = P(X|Y)/P(X) = \text{Confidence}(X \Leftarrow Y)/P(X)$$



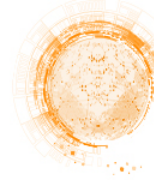
# 民生银行数据库智能运维实践

案例分享

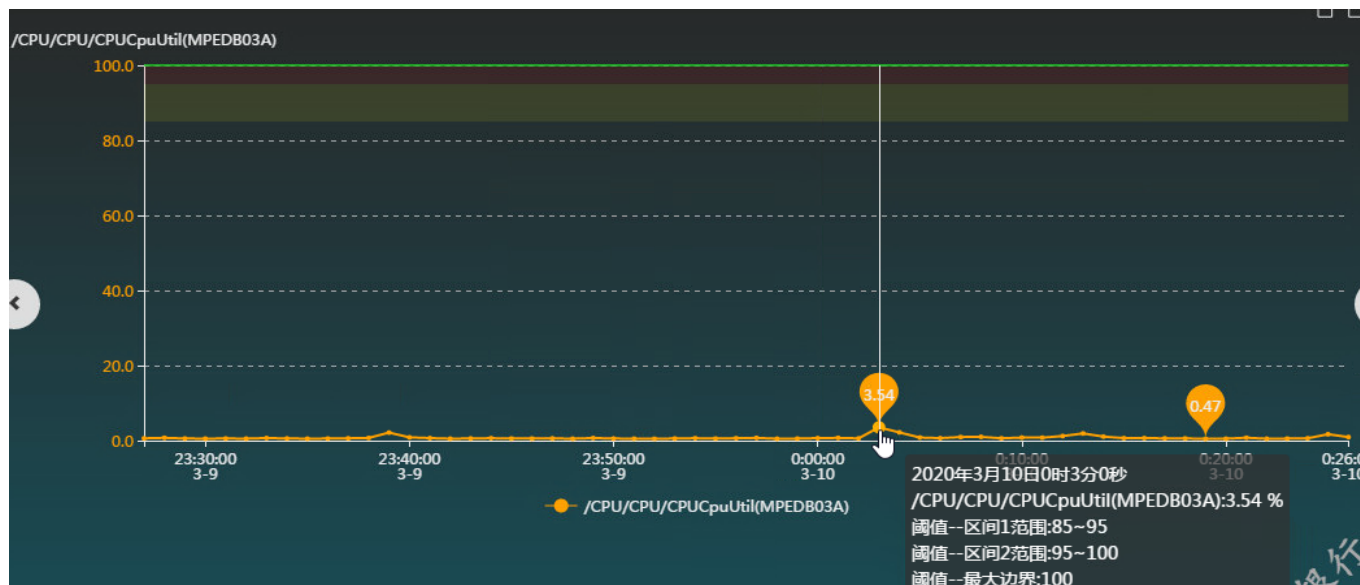


架构革新 11<sup>th</sup>  
自主可控



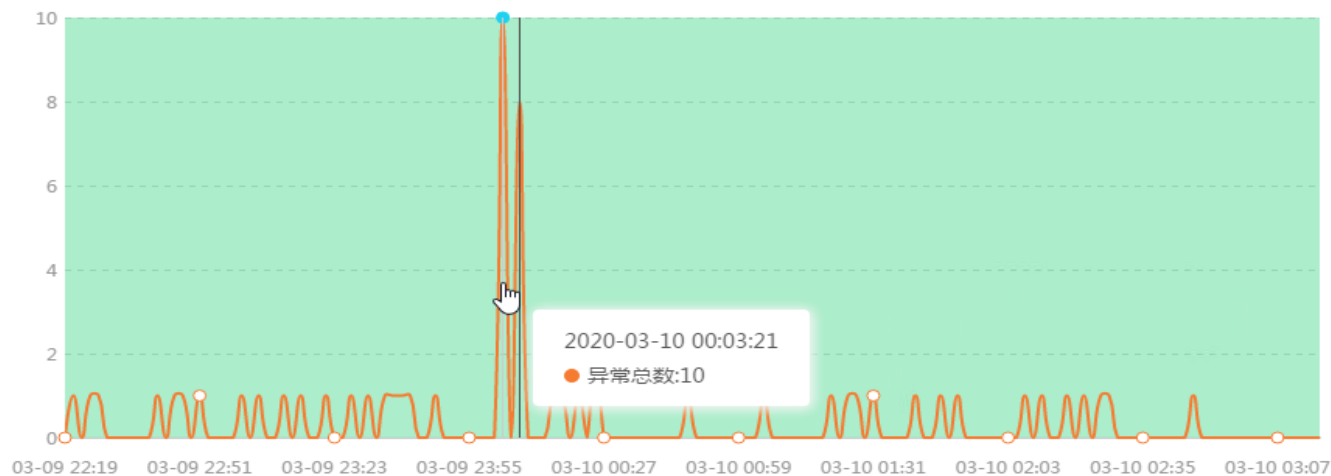


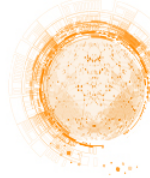
某系统反馈在日终集中挪大量已核销订单的时候，在线交易受到了影响，当时平均交易响应时间1秒左右。最近问题比较突出的时间是2020年3月10日凌晨。  
当时的cpu使用率最高是03分，也只有3.54%。因此硬件资源应该不是问题。



从AIOPS平台检查当时的时间点数据库有什么异常表现。从当时的时间点看，确实有多个指标同时发生了异常。

### 异常告警分布





点击智能分析命中场景发现有日志写磁盘的告警，其中第一个关联sql是影响最大的sql。

## 场景分析

### 场景1:日志写盘 ▼

解析

当前异常表示数据库写活动日志到磁盘出现异常。

异常分析:

1. 一种原因是IO延时长，需要警惕。建议结合操作系统，存储信息来一起判断。
2. 另一种原因是出现临时大量INSERT|UPDATE|DELETE操作，写日志成为瓶颈。可以参考当前时间的ROWS\_MODIFIED, ROWS\_DELETED, ROWS\_INSERTED, ROWS\_UPDATED判断是否发生突变。

解决方案:

1. 找到写数据的原因，分析是否是正常业务。
2. 如果伴随着LOG\_BUFFER\_WAIT\_TIME, NUM\_LOG\_BUFFER\_FULL异常，可以考虑增加数据库参数Logbufsz的设置。
3. log 排除是否是磁盘IO缓慢原因，如有问题请联系系统组分析解决。

异常DB指标

LOG\_DISK\_WAIT\_TIME TOTAL\_COMMIT\_TIME

关联SQL1 ▼

```
create index I_T_ORDER1_20200310
on t_order_20200310 ( PLAT_DATE,
SEQ_NO ) ALLOW REVERSE SCANS
```

SQL指标: LOG\_DISK\_WAIT\_TIME

关联SQL2 ▼



经检查发现当时很多的DDL都发生了等待日志写盘的行为。可见当时对日志写盘压力是比较大的。

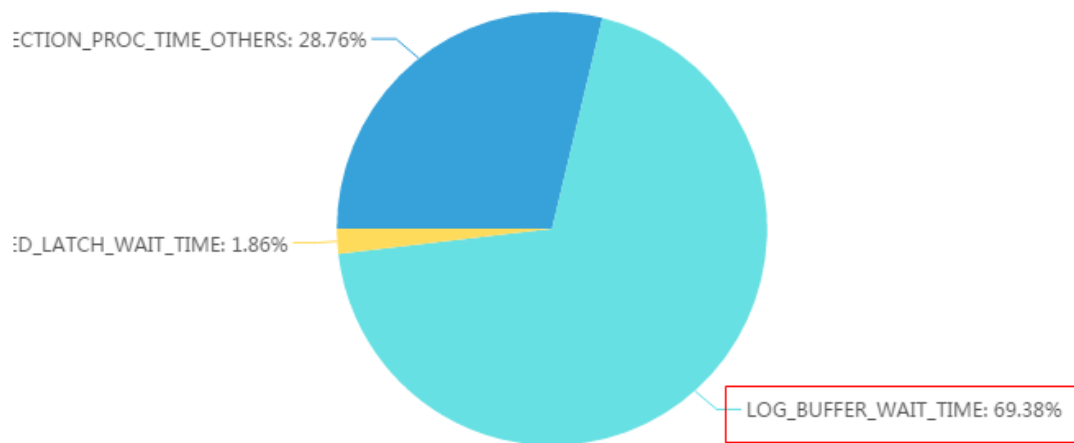
### 👉 SQL指标排行

LOG_DISK_W AIT_TIME	执行时间	执行次数	平均执行 时间	等待时间 占比	SQL文本
79822	92589	16	5786.81	99.91%	create index I_T_ORDER1_20200310 o...
73488	78948	16	4934.25	99.89%	create table t_order_20200310 like t...
9306	22362	16	1397.62	99.72%	create index I_T_ORDER4_20200310 o...
8608	11271	16	704.44	99.43%	create index I_T_ORDER_BCI2_202003...
8353	10599	16	662.44	99.45%	create index I_T_ORDER2_20200310 o...
8173	10977	16	686.06	99.44%	create unique index UIDX_T_ORDER_...
7971	10361	16	647.56	99.47%	create index I_T_ORDER5_20200310 o...
7925	14450	16	903.12	99.57%	create index I_T_ORDER_BCI1_202003...
7060	10227	16	639.19	99.46%	create index I_T_ORDER3_20200310 o...
6660	9331	16	583.19	99.41%	create index I_T_ORDER6_20200310 o...



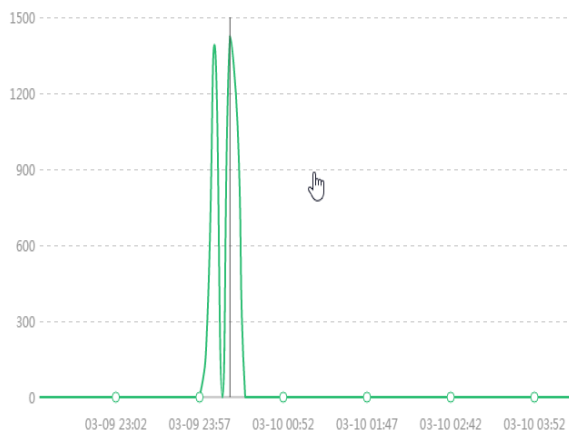
继续查看转订单SQL的执行详情，等待时间占比很高，并且在等待事务写入日志缓存上。

### SQL时间分布图



看一下受到影响的业务语句，在这个时间点出现等待日志缓存，而其他时间点是没有的。

### SQL指标

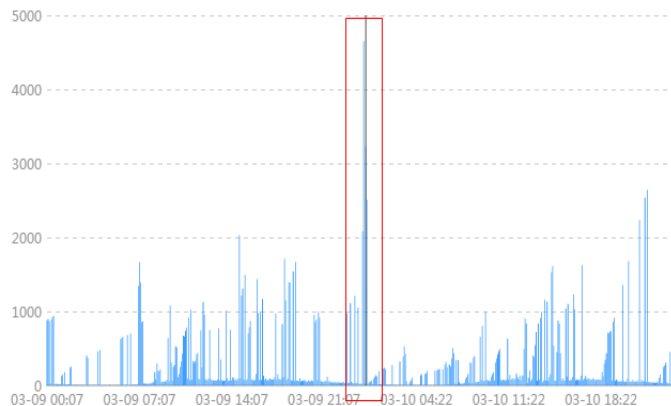


### SQL指标排行

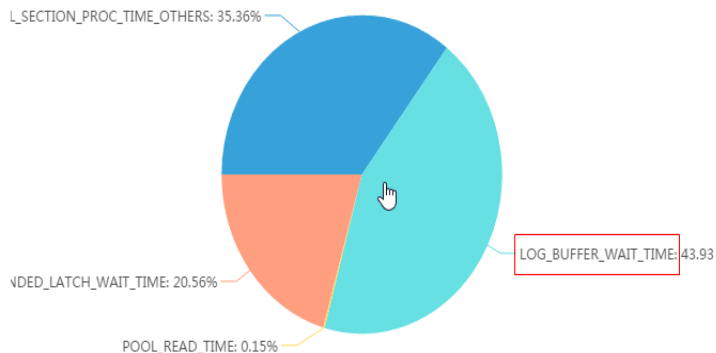
LOG_BUFFER_WAIT_TIME	执行时间	执行次数	平均执行时间	等待时间占比	SQL文本
2637	3021	2241	1.35	95.27%	INSERT INTO t_chn_seqno_reg (PLAT...
1425	3244	8661	0.37	64.64%	INSERT INTO t_order (ORDER_ID, PLA...

从图中可以看到，业务sql当时出现了日志缓存等待，而其他时间没有。另外的latch等待是次生产物。

SQL执行时间



SQL时间分布图





## 分析结论：

最终的分析结果是大量的ddl语句但是产生了大量日志，造成日志写盘繁忙。同一时间内，大量订单转表语句执行，对日志缓存需求较高，导致业务语句使用日志缓存也受到影响，因此造成了业务sql抖动。

## 优化建议：

数据库参数调整： 建议本次维护窗口调整数据库日志缓存为64M。

应用优化建议：

1. 建表建索引等语句建议和跑批错开，减少资源竞争
2. 建议跑批语句拆分为更细粒度事务。减少跑批并发，错峰跑批。

# 智能运维与DBA的工作



架构革新 11<sup>th</sup>  
自主可控



智能化

## 智运运维

- 都智能运维了，都AI了，还要DBA干什么？
- 要做个懂AI的DBA
- 为AIOps整理提供数据
- 在AIOps中建立智能场景
- 从AIOps中获取知识
- AIOps让DBA的经验和能力更上一层楼

自动化

## 工具运维

- 建造各种运维工具实现监控管理等工作需求
- 自动化变更发布
- 应急，调优

人力

## 经验运维

- 装机，变更
- 监控，应急
- 巡检，调优

# THANKS

