微服务应用系统故障发现和根因定位解题思路与算法迭代

张世泽、赵鋆峰

一行bug

网络科学与网络空间研究院

清華大學
Tsinghua University

目录

- 团队介绍
- baseline方法
- 准确性提升
- 速度提升
- 可能的不足

清华大学
Tsinghua University

目录

- 团队介绍
- baseline方法
- 准确性提升
- 速度提升
- 可能的不足

团队介绍

- 成员
  - 张世泽　　博士研究生　　清华大学网络科学与网络空间研究院
  - 赵鋆峰　　硕士研究生　　清华大学网络科学与网络空间研究院
  - 指导老师：杨家海、王之梁老师

实验室主页地址

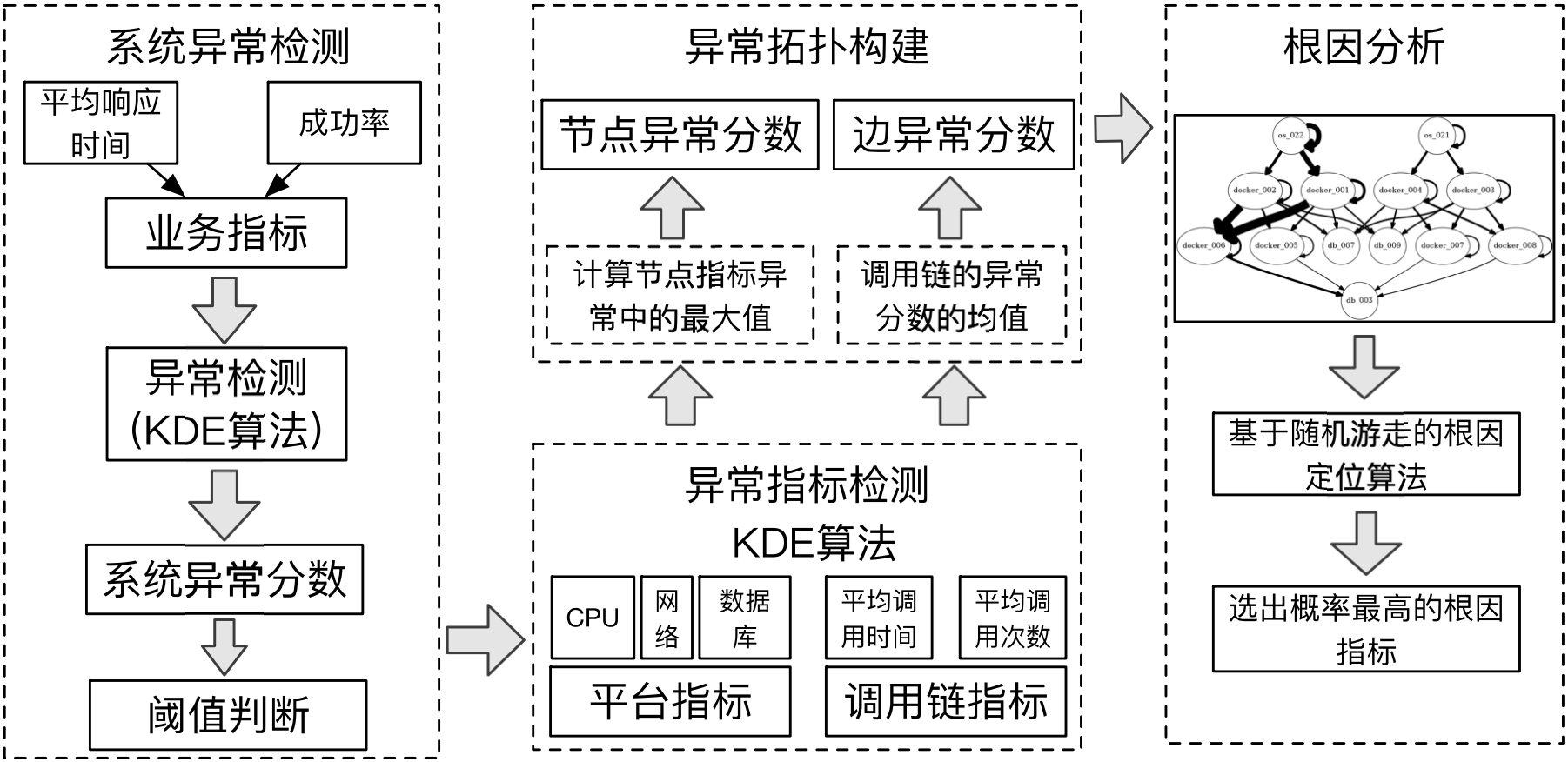
http://nmgroup.tsinghua.edu.cn

清華大學
Tsinghua University

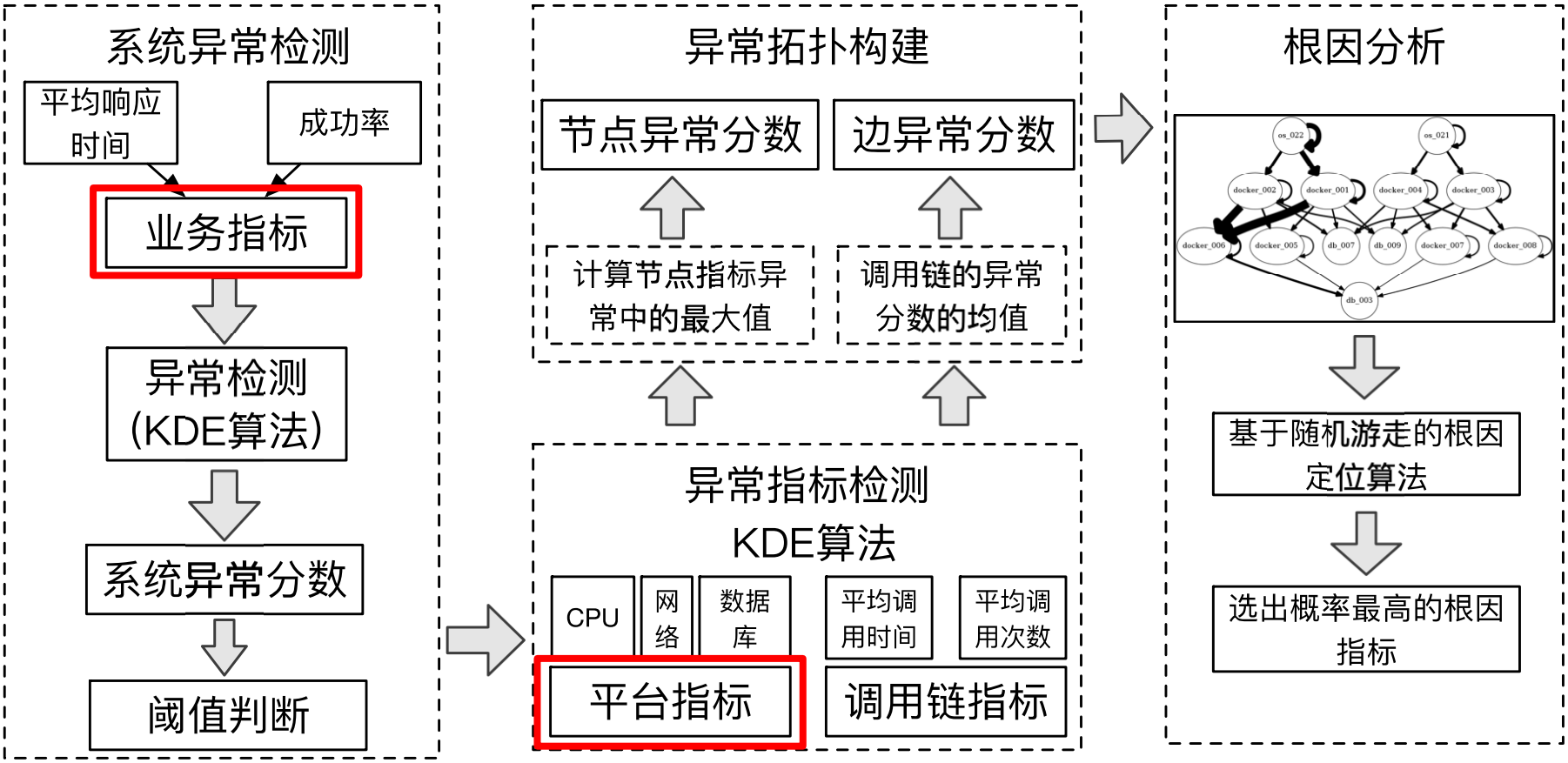
目录

- 团队介绍
- baseline方法
- 准确性提升
- 速度提升
- 可能的不足

算法整体框架


系统异常检测
平均响应时间
成功率
业务指标
异常检测(KDE算法)
系统异常分数
阈值判断
异常拓扑构建
节点异常分数
边异常分数
计算节点指标异常中的最大值
调用链的异常分数的均值
异常指标检测
KDE算法
CPU
网络
数据库
平均调用时间
平均调用次数
平台指标
调用链指标
根因分析
os_022
os_021
docker_002
docker_001
docker_004
docker_003
docker_006
docker_005
db_007
db_009
docker_007
docker_008
db_003
基于随机游走的根因定位算法
选出概率最高的根因指标

业务、平台指标


系统异常检测
平均响应时间
成功率
业务指标
异常检测(KDE算法)
系统异常分数
阈值判断
异常拓扑构建
节点异常分数
边异常分数
计算节点指标异常中的最大值
调用链的异常分数的均值
异常指标检测
KDE算法
CPU
网络
数据库
平均调用时间
平均调用次数
平台指标
调用链指标
根因分析
基于随机游走的根因定位算法
选出概率最高的根因指标

业务、平台指标

应用需求

通用

轻量

无需训练

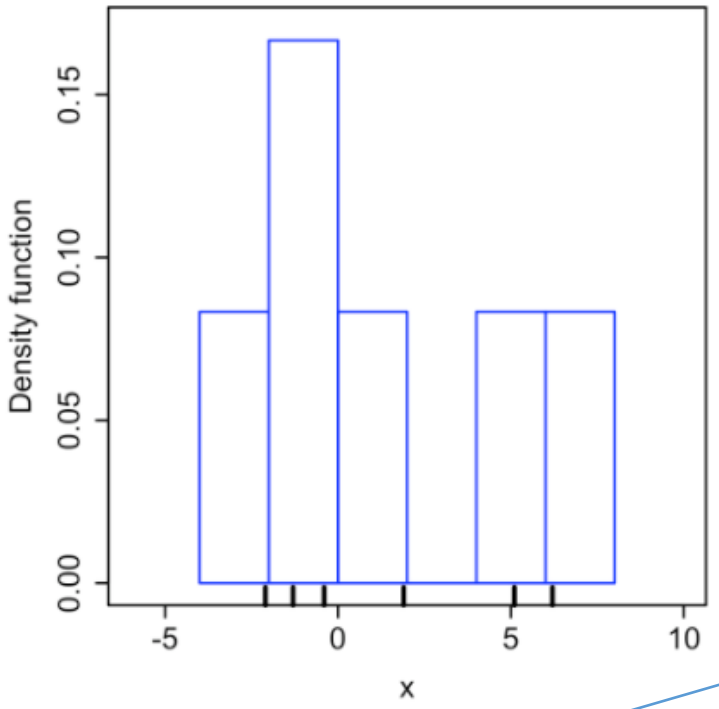
可以输出异常分数

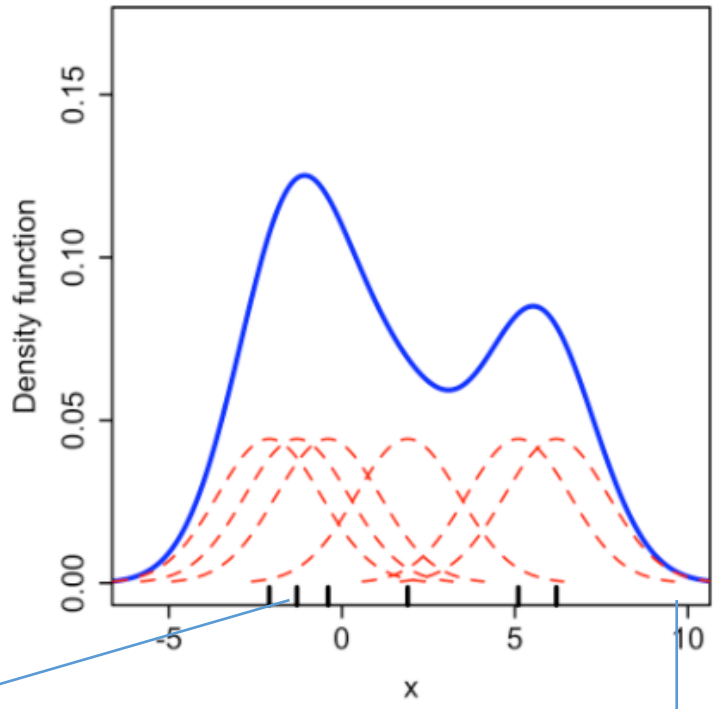
数据特点

数据整体平稳

异常局部明显

核函数估计KDE



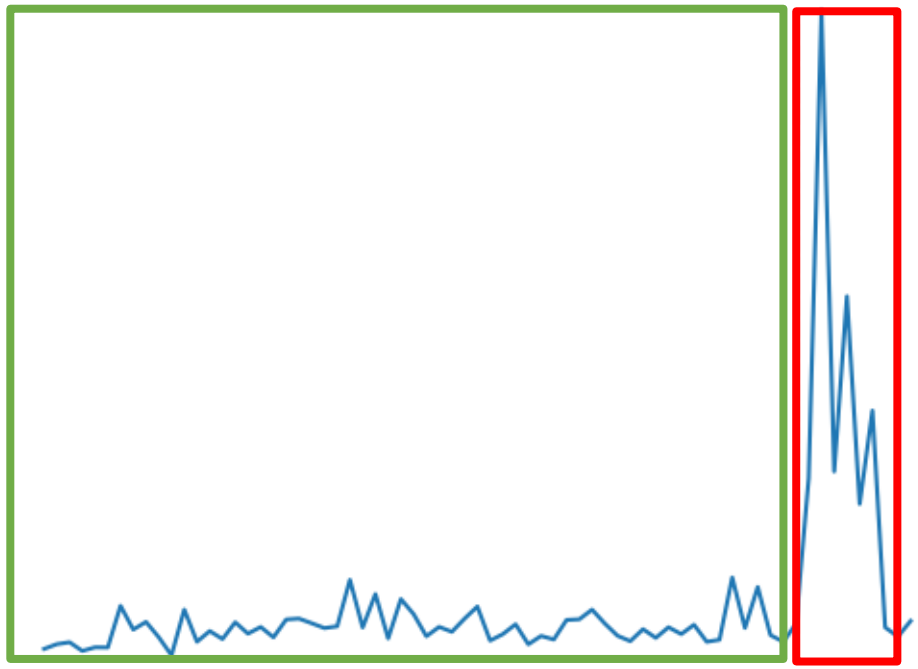


$-\log(0.12) \approx 2.12$

$-\log(0.005) \approx 5.30$

[1] Liu P, Chen Y, Nie X, et al. FluxRank: A Widely-Deployable Framework to Automatically Localizing Root Cause Machines for Software Service Failure Mitigation[C]//2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE). IEEE, 2019: 35-46.

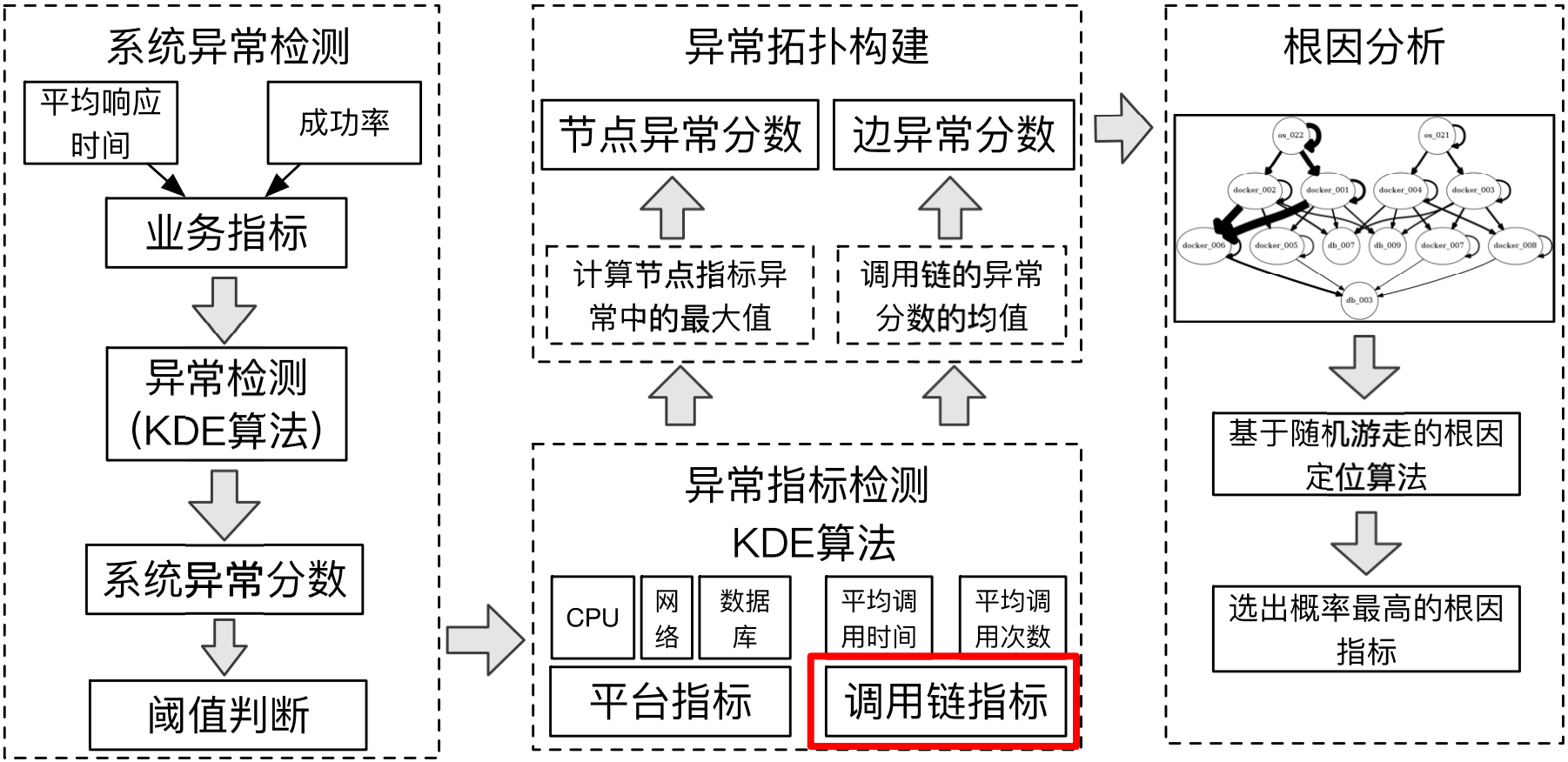
核函数估计KDE



构建KDE模型，得到概率密度P

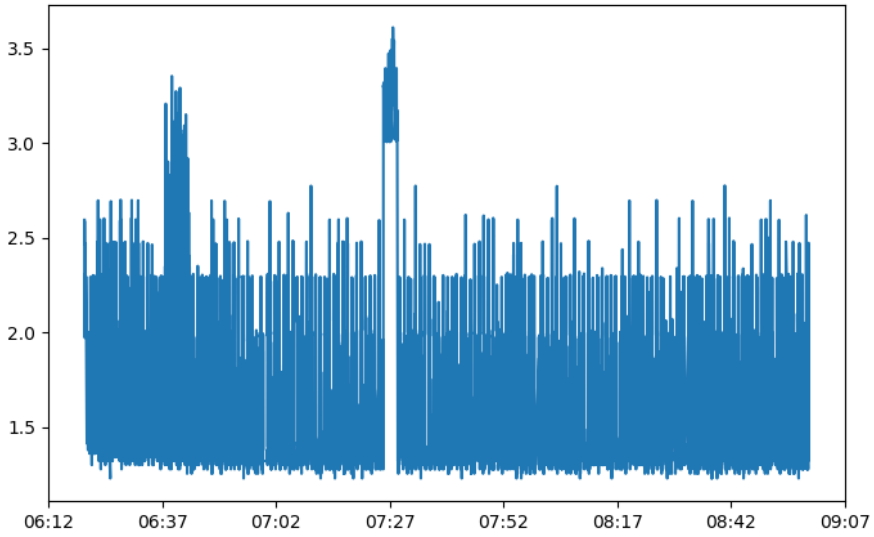
$\max(-\log(P(x)))$作为异常分数

算法整体框架

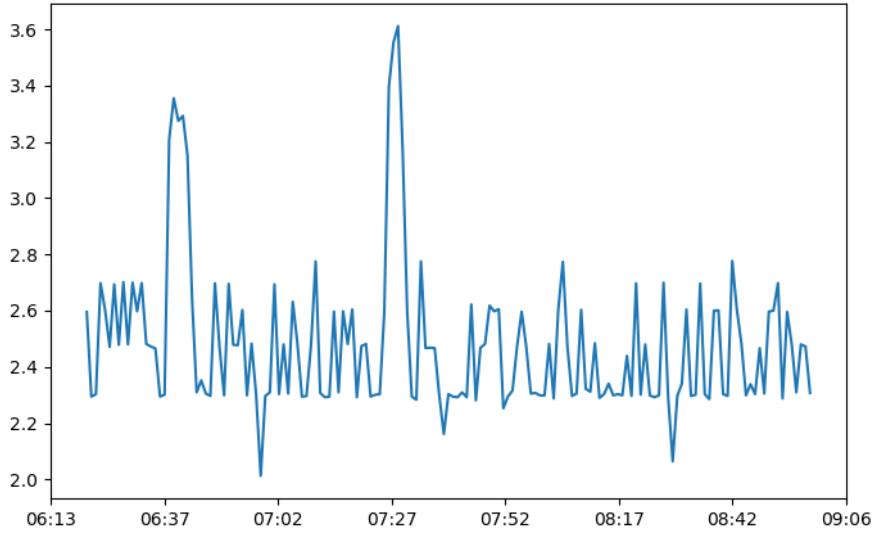

系统异常检测
平均响应时间
成功率
业务指标
异常检测(KDE算法)
系统异常分数
阈值判断
异常拓扑构建
节点异常分数
边异常分数
计算节点指标异常中的最大值
调用链的异常分数的均值
异常指标检测
KDE算法
CPU
网络
数据库
平均调用时间
平均调用次数
平台指标
调用链指标
根因分析
基于随机游走的根因定位算法
选出概率最高的根因指标

调用链指标

- 原始数据
  - 数据量很大？
    - 100 +条 / min
  - 网络丢包 & 网络延时

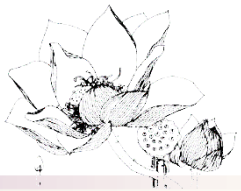


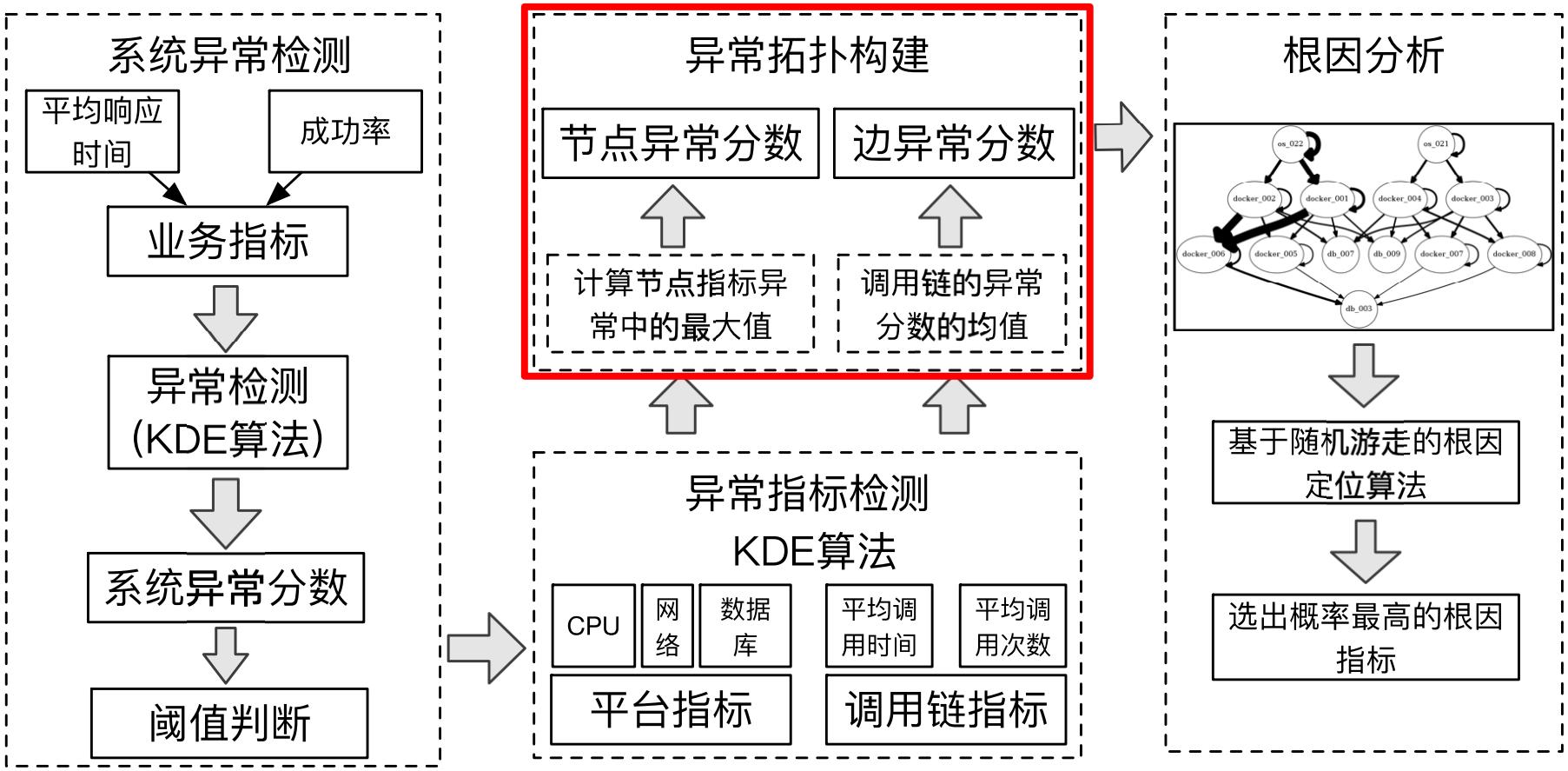
- 统计数据
  - 转化成每分钟内的平均调用时间
    - 1 条/ min
  - 依旧能够检测出两种网络故障



Tsinghua University

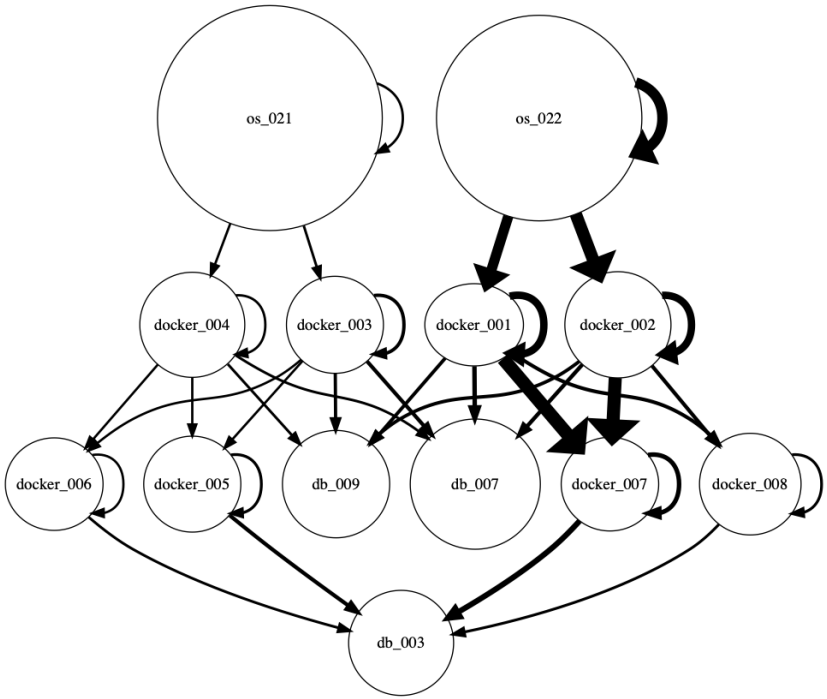
算法整体框架



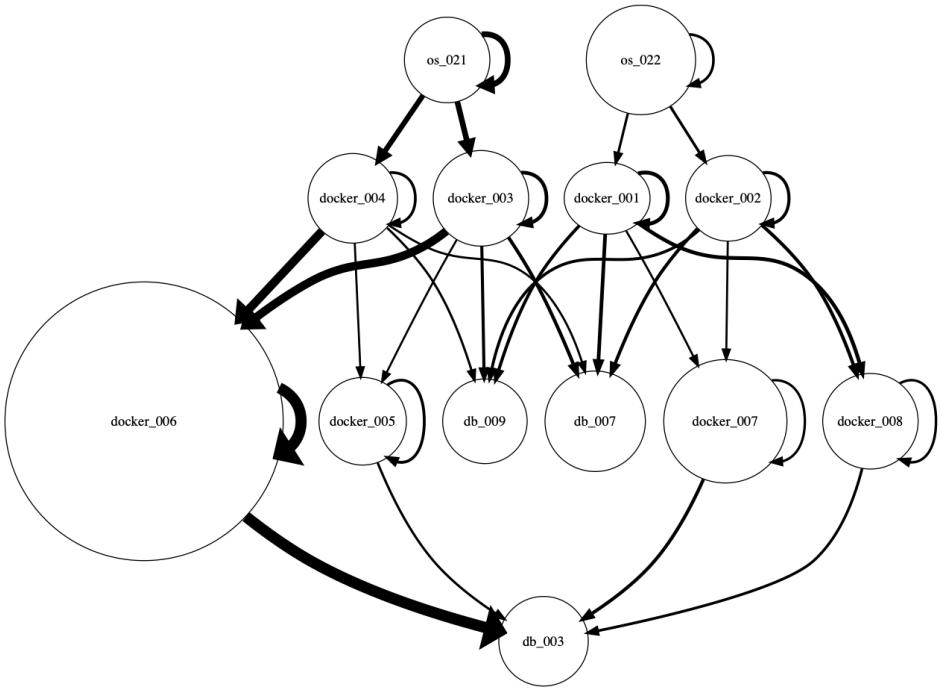


异常图部分示例

把异常分数与静态拓扑图结合起来



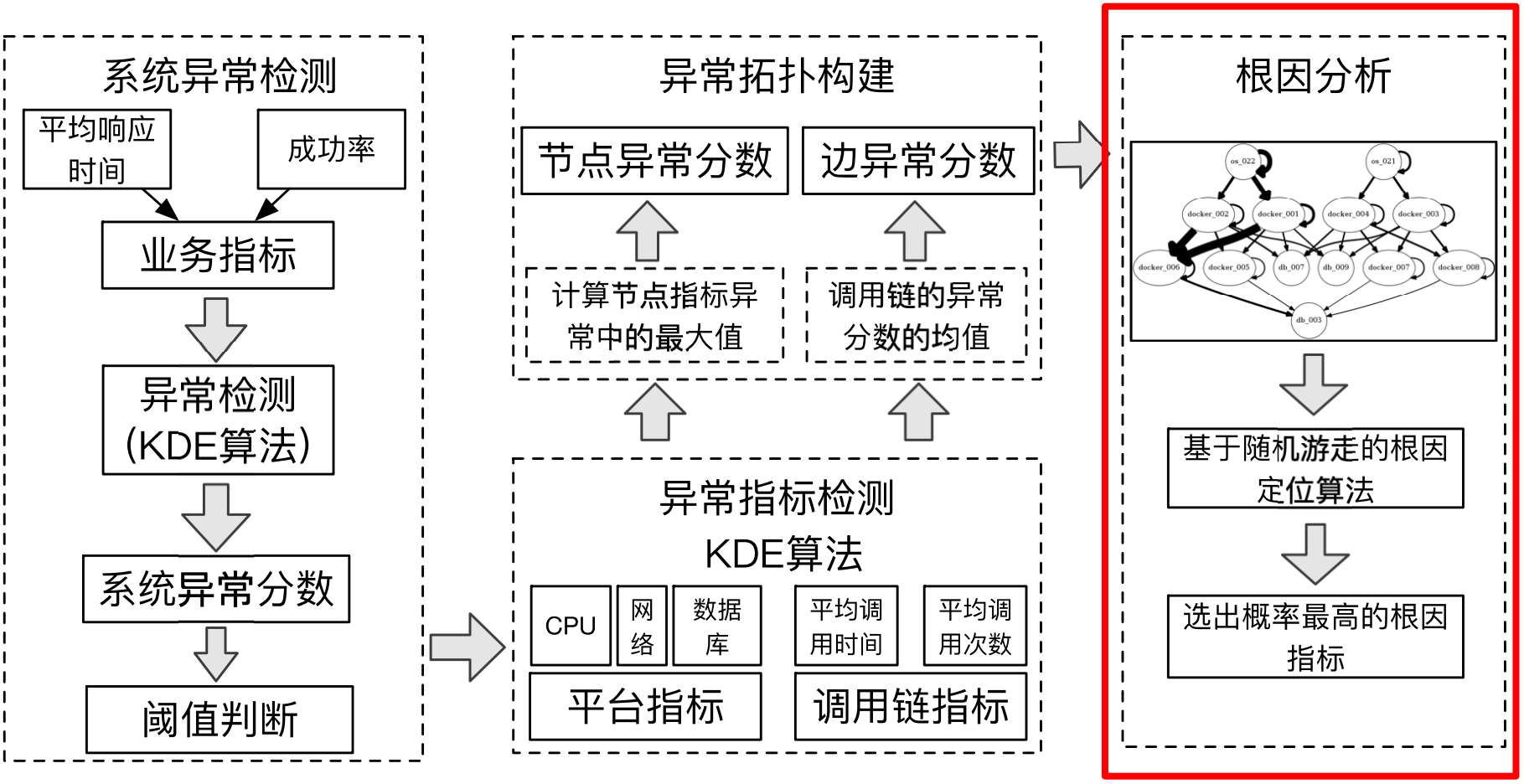
docker_007网络



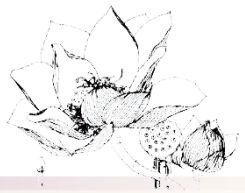
docker_006 CPU

Tsinghua University

算法整体框架



随机游走



- 如何让算法在有明显异常传播关系的图上找到根因呢？
  - 随机游走

$$q_{i,j} = \begin{cases} w_{i,j}, & if\ (i,j) \in E \\ \rho_{back} w_{j,i}, & if\ (i,j) \notin E\ and\ (j,i) \in E \\ \rho_{self} v_i + \rho_{in} \frac{\sum_{(j,i)\in E} w_{j,i}}{\sum_{(j,i)\in E} 1} + \rho_{out} \frac{\sum_{(i,j)\in E} w_{i,j}}{\sum_{(i,j)\in E} 1}, & if\ i = j \end{cases}$$

Algorithm 6 随机游走过程

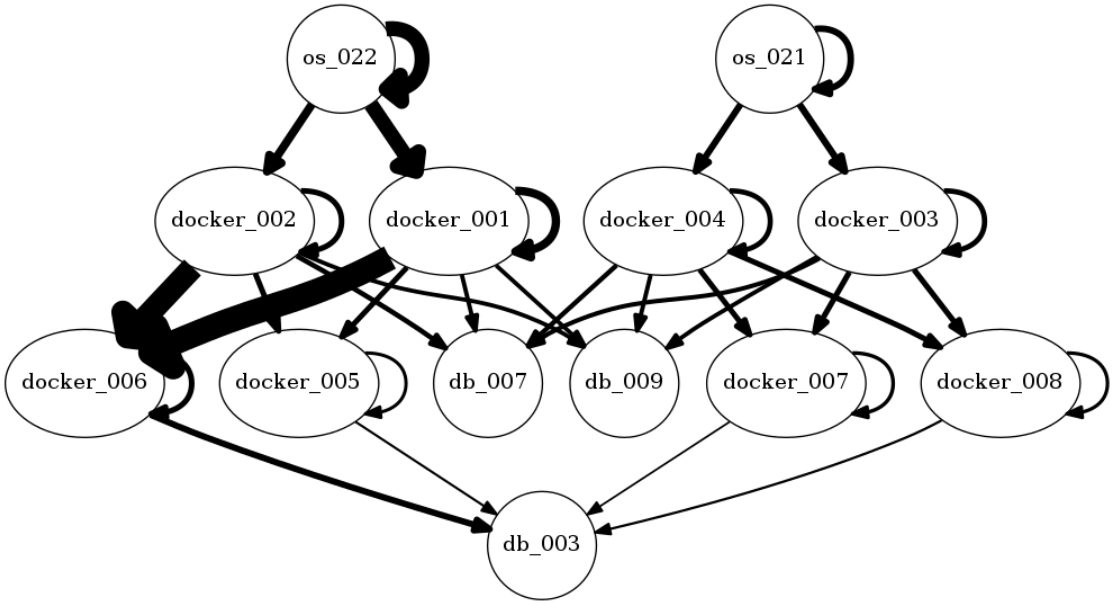
输入: $p, V, \overline{w}, step$

输出: R

```
v ←randomly choose from V
repeat
    r ← random(0, 1)
    if r < p then
        v ←randomly choose from V
    else
        v ←randomly choose j by q̄_{v,j} probability
    end if
    R_i ← R_i + 1
until step rounds
sort R in descending order
```

[2] Weng J, Wang J H, Yang J, et al. Root cause analysis of anomalies of multitier services in public clouds[J]. IEEE/ACM Transactions on Networking, 2018, 26(4): 1646-1659.

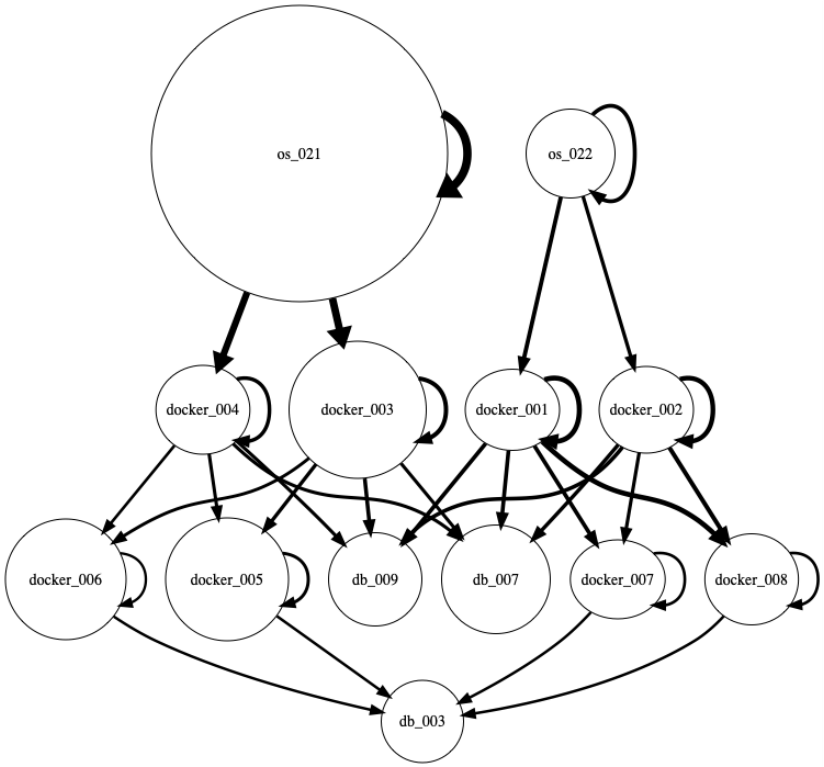
评估示例一



docker_006的网络故障

```
Random walk answer is :
  docker_006: 3427
  docker_001: 1593
  docker_002: 863
  docker_003: 860
  docker_004: 857
  docker_007: 579
  docker_008: 496
  os_021: 447
  db_007: 285
  db_009: 237
  os_022: 160
  docker_005: 99
```

评估示例二



os_021的网络故障

```
Random walk answer is :
  os_021: 1554
  docker_003: 1162
  docker_001: 1151
  docker_004: 919
  docker_008: 882
  docker_002: 835
  docker_007: 639
  docker_005: 583
  db_007: 570
  db_009: 536
  docker_006: 458
```

清華大學
Tsinghua University

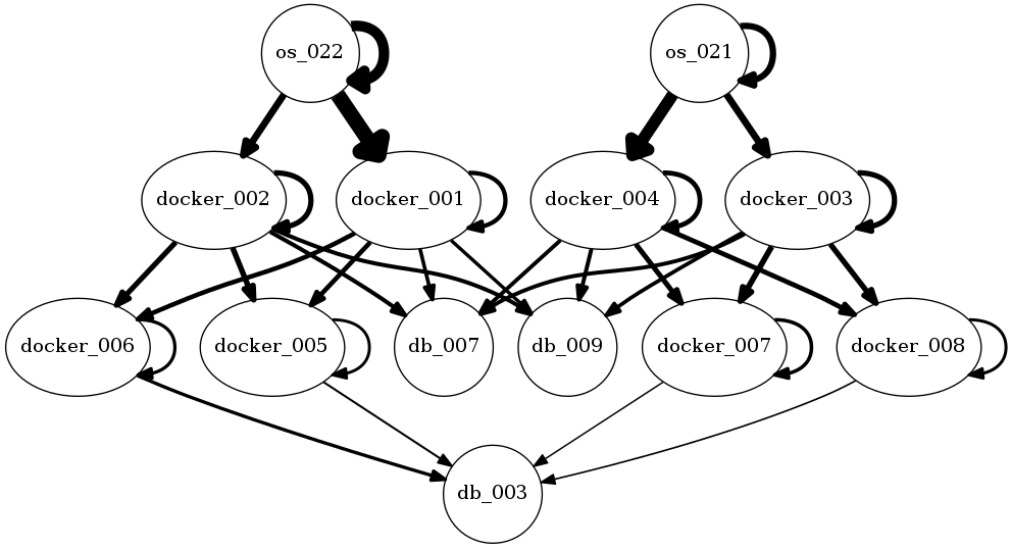
目录

- 团队介绍
- baseline方法
- 准确性提升
- 速度提升
- 可能的不足

存在的准确性问题

Baseline方法存在的准确性问题

- 多个根因
- 多个docker所属的统一os网络故障
- 随机游走结果偏差
- 相邻异常对KDE效果干扰



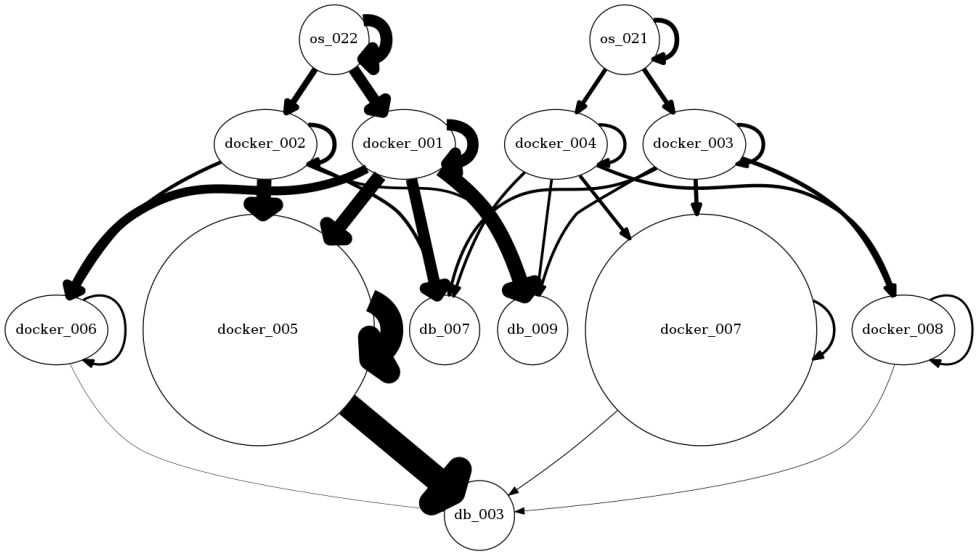
docker_001和docker_004的网络故障

```
Random walk answer is
  docker_001: 2406
  docker_004: 1806
  os_022: 1303
  os_021: 901
  docker_003: 744
  docker_002: 701
  docker_006: 474
  docker_007: 444
  docker_008: 335
  docker_005: 301
  db_009: 235
  db_007: 234
  db_003: 116
```

存在的准确性问题

Baseline方法存在的准确性问题

- 多个根因
- 多个docker所属的统一os网络故障
- 随机游走结果偏差
- 相邻异常对KDE效果干扰



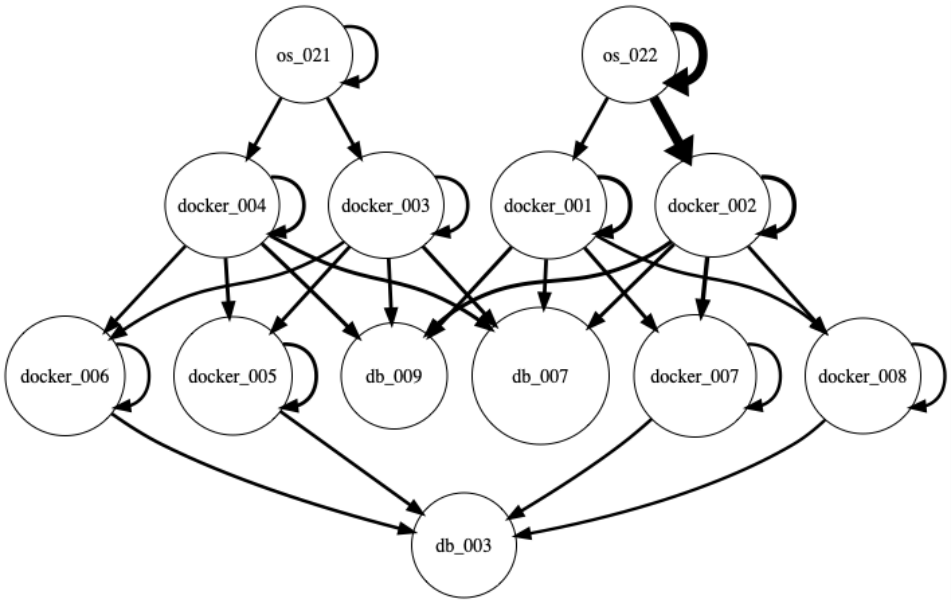

```
Random walk answer is :
  docker_005: 2621
  db_003: 1897
  docker_007: 1287
  docker_001: 1099
  db_009: 782
  docker_003: 564
  docker_008: 352
  docker_004: 338
  os_021: 294
  os_022: 280
  db_007: 249
```

docker_001和docker_005同属的os_017的网络故障

存在的准确性问题

Baseline方法存在的准确性问题

- 多个根因
- 多个docker所属的统一os网络故障
- 随机游走结果偏差
- 相邻异常对KDE效果干扰



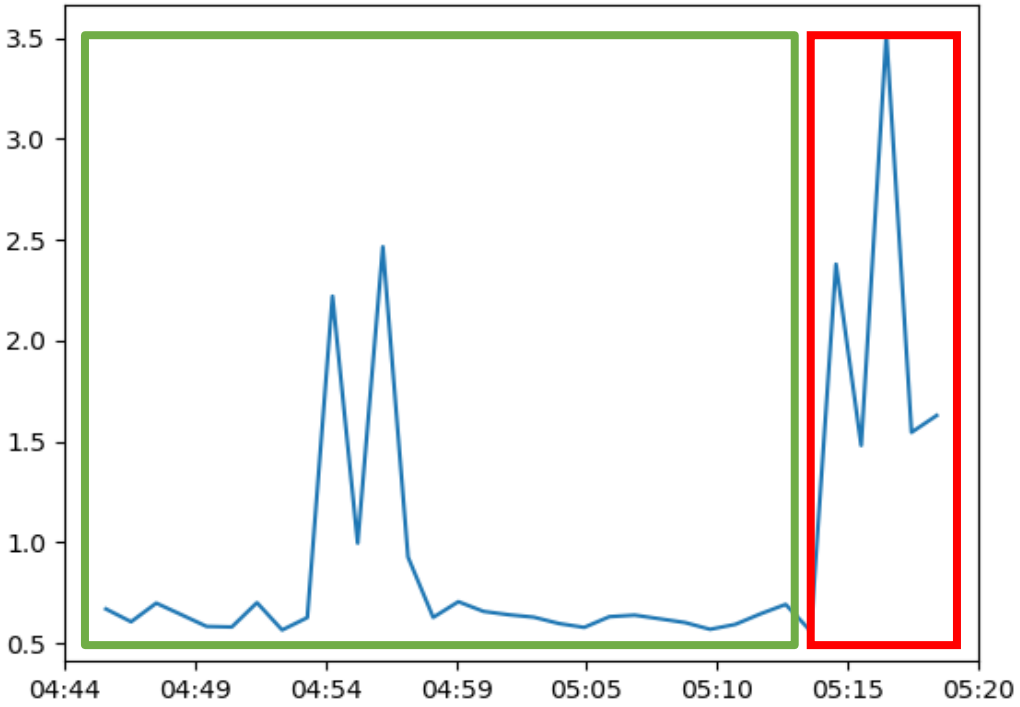
docker_002的网络故障

```
Random walk answer is :
  os_022: 3078
  docker_002: 3001
  docker_001: 474
  docker_004: 461
  docker_007: 441
  docker_003: 421
  db_007: 418
  db_009: 385
  docker_005: 377
  docker_006: 300
```

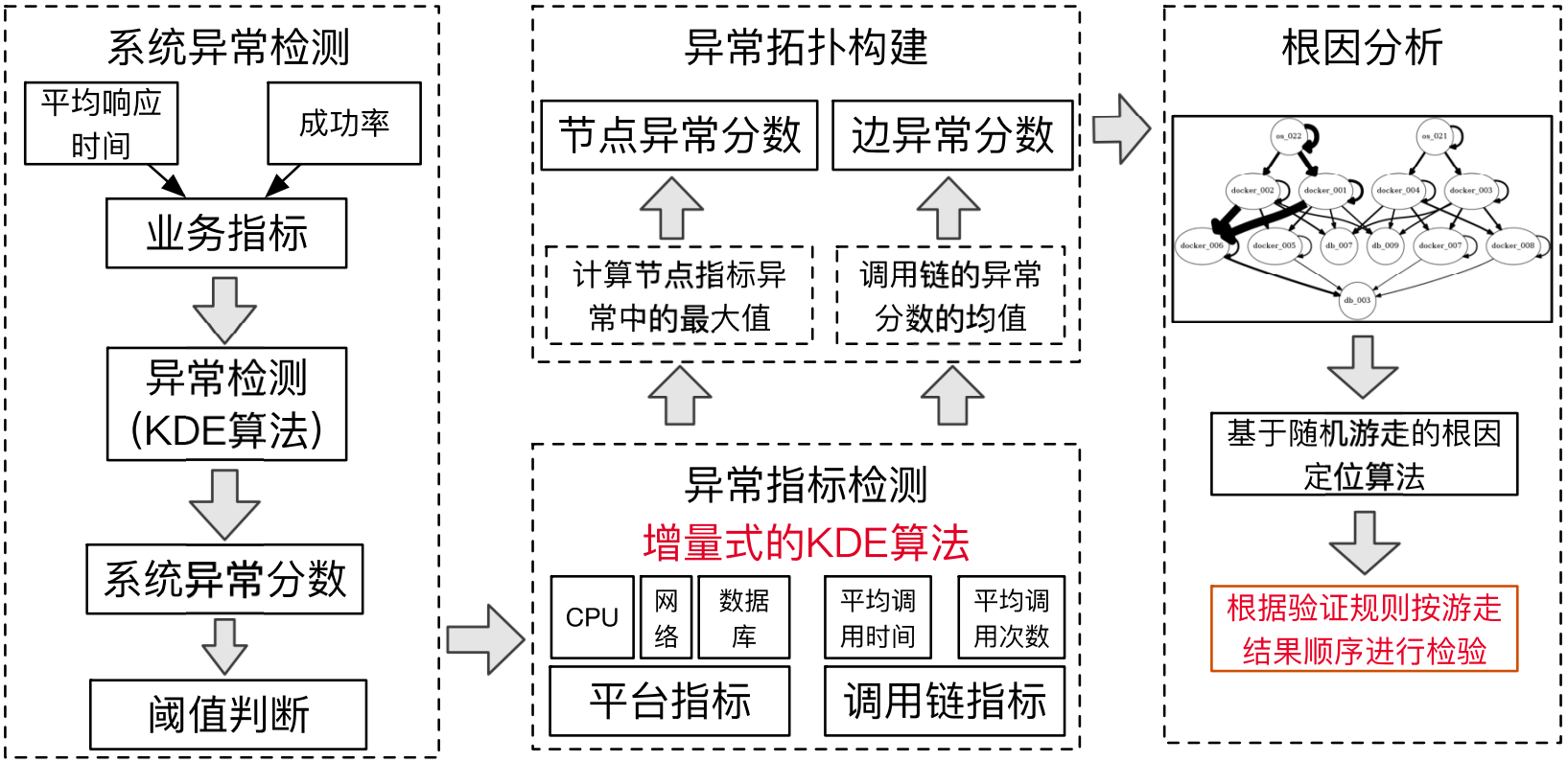
存在的准确性问题

Baseline方法存在的准确性问题

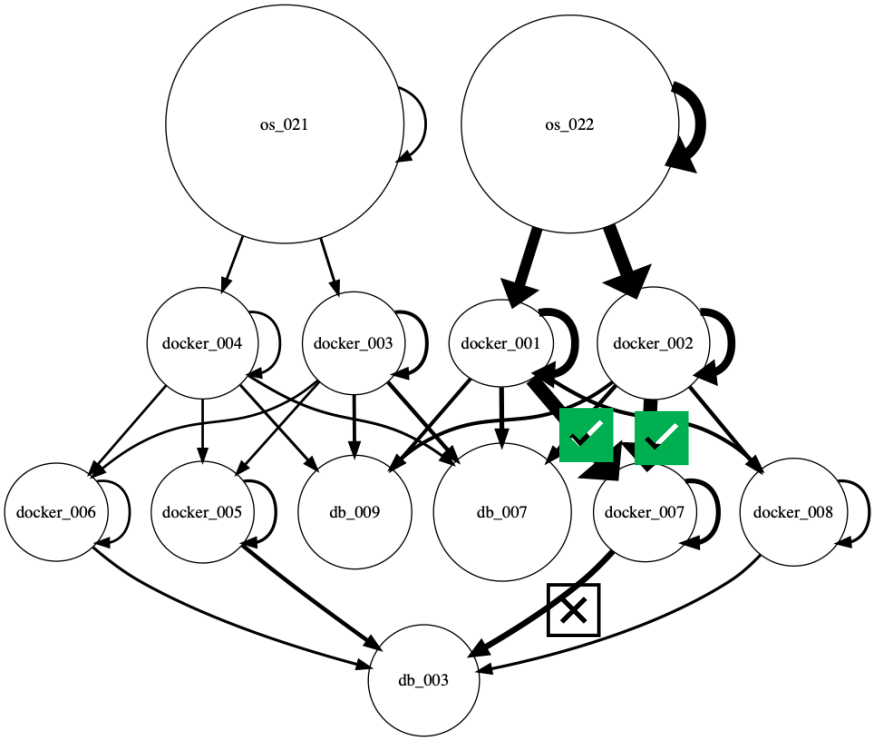
- 多个根因
- 多个docker所属的统一os网络故障
- 随机游走结果偏差
- 相邻异常对KDE效果干扰



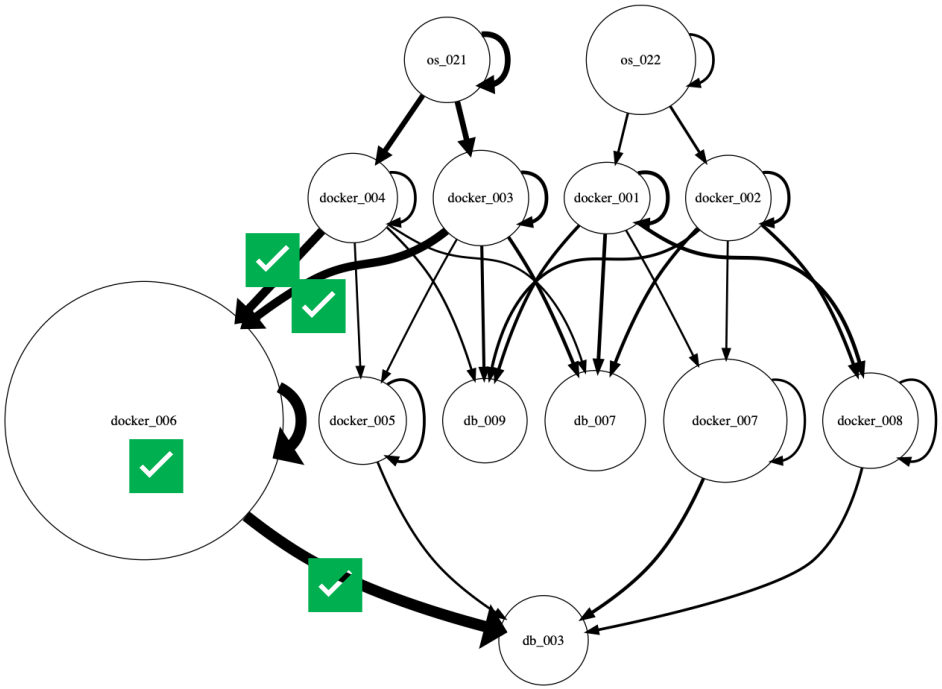
算法整体框架


系统异常检测
平均响应时间
成功率
业务指标
异常检测(KDE算法)
系统异常分数
阈值判断
异常拓扑构建
节点异常分数
边异常分数
计算节点指标异常中的最大值
调用链的异常分数的均值
异常指标检测
增量式的KDE算法
CPU
网络
数据库
平均调用时间
平均调用次数
平台指标
调用链指标
根因分析
基于随机游走的根因定位算法
根据验证规则按游走结果顺序进行检验

专家经验验证



对于网络故障的验证

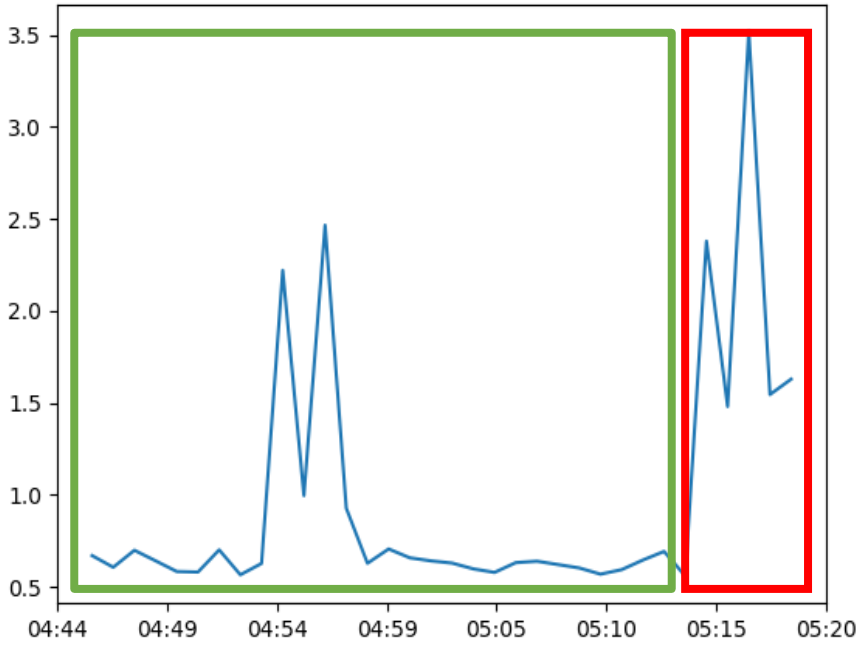


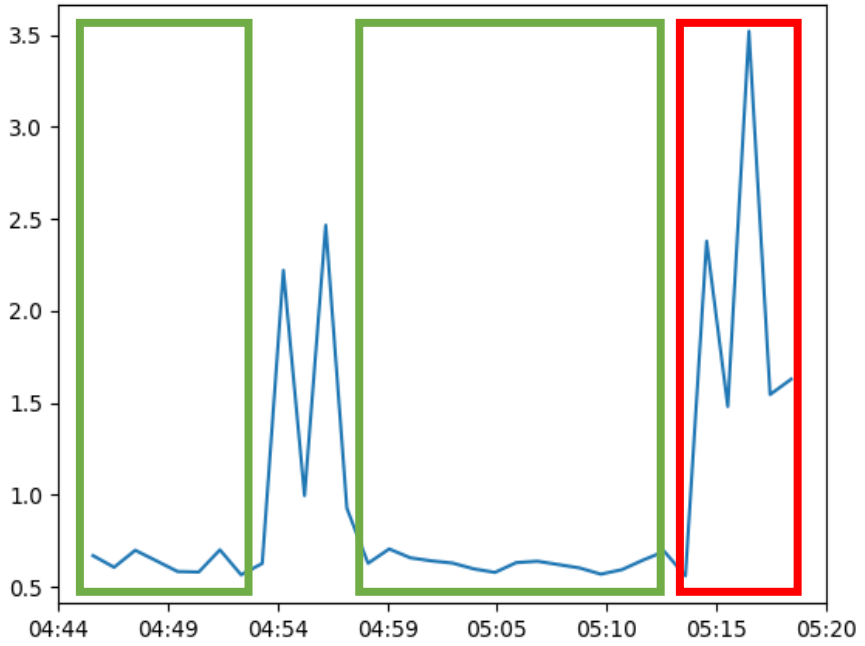
对于CPU的验证

……

增量式的KDE算法

- 维护一个历史的正常数据（固定大小）及其构建的KDE模型
  - 新来的历史数据，如果用KDE模型得到的异常分数 < threshold，才将其加入正常数据中
  - 为了防止太多的重构，每次更新后以一定几率重构





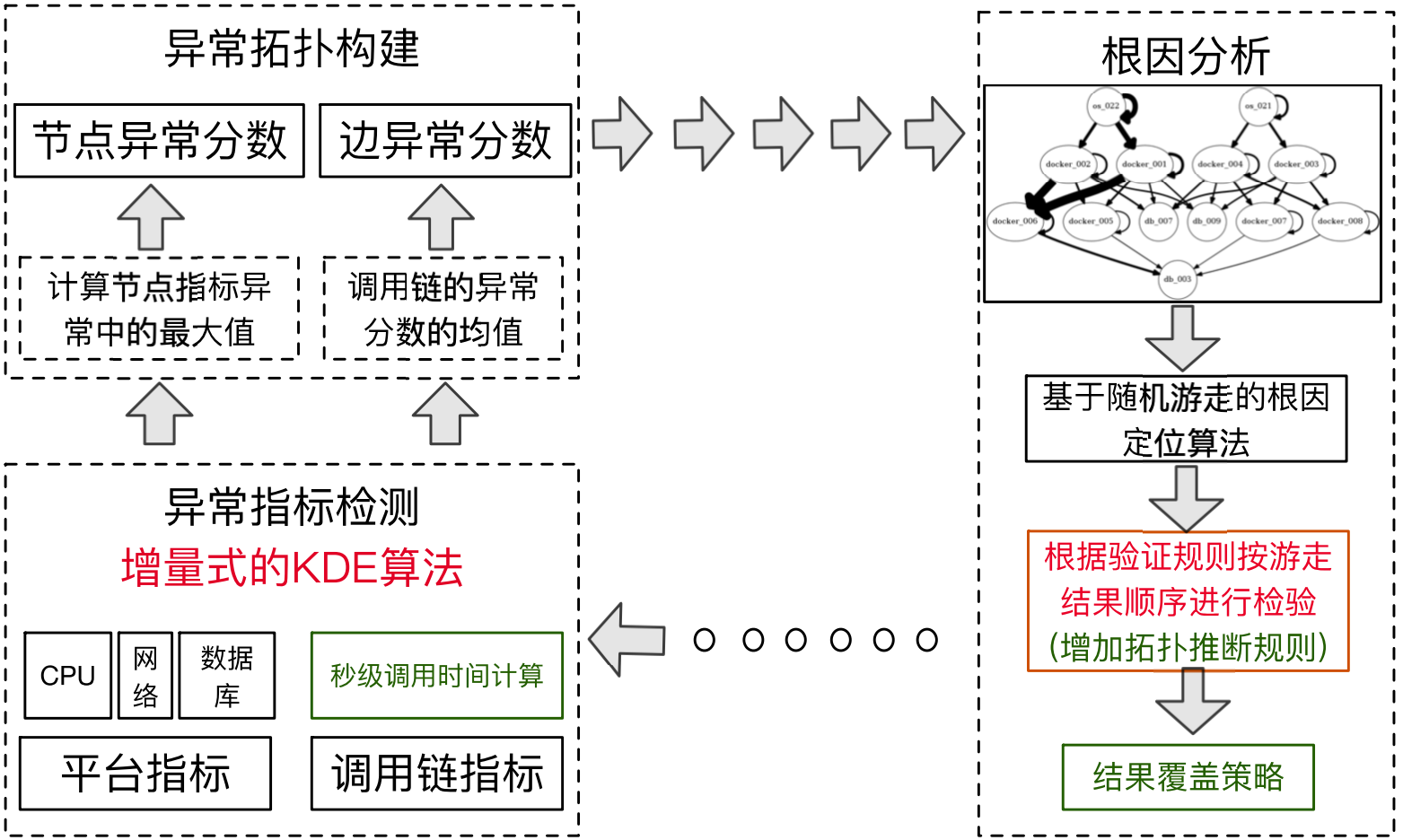
目录

- 团队介绍
- baseline方法
- 准确性提升
- 速度提升
- 可能的不足

速度提升

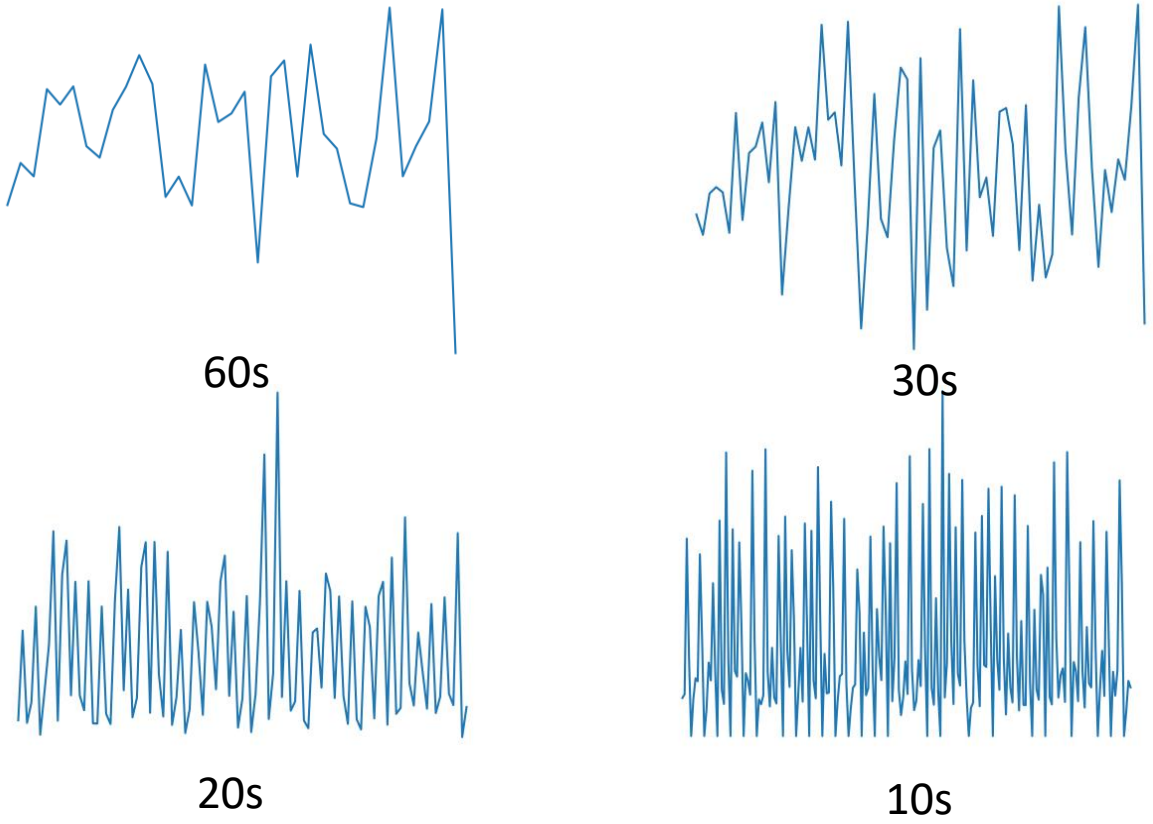
- 存在的问题
  - 调用链指标聚合慢
  - 指标采样频率不同、相位不同
  - 业务指标延时一分钟
- 解决方案
  - 秒级累计平均延时统计分析
  - 拓扑推断
  - 主动探测 + 结果覆盖策略

算法整体框架

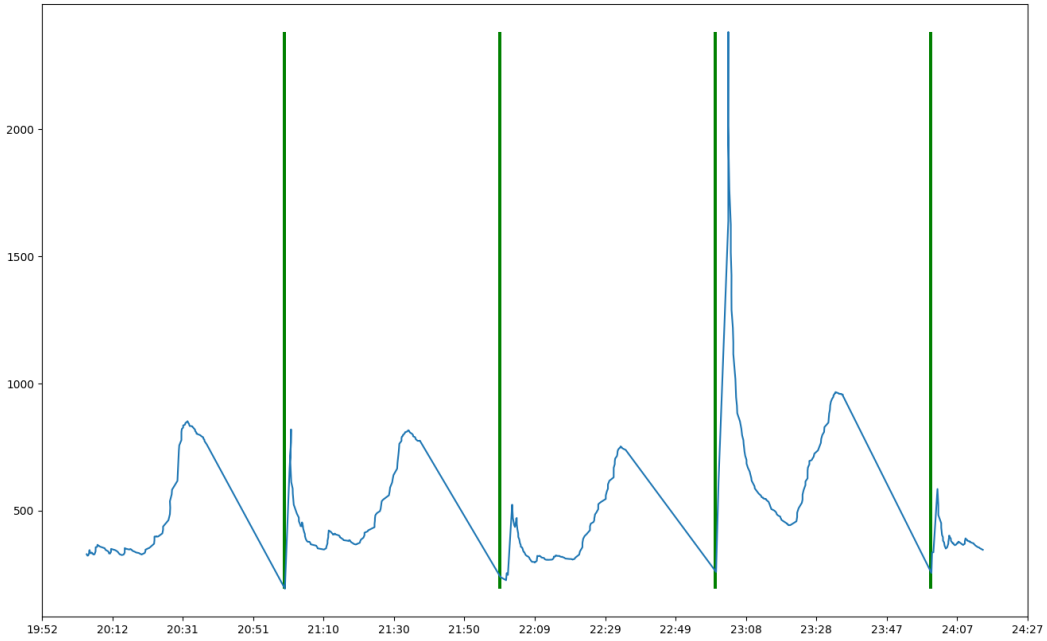

异常拓扑构建
节点异常分数
边异常分数
计算节点指标异常中的最大值
调用链的异常分数的均值
异常指标检测
增量式的KDE算法
CPU
网络
数据库
秒级调用时间计算
平台指标
调用链指标
根因分析
os_022
os_021
docker_002
docker_001
docker_004
docker_003
docker_006
docker_005
db_007
db_009
docker_007
docker_008
db_003
基于随机游走的根因定位算法
根据验证规则按游走结果顺序进行检验
(增加拓扑推断规则)
结果覆盖策略

秒级累计平均延时统计分析

- 是否可以通过缩小窗口的大小来加快得到数据的速度？
  - 答案是不可以



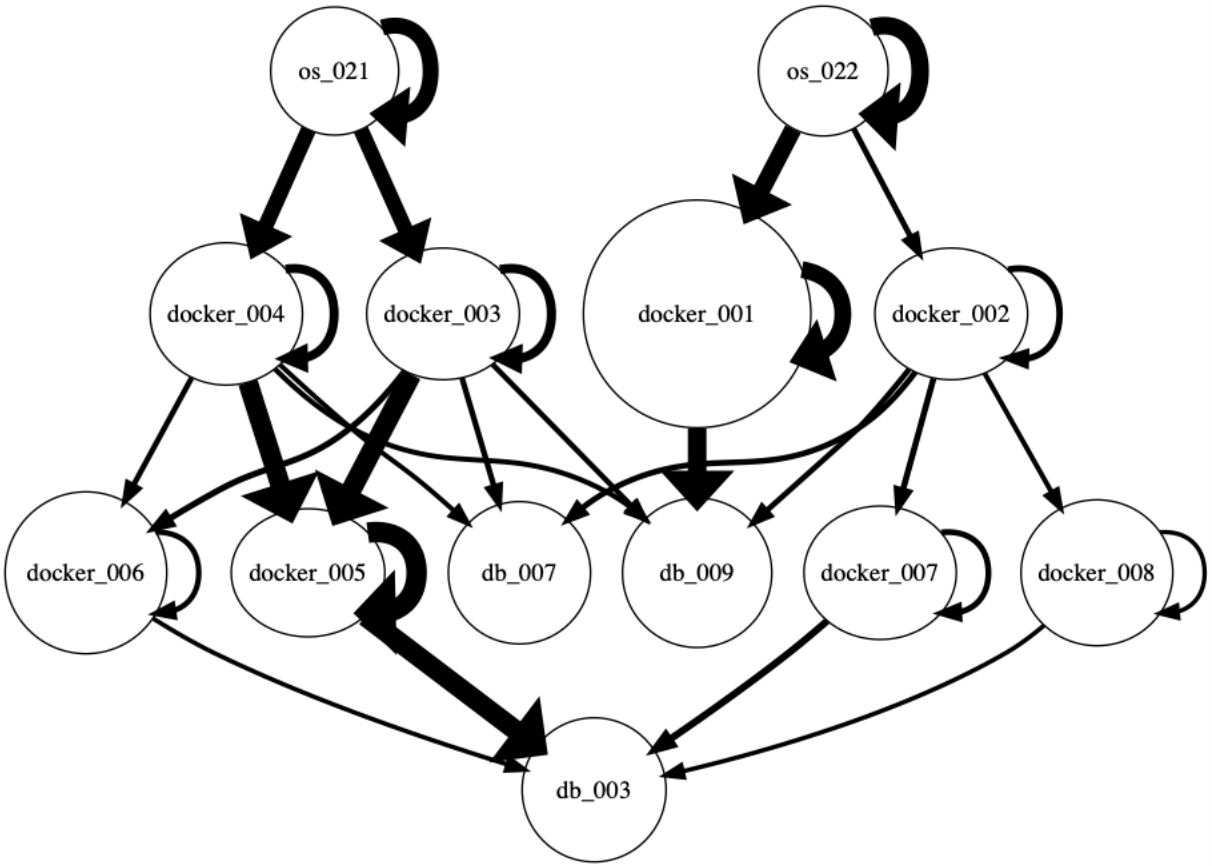
秒级累计平均延时统计分析



某调用链每分钟秒级累计平均延时

- 原始的调用时长序列并不是持续平稳序列，而是在以一分钟为周期，并且只存在于前40秒
- 因此当异常检测时，获取当前分钟的累计平均调用时间秒数，与前几分钟此时的平均调用时长快照做比较也就是异常检测即可

拓扑推断



当os_017发生故障时，其上的两个docker调用链特征已经足够明显，此时无需等待os_017的Sent_queue指标

主动探测、结果覆盖

- 主动循环探测
- 结果覆盖：
  - 缺少黄金业务指标的检测，导致系统故障检测会产生误报
  - 由于规则判断故障产生的**10分钟内的最后一次答案**为最终答案，因此存在误报覆盖正确答案的问题

- 解决方案：
  - 引入了一系列覆盖调整策略：
    - 故障优先级
    - 故障置信度
    - 故障报告次数
    - ……

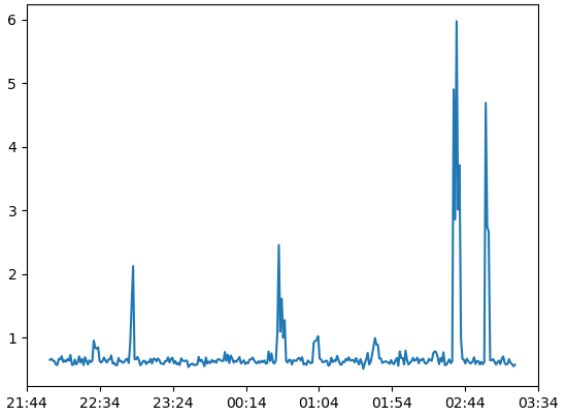
清華大學
Tsinghua University

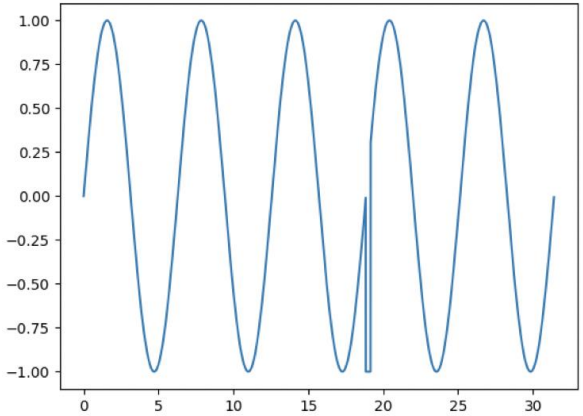
目录

- 团队介绍
- baseline方法
- 准确性提升
- 速度提升
- 可能的不足

未来可能的研究问题

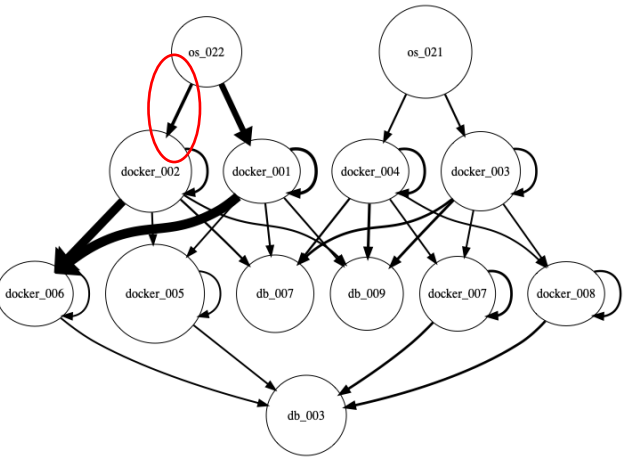
- KDE算法的局限性
  - 对于强周期性数据无可奈何

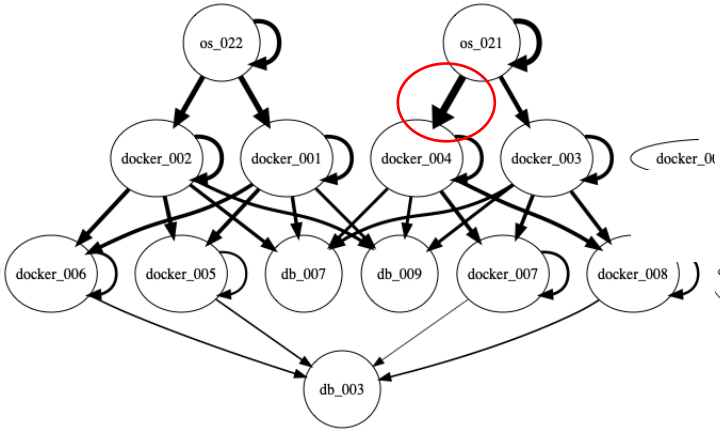


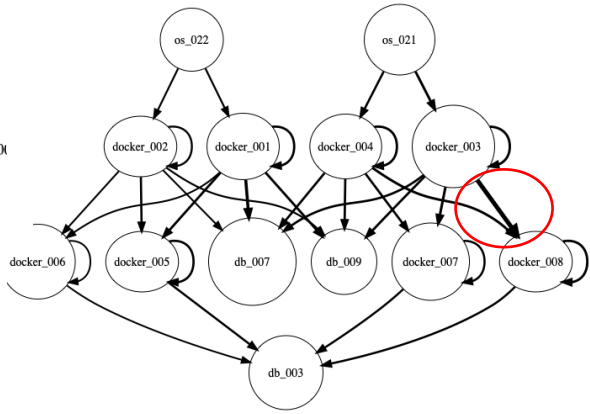


未来可能的研究问题

- 随机游走模型的鲁棒性
  - 尤其是当docker_001至docker_004网络故障的时候，一条边的异常分数误报就可能导致结果的误报

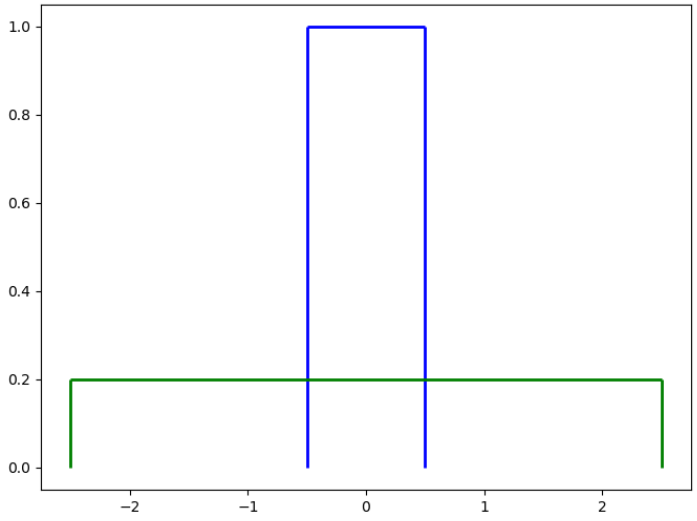


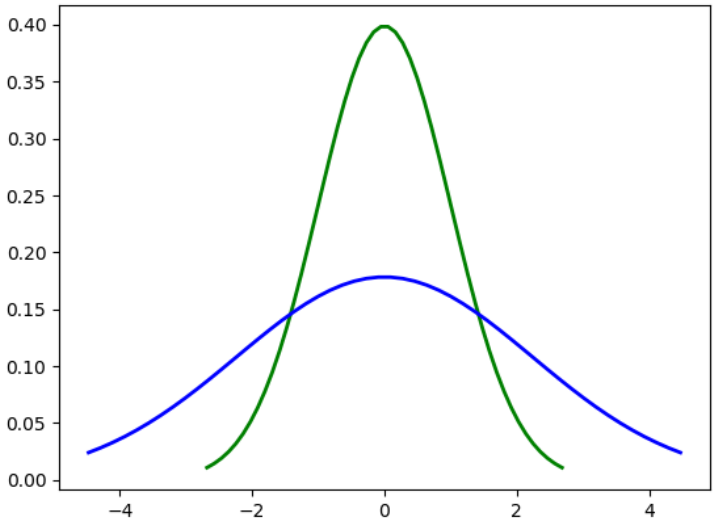




未来可能的研究问题

- 概率密度越低，就真的越异常嘛?





清華大學
Tsinghua University

谢谢！

清華大學
Tsinghua University

Q & A