AIOps Challenge

2023 CCF国际AIOps挑战赛决赛
暨“大模型时代的AIOps”研讨会

构建由大模型辅助的基于多模态数据融合的异常检测、根因诊断和故障报告生成系统

队伍：DDopS

机构：中山大学

演讲人：黄海宇

主办单位：中国计算机学会（CCF）、清华大学、中国建设银行股份有限公司、南开大学

承办单位：中国计算机学会互联网专委会、清华大学计算机科学与技术系、中国建设银行股份有限公司运营数据中心、南开大学软件学院、北京必示科技有限公司

赞助单位：华为技术有限公司、国网宁夏电力有限公司电力科学研究院、软通动力信息技术（集团）股份有限公司

AIOps Challenge

2023 CCF国际AIOps挑战赛决赛
暨“大模型时代的AIOps”研讨会

目录
CONTENTS

中山大學 SUN YAT-SEN UNIVERSITY

第一章节

团队介绍

1 团队介绍

AIOps Challenge

参赛人员：

- 黄海宇
- 何子龙
- 李晓芸
- 余广坝
- 李民
- 麦艮廷
- 谭苟
- 徐俊杰龙
- 张晓玉
- 张震宇
- 刘祺瀚

DDopS来自中山大学计算机学院Intelligent DDS 实验室。实验室主要方向为云计算、智能运维(AIOps)、软件定义网络、分布式软件资源管理与优化、eBPF 性能监控与优化等。多篇论文工作发表于国际会议如ICSE、ESEC/FSE、ASE、WWW、DSN等。实验室积极与阿里巴巴、华为、腾讯、蚂蚁金服、深信服等企业开展校企合作项目，并且将部分研究成果在企业落地。

中山大學 SUN YAT-SEN UNIVERSITY

第二章节

选题分析

2.1 选题分析 —— 背景与挑战

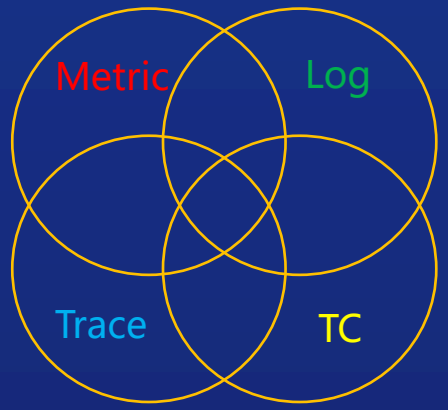
AIOps Challenge

竞赛数据洞察

挑战1：

多模态数据源：融合多模态可观测性数据是全场景、细粒度故障诊断的基石。，

此次比赛提供了四种模态的数据。囊括了传统的trace、log、metric三种模态数据，并提供了一种tc数据，表示一次交易。tc数据我们认为可以视为一种有特定含义(交易)的event去分析。



中文	时间戳	指标所在对象	指标名_类别	指标值	子部件
英文	timestamp	cmdb_id	kpi_name	value	device
样例	1685462400	Kafka_03	system.cpu.pct_usage	25.74	
样例	1685462400	Weblogic_01	system.disk.pct_usag	1.42	/home/

英文	中文	样例
sysId	物理子系统id	Sys001
traceId	traceId	c6ffcc1c59e607d099548b416cb90caf
cost	耗时 单位毫秒	0
kind	事务类型	CLIENT
bizCode	业务类型编码	
bizStatusCode	业务状态码	
attributes	基本属性	
attributesJson	基本属性json字符串	"net.peer.port": 6003, "db.statement": "SET
parentSpanId	父spanId	d06c508e6aa09881
spanId	spanId	75e1cc508e6cdd8f
CMDB_ID	CMDB_ID	Weblogic_09
appId	应用模块id	64701a8e58d7742d51c557f1
appName	应用模块名称	批量结果样
name	名称	SET
deployUnitName	部署单元名称	
startTime	链路开始时间（时间戳）	1687737600004
endTime	链路结束时间（时间戳）	1687737600004
statusCode	业务状态码（UNSET:应用成功 OK:应用成功 ERROR:应用失败）	UNSET
status	状态（1:应用成功 2:业务错误 3:应用错误）	1

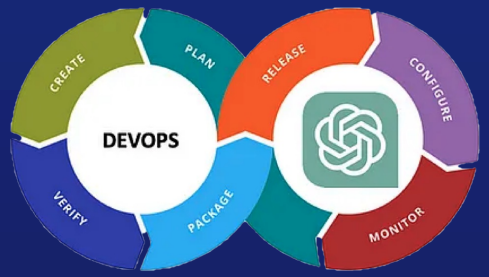
英文名	中文名
tran_code	交易码
timestamp	时间
amount	交易量
bus_success_rate	业务成功率
sys_success_rate	系统成功率
avg_rsp_time	平均响应时间
stall_amount	长交易数
avg_proc_time	平均处理时间
stall_rate	长交易率
apdex	性能指标

```
[PSYoungGen: 6784K->0K(634880K)] [ParOldGen: 626981K->332970K(1398272K)] 633765K->332970K(2033152K), [Metaspace: 211821K->211813K(1241088K)], 0.3833348 secs] [Times: user=0.88 sys=0.00, real=0.39 secs]
1694620833: 4344813.016: [GC (Allocation Failure) [PSYoungGen: 574464K->9248K(644096K)] 907434K->342218K(2042368K), 0.0084012 secs] [Times: user=0.04 sys=0.00, real=0.01 secs]
1694620881: 4344861.071: [GC (Allocation Failure) [PSYoungGen: 595488K->2592K(641024K)] 928458K->335562K(2039296K), 0.0082342 secs] [Times: user=0.03 sys=0.00, real=0.01 secs]
1694621001: 4344981.454: [GC (Allocation Failure) [PSYoungGen: 588832K->51584K(626176K)] 921802K->385505K(2024448K), 0.0114976 secs] [Times: user=0.04 sys=0.01, real=0.01 secs]
1694621173: 4345153.282: [GC (Allocation Failure) [PSYoungGen: 624512K->15168K(635904K)] 958433K->391768K(2034176K), 0.0120649 secs] [Times: user=0.05 sys=0.01, real=0.01 secs]
1694621348: 4345328.875: [GC (Allocation Failure) [PSYoungGen: 588096K->4640K(641536K)] 964696K->392970K(2039808K), 0.0099335 secs] [Times: user=0.03 sys=0.00, real=0.01 secs]
```

时代趋势

挑战2：

融合大模型：AI大模型是“大数据+大算力+强算法”结合的产物，凝聚了大数据内在精华的“隐式知识库”。构建运维大模型是智能化、通用化故障诊断的未来。



构建由大模型辅助的基于多模态数据融合的异常检测、根因诊断和故障报告生成系统

中山大學 SUN YAT-SEN UNIVERSITY

2.2 选题分析 —— 达到的运维能力

AIOps Challenge

构建由大模型辅助的基于多模态数据融合的异常检测、根因诊断和故障报告生成系统

本小组拟定的赛题及解决方案达到以下运维能力，力求在解决热点问题的同时做出创新，且贴合赛题场景

达到的运维能力：

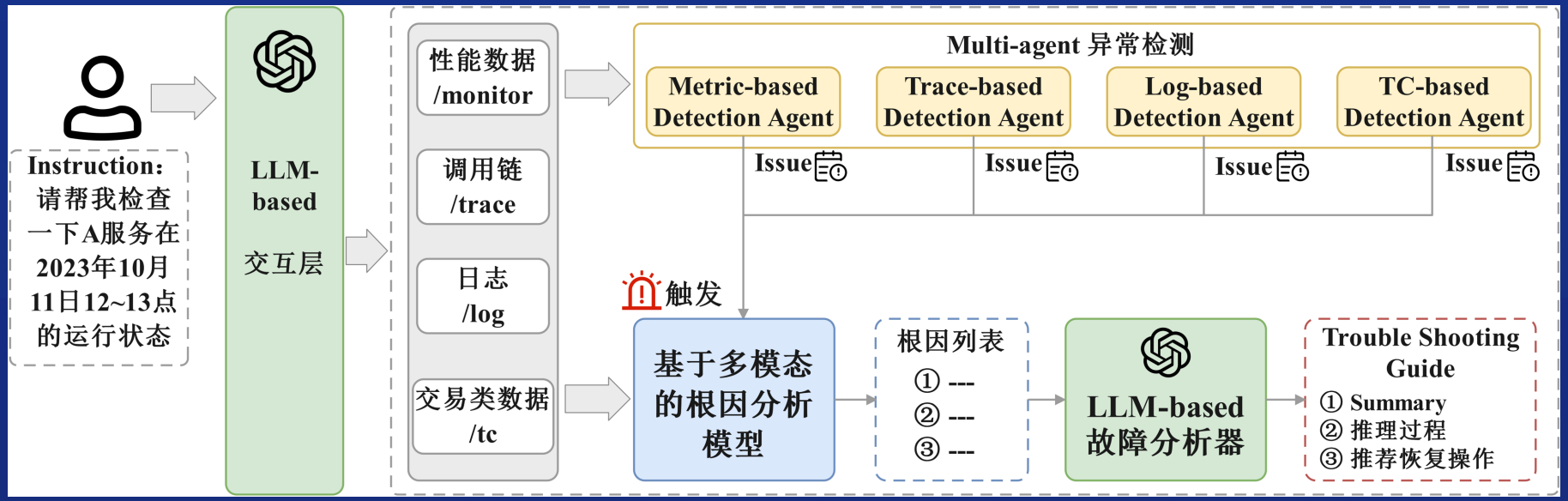
- **故障检测能力。**从多模态数据源中检测系统是否存在异常
- **故障分类能力。**在异常检测能力的基础上，分析出大致的异常类型。本次方案中能识别到的异常类型包括：耗时异常、流量异常(某事件触发次数增加)、业务逻辑异常(表现为断链)
- **根因定位能力。**即在众多异常中，找到问题根本原因
- **故障报告生成能力。**即根据分析结果生成故障报告和恢复建议
- **识别用户自然语言提问的能力。**用户可以使用自然语言进行提问，模型会理解用户语义并分析出用户给出的任务

中山大學 SUN YAT-SEN UNIVERSITY

第三章节

方案介绍

3.1 整体架构介绍



1. **LLM-based交互层**：此模块用于理解用户的query instruction，提取出用户给出的基础任务和参数。我们使用ChatGLM2作为base model，在源文件代码中加入**self-consistency 、CoT 和 in-context learning**的逻辑，使模型更能理解我们的场景并能更好地做出回答

2. **Multi-agent异常检测**：由于涉及多种模态的数据源，采用单一检测模块难以获得高准确率且容易产生假阳性，因此我们**采用multi-agent的检测方案**。我们针对trace、log、metric三种模态数据均设计了异常检测agent

3. **基于多模态数据融合的根因分析模型**：算法将调用链、日志和指标等**多模态数据转换成统一的事件表达**，利用无监督的频繁项集挖掘的方法找出故障模式，在资源和代码块级别定位细粒度根因。同时该方法还能通过对比故障前后的模式变化对故障进行**解释**

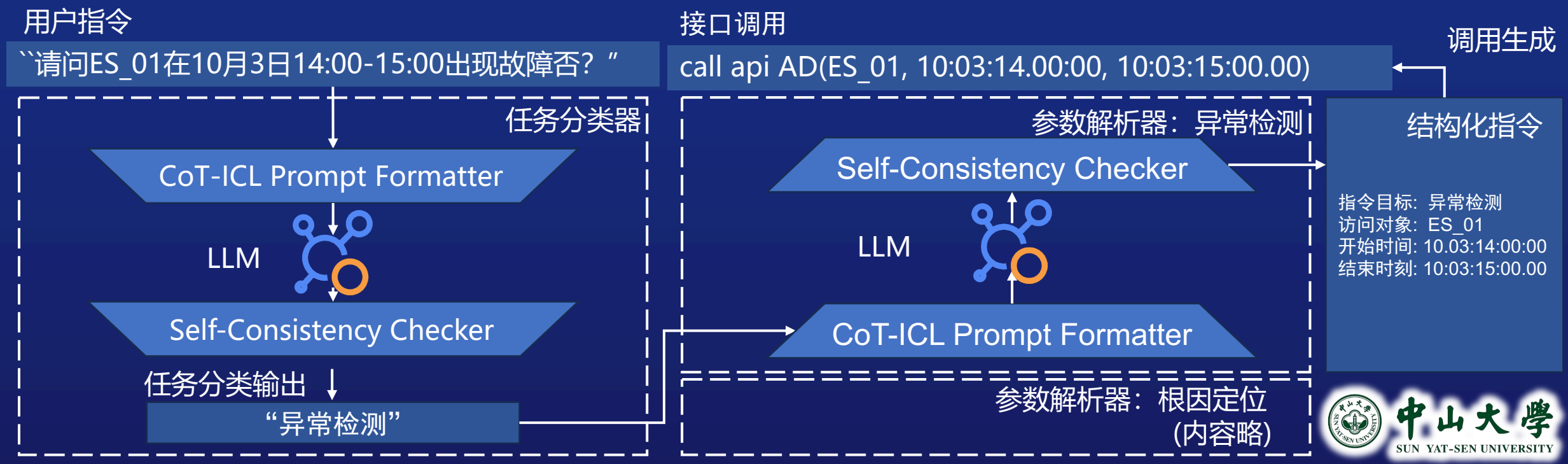
4. **LLM-based故障分析器**：采用**多LLM Agent轮询问答**的方式，不同的LLM Session作为不同角色，生成故障报告工单

3.2 LLM-based 交互层

在真实业务场景中，**并非每个云产品的用户都具备充分的运维领域的知识或专家经验。**为了让我们的多模态异常检测、根因定位端到端系统能够更好地服务于用户，我们使用基于LLM的交互层理解使用者的意图，提取用户指令中的参数，并最终生成调用异常检测或根因定位模块的API。该模块包含两步：

1. LLM-based 任务分类
2. LLM-based 参数解析

具体而言，LLM-based交互层将会首先基于包含上下文例的思维链提示词，区分用户的意图是指向哪一类任务。之后按照相关任务，对用户指令中出现的参数进行解析。解析后的指令将被组装成api call。流程图如下所示：



3.2 LLM-based 交互层

AIOps Challenge

考虑到没有相关的benchmark进行评估，我们在确保指令质量和多样性的前提下，手动设计并打标了50个测试数据进行评估。结果显示我们在分类任务上达到**92%**的准确率，在解析任务上达到了**90%**的准确率。

我们认为这种较好的表现是因为LLM本身具有较强的语义理解能力，也因为分类任务和解析任务都不涉及复杂的推理过程：分类任务实质是一个二分类任务，而解析任务实质是文本摘要提取任务。我们通过majority vote的方式进行自我集成，即self-consistency，进而更加避免了错误结果的产生。

在工业部署上，我们的方案具有额外的优势，即**无需模型白盒权限**，也**无需设计梯度计算**。这保证在实际应用中，LLM无需负担除推理之外的其他计算开销。

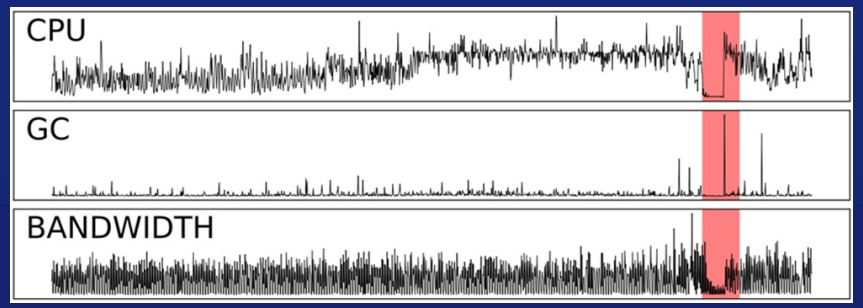
```
(py10) root@VM-2-145-kylin:/dev/data/dev>python3 simple_chat.py
Loading checkpoint shards: 100%|████████████████████████████████████████████████████████████| 7/7 [00:09<00:00,  1.38s/it
Instruction: 请问12月1日12：00到13：00之间，系统服务是否出现故障？
Response:
Sample#1:您提供的查询包括以下信息：
 - 组件名称："System服务"
 - 开始时间："12月1日12:00"
 - 结束时间："12月1日13:00"
根据提供的信息，System服务在12月1日12:00到13:00之间出现了故障。
Sample#2:从您提供的指令中，我们可以分析出以下信息：
 - 组件名称："NULL"
 - 开始时间："12月1日12:00"
 - 结束时间："12月1日13:00"
您的指令没有提供组件名称或系统的标识，因此组件名称被设置为"NULL"。
Sample#3:抱歉，您提供的查询时间范围是错误的。您询问的是12月1日12:00到13:00之间系统服务是否出现故障，但您的指令中给出的时间范围是10月20日内。

如果您想查询12月1日12:00到13:00之间系统服务是否出现故障，请提供以下信息：
 - 组件名称："系统服务"
 - 开始时间："12月1日12:00"
 - 结束时间："12月1日13:00"
Sample#4:从您提供的指令中，我们可以分析出以下信息：
 - 组件名称："系统服务"
 - 开始时间："12月1日12:00"
 - 结束时间："12月1日13:00"

根据您提供的信息，系统服务在12月1日12:00到13:00之间出现了故障。
Sample#5:从您提供的指令中，我们可以分析出以下信息：
 - 组件名称："NULL"
 - 开始时间："12月1日12:00"
 - 结束时间："12月1日13:00"
您的指令没有提供组件名称或系统的标识，因此组件名称被设置为"NULL"。
```

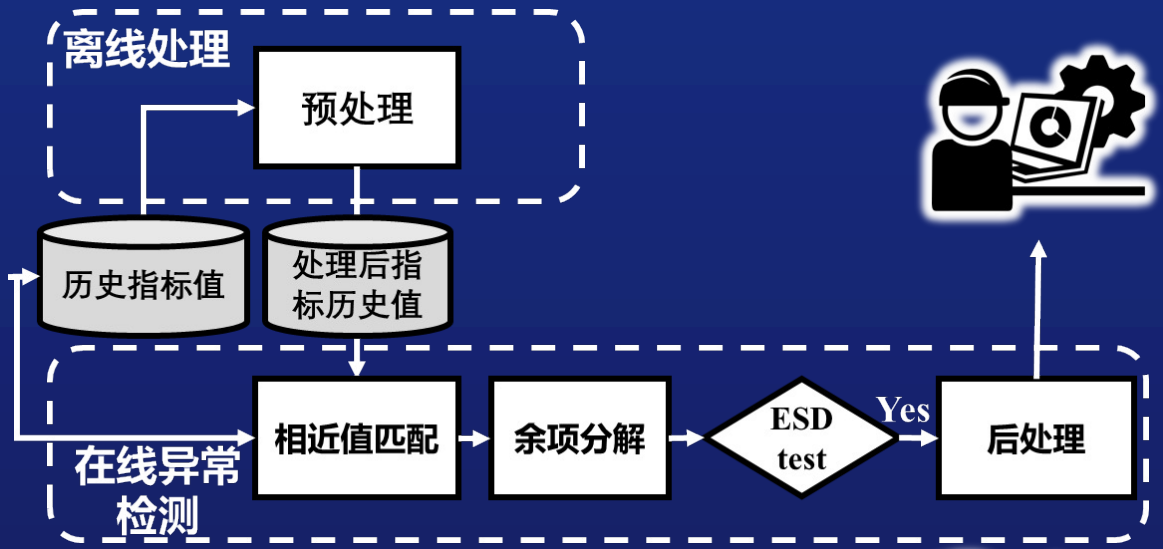
中山大學 SUN YAT-SEN UNIVERSITY

3.3.1 基于统计的指标异常检测

AIOps Challenge

- 指标为反映系统运行时状态的时间序列数据，通常包含的信息最为全面
- 洞察：运维人员通常将前几天的指标与当前指标放在一起，观察同环比等判断异常
- 对某个指标值X_t，假设其历史相近时间段值集合为H_t
- 余项分解：$R_t = \min(\{abs(X_t - h) \mid h \in H_t\})$
- ESD检验：$Z_t = \frac{|R_t - \text{median}(R_{t-W:t})|}{MAD(R_{t-W:t})}$





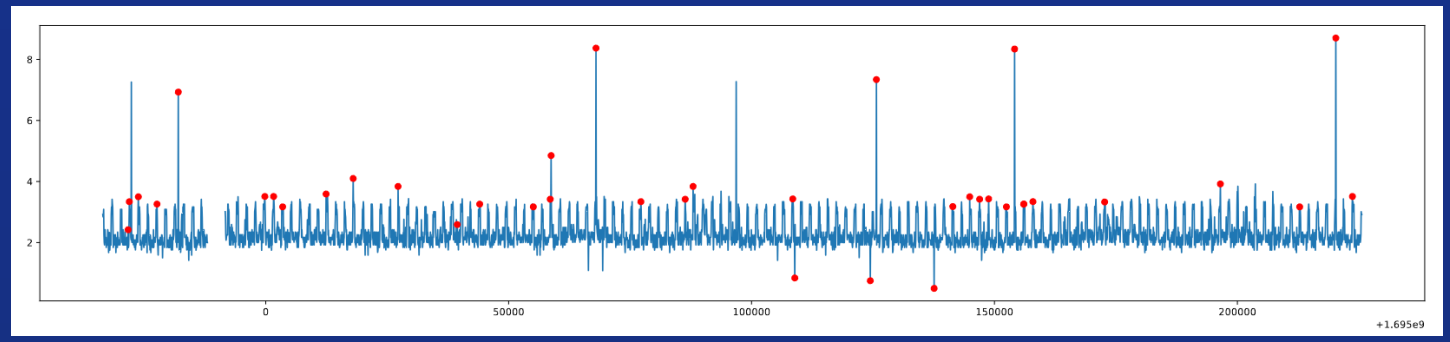
3.3.1 基于统计的指标异常检测

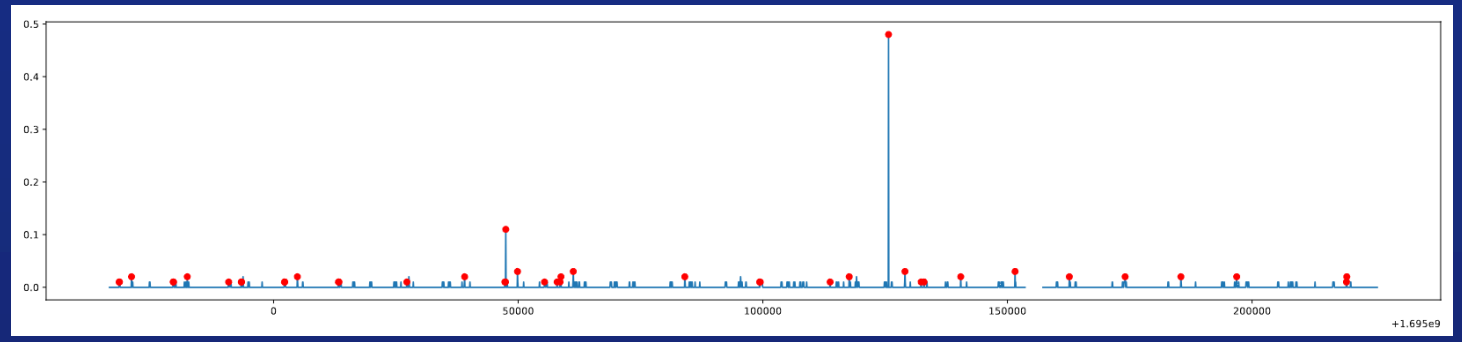
AIOps Challenge

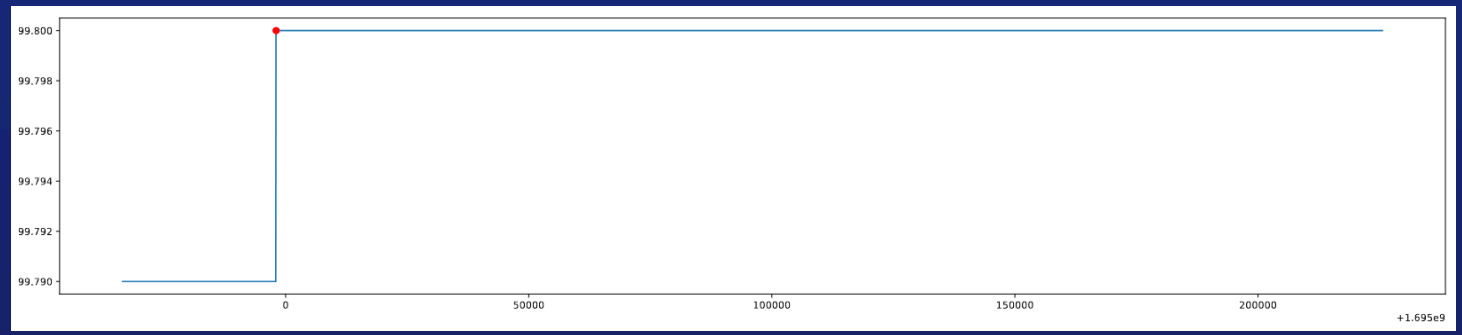
- 返回结果例子：

```
[{'timestamp': 1694971680,
  'cmdb_id': 'ES_01',
  'kpi_name': 'system.cpu.i_dle',
  'device': 'NaN',
  'value': 97.58},
 {'timestamp': 1694971920,
  'cmdb_id': 'ES_01',
  'kpi_name': 'system.cpu.i_dle',
  'device': 'NaN',
  'value': 96.66},
 {'timestamp': 1694973780,
  'cmdb_id': 'ES_01',
  'kpi_name': 'system.cpu.i_dle',
  'device': 'NaN',
  'value': 96.5},
```

- 检测效果如右：







中山大學 SUN YAT-SEN UNIVERSITY

3.3.2 基于神经网络的指标异常检测

AIOps Challenge

难点 传统方法在处理复杂系统的非线性关系和复杂异常模式时存在局限性。

应对 引入神经网络的指标异常检测模块，采用LTSF-Linear算法

效果 克服了传统方法在处理复杂系统中的限制，弥补了传统方法的不足。

离线学习

正常模式数据

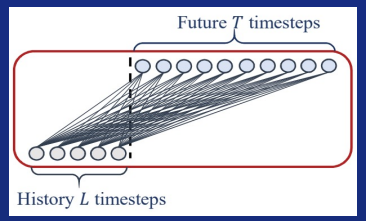
数据格式化

单指标数据

数据分解

移动平均分量
季节趋势分量

LTSF-Linear模型构建



训练学习

估计正态分布参数
均值和标准差

在线检测

异常模式数据

异常检测

正态分布生存函数
计算p-values

输出异常

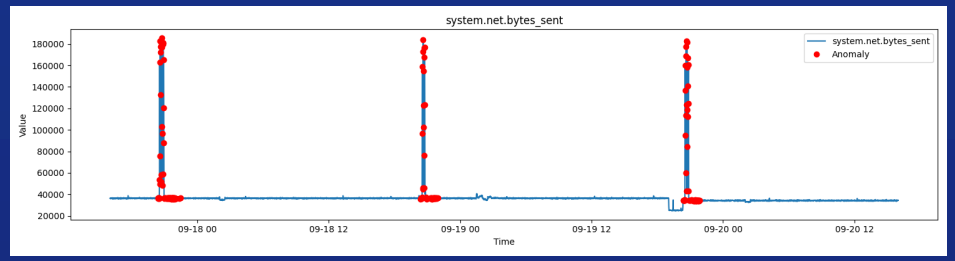
显著性水平
异常点的索引

中山大學 SUN YAT-SEN UNIVERSITY

3.3.2 基于神经网络的指标异常检测

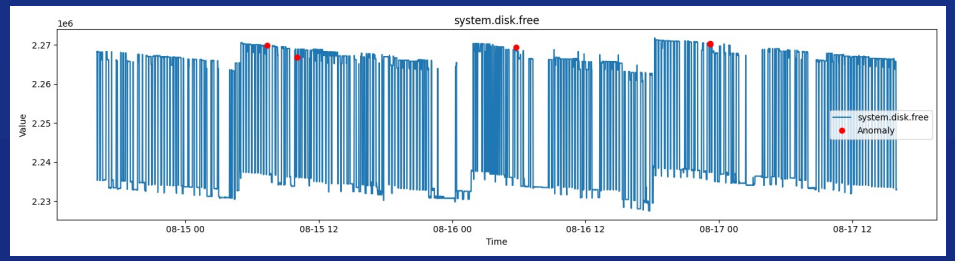
AIOps Challenge

根据 'cmdb_id'、'kpi_name'、'device' 这三个属性对数据进行分组。其目的是将异常数据按照具体的业务和设备进行划分，以便更精细地分析每个指标的异常情况。接着使用LTSF-Liner异常检测方法进行实验。



对异常数据进行检测结果例子：('Kafka_03', 'system.net.bytes_sent', 'eth0')

相对较多的红色点，这表明该方法能够较为准确地识别异常模式。



对正常数据进行检测结果例子：('Weblogic_13', 'system.disk.free', '/home').

红色点只分布在极少的区域。这说明误检率低，从而为异常检测提供了可靠的性能。

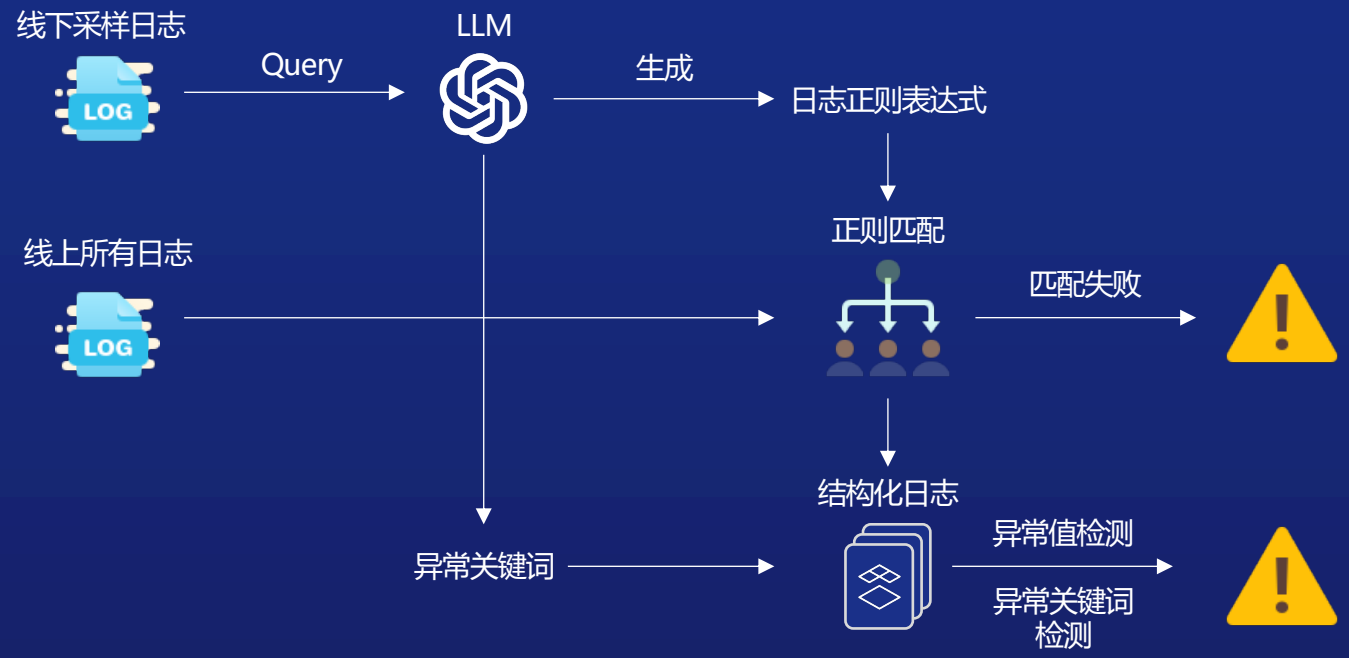
LTSF-Liner能够有效地区分正常模式和异常模式。

中山大學 SUN YAT-SEN UNIVERSITY

3.3.3 日志异常检测

AIOps Challenge

- 日志记录了系统在运行时发生的事件和系统状态，给运维人员提供细粒度的故障信息
- 目前日志数量较大，格式简单->采用高性价比的处理策略，仅进行日志结构化和异常值检测，使用LLM做类日志解析
- 异常检测策略：异常格式日志或异常日志检测




```
[Prompt:]Please write regular expressions to match the fol-
lowing logs:
1694873301 POST /coupon/jym=A0001 200 0.002
1694873301 POST /coupon/jym=A0010 200 0.026
1694873302 POST /coupon/jym=A0005 200 0.011

[Output:] To match the provided logs, regular expressions can
be used to capture the different sections of each log entry.
Below are regular expressions that can be used to match and
extract the different components of the given log lines:
Regular expression pattern to match the log entries:
^(\d+)\s+(\w+)\s+([^\s]+)\s+(\d+)\s+([0-9.]+)$
```

LLM 交互例子

3.3.3 日志异常检测

AIOps Challenge

- 运行结果

线下运行

线上异常检测

结果统计

日志告警

```
-rw-r--r--  1 leexy  staff   230K Nov  7 11:35 visualize.ipynb
(base) leexy@lixiaoyuns-MBP ~/workspace/projects/aiops2023/src bash run.sh
2023-11-19 21:41:49,411 - INFO - Filtering log file logad.log based on provided fault-indicating keywords.
2023-11-19 21:41:49,412 - INFO - ================================Offline Training================================
2023-11-19 21:41:49,412 - INFO - Start offline training
2023-11-19 21:41:49,412 - INFO - Processing log file Weblogic_18_access.log_20230815_28801367.20230816090002310
2023-11-19 21:41:49,537 - INFO - Processing log file Weblogic_23_access.log_20230815_28801325.20230816090002254
2023-11-19 21:41:49,673 - INFO - Processing log file Weblogic_18_access.log_20230815_28801357.20230816090002066
```

```
2023-11-19 21:42:06,520 - INFO - Processing log file Weblogic_06_access.log_20230815_28801293.20230816090001417
2023-11-19 21:42:06,647 - INFO - ================================Offline Training Finished================================
2023-11-19 21:42:06,655 - INFO - ================================Online Run================================
2023-11-19 21:42:06,656 - INFO - Processing log file Weblogic_27_access.log_20230918_66401601.20230919090001731
2023-11-19 21:42:06,780 - INFO - Anomalies found in file Weblogic_27_access.log_20230918_66401601.20230919090001731: 0
2023-11-19 21:42:06,780 - INFO - Processing log file Weblogic_20_access.log_20230918_66401433.20230919090002033
2023-11-19 21:42:06,910 - INFO - Anomalies found in file Weblogic_20_access.log_20230918_66401433.20230919090002033: 0
2023-11-19 21:42:06,910 - INFO - Processing log file Weblogic_10_access.log_20230918_66401801.20230919090001634
2023-11-19 21:42:07,044 - INFO - Anomalies found in file Weblogic_10_access.log_20230918_66401801.20230919090001634: 45
```

```
2023-11-19 21:42:23,410 - INFO - Anomalies found in file Weblogic_34_gc_mSrv1.log_20230918_66401378.20230919090001995: 15
2023-11-19 21:42:23,410 - INFO - ================================Online Run Finished================================
2023-11-19 21:42:23,410 - INFO - Total log file parsed: 145
2023-11-19 21:42:23,410 - INFO - Total anomalies found: 2030
2023-11-19 21:42:23,413 - INFO - Result saved to log.pkl
```

```
{'timestamp': '1694883602',
 'log_msg': '1694883602: 278106.599: [Full GC (Heap Inspection Initiated GC) [PSYoungGen: 1024K->0K(693248K)]
 [ParOldGen: 234152K->149477K(1398272K)] 235176K->149477K(2091520K), [Metaspace: 212734K->212726K(1241088K)], 0.
 3204906 secs] [Times: user=0.65 sys=0.00, real=0.32 secs]',
 'cmdb_id': 'Weblogic_22',
 'app_id': 'gc_mSrv1',
 'sys_id': '20230918'},
```

中山大學 SUN YAT-SEN UNIVERSITY

3.3.4 调用链异常检测

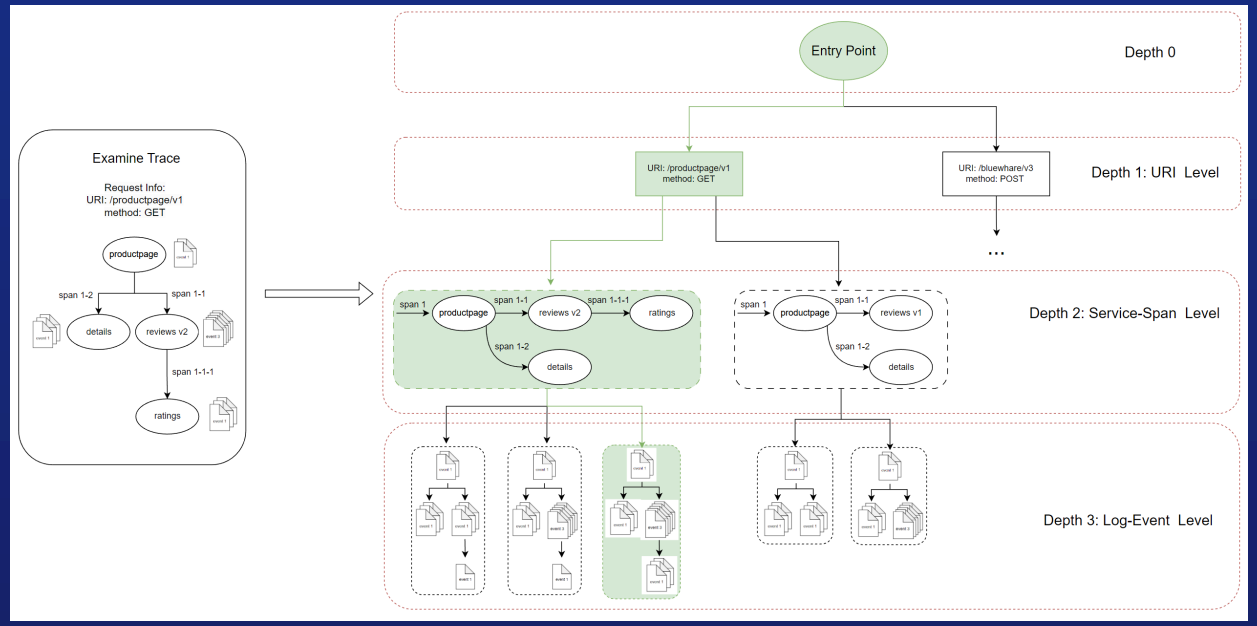
AIOps Challenge

采样 & 预处理：

- **调用链拼接 & 保证trace完整性**：首先根据trace_id和span父子关系将trace拼成完整的树形结构。同时对不完整的trace，我们予以丢弃，因为不完整的trace基本不具有分析价值且容易污染数据
- **粗粒度的采样**。由于全量分析trace时空复杂度太大，我们对同一入口的traces，只取耗时P75以上的部分及状态码异常的部分进行分析，这使我们能大大提高分析效率

离线建立正常模式，在线匹配检测：

- 对正常时间窗口的调用链进行汇聚，并构建多层正常模板
- 对于一条待测调用链，首先匹配与之最相近的匹配模板
- 分析待测链与模板链的结构、信息差异，产生不同种类的调用链异常类型



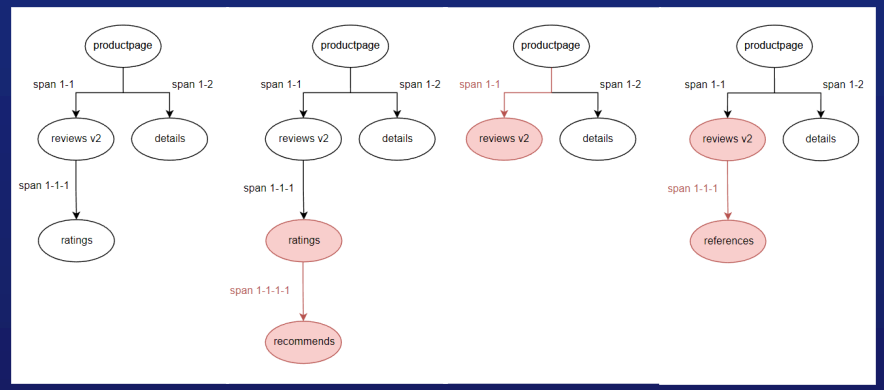
中山大學 SUN YAT-SEN UNIVERSITY

3.3.4 调用链异常检测

AIOps Challenge

异常类型识别结果：


```
{
    "summary": "接口/coupon/txCtrl的rpc方法rpc调用耗时加剧",
    "rank": 3,
    "trace_id": "trace_f5fe9530f01b1c614cd057cd92e6869b",
    "event_identity": "3a41df507be7a6157905a500b4532745_bc3ce9e4bd
    "finish_timestamp": 1695020083.0,
    "service_id": "Weblogic_19,AppId008,Sys001",
    "call_freq": 4,
    "time_offset": 277.43,
    "detail": {
        "service": "Weblogic_19,AppId008,Sys001",
        "url": "/coupon/txCtrl",
        "type": "rpc",
        "method": "rpc",
        "event_span_nodes": [
            "root",
            "Weblogic_19,AppId008,Sys001"
        ],
        "reason_type": "self",
        "finish_timestamp": 1695020083.0,
        "reason": "调用耗时971ms超过正常上限693.57ms",
        "current_argument": "/coupon/txCtrl",
        "base_argument": "/coupon/txCtrl"
    }
}
```

检测得到详细的异常issue单：

- 异常所处调用链
- 异常所处位置
- 详细异常信息
- 对应的正常模式


```
{
    "summary": "接口/coupon/txCtrl的rpc方法rpc调用耗时加剧",
    "detail": "{\"service\": \"Weblogic_19,AppId008,Sys001\", \"uri\": \"/coupon/txCtrl\"
    "score": 100,
    "score_on_trace": 100,
    "trace_id": "trace_d53637fc84a89c04c40bdc030efe5d91",
    "service_flag": "/coupon/txCtrl",
    "service_id": "Weblogic_19,AppId008,Sys001",
    "param_type": "time",
    "examined_lines": "[0, 6, 8, 10]",
    "sub_idx": 0,
    "time_offset": 277.43,
    "score_SBFL_index": 86,
    "span_id": "1",
    "span_event_id": "1",
    "event_identity": "3a41df507be7a6157905a500b4532745_bc3ce9e4bdf53ffb",
    "matched_template_index": 5,
    "apm_line": 1.0,
    "finish_timestamp": 1695020083.0,
    "score_SBFL": 154.79,
    "trans_poss": 8600
},
```

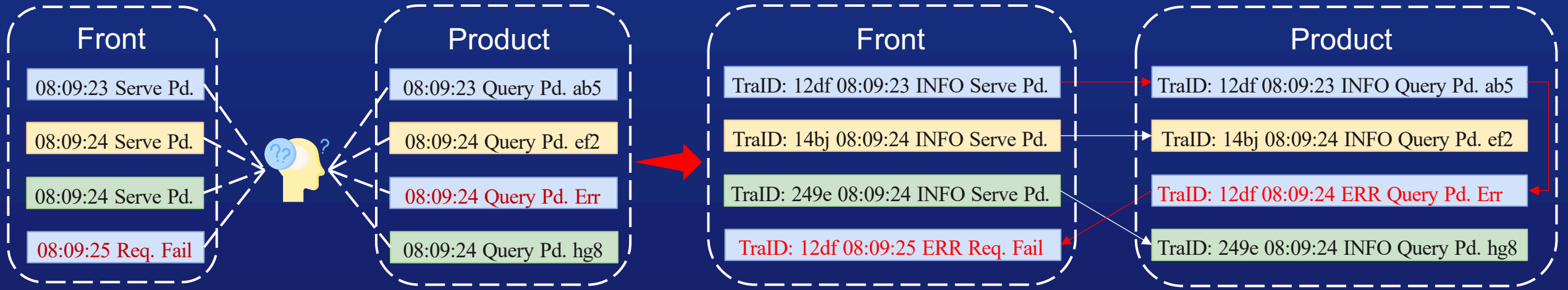

中山大學 SUN YAT-SEN UNIVERSITY

3.4 基于多模态数据融合的根因诊断

AIOps Challenge

动机：通过整合异构的 Metric、Log 和 Trace 来增强根因诊断

- 将 Trace ID 插入 Log 中以关联 Log 与 Trace
- 使用统一的 Event 表征克服多模态数据的异构性
- 基于 Trace ID 和时间戳关联跨服务和主机的事件表征



将日志和Trace集成之前

将日志和Trace集成之后

中山大學 SUN YAT-SEN UNIVERSITY

3.4 基于多模态数据融合的根因诊断

AIOps Challenge

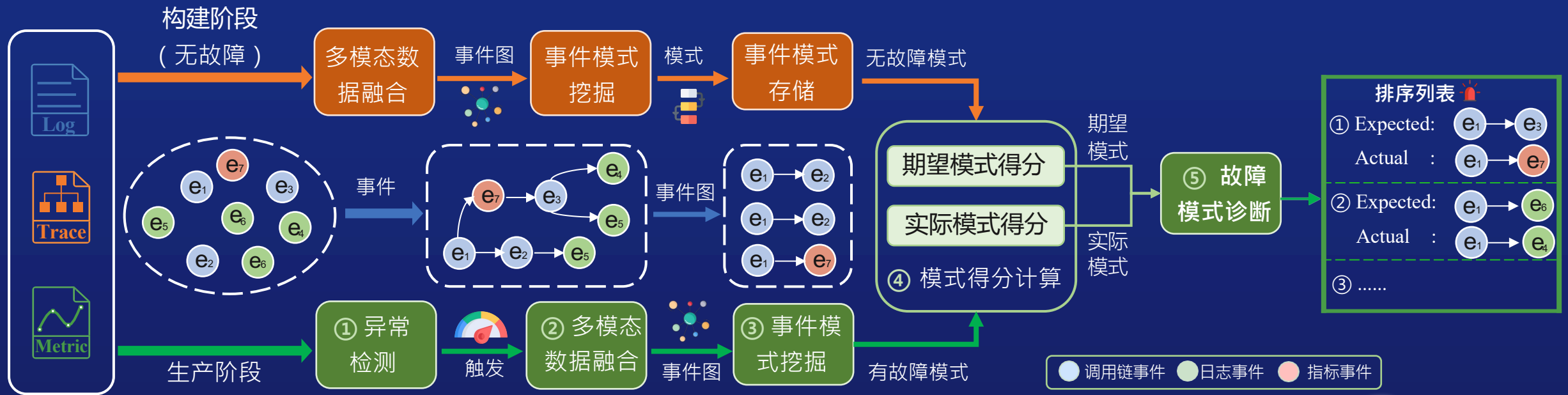
方法：基于多模态数据融合的无监督根因诊断方法研究

① 由上游异常检测模块触发

② 将多模态数据统一表征为事件，并整合到事件图中

③ 分别从 Fault-free 和 Fault-suffering 的事件图中挖掘频繁模式

④ 通过对 Fault-free 和 Fault-suffering 模式的差异计算可疑得分并排名

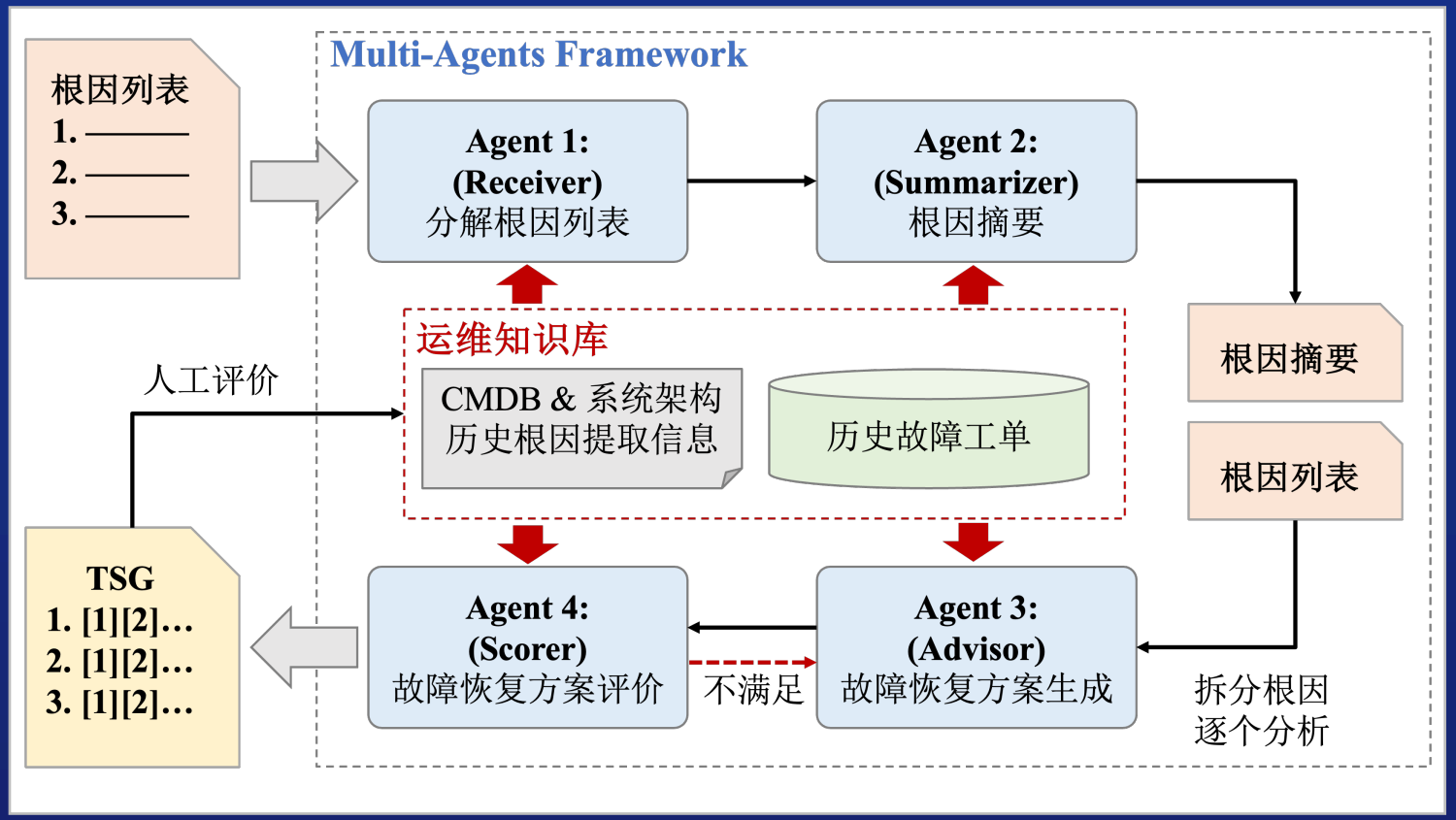


中山大學 SUN YAT-SEN UNIVERSITY

3.5 LLM-based 故障报告生成

AIOps Challenge

- Agent 1(Receiver)：接收上游输入的根因列表(结构化数据，机器可读)，并对其拆解成多个自然语言描述的根因
- Agent 2(Summarizer)：根据上游产生的多个自然语言描述根因，汇聚形成根因摘要
- Agent 3(Advisor)：对每个根因单独分析，生成故障恢复方案
- Agent 4(Scorer)：对方案进行评价审查

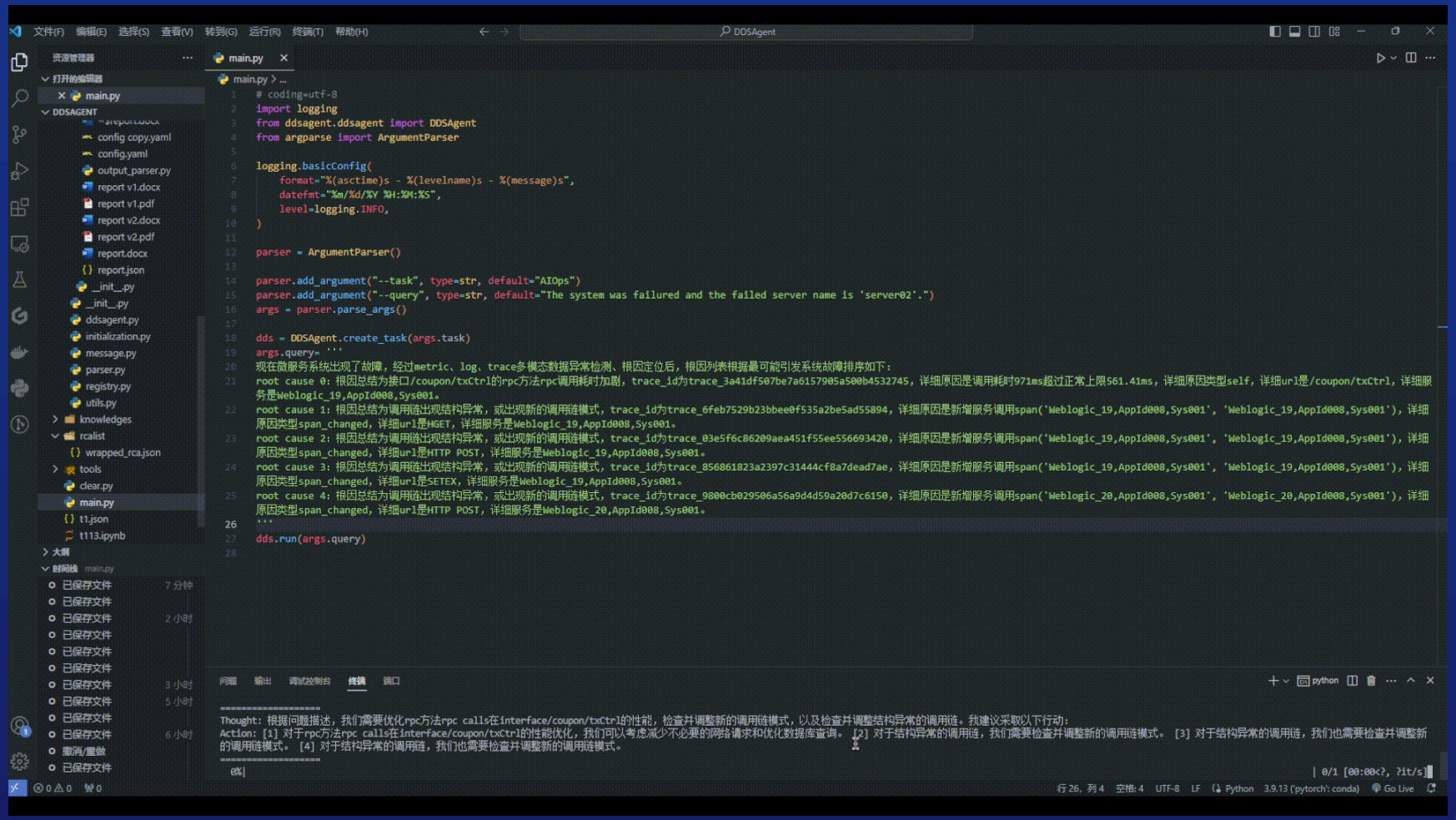


中山大學 SUN YAT-SEN UNIVERSITY

3.5 LLM-based 故障报告生成

AIOps Challenge

演示样例：Multi-Agent程序的运行过程和结果



中山大學 SUN YAT-SEN UNIVERSITY

3.5 LLM-based 故障报告生成

AIOps Challenge

演示样例：自动运维报告单

AI Agents 自动运维报告单

故障问题清单

- **故障报告 ID**: 5230c984-7e16-11ee-a836-965a262bc0e6
- **提交时间**: 2023-11-18 15:45:39.627260
- **问题描述**: 对以下故障根因进行分析并给出恢复建议

疑似根因 1

- **故障问题**: 接口/coupon/txCtrl 的 rpc 方法 rpc 调用耗时加剧
- **排序**: 1
- **trace_id**: trace_3a41df507be7a6157905a500b4532745
- **问题说明**: 调用耗时 971ms 超过正常上限 561.41ms

疑似根因 2

- **故障问题**: 调用链出现结构异常，或出现新的调用链模式
- **排序**: 2
- **trace_id**: trace_6feb7529b23bbee0f535a2be5ad55894
- **问题说明**: 新增服务调用 span('Weblogic_19,AppId008,Sys001', 'Weblogic_19,AppId008,Sys001')

疑似根因 3

- **故障问题**: 调用链出现结构异常，或出现新的调用链模式
- **排序**: 3
- **trace_id**: trace_06f9aa5101e0a2657fadcd9117e44af9
- **问题说明**: 缺失服务调用 span('Weblogic_21,AppId008,Sys001', 'Weblogic_21,AppId008,Sys001')

根因摘要

- **解析结果**: 根据历史故障信息，这些故障同时发生时最大可能的根因是

1. 接口/coupon/txCtrl 的 rpc 方法 rpc 调用耗时加剧，可能是由于服务端处理能力不足或者网络延迟导致的。这可能导致调用链中的其他服务受到影响，从而出现结构异常或新的调用链模式。
2. 调用链出现结构异常或新的调用链模式，可能是由于服务间的依赖关系发生变化或者某个服务出现了问题。新增的服务调用 span('Weblogic_19,AppId008,Sys001', 'Weblogic_19,AppId008,Sys001')可能是为了解决某个问题而引入的新的逻辑。
3. 调用链出现结构异常或新的调用链模式，缺失服务调用 span('Weblogic_21,AppId008,Sys001', 'Weblogic_21,AppId008,Sys001')可能是由

恢复方案

- **故障恢复清单**: 每个故障根因的恢复建议如下

疑似根因 1

- **故障问题**: 调用链出现结构异常，或出现新的调用链模式
- **恢复方案**:
  1. 检查服务端的代码，确保逻辑正确；
  2. 检查服务的配置文件，确保配置正确；
  3. 如果可能的话，可以考虑使用调试工具或者日志来定位问题所在。

疑似根因 2

- **故障问题**: 接口/coupon/txCtrl 的 rpc 方法 rpc 调用耗时加剧
- **恢复方案**:
  1. 检查网络连接，确保客户端和服务端之间的通信畅通；
  2. 优化服务端的代码，提高处理速度；
  3. 如果可能的话，可以考虑使用负载均衡或者分布式架构来分散请求压力。

疑似根因 3

- **故障问题**: 调用链出现结构异常，或出现新的调用链模式
- **恢复方案**:
  1. 检查服务端的代码，确保逻辑正确；
  2. 检查服务的配置文件，确保配置正确；

2023-11-19 14:41:23 Sunday November

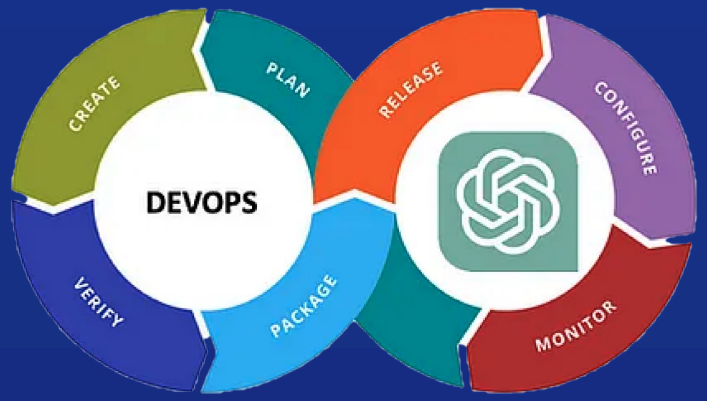
第四章节

总结与思考

4 总结与思考

AIOps Challenge

LLM 在 Ops 中的应用：端到端 or 垂向应用?



- 在我们的解决方案中，我们并非直接仅仅使用 LLM 去搭建一个完整的端到端系统，比如直接训练一个集成大模型作为一个整体解决方案。而是采用大小模型结合的思路，让LLM做一些不太涉及具体系统本身，但是需要按人类思维分析的任务，替代以往需要人工的部分（例如，理解语义、总结、建议等工作），垂向的领域问题交给对口小模型。这样更能充分利用 LLM 的 general 知识，也更具有实用性，减轻因为 LLM 的幻觉对分析造成的威胁。

LLM 在 Ops 中的挑战:

- 冷启动：一开始 pretrained model 并没有对所应用的特定场景的知识，很可能无法正确作答，比如根本不认识问题中的指标名、物理节点名等，因为这些并不在 LLM 用于预训练的知识里
- 微服务系统快速迭代：生产微服务系统是一个高速迭代和不断变化的环境，供 LLM 训练的知识可能很快就会过时，之前的经验很可能反而成为噪声

中山大學 SUN YAT-SEN UNIVERSITY

2023 CCF国际AIOps挑战赛决赛暨“大模型时代的AIOps”研讨会

THANKS