



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

LLM和Agent技术在AIOps领域的应用探索

胡牛犇 中原银行

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

目录

CONTENTS

选题背景

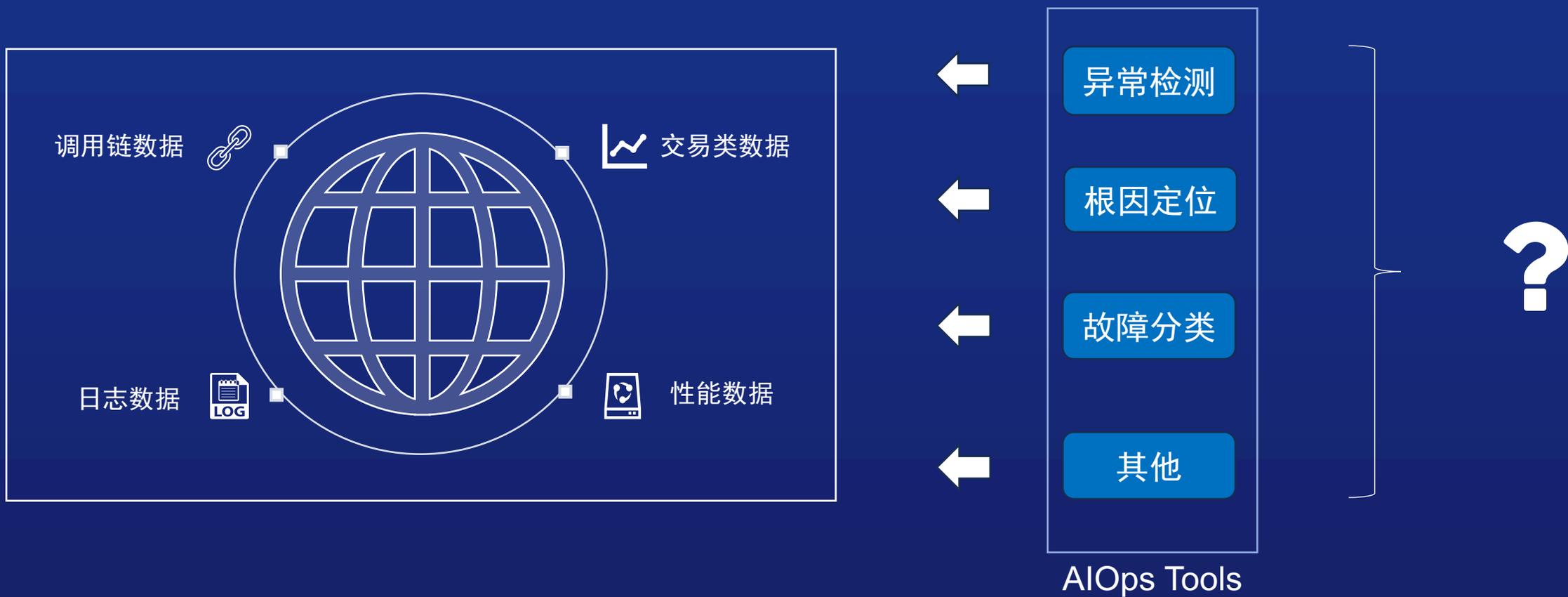
方案设计

方案效果

未来探索

选题背景

问题背景：运维领域有大量多维度的监控数据，同时随着AIOps的发展也积累了众多AIOps工具，但是工具使用门槛较高，往往需要编写代码，而且不同的场景使用何种工具都需要硬编码调度，这些都大幅降低了运维的效率。



选题背景

解决思路：将LLM智能体和AIOps工具有机融合，模拟运维专家进行任务规划和决策，同时主动使用AIOps工具，降低运维门槛，提升运维效率。

$$\text{Agent} = \text{LLM} + \text{Memory} + \text{Planning} + \text{Tools}$$

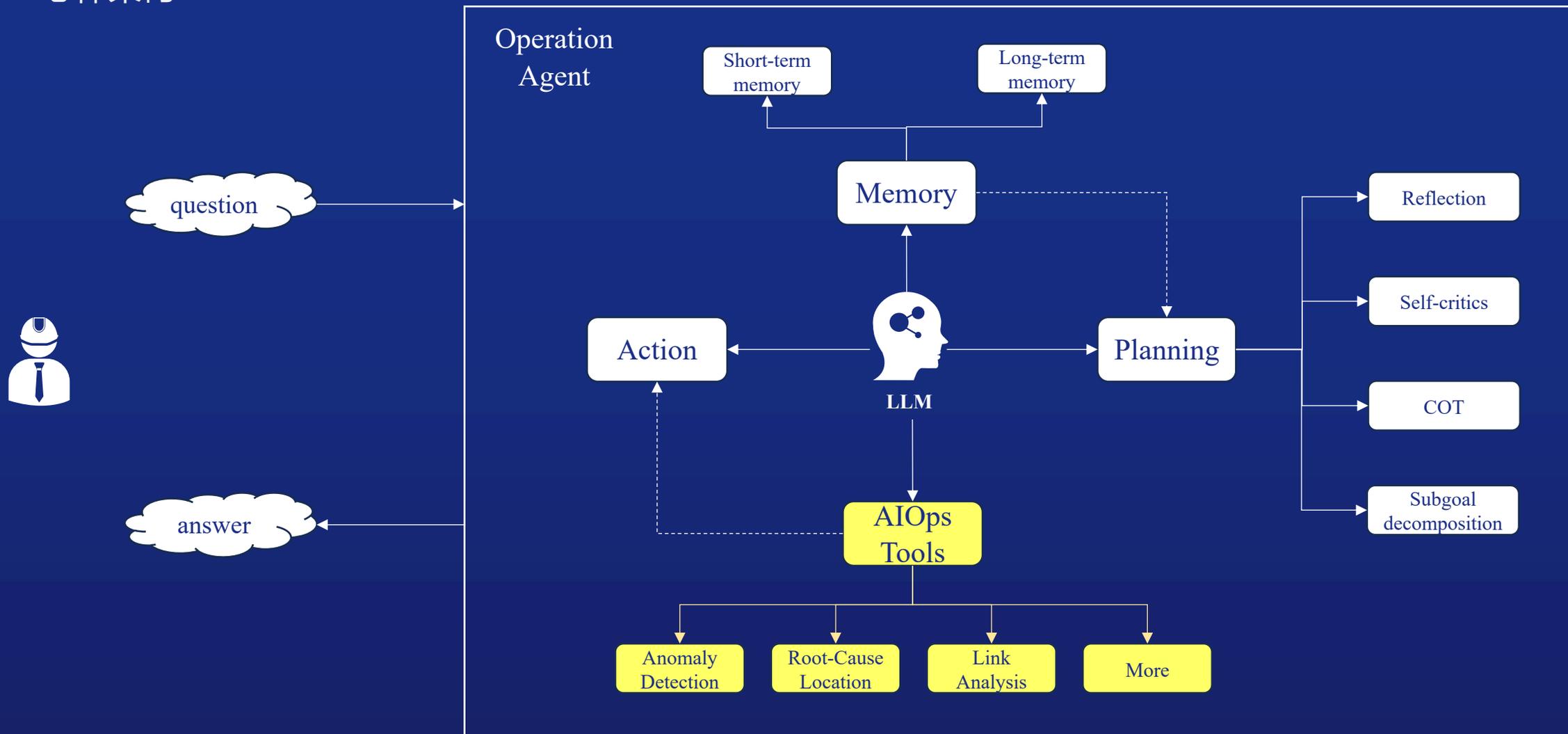
---- OpenAI



$$\text{Operation Agent} = \text{LLM} + \text{Memory} + \text{Planning} + \text{AIOps tools}$$

参赛运维能力	是否主要
AIOps与LLM结合能力	是
可视化能力	否

总体架构



可行性分析

创新性



1. 创造运维Agent自动调用AIOps工具解决运维问题
2. 对话形式，用户0代码参与
3. 可交互对话窗口
4. 框架易搭建、易扩展

挑战性



1. 开源大模型能否准确识别运维工具?
2. 运维工具是否能够快速得出结果?
3. 是否有GPU? 能否推理加速?
4. 前端是否支持

可行性



1. Qwen-LLM工具使用能力满足要求
2. 行内自研数项AIOps能力均达到秒级响应
3. 可用GPU，可采用量化模型和投机解码策略
4. 支持Windows主机，可进行可视化展示

得益于赛题资源和行内实践，为达到整体目标提供支持

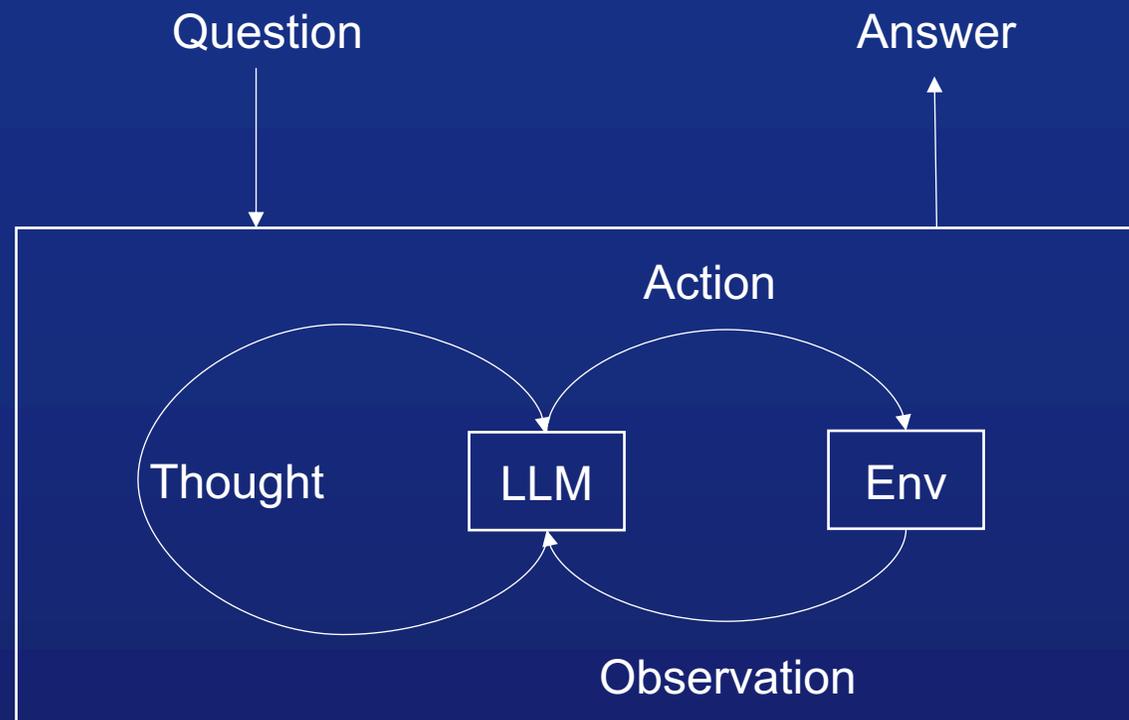
技术选型

LLM

Model	Tool Selection (Acc)
GPT-4	98.0%
GPT-3.5	74.5%
Qwen-14B-Chat	96.9%

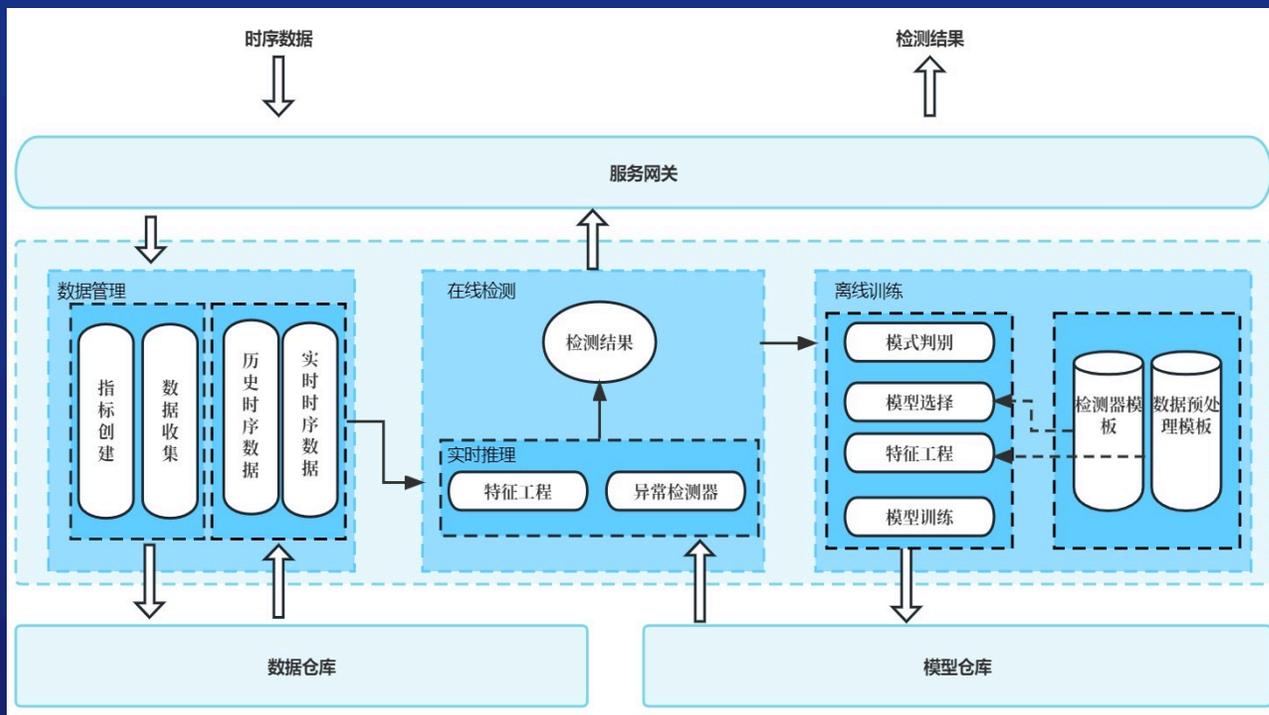
Quantization	Speed (Tokens/s)
BF16	32.22
Int8	29.28
Int4	38.72

Agent 框架 - ReAct



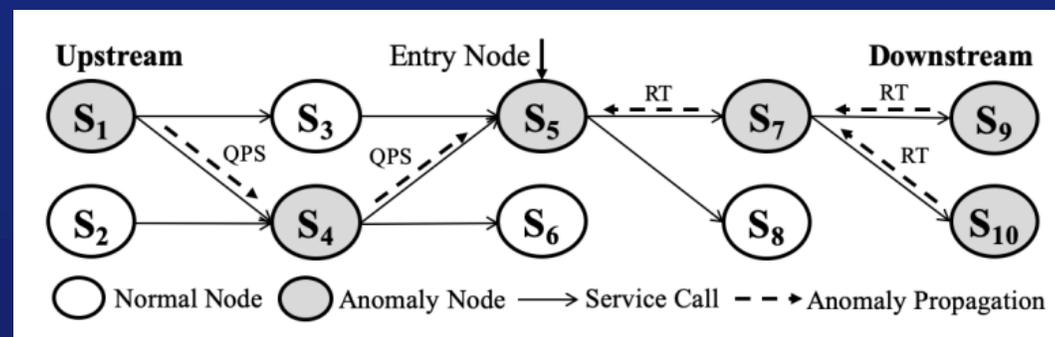
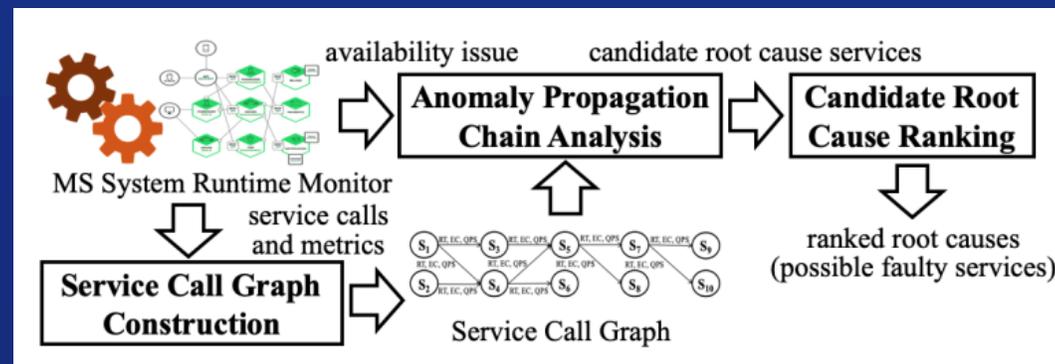
技术选型-AIOps Tools

异常检测



F1-Score>90% FPR<2%

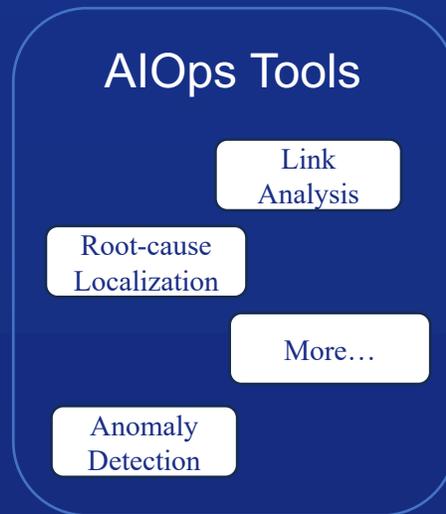
根因定位



Recall@1>70% Recall@3>90%

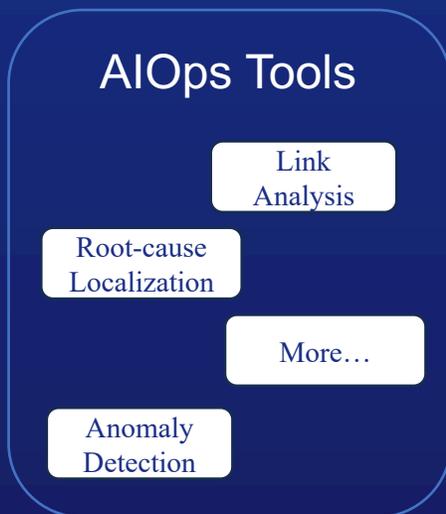
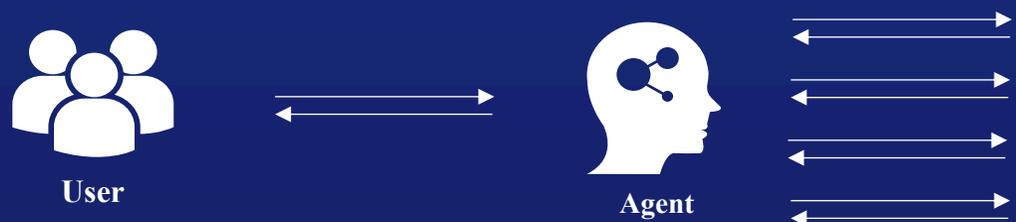
方案效果--以完成一次异常发现到根因定位过程为例

人机协同 – 用户与Agent通过对话协同解决:



- ① 获取节点最近发生的异常
- ② 根因定位
- ③ 根因节点的异常检测
- ④ KPI的异常情况

自我规划 – 用户提出问题, Agent自动思考解决问题:



模型自动规划步骤
调用工具获取结果

方案效果--以完成一次异常发现到根因定位过程为例

人机协同：



查询一下cmdb_id为nginx_03最近的异常

好的，您想知道cmdb_id为nginx_03的最近一次异常是什么？

Thought: 我需要使用查询最近异常API来获取这个信息。

Action: select_latest_anomaly_cmdb_id

Action Input: {"cmdb_id": "nginx_03"}

Observation: 异常检测结果为：

kpi_name	timestamp	current_value	last_week_value	last_day_value	is_anomaly
nginx_03^system.cpu.user^null	1695366120	7.86	17.1	26.32	True

Thought: cmdb_id为nginx_03最近的异常为7.86%的CPU用户率。

Final Answer: cmdb_id为nginx_03最近的异常为7.86%的CPU用户率。建议查看日志以确定原因，并采取相应措施。

方案效果--以完成一次异常发现到根因分析过程为例

人机协同：



ok,对这条异常执行根因分析

好的,我将帮助您对这条异常执行根因分析。

Thought: 需要调用根因定位API来完成根因定位任务。

Action: root_analysis

Action Input: {"kpi_name": "nginx_03", "timestamp": 1695366120}

Observation: root search ans:

trace_name	trace_score
nginx_03-->Weblogic_12-->Redis_01	374
nginx_03-->Weblogic_12-->Redis_02	374

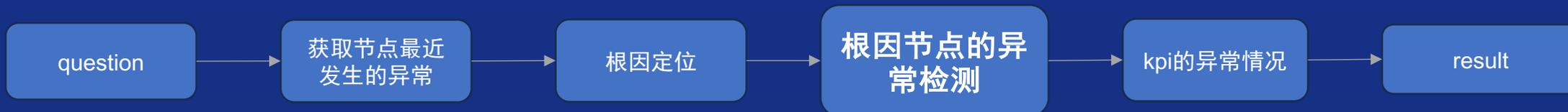
Thought: 根据观察,可以得出最后的答案。

Final Answer: 根据根因定位插件的推荐,推测可能是因为Weblogic12与Redis节点之间出现连接问题,进而影响了Nginx服务的正常运行。建议先尝试通过排查Weblogic和Redis节点的相关状态,进一步确认是否存在问题,并及时进行修复。



方案效果--以完成一次异常发现到根因分析过程为例

人机协同：



好的，那查询一下Redis_01最近的异常

Thought: 需要调用查询最近异常API来完成查询任务。

Action: select_latest_anomaly_cmdb_id

Action Input: {"cmdb_id": "Redis_01"}

Observation: 异常检测结果为：

kpi_name	timestamp	current_value	last_week_value	last_day_value	is_anomaly
Redis_01^system.io.wrqm_s^vda	1695398340	3	0	0	True
Redis_01^system.io.w_s^vda	1695398340	31	10	13	True
Redis_01^system.net.packets_in.count^eth0	1695398340	1407.45	1606.78	1449.05	True
Redis_01^system.net.packets_out.count^eth0	1695398340	919.52	1085.77	955.32	True

Thought: 根据观察，可以得出最后的答案。

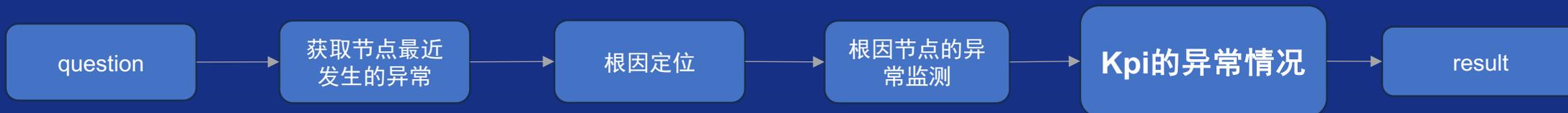
Final Answer: cmdb_id为Redis_01最近的异常有以下四个：

1. system.io.wrqm_s的值为3，说明其运行中的请求等待响应的平均时间为3毫秒，超出阈值范围；
2. system.io.w_s的值为31，说明其空闲套接字数量为31个，超出阈值范围；
3. system.net.packets_in.count的值为1407.45，表示网络接收包的数量超过了阈值；
4. system.net.packets_out.count的值为919.52，表示网络发送包的数量也超过了阈值。

这四个指标都表明Redis_01节点出现了性能问题，可能导致整个应用系统的响应速度变慢或者频繁出现错误。建议尽快采取措施，如增加硬件资源、优化配置等，以提升Redis_01节点的性能和稳定性。

方案效果--以完成一次异常发现到根因分析过程为例

人机协同：



好的，对kpi_name为Redis_01^system.io.w_s^vda的指标,使用异常检测工具查询一下在2023-09-22 20:00:00 到2023-09-22 23:59:00之间的异常

.....

Redis_01^system.io.w_s^vda	1695398040	24	9	9	True
Redis_01^system.io.w_s^vda	1695398160	32	8	10	True
Redis_01^system.io.w_s^vda	1695398220	22	8	9	True
Redis_01^system.io.w_s^vda	1695398280	26	9	9	True
Redis_01^system.io.w_s^vda	1695398340	31	10	13	True

Thought: 根据API返回的数据，在2023-09-22 20:00:00 到2023-09-22 23:59:00之间，cmdb_id为Redis_01^system.io.w_s^vda有14条异常记录，均显示为True，这意味着该指标在这段时间内一直存在异常。

Final Answer: 在2023-09-22 20:00:00 到2023-09-22 23:59:00之间，cmdb_id为Redis_01^system.io.w_s^vda有14条异常记录，均显示为True。需要对该指标进行详细调查，以找到是否存在真正的问题，然后决定如何处理它。建议可以考虑增加系统的负载均衡能力，减少一些不必要的压力，或者定期清理磁盘空间，以提高系统的性能和稳定性。



方案效果--以完成一次异常发现到根因分析过程为例

自我规划:



查询一下cmdb_id为nginx_03最近的异常，并对该异常执行根因分析，最后查询第一个异常链路根因节点最近的异常信息

Thought: 首先, 我需要使用 `select_latest_anomaly_cmdb_id` 工具来查询cmdb_id为nginx_03最近的异常, 然后将这个异常的ID作为输入, 再使用 `root_analysis` 工具进行根因分析, 最后再用这个根因分析结果中异常节点的ID作为输入再次调用 `select_latest_anomaly_cmdb_id` 工具查询最近的异常。

Action: `select_latest_anomaly_cmdb_id`

Action Input: {"cmdb_id": "nginx_03"}

Observation: 异常检测结果为:

kpi_name	timestamp	current_value	last_week_value	last_day_value	is_anomaly
nginx_03^system.cpu.user^null	1695366120	7.86	17.1	26.32	True

Thought: 现在我已经知道了一个最近的异常 (timestamp为1695366120), 我可以这个异常的cmdb_id和根因分析API进行根因分析。

Action: `root_analysis`

Action Input: {"kpi_name": "nginx_03", "timestamp": 1695366120}

Observation: root search ans:

trace_name	trace_score
nginx_03-->Weblogic_12-->Redis_01	374
nginx_03-->Weblogic_12-->Redis_02	374

Thought: 根据根因分析的结果, 我发现一个可能的原因是nginx与Weblogic以及Redis之间的通信问题。为了验证这一点, 我需要获取这些节点最近的异常信息。

方案效果--以完成一次异常发现到根因分析过程为例

自我规划:



Action: select_latest_anomaly_cmdb_id

Action Input: {"cmdb_id": "Redis_01"}

Observation: 异常检测结果为:

kpi_name	timestamp	current_value	last_week_value	last_day_value	is_anomaly
Redis_01 ^system.io.wrqm_s ^vda	1695398340	3	0	0	True
Redis_01 ^system.io.w_s ^vda	1695398340	31	10	13	True
Redis_01 ^system.net.packets_in.count ^eth0	1695398340	1407.45	1606.78	1449.05	True
Redis_01 ^system.net.packets_out.count ^eth0	1695398340	919.52	1085.77	955.32	True

Thought: 根据这些异常信息,我也发现Redis_01节点存在一些问题,可能是导致了整个系统的性能下降。

Final Answer: cmdb_id为nginx_03最近的异常在根因分析后得出可能是因为nginx与Weblogic以及Redis之间的通信问题,但是查询这两个节点最近的异常信息发现只有Redis_01节点存在异常。根据这些异常信息,可以推断出Redis_01节点可能是导致系统性能下降的原因之一。建议针对这个问题进一步深入调查和解决。





2023 CCF国际AIOps挑战赛决赛暨“大模型时代的AIOps”研讨会

THANKS



2023 CCF国际AIOps挑战赛决赛暨“大模型时代的AIOps”研讨会

THANKS