



中國計算機學會  
CHINA COMPUTER FEDERATION



Nightingale



GitLink  
— 确实 · 开源 —

# 第二届CCF·夜莺开发者创新论坛

中国北京 2024.7.26

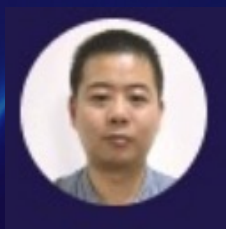
主办方: 中国计算机学会 | 承办方: CCF开源发展委员会、夜莺项目开源社区



中國計算機學會  
CHINA COMPUTER FEDERATION



# Ai Agent在运维体系中的探索与实践



钱誉

Zenlayer GFS Monitoring Automation 负责人

中国北京 2024.7.26

主办方: 中国计算机学会 | 承办方: CCF开源发展委员会、夜莺项目开源社区



# 大纲

- Zenlayer在运维体系中对Copilot & Agent理解
- Zenlayer在私域数据尝试：FT or RAG(Baseline & Graph)
- 场景实现案例
- 实践预期
- 未来展望
















## 1. 数据提取与汇总

AS-PATH	Count	Avg Delay (ms)
(21859, 23853)	19	1.06
(21859, 1299, 23853)	5	0.70
(21859, 4134, 4812, 23853)	63	168.62
(21859, 4837, 17621, 23853)	2	211.15

- **AS-PATH连续性分析:** 从上表可以看出, ASN23853节点通过不同的AS-PATH与ASN21859节点相连, 表明网络路径的多样性。
- **主路径:** (21859, 4134, 4812, 23853)路径流量最大, 占比为63, 平均延迟为168.62ms, 是主路径。
- **次要路径与间接路径:** (21859, 1299, 23853)和(21859, 4837, 17621, 23853)路径流量较小, 分别为5和2, 平均延迟较低, 可作为次要路径或间接路径。
- **关键上游ASNs:** 4134和4812是关键上游ASNs, 特别集中在主路径中。
- **抖动与延迟分析:** 平均延迟反映了路径的稳定性, (21859, 4134, 4812, 23853)路径的延迟最高, 可能需要关注其稳定性。

网络多样性体现在不同的AS-PATH上。这有助于提高网络的可靠性和冗余。从数据中可以看出,ASN23853节点通过多条路径与ASN21859节点相连,这表明网络设计中考虑了路径多样性。

- **整体连通性:** ASN23853节点与多个ASNs相连, 显示了良好的网络连通性。
- **局部连通性:** 与ASN21853直接相连的ASNs包括21859、1299、4134、4812、4837和17621, 显示了局部网络的连通性。

- **技术问题**: 高延迟路径可能需要优化。例如(21859, 4134, 4812, 23853)。
- **运维问题**: 监控网络路径的稳定性, 确保关键路径的健康。
- **资源问题**: 确保网络带宽资源充足, 避免瓶颈。
- **供应链问题**: 与供应商合作, 确保网络设备的及时更新和维护。

本报告的Sankey图链接如下:

<http://ai.zenlayer.win:8081/2024-06-24/d601ea8c-04a7-492f-b93d-ba3ac66686a0.html>

此链接提供了基于ASN23853节点的网络路径的可视化分析。

多组 AI 对话 134.59s 查看详情



# 场景实现案例

系统配置

可以配置应用的系统参数

对话开场白

每次对话开始前，发送一个初始内容。支持标准 Markdown 语法。可使用的特殊标记：[快捷按钮]：用户点击后可以直接发送问题

全局变量

新增

语音播放

浏览器自审(免费)

语音输入

关闭

猜你想问

未开启

定时执行

未开启

输入引导

关闭

流程开始

输出

用户问题 string

全局变量

应用 ID string

当前对话 ID string

AI 回复的 ID string

最近 10 条聊天记录 历史记录

当前时间 string

知识库搜索

调用“语义检索”和“全文检索”能力，从“知识库”中查找可能与问题相关的参考内容

输入

选择知识库 手动选择

选择

VictoriaMetrics

搜索参数设置

搜索模式	引用上限	最低相关性	结果重排	问题优化
混合检索	2000	0.8	✓	语义干预24c-72b3

用户问题 string 变量引用

流程开始 > 用户问题

输出

知识库引用 知识库引用

AI 对话

AI 大模型对话

输入

AI 模型 手动选择

通义千问24c-72b大模型

提示词 手动输入

You are a Prometheus and VictoriaMetrics expert and you are required to answer all the questions for these two technologies.  
If PromQL is applicable, give an example of writing PromQL.  
If MetricsQL is applicable, give an example of writing MetricsQL.  
Summarize the logic and scenarios of configuration usage, give suggestions and examples of configuration.  
If you don't know how to answer, say you don't know, don't make it up.

聊天记录 手动输入

6

用户问题 string 变量引用

流程开始 > 用户问题

知识库引用 知识库引用

知识库搜索 > 知识库引用

输出

新的上下文 历史记录

AI 回复内容 string

我需要查询BGP路由条目在1小时内的收方向和发方向的路由变化次数。收方向的label为bgp\_received, 发方向的label为bgp\_send

根据您的请求，您想要查询的是BGP路由条目在1小时内的变化次数，分别针对收方向和发方向。但是，您提到的 bgp\_received 和 bgp\_send 看起来像是描述方向的标签值，而不是标签名。通常，Prometheus和VictoriaMetrics的metric会有一个特定的名称，例如 bgp\_routes，并且可能带有 direction 这样的标签来区分收发方向。

假设我们有一个名为 bgp\_routes 的metric，它带有 direction 标签，其中 direction 的值可以是 received 或 send，那么，您可以使用以下PromQL查询来获取所需的信息：

markdown

# 收方向路由变化次数  
increase(bgp\_routes(direction="received"), 1h)

# 发方向路由变化次数  
increase(bgp\_routes(direction="send"), 1h)

increase 函数用于计算指定时间范围内的增加量。在这个例子中，我们使用 [1h] 时间范围来计算过去1小时内的变化次数。

如果您使用的是VictoriaMetrics，并且想要使用MetricsQL，那么查询将非常相似：

markdown

# 收方向路由变化次数  
increase(bgp\_routes(direction="received"), 1h)

# 发方向路由变化次数  
increase(bgp\_routes(direction="send"), 1h)

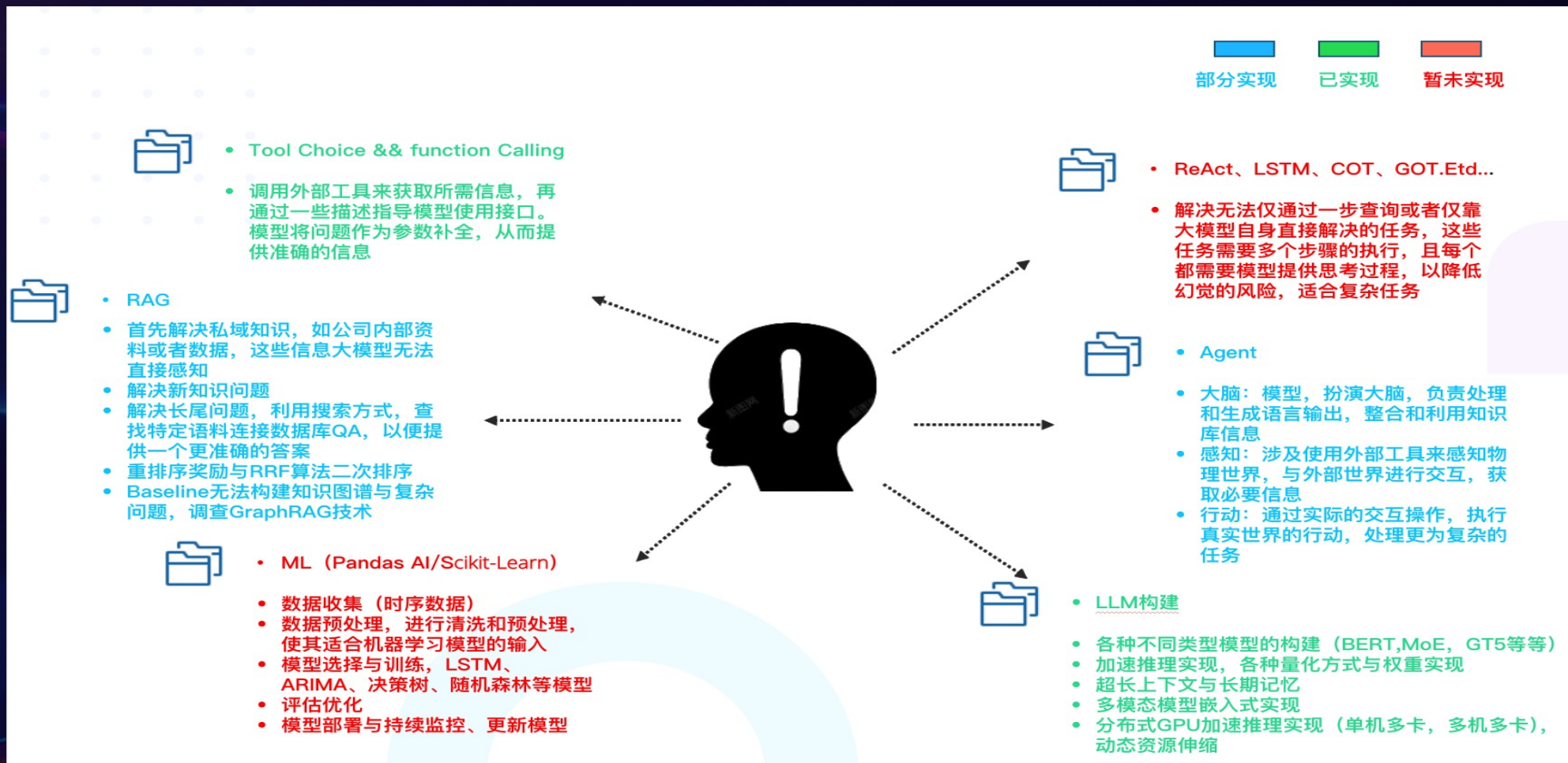
请注意，您需要将 bgp\_routes 替换为您实际的metric名称，将 direction 替换为您实际的标签名，将 received 和 send 替换为您实际的标签值。如果您的metric名称和标签与示例不同，那么请根据实际情况调整上述查询。

引用

vmgateway Cluster version VictoriaMetrics MetricsQL vmauth Writer

9条引用 3条上下文 38.77% 查看详情

# 实践预期





# 未来展望

## 基于 LLM 的 RCA-Agent 构建

目标是先将基于大语言模型的根因诊断 ( RCA ) Agent 框架落地应用，因为根因诊断是所有运维团队面临的一个主要挑战，它占用了大量的时间和精力，日常的 On Call 问题定位也给团队成员带来了沉重的负担。我们希望专注于解决这些实际问题，真正缓解同事的痛点。

我们决定定义了一些工具和插件，是在出现故障时用来进行检测的工具。除了工具和插件，我们还设计了工作流编排，以自动化和优化故障处理流程。我们构建了一个知识库，它包含了历史故障数据、专家经验和故障处理策略，这些都是进行有效根因分析的关键资源

## 知识库/知识图谱的构建

构建知识库方面所做的工作主要包括以下几个部分，并且我们计划未来会引入更多用户原始文档、历史 On Call 记录等不同类型的数据。

排障专家经验：这部分是根据根因诊断的场景特别设计的，目的是让业务团队的成员能够管理和记录他们的知识和经验。定义的每一个经验都是一组根因故障，包括故障发生时的描述和一些止损措施的组合。这些信息将被用来训练大语言模型推理。

故障场景 SOP 文档：我们希望用户输入的是一些 SOP 文档。这种方式给组件团队提供一种灵活管理知识的方法。选择这种半规范化文档的形式，是因为当前大语言模型的能力还有局限，需要通过文档梳理来帮助模型更好地理解。

历史故障信息：建立维护一个历史故障信息库，记录每一次通过大语言模型检测到的故障，这些记录会用来对组件团队进行训练和打标。

## 基础工具的构建

在构建框架的基础工具方面，我们参考了 OpenAI GPTs 将工具集成到平台时所遵循的规范。我们将运维场景中的一些关键指标和基础工具进行了统一管理，把传统的异常检测方法统一成一个工具，用户只需要维护他们需要进行异常检测的指标即可。

用户可以自定义检测项，包括指标名称、指标的标签或指标描述，以及定义何为异常表现。因为是由用户自定义的工具，所以可以根据具体需求设置检测标准。

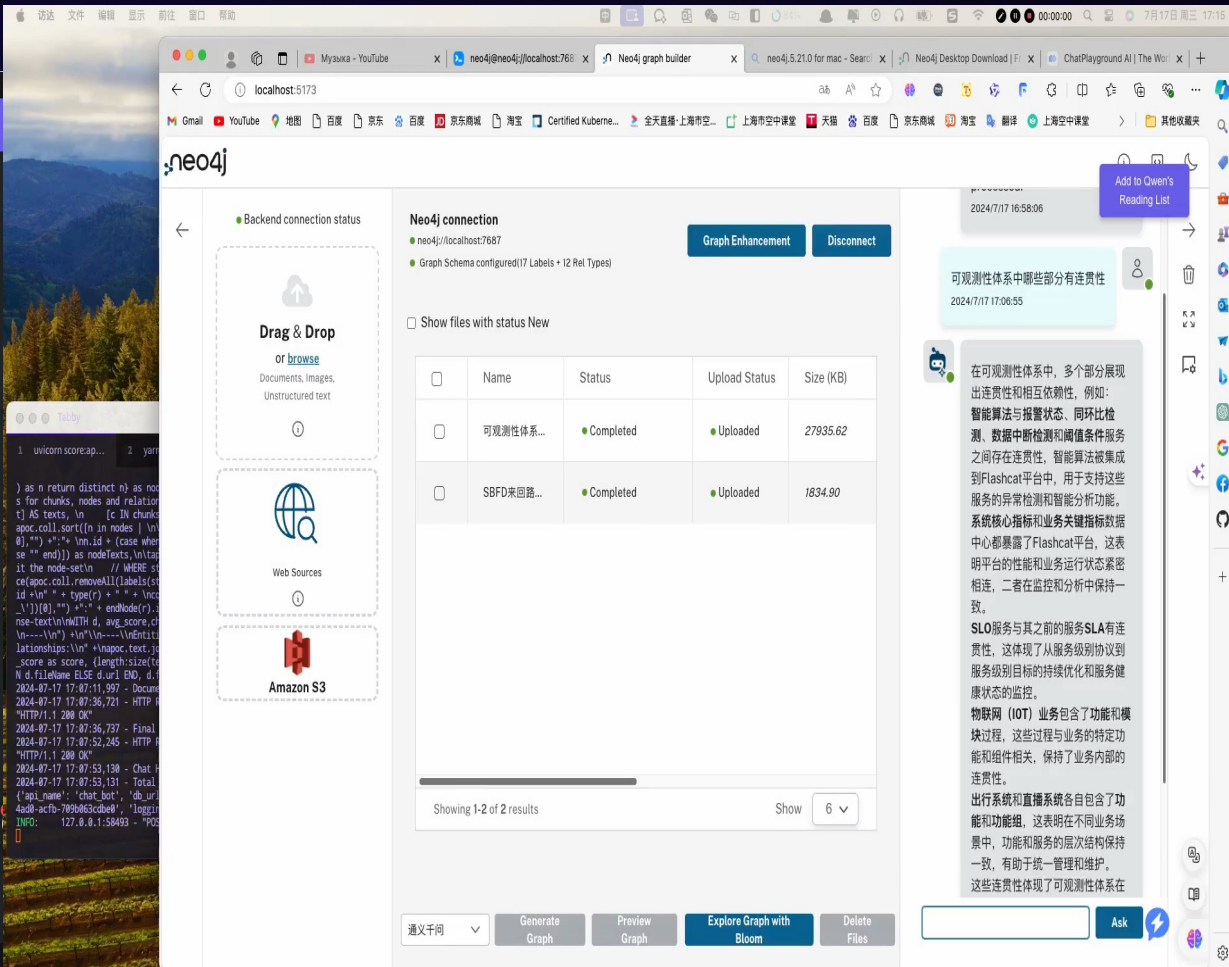
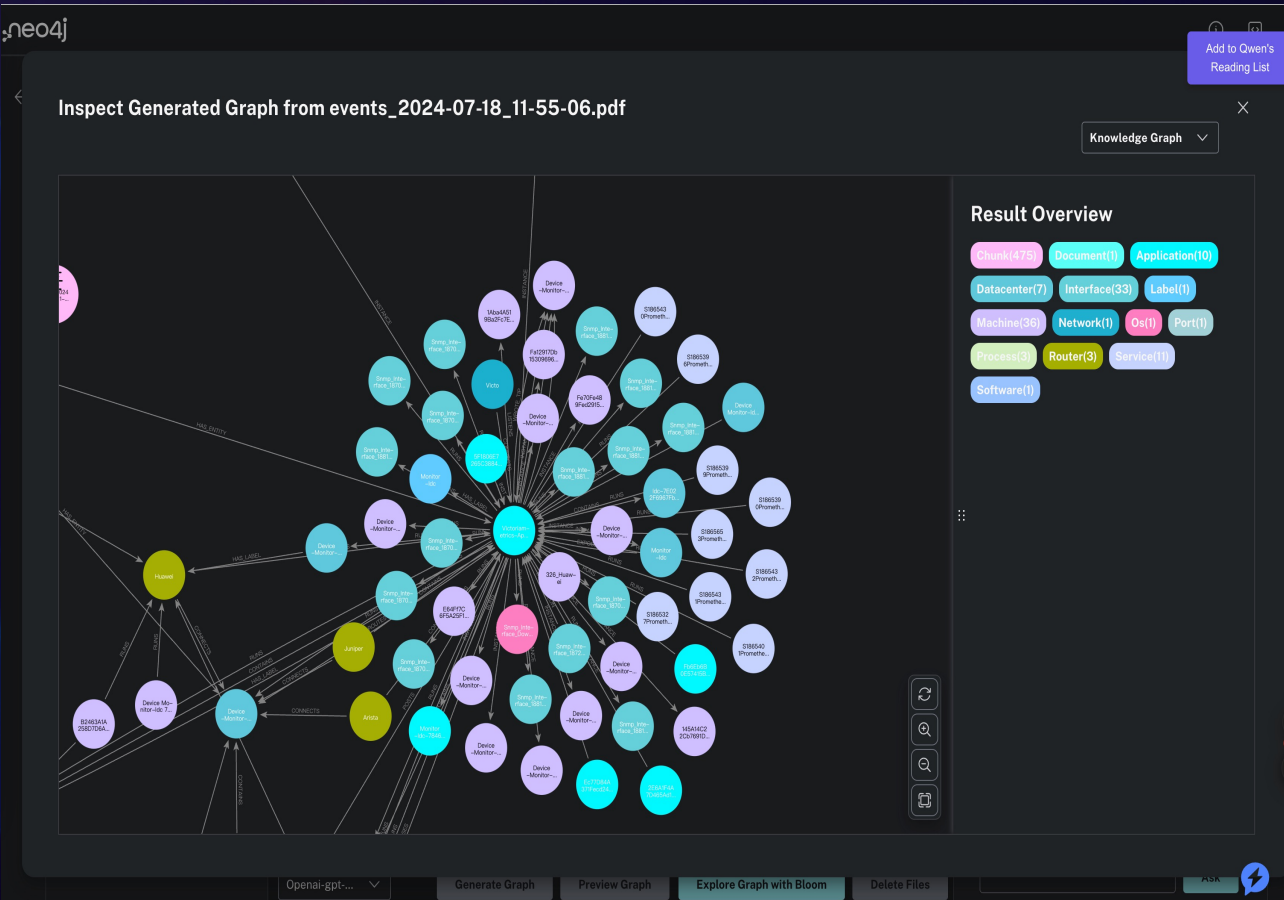
实现了一个变更事件查询工具，当出现故障时，用户可以通过调用这个接口来确定是否由线上变更导致。我们在平台上部署的组件配置了一些工具，例如异常检测、变更和事件查询等，还包括了自然语言的意图理解和大语言模型的根因推理功能。

## 工作流的构建 ( SOP )

构建工作流，目前这一过程仍然需要用户自行配置，这主要是由于大语言模型当前能力的限制所做出的妥协。不过，我们正在探索一种新的方法，即允许用户在其 SOP 文档中预先设定工作流，例如，文档中可以指明首先需要检查哪些指标，以及根据这些指标的结果接下来应该检查哪些指标。

我们希望能够训练大语言模型，使其能够直接根据用户的 SOP 文档生成工作流。最终，运维团队能够向大语言模型提供一个简单的文档，甚至是未经格式化的文本，而模型能够根据文档中的指标或检测项动态地编排诊断步骤，并根据每一步的检测结果，智能地调度后续的执行流程。

# 彩蛋---Neo4J-RAG





# 感谢聆听

# Thank you for listening