

10.13 数据预处理调研

上次会议小结

Data Refinement

Metrics 特征分析, workload画像, pattern识别, 时序预测

Logs 优先级分类, 关键字过滤, LLM总结

Traces 用图的方法

Agent

Prompt template设计

是否加入领域知识, 知识图谱

调用远端大模型/用本地

《TVDiag: A Task-oriented and View-invariant Failure Diagnosis Framework for Microservice-based Systems with Multimodal Data》

Metrics

使用 3-sigma 规则[1], 虽然其性能略低, 但提供了快速的处理速度。对于待检测的某个指标, 收集其在一段时间内的数值波动。随后, 计算这些波动的均值和标准差。如果其值超过 $u + 3\text{-sigma}$ (或小于 $u - 3\text{-sigma}$), 则被视为该指标的“上升”(或“下降”)方向的告警。指标名称和异常方向随后会被记录为警报(alert, 即提取出来的有用信息)。

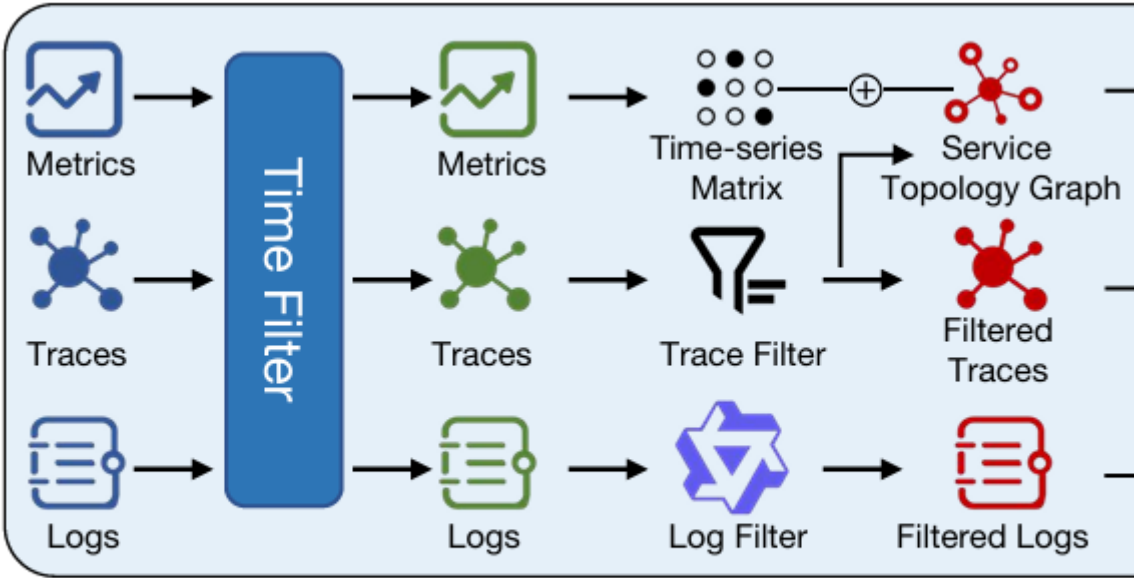
Logs

TVDiag 使用 Drain[2] 解析日志, 并将每个日志与一个日志键匹配。由于某些日志键通常与特定故障相关, TVDiag 将这些有价值的日志键检测为日志告警。此外, 应避免使用与故障无关且常见的日志键, 因为它们可能引入不必要的噪音, 甚至误导模型。作者认为ERROR级别和低频日志键在故障诊断中更有帮助。TVDiag 使用两条规则来生成告警: 规则 1: 所有 ERROR 级别的 logKeys 都被视为 alert。可以通过搜索预定义关键字来识别 ERROR 级别的 logKey。规则 2: 历史中出现频率最低的 top-k logKeys 被视为告警。请注意, k (默认值为 0.5) 是一个超参数, 操作人员可以根据自己的偏好进行调整。

Traces

采用Isolation Forest (IForest) [3] 来检测异常，并根据 Traces 中每对调用的响应时间和状态码生成 alert。IForest通过随机隔离数据点来构建决策树，并通过观察隔离它们所需的平均路径长度来识别异常。操作员通常通过交互、响应时间和 Traces 中的状态码来检测系统故障。较高的响应时间通常意味着性能下降，而异常状态码则表示业务错误。

《 TrioXpert: An automated incident management framework for microservice system 》



(a). Multimodal Data Preprocessing

Metrics

考虑到 Metrics 的时间序列特性，它被组织成一个三维时间序列矩阵 $\mathcal{M} \in \mathbb{R}^{T \times S \times F}$ 。该矩阵的三个维度分别表示时间戳、服务实例和特征通道。这种结构化的表示不仅有助于后续的时间序列特征提取，还能有效捕捉系统运行状态的动态变化。

Logs

TrioXpert采用基于大型语言模型（LLM）的两阶段过滤机制，以提取与事件相关的日志条目，从而减少噪声并提高跨场景适应性。受COMET [4] 启发，该方法保留了将关键字过滤与语义精炼相结合的核心思想。在第一阶段，LLM被提示从数据集中提取与事件相关的关键字（例如，根据”在以下日志中识别常与系统故障相关的关键术语”这一指令生成“error”、“failure”或“timeout”等术语）。然后使用这些关键字过滤候选日志，再将其输入第二阶段进行更深入的语义分析。在第二阶

段，LLM通过分析日志条目在事件模式中的语义内容来识别重要日志条目，例如它们是否包含异常操作、错误指示或系统关键事件。

Traces

由于 Traces 数据中不同调用类型（例如 HTTP、RPC）的延迟分布特征各异，作者分别计算每种调用类型所有跨度的 P95 延迟百分位数，并将其作为阈值。对于延迟超过该阈值的跨度，会递归追溯其父跨度直至根节点，以确保完整调用链的保留。这种方法在移除无关跟踪数据的同时，支持对异常调用链的上下文分析。此外，按照 ART [5] 和 DiagFusion [6] 的方法，TrioXpert 从跟踪数据中提取服务拓扑图 G，并保持与 Metrics 数据相同的采样频率，以描述任意时刻服务实例的调用关系。

[1] Friedrich Pukelsheim. 1994. The three sigma rule. *The American Statistician* 48, 2 (1994), 88–91.

[2] Pinjia He, Jieming Zhu, Zibin Zheng, and Michael R Lyu. 2017. Drain: An online log parsing approach with fixed depth tree. In *2017 IEEE international conference on web services (ICWS)*. IEEE, 33–40.

[3] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 eighth IEEE international conference on data mining*. IEEE, 413–422.

[4] Z. Wang, J. Li, M. Ma, Z. Li, Y. Kang, C. Zhang, C. Bansal, M. Chintalapati, S. Rajmohan, Q. Lin et al., “Large language models can provide accurate and interpretable incident triage,” in *2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2024, pp. 523–534.

[5] Y. Sun, B. Shi, M. Mao, M. Ma, S. Xia, S. Zhang, and D. Pei, “Art: A unified unsupervised framework for incident management in microservice systems,” in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 1183–1194.

[6] S. Zhang, P. Jin, Z. Lin, Y. Sun, B. Zhang, S. Xia, Z. Li, Z. Zhong, M. Ma, W. Jin et al., “Robust failure diagnosis of microservice system through multimodal data,” *IEEE Transactions on Services Computing*, vol. 16, no. 6, pp. 3851–3864, 2023.