

Sage: Practical and Scalable ML-Driven Performance Debugging in Microservices



阅读时间：2023/5/29

资源角度入手，分析由资源引起的故障

rpc级别的延迟跟踪和容器/节点级别的监控

只依靠现场的数据，不是离线训练

- **Sage**:一种机器学习驱动的根本原因分析系统，关注实用性和可伸缩性，QoS（服务质量）
- **研究内容**：利用**无监督**的方法去规避轨迹标记的开销，捕获微服务间的依赖关系的影响。利用因果贝叶斯网络（CBN）捕获端到端应用程序拓扑中微服务间的依赖关系，通过图形变分自动编码器（GVAE）生成假设的场景(反事实)，来检查微服务对端到端性能的影响

只使用用户级指标,用Prometheus收集，每次间隔10s

三种节点类型：指标节点X（服务器相关、网络相关）、延迟节点Y、潜在变量Z

案例系统：DeathStarBench

root cause analysis

rpc级别的延迟跟踪和容器/节点级别的监控

定期重新训练模型

训练模型时可以选用当前和之前的数据，以避免遗忘之前的知识

检测方法：当发生异常（QoS降低）后，把所有的指标都恢复成正常状态，然后通过调指标使得出现异常情况时，定位当前指标为根因。

如何适应新节点的加入？

迁移学习：动态可重塑GVAE实现选择一部分再训练和增量再训练

- (1) 只对有因果关系的节点进行再训练
- (2) 将参数初始化为前一个模型的参数，再修改相应的网络

微服务调用关系结构图

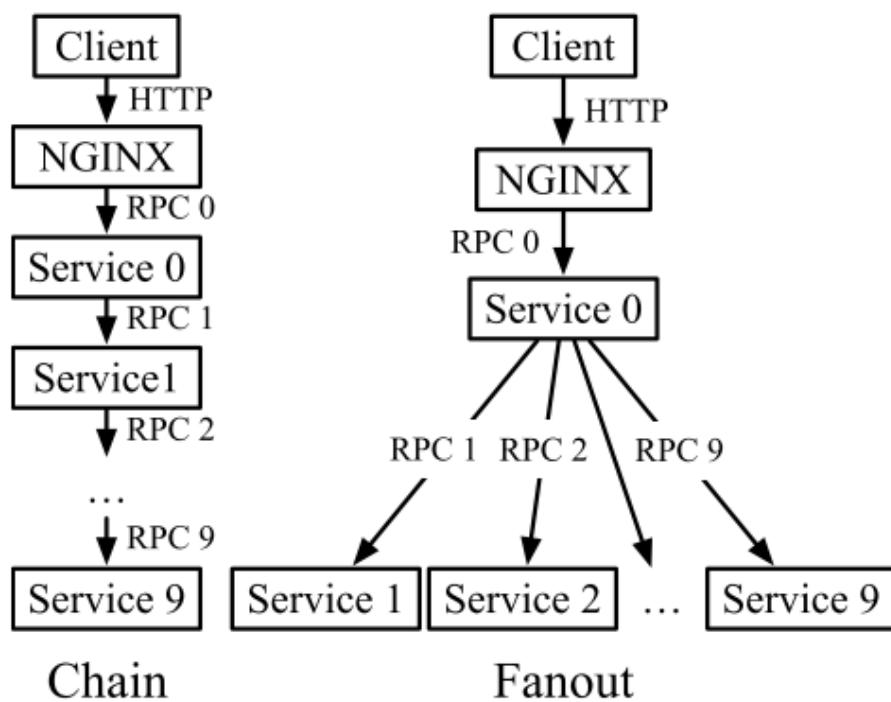


Figure 8: RPC dependency graph for the two synthetic Chain and Fanout services.

实验设置

- (1) 6个对比实验
- (2) 层不完整的情况下的准确性
- (3) 不同训练量对模型的影响
- (4) 不同采样频率的影响