

基于机器学习的分布式系统故障诊断系统

需求文档

张骁凯
2023-09-04





CONTENTS

- 项目背景
- 项目目标
- 业务场景
- 系统需求
- 技术选型



01

项目背景

项目背景

在大数据时代，分布式系统已经成为信息存储和处理的主流方式。由于其庞大和复杂的特性，分布式系统的故障发生率相对较高，运维难度和复杂度也随之增加。如何对分布式系统进行高效、准确的运维，成为保障信息系统高效、可靠运行的关键





02

项目目标

项目目标

- 利用机器学习、深度学习等技术对分布式系统的故障数据进行分析。
- 设计故障诊断模型，准确地分析并识别故障类别。
- 实现分布式系统故障运维的智能化，快速恢复故障，降低运维难度，减少人力资源消耗。



03

业务场景

业务场景



当分布式系统中的某个节点发生故障时，故障会沿着系统的拓扑结构传播，导致相关的KPI指标和日志异常。本项目将提供分布式数据库的故障特征数据和标签数据，其中特征数据是系统发生故障时的KPI指标数据，标签数据为故障类别数据。



04

系统需求

系统需求

1 基于机器学习的故障诊断功能 (SystemHealer-SR1)

2 WEB平台功能 (SystemHealer-SR2
SR2)

3 基于机器学习的故障诊断功能 (SystemHealer-SR3)

4 WEB平台功能 (SystemHealer-SR4
SR4)

1 基于机器学习的故障诊断功能 (SystemHealer-SR1)

1. 初始假设:

分布式系统中存在多个节点。

每个节点有一系列的KPI指标, 如feature0、feature1
...feature106。

故障会导致某些KPI指标异常。

2. 正常状态:

分布式系统的所有节点正常运行。

所有KPI指标在正常范围内。

3. 有哪些会出错:

节点可能会发生故障。
故障会导致相关的KPI指标异常。

故障可能会沿着分布式系统的拓扑结构传播。

4. 其他活动:

用户可以上传训练数据并在线训练模型。

用户可以上传单条或多条测试语句进行测试。
系统支持可视化测试结果和下载测试结果。

5. 完成的系统状态:

故障被准确地诊断并分类。
分布式系统恢复到正常状态。

1

2

3

4

5

2 WEB平台功能 (SystemHealer-SR2)

1. 初始假设:

用户需要一个界面来上传训练数据和测试数据。
用户希望在线进行模型的训练和测试。

1

2. 正常状态:

WEB平台正常运行。
用户可以顺利上传、训练和测试。

2

3. 有哪些会出错:

网络连接问题。
用户上传的数据格式不正确。
模型训练失败。

3

4. 其他活动:

WEB平台提供模型下载功能。
WEB平台支持测试结果的可视化和下载。

4

5. 完成的系统状态:

用户成功完成模型的训练和测试。
用户可以下载训练的模型和测试结果。

5

3 基于机器学习的故障诊断功能 (SystemHealer-SR3)

1. 初始假设:

分布式系统中存在多个节点。

每个节点有一系列的KPI指标, 如feature0、feature1
...feature106。

故障会导致某些KPI指标异常。

1

2. 正常状态:

分布式系统的所有节点正常运行。

所有KPI指标在正常范围内。

2

3. 有哪些会出错:

节点可能会发生故障。
故障会导致相关的KPI指标异常。

故障可能会沿着分布式系统的拓扑结构传播。

3

4. 其他活动:

用户可以上传训练数据并在线训练模型。

用户可以上传单条或多条测试语句进行测试。
系统支持可视化测试结果和下载测试结果。

4

5. 完成的系统状态:

故障被准确地诊断并分类。
分布式系统恢复到正常状态。

5

4 WEB平台功能 (SystemHealer-SR4)

1. 初始假设:

用户需要一个界面来上传训练数据和测试数据。
用户希望在线进行模型的训练和测试。

1

2. 正常状态:

WEB平台正常运行。
用户可以顺利上传、训练和测试。

2

3. 有哪些会出错:

网络连接问题。
用户上传的数据格式不正确。
模型训练失败。

3

4. 其他活动:

WEB平台提供模型下载功能。
WEB平台支持测试结果的可视化和下载。

4

5. 完成的系统状态:

用户成功完成模型的训练和测试。
用户可以下载训练的模型和测试结果。

5



05

技术选型

技术选型

机器学习部分：Python + PyTorch

WEB平台：Django + Bootstrap + JavaScript

THE END

THANKS

