<Hive反模式检测与修复>

测试评估报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <29/11/2020> | <1.0> | 测试评估报告1.0 | 任姚丹珺、贾兴国 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 测试结果摘要 5

3. 基于需求的测试覆盖 7

4. 基于代码的测试覆盖 7

5. 建议措施 8

测试评估报告

# 简介

## 目的

本测试报告用来阐述Hive反模式检测与修复系统的针对测试计划的测试结果

## 范围

本测试评估报告仅针对Hive反模式检测与修复系统

## 定义、首字母缩写词和缩略语

Hive反模式检测与修复系统：项目名称

静态检测：通过解析AST抽象语法树，用户可以通过自定义的Antlr的Listener类继承BaseListener类，在监听器方法中编写规则来进行反模式的检测。

动态检测：HiveQL经过Hive框架优化后生成一系列的Map-Reduce任务，由Hadoop进行资源分配和任务的执行。通过一定的规则，对运行时任务的运行时间，负载等日志信息进行分析来找出可能的反模式

反模式：指用来解决问题的带有共同性的不良方法。它们已经经过研究并分类，以防止日后重蹈覆辙，并能在研发尚未投产时辨认出来。

## 参考资料

无

## 概述

本报告主要用来展示测试结果

# 测试结果摘要

功能测试摘要，详见Antlr静态检测测试用例：

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“过多使用join”的反模式 |
| 方法： | 检测语句“select t3.subname,t4.age from (select t1.subname,t2.age from t1 inner join t2 on t1.id = t2.id) as t3 inner join t4 on t4.name = t3.name;”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“过多使用join” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“在join或where子句中进行运算”的反模式 |
| 方法： | 检测语句“select t1.col1, t2.col2 from table1 as t1 join table2 as t2 on t1.col1 + 100 = t2.col2;”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“在join或where子句中进行运算” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“使用 select \*”的反模式 |
| 方法： | 检测语句“select \* from t1;”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“使用 select \*” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“使用 order by”的反模式 |
| 方法： | 检测语句“select \* from t1 order by age;”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“使用 order by” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“使用having进行过滤”的反模式 |
| 方法： | 检测语句“select t1.col1, t2.col2 from t1 join t2 on t1.id = t2.id group by t1.col1 having t2.col2 > 2000;”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“使用having进行过滤” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“在谓词中使用函数”的反模式 |
| 方法： | 检测语句“select t1.col1, t2.col2 from t1 inner join t2 on abs(t1.col1) = t2.col2;”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“在谓词中使用函数” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“select的列未在group by中”的反模式 |
| 方法： | 检测语句“select col1,col2 from t1 group by col1;”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“select的列未在group by中” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“在date\_sub()中使用interval”的反模式 |
| 方法： | 检测语句“select date\_sub('2020-9-16', 2) from a;”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“在date\_sub()中使用interval” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“case语句中then和else后数据类型不一致”的反模式 |
| 方法： | 检测语句“SELECT ID,CASE WHEN col\_a = 0 THEN 0 ELSE 2.2 END AS math\_is\_fun FROM t1;”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“case语句中then和else后数据类型不一致” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“group by没有与聚集函数搭配使用”的反模式 |
| 方法： | 检测语句“select sum(col3),col2 from t1 group by col1,col2;”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“group by没有与聚集函数搭配使用” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

|  |  |
| --- | --- |
| 测试目标： | 测试静态检测模块能否检测出“在大表中(频繁)使用count(distinct)”的反模式 |
| 方法： | 检测语句“select count(cookie) from weblogs where dt <= ${today} and dt >= ${90daysAgo};”中存在的反模式 |
| 测试结果： | 检测出存在的反模式有“在大表中(频繁)使用count(distinct)” |
| 需考虑的特殊事项： | 被检测语句中存在嵌套语句、不同表内属性名重复 |

接口测试：

见接口测试文档

压力测试：

见压力测试文档

准确率测试：

|  |  |
| --- | --- |
| 测试目标： | 检测建议的准确率大于90% |
| 方法： | 人工检测所有的测试用例，并给出其检测建议的正确率 |
| 测试结果： | 在96个测试用例中，检测建议正确的有93个，正确率为90% |
| 需考虑的特殊事项： | 测试环境正常运行，系统不出现无关的报错 |

|  |  |
| --- | --- |
| 测试目标： | 修复结果的准确率大于90% |
| 方法： | 对于系统已实现修复的Anti-Pattern进行人工判断 |
| 测试结果： | 修复正确的用例有35个，错误的有3个，准确率为92.1% |
| 需考虑的特殊事项： | 测试环境正常运行，系统不出现无关的报错 |

|  |  |
| --- | --- |
| 测试目标： | MLP reduce预测准确率大于90% |
| 方法： | 直接运行预测程序 |
| 测试结果： | 预测完全正确的准确率为：17.58%  预测结果+-1个reduce准确率：87.91%  预测结果+-3准确率：94%，达到了测试目标 |
| 需考虑的特殊事项： | 预测reduce不准确时，实际reduce和预测reduce设置下，运行时间差距不大，且此时预测reduce是局部最低点，而不是全局最低。 |

# 基于需求的测试覆盖

测试已覆盖几乎所有需求，从测试的结果来看，测试发现了一些存在的缺陷，这些缺陷中不严重的缺陷均已被修复。

# 基于代码的测试覆盖

代码仍在编写中，无法得知基于代码的测试覆盖率。

# 建议措施

深入理解HiveQL的语句结构，尤其是Antlr语法分析器的Listener类对AST树的解析，从而更精确定位反模式。