Hive反模式的检测与修复

测试计划

版本 1.0

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 29/11/2020 | 1.0 | 撰写测试计划 | 任姚丹珺，贾兴国 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 背景 4

1.3 范围 4

1.4 项目标识 5

2. 测试需求 6

3. 测试策略 7

3.1 测试类型 7

3.1.1 数据和数据库完整性测试 7

3.1.2 功能测试 7

3.1.3 业务周期测试 9

3.1.4 用户界面测试 10

3.1.5 性能评价 11

3.1.6 负载测试 12

3.1.7 强度测试 13

3.1.8 容量测试 14

3.1.9 安全性和访问控制测试 15

3.1.10 故障转移和恢复测试 16

3.1.11 配置测试 18

3.1.12 安装测试 19

3.2 工具 20

4. 资源 21

4.1 角色 21

4.2 系统 23

5. 项目里程碑 24

6. 可交付工件 25

6.1 测试模型 25

6.2 测试日志 25

6.3 缺陷报告 25

7. 附录 A：项目任务 26

测试计划

# 简介

## 目的

Hive反模式的检测与修复项目的这一“测试计划”文档有助于实现以下目标：

• 确定现有项目的信息和应测试的软件构件。

• 列出需要满足的测试需求（高层次）。

• 确定采用的测试策略，并对这些策略加以说明。

• 确定所需的资源，并对测试的工作量进行估计。

• 列出测试项目的可交付元素。

## 背景

Hive反模式的检测与修复系统是为了方便程序员对自己编写的HiveQL的正确性以及性能进行检测而开发的工具。目的是通过反模式的静态检测和动态监测，精准定位出HiveQL中的不良代码，并给出恰当的修改意见，从而方便程序员即使在工期紧张时，也能快速生产出高质量的代码。

本项目提供的功能包括：用户通过交互界面提交HiveQL代码，提交静态/动态检测与修复请求；用户对数据库相关配置项做出设置；用户获得系统返回的修复与检测结果。

本项目采取B/S架构，用户采用浏览器作为客户端功能访问界面，本项目的后端以及Hadoop集群运行在实验室服务器上，为客户端发来的请求提供相应的检测和反馈。

本项目从10月1日立项，到1月6日结项，计划基于Scrum开发模型采用4次迭代，前三次迭代每次4周，最后一次耗时2周，每次迭代会提交相应的系统版本，最终产生稳定的系统版本与全面的文档材料。

## 范围

本测试计划书针对Hive反模式的检测与修复项目的需求规约书中提出的功能性与非功能性需求进行测试，要测试的内容包括：

|  |  |
| --- | --- |
| 测试内容 | 测试范围 |
| 功能测试 | * 请求反模式的静态检测与修复 * 请求反模式的动态检测与修复 * 设定数据库服务器相关配置项 * 获取检测与修复结果 |
| 性能测试 | 1. 模块 2. 静态检测模块 3. 动态检测模块 4. 性能 5. 系统最多可支持200个并发请求 6. 在95%的情况下，系统在10s内对用户请求给出响应 7. 可支持性 8. Web前端支持IE10以上和Chrome、Firefox、Edge这些主流浏览器。 9. 前端采用Airbnb JavaScript命名规范，后端使用阿里巴巴Java开发命名规范。 |

我们不对本项目的反模式表中未涉及的anti-pattern进行测试。

## 项目标识

下表列出了制定*测试计划*所用的文档，并标明了文档的可用性：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档 （版本/日期）** | **已创建或可用** | **已被接受或已经过复审** | **作者或来源** | **备注** |
| 需求规约 | √ 是 🞏 否 | √是 🞏 否 | 全体成员 |  |
| 功能性规约 | √是 🞏 否 | √是 🞏 否 | 全体成员 |  |
| 用例报告 | √ 是 🞏 否 | 🞏 是 √否 | 全体成员 | 根据开发进程，持续完善 |
| 项目计划 | √是 🞏 否 | √ 是 🞏 否 | 茅悦田、崔楠 |  |
| 设计规约 | √ 是 🞏 否 | √是 🞏 否 | 原帅 |  |
| 原型 | √ 是 🞏 否 | 🞏 是 √ 否 | 任姚丹珺 | 尚未完善 |
| 用户手册 | 🞏 是 √ 否 | 🞏 是 √否 | 贾兴国 | 是计划交付的文档 |

# 测试需求

下面列出了那些已被确定为测试对象的项目（用例、功能性需求和非功能性需求）。此列表说明了测试的对象。

* 请求反模式的静态检测与修复
* 请求反模式的动态检测与修复
* 设定数据库服务器相关配置项
* 获取检测与修复结果
* 静态检测模块
* 动态检测模块
* 系统最多可支持200个并发请求
* 在95%的情况下，系统在10s内对用户请求给出响应
* Web前端支持IE10以上和Chrome、Firefox、Edge这些主流浏览器。
* 前端采用Airbnb JavaScript命名规范，后端使用阿里巴巴Java开发命名规范。

# 测试策略

[测试策略提供了推荐用于测试对象的方法。上一节“测试需求”中说明了将要测试哪些对象，而本节则要说明如何对测试对象进行测试。

对于每种测试，都应提供测试说明，并解释其实施和执行的原因。

如果不实施和执行某种测试，则应该用一句话加以说明，并陈述这样做的理由。例如，“将不实施和执行该测试。。该测试不合适。”

制定测试策略时所考虑的主要事项有：将要使用的方法以及判断测试何时完成的标准。

下面列出了在进行每项测试时需考虑的事项，除此之外，测试还只应在安全的环境中使用已知的、受控的数据库来执行。 ]

## 测试类型

### 功能测试

[测试对象的功能测试应该侧重于可以被直接追踪到用例或业务功能和业务规则的所有测试需求。这些测试的目标在于核实能否正确地接受、处理和检索数据以及业务规则是否正确实施。这种类型的测试基于黑盒方法，即通过图形用户界面 (GUI) 与应用程序交互并分析输出结果来验证应用程序及其内部进程。以下列出的是每个应用程序推荐的测试方法概要：]

|  |  |
| --- | --- |
| 测试目标： | [确保测试对象的功能正常，其中包括导航、数据输入、处理和检索等。] |
| 方法： | [利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以下内容：  • 在使用有效数据时得到预期的结果。  • 在使用无效数据时显示相应的错误消息或警告消息。  • 各业务规则都得到了正确的应用。] |
| 完成标准： | • [所计划的测试已全部执行。  • 所发现的缺陷已全部解决。] |
| 需考虑的特殊事项： | [确定或说明那些将对功能测试的实施和执行造成影响的事项或因素（内部的或外部的）] |

### 性能评价

[性能评价是一种性能测试，它对响应时间、事务处理速率和其他与时间相关的需求进行评测和评估。性能评价的目标是核实性能需求是否都已满足。实施和执行性能评价的目的是将测试对象的性能行为当作条件（例如工作量或硬件配置）的一种函数来进行评价和微调。

注：以下事务均指“逻辑业务事务”。这种事务被定义为将由系统的某个主角通过使用测试对象来执行的特定用例，例如，添加或修改某个合同。]

|  |  |
| --- | --- |
| 测试目标： | [核实所指定的事务或业务功能在以下情况下的性能行为：  • 正常的预期工作量  • 预期的最繁重工作量] |
| 方法： | • [使用为功能或业务周期测试制定的测试过程。  • 通过修改数据文件来增加事务数量，或通过修改脚本来增加每项 事务的迭代次数。  • 脚本应该在一台计算机上运行（最好是以单个用户、单个事务为 基准），并在多台客户机（虚拟的或实际的客户机，请参见下面 的“需考虑的特殊事项”）上重复。] |
| 完成标准： | • [单个事务或单个用户：在每个事务所预期或要求的时间范围内 成功地完成测试脚本，没有发生任何故障。]  • [多个事务或多个用户：在可接受的时间范围内成功地完成测试 脚本，没有发生任何故障。] |
| 需考虑的特殊事项： | [综合的性能测试还包括在服务器上添加后台工作量。  可采用多种方法来执行此操作，其中包括：  • 直接将“事务强行分配到”服务器上，这通常以“结构化查询语 言”(SQL) 调用的形式来实现。  • 通过创建“虚拟的”用户负载来模拟许多个（通常为数百个）客 户机。 此负载可通过“远程终端仿真”(Remote Terminal Emulation) 工具来实现。 此技术还可用于在网络中加载“流 量”。  • 使用多台实际客户机（每台客户机都运行测试脚本）在系统上添 加负载。  性能测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。  性能测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。] |

### 负载测试

[负载测试是一种性能测试。在这种测试中，将使测试对象承担不同的工作量，以评测和评估测试对象在不同工作量条件下的性能行为，以及持续正常运行的能力。负载测试的目标是确定并确保系统在超出最大预期工作量的情况下仍能正常运行。此外，负载测试还要评估性能特征，例如，响应时间、事务处理速率和其他与时间相关的方面。]

[注：以下事务均指“逻辑业务事务”。这些事务被定义为将由系统的最终用户通过使用应用程序来执行的具体功能，例如，添加或修改某个合同。]

|  |  |
| --- | --- |
| 测试目标： | [核实所指定的事务或商业理由在不同的工作量条件下的性能行为时间。] |
| 方法： | • [使用为功能或业务周期测试制定的测试。  • 通过修改数据文件来增加事务数量，或通过修改测试来增加每项 事务发生的次数。] |
| 完成标准： | [多个事务或多个用户：在可接受的时间范围内成功地完成测试，没有发生任何故障。] |
| 需考虑的特殊事项： | • [负载测试应该在专用的计算机上或在专用的机时内执行，以便 实现完全的控制和精确的评测。  • 负载测试所用的数据库应该是与实际大小相同或等比例缩放的数 据库。] |

### 安全性和访问控制测试

[安全性和访问控制测试侧重于安全性的两个关键方面：

• 应用程序级别的安全性，包括对数据或业务功能的访问

• 系统级别的安全性，包括对系统的登录或远程访问。

应用程序级别的安全性可确保：在预期的安全性情况下，主角只能访问特定的功能或用例，或者只能访问有限的数据。例如，可能会允许所有人输入数据，创建新账户，但只有经理才能删除这些数据或账户。如果具有数据级别的安全性，测试就可确保“用户类型一”能够看到所有客户信息（包括财务数据），而“用户二”只能看见同一客户的统计数据。

系统级别的安全性可确保只有具备系统访问权限的用户才能访问应用程序，而且只能通过相应的网关来访问。]

|  |  |
| --- | --- |
| 测试目标： | * 应用程序级别的安全性：[*核实主角只能访问其所属用户类型已被授权使用的那些功能或数据。]* * 系统级别的安全性：*核实只有具备系统和应用程序访问权限的主角才能访问系统和应用程序。]* |
| 方法： | * 应用程序级别的安全性：[确定并列出各用户类型及其被授权使用的功能或数据。*]*   • [为各用户类型创建测试，并通过创建各用户类型所特有的事务来核实其权限。]  • 修改用户类型并为相同的用户重新运行测试。对于每种用户类型，确保正确地提供或拒绝了这些附加的功能或数据。   * 系统级别的访问：*[请参见下面的“需考虑的特殊事项”]* |
| 完成标准： | [各种已知的主角类型都可访问相应的功能或数据，而且所有事务都按照预期的方式运行，并在先前的应用程序功能测试中运行了所有的事务。] |
| 需考虑的特殊事项： | [必须与相应的网络或系统管理员一起对系统访问权进行检查和讨论。由于此测试可能是网络管理或系统管理的职能，可能不需要执行此测试。] |

### 配置测试

[配置测试核实测试对象在不同的软件和硬件配置中的运行情况。在大多数生产环境中，客户机工作站、网络连接和数据库服务器的具体硬件规格会有所不同。客户机工作站可能会安装不同的软件,例如，应用程序、驱动程序等。而且在任何时候，都可能运行许多不同的软件组合，从而占用不同的资源。]

|  |  |
| --- | --- |
| 测试目标： | [核实测试对象可在要求的硬件和软件配置中正常运行。] |
| 方法： | • [使用功能测试脚本。  • 在测试过程中或在测试开始之前，打开各种与非测试对 象相关的软件（例如 Microsoft 应用程序：Excel 和 Word），然后将其关闭。  • 执行所选的事务，以模拟主角与测试对象软件和非测试 对象软件之间的交互。  • 重复上述步骤，尽量减少客户机工作站上的常规可用内 存。] |
| 完成标准： | [对于测试对象软件和非测试对象软件的各种组合，所有事务都成功完成，没有出现任何故障。] |
| 需考虑的特殊事项： | • [需要、可以使用并可以通过桌面访问哪种非测试对象软 件？  • 通常使用的是哪些应用程序？  • 应用程序正在运行什么数据？例如，在Excel 中打开的 大型电子表格，或是在 Word 中打开的 100 页文档。  • 作为此测试的一部分，应将整个系统、Netware、网络服 务器、数据库等都记录下来。] |

### 

## 工具

此项目将使用以下工具：

[注：可以视情况删除或添加项目。]

|  |  |  |  |
| --- | --- | --- | --- |
|  | 工具 | 厂商/自行研制 | 版本 |
| 测试管理 |  |  |  |
| 缺陷跟踪 |  |  |  |
| 用于功能性测试的 ASQ 工具 |  |  |  |
| 用于性能测试的 ASQ 工具 |  |  |  |
| 测试覆盖监测器或评价器 |  |  |  |
| 项目管理 |  |  |  |
| DBMS 工具 |  |  |  |

# 资源

[本节列出推荐 <项目名称> 项目使用的资源，及其主要职责、知识或技能。]

## 角色

下表列出了在此项目的人员配备方面所作的各种假定。

[注：可视情况删除或添加项目。]

|  |  |  |
| --- | --- | --- |
| 人力资源 | | |
| 角色 | 推荐的最少资源  （所分配的专职角色数量） | 具体职责或注释 |
| 测试经理，  测试项目经理 |  | 进行管理监督。  职责：   * 提供技术指导 * 获取适当的资源 * 提供管理报告 |
| 测试设计员 |  | 确定测试用例、确定测试用例的优先级并实施测试用例。  职责：   * 生成测试计划 * 生成测试模型 * 评估测试工作的有效性 |
| 测试员 |  | 执行测试。  职责：   * 执行测试 * 记录结果 * 从错误中恢复 * 记录变更请求 |
| 测试系统管理员 |  | 确保测试环境和资产得到管理和维护。  职责：   * 管理测试系统 * 授予和管理角色对测试系统的访问权 |
| 数据库管理员 |  | 确保测试数据（数据库）环境和资产得到管理和维护。  职责：   * 管理测试数据（数据库） |
| 设计员 |  | 确定并定义测试类的操作、属性和关联。  职责：   * 确定并定义测试类 * 确定并定义测试包 |
| 实施员 |  | 实施测试类和测试包，并对它们进行单元测试。  职责：   * 创建在测试模型中实施的测试类和测试包 |

## 系统

下表列出了测试项目所需的系统资源。

[此时并不完全了解测试系统的具体元素。建议让系统模拟生产环境，并在适当的情况下减小访问量和数据库大小。]

[注：可以视情况删除或添加项目。]

|  |  |
| --- | --- |
| 系统资源 | |
| 资源 | 名称/类型 |
| 数据库服务器 |  |
| —网络或子网 | TBD |
| —服务器名服务器名 | TBD |
| —数据库名 | TBD |
| 客户端测试 PC |  |
| —包括特殊的配置需求 | TBD |
| 测试存储库 |  |
| —网络或子网 | TBD |
| —服务器名服务器名 | TBD |
| 测试开发 PC | TBD |

# 项目里程碑

[对 <项目名称> 的测试应包括上面各节所述的各项测试的测试活动。应该为这些测试确定单独的项目里程碑，以通知项目的状态和成果。]

|  |  |  |  |
| --- | --- | --- | --- |
| **里程碑任务** | **工作量** | **开始日期** | **结束日期** |
| 制定测试计划 |  |  |  |
| 设计测试 |  |  |  |
| 实施测试 |  |  |  |
| 执行测试 |  |  |  |
| 评估测试 |  |  |  |

# 可交付工件

[本节列出了将要创建的各种文档、工具和报告，及其创建人员、交付对象和交付时间。]

## 测试模型

[本节确定将要通过测试模型创建并分发的报告。测试模型中的这些工件应该用 ASQ 工具来创建或引用。]

## 测试日志

[说明用来记录和报告测试结果和测试状态的方法和工具。]

## 缺陷报告

[本节确定用来记录、跟踪和报告测试中发生的意外情况及其状态的方法和工具。]

# 附录 A：项目任务

以下是一些与测试有关的任务：

• 制定测试计划

* + 确定测试需求
  + 评估风险
  + 制定测试策略
  + 确定测试资源
  + 创建时间表
  + 生成测试计划

• 设计测试

- 准备工作量分析文档

- 确定并说明测试用例

- 确定并结构化测试过程

- 复审和评估测试覆盖

• 实施测试

* + 记录或通过编程创建测试脚本
  + 确定设计与实施模型中的测试专用功能
  + 建立外部数据集

• 执行测试

- 执行测试过程

- 评估测试的执行情况

- 恢复暂停的测试

- 核实结果

- 调查意外结果

- 记录缺陷

• 评估测试

- 评估测试用例覆盖

- 评估代码覆盖

- 分析缺陷

- 确定是否达到了测试完成标准与成功标准