|  |
| --- |
| 北京航空航天大学 |
| 测试规格说明书 |
| **Redis** |
|  |
| SY1406108 陈志伟 SY1406112 王珊珊 SY1406311 林 璐 SY1406117 王志鹏 |
| **2015/04/28** |

**版本变更历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 说明 |
| Version 1.0 | 2015-04-28 | 全体成员 | 初步完成测试规格说明书的第一版 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 范围 1](#_Toc417978633)

[1.1 标识 1](#_Toc417978634)

[1.2 编写目的 1](#_Toc417978635)

[1.3 测试职责 1](#_Toc417978636)

[1.3.1 职责 1](#_Toc417978637)

[1.3.2 职责划分 1](#_Toc417978638)

[1.4 文档概述 2](#_Toc417978639)

[2 测试规范及对照表 2](#_Toc417978640)

[2.1 编写测试计划 2](#_Toc417978641)

[2.2 设计测试用例 3](#_Toc417978642)

[2.3 测试用例与需求用例对照表 3](#_Toc417978643)

[3 服务器模块 4](#_Toc417978644)

[3.1 启动服务器测试 4](#_Toc417978645)

[3.1.1 测试策略描述 4](#_Toc417978646)

[3.1.2 测试用例 4](#_Toc417978647)

[3.1.3 测试结果及分析 5](#_Toc417978648)

[3.2 自定义服务器配置测试 6](#_Toc417978649)

[3.2.1 测试策略描述 6](#_Toc417978650)

[3.2.2 测试用例 6](#_Toc417978651)

[3.2.3 测试结果及分析 7](#_Toc417978652)

[3.3 服务器性能测试 8](#_Toc417978653)

[3.3.1 测试策略描述 8](#_Toc417978654)

[3.3.2 测试用例 8](#_Toc417978655)

[3.3.3 测试结果及分析 9](#_Toc417978656)

[4 RDB持久化模块 11](#_Toc417978657)

[4.1 同步回写SAVE测试 11](#_Toc417978658)

[4.1.1 测试策略描述 11](#_Toc417978659)

[4.1.2 测试用例 12](#_Toc417978660)

[4.2 启动服务器测试 13](#_Toc417978661)

[4.2.1 测试策略描述 13](#_Toc417978662)

[4.2.2 测试用例 13](#_Toc417978663)

[4.3 载入数据测试 14](#_Toc417978664)

[4.3.1 测试策略描述 14](#_Toc417978665)

[4.3.2 测试用例 14](#_Toc417978666)

[5 AOF持久化模块 15](#_Toc417978667)

[5.1 命令同步测试 15](#_Toc417978668)

[5.1.1 测试策略描述 15](#_Toc417978669)

[5.1.2 测试用例 15](#_Toc417978670)

[5.1.3 测试结果及分析 16](#_Toc417978671)

[5.2 缓存追加测试 17](#_Toc417978672)

[5.2.1 测试策略描述 17](#_Toc417978673)

[5.2.2 测试用例 17](#_Toc417978674)

[5.3 AOF文件读取和数据还原测试 18](#_Toc417978675)

[5.3.1 测试策略描述 18](#_Toc417978676)

[5.3.2 测试用例 18](#_Toc417978677)

[6 客户端模块 19](#_Toc417978678)

[6.1 用户命令处理测试 19](#_Toc417978679)

[6.1.1 测试策略描述 19](#_Toc417978680)

[6.1.2 测试用例 19](#_Toc417978681)

[6.2 命令请求读取测试 20](#_Toc417978682)

[6.2.1 测试策略描述 20](#_Toc417978683)

[6.2.2 测试用例 20](#_Toc417978684)

[6.3 命令请求执行测试 20](#_Toc417978685)

[6.3.1 测试策略描述 20](#_Toc417978686)

[6.3.2 测试用例 21](#_Toc417978687)

[6.4 命令请求回复测试 22](#_Toc417978688)

[6.4.1 测试策略描述 22](#_Toc417978689)

[6.4.2 测试用例 22](#_Toc417978690)

[7 参考文献 23](#_Toc417978691)

# 1 范围

## 1.1 标识

Redis版本号： redis-release-2.8.19

模 块 名 称： Redis服务器、客户端、RDB持久化、AOF持久化模块

测试规格版本：V1.0

## 1.2 编写目的

本文档主要明确了本次测试的主要内容和测试小组各成员的分工，并对测试用例进行了初步的设计。测试技术和策略等问题不在本文档描述范围内。

## 1.3 测试职责

### 1.3.1 职责

测试是软件开发过程中的重要组成部分，肩负着如下责任：

* 编写合理的测试计划，并与需求分析相对应；
* 编写覆盖率高的测试用例；
* 针对测试需求进行相关测试技术的研究；
* 认真仔细地实施测试工作，并提交测试报告；
* 撰写工作进度日志。

### 1.3.2 职责划分

|  |  |
| --- | --- |
| 成员 | 主要职责 |
| 陈志伟 | 根据服务器模块的相关需求及代码做测试需求分析； |
| 林璐 | 根据RDB持久化模块的相关需求及代码做测试需求分析； |
| 王珊珊 | 根据AOF持久化模块的相关需求及代码做测试需求分析； |
| 王志鹏 | 根据客户端模块的相关需求及代码做测试需求分析； |
| 全体成员 | 撰写测试规格说明书； |

## 1.4 文档概述

本文档主要说明的测试阶段的工作内容。首先叙述了本组对测试工作的分析过程，给出了决定撰写测试用例的需求用例与对应的测试用例的对照表，对每个测试用例按照RUCM4test的标准进行了说明。最后，本文对“非功能性需求”的“服务器性能测试”从效率的角度做了详细描述，给出了测试过程、测试数据和测试结果分析。

# 2 测试规范及对照表

## 2.1 编写测试计划

在进行软件测试前，周密的计划和合理的安排是必不可少的。测试计划的制定是要建立在充分理解需求的基础上的，是测试的起始步骤和重要环

|  |  |
| --- | --- |
| 过程要点 | 详细说明 |
| 启动条件 | 需求分析文档终稿的完成； |
| 工作内容 | 根据项目的需求分析文档，测试文档中应该至少包括以下关键内容：   * 根据已有的需求分析文档终稿，明确相应的测试用例，包括四大部分的测试用例模块。 * 具体设计各大模块的测试用例，设计完成后进行评审工作。 * 根据评审后的测试用例编写测试代码 * 部署环境，具体测试各模块功能，记录测试数据及结果。 * 结合需求文档终稿，对测试结果进行分析。 * 完成整个测试需求文档，由评审组完成最终评审。   测试计划编写完毕后，必须提交给项目组全体成员，并由项目组组中各成员联合评审。 |
| 退出标准 | * 测试获得预期成果。 * 测试由评审组评审通过. |
| 责任人 | 全体成员 |

## 2.2 设计测试用例

一个测试用例描述了针对某个目标对程序进行测试所采用的一组实际输入、程序执行条件、测试步骤和预期的输出，以核实某个程序或其中的特定路径是否满足特定需求。在需求分析文档确立基线以后，需编写项目的测试用例，具体的任务和责任人如下：

|  |  |
| --- | --- |
| 过程要点 | 详细说明 |
| 输入条件 | 测试需求明确，测试计划明确 |
| 工作内容 | 根据每一步测试计划编写全部的测试用例 |
| 退出标准 | 测试用例需要覆盖大部分测试需求 |
| 责任人 | 全体组员 |

## 2.3 测试用例与需求用例对照表

本次测试中，小组挑选了各模块中核心及具有代表性需求用例进行测试用例的设计。测试用例与需求用例的对应表如下：

|  |  |  |
| --- | --- | --- |
| 模块 | 需求用例 | 测试用例 |
| 服务器 | 启动服务器 | 启动服务器测试 |
| 自定义服务器配置 | 自定义服务器配置测试 |
| 处理事件 | —— |
| 高效率 | 服务器性能测试 |
| RDB持久化 | 保存快照 | —— |
| 同步回写SAVE | 同步回写SAVE测试 |
| 异步回写BGSAVE | 异步会写BGSAVE测试 |
| 载入数据 | 载入数据测试 |
| AOF持久化 | 命令同步 | 命令同步测试 |
| 命令传播 | —— |
| 缓存追加 | 缓存追加测试 |
| AOF文件读取和数据还原 | AOF文件读取和数据还原测试 |
| 客户端 | 用户命令处理 | 用户命令处理测试 |
| 读取命令请求 | 命令请求读取测试 |
| 执行命令请求 | 命令请求执行测试 |
| 回复命令请求 | 命令请求回复测试 |

注：“——”表示不进行此项测试；

# 3 服务器模块

## 3.1 启动服务器测试

### 3.1.1 测试策略描述

为了测试服务器是否正常启动，计划编写脚本分别测试服务器正常启动和异常启动两种情况。正常情况下，会直接输出服务器返回的ping值，并进行简单的字符串存储操作；异常情况下，捕获“ConnectException”异常并抛出此异常。

### 3.1.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 启动服务器测试 | |
| Brief Description | 测试Redis服务器是否正常启动 | |
| Precondition | 系统已安装Redis数据库 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 准备Java测试脚本 |
| Description | 测试员编写好用于测试的Java源文件 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 测试员在eclipse中配置好编写Redis测试程序所需的jar包； |
| 2 | 测试员编写Java测试源文件； |
| 3 | 测试员通过人工走查方式检查源文件； |
| Postcondition  (Test Oracle) | 指定目录的Java测试源文件已存在；  Java测试代码没有逻辑和语法错误； |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 测试员在eclipse中编译Java测试程序； |
| 2 | 测试员在eclipse中运行Java测试程序； |
| 3 | 测试程序使用指定的IP地址连接redis服务器； |
| 4 | 服务器返回ping值； |
| 5 | 测试程序VALIDATES THAT服务器正常启动； |
| 6 | 测试程序执行简单的字符串存储操作； |
| Postcondition  (Test Oracle) | 测试程序输出服务器正常运行的提示；  测试程序输出存储的字符串；  服务器正常启动的情况被测试； |
| Specific Alternative Flows  (Test Sequence) | RFS 5 | |
| 1 | 测试程序抛出服务器连接异常； |
| 2 | ABORT |
| Postcondition  (Test Sequence) | 服务器未启动的情况被测试； |

### 3.1.3 测试结果及分析

（1）测试代码

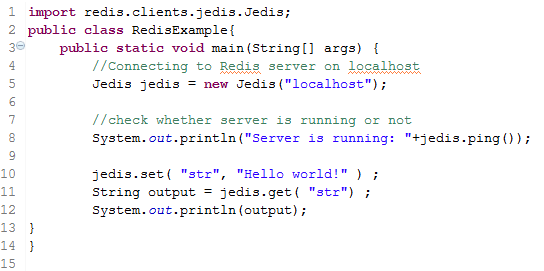


图 1 启动服务器测试代码

（2）测试结果

* 服务器正常启动

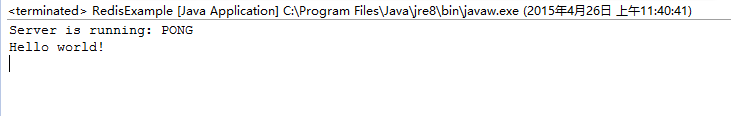


图 2 正常启动的结果图

测试程序输出了服务器端返回的ping值，同时完成了一个简单的字符串存储和查询操作，证明了服务器正常启动且正在运行。

* 服务器启动异常

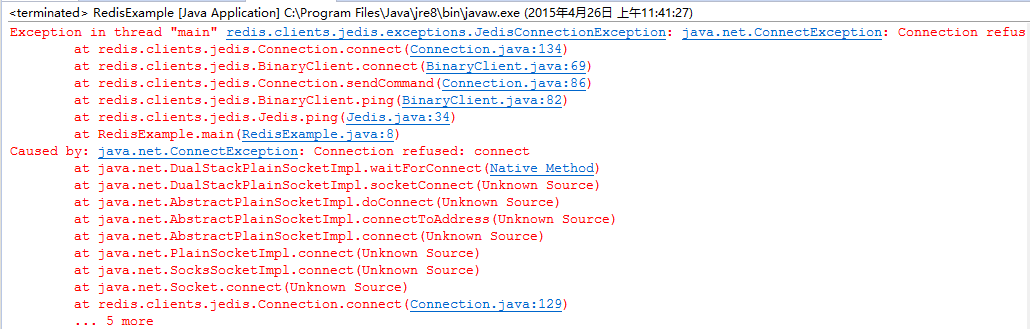


图 3 异常启动的结果图

服务器未启动时，客户端连接服务器且进行操作时，会产生连接的异常。

## 3.2 自定义服务器配置测试

### 3.2.1 测试策略描述

由于Redis配置文件中的参数较多，包括daemonize、pidfile、bind等。为了减少测试的难度，我们并不进行所有参数的测试，而是选取了一个关键的参数，即：maxheap，用来表示堆栈的最大内存，它关系到redis能否正常启动。通过在自定义的文件中，修改maxheap的不同取值，得到服务器在这些参数下的运行结果，从而完成测试。

### 3.2.2 测试用例

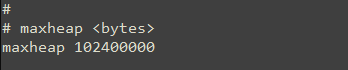
|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 自定义服务器配置测试 | |
| Brief Description | 测试在用户指定的配置下对服务器的影响 | |
| Precondition | 系统已安装Redis数据库 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 准备自定义的配置文件 |
| Description | 测试员编写好用于测试的自定义配置文件 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 测试员编写Redis的配置文件； |
| 2 | 测试员将配置文件保存到指定目录； |
| 3 | 测试员通过人工走查方式检查配置文件； |
| Postcondition  (Test Oracle) | 指定目录的自定义配置文件已存在；  配置文件符合Redis的配置文件相关规范； |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 测试员在命令行中输入服务器可执行程序名和自定义的配置文件名； |
| 2 | 服务器读入配置文件中的相关配置参数的值； |
| 3 | 服务器VALIDATES THAT配置文件中的参数有效； |
| 4 | 服务器按配置文件中的参数的值设置自身的参数； |
| Postcondition  (Test Oracle) | 服务器开始运行，在命令行中输出启动界面；  配置文件中的参数有效的情况被测试； |
| Specific Alternative Flows  (Test Sequence) | RFS 3 | |
| 1 | 服务器启动失败并给出配置文件中参数设置错误； |
| 2 | ABORT |
| Postcondition  (Test Oracle) | 配置文件中参数设置错误的情况被测试； |

图 10 读取命令请求用例规格

### 3.2.3 测试结果及分析

（1）测试代码

* 自定义配置文件（显示将改变的参数）



* 测试指令



（2）测试结果

* 参数有效情况

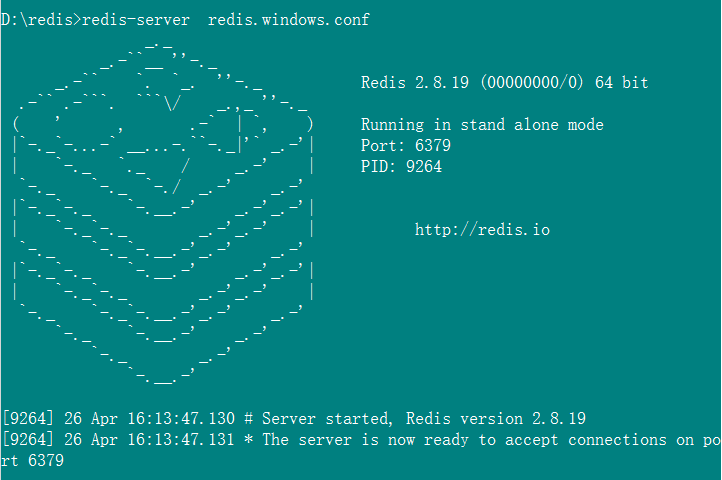


图 4 参数有效结果图

这里是把maxheap设置为一个较小的值，即102,400,000byte所得到的结果，可以看到当给堆栈的最大内存分配一个合适的值时，服务器可以正常启动。

* 测试设置错误

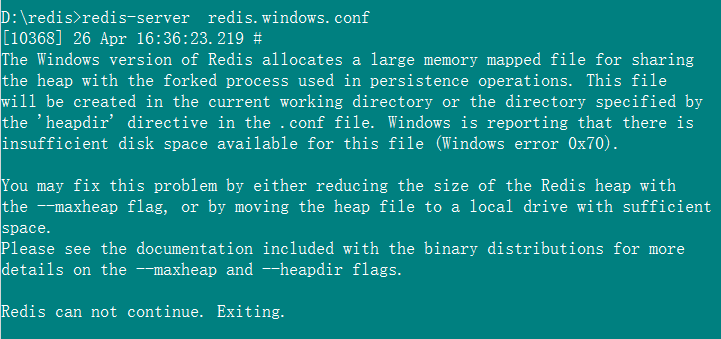


图 5 参数错误结果图

这里是把maxheap设置为一个较大的值，即10,240,000,000bytes时所得到的结果，由于堆栈的最大内存分配过大，导致磁盘没有足够的容纳空间，所以服务器启动失败。

## 3.3 服务器性能测试

### 3.3.1 测试策略描述

在Redis服务器性能时测试中，我们选择的是在请求数量一定的情况下，研究不同的数量的客户端，即并发数对服务器的性能的影响。通过把请求数量设置为恒定的100000次，并按一定的规则改变并发数，从而分别得到完成所有请求所用的时间，就可推算出服务器的性能，即每秒所能处理的请求的个数。

### 3.3.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 服务器性能测试 | |
| Brief Description | 模拟测试不同的并发数对服务器的性能影响 | |
| Precondition | 服务器已正常启动 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 打开命令行 |
| Description | 打开命令行界面以启动redis-benchmark.exe性能测试工具 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 测试员启动操作系统中的命令行界面； |
| 2 | 测试员进入redis-benchmark.exe的安装目录； |
| Postcondition  (Test Oracle) | 命令行已经启动，处于等待输入命令的状态 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 测试员在命令行调用性能测试工具向服务器以不同的并发数发送恒定数量的请求； |
| 2 | 测试员改变并发数，重复第一步； |
| 3 | 测试员记录测试数据； |
| 4 | 测试员对测试数据进行分析； |
| Postcondition  (Test Oracle) | 在请求数量一定的情况下，服务器在不同的并发数下的性能得到测试 |

### 3.3.3 测试结果及分析

（1）测试指令



其中，“-c”后面的50表示测试的客户端数量，可改为不同的值。

（2）测试结果

* 当并发数为50的测试结果

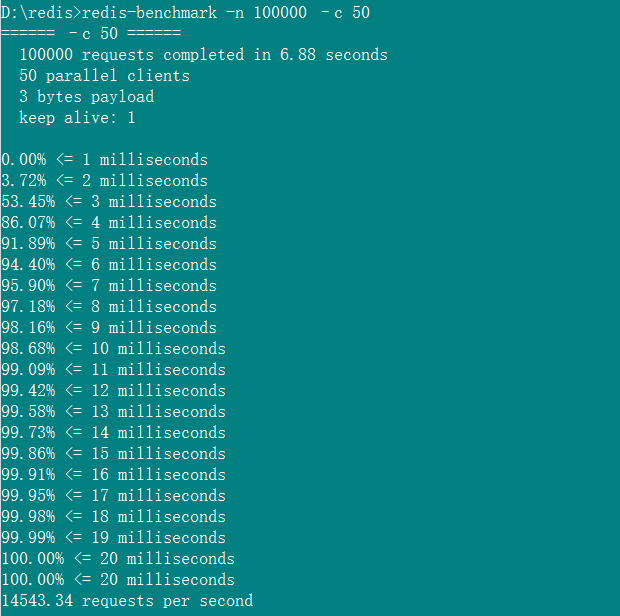


图 6 并发数为50的测试结果结果图

图中表示100000在6.88秒内完成，50个并发客户端，3字节有效载荷。

* 不同并发数下的测试结果

图 7 Redis性能图

具体数据：

|  |  |
| --- | --- |
| 并发数（个） | 每秒请求次数（requests per second） |
| 10 | 14359.56 |
| 20 | 14617.75 |
| 30 | 14480.16 |
| 40 | 14096.42 |
| 50 | 14543.34 |
| 60 | 14622.02 |
| 70 | 14887.60 |
| 80 | 14705.88 |
| 90 | 14634.86 |
| 150 | 14747.09 |
| 180 | 14579.38 |
| 200 | 14581.51 |
| 300 | 14154.28 |
| 500 | 14983.52 |
| 1000 | 14624.16 |

由上述的图表可知，在测试的并发数范围10～1000内，Redis服务器的性能恒定在每秒处理14000～15000个请求的范围内，即Redis服务器并未随着并发数的上升性能会下降，而是保持相对稳定的高性能。

# 4 RDB持久化模块

## 4.1 同步回写SAVE测试

### 4.1.1 测试策略描述

同步回写SAVE在Redis主进程中直接调用rdbSave()函数，阻塞主进程，直到保存完成为止。在主进程阻塞期间，服务器不能处理客户端任何请求。

一般来说，在生产环境中很少用SAVE操作，因为会阻塞所有客户端的请求，保存数据库的任务通常由BGSAVE命令异步保存。然而，如果负责保存数据的后台子进程不幸出现问题时，SAVE可以作为保存数据的最后手段来使用。

本部分测试在window环境下，当Redis开启rdb方式，并配置m秒时间内发生n次变更进行rdb文件保存。同时，验证其主进程被阻塞时，服务器发送异常信息的功能。

### 4.1.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 同步回写SAVE测试 | |
| Brief Description | 测试能否正确读取用户配置命令，服务器进行同步回写SAVE操作 | |
| Precondition | Redis在windows下正确安装 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 配置redis.conf文件 |
| Description | 配置redis.conf文件，以选择RDB持久化方式，并调用SAVE命令 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 测试员打开redis.conf文件 |
| 2 | 测试员写入命令”**save 300 10**”，设置快照保存的策略为300秒内发生10次变更则进行保存 |
| 3 | 写入命令”dbfilename dump.rdb”，设置快照保存文件的文件名 |
| 4 | 写入命令”dir ./”，设置备份文件放置路径 |
| 5 | 写入命令”appendonly no”，选择非AOF方式，也即RDB方式 |
| Postcondition  (Test Oracle) | 配置完成，准备启动Redis进行快照SAVE测试 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 打开windows下的cmd运行窗口 |
| 2 | 输入”cd PATH-TO-REDIS”，切换到Redis文件夹下 |
| 3 | 输入”redis-server.exe redis.conf”，启动Redis |
| 4 | 输入”redis-cli.exe”，打开一个窗口运行客户端A |
| 5 | 创建Eclipse项目，引入jedis客户端包 |
| 6 | 编写测试程序A-test，实现每秒修改一次数据，运行5分钟 |
| 7 | 在测试程序运行期间，输入”redis-cli.exe”，打开另一个窗口运行客户端B |
| 8 | 在客户端B发送命令”get xxx” VALIDATES THAT Redis服务器主进程未被阻塞 |
| 9 | 测试程序A-test运行完成，在客户端A输入命令”LASTSAVE”，获取最近一次成功保存数据的时间 |
| Postcondition  (Test Oracle) | 最近的保存时间恰是测试程序A-test执行结束时间，同步保存SAVE的每300秒发生10次变更则保存的功能被测试 |
| Specific Alternative Flows  (Test Sequence) | RFS 1 | |
| 1 | Redis服务器向客户端发送错误信息，告知在SAVE执行期间，不能处理其他命令 |
| Postcondition  (Test Sequence) | SAVE执行期间，主进程被阻塞的特点被测试 |

## 4.2 启动服务器测试

### 4.2.1 测试策略描述

异步回写BGSAVE与SAVE的不同点在于，主进程会fork出一个子进程，子进程负责调用rdbSave()函数，并在保存完成后向主进程发送信号，通知保存已完成。因为rdbSave()执行期间，是被子进程调用的，所以Redis服务器在BGSAVE期间仍可以继续处理客户端的请求。

本部分测试在window环境下，当Redis开启rdb方式，并配置m秒时间内发生n次变更进行rdb文件保存。同时，验证其主进程被阻塞时，服务器仍能接受客户端命令，但是拒绝处理同时调用的SAVE或BGSAVE命令以避免竞争。

### 4.2.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 异步回写BGSAVE测试 | |
| Brief Description | 测试能否正确读取用户配置命令，服务器进行异步回写BGSAVE操作 | |
| Precondition | Redis在windows下正确安装 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 配置redis.conf文件 |
| Description | 配置redis.conf文件，以选择RDB持久化方式，并调用BGSAVE命令 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 测试员打开redis.conf文件 |
| 2 | 测试员写入命令”**bgsave 300 10**”，设置快照保存的策略为300秒内发生10次变更则进行保存 |
| 3 | 写入命令”dbfilename dump.rdb”，设置快照保存文件的文件名 |
| 4 | 写入命令”dir ./”，设置备份文件放置路径 |
| 5 | 写入命令”appendonly no”，选择非AOF方式，也即RDB方式 |
| Postcondition  (Test Oracle) | 配置完成，准备启动Redis进行快照BGSAVE测试 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 打开windows下的cmd运行窗口 |
| 2 | 输入”cd PATH-TO-REDIS”，切换到Redis文件夹下 |
| 3 | 输入”redis-server.exe redis.conf”，启动Redis |
| 4 | 输入”redis-cli.exe”，打开一个窗口运行客户端A |
| 5 | 创建Eclipse项目，引入jedis客户端包 |
| 6 | 编写测试程序A-test，实现每秒修改一次数据，运行5分钟 |
| 7 | 在测试程序运行期间，输入”redis-cli.exe”，打开另一个窗口运行客户端B |
| 8 | 在客户端B发送命令”get xxx” |
|  | 在客户端B发送命令”bgsave” VALIDATES THAT Redis服务器主进程未被阻塞 |
| 9 | 测试程序A-test运行完成，在客户端A输入命令”LASTSAVE”，获取最近一次成功保存数据的时间 |
| Postcondition  (Test Oracle) | 最近的保存时间恰是测试程序A-test执行结束时间，同步保存SAVE的每300秒发生10次变更则保存的功能被测试 |
| Specific Alternative Flows  (Test Sequence) | RFS 1 | |
| 1 | Redis服务器向客户端发送错误信息，告知在其他SAVE或BGSAVE执行期间，不能执行bgsave命令 |
| Postcondition  (Test Sequence) | save或bgsave命令同时调用造成竞争的问题被避免 |

## 4.3 载入数据测试

### 4.3.1 测试策略描述

当Redis服务器启动时，执行载入函数rdbLoad，读取备份.rdb文件，并将文件中的数据库数据载入到内存中。

本部分测试当修改数据键值，并手动调用save命令将更改备份后，服务器关闭又重启后，是否能正确载入备份的数据到内存。

### 4.3.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 载入数据测试 | |
| Brief Description | 测试Redis服务器是否能正确从.rdb备份文件中恢复数据到内存中 | |
| Precondition | Redis服务器已启动，正在运行并未断电 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 修改数据库值 |
| Description | 修改某个数据库的值，以验证修改能被保存和恢复 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 测试员在运行窗口输入”redis-cli.exe”，打开一个窗口运行客户端A |
| 2 | 输入”set test1 testLoad”，设置一个数据值 |
| 3 | 输入”save”，保存数据快照 |
| 4 | 输入”get test1”，验证该值已被修改 |
| 5 | 输入”shutdown”，停止Redis服务器 |
| Postcondition  (Test Oracle) | 数据库的一个值被修改 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 输入“redis-server.exe redis.conf”，重启服务器 |
| 2 | 输入”redis-cli.exe”，再启动客户端 |
| 3 | 输入”get test1”，获取之前被修改的数据 |
| Postcondition  (Test Oracle) | 数据为testLoad，说明服务器重启后，重新从备份文件载入数据到内存成功 |

# 5 AOF持久化模块

## 5.1 命令同步测试

### 5.1.1 测试策略描述

以AOF持久化方式打开Redis服务器，在Redis客户端输入数据库修改写入命令，通过查看AOF文件，命令是否正确的以AOF网络通讯协议的格式保存下来,来判断Redis是否能够实现命令同步功能。

### 5.1.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 命令同步测试 | |
| Brief Description | 测试Redis将所有对数据库进行过写入的命令（及其参数）记录到 AOF文件中 | |
| Precondition | Redis服务器与客户端正常运行 | |
| Tester | None | |
| Dependency | |  |  |  |  | | --- | --- | --- | --- | | |  | | --- | | **INCLUDE TEST CASE SPECIFICATION** 缓存追加测试 | | **INCLUDE TEST CASE SPECIFICATION** 文件写入和保存测试 | |  | | | |
| Test Setup | Name | 准备命令并打开Redis服务器和客户端 |
| Description | 准备好向Redis写入的命令并以aof方式打开Redis服务器，打开Redis客户端等待命令写入 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 以管理员的方式打开windows下的cmd运行窗口 |
| 2 | 切换到Redis/src/bin文件夹下 |
| 3 | 输入”redis-server.exe --appendonly yes”启动命令参数，启动Redis服务器并且开启aof功能 |
| 4 | 输入”redis-cli.exe”，打开一个窗口运行客户端 |
| 5 | 准备好写入数据库的命令 |
| Postcondition  (Test Oracle) | 客户端命令行已经打开，处于等待数据库写入命令状态 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 在客户端命令行里输入已经准备好的数据库写入命令 |
| 2 | 切换到Redis/src/bin文件夹下 |
| 3 | 输入”cat appendonly.aof” 查看AOF文件 |
| Postcondition  (Test Oracle) | 命令输入完成后，查看AOF文件命令被以aof网络通讯协议的格式保存下来，测试通过 |

### 5.1.3 测试结果及分析

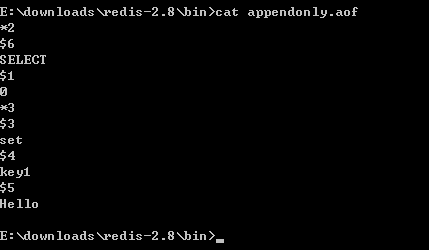
（1）打开Redis服务器



（2）写入命令



（3）查看appendonly.aof文件



## 5.2 缓存追加测试

### 5.2.1 测试策略描述

由于aof\_buf并没有一个直接的查看文件，只有通过重新修改重新编译源代码将缓存打印出来查看缓存的内容是否对应命令的AOF网络通讯协议的格式。

### 5.2.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 缓存追加测试 | |
| Brief Description | 测试Redis将所有对数据库进行过写入的命令（及其参数）经过AOF程序后缓存追加到 aof\_buf文件中 | |
| Precondition | Redis服务器与客户端正常运行 | |
| Tester | None | |
| Dependency | |  |  |  |  | | --- | --- | --- | --- | | |  | | --- | | **INCLUDE TEST CASE SPECIFICATION** 命令同步测试 | |  | |  | | | |
| Test Setup | Name | 编译aof.c |
| Description | 使用VS编译工具编译aof.c |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 打开VS工具 |
| 2 | 将源文件导入VS中 |
| 3 | 编辑aof.c，加入print server.aof\_buf 将缓冲区的内容打印出来 |
| 4 | VS重新编译代码 |
| Postcondition  (Test Oracle) | 源文件已经被编译成功，等待执行 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 切换到Redis/src/bin文件夹下 |
| 2 | 打开redis\_cli.exe |
| 3 | 输入数据库写入命令 |
| 4 | 查看打印出来的aof\_buf的内容 |
| Postcondition  (Test Oracle) | 看见打印出来的aof\_buf的内容与写入的数据库命令的AOF网络通讯协议的格式相同，即测试通过 |

## 5.3 AOF文件读取和数据还原测试

### 5.3.1 测试策略描述

通过将输入的命令和从AOF文件还原出的命令对比，可以测试数据还原功能。

### 5.3.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | AOF文件读取和数据还原测试 | |
| Brief Description | 测试读取AOF文件，并且将AOF网络通讯协议的格式还原为数据库写入命令，再执行该命令还原数据库的状态 | |
| Precondition | Redis服务器与客户端正常运行 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 打开服务器和客户端 |
| Description | 打开服务器和客户端命令行 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 以管理员的方式打开windows下的cmd运行窗口 |
| 2 | 切换到Redis/src/bin文件夹下 |
| 3 | 输入”redis-server.exe --appendonly yes”启动命令参数，启动Redis服务器并且开启aof功能 |
| 4 | 输入”redis-cli.exe”，打开一个窗口运行客户端 |
| Postcondition  (Test Oracle) | 命令行被打开，等待输入命令 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 准备一段Redis写入命令自己单独记录 |
| 2 | 打开客户端，输入数据库命令 |
| 3 | 读取AOF文件 |
| 4 | 根据还原程序还原AOF文件为数据库命令 |
| Postcondition  (Test Oracle) | 对比开始写入的命令和从AOF文件还原出来的数据库命令，若两者相等则测试通过 |

# 6 客户端模块

## 6.1 用户命令处理测试

### 6.1.1 测试策略描述

在 redis 客户端中输入一系列正确的命令请求和异常的命令请求，分别查看redis客户端保存的协议格式，判断客户端是否能将各种命令请求正确转换成协议格式。

### 6.1.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 用户命令请求处理测试 | |
| Brief Description | 测试客户端收到来自用户的命令请求，并将其转换为协议格式 | |
| Precondition | Redis服务器正在运行 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 打开命令行 |
| Description | 打开命令行界面以输入命令请求 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 打开操作系统的命令行界面 |
| 2 | 打开命令行，进入redis源码路径 |
| Postcondition  (Test Oracle) | 命令行已经打开，处于等待启动客户端状态 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 测试员进入命令行 |
| 2 | 测试员输入命令”cd redis/src” 进入redis目录 |
| 3 | 输入”./redis-cli” 启动redis客户端 |
| 4 | 输入“set key value ”发送命令请求 |
| 5 | 输入“get key” 发送命令请求 |
| 6 | 输入一系列错误命令请求 |
| 7 | 测试员查看客户端保存的命令协议格式 |
| Postcondition  (Test Oracle) | 命令输入完成，客户端保存的命令协议格式与预估格式相同，测试通过 |

## 6.2 命令请求读取测试

### 6.2.1 测试策略描述

在 redis 客户端中输入一系列正确的命令请求和异常的命令请求，分别查看redis客户端保存的args 和 argv属性，判断redis服务器是否正确读取并分析了redis客户端发送过来的协议格式。

### 6.2.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 命令请求读取测试 | |
| Brief Description | 测试服务器是否能正确读取客户端的请求 | |
| Precondition | Redis服务器与客户端正在运行 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 打开命令行 |
| Description | 打开命令行界面以输入命令请求 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 打开操作系统的命令行界面 |
| 2 | 打开命令行，进入redis源码路径 |
| Postcondition  (Test Oracle) | 命令行已经打开，处于等待输入命令状态 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 测试员进入命令行 |
| 2 | 测试员输入命令“cd redis/src” 进入redis目录 |
| 3 | 输入”./redis-cli” 启动redis客户端 |
| 4 | 输入“set key value ”发送命令请求 |
| 5 | 输入“get key” 发送命令请求 |
| 6 | 输入一系列错误命令请求 |
| 7 | 测试员查看客户端状态的argv和argc属性 |
| Postcondition  (Test Oracle) | 命令输入完成，客户端状态中的argv与argc属性被成功保存，测试通过 |

## 6.3 命令请求执行测试

### 6.3.1 测试策略描述

在 redis 客户端中输入一系列正确的命令请求和异常的命令请求，分别查看redis服务器保存在客户端的输出缓冲区里的命令执行结果，判断在命令请求异常的情况下，命令执行器能否成功识别；在命令请求正确的情况下，命令执行器能否正确执行命令并产生结果。

### 6.3.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 命令请求执行测试 | |
| Brief Description | 测试命令执行器是否可以正确执行命令请求 | |
| Precondition | Redis服务器与客户端正常运行 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 打开命令行 |
| Description | 打开命令行界面以输入命令请求 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 打开操作系统的命令行界面 |
| 2 | 打开命令行，进入redis源码路径 |
| Postcondition  (Test Oracle) | 命令行已经打开，处于等待输入命令状态 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 测试员进入命令行 |
| 2 | 测试员输入命令“cd redis/src” 进入redis目录 |
| 3 | 输入”./redis-cli” 启动redis客户端 |
| 4 | 输入“set key value ”发送命令请求 |
| 5 | 输入“get key” 发送命令请求 |
| 6 | 输入一系列错误命令请求 |
| 7 | 命令执行器 VALIDATE THAT 客户端状态不为null |
| 8 | 命令执行器 VALIDATE THAT 参数个数正确 |
| 9 | 命令执行器 VALIDATE THAT 客户端通过身份验证 |
| 10 | 测试员查看客户端状态的输出缓冲区的命令结果 |
| Postcondition  (Test Oracle) | 命令输入完成，客户端命令被成功执行，并保存到客户端状态的输出缓冲区 |
| Specific Alternative  Flows  (Test Sequence) | RFS 1 | |
| 1 | 命令执行器返回客户端状态为null的错误信息 |
| Postcondition  (Test Oracle) | 命令请求使客户端状态为null 的情况被测试 |
| RFS 2 | |
| 1 | 命令执行器返回参数个数不正确的错误信息 |
| Postcondition  (Test Oracle) | 参数个数不正确的命令请求被测试 |
| RFS 3 | |
| 1 | 命令执行器返回客户端未通过身份验证的错误信息 |
| Postcondition  (Test Oracle) | 客户端未通过验证的请求被测试 |

## 6.4 命令请求回复测试

### 6.4.1 测试策略描述

在 redis 客户端中输入一系列正确的命令请求和异常的命令请求，分别查看命令行中打印出的命令回复结果，判断命令请求是否被成功回复。

### 6.4.2 测试用例

|  |  |  |
| --- | --- | --- |
| **Test Case Specification** | | |
| Name | 命令请求回复测试 | |
| Brief Description | 测试服务器是否能正确回复客户端的请求 | |
| Precondition | Redis服务器与客户端正在运行 | |
| Tester | None | |
| Dependency | None | |
| Test Setup | Name | 打开命令行 |
| Description | 打开命令行界面以输入命令请求 |
| Basic Flow  (Test Setup) | Steps | |
| 1 | 打开操作系统的命令行界面 |
| 2 | 打开命令行，进入redis源码路径 |
| Postcondition  (Test Oracle) | 命令行已经打开，处于等待输入命令状态 |
| Basic Flow  (Test Sequence) | Steps | |
| 1 | 测试员进入命令行 |
| 2 | 测试员输入命令“cd redis/src” 进入redis目录 |
| 3 | 输入”./redis-cli” 启动redis客户端 |
| 4 | 输入“set key value ”发送命令请求 |
| 5 | 输入“get key” 发送命令请求 |
| 6 | 输入一系列错误命令请求 |
| 7 | 查看命令行的输出结果 |
| Postcondition  (Test Oracle) | 命令输入完成，命令行中显示服务器返回的输出结果，测试通过。 |

# 7 参考文献

[1] http://redis.io/

[2] 黄健宏. Redis设计与实现. 机械工业出版社[M]. 2014-06.

[3] 肖丁、吴建林等. 软件工程模型与方法[M]. 北京邮电大学出版社. 2008-03.

[4] http://blog.csdn.net/renfufei/article/details/38474435

[5] 蔡建平. 软件测试实验指导教程[M]. 清华大学出版社. 2009-11.