**内部文件**

**秘密**

**技术预研**

**灾难备份与恢复系统平台代码文档**

**成都世纪顶点科技有限公司**

**2017.11.02**

**版本情况**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **V1.0** | **V1.1** |  |
| **发布日期** | 2017.11.02 | 2017.11.16 |  |
| **负责人** | 李帅 | 李帅 |  |
| **小组成员：** | 李帅 | 李帅 |  |
| **审批人** | 严涛 | 严涛，周计 |  |

**目 录**

[第1章 DrcClient流程分析 5](#_Toc497386454)

[1.1 入口文件和main函数 5](#_Toc497386455)

[1.2 main函数执行的操作和该函数的流程图 5](#_Toc497386456)

[1.3 分析ClientApp::run函数中做的事情 5](#_Toc497386457)

[1.4 分析sendKeepAliveMessage如何KeepAlive 5](#_Toc497386458)

[1.5 LisenceHandller的作用和在程序中的流程，并详细分析LisenceHandller::run() 5](#_Toc497386459)

[1.6 详细分析LisenceHandller::handleModuleInfo函数 6](#_Toc497386460)

[1.7 详细分析TEST\_MODULE\_ACTIVATE宏 6](#_Toc497386461)

[1.8 startClientListner的作用，为什么要监听 6](#_Toc497386462)

[1.9 clientInit的作用 6](#_Toc497386463)

[1.10 客户端的注册上回调函数ClentCallback流程 6](#_Toc497386464)

[1.11 客户端网络离线后又重新上线，是如何实现的 6](#_Toc497386465)

[1.12 客户端一旦运行后，启动了哪些线程，线程的作用，线程的变化 7](#_Toc497386466)

[1.13 UUID如何生成的 7](#_Toc497386467)

[1.14 UUID是否固定不变，仅仅与硬件相关还是随机生成 7](#_Toc497386468)

[1.15 客户端配置完成连接上灾备机后，向灾备机发送了哪些信息 7](#_Toc497386469)

[1.16 灾备机如何处理新添加的客户端 7](#_Toc497386470)

[1.17 客户端如何处理自己被灾备机上人为删除的 7](#_Toc497386471)

[1.18 MsgHandller线程的作用和流程 8](#_Toc497386472)

[1.19 ClientCallbackI提供的功能有哪些 8](#_Toc497386473)

[第2章 DrcServer流程分析 9](#_Toc497386474)

[2.1 入口文件和main函数 9](#_Toc497386475)

[2.2 Main函数流程图和执行的操作 9](#_Toc497386476)

[2.3 DrcServer启动了哪些线程 9](#_Toc497386477)

[2.4 DRCService::run中执行的操作（监听和注册回调） 9](#_Toc497386478)

[2.5 授权部分 9](#_Toc497386479)

[2.5.1 用户操作对Oracle授权时发生的总体流程概要 9](#_Toc497386480)

[2.5.2 用户点击向页面发送的请求详细字段内容，对应字段作用等 9](#_Toc497386481)

[2.5.3 每种授权对应的代码 10](#_Toc497386482)

[2.5.4 授权写入数据库的信息是如何安排的 10](#_Toc497386483)

[2.5.5 授权信息是如何发送到DrcClient的 10](#_Toc497386484)

[2.5.6 授权信息如何组织之后传递到客户端 10](#_Toc497386485)

[2.5.7 客户端如何检测自己已经授权，如何启动Oracle模块的 10](#_Toc497386486)

[2.6 灾备机如何检测到客户端的在线，离线状态（ClientCleaner::run） 10](#_Toc497386487)

[2.7 TaskControll回调是在哪里注册的 10](#_Toc497386488)

[2.8 TaskControll提供的哪些功能（列表和简要功能介绍） 10](#_Toc497386489)

[2.9 日志线程 11](#_Toc497386490)

[2.9.1 分析JobLogUpdater::run()流程 11](#_Toc497386491)

[2.9.2 分析SystemLogUpdater::run()流程 11](#_Toc497386492)

[2.9.3 分析上述日志队列的实现流程 11](#_Toc497386493)

[2.9.4 分析TaskControlI::testJobLogCode中日志是如何被过滤的 11](#_Toc497386494)

[第3章 新模块授权添加概要设计 12](#_Toc497386495)

[3.1 参照上述流程分析，简述添加OracleRAC模块，在原系统上如何修改，哪些地方需要修改，模块的定义是多少，哪些函数需要修改编码，对外提供给PHP调用的脚本如何修改，客户端如何检测到并写入正确的标志 12](#_Toc497386496)

[第4章 其它 13](#_Toc497386497)

[4.1 DrcClient和DrcServer的编译步骤 13](#_Toc497386498)

[4.1.1 需要哪些组件/依赖才能编译 13](#_Toc497386499)

[4.1.2 编译的操作系统环境 13](#_Toc497386500)

[第5章 对此次研究的总结 14](#_Toc497386501)

DrcClient流程分析

* 1. 入口文件和main函数

列举入口文件和main函数入口位置

Main入口位置 DRCClient->ChinArkDRCClient.cpp

Run入口位置DRCCLient->ClientApp.cpp 程序主进程

* 1. main函数执行的操作和该函数的流程图



* 1. 分析ClientApp::run函数中做的事情

1. 登记中断处理函数，处理中断。

2.获取客户端版本

3.获取context

a获取灾备机连接信息：ip/port/name/protocal,根据连接信息建立任务管理。

b获取生产机信息：ip/mac/

4.创建客户端回调适配器

5.创建各种子进程，并启动，并将子线程控制器加入到相应的表中。

6.陷入判断重启的循环中。如果没有发生重启，就保持连接。 如果发生重启，将所有子进程销毁。

之后应该是等待watchdog将主进程自动开启，从而完成重启的过程。

* 1. 分析sendKeepAliveMessage如何KeepAlive

通过一个保活计时器keepAlive线程，通过sendKeepAliveMessage不断的更新相应UUID对应的context中的时间戳。

1.加锁

2.根据UUID得到context

3.利用context更新lastTimeIntval与timeIntval

* 1. LisenceHandller的作用和在程序中的流程，并详细分析LisenceHandller::run()

LisenceHandller

循环判断客户端是否退出，如果没有退出，则一直监听。

1.读取配置信息

2.检测数据库是否连接，客户端配置信息是否存在；如果是否，则等待一个时刻，继续检测。

3.检测客户端是否已经删除，如果是，则删除模块信息，备份信息。

4.同步本地与远端的模块信息。

* 1. 详细分析LisenceHandller::handleModuleInfo函数

作用：同步本地与远程的模块开启状态

1.检测模块信息是否发生变化，如果未发生变化则什么都不做。如果发生变化则进行下面的步骤。

2.打印相关调试信息，表明本地模块信息与远程注册模块不一致。

3.判断模块信息不一致的原因。如果远程模块开启则在本地开启模块，如果远程模块已经关闭而本地模块仍然开启，则关闭本地的模块。

* 1. 详细分析TEST\_MODULE\_ACTIVATE宏
  2. startClientListner的作用，为什么要监听
  3. clientInit的作用
  4. 客户端的注册上回调函数ClentCallback流程
  5. 客户端网络离线后又重新上线，是如何实现的
  6. 客户端一旦运行后，启动了哪些线程，线程的作用，线程的变化
  7. UUID如何生成的
  8. UUID是否固定不变，仅仅与硬件相关还是随机生成
  9. 客户端配置完成连接上灾备机后，向灾备机发送了哪些信息
  10. 灾备机如何处理新添加的客户端
  11. 客户端如何处理自己被灾备机上人为删除的
  12. MsgHandller线程的作用和流程
  13. ClientCallbackI提供的功能有哪些

DrcServer流程分析

* 1. 入口文件和main函数
  2. Main函数流程图和执行的操作
  3. DrcServer启动了哪些线程
  4. DRCService::run中执行的操作（监听和注册回调）
  5. 授权部分
     1. 用户操作对Oracle授权时发生的总体流程概要
     2. 用户点击向页面发送的请求详细字段内容，对应字段作用等
     3. 每种授权对应的代码
     4. 授权写入数据库的信息是如何安排的
     5. 授权信息是如何发送到DrcClient的
     6. 授权信息如何组织之后传递到客户端
     7. 客户端如何检测自己已经授权，如何启动Oracle模块的
  6. 灾备机如何检测到客户端的在线，离线状态（ClientCleaner::run）
  7. TaskControll回调是在哪里注册的
  8. TaskControll提供的哪些功能（列表和简要功能介绍）
  9. 日志线程
     1. 分析JobLogUpdater::run()流程
     2. 分析SystemLogUpdater::run()流程
     3. 分析上述日志队列的实现流程
     4. 分析TaskControlI::testJobLogCode中日志是如何被过滤的

新模块授权添加概要设计

* 1. 参照上述流程分析，简述添加OracleRAC模块，在原系统上如何修改，哪些地方需要修改，模块的定义是多少，哪些函数需要修改编码，对外提供给PHP调用的脚本如何修改，客户端如何检测到并写入正确的标志

其它

* 1. DrcClient和DrcServer的编译步骤
     1. 需要哪些组件/依赖才能编译
     2. 编译的操作系统环境

对此次研究的总结