# 诛仙手游数据存储系统

## 1 概述

诛仙手游数据存储系统工作包括游戏服务器运行期间游戏数据的内存存储（SharedMemory）和持久化存储（MySQL）的代码设计和实现，也包含游戏服务器非运行期间数据库升级，合服和缩库等脚本的开发和维护。

## 2 存储架构

## 2.1存储架构概述

诛仙手游服务器端分为 GameServer，DBAgent，HttpAgent三个进程，其中GameServer和DBAgent有存储相关逻辑。存储相关服务器架构图如2-1所示

Login

Routine

Recharge

Routine

DBRoutine

GameServer

…

MySQL Master DB

DBSupRoutine

DBSubRoutine

DBSubRoutine

DBAgent

…

.

sm\_rechargee

MySQL Slave DB

图2-1 诛仙手游存储架构

GameServer的DBRoutine处理GameServer的存储逻辑，主要职能：

* 主动连接DBAgent，连接断开时进行重连
* 缓存并发送GDPacketBuf。
* 接收和发送与其他Routine之间的DBMsg。
* 接收并处理DBAgent发送的DGPacket。

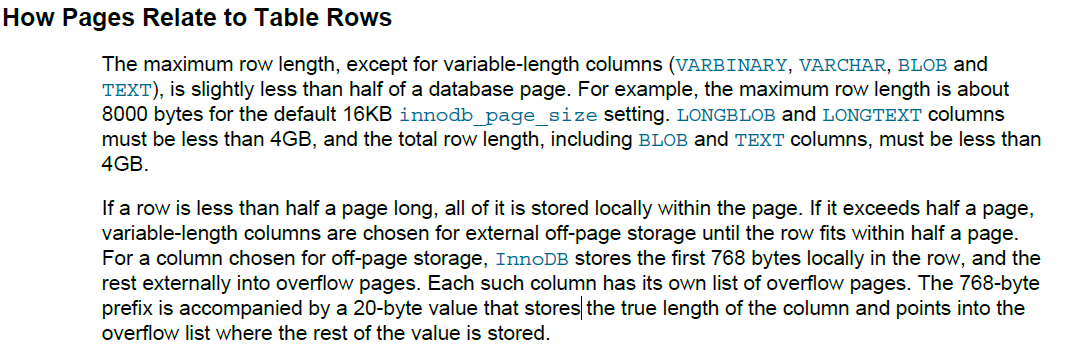
DBAgent的DBSupRoutine处理DBAgent的主要存储逻辑，主要职能：

* 接收并处理GameServer发送的GDPacket
* 创建并管理DBTask
* 接收和发送与DBSubRoutine之间的DBMsg
* 缓存并发送DGPacketBuf

DBAgent的DBSubRoutine是和DBSupRoutine并行的线程，主要执行DBTask的逻辑。

## 2.2 MySQL应用

* MySQL版本5.6.29
* MySQL主从复制机制
* 定时从库全备
* 数据库字符集utf8mb4(emoji表情符号)
* 序列化存储数据
* 数据库表自增列名字为aid，便于合服统一处理。
* 数据库表单行最大实际长度不要超过8000 Bytes。



* 即使是varchar/char类型，定义的长度也要尽可能的小，实际需要多少就定义多少。

## 3 数据存储方式

诛仙手游服务器运行期间数据，可以分为角色数据和公共数据。存储方式分为内存存储和持久化存储两种。内存存储是采用共享内存机制存储公共数据，持久化存储是将角色数据和公共数据存储到MySQL数据库。

### 3.1内存存储

每个Routine中存在的需要存储的数据称为公共数据，例如GuildRoutine中的帮会列表，FamilyRoutine中的家族列表等。诛仙手游内存存储数据，是指将公共数据存入共享内存（SharedMemory）中。将公共数据写入共享内存中，主要是缓解MySQL的写入压力，缩短宕机时公共数据的回档时间。

存储公共数据的所有SharedMemory类声明都在SharedMemory\_GameServer.h中，每个需要存储公共数据的Routine对应一个SharedMemroy类，每个类必须要有以下变量：

int32\_t m\_type\_size;

每个类必须要实现以下函数：

void init(void);

bool check(void) const;

void clean(void);

void datavalidate\_begin(void);

void datevalidate\_end(void);

公共数据存入SharedMemory是内存间数据拷贝，在Routine的Tick中定时处理，因速度快，所以存储的间隔基本在60秒左右，此控制时间在GameConfig.ini中配置。

公共数据从SharedMemory读取，是发生宕机或其他异常停服情况下使用，启动GameServer时，需要指定参数-loadmode=9999；

### 3.2 持久化存储

诛仙手游一切需要存储的数据，都会存储MySQL数据库中。MySQL采用主从机制，从库会定时进行数据库全备。数据库的命名规则为zxdb\_worldid（worldid替换为对应的世界号）。

运行期间会将数据进行Tick存储，停服会将数据全部存储，正常停服（非宕机等异常情况）会从数据库中加载数据。

#### 3.2.1 数据库存储逻辑

诛仙手游的数据库存储逻辑都是GameServer主动发起存储请求（加载或者存储数据）到DBAgent，DBAgent处理存储请求后返回结果消息给GameServer。详见图3-1。

GameServer DBRoutine

DBAgent DBSupRoutine

DBAgent DBSubRoutine

DB\_Msg

GDPacket

DBAssignTaskMsg

DBTaskRetMsg

DBTaskExecuteResultMsg

DGPacket

DB\_Msg

图3-1 诛仙手游存储逻辑时序图

#### 3.2.2 DBTask管理逻辑

DBTask是DBAgent与MySQL交互处理的封装类，DBTask的管理逻辑存在于DBAgent的DBSupRoutine中，主要管理DBTask的以下方面：

* 相同类型相同Key值的DBTask的串行执行控制，避免DBTask之间出现并行顺序错乱情况。
* 相同类型相同Key值的DBTask的覆盖处理，如果DBTask执行过程中，接收到多个新的DBTask，则根据接收顺序保留最新的一个，避免DBAgent申请过多内存。
* 存储数据的DBTask在存储失败后，如果队列中没有新的相同类型相同Key值的DBTask，会重新加入等待队列，继续进行存储，如果已有新的相同类型相同Key值的DBTask，则会忽略不再进行存储。

具体处理逻辑，见图3-2

Handle GD\_Packet

Create New DBTask

Ask Add DBTask To DBTaskWaitList

Same DBTask In DBTaskWaitList

Add New DBTask To DBTaskWaitList

Replace Old DBTask

Tick\_AssignTask

Can DBTask Direct Execute

Same DBTask Executing

Send Task To SubRoutine

Insert DBTask To

DBTaskExecutMap

Erase DBTask From

DBTaskWaitList

Yes

No祝亮哥早日拿到手机。

Yes

No

Yes

No

DBTask Execute

Success

DBTask For Save

Yes

Erase DBTask From

DBTaskExecutMap

DBTask Execute

DBTaskExecutMap

No

Yes

No

## 4 数据库新建和升级

### 4.1新建数据库

线上开新服，内网研发会需要新建数据库。

脚本路径：服务器代码目录/SqlScript/CreateDB/，路径下主要包含：

* 建库数据初始化脚本
* 建库脚本
* 建库控制执行脚本，供运维平台线上建库调用，内部研发Jenkins建库调用。
* 随机名字数据，建库时加载到随机名字数据表中。

### 4.2 升级数据库

线上数据库版本更新时，需要升级数据库。主要包含：

* 更新脚本
* 升级检查脚本，升级后检查本次升级是否正确。
* 升级控制执行脚本，供运维平台线上数据库升级调用

## 5 合服

游戏线上运营，会有合服需求。

## 6 缩库

游戏线上运行，随着数据积累，数据库越来越大，为避免影响正常运行，减轻定时全备压力，会有缩库需求。

## 7 数据库优化

### 7.1 MySQL数据库表空间优化

定期使用MySQL的optimize table命令，降低存储空间，提升I/O效率。

脚本目录：

## 8存储结构

增加存储时，需要存储的数据及数据结构，需要满足以下条件：

* 逻辑上需要存储的数据Struct要有对应的DBStruct。
* 需要存储的数据必须是原生数据类型，DBStruct中字符串需要用char类型数组。
* 需要存储的数据DBStruct及其子DBStruct不能包含虚函数，指针和引用。
* 需要存储的数据DBStruct包含CleanUp函数，方便数据清除。
* 已经序列化的内存数据，一旦开始使用，不能调整数据的结构，包括字段的顺序、字节对齐方式。
* 对于旧存储的扩展，必须要考虑新版本是否能兼容旧数据。
* 结构体数组使用序列化内存的方式存盘，确实需要时使用，尽量不要使用。