DBS

DBS部署于Resin容器中，作为Redis缓存和Mysql数据库之间的桥梁，负责对缓存中数据延时存储到数据库中，下面具体介绍这种机制。

# 原理

在缓存中标记数据的变更状态，每隔一段时间，将标记的数据集中入库一次。这样可以减少SQL生成量，减轻对数据库的压力。

# 工作流程图

定时调度

（1分钟）

调用HumanDataUpdater和CacheCommonDataUpdater的run方法

依次调用HumanDataUpdater的switchCache或flushOpposite方法

每个Redis启动一个线程，根据ModifiedProperty将玩家在redis中变化的数据异步更新入库

UpdateCacheModified的saveOrUpdate方法根据传进来的ModifiedProperty的操作类型，使用相应的Dao更新Entity

# 补充说明

1. 数据变更标示缓存

游戏存入缓存的时候是Java的内存模型，不是Entity，例如Human整个对象会序列化成二进制数据，然后存入缓存，取出来在反序列化。

游戏向缓存中存储数据时，如果数据有变更，则同时向数据变更标示缓存写入变更的标示。DBS存库的时候，遍历数据变更标示缓存中的数据，将变更的缓存数据同步入库。

数据变更标示缓存使用SortedSet进行数据存储，通过Java的当前时间进行排序，保证先标记的先存储。

当有多个Redis的时候，每个Redis上都会有一个数据变更标示缓存。每个Reids的数据变更标示缓存分为两块，二者交替进行，DBS存储的时候使用一个，游戏存缓存的时候用另一个，当DBS存完的时候，更换另一个，游戏也更换。

1. 数据变更标示（CacheProperty）

变更标示需要包含以下属性：

|  |  |
| --- | --- |
| 属性 | 用途 |
| 玩家ID | 查询玩家在哪个数据库 |
| 角色ID | 数据库具体某一列主键 |
| 缓存属性类型 | 区分不同的数据表，玩家、角色、建筑、成就等 |
| 缓存操作类型 | 区分增删改操作 |
| 标示ID | 可选项，针对一个对象中包含多个数据存储。  如BuildingManager有许多building，每个building为数据库中一行，如果其中多个building改变，那么只需要存储这些变更的building，标示ID就标示这些变更的building的ID. |

1. 数据库存库方案：

使用Hibernate的save或者update，存储数据。单表所有列数据会一起保存，而不是针对某一列数据变更只存储变更的数据。从而降低Hibernate的开销。例如HumanEntity中有多个字段，其中一列gold数据有变更，那么会把所有HumanEntity的所有数据列全部存库。