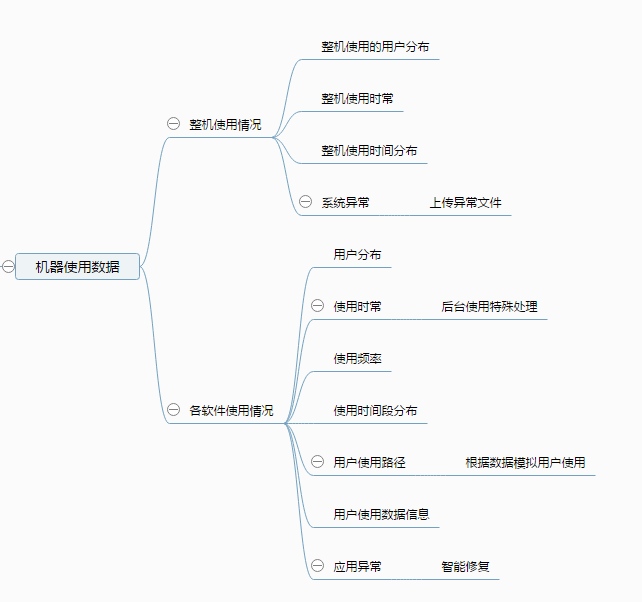
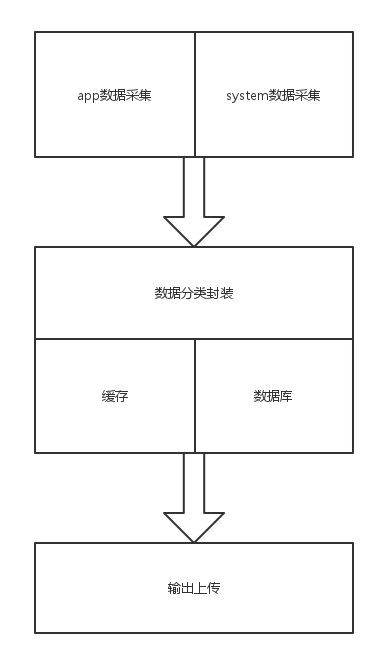
## 需求



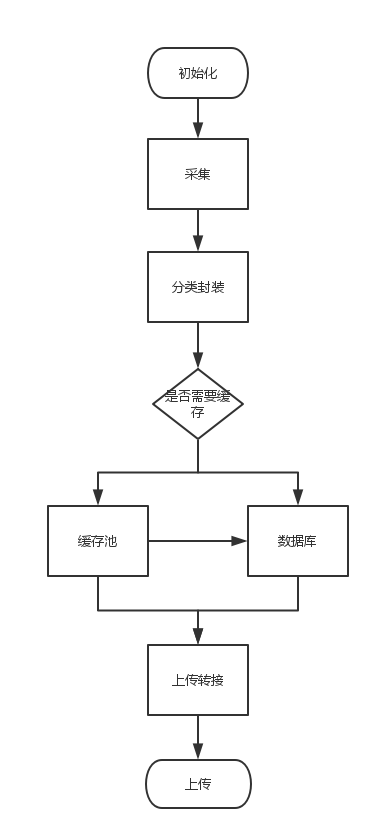
## 目标

实现大数据采集、整理、缓存、入库、提供上传。

## 框架图



## 流程图



## 模块

各个模块具体功能。

### 采集

功能：提供给各个app调用，采集数据。

接口：

ICollecter定义提供app调用的接口。

接口实现类：

BehaviorCollector app应用和DA采集解耦。

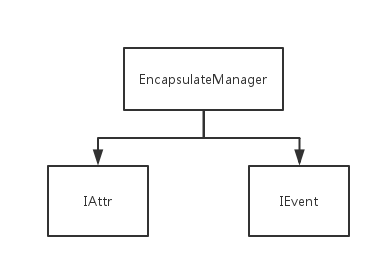
注意：

1. 初始化问题。很多东西都需要初始化才能正常使用，必须调用init()函数。
2. UserAttr用户信息。
3. 传参形式：bean? 多参数？
4. 如果命名没有特别大问题，提供app调用包名和类名、函数尽量和之前保持一致，就减少各个app修改的负担。

### 分类封装

功能：

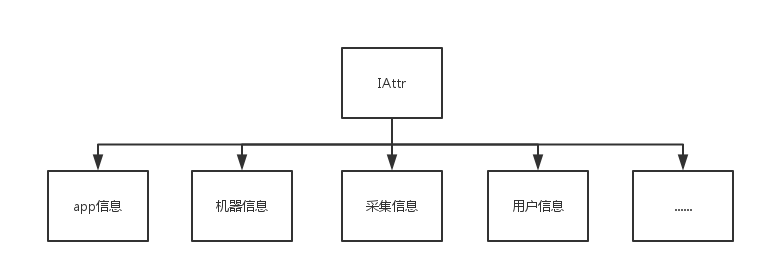
1. 数据合法性。
2. 数据分类。
3. 添加必要的默认数据，如app信息、机器信息、用户信息等等。



接口：

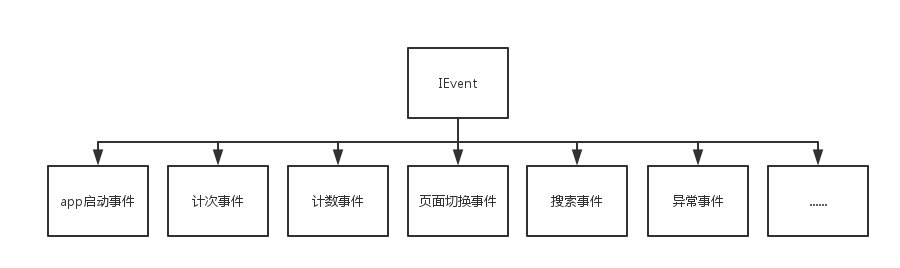
IAttr 对接数据库接口。

IEvent 用户采集数据整理。



IAttr实例类型：

采集信息、App信息、机器信息、用户信息等等….



IEvent实例类型：

计次事件、计数事件、搜索事件、异常捕获事件等等….

管理类：

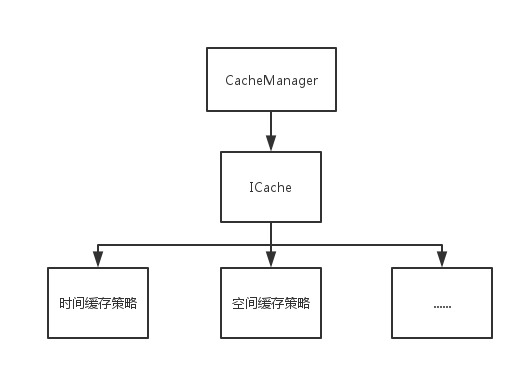
EncapsulationManager 封装分类管理类。

输出：IEvent实例

### 缓存池

功能：

1. 减少入库操作，提高效率。
2. 根据数据类型，进行提交上传、缓存和直接入库操作。
3. 根据多个或者单个缓存机制。



接口：

ICache 缓存的逻辑处理和触发入库的接口

ICache实例类型：

时间缓存、空间缓存

管理类：

CacheManager 组合、管理ICache,触发入库和上传转接。

注意：

1. 需不需要一个app控制是否缓存的开关？
2. App多进程的话，缓存池是否也会有多个？是优化机制使多个缓存池不影响使用，还是想办法统一用一个缓存池？
3. Sessionid也有同样进程问题。
4. 进程被杀，缓存池不能及时入库导致丢失。

输出：

List<IEvent>

### 数据库

功能：提供更删改查。

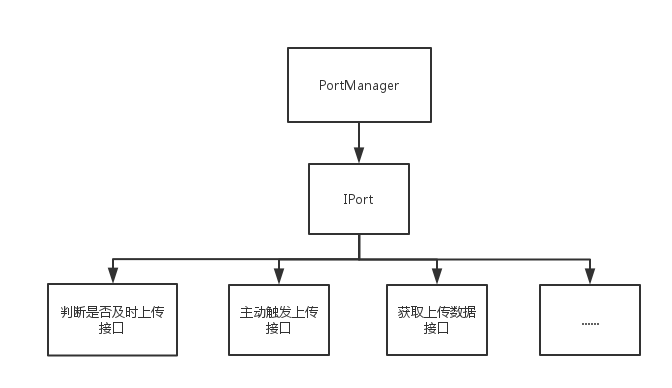
注意：

1. 线程安全。
2. 批量。

### 上传转接

功能：

1. 连接采集和上传。
2. 数据整合（IEvent🡪上传需要的类型）。
3. 根据事件类型判断是否及时上传接口。
4. 提供主动触发上传接口。
5. 提供上传部分获取数据接口。



接口：

IPort 提供主动调用上传和获取上传数据的接口。

管理类：

PortManager 整合采集数据、隔离采集和上传的管理类。

注意：

1. 具体数据类型要和上传部分协商。