**精准DB拆分**

**技术方案**

版本：<文档版本>

**文档修订历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 作者 | 内容提要 | 修订日期 |
|  |  |  |  |
|  |  |  |  |

**文档保密声明**

本文档仅限支付宝公司内部使用，未经许可，不得外传。

**目录**

[1 概述 4](#_Toc390105494)

[1.1 术语定义 4](#_Toc390105495)

[1.2 需求背景 4](#_Toc390105496)

[2 数据架构分析与设计 5](#_Toc390105497)

[2.1 数据库拆分方案 5](#_Toc390105498)

[2.1.1 总体拆分原则 5](#_Toc390105499)

[2.1.2 整体拆分方案 6](#_Toc390105500)

[2.1.3 具体拆分方案 6](#_Toc390105501)

[2.1.4 数据后双数据源方案 7](#_Toc390105502)

[2.2 数据库拆分实施方案 8](#_Toc390105503)

[2.2.1 低峰暂停修改发布 8](#_Toc390105504)

[2.2.2 逐步引流方案 9](#_Toc390105505)

[2.3 跨数据库事务评估 11](#_Toc390105506)

[2.4 拆分影响 11](#_Toc390105507)

[2.4.1 查询业务 11](#_Toc390105508)

[2.4.2 积极影响 11](#_Toc390105509)

[2.4.3 数据仓库影响 11](#_Toc390105510)

[附： 12](#_Toc390105511)

[1. 运维能力分析与设计 12](#_Toc390105512)

[1.1 发布兼容能力分析 12](#_Toc390105513)

[1.1 发布回滚能力分析 15](#_Toc390105514)

[1.2 动态配置能力分析 16](#_Toc390105515)

[1.3 监控能力分析 17](#_Toc390105516)

[2. 性能、容量与稳定性分析 17](#_Toc390105517)

[2.1 性能、容量与可伸缩性分析 17](#_Toc390105518)

[2.2 高可用性分析 19](#_Toc390105519)

[2.2.1 单点分析 19](#_Toc390105520)

[2.2.2 故障隔离分析 21](#_Toc390105521)

[2.2.3 可恢复能力分析 23](#_Toc390105522)

[2.3 正确性分析 23](#_Toc390105523)

[2.3.1 资损控制能力分析 23](#_Toc390105524)

[2.3.2 业务平衡检查能力分析 24](#_Toc390105525)

[2.3.3 出错数据订正分析 24](#_Toc390105526)

[2.4 可靠性分析 25](#_Toc390105527)

# 

# 概述

## 术语定义

|  |  |
| --- | --- |
| **术语** | **说明** |
| 数据库拆分 | 一般分为垂直拆分，水平拆分，用于提升系统容量和稳定性的常用方法。 |
| failover | 一种数据库架构，可以在主库挂掉的情况下由应用切换到备库，主库和备库间数据不同步，所以可以很好的保证存储稳定性。 |
|  |  |

## 需求背景

本次改造项目的需求背景有两个：

1. 随着新接入业务和已有业务的不断增长，系统承载的业务量不断增加， 就近7日统计每分钟service 5w/分；sal（向外调用）已达到35W/分；同时系统对请求记录、推荐、推荐信息记录日志、过程数据记录都进行了详细记录。
2. 不断增长的业务量，对系统的容量和稳定性提出了更多的挑战。对于日志和推荐信息记录。在不影响主流程的情况下，异步降级，最大程度降低系统对数据库的依赖。

# 数据架构分析与设计

## 数据库拆分方案

### 总体拆分原则

#### 是否做failover架构

failover架构，可以在主库挂掉的情况下由应用切换到备库，可以很好的保证存储稳定性。failover架构的特点适合主库宕机后，新数据可以入备库，新业务不受影响。

精准业务目前越来越核心，服务提供不稳定，直接会导致线上小二服务量剧增，影响公司的服务质量，所以有必要作主备库方案。

### 整体拆分方案

精准规则

场景/分类

快照记录

请求记录

精准规则

精准数据依赖

DB（异步插入）

DB（高可靠）

结果记录

### 具体拆分方案

1. 确定主拆分维度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **方案名称** | **优点** | **缺点** | **备注** |
| 方案一 | 根据各业务数据为维度拆分 | 每个业务数据是否需要高可用保证，拆分成2个库，拆分简单。 | 对于某些整合查询需要通过双数据源查询。 | 拆分为要求告可用的，和允许异步插入两部分 |
| 方案二 | 用户维度拆分 | 个体数据聚合在一起，方面个体数据的查询；事务处理简单。 | 满足不了某些业务拆分需求。 |  |

根据本次业务场景，采用方案一进行拆分。

1. 验证拆分维度

现有业务数据简单梳理如下：拆分维度业务数据维度。下表中插入操作可以异步，即精准服务中非要求高可用性表。剩余的精准服务 的其他表迁移到新的库中(高可用性，主备库)

|  |  |  |
| --- | --- | --- |
| 业务场景 | 表说明 | 备注 |
| ac\_service\_source | 精准服务模块-服务请求来源，记录精准服务请求来源 | 保留表 |
| ac\_accurate\_record | 业务数据维度 | 保留表 |
| ac\_operate\_log | 系统操作日志 | 保留表 |

### 迁移后双数据源方案

1. （若需要重新通过数据库表生成xml文件，需要第一步配置修改，本次迁移不涉及）修改anygen项目中db.xml、gen\_config.xml文件及代码生成渲染模板${appModule}-dal.xml.auto
2. 修改dal层，本次拆分后需要修改common-dal.xml文件，需要根据DBA申请到最新资源给出dataSouce及sqlMapClient 双bean

截图示例如下：

|  |
| --- |
|  |
|  |

## 数据库拆分实施方案

### 低峰暂停修改发布

该方案是发布中成本最低的修改方案。主要方式是

1. 通知因子及规则修改角色人员暂停修改
2. DBA同步旧库(关键表)到新库
3. 链接新库旧库的双数据源代码发布

### 逐步引流方案

因为系统上线后，不能一刀切到拆分后的数据库，需要逐步的引流到新存储上，新老存储需要兼容并存一小段时间，在新老存储并存阶段，需要所有功能运行正常。并且能够动态的引导流量。

增加一个流量动态控制模块，通过drm来确定引流的量。若新存储出现问题，可以通过drm开关将所有流量切到老存储，需要全部切到新存储时同理。

在新老存储并存阶段，将不会进行数据迁移，待需要全部切到新存储时，再做。所以系统完成上线，将分为五步走，任何一步都保持向前一状态回滚的能力：



经过评估选取，选择低峰暂停修改 发布的方案，因为涉及业务修改角色和人员很少，方案简单高效有效。

## 跨数据库事务评估

拆离出去的正是和留下的是要异步处理的数据，所以不存在跨数据库事务。

## 拆分影响

### 查询业务

随着业务发展也许会出现遍历多库进行联合查询，不够目前数据都同步到数据仓库的，可以在数据仓库提供查询分析服务，所以总体问题不大。

### 积极影响

拆分后稳定性进一步提升。同时完成精准DB去单点，一写多读，大促等可以更有力应对。外围系统影响分析

### 数据仓库影响

初步评估：因为此次拆分不涉及分库分表，属于数据库表之间水平拆分，所以对数据仓库除ETL链接属性修改外，其他影响应该不大。但是数据那边是实时取的（待文宋进一步答复）

2.4.4 其他系统影响