前端消息缓存方案设计

## 需求背景:

当前APP所有聊天记录每次都从服务端API接口分页拉取,每次拉取都会有大量重复的消息.

第一.会造成服务器资源与流量的浪费.

第二.客户端体验不好,手机在无网络状态时无法查看之前的聊天记录,

第三.每次进入会话时,需要等待服务端返回聊天历史记录,在网络不好的情况下需要等待很长时间

第四:未缓存到本地的聊天记录无法进行文字消息搜索.后续服务端消息会采取加密存储,无法提供搜索聊天记录服务,如果保存在前端,则可以直接在前端数据库中搜索聊天记录

所以,前端需要把已经接受到的消息,全部保存至前端的本地数据库中,已经在本地保存过的消息将不再从服务端数据库拉取.

如果用户换了设备进行登录, 则需要全量拉取之前的聊天历史记录到前端,如果数据量过大,需要考虑分页拉取,或只拉取最近x个月/最近xxx条数量的历史记录

## 后端技术方案:

## 方案一: seq 序号id (不考虑)

### 如何保证客户端缓存消息丢失?多端同时登录情况下,在其他端的聊天内容如何同步缓存?

#### 后端数据库表设计:

**[im\_message] 消息存储表:**

message\_id: 主键,消息唯一ID

sender:发送方

content:消息内容

**[im\_inbox] 消息收件箱:**

**inbox\_id:主键ID**

message\_id: 外键:消息唯一ID

receiver:接收方

conversation:会话ID

read\_flag:已读/未读状态

**seq:消息序号,用户在每个会话都有一个seq,每个会话都是从1开始递增,每次递增为1,后端一定保证是顺序递增+1的ID,且保证后发送的seq大于前发送的seq**

当前后端采用的是收件箱模型:

假设现在**A B C**三个用户加入了**会话1,**

A发送一条消息 "你好", 后端会生成的一条消息唯一ID(MessageID)为: **10001** 保存至[im\_message]消息存储表, 然后在[im\_inbox]消息 ABC分别都会生成一条seq为1的记录.

此时,**用户D**加入群聊, B再发送一条消息,后端会生成的一条消息Message ID为: **10002**保存至[im\_message]消息存储表, 然后在[im\_inbox]消息 ABC分别都会生成一条seq为2的记录. 但用户D生成的seq则为1

### 已缓存在客户端的消息,对方撤回或删除后,如何同步?

调用撤回消息接口时将新生成一条"type=撤回通知"类型的消息存入[im\_message] 表, 其中content内容为原msgid

同时保存[im\_inbox] 收件箱表, 新的[im\_inbox]记录中seq序号将大于原消息的seq

### 服务端分布式场景下,如何保证seq的生成顺序是+1递增?

新增数据库表: 用来维护用户在每个会话的seq.

**[im\_conversation\_seq] 用户-会话-序号维护表:**

**id:主键ID**

receiver:接收方

conversation:会话ID

seq:序号

每次保存inbox收件箱时,从[im\_conversation\_seq] 表取出seq,并进行seq+1作为inbox的seq, 然后再将seq+1后的值同步修改到im\_conversation\_seq表. 由于后端是分布式部署的, **如何保证分布式多线程进行seq+1数据修改操作的ACID ?**

添加分布式锁,采用Lock4J分布式锁,每次获取seq时将该条数据锁住,后接受的消息必须等前一条消息存入库提交完,再获取锁,来保证seq的顺序性

接口 :

#### 拉取会话历史消息接口

背景说明

* 即时通信 IM 的消息是按 Seq 排序的，按照服务端收到消息的顺序分配 Seq，先发的消息 Seq 小，后发的 Seq 大,后发的消息每次递增+1。
* 如果用户想拉取一个会话的全量消息，首次拉取时不用填 Seq，服务端会自动返回最新的消息，以后拉取时 Seq 填上次返回的最小 Seq 减1。
* 如果返回消息的 IsPlaceMsg 为1，表示这个 Seq 的消息或者过期、或者被对方撤回、或者被删除了。

## 方案二: 消息id (选用)

当前数据库表结构

**[im\_message] 消息存储表:**

message\_id: 主键,消息唯一ID,后发送的id大于前的消息id

sender:发送方

content:消息内容

**[im\_inbox] 消息收件箱:**

**inbox\_id:主键ID**

message\_id: 外键:消息唯一ID

receiver:接收方

conversation:会话ID

read\_flag:已读/未读状态

消息表与收件箱为一对多关系, 一条消息将会生成多条收件箱记录,会话的每个接收方都会有一条收件箱记录,每条消息有已读和未读状态, **未读的消息=离线消息**

**缓存消息的流程,场景1:**

**用户换新设备登录, 不需要同步之前的聊天记录, 只需要同步未读的离线消息**

T+1. 用户A在手机1登录, 产生聊天数据 1,2,3,4,5,6 总共6条消息

并且全部为已读状态,已经缓存在手机1

T+2.用户A在手机2登录, 不用展示在手机1的聊天内容: 1,2,3,4,5,6 , 也就是说换设备登录不需要同步在之前的设备的聊天记录

T+3,用户A在手机2与用户B产生聊天记录7,8,9并且全部为已读状态,已经缓存在手机2.

T+4.用户A在手机2退出登录.

T+5,用户C给用户A,发送消息10,11,12, 因用户A退出了登录,10,11,12为离线消息保存在服务端数据库.

T+6.用户A在手机1登录,不需要展示在手机2的聊天记录7,8,9, 但需要展示退出登录期间的离线消息10,11,12

**缓存消息的流程,场景2:**

**收到消息并缓存在本地,但没有点开进入会话查看,该消息还是未读状态消息,换新设备登录后需要展示未读的消息.**

T+1,用户D在手机1登录, 用户F给用户D发送消息1,2,3

T+2,用户D进入该会话聊天界面查看了该3条消息

T+3,用户D从聊天界面返回至会话列表

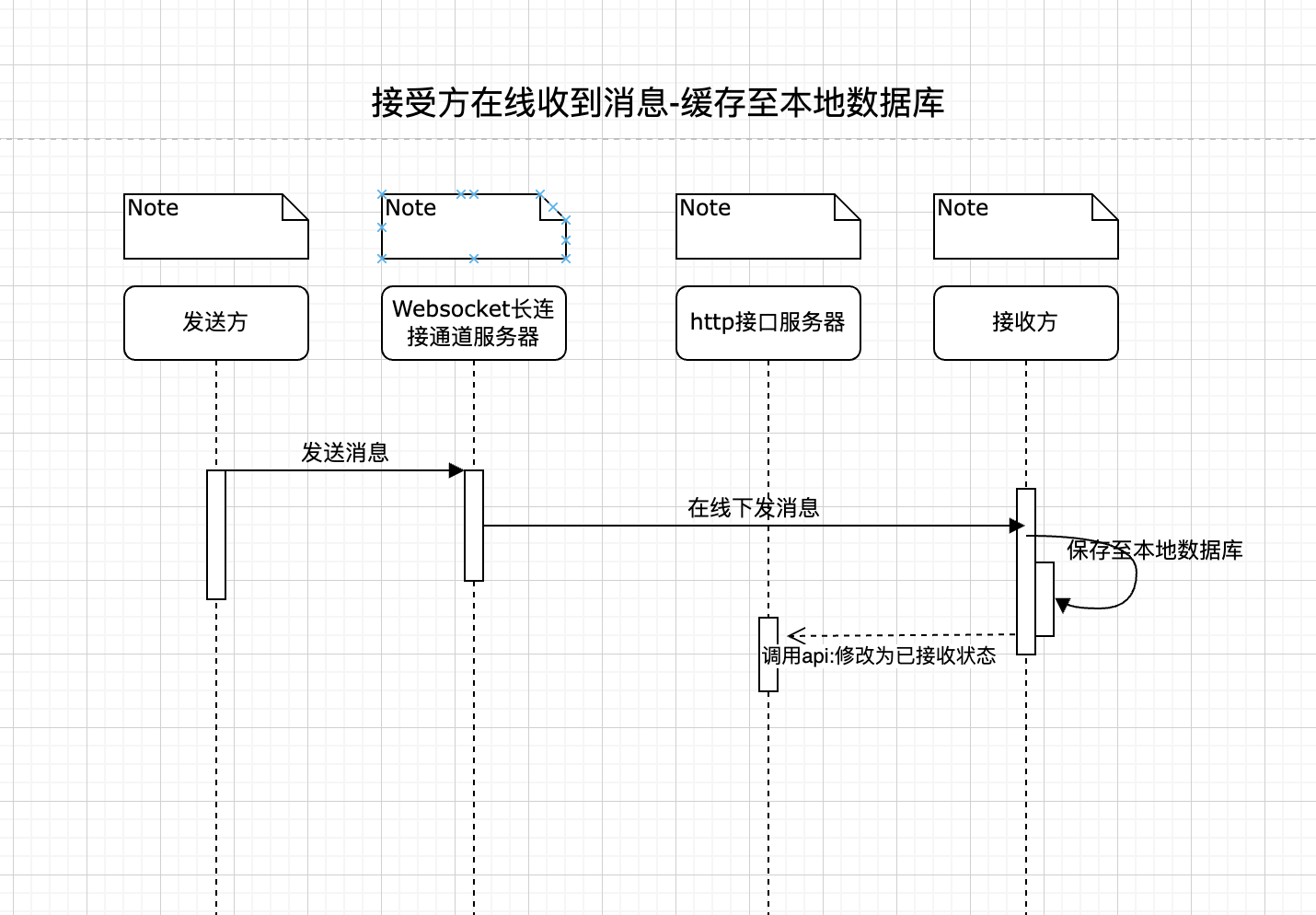
T+4, 用户F继续给用户D发送消息5,6,7

T+5, 用户D在会话列表接受到了5,6,7, 但是没有点击进入会话进行查看消息

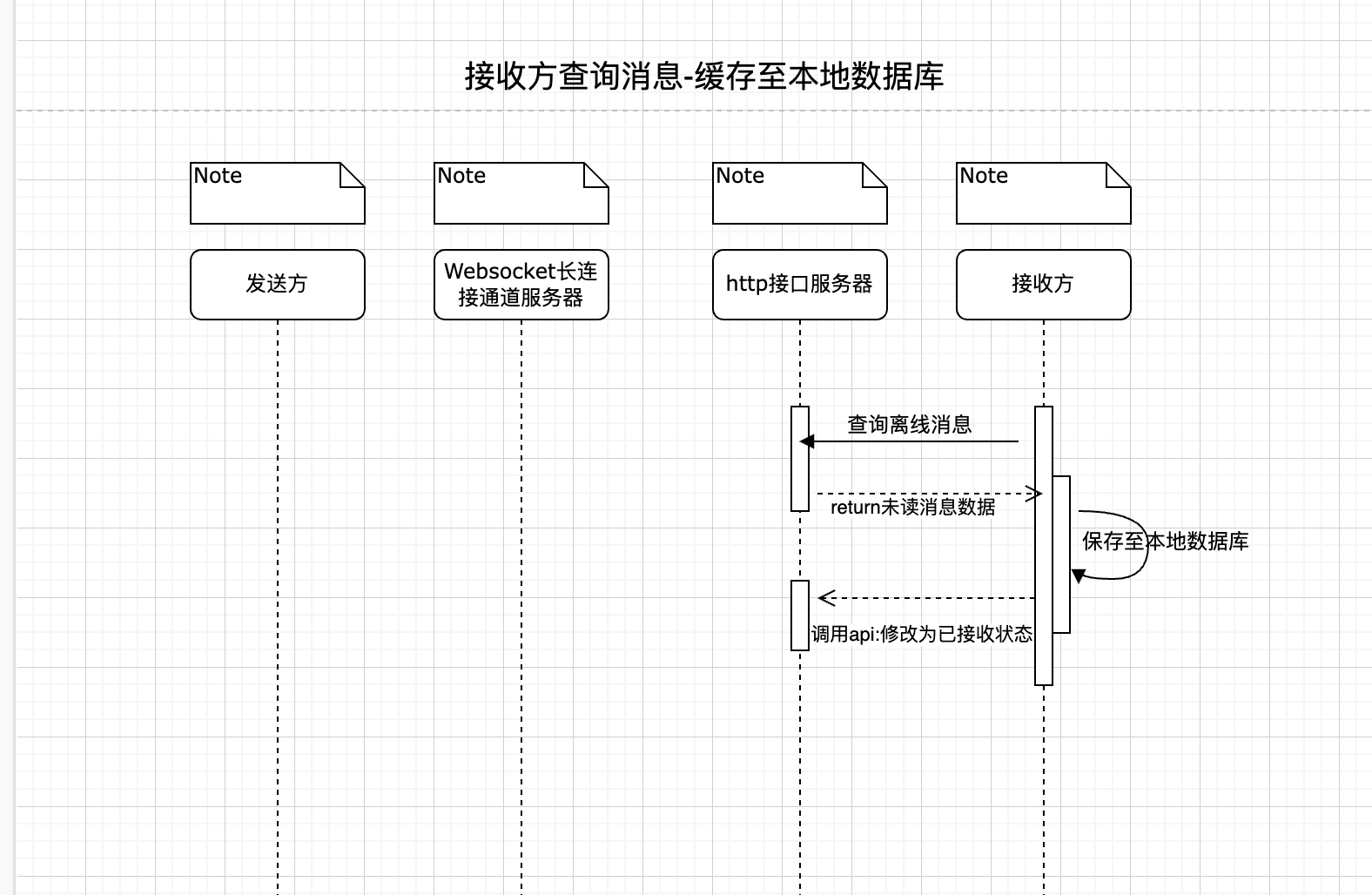
T+6, 用户D在手机2登录,不需要展示1,2,3, 但需要展示用户F发过来的未读的消息5,6,7.

### 流程图

#### 在线:接受方在线收到消息-缓存至本地数据库



#### 离线: 接收方查询离线消息-缓存至本地数据库



### 提供接口:

**拉取全部离线消息接口:**

返回数据:发给该用户的所有未读状态的消息

### 已缓存在客户端的消息,对方撤回或删除后,如何同步?

调用撤回消息接口时将新生成一条"type=撤回通知"类型的消息存入[im\_message] 表, 其中content内容为原msgid

同时保存[im\_inbox] 收件箱表, read\_flag:为未读状态(**离线消息**)

**客户端在拉取离线消息时,就能把已缓存在本地的消息修改为撤回状态**

**示例:**

|  |
| --- |
| **假设有以下消息ID** |
| 1 |
| 4 |
| 5 |
| 7 |
| 8 |
| 10 |
| 13 |
| 15 |
| 16 |

|  |
| --- |
| 红色字体表示本地已经缓存 |
| 蓝色字体表示未读的消息(离线时别人发送的消息) |
| 黑色为云端存储的历史漫游消息 |

|  |  |
| --- | --- |
| **场景1:** | 15.16为别人发送的离线消息, 客户端该如何同步到本地缓存? |
| 1 |  |
| 4 | 如何把末尾的离线消息缓存至本地: 服务端提供离线消息查询接口, 查询该会话中所有未读的离线消息 |
| 5 |  |
| 7 |  |
| 8 |  |
| 10 |  |
| 13 |  |
| 15 |  |
| 16 |  |

|  |
| --- |
| **场景2:** |
| 1 |
| 4 |
| 5 |
| 7 |
| 8 |
| 10 |
| 13 |
| 15 |
| 16 |

# 前端设计统一

数据库表字段: