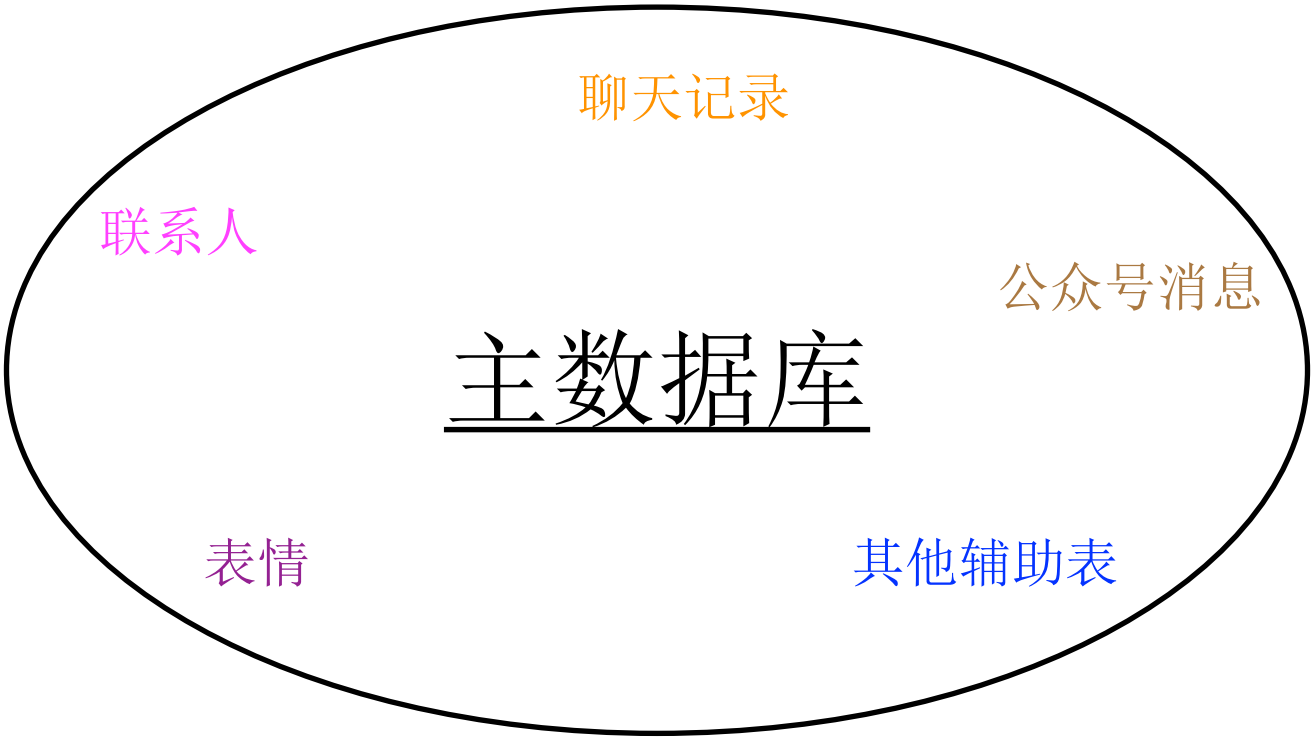
与「十」俱进，码出未来

2018源创会年终盛典

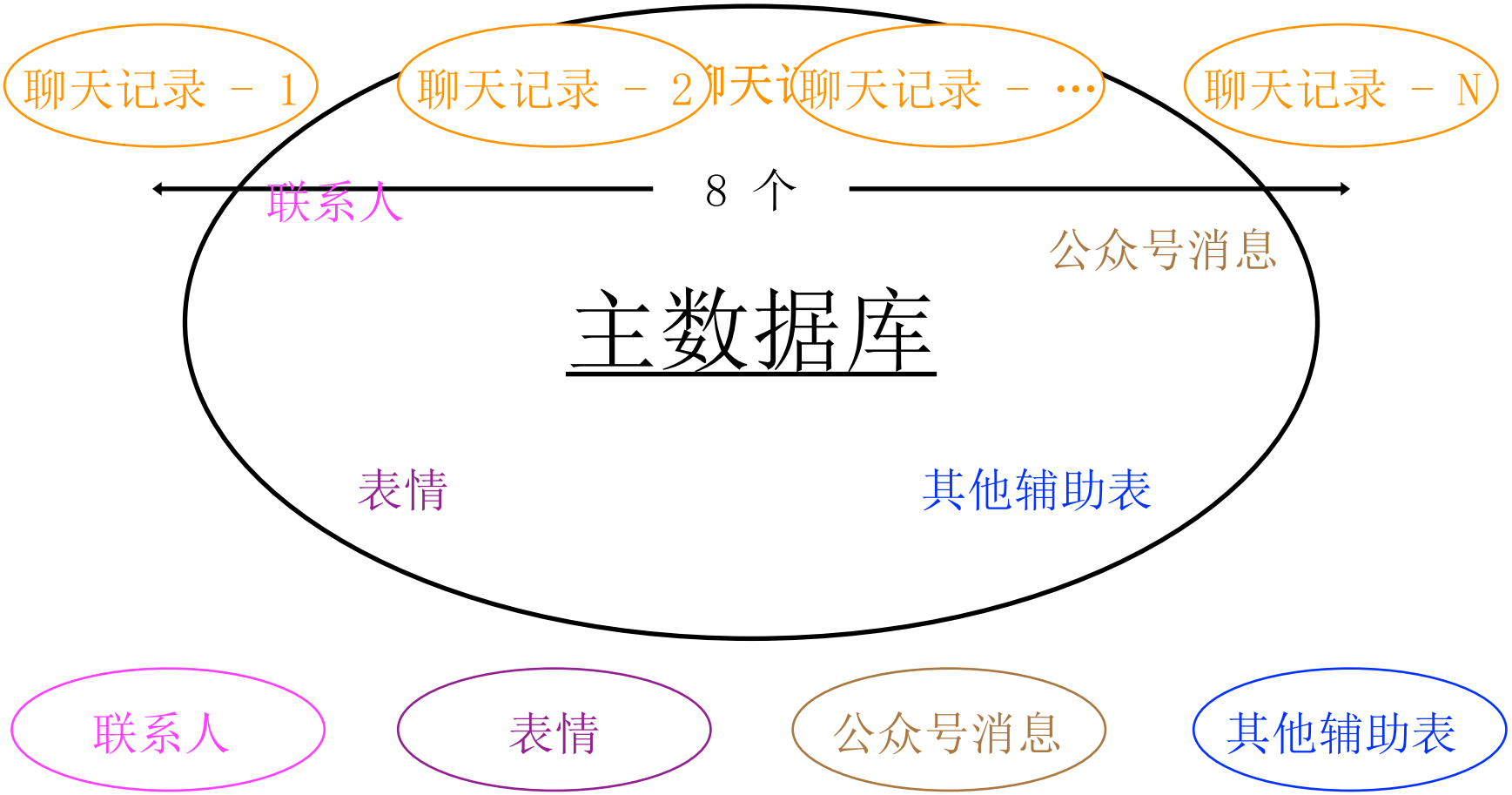
主办方 开源中国 oschina.net

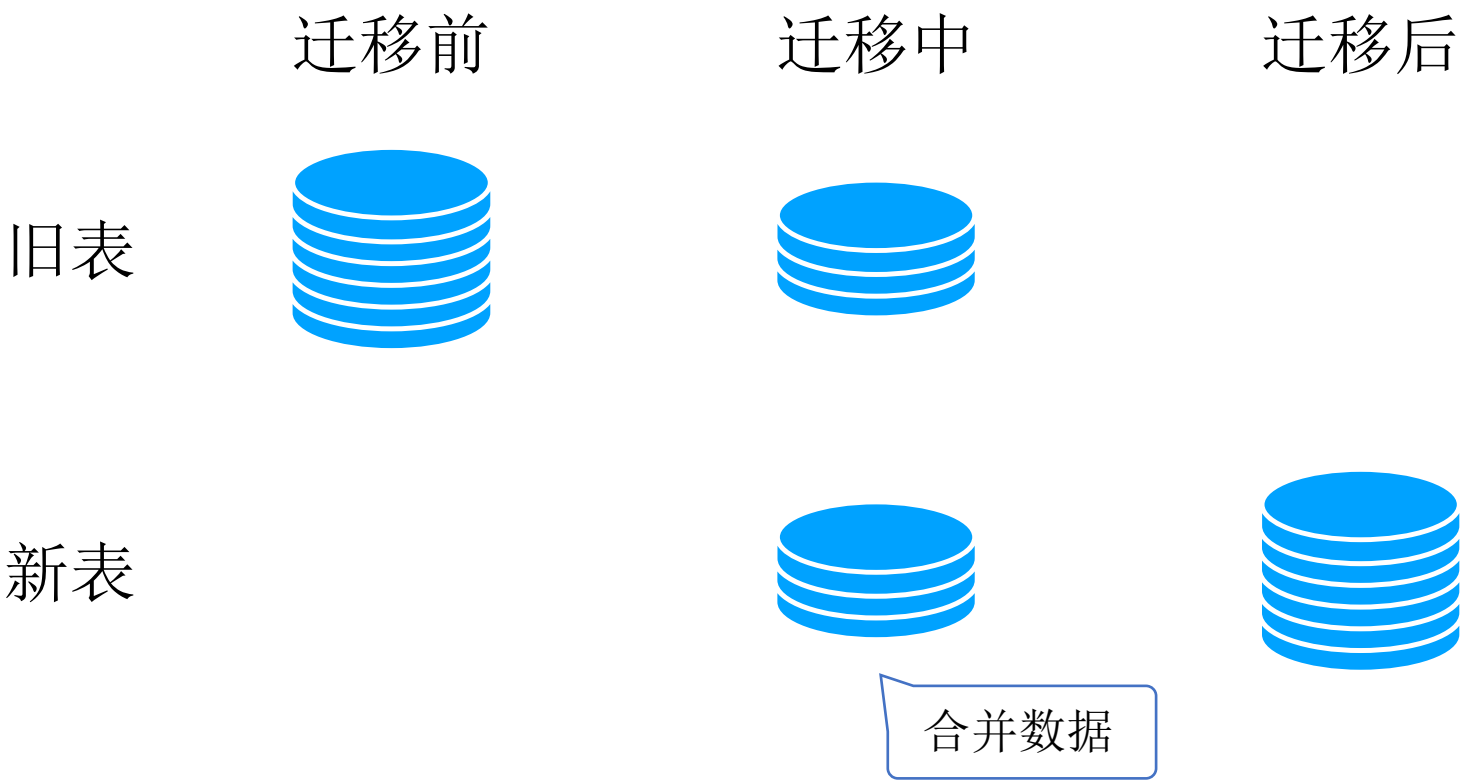
微信终端开源数据库 WCDB

数据库修复与无感迁移方案



- 代码耦合
- 并发性差
- 健壮性差


聊天记录 - 1
聊天记录 - 2
聊天记录 - …
聊天记录 - N
8 个
联系人
公众号消息
主数据库
表情
其他辅助表
联系人
表情
公众号消息
其他辅助表


迁移前
迁移中
迁移后
旧表
新表
合并数据

Write Once, Run Everywhere ?

WODB 无感迁移方案！

让开发者以迁移完成为假定前提进行开发

```
// 旧表
CREATE TABLE oldTable(identifier INTEGER,
                      content    TEXT)

// 新表
CREATE TABLE newTable(identifier INTEGER,
                      content    TEXT)
```

```
SELECT identifier, content FROM newTable LIMIT 1
```



```
SELECT identifier, content FROM unionedView LIMIT 1
```

```
CREATE TEMP VIEW unionedView AS
    SELECT * FROM oldTable
    UNION ALL
    SELECT * FROM newTable
```

```
WITH cte(t) AS (SELECT max(t) FROM t WHERE t LIKE
"DELETE") UPDATE t.t SET t=(SELECT sum(t) FROM other.t
WHERE cte.t="t" and main.t="INSERT")
```

- schema 名
- 字段名
- 临时字段
- 字符串
- 其他数据库的表名
- 当前数据库的表名

```swift
/* WINQ: WCDB 语言集成查询 */
var column = Column(named: "t")
var table = TableOrSubquery("t");

// SELECT sum(t) FROM t WHERE t > 1
var select = StatementSelect()
                .select(column.sum())
                .from(table)
                .where(column > 1)
```

```
INSERT INTO newTable VALUES(1, "Hello, WCDB")
```

```
SAVEPOINT

INSERT INTO oldTable VALUES(1, "Hello, WCDB")

--保存 lastInsertedRowID

INSERT INTO newTable(rowID, identifier, content)
    SELECT rowID, identifier, content
    FROM oldTable
    WHERE rowID = lastInsertedRowID


DELETE FROM oldTable
RELEASE
    WHERE rowID = lastInsertedRowID
```

```
UPDATE newTable SET content = "Hello, WCDB"
    WHERE identifier > 10
```

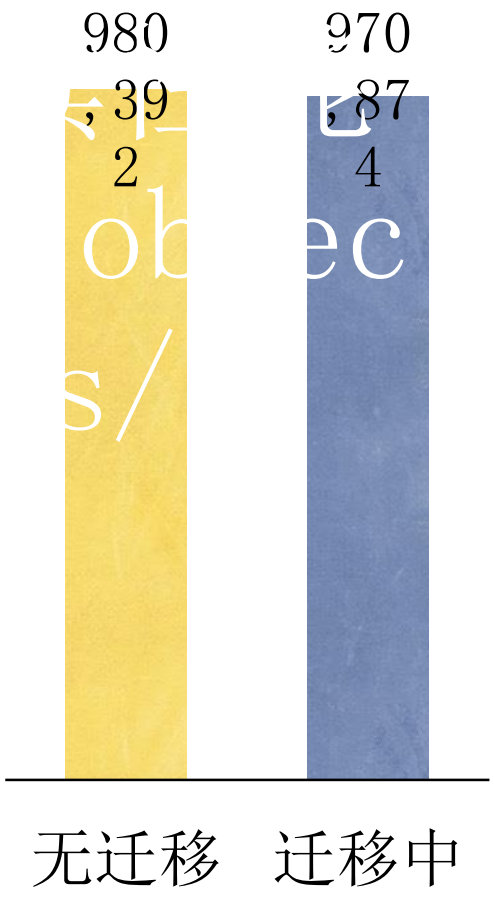


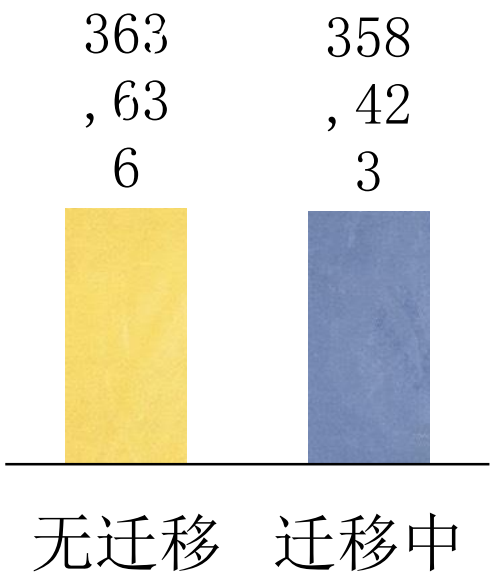
```
SAVEPOINT
UPDATE oldTable SET content = "Hello, WCDB"
    WHERE identifier > 10

UPDATE newTable SET content = "Hello, WCDB"
    WHERE identifier > 10
RELEASE
```

```
DELETE newTable WHERE identifier > 10
```

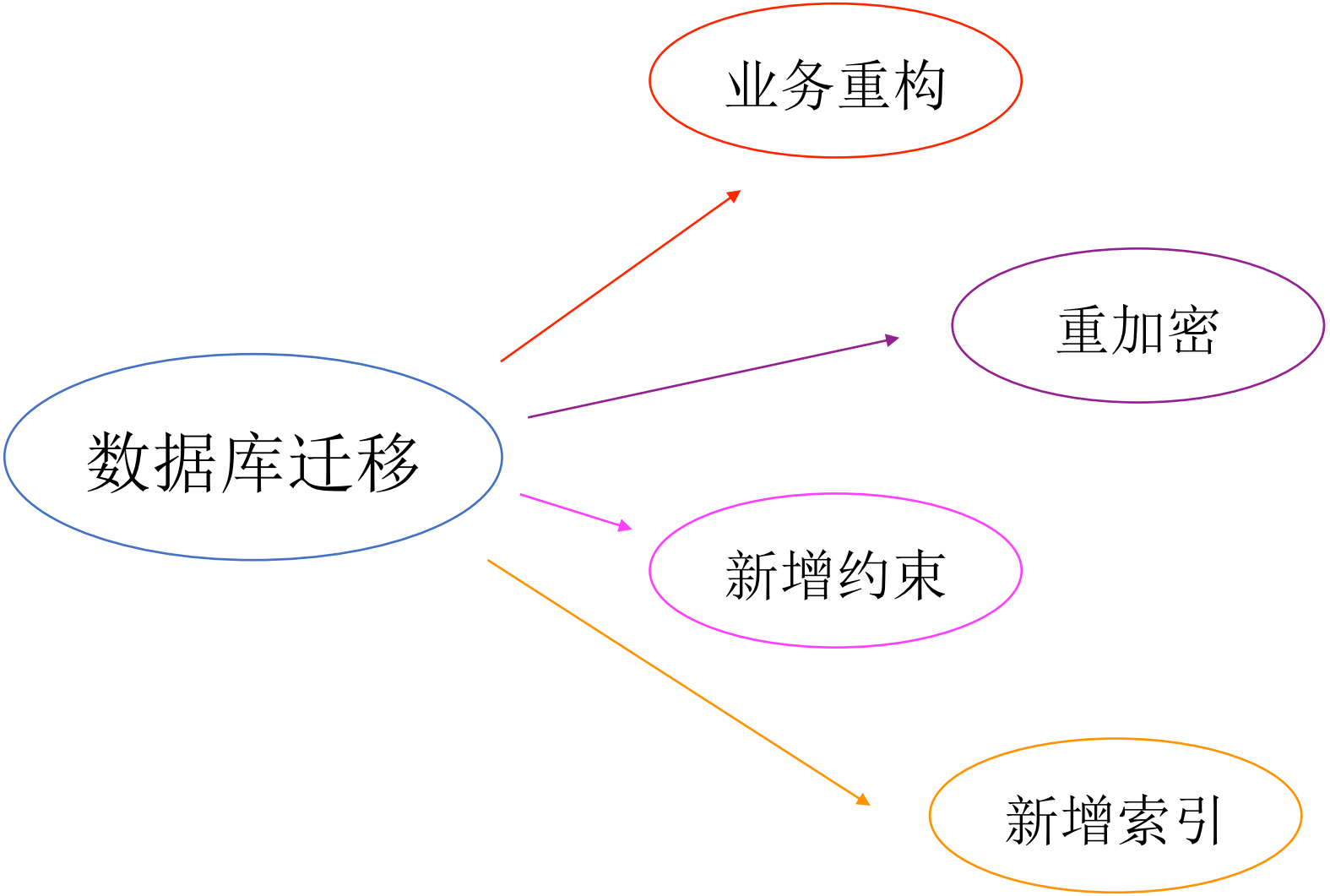


```
SAVEPOINT
UPDATE FROM oldTable WHERE identifier > 10

UPDATE FROM newTable WHERE identifier > 10
RELEASE
```


980,392
970,874
无迁移
迁移中


363,636
358,423
无迁移
迁移中

```
-   var database = Database(withPath: oldPath)
+   var database = Database(withPath: newPath)
+   database.filterMigration({ (migrationInfo) in
+     if migrationInfo.targetTable == "newTable" {
+       migrationInfo.sourceTable = "oldTable"
+       migrationInfo.sourceDatabase = oldPath
+     }
+   })
-   var *table =
-     database.getTable(named: "oldTable");
+     database.getTable(named: "newTable");
```


业务重构
重加密
数据库迁移
新增约束
新增索引

- 代码耦合
- 并发性差
- 健壮性差

可规避：

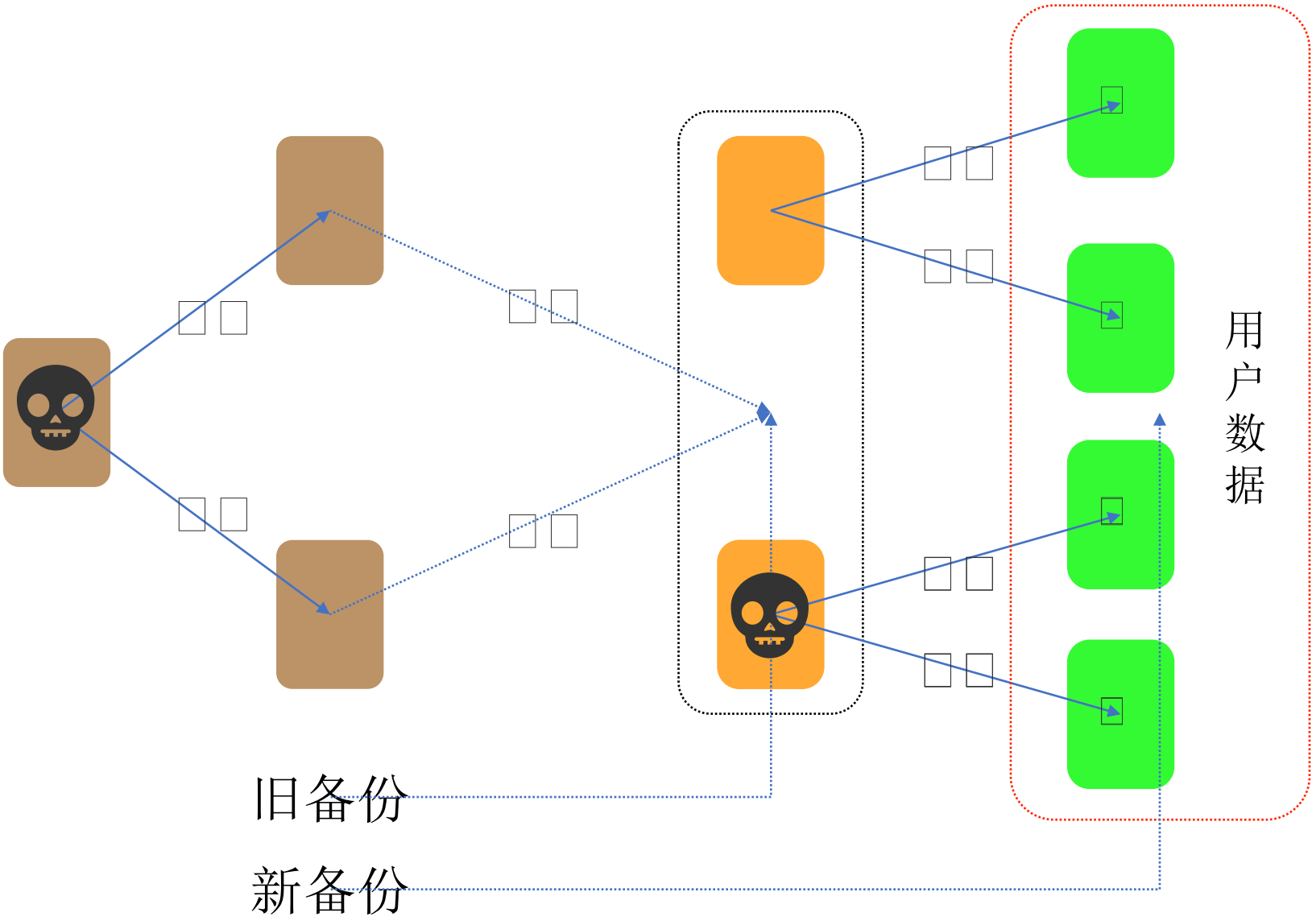
SQLite 使用不当
……

难以规避：

内存野指针
文件错写
……

无法规避：

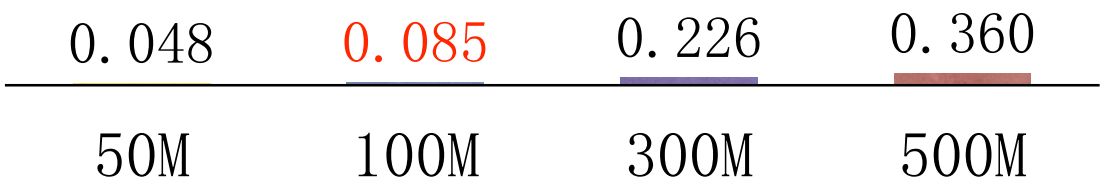
文件系统错误
磁盘损坏
……


master 表
用户表
用户数据
旧备份
新备份

I/O → mmap

- 分段
- 按需
- LRU
- 引用计数

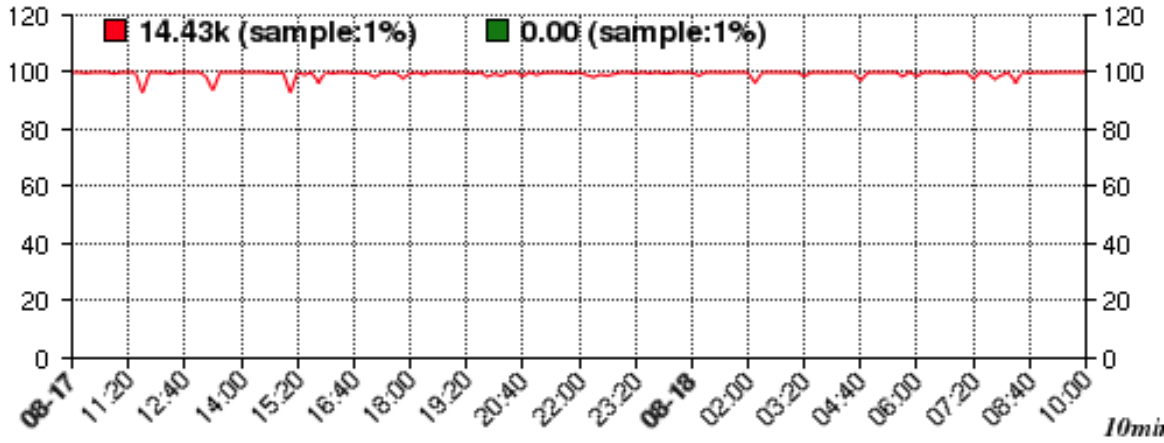
内存 → 事务化备份（文件不会变化）



$$修复率 = \frac{修复成功的页}{应该修复的页}$$

$$= \frac{修复的叶子页数 * 修复的\ Cell\ 权重}{叶子页或未知页的总数}$$

$$= 96.71\%$$



主db修复率 (312:23)

微信终端开源数据库 WCDB

无感迁移方案

- 非阻塞迁移
- 方便接入
- 性能损耗小

新修复方案

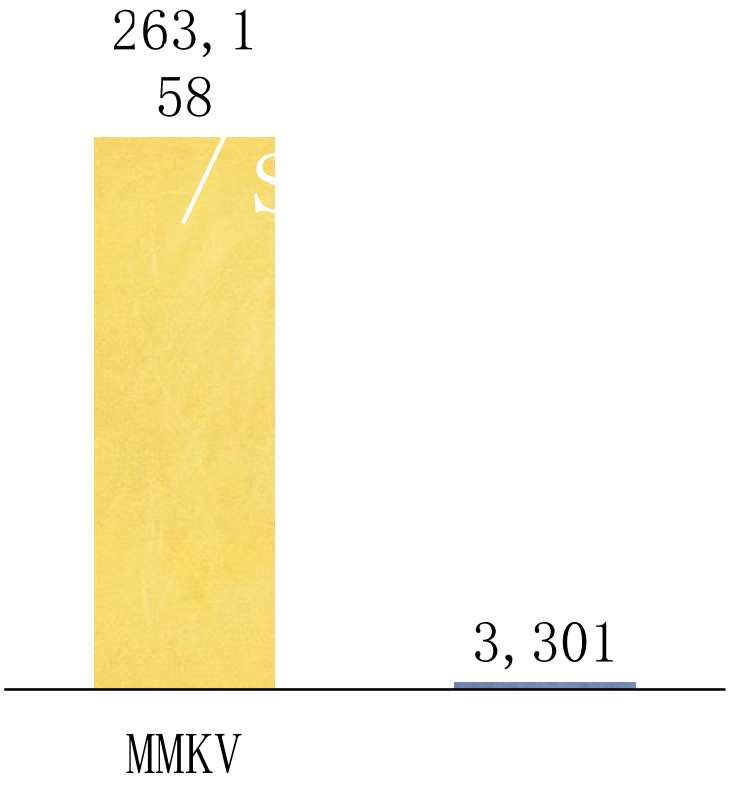
- 高效
- 修复率高

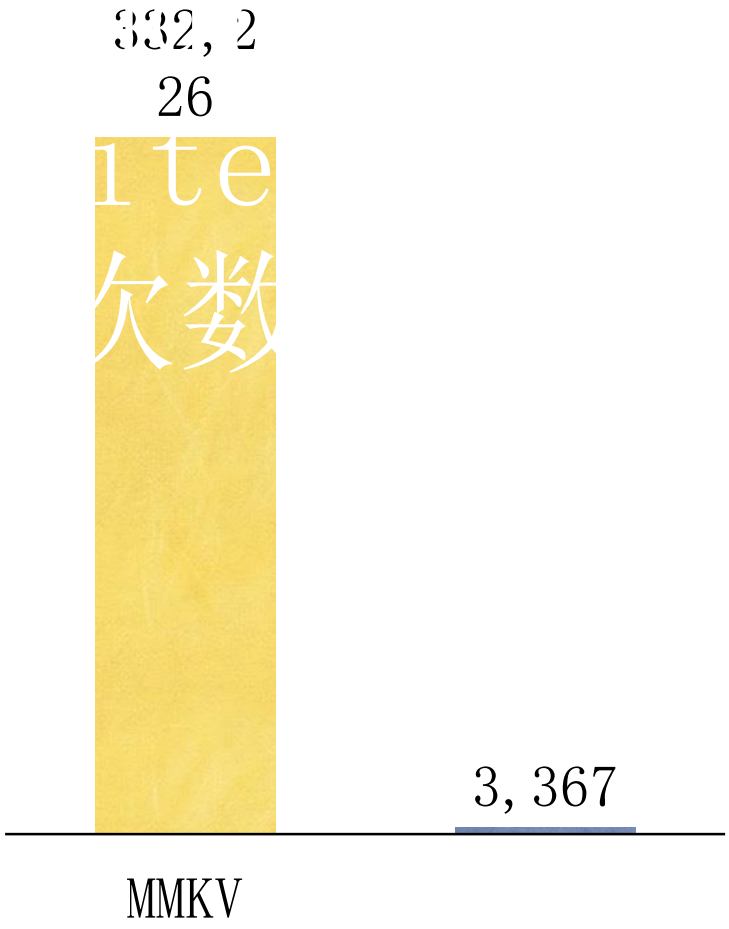
https://github.com/Tencent/ WCDB

https://github.com/Tencent/ MMKV

MMKV

- key-value 组件
- mmap/protobuf
- 微信上稳定使用 3 年多
- 18 年 9 月对外开源
- 支持 iOS/MacOS/Android
- 小而美（二进制 30KB 左右）


263,158
/s
3,301
MMKV


332,226
ite
次数
3,367
MMKV

```objc
// iOS

MMKV *mmkv = [MMKV defaultMMKV];

[mmkv setObject:@"Hello, MMKV!"  forKey:@"string"];
NSString *str = [mmkv getObjectOfClass:NSString.class
                                forKey:@"string"];
```

```java
// Android

MMKV kv = MMKV.defaultMMKV();

kv.encode("string", "Hello, MMKV!");
String str = kv.decodeString("string");
```

微信终端团队 - WeMobileDev

- https://github.com/Tencent/WCDB
- https://github.com/Tencent/MMKV
- https://github.com/Tencent/Mars
- https://github.com/Tencent/Tinker
- https://github.com/Tencent/SOTER
- ……