MQ结构

[1. 前言 1](#_Toc1909)

[2. 概览 2](#_Toc3961)

[3. 文件存储 2](#_Toc31617)

[3.1. 文件服务 2](#_Toc25503)

[3.2. 存储模型 2](#_Toc15699)

[3.3. 缓存服务 3](#_Toc30206)

[4. 表级服务 3](#_Toc18139)

[5. SQL解析 3](#_Toc16147)

[6. WAL 4](#_Toc17324)

[7. 事务 4](#_Toc9487)

[8. 执行计划 4](#_Toc20113)

[9. 结束语 4](#_Toc22400)

[10. 附件 4](#_Toc8451)

# 前言

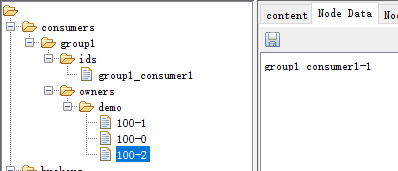
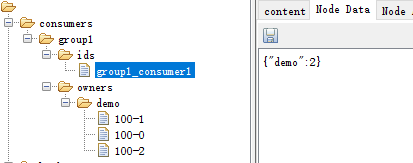
2017年下半年，因工作原因利用业余时间对数据库学习研究的较多。偶然在github上找到了一个Java版的数据库（教学用途），代码简洁清晰，仔细阅读了其代码后，将其核心技术点记录下来。

# 概览

此数据库的核心模块包括SQL解析，表管理，事务管理，WAL和存储服务。

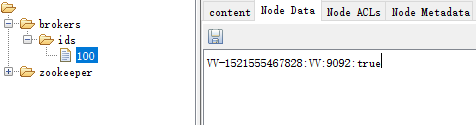


# Consumer



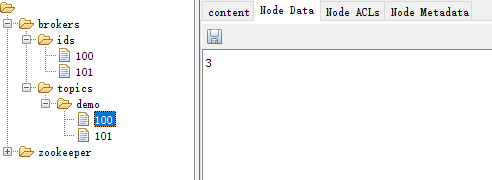
# Broker

Jafka启动后在ZK上注册节点/brokers/ids/100，节点内容为creatorId:host:port:autocreated。其中creatorId为host-currentTime。



# Producer

Producer启动后，创建的BrokerPartitionInfo持有各topic对应的partition(brokerId,partionId)集合。其通过监听节点/brokers/topics，实时更新此映射关系。然后通过/brokers/ids可以获取到当前在线的所有broker信息。下图中topic为demo的消息发送到broker 100和borker 101两个服务实例上，且每个broker下有三个分区（配置的），因此一条消息在没有给定key值时，随机被分在这6个分区上。



文件结构

# 结束语

文章的结束语部分，一般是对文章内容或技术的总结，或其它启发性思考等。

# 附件

<https://github.com/williamxww/columndb-demo>