Canal开源产品介绍

七锋 @ taobao

Agenda

1. 产生背景
2. 项目介绍
3. 周边产品
4. roadmap

产生背景

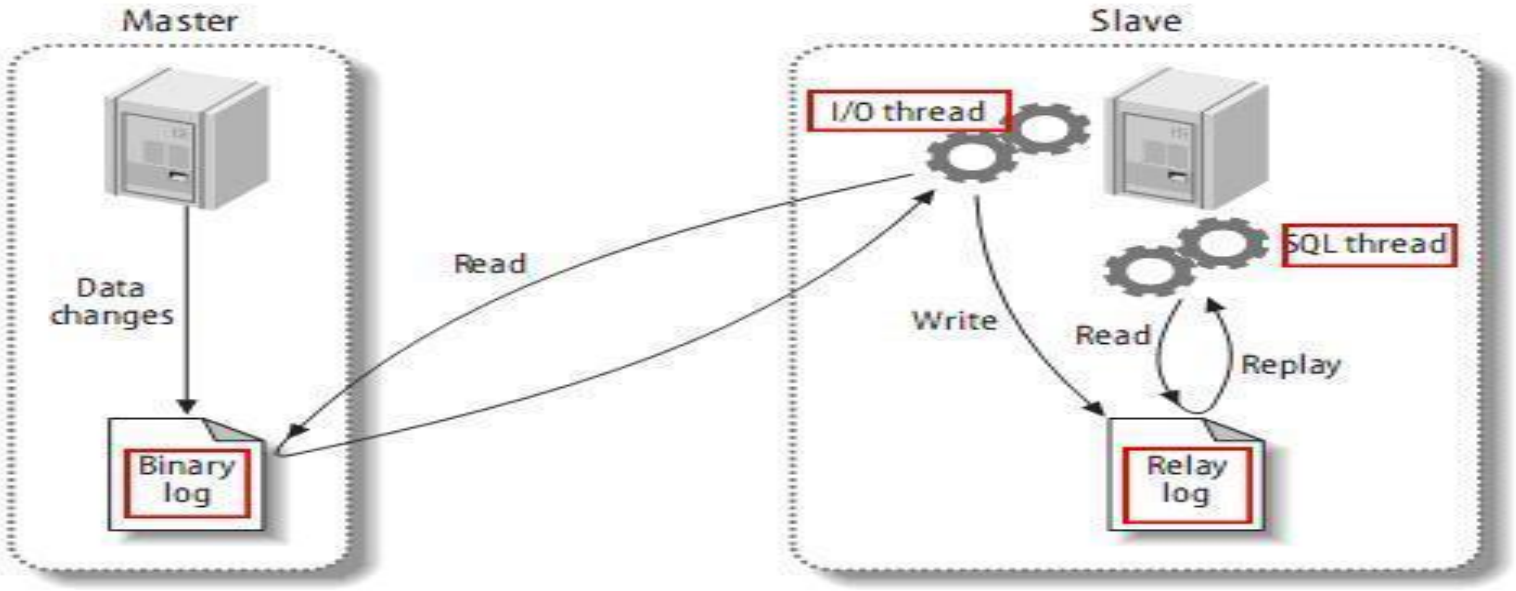
早期，阿里巴巴B2B公司因为存在杭州和美国双机房部署，存在跨机房同步的业务需求，当时早期的数据库同步业务，主要是基于trigger的方式获取增量变更。

从2010年开始，阿里系公司开始逐步的尝试基于数据库的日志解析，获取增量变更进行同步，由此衍生出了增量订阅&消费的业务，从此开启了一段新纪元.

Canal介绍

- 名称：canal [kə'næl]
- 译意：水道/管道/沟渠
- 语言：纯java开发
- 定位：　基于数据库增量日志准实时解析，提供增量数据订阅&消费(目前开源版本主要支持了mysql)

Mysql同步原理



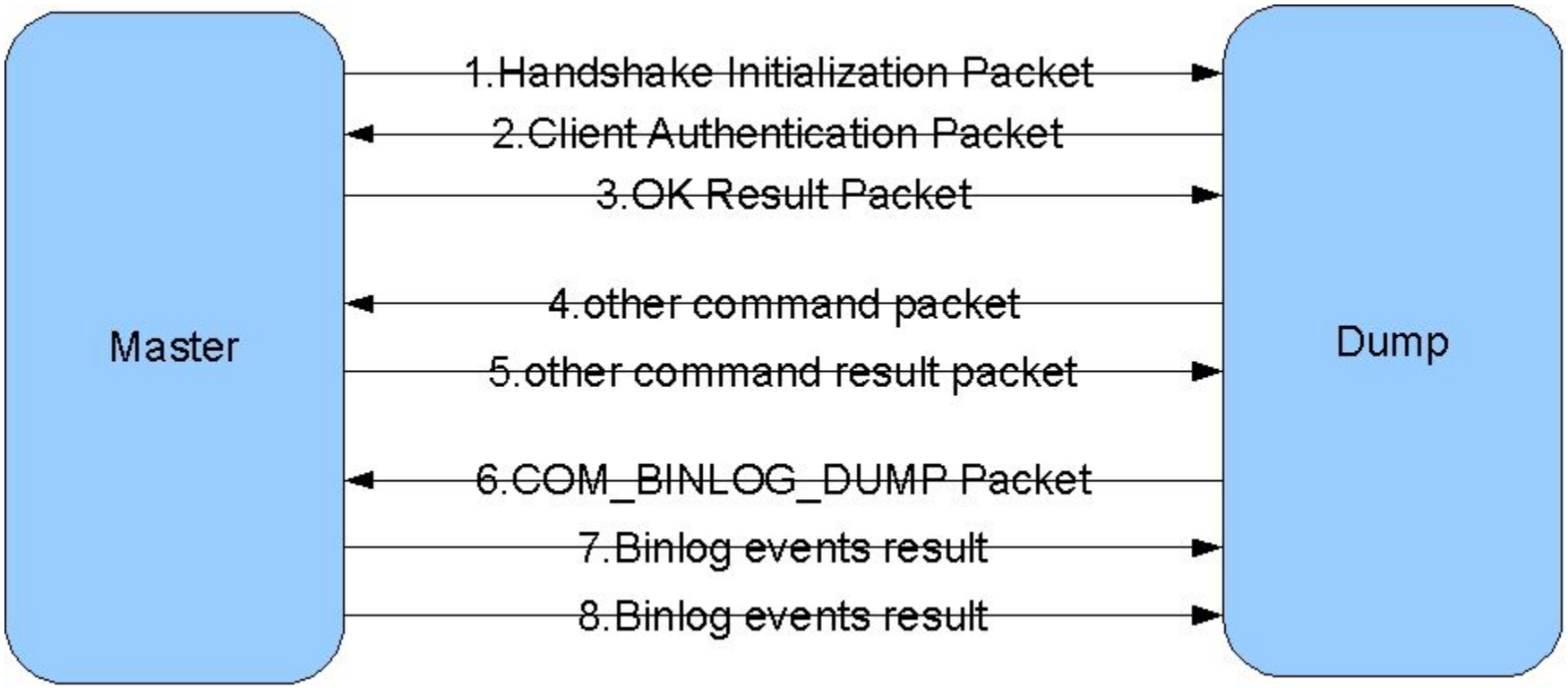
Mysql Slave同步原理:

a. I/O thread接收binlog

b. SQL thread执行变更

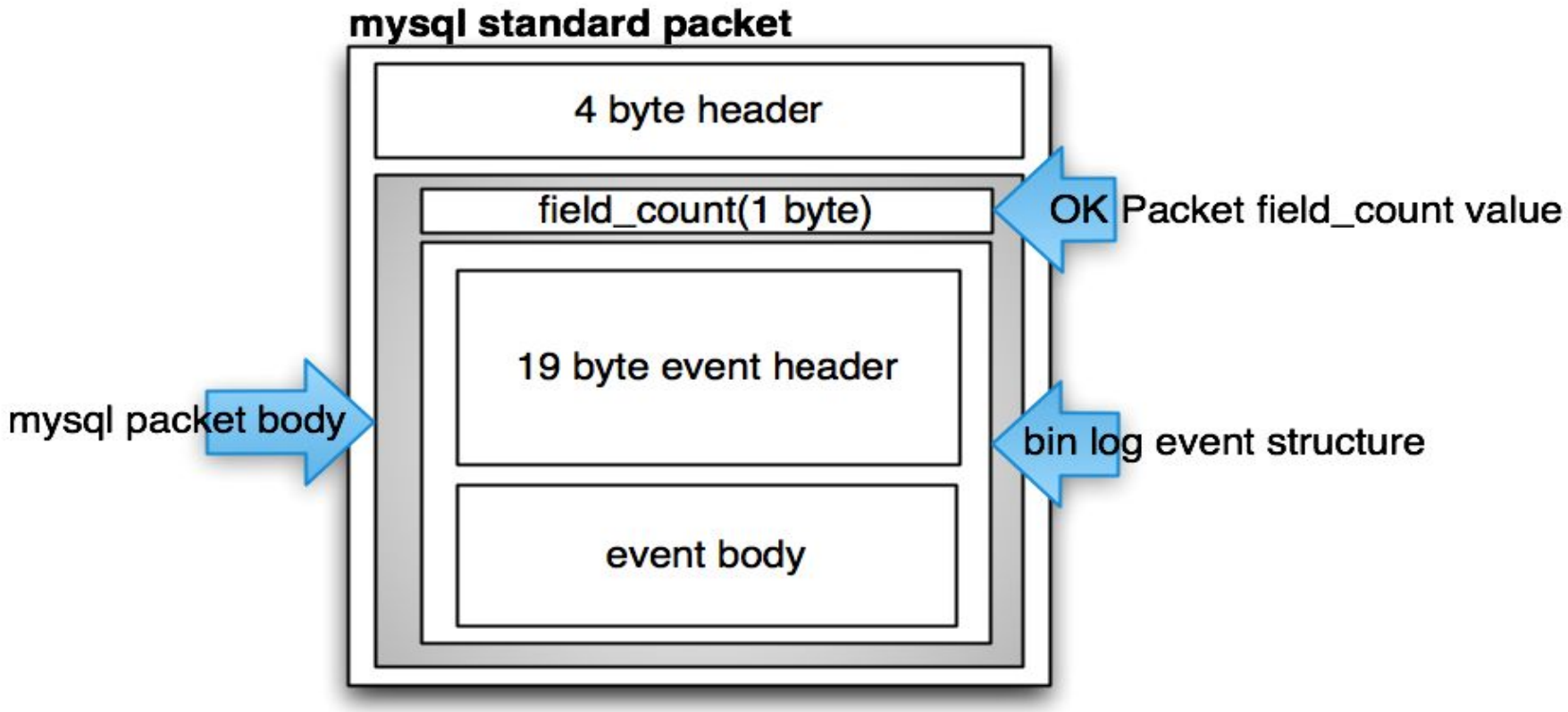
Mysql同步原理

Binlog Dump交互



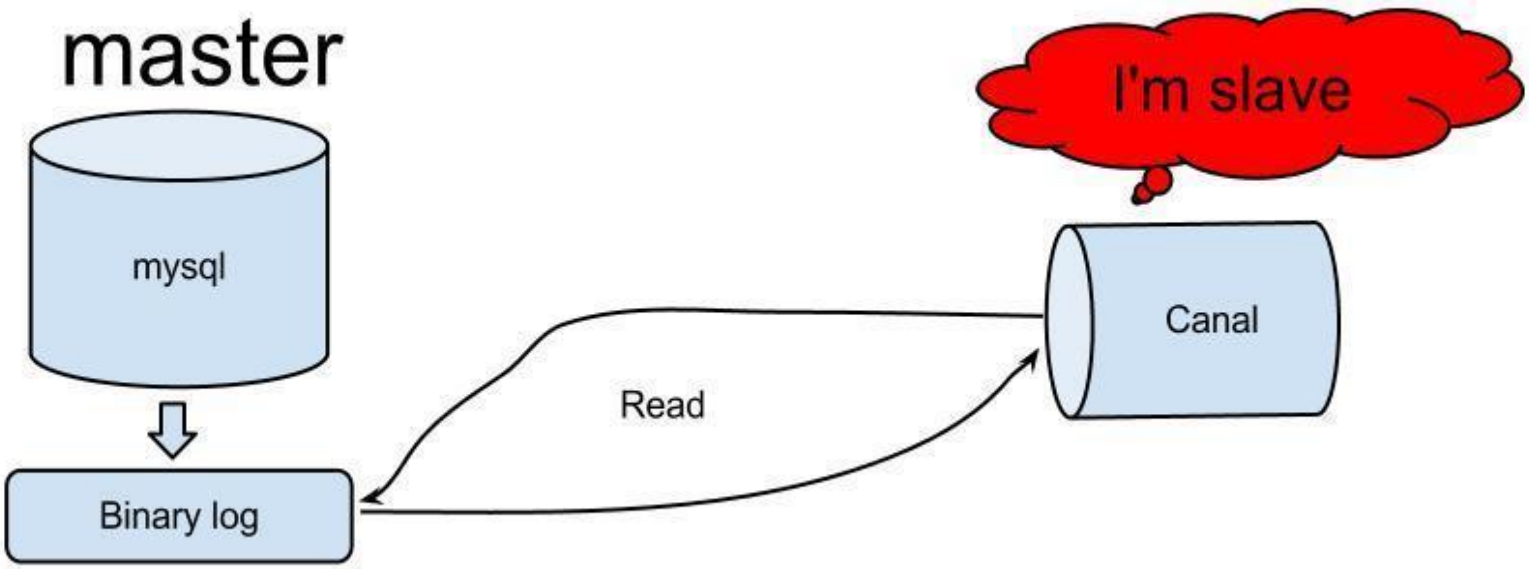
Mysql同步原理

Binlog Event Structure Demonstration



更多协议参考：http://dev.mysql.com/doc/internals/en/binary-log.html

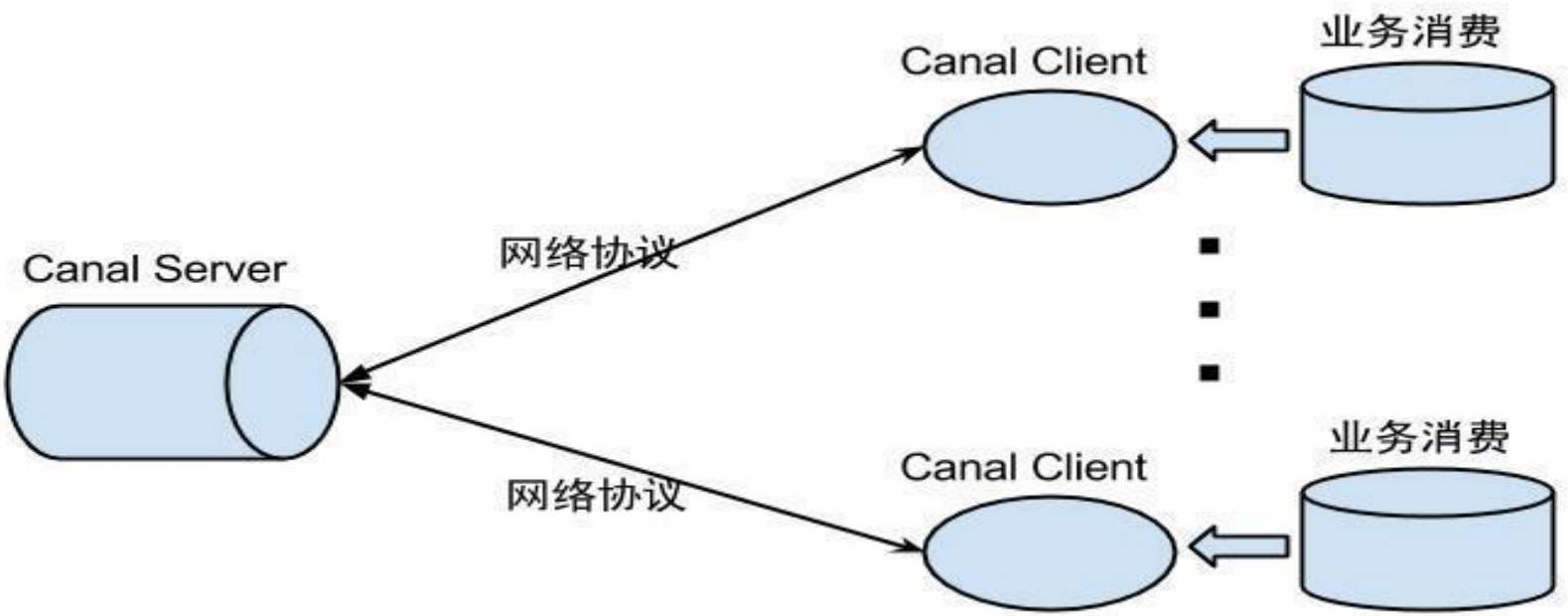
Canal工作原理



实时增量数据获取原理：

模拟slave的交互协议，伪装自己为mysql slave (类似于I/O thread线程)

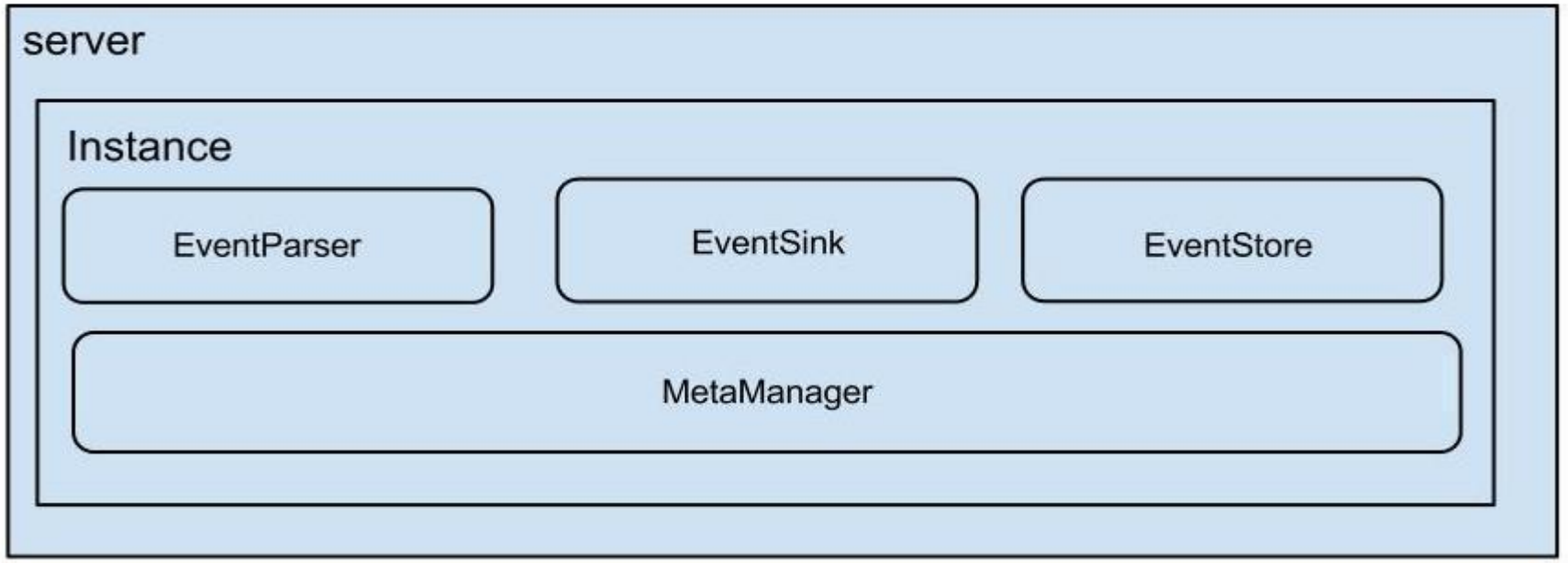
Canal工作原理



数据消费原理:

基于网络协议，提供数据订阅&消费，类似于SQL Thread实现业务自定义

Canal Server模块



1. server代表一个canal server运行实例，对应于一个jvm
2. instance对应于一个数据队列（1个server对应0..n个instance)

Canal Server模块

server模块:

基于netty网络处理 + protobuf数据传输格式

instance模块:

a. eventParser

增量数据解析器，目前仅支持mysql

b. eventSink

数据过滤，加工，分发的工作

c. eventStore

数据存储，目前1.0.6仅支持memory，file存储开发中

d. metaManager

增量订阅&消费信息管理器

Canal Server配置示例


```
vi conf/example/instance.properties
```

```
#################################################
## mysql serverId
canal.instance.mysql.slaveId = 1234

# position info
canal.instance.master.address = 127.0.0.1:3306 #改成自己的数据库地址
canal.instance.master.journal.name =
canal.instance.master.position =
canal.instance.master.timestamp =

#canal.instance.standby.address =
#canal.instance.standby.journal.name =
#canal.instance.standby.position =
#canal.instance.standby.timestamp =

# username/password
canal.instance.dbUsername = canal  #改成自己的数据库信息
canal.instance.dbPassword = canal  #改成自己的数据库信息
canal.instance.defaultDatabaseName =   #改成自己的数据库信息
canal.instance.connectionCharset = UTF-8  #改成自己的数据库信息

# table regex
canal.instance.filter.regex = .*\\..*

#################################################
```

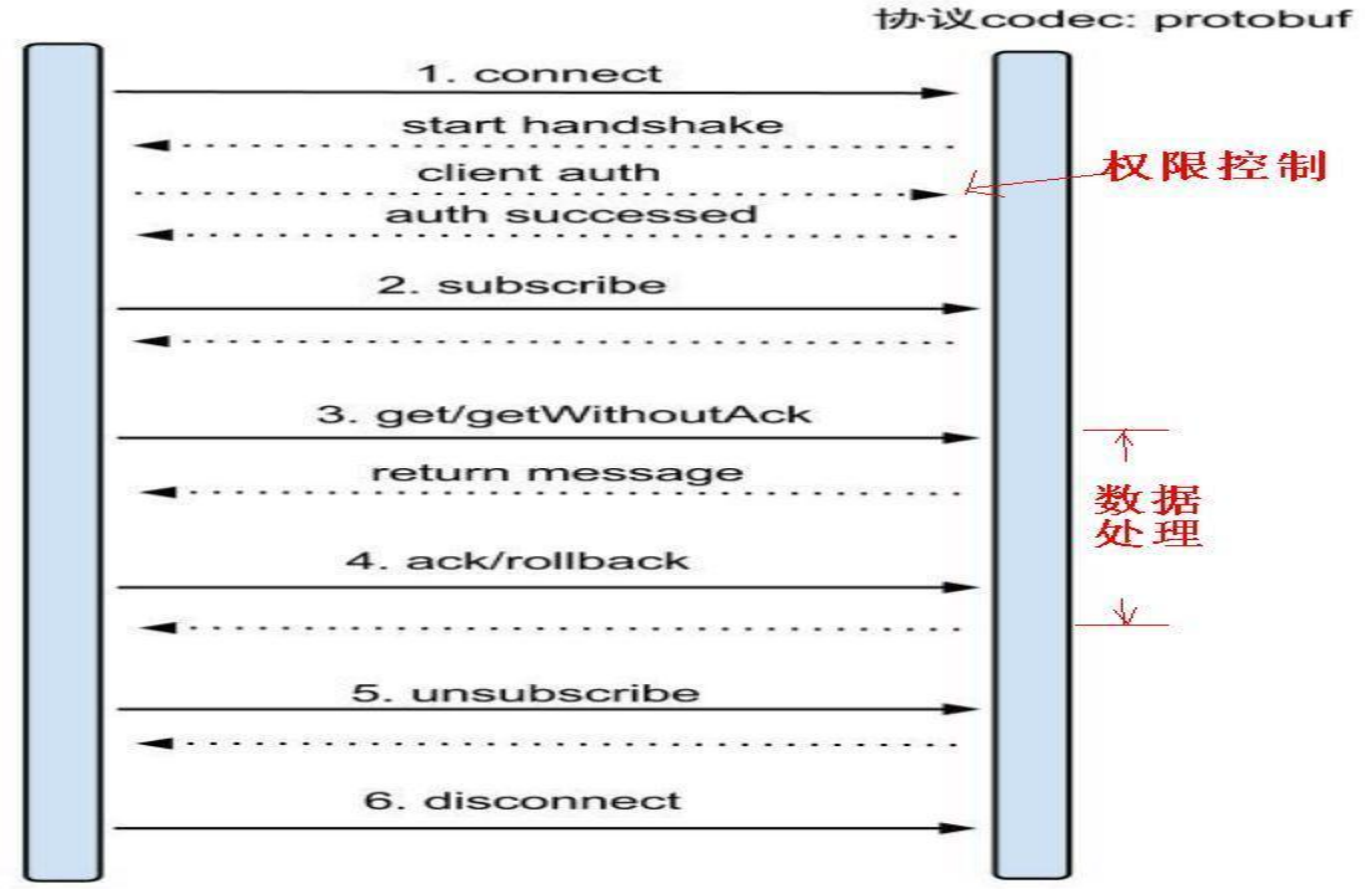
类似于mysql serverId
保证唯一

启动的初始位点

数据库信息

声明关注的表

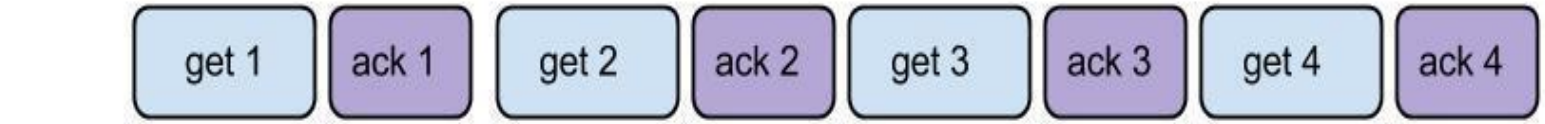
Client/Server交互



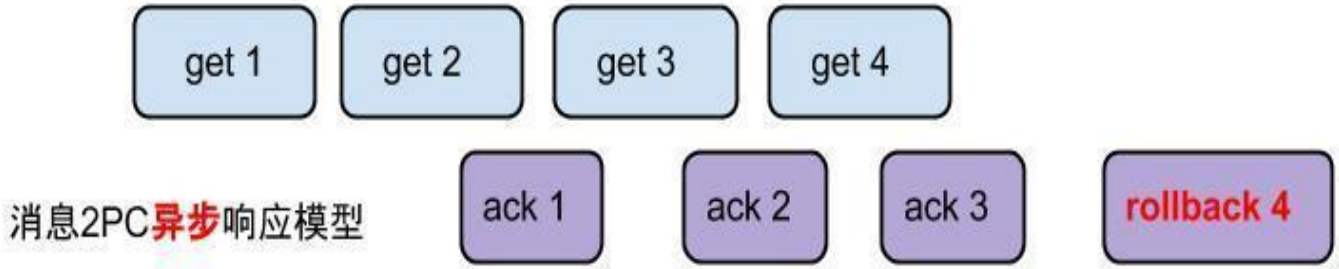
1. subscribe/unsubscribe只在第一次需要
2. subscribe允许重复调用，每次提交新的filter

Client/Server交互

消息2PC串行响应模型



消息2PC**异步**响应模型



设计出发点：

1. 数据处理 成功概率(ack)>>出错概率(rollback)
2. 数据get和数据处理并行
3. get/ack双工，减少ack时间消耗

数据对象格式

数据对象格式： EntryProtocol.proto

```
Entry
    Header
        logfileName [binlog文件名]
        logfileOffset [binlog position]
        executeTime [binlog里记录变更发生的时间戳]
        schemaName [数据库实例]
        tableName [表名]
        eventType [insert/update/delete类型]
    entryType   [事务头BEGIN/事务尾END/数据ROWDATA]
    storeValue  [byte数据,可展开，对应的类型为RowChange]

RowChange
    isDdl       [是否是ddl变更操作，比如create table/drop table]
    sql     [具体的ddl sql]
    rowDatas    [具体insert/update/delete的变更数据，可为多条，1个binlog event事件可对
        beforeColumns [Column类型的数组]
        afterColumns [Column类型的数组]

Column
    index       [column序号]
    sqlType     [jdbc type]
    name        [column name]
    isKey       [是否为主键]
    updated     [是否发生过变更]
    isNull      [值是否为null]
    value       [具体的内容，注意为文本]
```

binlog位点信息

变更前后数据

字段信息

Canal Client示例


```
int batchSize = 5 * 1024;
while (running) {
    try {
        MDC.put("destination", destination);
        connector.connect();
        connector.subscribe("");
        while (running) {
            Message message = connector.getWithoutAck(batchSize); // 获取指定数量的数据
            // ackQueue.add(batchId);//添加到ack处理队列，由异步线程进行ack，可直接获取下一批数据
            long batchId = message.getId();
            int size = message.getEntries().size();
            if (batchId == -1 || size == 0) { //空数据，可选择进行sleep
                // try {
                // Thread.sleep(1000);
                // } catch (InterruptedException e) {
                // }
            } else {
                printSummary(message, batchId, size);
                printEntry(message.getEntries());
            }

            connector.ack(batchId); // 提交确认
            // connector.rollback(batchId); // 处理失败，回滚数据
        }
    } finally {
        connector.disconnect();
        MDC.remove("destination");
    }
}
```

数据连接&订阅

数据处理

Canal Client示例


```
    RowChange rowChage = null;
    try {
        rowChage = RowChange.parseFrom(entry.getStoreValue());
    } catch (Exception e) {
        throw new RuntimeException("parse event has an error , data:" + entry.toString(), e);
    }

    EventType eventType = rowChage.getEventType();
    long executeTime = entry.getHeader().getExecuteTime();
    long delayTime = new Date().getTime() - executeTime;
    logger.info(row_format, new Object[] { entry.getHeader().getLogfileName(),
            String.valueOf(entry.getHeader().getLogfileOffset()), entry.getHeader().getSchemaName(),
            entry.getHeader().getTableName(), eventType, String.valueOf(entry.getHeader().getExecuteTime()),
            String.valueOf(delayTime) });

    if (eventType == EventType.QUERY || rowChage.getIsDdl()) {
        logger.info(" sql ----> " + rowChage.getSql() + SEP);
    }

    for (RowData rowData : rowChage.getRowDatasList()) {
        if (eventType == EventType.DELETE) {
            printColumn(rowData.getBeforeColumnsList());
        } else if (eventType == EventType.INSERT) {
            printColumn(rowData.getAfterColumnsList());
        } else {
            printColumn(rowData.getAfterColumnsList());
        }
    }
}
```

数据反序列化

DDL数据，打印SQL

DML数据，打印字段信息

基于Canal能做什么？

1. 数据库镜像&备份
2. 异构数据库同步
3. 多地机房
4. 二级索引
5. 搜索引擎增量build
6. 数据库操作审计
7. 业务cache刷新
8. 价格变化等重要业务变更消息

…… ……

Canal目前使用情况

- Alibaba 200+ 数据解析任务
  a. 数据规模:6亿+
  b. 支持mysql5.1.40/48 , mysql 5.5.18

2. Canal使用群人数已超70+

类似开源产品

1. linkedin databus
   https://github.com/linkedin/databus
2. tungsten-replicator
   http://code.google.com/p/tungsten-replicator/
3. open-replicator
   http://code.google.com/p/open-replicator/

Roadmap

1. topic模式支持
2. web管理系统
   a.Auth权限管理
   b.监控体系
3. 新数据源接入
   a. Hbase增量
4. client代码共建(共性业务场景)
   a.数据库同步
   b.nosql同步(如hbase)

相关资料

1. canal wiki

   https://github.com/alibaba/canal/wiki

2. mysql binary log

   http://dev.mysql.com/doc/internals/en/binary-log.html

3. mysql replication-protocol

   http://dev.mysql.com/doc/internals/en/replication-protocol.html

问题反馈

1. qq交流群：161559791
2. 邮件交流：jianghang115@gmail.com
3. 新浪微博：agapple0002
4. 报告issue：issues

最后

Q & A