**1.4.0 发行日志**

* MessageProducer添加callback方式的异步发送方法：

/\*\*

\* 异步发送消息，在指定时间内回调callback,此模式下无法使用事务

\*

\* @param message

\* @param cb

\* @param time

\* @param unit

\* @since 1.4

\*/

public void sendMessage(Message message, SendMessageCallback cb, long time, TimeUnit unit);

/\*\*

\* 异步发送消息，在默认时间内（3秒）回调callback，此模式下无法使用事务

\*

\* @param message

\* @param cb

\* @since 1.4

\*/

public void sendMessage(Message message, SendMessageCallback cb);

* 修改offset在zookeeper上的存储格式，从单纯的offset修改为msgId-offset，主要是为了做同步复制的实现。客户端升级，同一个分组必须批量升级，否则会遇到兼容问题。
* ZkConfig引入zkRoot配置项，可以配置meta的zk根路径，默认仍然为/meta
* ZkConfig引入zkEnable配置项，false表示不注册broker到zk，默认为true
* ConsumerConfig废弃maxDelayFetchTimeWhenExceptionInMills，统一使用maxDelayFetchTimeInMills。
* Recover只在第一次创建consumer的时候才启动。
* DeletePolicy可以使用秒和分钟为单位，如"delete,72s"，表示使用删除策略并保存最长72秒，72m表示72分钟。默认没有单位或者h表示使用小时。
* 服务端添加deleteWhen配置项，全局和topic都可以配置。这是一个cron表达式，使用quartz的cron语法，指定删除策略的执行周期，默认仍然为早晚6点。
* 服务端添加quartzThreadCount配置项，用于设置quartz的线程池大小，默认为5。
* 服务端添加dataLogPath，配置事务日志的单独路径。
* 服务端的配置项dataPath，现在可以被topic配置所覆盖，也就是不同topic可以配置不同的数据路径,对于使用多块盘的机器有利。
* 客户端引入了新的负载均衡策略，抽象出LoadBalanceStrategy接口并实现了ConsisHashStrategy的一致性哈希策略。ConsisHashStrategy可以减少机器变更带来的重新负载均衡，但是可能会造成消费分布的不均匀，暂时不推荐使用。LoadBalanceStrategy可以通过ConsumerConfig的loadBalanceStrategyType配置项来设置，目前支持DEFAULT和CONSIST。
* 服务端对于设置unflushThreshold<=0的时候，也就是严格每次刷盘模式，使用group commit来提升吞吐量。经测试表明，吞吐量相比于非严格刷盘模式没有下降，但是服务端load会上升。
* 异步复制相关配置从server.ini迁移到slave.properties文件中，异步复制slave作为插件启动。
* 实现同步复制的高可用方案，具体设计和配置 TODO
* log4j appender实现，可以透明地使用log4j写日志来发送消息。配置和使用(TODO)
* hdfs writer，将meta消息接入hadoop。配置和使用(TODO)
* 服务端和客户端内部的代码重构，重点减轻对zk的压力。

**1.3.2 发行日志**

* MessageProducer添加setDefaultTopic方法：

/\*\*

\* 设置发送消息的默认topic，当发送的message的topic没有找到可用broker和分区的时候，选择这个默认topic指定的broker发送

\* 。调用本方法会自动publish此topic。

\*

\* @param topic

\*/

public void setDefaultTopic(String topic);

* 减少在本地恢复消息的时候，MessageListener为null导致NPE的日志打印
* 发送消息重试次数减少为1，也就是不重试。
* Partition添加一系列方法用于设置和获取属性:

/\*\*

\* 设置属性，覆盖已有的任何关联值

\*

\* @param key

\* @param value

\*/

public Object setAttribute(final String key, final Object value)

/\*\*

\* 当key的value不存在的时候，关联key到传入的value，此操作是原子的

\*

\* @see java.util.concurrent.ConcurrentHashMap#putIfAbsent(Object, Object)

\* @param key

\* @param value

\*/

public Object setAttributeIfAbsent(final String key, final Object value)

/\*\*

\* 获取key指定的属性

\*

\* @param key

\* @return

\*/

public Object getAttribute(final String key)

/\*\*

\* 返回属性的key集合，弱一致性

\*

\* @return

\*/

public Set<String> attributeKeySet()

/\*\*

\* 移除key指定的属性

\*

\* @param key

\* @return

\*/

public Object removeAttribute(final String key)

/\*\*

\* 清空所有属性

\*/

public void clearAttributes()

## 1.3.1

修复消费端消息存入recover store时可能存在重复key而造成消费端卡住的问题

**1.3**

修复消费者负载均衡bug  
修复http读取数据偏移量的bug  
增加消息处理失败存本地时打印日志  
支持异步单向发送消息  
修复发送带有空字符串属性消息的bug  
增加消息处理失败时可以保存到服务端的选择  
fetch失败时(例如无可用连接)，更新延时,以免客户端打印过多异常日志  
增加拉取数据不足重试次数的阈值参数,把消息处理失败重试跟拉取数据不足重试分开设置（tddl使用中发现的问题）  
修复拥有相同brokerId的同类型服务器启动时，注册zk失败时，影响正在运行的另一个正常server  
修复消费者recover的bug  
发送顺序消息增加选项,当发送失败时可以选择保存本地还是返回错误.  
完善发送recover时可能存在的问题

**1.2**

支持http协议  
支持本地事务和XA事务  
增加消费者批量接收事务  
增加针对每个topic配置定时和定量刷硬盘  
增加动态改变topic的统计属性.topic没变的情况下动态增加和减少对某topic的统计;  
修复每次重新加载配置时存放Pattern的Set一直会增加的bug  
修复客户端cli\_get\_failed统计的bug  
修复定时flush bug；  
修复客户端拉取数据量大小自适应bug；  
改善限制消息大小的代码  
修复每次重新加载配置时存放Pattern的Set一直会增加的bug  
修复使用remotingClint方式发送统计命令item为空时的解码问题  
修改topic实时统计的输出项(原先全部写成realtime\_put了),增加resultCode项表示topic是否存在  
修复一个关闭sessionFactory zkClient的潜在bug(直连方式时zkClient是为null的)

**1.1**

增加分区总数的配置和PartitionSelector支持  
支持顺序消息的分区迁移,消息保存本地和重试  
增加针对topic配置flush的时间间隔和条数阀值,并支持运行时动态加载改变  
修改了对巨型消息的限制  
消息接收端的批量接收事务  
服务端的一些重构  
修复可能删除不掉过期数据的bug

**1.0.1**

完成HA方案的实现，支持一主多备  
增加服务端启动的插件机制,灵活的支持meta接收其他数据  
新增支持指定每个topic不同的数据删除策略  
增加消息的广播接收方式  
支持动态增加topic  
改变配置文件为ini格式  
修复分区排序bug  
修复客户端本地存储offset的bug  
修复服务器重启时可能误删数据的bug  
修复客户端负载均衡方面的一些缺陷