



图文 113 一条消息写入CommitLog文件之后，如何实时更新索引文件？

101 人次阅读 2020-03-27 11:04:29

详情 评论



猴猫技术

一条消息写入CommitLog文件之后，如何实时更新索引文件？

从零开始
带你成为
MySQL实战优化高手

继爆款专栏《从零开始带你成为JVM实战高手》后，又一重磅力作

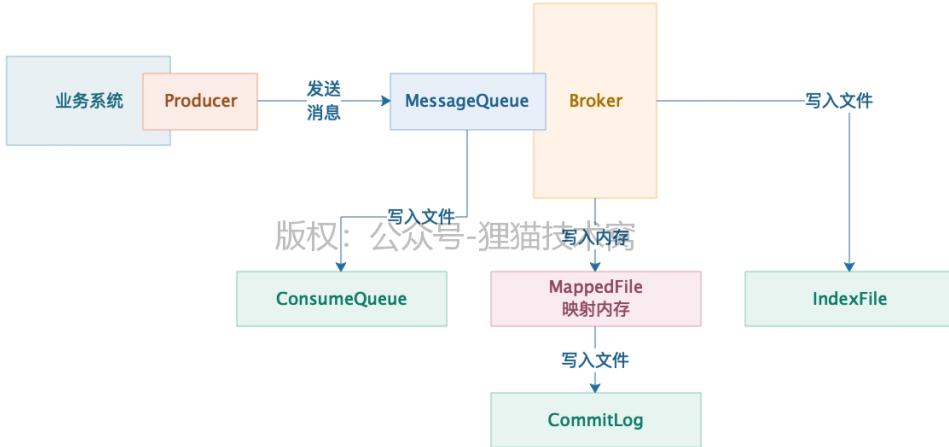
专栏作者：救火队队长
阿里资深技术专家

继《从零开始带你成为JVM实战高手》后，阿里资深技术专家携新作再度出山，重磅推荐：

(点击下方蓝字试听)

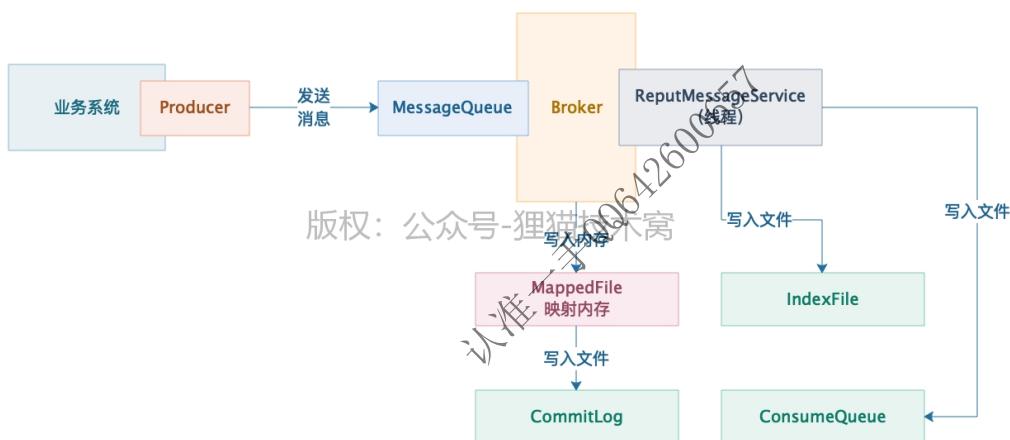
[《从零开始带你成为MySQL实战优化高手》](#)

昨天我们讲到，Broker收到一条消息之后，其实就会直接把消息写入到CommitLog里去，但是他写入刚开始仅仅是写入到MappedFile映射的一块内存里去，后续是根据刷盘策略去决定是否立即把数据从内存刷入磁盘的，我们看下图。



关于这个同步刷盘和异步刷盘的问题，我们后续再讲，今天先来说说，这个消息写入CommitLog之后，然后消息是如何进入ConsumeQueue和IndexFile的。

实际上，Broker启动的时候会开启一个线程，ReputMessageService，他会把CommitLog更新事件转发出去，然后让任务处理器去更新ConsumeQueue和IndexFile，如下图。



我们看下面的源码片段，在DefaultMessageStore的start()方法里，在里面就是启动了这个ReputMessageService线程。

这个DefaultMessageStore的start()方法就是在Broker启动的时候调用的，所以相当于是Broker启动就会启动这个线程。

```

1 this.reputMessageService.setReputFromOffset(maxPhysicalPosInLogicQueue);
2 this.reputMessageService.start();

```

下面我们看这个ReputMessageService线程的运行逻辑，源码片段如下所示。

```

1 @Override
2 public void run() {
3     DefaultMessageStore.log.info(this.getServiceName() + " service started");
4
5     while (!this.isStopped()) {
6         try {
7             Thread.sleep(1);
8             this.doReput();
9         } catch (Exception e) {
10            DefaultMessageStore.log.warn(this.getServiceName() + " service has exception. ", e);
11        }
12    }
13
14    DefaultMessageStore.log.info(this.getServiceName() + " service end");
15 }

```

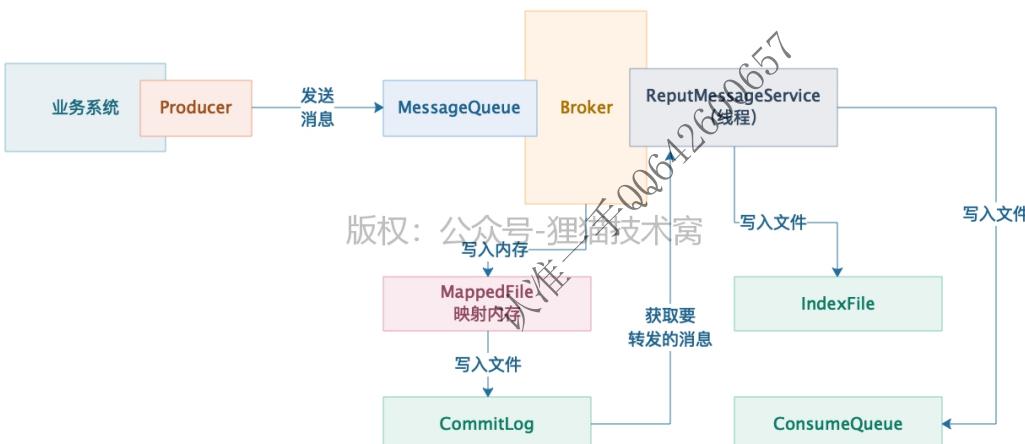
也就是说，在这个线程里，每隔1毫秒，就会把最近写入CommitLog的消息进行一次转发，转发到ConsumeQueue和IndexFile里去，通过的是doReput()方法来实现的，我们再看doReput()方法里的实现逻辑，先看下面源码片段。

```

1 DispatchRequest dispatchRequest = DefaultMessageStore.this.commitLog
2     .checkMessageAndReturnSize(result.getByteBuffer(), false, false);

```

这段代码意思非常的清晰明了，就是从commitLog中去获取到一个DispatchRequest，拿到了一份需要进行转发的消息，也就是从CommitLog中读取的，我们画在下面示意图里



接着他就会通过下面的代码，调用doDispatch()方法去把消息进行转发，一个是转发到ConsumeQueue里去，一个是转发到IndexFile里去

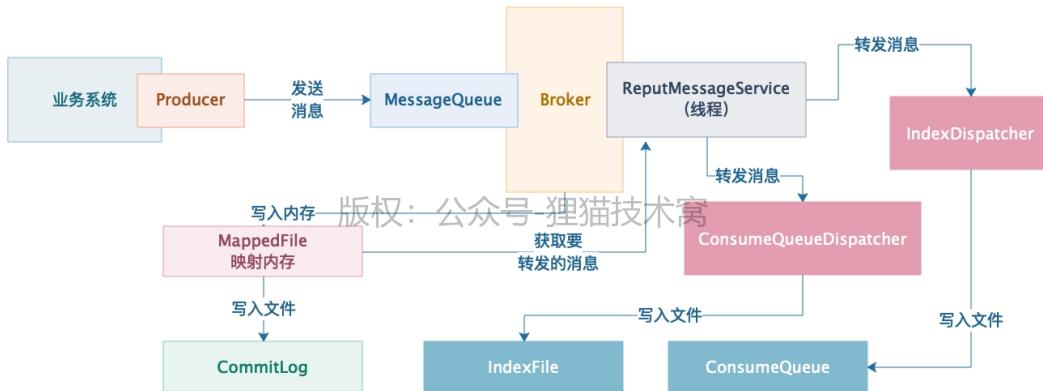
大家看下面的源码片段，里面走了CommitLogDispatcher的循环

```

1 public void doDispatch(DispatchRequest req) {
2     for (CommitLogDispatcher dispatcher : this.dispatcherList) {
3         dispatcher.dispatch(req);
4     }
5 }

```

实际上正常来说这个CommitLogDispatcher的实现类有两个，分别是CommitLogDispatcherBuildConsumeQueue和CommitLogDispatcherBuildIndex，他们俩分别会负责把消息转发到ConsumeQueue和IndexFile，我画在下图中：



接着我们看一下ConsumeQueueDispatche的源码实现逻辑，其实非常的简单，就是找到当前Topic的messageQueueId对应的一个ConsumeQueue文件

一个MessageQueue会对应多个ConsumeQueue文件，找到一个即可，然后消息写入其中。

```

1 public void putMessagePositionInfo(DispatchRequest dispatchRequest) {
2     ConsumeQueue cq = this.findConsumeQueue(dispatchRequest.getTopic(), dispatchRequest.getQueueId());
3     cq.putMessagePositionInfoWrapper(dispatchRequest);
4 }

```

再来看看IndexFile的写入逻辑，其实也很简单，无非就是在IndexFile里去构建对应的索引罢了，如下面的源码片段。

```

1 if (DefaultMessageStore.this.messageStoreConfig.isMessageIndexEnable()) {
2     DefaultMessageStore.this.indexService.buildIndex(request);
3 }

```

因此到目前为止，我想大家基本就看明白了，当我们把消息写入到CommitLog之后，有一个后台线程每隔1毫秒就会去拉取CommitLog中最新更新的一批消息，然后分别转发到ConsumeQueue和IndexFile里去，这就是他底层的实现原理。

那么明天我们再来继续看同步刷盘和异步刷盘的实现。

End

专栏版权归公众号**狸猫技术窝**所有

未经许可不得传播，如有侵权将追究法律责任

狸猫技术窝精品专栏及课程推荐：

- [《从零开始带你成为JVM实战高手》](#)
- [《21天互联网Java进阶面试训练营》（分布式篇）](#)
- [《互联网Java工程师面试突击》（第1季）](#)
- [《互联网Java工程师面试突击》（第3季）](#)

重要说明：

- 如何提问：每篇文章都有评论区，大家可以尽情留言提问，我会逐一答疑
- 如何加群：购买狸猫技术窝专栏的小伙伴都可以加入狸猫技术交流群，一个非常纯粹的技术交流的地方

具体加群方式，请参见目录菜单下的文档：《付费用户如何加群》（[购买后可见](#)）

2020.12.1
1000x1000mm

Copyright © 2015-2020 深圳小鹅网络技术有限公司 All Rights Reserved. [粤ICP备15020529号](#)

 小鹅通提供技术支持

之准→手00642600657