

昵称：timfruit

园龄：2年10个月

粉丝：2

关注：28

+加关注

< 2021年5月 >						
日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

搜索

找找看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

我的标签
设计模式(12)
mybatis(9)
java(8)
工具集(7)
java并发(4)
jvm(4)
网络(4)
nio(2)
TCP/IP(2)
线程池(1)
更多

随笔分类
dubbo(1)
java(1)
java并发(6)
java基础(1)
jvm(4)
leetcode(2)
maven(1)
mongodb(1)
mybatis(8)
netty(1)
nio(2)
redis(6)
spring(1)
spring-cloud(1)
单元测试(2)
更多

随笔档案
2020年6月(1)
2020年5月(2)
2020年4月(6)
2020年3月(1)
2020年1月(3)
2019年12月(4)
2019年11月(6)
2019年10月(2)
2019年9月(6)
2019年8月(9)
2019年7月(10)
2019年6月(6)
2019年2月(2)
2018年12月(1)
2018年10月(1)
更多

最新评论
1. Re:java中转换不同时区的时间

消息队列 - Spring Boot 对rabbitmq批量处理数据的支持

消息队列 - Spring Boot 对rabbitmq批量处理数据的支持

一、前言

在生产中，存在一些场景，需要对数据进行批量操作。如，可以先将数据存放到redis，然后将数据进行批量写进数据库。但是使用redis，不得不面对一个数据容易丢失的问题。也可以考虑使用消息队列进行替换，在数据持久化，数据不丢失方面，消息队列确实比redis好一点，毕竟设计不一样。是不是使用消息队列，就一定好呢？不是的，首先使用消息队列，不能确保数据百分百不丢失，（如果要做到百分百不丢失，设计上就会比较复杂），除此之外，还要面对数据重复的问题。消息丢失，消息重复，是使用消息队列必须面对的问题。

AMQP在协议上规定每次只能传送一条数据，因此做批量数据操作，需要在应用层上定义，Spring 目前已经提供 [\(来源资料\)](#)

M-m-m. No, there is no such a functionality. Only one message can be read at a time from the queue. And it is on the protocol level. That's why we have introduced that artificial `BatchingRabbitTemplate` to batch on the application level, before protocol. – Artem Bilan Nov 30 '16 at 12:02

二、spring rabbit mq 支持批量操作的版本

关于起始版本，笔者尚未查找到佐证资料，目前笔者所使用的是2.2.2版本

```
批量发送

/**
 * A {@link RabbitTemplate} that permits batching individual messages into a larger
 * message. All {@code send()} methods (except
 * {@link #send(String, String, org.springframework.amqp.core.Message,
 * org.springframework.amqp.rabbit.connection.CorrelationData)})
 * are eligible for batching.
 * <b>Experimental - APIs may change.</b>
 *
 * @author Gary Russell
 * @since 1.4.1
 */
public class BatchingRabbitTemplate extends RabbitTemplate {
    private final BatchingStrategy batchingStrategy;

    private final TaskScheduler scheduler;

    private volatile ScheduledFuture<?> scheduledTask;
}
```

批量监听(其实不一定需要实现该接口，目前笔者的实现是使用该接口，其他可以自行查看官方文档)

```
/**
 * Used to receive a batch of messages if the container supports it.
 *
 * @author Gary Russell
 * @since 2.2
 */
public interface BatchMessageListener extends MessageListener {

    @Override
    default void onMessage(Message message) {
        throw new UnsupportedOperationException("Should never be called by the contain
    }

    @Override
    void onMessageBatch(List<Message> messages);
}
```

官方文档资料

← → ↺ 安全 | https://docs.spring.io/spring-amqp/reference/html/#receiving-batch

应用 有道

其他书签

2. What's New

3. Introduction

4. Reference

4.1. Using Spring AMQP

4.1.1. AMQP Abstractions

4.1.2. Connection and Resource Management

4.1.3. Adding Custom Client Connection Properties

4.1.4. AmqpTemplate

4.1.5. Sending Messages

4.1.6. Receiving Messages

Polling Consumer

Asynchronous Consumer

Batched Messages

Consumer Events

Consumer Tags

Annotation-driven Listener Endpoints

Using Container Factories

Asynchronous @RabbitListener Return

@RabbitListener with Batching

When receiving a **a batch** of messages, the de-batching is normally performed by the container and the listener is invoked with one message at a time. Starting with version 2.2, you can configure the listener container factory and listener to receive the entire batch in one call, simply set the factory's `batchListener` property, and make the method payload parameter a `List`:

```
@Bean
public SimpleRabbitListenerContainerFactory rabbitListenerContainerFactory() {
    SimpleRabbitListenerContainerFactory factory = new SimpleRabbitListenerContainerFactory();
    factory.setConnectionFactory(connectionFactory());
    factory.setBatchListener(true);
    return factory;
}

@RabbitListener(queues = "batch.1")
public void listen1(List<Thing> in) {
    ...
}

// or

@RabbitListener(queues = "batch.2")
public void listen2(List<Message<Thing>> in) {
    ...
}
```

三、实现

1. 简单配置测试队列

```
//测试批量
public static final String BATCH_QUEUE_NAME="batch.queue";

@Bean
public Queue batchQueue() {
    return new Queue(BATCH_QUEUE_NAME);
}
```

2. 配置批量发送template

```
@Bean("batchQueueTaskScheduler")
public TaskScheduler batchQueueTaskScheduler() {
    TaskScheduler taskScheduler=new ThreadPoolTaskScheduler();
    return taskScheduler;
}

//批量处理rabbitTemplate
@Bean("batchQueueRabbitTemplate")
public BatchingRabbitTemplate batchQueueRabbitTemplate(ConnectionFactory connectionFactory,
    @Qualifier("batchQueueTaskScheduler") TaskScheduler taskScheduler){

    //!!!重点: 所谓批量, 就是spring 将多条message重新组成一条message, 发送到mq, 从mq接受到这条message后, 在重新解析成多条message

    //一次批量的数量
    int batchSize=10;
    // 缓存大小限制,单位字节,
    // simpleBatchingStrategy的策略, 是判断message数量是否超过batchSize限制或者message的大小是否超过缓存限制,
    // 缓存限制, 主要用于限制"组装后的一条消息的大小"
    // 如果主要通过数量来做批量("打包"成一条消息), 缓存设置大点
    // 详细逻辑请看simpleBatchingStrategy.addToBatch()
    int bufferLimit=1024; //1 K
```


lz, 你的转换少考虑了一种情况, 就是本机本身支持夏令时, 当然如果你在北京时区操作, 是没问题的, 如果程序本身在支持夏令时的时区, 比如在洛杉矶, new Date, 转纽约, 则会差一个小时 应改为: `int...`

--silyvin

2. Re-学习设计模式 - 中介者模式

@ 前世迟来者 挺好用的 ^_^...

--timfruit

3. Re-学习设计模式 - 中介者模式

请问一下。博主你是用的什么画的UML图呀

--前世迟来者

阅读排行榜

1. Redis - RedisTemplate的切换库实现(5638)

2. dubbo - dubbo2.7.5 dubbo admin "无元数据信息, 请升级至Dubbo2.7及以上版本"问题解决(4827)

3. java中转换不同时区的时间(4543)

4. maven - 配置指定1.8jdk(3986)

5. 消息队列 - Spring Boot 对rabbitmq批量处理数据的支持(3548)

评论排行榜

1. 学习设计模式 - 中介者模式(2)

2. java中转换不同时区的时间(1)

推荐排行榜

1. Redis - RedisTemplate的切换库实现(3)

2. mybatis从入门到精通(五) sqlSession API的使用(2)

3. mybatis源码分析(四) mybatis与spring事务管理分析(1)

4. 学习TCP/IP - TCP三次握手连接和四次握手断开连接(1)

```
long timeout=10000;

//注意, 该策略只支持一个exchange/routingKey
//A simple batching strategy that supports only one exchange/routingKey
BatchingStrategy batchingStrategy=new SimpleBatchingStrategy(batchSize,bufferLimit,timeout);
return new BatchingRabbitTemplate(connectionFactory,batchingStrategy,taskScheduler);
}
```

3. 批量监听 (注意, 批量发送和批量监听可以各自独立使用)

a. 配置监听容器(这里是必须的!!!)

```
@Bean("batchQueueRabbitListenerContainerFactory")
public SimpleRabbitListenerContainerFactory batchQueueRabbitListenerContainerFactory(ConnectionFactory connectionFactory) {
    SimpleRabbitListenerContainerFactory factory = new SimpleRabbitListenerContainerFactory();
    factory.setConnectionFactory(connectionFactory);
    //设置批量
    factory.setBatchListener(true);
    factory.setConsumerBatchEnabled(true); //设置BatchMessageListener生效
    factory.setBatchSize(10); //设置监听器一次批量处理的消息数量
    return factory;
}
```

b. 配置监听器

```
@Slf4j
@Component
public class BatchQueueListener implements BatchMessageListener {

    //批量接收处理
    @RabbitListener(queues = RabbitMqConfig2.BATCH_QUEUE_NAME,containerFactory = "batchQueueRabbitListenerContainerFactory")
    @Override
    public void onMessageBatch(List<Message> messages) {
        log.info("batch.queue.consumer 收到{}条message", messages.size());
        if(messages.size()>0){
            log.info("第一条数据是: {}", new String(messages.get(0).getBody()));
        }
    }
}
```

4. 测试

```
// ----- 测试batch
@Autowired
BatchingRabbitTemplate batchQueueRabbitTemplate;

@Test
public void batchSend() throws InterruptedException {
    // 除了send(String exchange, String routingKey, Message message, CorrelationData correlationData)方法是发送单条数据
    // 其他send都是批量发送

    //批量发送

    long timestamp=System.currentTimeMillis();

    String msg;
    Message message;
    MessageProperties messageProperties=new MessageProperties();
    for(int i=0;i<1000;i++){
        msg="batch."+timestamp+"-"+i;

        message=new Message(msg.getBytes(), messageProperties);
        batchQueueRabbitTemplate.send(RabbitMqConfig2.BATCH_QUEUE_NAME,message);

        defaultRabbitTemplate.convertAndSend(RabbitMqConfig2.BATCH_QUEUE_NAME, msg.getBytes());
    }

    System.out.println("发送数据完毕");
    System.out.println("等待30s");
    TimeUnit.SECONDS.sleep(30); //等待消费者消费
}
```

5. 输出结果

```
Tests passed: 1 of 1 test — 31s 691ms
2020-04-04 21:40:09.694 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : batch.queue.consumer 收到100条message
2020-04-04 21:40:09.776 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : 第一条数据是: batch.1586007609407-0
2020-04-04 21:40:09.783 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : batch.queue.consumer 收到100条message
2020-04-04 21:40:09.817 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : 第一条数据是: batch.1586007609407-100
2020-04-04 21:40:09.818 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : batch.queue.consumer 收到100条message
2020-04-04 21:40:09.818 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : 第一条数据是: batch.1586007609407-200
2020-04-04 21:40:09.857 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : batch.queue.consumer 收到100条message
2020-04-04 21:40:09.857 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : batch.queue.consumer 收到100条message
2020-04-04 21:40:09.881 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : 第一条数据是: batch.1586007609407-300
2020-04-04 21:40:09.882 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : batch.queue.consumer 收到100条message
2020-04-04 21:40:09.882 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : 第一条数据是: batch.1586007609407-400
2020-04-04 21:40:09.955 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : batch.queue.consumer 收到100条message
2020-04-04 21:40:09.956 INFO 29852 --- [tContainer#11-1] c.t.r.demo.mq2.batch.BatchQueueListener : 第一条数据是: batch.1586007609407-500
```

解析:

设置批量监听处理的数量为10, 为什么输出是100呢?

因为使用了批量发送, 配置批量发送是将10条数据压缩成1条, 批量监听收到的是压缩后的10条, 解析后, 变成100条, 没毛病

完整demo源码

参考资料:

[官方文档](#)

人生没有彩排, 每一天都是现场直播

分类: 消息队列

好文置顶

关注我

收藏该文

timfruit

关注 - 28

粉丝 - 2

+加关注

« 上一篇: [消息队列 - Spring boot 中RabbitMQ的使用](#)

» 下一篇: [消息队列 - 关于消息队列的消息丢失问题的一点思考](#)

登录后才能查看或发表评论, 立即 登录 或者 逛逛 博客园首页

- 【推荐】阿里云专家倾力打造, 实时计算Flink版+Hologres实时数仓 “王炸组合”
- 【推荐】大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载!
- 【推荐】玩转开发板: 旧键盘+OpenHarmony 变身蓝牙键盘 v0.1
- 【推荐】阿里云爆品销量榜单, 精选爆款产品低至0.55折
- 【推荐】限时秒杀! 国云大数据魔镜, 企业级云分析平台

刷新评论 刷新页面 返回顶部

园子动态:

- 致园友们的一封检讨书：都是我们的错
- 数据库实例 CPU 100% 引发全站故障
- 发起一个开源项目：博客引擎 fluss

最新新闻:

- 四月新能源车销量出炉：五个新造车加起来顶个特斯拉
- 老艺术家，别碰直播带货
- 微软如何走出反垄断？
- 芒格、李录与段永平 “battle” 比亚迪
- 中科驭数宣布与中移物联网达成战略合作
- » 更多新闻...