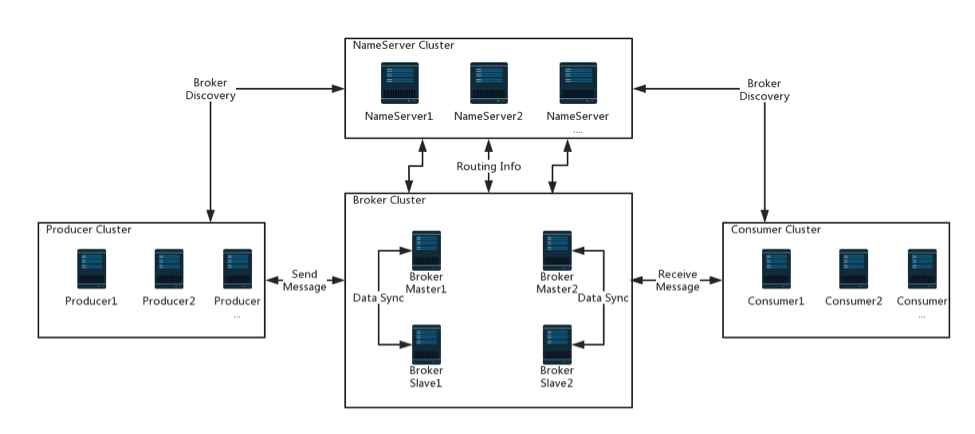
RocketMQ简介

RocketMQ是低延迟、高可靠、可伸缩和易于使用的消息中间件，具有以下特性：

* 支持发布/订阅和点对点P2P消息模型
* 在一个队列中可靠的先进先出和严格的顺序传递
* 支持pull和push两种消息模式
* 单一队列百万消息的堆积能力
* 支持多种消息协议，如JMS/MQTT等
* 分布式高可用的部署架构，满足至少一次消息传递语义
* 提供Docker镜像用于隔离测试和云集群部署
* 提供配置、指标和监控等功能丰富的Dashboard

1. **RocketMQ系统架构及概念**

RocketMQ的系统架构如下图：



其包括4个部分：name servers、brokers、producers及consumers，每种组件都可以水平扩展，不存在单点问题，各组件的介绍如下：

1. NameServer，提供轻量级的服务器发现和路由，每个NameServer记录完整的路由信息，提供等效的读写服务，并支持快速存储扩展。Broker与NameServer集群的所有节点建立长连接，定时注册Topic消息到所有的NameServer中。
2. Broker，通过提供轻量级的Topic和Queue机制来处理消息存储，同时支持pull和push模式，其采用主从结构的容错机制。Broker Master和Slave之间会进行数据同步。
3. Producer，生产者，产生消息，相同Producer Group的Producer组成一个集群。Producer与NameServer集群中某个节点建立长连接，定期从NameServer获取Topic路由信息，同时向提供Topic服务的Broker Master建立长连接，并定时发送心跳。
4. Consumer，消费者，接收并消费，相同的Consumer Group的Consumer组成一个集群。其同时与Topic服务的Master和Slave建立长连接。

其他概念：

1. Topic，消息的逻辑分类，如订单类消息，通过创建订单Topic存放订单相关的消息
2. Message，消息的载体，发送时指定Topic
3. Tag，对Topic进一步细化，相同页面模块通过引入标签来标记不同用途的消息
4. **RocketMQ的部署**

RocketMQ集群部署的模式很多，下面仅以测试为目的部署单主模式

1）下载RocketMQ安装包

*https://www.apache.org/dyn/closer.cgi?path=rocketmq/4.2.0/rocketmq-all-4.2.0-bin-release.zip*

2）解压并配置环境变量

*unzip rocketmq-all-4.2.0-bin-release.zip*

环境变量.bashrc配置如下：

*export ROCKETMQ\_HOME=/opt/rocketmq*

*export PATH="$ROCKETMQ\_HOME/bin:$PATH"*

1. 启动Name Server

*#nohup sh /opt/rocketmq/bin/mqnamesrv &*

*# jps*

*4010 NamesrvStartup*

*# netstat -anp|grep 4010*

*tcp6 0 0 :::9876 :::\* LISTEN 4010/java*

*# tailf ~/logs/rocketmqlogs/namesrv.log*

*.......*

*2018-07-07 23:16:14 INFO main - The Name Server boot success. serializeType=JSON*

默认使用端口9876

1. 启动broker

*# nohup sh /opt/rocketmq/bin/mqbroker -n fys1.cmss.com:9876 &*

*# jps*

*4354 BrokerStartup*

*#tailf ~/logs/rocketmqlogs/broker.log*

*.....*

*register broker to name server fys1.cmss.com:9876 OK*

启动成功

1. **安装RocketMQ Dashboard**

RocketMQ提供RocketMQ Console，图形化管理控制台，可以查看Broker集群状态、Topcic管理、消息查询等常用功能

1. 下载源代码

*git clone -b release-rocketmq-console-1.0.0 https://github.com/apache/rocketmq-externals.git*

1. 编译

*cd rocketmq-console*

*mvn clean package -DskipTests*

*scp rocketmq-console-ng-1.0.0.jar root@fys1.cmss.com:~/*

1. 启动

*java -jar rocketmq-console-ng-1.0.0.jar*

*. \_\_\_\_ \_ \_\_ \_ \_*

*/\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \*

*( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \*

*\\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )*

*' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /*

*=========|\_|==============|\_\_\_/=/\_/\_/\_/*

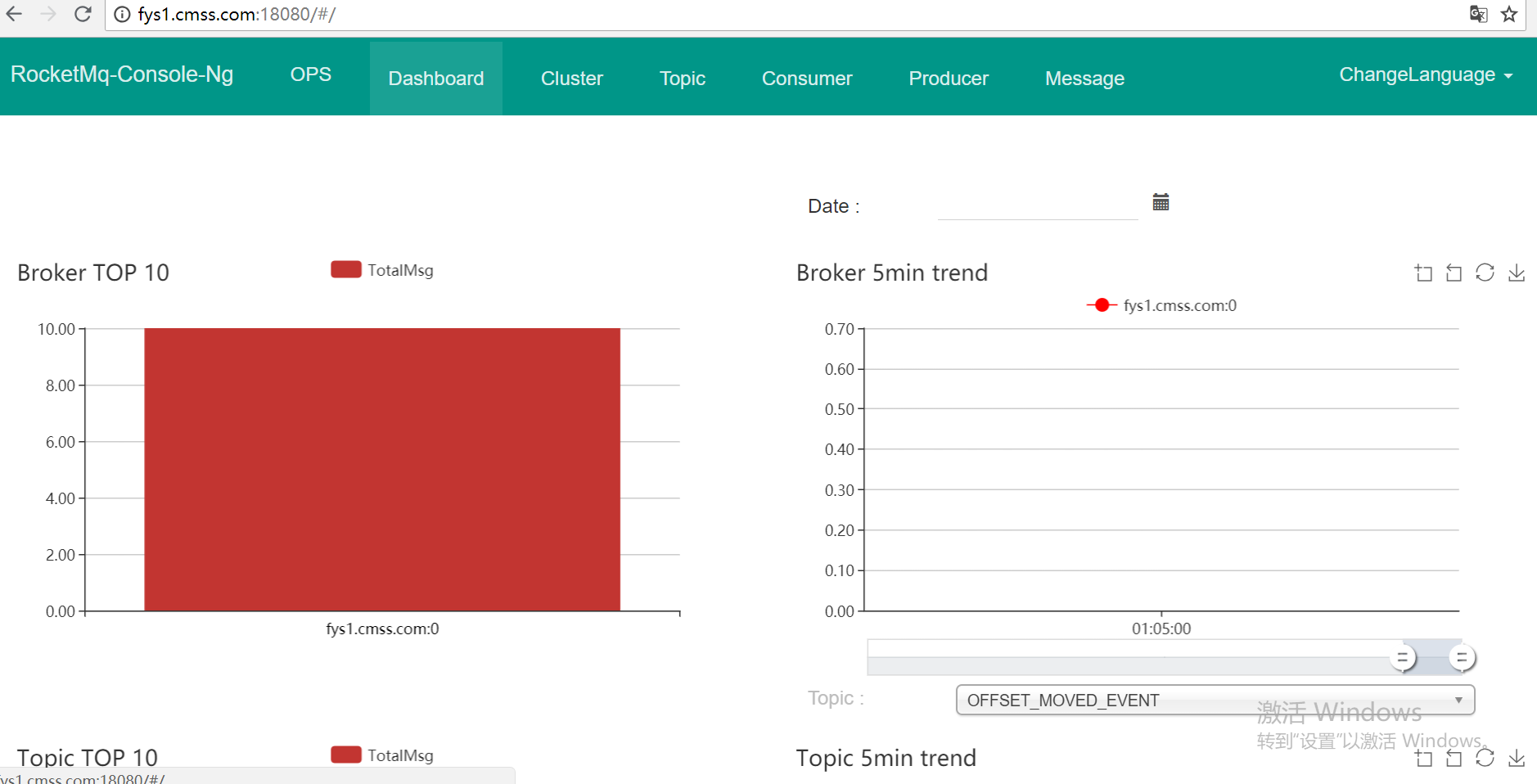
*:: Spring Boot :: (v1.4.3.RELEASE)*

*....*

*[2018-07-08 01:02:29.787] INFO Tomcat started on port(s): 18080 (http)*

1. 打开页面如下：

*<http://fys1.cmss.com:18080/#/>*



1. **RocketMQ的使用**
2. 定义生产者

*public class RocketMQProducer {*

*public static void main(String[] args) throws MQClientException {*

*DefaultMQProducer producer = new DefaultMQProducer("FirstProducer");*

*producer.setNamesrvAddr("10.139.4.98:9876");*

*producer.start();*

*for(int i = 0 ; i < 10 ;i++) {*

*try {*

*Message msg = new Message("TopicTest",*

*"TagA",*

*("Hello RocketMQ " + i).getBytes(RemotingHelper.DEFAULT\_CHARSET));*

*SendResult sendResult = producer.send(msg);*

*System.out.println(sendResult);*

*} catch (Exception e) {*

*e.printStackTrace();*

*}*

*}*

*producer.shutdown();*

*}*

*}*

1. 定义消费者

*public class RocketMQConsumer {*

*public static void main(String[] args) throws MQClientException {*

*DefaultMQPushConsumer consumer = new DefaultMQPushConsumer("FirstConsumer");*

*consumer.setNamesrvAddr("10.139.4.98:9876");*

*consumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME\_FROM\_FIRST\_OFFSET);*

*consumer.subscribe("TopicTest","\*");*

*consumer.registerMessageListener(new MessageListenerConcurrently() {*

*public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt> msgs,*

*ConsumeConcurrentlyContext context) {*

*System.out.println(Thread.currentThread().getName() + " Receive New Messages: " + msgs);*

*return ConsumeConcurrentlyStatus.CONSUME\_SUCCESS;*

*}*

*});*

*//调用start()方法启动consumer*

*consumer.start();*

*System.out.println("Consumer Started.");*

*}*

*}*

1. 执行输出如下：

Producer:

*SendResult [sendStatus=SEND\_OK, msgId=AC110001154160E53B932436A39B0000, offsetMsgId=AC11000100002A9F0000000000000000, messageQueue=MessageQueue [topic=TopicTest, brokerName=fys1.cmss.com, queueId=2], queueOffset=0]*

*......*

Consumer：

*Consumer Started.*

*ConsumeMessageThread\_4 Receive New Messages: [MessageExt [queueId=2, storeSize=178, queueOffset=0, sysFlag=0, bornTimestamp=1530981960604, bornHost=/10.139.4.98:40920, storeTimestamp=1530981960631, storeHost=/172.17.0.1:10911, msgId=AC11000100002A9F0000000000000000, commitLogOffset=0, bodyCRC=613185359, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message [topic=TopicTest, flag=0, properties={MIN\_OFFSET=0, MAX\_OFFSET=3, CONSUME\_START\_TIME=1530981975023, UNIQ\_KEY=AC110001154160E53B932436A39B0000, WAIT=true, TAGS=TagA}, body=16]]]*

https://my.oschina.net/buru1kan/blog/1814356

https://www.jianshu.com/p/824066d70da8