

<http://blog.csdn.net/juny9123/article/details/72632812>

Kafka入门-windows安装测试

一、kafka是什么？

Kafka是一种高吞吐量的分布式发布订阅消息系统，它可以处理消费者规模的网站中的所有动作流数据。有如下特性：

- 1、以时间复杂度为 $O(1)$ 的方式提供消息持久化能力，即使对TB级以上数据也能保证常数时间复杂度的访问性能。
- 2、高吞吐率。即使在非常廉价的商用机器上也能做到单机支持每秒100K条以上消息的传输。
- 3、支持Kafka Server间的消息分区，及分布式消费，同时保证每个Partition内的消息顺序传输。
- 4、同时支持离线数据处理和实时数据处理。
- 5、Scale out：支持在线水平扩展。

$O(1)$ ：时间复杂度的算法

为何使用消息系统：

- 解耦

在项目启动之初来预测将来项目会碰到什么需求，是极其困难的。消息系统在处理过程中插入了一个隐含的、基于数据的接口层，两边的处理过程都要实现这一接口。这允许你独立的扩展或修改两边的处理过程，只要确保它们遵守同样的接口约束。

- 冗余

有些情况下，处理数据的过程会失败。除非数据被持久化，否则将造成丢失。消息队列把数据进行持久化直到它们已经被完全处理，通过这一方式规避了数据丢失风险。许多消息队列所采用的"插入-获取-删除"范式中，在把一个消息从队列中删除之前，需要你的处理系统明确的指出该消息已经被处理完毕，从而确保你的数据被安全的保存直到你使用完毕。

- 扩展性

因为消息队列解耦了你的处理过程，所以增大消息入队和处理的频率是很容易的，只要另外增加处理过程即可。不需要改变代码、不需要调节参数。扩展就像调大电

力按钮一样简单。

- 灵活性 & 峰值处理能力

在访问量剧增的情况下，应用仍然需要继续发挥作用，但是这样的突发流量并不常见；如果为以能处理这类峰值访问为标准来投入资源随时待命无疑是巨大的浪费。使用消息队列能够使关键组件顶住突发的访问压力，而不会因为突发的超负荷的请求而完全崩溃。

- 可恢复性

系统的一部分组件失效时，不会影响到整个系统。消息队列降低了进程间的耦合度，所以即使一个处理消息的进程挂掉，加入队列中的消息仍然可以在系统恢复后被处理。

- 顺序保证

在大多使用场景下，数据处理的顺序都很重要。大部分消息队列本来就是排序的，并且能保证数据会按照特定的顺序来处理。Kafka保证一个Partition内的消息的有序性。

- 缓冲

在任何重要的系统中，都会有需要不同的处理时间的元素。例如，加载一张图片比应用过滤器花费更少的时间。消息队列通过一个缓冲层来帮助任务最高效率的执行——写入队列的处理会尽可能的快速。该缓冲有助于控制和优化数据流经过系统的速度。

- 异步通信

很多时候，用户不想也不需要立即处理消息。消息队列提供了异步处理机制，允许用户把一个消息放入队列，但并不立即处理它。想向队列中放入多少消息就放多少，然后在需要的时候再去处理它们。

常用Message Queue对比：

- RabbitMQ

RabbitMQ是使用Erlang编写的一个开源的消息队列，本身支持很多的协议：AMQP, XMPP, SMTP, STOMP，也正因此，它非常重量级，更适合于企业级的开发。同时实现了Broker构架，这意味着消息在发送给客户端时先在中心队列排队。对路由，负载均衡或者数据持久化都有很好的支持。

- Redis

Redis是一个基于Key-Value对的NoSQL数据库，开发维护很活跃。虽然它是一个Key-Value数据库存储系统，但它本身支持MQ功能，所以完全可以当做一个轻量级的队列服务来使用。对于RabbitMQ和Redis的入队和出队操作，各执行100万次，每10万次记录一次执行时间。测试数据分为128Bytes、512Bytes、1K和10K四个不同大小的数据。实验表明：入队时，当数据比较小时Redis的性能要高于RabbitMQ，而如果数据大小超过了10K，Redis则慢的无法忍受；出队时，无论数据大小，Redis都表现出非常好的性能，而RabbitMQ的出队性能则远低于Redis。

- ZeroMQ

ZeroMQ号称最快的消息队列系统，尤其针对大吞吐量的需求场景。ZeroMQ能够实现RabbitMQ不擅长的高级/复杂的队列，但是

开发人员需要自己组合多种技术框架，技术上的复杂度是对这MQ能够应用成功的挑战。ZeroMQ具有一个独特的非中间件的模式，你不需要安装和运行一个消息服务器或中间件，因为你的应用程序将扮演这个服务器角色。你只需要简单的引用ZeroMQ程序库，可以使用NuGet安装，然后你就可以愉快的在应用程序之间发送消息了。但是ZeroMQ仅提供非持久性的队列，也就是说如果宕机，数据将会丢失。其中，Twitter的Storm 0.9.0以前的版本中默认使用ZeroMQ作为数据流的传输（Storm从0.9版本开始同时支持ZeroMQ和Netty作为传输模块）。

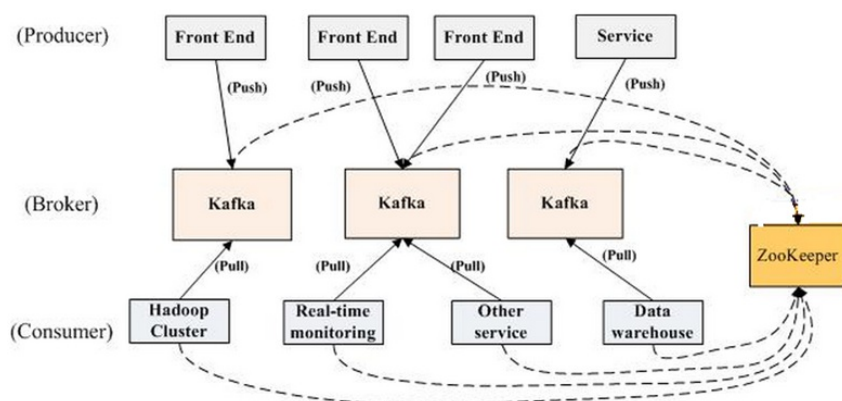
- **ActiveMQ**

ActiveMQ是Apache下的一个子项目。类似于ZeroMQ，它能够以代理人和点对点的技术实现队列。同时类似于RabbitMQ，它少量代码就可以高效地实现高级应用场景。

- **Kafka/Jafka**

Kafka是Apache下的一个子项目，是一个高性能跨语言分布式发布/订阅消息队列系统，而Jafka是在Kafka之上孵化而来的，即Kafka的一个升级版。具有以下特性：快速持久化，可以在O(1)的系统开销下进行消息持久化；高吞吐，在一台普通的服务器上既可以达到10W/s的吞吐速率；完全的分布式系统，Broker、Producer、Consumer都原生自动支持分布式，自动实现负载均衡；支持Hadoop数据并行加载，对于像Hadoop的一样的日志数据和离线分析系统，但又要求实时处理的限制，这是一个可行的解决方案。Kafka通过Hadoop的并行加载机制统一了在线和离线的消息处理。Apache Kafka相对于ActiveMQ是一个非常轻量级的消息系统，除了性能非常好之外，还是一个工作良好的分布式系统。

Kafka拓扑结构：



如上图所示，一个典型的Kafka集群中包含若干Producer（可以是web前端产生的Page View，或者是服务器日志，系统CPU、Memory等），若干broker（Kafka支持水平扩展，一般broker数量越多，集群吞吐率越高），若干Consumer Group，以及一个Zookeeper集群。Kafka通过Zookeeper管理集群配置，选举leader，以及在Consumer Group发生变化时进行rebalance。Producer使用push模式将消息发布到broker，Consumer使用pull模式从broker订阅并消费消息。

二、Kafka相关术语介绍

- **Broker（服务器）**

Kafka集群包含一个或多个服务器，这种服务器被称为broker（硬件）

- **Topic（频道）**

每条发布到Kafka集群的消息都有一个类别，这个类别被称为Topic。（物理上不同Topic的消息分开存储，逻辑上一个Topic的消息虽然保存于一个或多个broker上但用户只需指定消息的Topic即可生产或消费数据而不必关心数据存于何处）

- Partition (区分、隔离)

Partition是物理上的概念，每个Topic包含一个或多个Partition.

- Producer (生产者)

负责发布消息到Kafka broker

- Consumer (消费者)

消息消费者，向Kafka broker读取消息的客户端。

- Consumer Group (消费者组)

每个Consumer属于一个特定的Consumer Group (可为每个Consumer指定group name，若不指定group name则属于默认的group)。

三、下载

下载地址：<http://kafka.apache.org/downloads>

下载后是一个压缩包

 kafka_2.11-0.10.1.0.gz

四、相关配置

1、依赖jdk，所以必须安装jdk并配置环境变量

2、启动zookeeper

kafka使用zookeeper来实现动态的集群扩展，不需要更改客户端（producer和consumer）的配置。broker会在zookeeper注册并保持相关的元数据（topic，partition信息等）更新。

而客户端会在zookeeper上注册相关的watcher。一旦zookeeper发生变化，客户端能及时感知并作出相应调整。这样就保证了添加或去除broker时，各broker间仍能自动实现负载均衡。

3、启动kafka

3.1、解压kafka安装包

3.2、编辑文件config/server.properties

3.3、找到并编辑**log.dirs**=D:\Java\Tool\kafka_2.11-0.10.0.1\kafka-log,这里的目录自己修改成自己喜欢的

```
59 # A comma seperated list of direct,
60 log.dirs=../tmp/kafka-logs
```

3.4、找到并编辑**zookeeper.connect**=localhost:2181。默认表示本地运行（不改）

3.5、Kafka会按照默认，在**9092**端口上运行，并连接zookeeper的默认端口：

2181。

3.6、dos命令进入kafka/bin/windows/这个路径，

执行命令：`kafka-server-start.bat ..\..\config\server.properties`

假如报如下错误：

```
D:\devtools\kafka\kafka_2.11_2\bin\windows>kafka-server-start.bat ..\..\config\server.properties
错误：找不到或无法加载主类 Files\Java\jdk1.7.0_71\bin;D:\Program
D:\devtools\kafka\kafka_2.11_2\bin\windows>
```

解决方案：

在kafka安装目录中找到bin\windows目录中的**kafka-run-class.bat**找到142行为%CLASSPATH%加上双引号

修改前：`set COMMAND=%JAVA% %KAFKA_HEAP_OPTS% %KAFKA_JVM_PERFORMAN
CE_OPTS% %KAFKA_JMX_OPTS% %KAFKA_LOG4J_OPTS% -cp %CLASSPATH%KAFK
A_OPTS% %*`

修改后：`set COMMAND=%JAVA% %KAFKA_HEAP_OPTS% %KAFKA_JVM_PERFORMAN
CE_OPTS% %KAFKA_JMX_OPTS% %KAFKA_LOG4J_OPTS% -cp "%CLASSPATH%"%KA
FKA_OPTS% %*`

改完启动，成功页面如下：

```
[2017-12-14 09:39:31,985] INFO [GroupCoordinator 0]: Startup complete. (kafka.coordinator.GroupCoordinator)
[2017-12-14 09:39:31,986] INFO [Group Metadata Manager on Broker 0]: Removed 0 expired offsets in 1 milliseconds. (kafka
coordinator.GroupMetadataManager)
[2017-12-14 09:39:32,000] INFO Will not load MX4J, mx4j-tools.jar is not in the classpath (kafka.utils.Mx4jLoader$)
[2017-12-14 09:39:32,032] INFO Creating /brokers/ids/0 (is it secure? false) (kafka.utils.ZKCheckedEphemeral)
[2017-12-14 09:39:32,063] INFO Result of znode creation is: OK (kafka.utils.ZKCheckedEphemeral)
[2017-12-14 09:39:32,065] INFO Registered broker 0 at path /brokers/ids/0 with addresses: PLAINTEXT -> EndPoint(DESKTOP-
ASN29JH,9092,PLAINTEXT) (kafka.utils.ZkUtils)
[2017-12-14 09:39:32,074] INFO Kafka version : 0.10.1.0 (org.apache.kafka.common.utils.AppInfoParser)
[2017-12-14 09:39:32,074] INFO Kafka commitId : 3402a74efb23d1d4 (org.apache.kafka.common.utils.AppInfoParser)
[2017-12-14 09:39:32,076] INFO [Kafka Server 0], started (kafka.server.KafkaServer)
[2017-12-14 09:39:32,223] INFO [ReplicaFetcherManager on broker 0] Removed fetcher for partitions linlin-0 (kafka.server
ReplicaFetcherManager)
[2017-12-14 09:39:32,241] INFO [ReplicaFetcherManager on broker 0] Removed fetcher for partitions linlin-0 (kafka.server
ReplicaFetcherManager)
```

3.7、创建topic (频道)

打开新的dos命令框，进入kafka/bin/windows/这个路径，

执行命令：`kafka-topics.bat --create --zookeeper localhost:2181`

`--replication-factor 1 --partitions 1 --topic chenjh`

创建一个chenjh频道

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation. 保留所有权利。

D:\devtools\kafka\kafka_2.11_2\bin\windows>kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --
-partitions 1 --topic chenjh
Created topic "chenjh".

D:\devtools\kafka\kafka_2.11_2\bin\windows>
```

3.8、客户端 (生产者) 连接频道

打开新的dos命令框，进入kafka/bin/windows/这个路径，

执行命令：`kafka-console-producer.bat --broker-list localhost:9`

`092 --topic chenjh`

创建一个关注chenjh频道的生产者

```
C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092 --topic chenjh
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

D:\devtools\kafka\kafka_2.11_2\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic chenjh
```

3.9、客户端（消费者）连接频道

打开新的dos命令框，进入kafka/bin/windows/这个路径，

执行命令：`kafka-console-consumer.bat --zookeeper localhost:2181`

`1 --topic chenjh`

创建一个关注chenjh频道的消费者

```
C:\Windows\System32\cmd.exe - kafka-console-consumer.bat --zookeeper localhost:2181 --topic chenjh
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

D:\devtools\kafka\kafka_2.11_2\bin\windows>kafka-console-consumer.bat --zookeeper localhost:2181 --topic chenjh
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider using
the new consumer by passing [bootstrap-server] instead of [zookeeper].
```

3.10、测试

在生产者命令窗口输入一句话进行测试，如：天王盖地虎，然后回车。可以看到消费之窗口获取到了这一条消息。

```
C:\Windows\System32\cmd.exe - kafka-console-consumer.bat --zookeeper localhost:2181 --topic chenjh
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

D:\devtools\kafka\kafka_2.11_2\bin\windows>kafka-console-consumer.bat --zookeeper localhost:2181 --topic chenjh
Using the ConsoleConsumer with old consumer is deprecated and will be removed in a future major release. Consider
the new consumer by passing [bootstrap-server] instead of [zookeeper].
天王盖地虎

C:\Windows\System32\cmd.exe - kafka-console-producer.bat --broker-list localhost:9092 --topic chenjh
Microsoft Windows [版本 10.0.14393]
(c) 2016 Microsoft Corporation。保留所有权利。

D:\devtools\kafka\kafka_2.11_2\bin\windows>kafka-console-producer.bat --broker-list localhost:9092 --topic chenjh
天王盖地虎
```

常用命令

- 查看现有的topic

```
.\bin\windows\kafka-topics.bat --zookeeper localhost:2181 --list
```

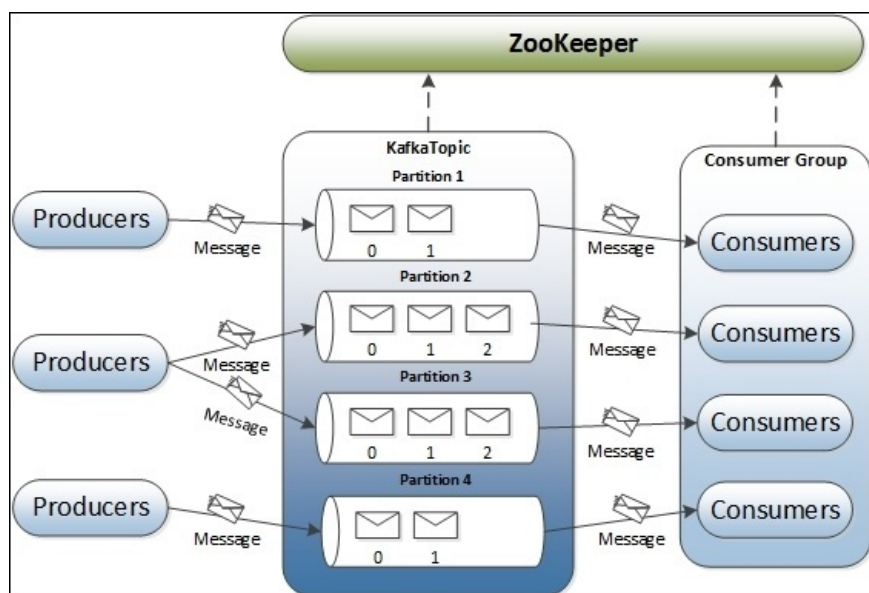
- 1

- 查看topic详细信息

```
.\bin\windows\kafka-topics.bat --zookeeper localhost:2181 --describe
```

后话

从上面的安装测试也大概可以看出kafka的结构。下图为kafka的结构图，摘自网络



java操作kafka

http://blog.csdn.net/zero_007/article/details/51068165

<http://blog.csdn.net/u012191627/article/details/53338503>

(参考)

<https://www.cnblogs.com/hd3013779515/p/6939013.html>