



OPC Router 实现 与 Apache Kafka 通讯

北京东方鼎晨科技有限公司

2021 年 02 月 05 日

目录

1. Apache Kafka 概述.....	2
1.1. 生产者和消费者（producer 和 consumer）	2
1.2. Broker	2
1.3. 主题（topic）	2
1.4. 分区（partition）	2
1.5. 偏移量（Offset）	3
2. Apache Kafka 安装及配置	3
2.1. 准备工作	3
2.1.1. JDK	3
2.1.2. Zookeeper	3
2.1.3. Kafka	4
2.2. 运行和创建	4
3. OPC Router 与 Apache Kafka 通讯配置	5
3.1. 通讯需求	5
3.2. 通讯配置	5
3.2.1. 新建 Plug-in	5
3.2.2. 新建 Connection	7
3.2.3. 运行与监视	7

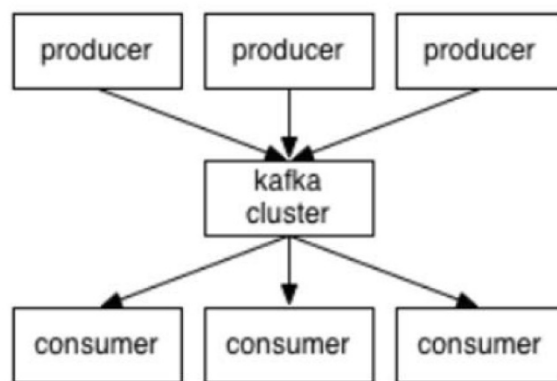
1. Apache Kafka 概述

Apache Kafka 是一个分布式发布/订阅消息系统，构建在 ZooKeeper 同步服务之上，用于实时流式数据分析。具有高性能、持久化、多副本备份、横向扩展能力。生产者往队列里写消息，消费者从队列里取消息进行业务逻辑。可以处理大量的数据，使你能够将消息从一个端点传递到另一个端点。Kafka 消息保留在磁盘上，并在群集内复制以防止数据丢失。

Apache Kafka 专为分布式高吞吐量系统而设计，具有更好的吞吐量，内置分区，复制和固有的容错能力，这使得它非常适合大规模消息处理应用程序。

1.1. 生产者和消费者 (producer 和 consumer)

消息的发送者叫 Producer，消息的使用者和接受者是 Consumer，生产者将数据保存到 Kafka 集群中，消费者从中获取消息进行业务的处理。



1.2. Broker

Kafka 集群中有很多台 Server，其中每一台 Server 都可以存储消息，将每一台 Server 称为一个 Kafka 实例，也叫做 broker。

1.3. 主题 (topic)

一个 topic 里保存的是同一类消息，相当于对消息的分类，每个 producer 将消息发送到 Kafka 中，都需要指明要存的 topic 是哪个，也就是指明这个消息属于哪一类。

1.4. 分区 (partition)

每个 topic 都可以分成多个 partition，每个 partition 在存储层面是 append log 文件。任何发布到此 partition 的消息都会被直接追加到 log 文件的尾部。为什么要进行分

区呢？最根本的原因就是：Kafka 基于文件进行存储，当文件内容大到一定程度时，很容易达到单个磁盘的上限，因此，采用分区的办法，一个分区对应一个文件，这样就可以将数据分别存储到不同的 server 上去，另外这样做也可以负载均衡，容纳更多的消费者。

1.5. 偏移量 (Offset)

一个分区对应一个磁盘上的文件，而消息在文件中的位置就称为 offset (偏移量)，offset 为一个 long 型数字，它可以唯一标记一条消息。由于 Kafka 并没有提供其他额外的索引机制来存储 offset，文件只能顺序的读写，所以在 Kafka 中几乎不允许对消息进行“随机读写”。

综上，我们总结 Kafka 的几个要点：

- Kafka 是一个基于发布-订阅的分布式消息系统 (消息队列)。
- Kafka 面向大数据，消息保存在主题中，而每个 topic 分为多个分区。
- Kafka 的消息数据保存在磁盘，每个 partition 对应磁盘上的一个文件，消息写入就是简单的文件追加，文件可以在集群内复制备份以防丢失。即使消息被消费，Kafka 也不会立即删除该消息，可以通过配置使得过一段时间后自动删除以释放磁盘空间。
- Kafka 依赖分布式协调服务 Zookeeper，适合离线/在线信息的消费，与 storm 和 saprk 等实时流式数据分析常常结合使用。

2. Apache Kafka 安装及配置

2.1. 准备工作

2.1.1. JDK

安装jdk,配置好 JAVA_HOME 和 path。

注意:

环境变量 JAVA_HOME 不要选择默认的"C:\Program Files\Java\jdk1.8.0_151"，因为文件夹路径不能有空格，后面可能启动 Kafka 服务出错。

2.1.2. Zookeeper

将安装包解压目录 C:\Tool。

进入 Zookeeper 设置目录，C:\Tool\zookeeper-3.4.13\conf

将 "zoo_sample.cfg" 重命名为 "zoo.cfg"

在文本编辑器 (如 notepad) 中打开 zoo.cfg

找到并编辑 `dataDir=C:\Tool\zookeeper-3.4.13\tmp` (可以自定义文件夹)

与 Java 中的做法类似, 我们在系统环境变量中添加:

- a. 在系统变量中添加 `ZOOKEEPER_HOME = C:\Tool\zookeeper-3.4.13`
- b. 编辑 `path` 系统变量, 添加为路径 `%ZOOKEEPER_HOME%\bin;`

在 zoo.cfg 文件中修改默认的 Zookeeper 端口 (默认端口 2181, 可以不用修改)

测试: 打开新的 cmd, 输入 `zkServer`, 运行 Zookeeper, 成功如下所示:

```
uring NIO connection handler with 10s sessionless connection timeout, 1 selector
thread(s), 2 worker threads, and 64 kB direct buffers.
2021-02-03 14:55:13,764 [myid:] - INFO [main:NIOServerCnxnFactory@674] - bindin
g to port 0.0.0.0/0.0.0.0:2181
2021-02-03 14:55:13,780 [myid:] - INFO [main:WatchManagerFactory@42] - Using or
g.apache.zookeeper.server.watch.WatchManager as watch manager
2021-02-03 14:55:13,780 [myid:] - INFO [main:WatchManagerFactory@42] - Using or
g.apache.zookeeper.server.watch.WatchManager as watch manager
2021-02-03 14:55:13,780 [myid:] - INFO [main:ZKDatabase@132] - zookeeper.snapsh
otSizeFactor = 0.33
```

2.1.3. Kafka

解压到目录 `C:\Tool\Kafka`

编辑文件 Kafka 配置文件, `C:\Tool\Kafka\config\server.properties`

找到并编辑 `log.dirs=C:\Tool\Kafka\Kafka-log`, (自定义文件夹)

找到并编辑 `zookeeper.connect=localhost:2181`。表示本地运行(默认的不改)

Kafka 会按照默认, 在 9092 端口上运行, 并连接 zookeeper 的默认端口: 2181。

2.2. 运行和创建

在启动 Kafka 前, 先运行 Zookeeper 实例。

1) 启动 Kafka 服务器

cmd 窗口: `cd C:\Tool\Kafka`

- `.\bin\windows\Kafka-server-start.bat .\config\server.properties`

2) 创建主题

cmd 窗口。

```
cd C:\Tool\Kafka\bin\windows
```

```
Kafka-topics.bat --create --zookeeper localhost:2181 --replication-  
factor 1 --partitions 1 --topic haha
```

3) 创建生产者

cmd 窗口

```
cd C:\Tool\Kafka\bin\windows
```

```
Kafka-console-producer.bat --broker-list localhost:9092 --topic haha
```

4) 创建消费者

cmd 窗口

```
cd C:\Tool\Kafka\bin\windows
```

```
Kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic  
haha --from-beginning
```

3. OPC Router 与 Apache Kafka 通讯配置

3.1. 通讯需求

通过 OPC UA/DA Server 采集工业控制系统 PLC 中设备运行数据，采用 OPC Router 实现与 Apache Kafka 系统数据交互。

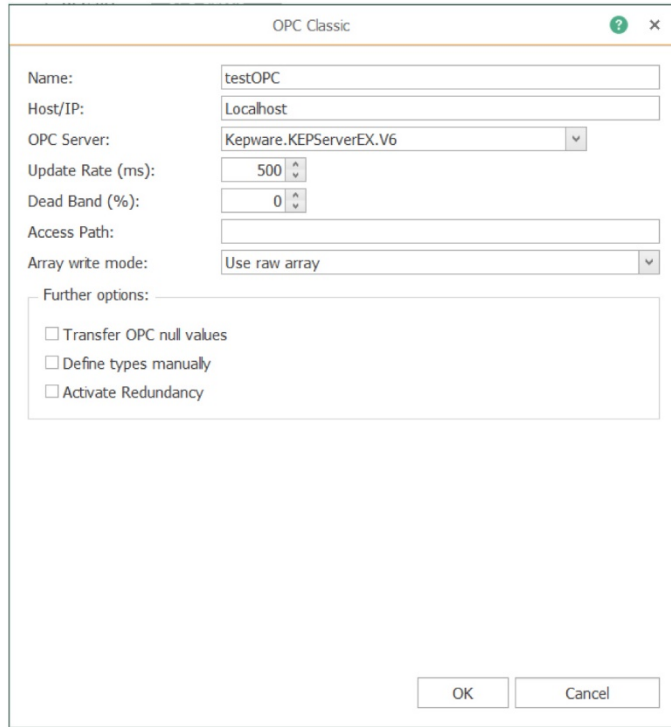


3.2. 通讯配置

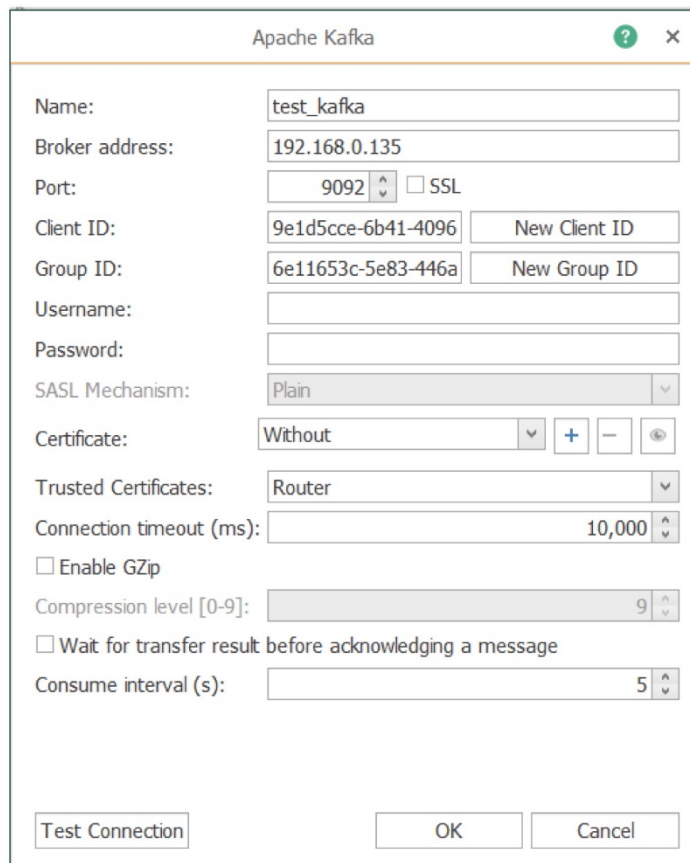
3.2.1. 新建 Plug-in

1) OPC Classic Plug-in

通过 OPC Classic Client 与 OPC DA Server 建立连接，读取到 OPC DA Serv 数据。

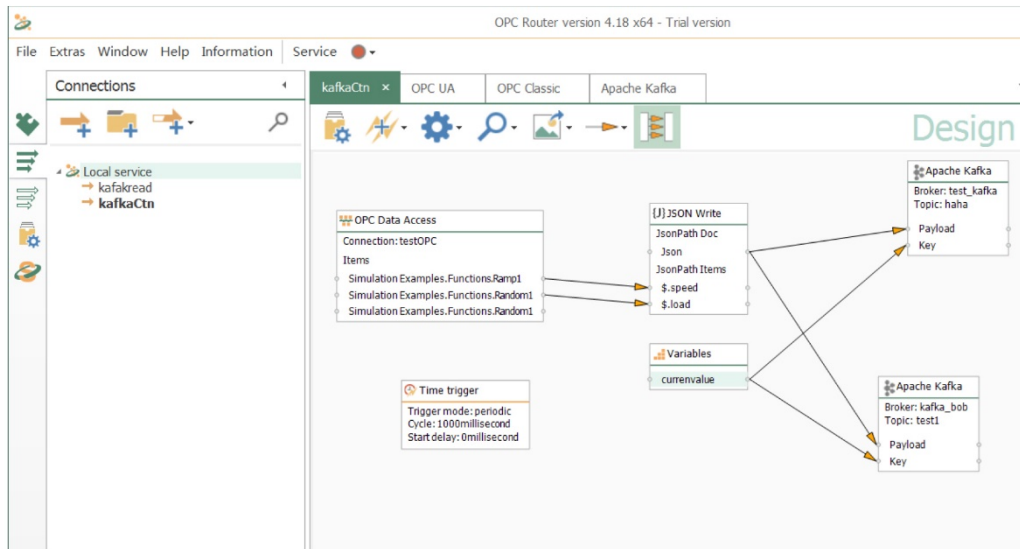


2) Apache Kafka Plug-in



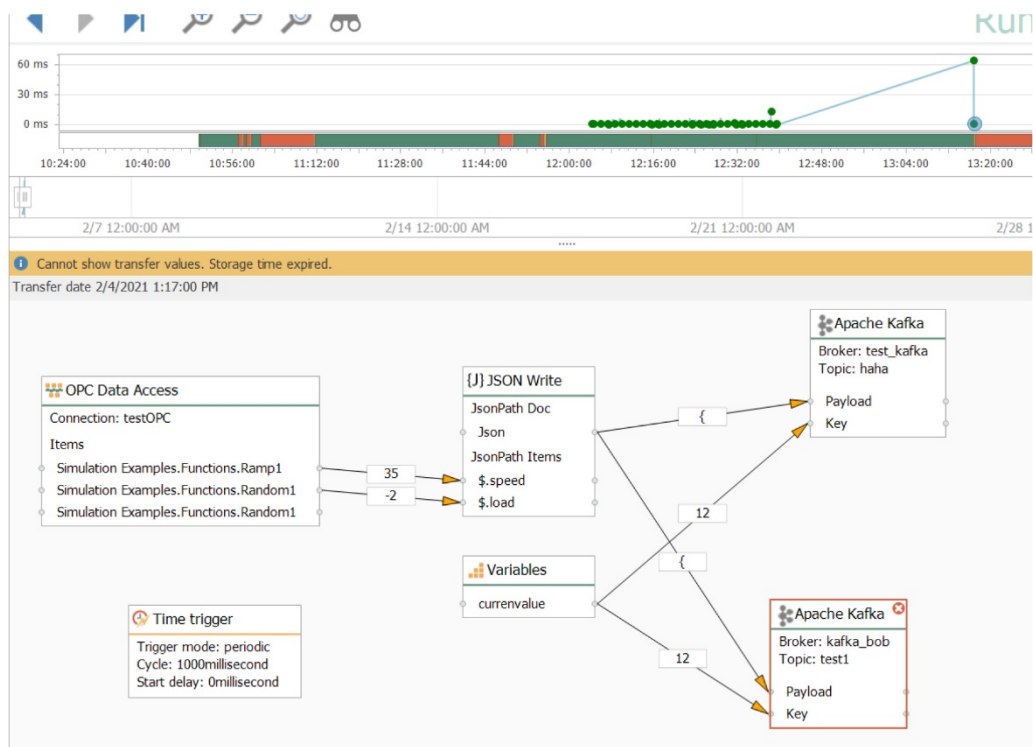
Broker address 为 Kafka 所在机器 IP

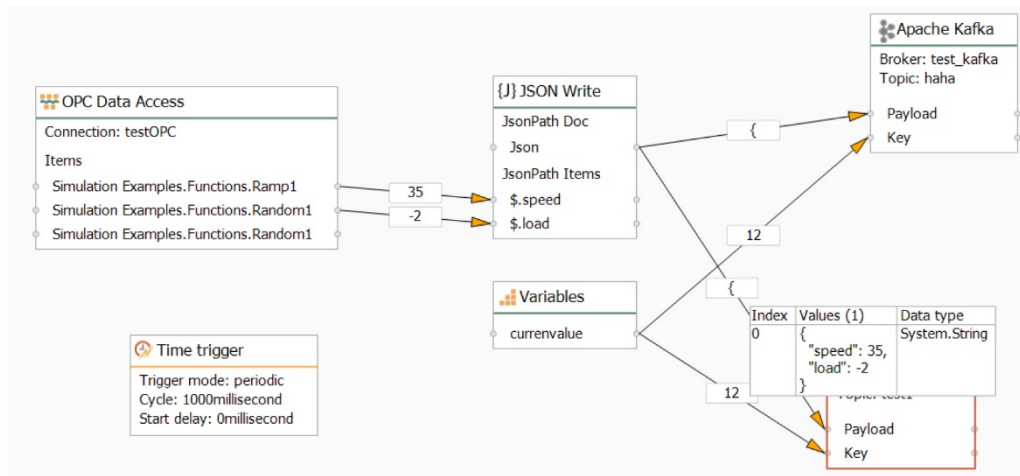
3.2.2. 新建 Connection



3.2.3. 运行与监视

OPC Router 软件中运行与监视界面：





读取 Kafka 中记录的数据信息。

```
kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic...
{
  "speed": 63,
  "load": 15
}
{
  "speed": 83,
  "load": 58
}
{
  "speed": 39,
  "load": 11
}
{
  "speed": 67,
  "load": 20
}
{
  "speed": 99,
  "load": 35
}
{
  "speed": 59,
  "load": 72
}
```