

**GBASE**

**Kafka 集群部署手册**



## **Kafka 集群部署手册，南大通用数据技术股份有限公司**

**GBase** 版权所有©2004-2017，保留所有权利。

### 版权声明

本文档所涉及的软件著作权、版权和知识产权已依法进行了相关注册、登记，由南大通用数据技术股份有限公司合法拥有，受《中华人民共和国著作权法》、《计算机软件保护条例》、《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权使用，不得非法使用。

### 免责声明

本文档包含的南大通用公司的版权信息由南大通用公司合法拥有，受法律的保护，南大通用公司对本文档可能涉及到的非南大通用公司的信息不承担任何责任。在法律允许的范围内，您可以查阅，并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经南大通用公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将视为侵权，南大通用公司具有依法追究其责任的权利。

本文档中包含的信息如有更新，恕不另行通知。您对本文档的任何问题，可直接向南大通用数据技术股份有限公司告知或查询。

未经本公司明确授予的任何权利均予保留。

### 通讯方式

南大通用数据技术股份有限公司

天津华苑产业区海泰发展六道 6 号海泰绿色产业基地 J 座(300384)

电话：400-013-9696            邮箱：info@gbase.cn

### 商标声明

**GBASE** 是南大通用数据技术股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由南大通用公司合法拥有，受法律保护。未经南大通用公司书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯南大通用公司商标权的，南大通用公司将依法追究其法律责任。

## 目 录

1	前言	1
2	安装软件包	1
2.1	JDK 配置	1
2.2	Kafka	1
2.2.1	安装	1
2.2.2	配置 Kafka 中 Zookeeper 集群	2
2.2.3	配置 Kafka 集群	3
3	启动	3
3.1	启动服务	3
3.2	测试 Kafka	4
4	创建 Topic	5
5	验证同步系统	5
6	异常问题处理	6
6.1	java 找不到 kafka.Kafka	6
6.1.1	现象	6
6.1.2	解决办法	7
6.2	Hostname 配置不统一	7
6.2.1	现象	7
6.2.2	解决办法	8
6.3	Kafka server 收不到远程 producer 消息	8
6.3.1	现象	8
6.3.2	解决办法	8
6.4	修改 broker id, Kafka server 启动失败	9
6.4.1	现象	9
6.4.2	解决办法	9
6.5	Kafka server 连接其他节点 zookeeper 失败	10
6.5.1	现象	10
6.5.2	解决办法	10
6.6	长时间使用消费脚本 kafka-console-consumer.sh 读数据导致 zookeeper 宕机	10
6.6.1	现象	10
6.6.2	解决办法	11
6.7	执行脚本报 Cannot allocate memory	11
6.7.1	现象	11

---

6.7.2	解决办法	11
6.8	无法生产或消费消息	11
6.8.1	现象	11
6.8.2	解决办法	11
6.9	执行脚本报 Too small initial heap for new size specified	12
6.9.1	现象	12
6.9.2	解决办法	12

# 1 前言

本文档描述 Goldengate\_Kafka 同步方案中, kafka 集群的部署 (以下用 3 节点 192.168.6.178-180 为例)

Kafka 版本: kafka\_2.11-0.10.1.0 (2.11 指 scala 版本号, 0.10.1.0 是 Kafka 的版本号)

OS: RedHat 6.2

JDK: jdk-7u80

## 2 安装软件包

java-1.7.0-openjdk-1.7.0.65-2.5.1.2.el6\_5.x86\_64.rpm

java-1.7.0-openjdk-devel-1.7.0.65-2.5.1.2.el6\_5.x86\_64.rpm

kafka\_2.11-0.10.1.0.tgz

### 2.1 JDK 配置

安装 java1.7.0

java-1.7.0-openjdk-1.7.0.65-2.5.1.2.el6\_5.x86\_64.rpm

java-1.7.0-openjdk-devel-1.7.0.65-2.5.1.2.el6\_5.x86\_64.rpm

相关 rpm 依赖

libjpeg-6b-46.el6.x86\_64

libjpeg-devel-6b-46.el6.x86\_64

### 2.2 Kafka

#### 2.2.1 安装

解压 kafka\_2.11-0.10.1.0.tgz

配置系统环境变量（以`~/.bash_profile`为例）

配置 kafka 路径

```
export kafka_home=/usr/local/kafka
```

配置 kafka 运行时的 jvm 环境（应根据机器性能进行调整，Xms：初始内存，Xmx：最大内存，Xmn：young generation 内存，配置过大或过小都会导致内存问题）

```
export KAFKA_HEAP_OPTS="-Xmx4G -Xms4G -Xmn2G -XX:PermSize=64m  
-XX:MaxPermSize=128m -XX:SurvivorRatio=6 -XX:CMSInitiatingOccupancyFraction=70  
-XX:+UseCMSInitiatingOccupancyOnly"
```

## 2.2.2 配置 Kafka 中 Zookeeper 集群

配置 kafka 中的 zookeeper（三个节点按照相同的配置）

```
mkdir -p $kafka_home/var/zookeeper  
  
# vim $kafka_home/config/zookeeper.properties （注意：这是 Kafka 的 config 目录中的 zookeeper.properties 文件）  
dataDir=/usr/local/kafka/zookeeper  
dataLogDir=/usr/local/kafka/var/zookeeper  
clientPort=2181  
maxClientCnxns=100  
tickTime=2000  
initLimit=10  
syncLimit=5  
server.1=192.168.6.178:2888:3888  
server.2=192.168.6.179:2888:3888  
server.3=192.168.6.180:2888:3888
```

在 dataDir 指定的路径上创建 myid 文件，

```
mkdir -p $kafka_home/zookeeper  
cd $kafka_home/zookeeper  
  
# vim myid  
1      （分配 zookeeper server 的 id 号，分别设置为 1,2,3）
```

## 2.2.3 配置 Kafka 集群

创建数据存储路径，配置三节点的 kafka

```
mkdir -p $kafka_home/logs
mkdir -p $kafka_home/var/kafka-logs

# vim $kafka_home/config/server.properties
broker.id=0    (每个节点要有不同的 broker id, 分别设置为 0, 1, 2)
delete.topic.enable=true    (允许管理人员删除 topic)
auto.create.topics.enable=false    (手动控制 topic 的创建, 否则 goldengate 可以自动创建 topic)
listeners=PLAINTEXT://192.168.6.178:9092    (配置为本节点的 ip)
log.dirs=/usr/local/kafka/var/kafka-logs
zookeeper.connect=192.168.6.178:2181, 192.168.6.179:2181, 192.168.6.180:2181
(配置所有 zookeeper 服务器的 ip, 注意逗号之间不要有空格)
```

对于 kafka 的 server.properties 中，log 存储数据的相关设置（此部分需要按现场需求针对性设置）

```
log.retention.hours=168    (多少个小时后, log 数据会被删除, 优先级低于 minutes 和 ms)
#log.retention.minutes=60    (多少分钟后, log 数据会被删除, 优先级低于 ms)
#log.retention.ms=6000    (多少毫秒后, log 数据会被删除, 如果被配置则优先使用)
log.retention.bytes=-1    (log 文件达到多少字节后, 数据会被删除。设置为 -1 则没有大小限制)
log.segment.bytes=1073741824    (每个 log 分片的大小, 超过后会新建分片)
log.roll.ms=60000    (间隔多久新创建一个数据文件)
log.retention.check.interval.ms=300000    (多长时间轮询检查是否会删除 log)
```

## 3 启动

### 3.1 启动服务

先启动三个节点的 zookeeper

```
$kafka_home/bin/zookeeper-server-start.sh  
$kafka_home/config/zookeeper.properties >$kafka_home/logs/zk.log &
```

再启动三个节点的 kafka

```
$kafka_home/bin/kafka-server-start.sh  
$kafka_home/config/server.properties >$kafka_home/logs/kk.log &
```

如下表示启动成功

```
# jps -m  
2930 QuorumPeerMain /usr/local/kafka/config/zookeeper.properties  
5188 Kafka /usr/local/kafka/config/server.properties  
5702 Jps -m
```

## 3.2 测试 Kafka

创建 kafka topic (其中--replication-factor 是配置互备节点数, 设置为 3 与集群节点数一致)

```
# $kafka_home/bin/kafka-topics.sh --create --partitions 1 --replication-factor  
3 --zookeeper 192.168.6.178:2181,192.168.6.179:2181,192.168.6.180:2181 --topic  
test1  
Created topic "test1".
```

检查 topic

```
# $kafka_home/bin/kafka-topics.sh --list --zookeeper  
192.168.6.178:2181,192.168.6.179:2181,192.168.6.180:2181  
test1
```

开启一个终端, 打开生产者

```
# $kafka_home/bin/kafka-console-producer.sh --broker-list 192.168.6.178:9092  
--topic test1  
hello world
```

开启一个终端, 打开消费者

```
# $kafka_home/bin/kafka-console-consumer.sh --zookeeper  
192.168.6.178:2181,192.168.6.179:2181,192.168.6.180:2181 --topic test1
```



```
hello world
```

如上所示，在生产者终端输入 hello world 并回车，可以在消费者终端看到 hello world 输出

## 4 创建 Topic

启动 Kafka 服务,创建两个 Topic,均在 goldengate 接收端的 kafka. props 中指定了

```
# $kafka_home/bin/kafka-topics.sh --zookeeper
192.168.6.178:2181,192.168.6.179:2181,192.168.6.180:2181 --create --topic
oggtopic --partitions 1 --replication-factor 3
Created topic "oggtopic".
```

验证 kafka topic 的情况

```
# $kafka_home/bin/kafka-topics.sh --zookeeper
192.168.6.178:2181,192.168.6.179:2181,192.168.6.180:2181 --list
oggtopic
# $kafka_home/bin/kafka-topics.sh --zookeeper
192.168.6.178:2181,192.168.6.179:2181,192.168.6.180:2181 --describe --topic
oggtopic
Topic:oggtopic PartitionCount:1 ReplicationFactor:1 Configs:
Topic: oggtopic Partition: 0 Leader: 0Replicas: 0 Isr: 0
```

## 5 验证同步系统

在任意 kafka 节点启动 consumer 脚本

```
$kafka_home/bin/kafka-console-consumer.sh --zookeeper
192.168.6.178:2181,192.168.6.179:2181,192.168.6.180:2181 --topic oggtopic
```

在 oracle 端插入一条数据

```
SQL> insert into gtest values(5, 5);
1 row created.
```

```
SQL> commit;  
Commit complete.
```

可以看到 consumer 正常接收数据

```
# $kafka_home/bin/kafka-console-consumer.sh --zookeeper  
192.168.6.178:2181,192.168.6.179:2181,192.168.6.180:2181 --topic oggtopic  
{"table": "BDTEST.GTEST", "op_type": "I", "op_ts": "2017-01-24  
07:17:44.000209", "current_ts": "2017  
-01-24T15:17:47.504000", "pos": "00000000010000001632", "primary_keys": ["A"], "  
after": {"A": "5", "B": "5"}},
```

## 6 异常问题处理

### 6.1 java 找不到 kafka.Kafka

#### 6.1.1 现象

```
# Exception in thread "main"  
java.lang.UnsupportedClassVersionError: kafka/Kafka : Unsupported  
major.minor version 51.0    at  
java.lang.ClassLoader.defineClass1(Native Method)  
    at java.lang.ClassLoader.defineClass(ClassLoader.java:634)  
    at  
java.security.SecureClassLoader.defineClass(SecureClassLoader.java:1  
42)  
    at  
java.net.URLClassLoader.defineClass(URLClassLoader.java:277)  
    at  
java.net.URLClassLoader.access$000(URLClassLoader.java:73)  
    at java.net.URLClassLoader$1.run(URLClassLoader.java:212)  
    at java.security.AccessController.doPrivileged(Native Method)  
    at  
java.net.URLClassLoader.findClass(URLClassLoader.java:205)  
    at java.lang.ClassLoader.loadClass(ClassLoader.java:321)
```

```
at
sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:294)
  at java.lang.ClassLoader.loadClass(ClassLoader.java:266)
  Could not find the main class: kafka.Kafka. Program will exit.
```

## 6.1.2 解决办法

安装 java1.7.0

```
java-1.7.0-openjdk-1.7.0.65-2.5.1.2.el6_5.x86_64.rpm
```

```
java-1.7.0-openjdk-devel-1.7.0.65-2.5.1.2.el6_5.x86_64.rpm
```

## 6.2 Hostname 配置不统一

### 6.2.1 现象

```
[2017-01-20 15:18:23,793] FATAL Fatal error during
KafkaServerStartable startup. Prepare to shutdown
(kafka.server.KafkaServerStartable) java.net.UnknownHostException:
V173: V173: Temporary failure in name resolution
  at java.net.InetAddress.getLocalHost(InetAddress.java:1473)
  at
kafka.server.KafkaHealthcheck$$anonfun$1.apply(KafkaHealthcheck.scala:61)
  at
kafka.server.KafkaHealthcheck$$anonfun$1.apply(KafkaHealthcheck.scala:59)
  at
scala.collection.MapLike$$MappedValues.get(MapLike.scala:249)
  at
scala.collection.MapLike$class.getOrElse(MapLike.scala:126)
  at scala.collection.AbstractMap.getOrElse(Map.scala:59)
  at
kafka.server.KafkaHealthcheck.register(KafkaHealthcheck.scala:69)
  at
```

```
kafka.server.KafkaHealthcheck.startup(KafkaHealthcheck.scala:51)
  at kafka.server.KafkaServer.startup(KafkaServer.scala:270)
  at
kafka.server.KafkaServerStartable.startup(KafkaServerStartable.scala:39)
  at kafka.Kafka$.main(Kafka.scala:67)
  at kafka.Kafka.main(Kafka.scala)
  Caused by: java.net.UnknownHostException: V173: Temporary failure
in name resolution
  at java.net.Inet6AddressImpl.lookupAllHostAddr(Native Method)
  at
java.net.InetAddress$1.lookupAllHostAddr(InetAddress.java:901)
  at
java.net.InetAddress.getAddressesFromNameService(InetAddress.java:1293)
  at java.net.InetAddress.getLocalHost(InetAddress.java:1469)
  ... 11 more
```

## 6.2.2 解决办法

统一/etc/hosts 和/etc/sysconfig/network 中的 hostname 配置。

## 6.3 Kafka server 收不到远程 producer 消息

### 6.3.1 现象

A 服务器生产 kafka 消息到 B 服务器的 kafka server，但是启动 consumer 发现没有收到消息。

### 6.3.2 解决办法

要在 kafka server 的配置文件里显式地指定 listener 的 ip

```
# vim $kafka_home/config/server.properties
```

```
listeners=PLAINTEXT://192.168.6.178:9092 （配置 kafka server 的
```

ip)

## 6.4 修改 broker id, Kafka server 启动失败

### 6.4.1 现象

```
查看 kk.log
[2017-01-24 14:48:55,511] FATAL Fatal error during
KafkaServerStartable startup. Prepare to s
hutdown (kafka.server.KafkaServerStartable)
kafka.common.InconsistentBrokerIdException: Configured broker.id
1 doesn't match stored broke
r.id 0 in meta.properties. If you moved your data, make sure your
configured broker.id matche
s. If you intend to create a new broker, you should remove all data
in your data directories
(log.dirs).
at
kafka.server.KafkaServer.getBrokerId(KafkaServer.scala:687)
at
kafka.server.KafkaServer.startup(KafkaServer.scala:213)
at
kafka.server.KafkaServerStartable.startup(KafkaServerStartable.scala
:39)
at kafka.Kafka$.main(Kafka.scala:67)
at kafka.Kafka.main(Kafka.scala)
```

### 6.4.2 解决办法

`$kafka_home/config/server.properties` 中修改了 broker id, 需要删除

log.dirs 中原有的数据再重新启动。

## 6.5 Kafka server 连接其他节点 zookeeper 失败

### 6.5.1 现象

Kafka 无法启动，在 log 中看到 hostname 无法识别

```
org.I0Itec.zkclient.exception.ZkException: Unable to connect to
gb69:2181, gb70:2181
```

```
Caused by: java.net.UnknownHostException: gb70: Name or service
not known
```

### 6.5.2 解决办法

修改 `$kafka_home/config/server.properties`，把逗号间的空格去掉  
`zookeeper.connect=gb69:2181,gb70:2181`。

## 6.6 长时间使用消费脚本

### kafka-console-consumer.sh 读数据导致

### zookeeper 宕机

#### 6.6.1 现象

长时间向 kafka 生产数据，同时使用 `kafka-console-consumer.sh` 不断读取数据。一段时间后，集群所有节点 kafka 和 zookeeper 宕机。查看作为 leader 节点 zookeeper 的 `zk.log`，报如下错误：

```
[2017-03-15 05:16:35,454] ERROR Unexpected exception causing
```

shutdown while sock still open

(org.apache.zookeeper.server.quorum.LearnerHandler)

## 6.6.2 解决办法

查看网上相关解决方案，是由 zookeeper 运行内存不足导致。在 `$kafka_home/config/zookeeper.properties` 中添加 `jute.maxbuffer=1000000000`，默认是 1M，增大后宕机发生的时间推迟了，没有从根本解决问题。

但如果只生产数据，不长时间运行 `kafka-console-consumer.sh`，则一切正常。怀疑是 kafka 消费脚本的 bug，此问题并不影响 8a 集群同步方案。

## 6.7 执行脚本报 Cannot allocate memory

### 6.7.1 现象

生产消息时，执行某些脚本会报出 `Cannot allocate memory`。

### 6.7.2 解决办法

内存不足，很可能是 jvm 中内存占用太高导致，需要调整 `KAFKA_HEAP_OPTS`。

## 6.8 无法生产或消费消息

### 6.8.1 现象

消费脚本或 kafka server log 中会报出类似 `java.lang.OutOfMemoryError: Java heap space` 的错误。

### 6.8.2 解决办法

消费时，为各个 partition 分配的初始内存总和过大，导致 `OutOfMemory`。

---

配置时总和不要超过 jvm 的内存

## 6.9 执行脚本报 Too small initial heap for new size specified

### 6.9.1 现象

启动 zookeeper 失败，\$kafka\_home/logs/zk.log 中有如下信息：

```
Error occurred during initialization of VM
```

```
Too small initial heap for new size specified
```

### 6.9.2 解决办法

Jvm 内存配置过大，减小 KAFKA\_HEAP\_OPTS 的 Xms、Xmx、Xmn。仍然报错可以尝试去掉 Xmn 的配置，使用系统默认。



# GBASE

南大通用数据技术股份有限公司  
General Data Technology Co., Ltd.



微博二维码



微信二维码

