分布式消息服务 Kafka

性能白皮书

文档版本 01

发布日期 2021-06-04

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 测试场景

本文从批处理大小、是否跨AZ生产、副本数、同步/异步复制的维度进行分布式消息服务Kafka的性能测试，对比客户端消息生产速率和服务端CPU消耗，得出性能测试结果。

- 测试场景一（批处理大小）：相同的Kafka专享版实例和Topic，不同的batch.size
- 测试场景二（是否跨AZ生产）：相同的Kafka专享版实例和Topic，生产客户端和服务端在不同的AZ中
- 测试场景三（副本数）：相同的Kafka专享版实例，不同的Topic副本数
- 测试场景四（同步/异步复制）：相同的Kafka专享版实例，不同复制机制的Topic

2 测试环境

进行性能测试前，您需要先构建如下的测试环境：

步骤一：购买 Kafka 专享版实例

购买一个Kafka专享版实例，参数信息如下，购买方法请参考**购买实例**。

- 区域：华北-北京四
- 项目：华北-北京四
- 可用区：可用区1
- 实例名称：kafka-test
- 版本：2.3.0
- 基准带宽：100MB/s，3节点，底层资源类型为c6.large.2
- 存储空间：超高I/O，600GB
- 容量阈值策略：自动删除
- 虚拟私有云：选择虚拟私有云，如果未创建，请参考**准备实例依赖资源**创建。
- 安全组：选择安全组，安全组需要满足分布式消息服务Kafka的要求，具体请参考**准备实例依赖资源**。
- Manager用户名：设置登录Kafka Manager的用户名。
- 密码：设置登录Kafka Manager的密码。
- 更多配置：不开启“公网访问”、“转储”、“Kafka SASL_SSL”和“Kafka自动创建Topic”

购买完成后，在实例详情页获取Kafka专享版实例的地址。

步骤二：创建 Topic

在购买的**Kafka专享版实例**中，创建如下参数的3个Topic，具体步骤请参考**创建Topic**。

- Topic-01：3分区1副本，异步复制
- Topic-02：3分区3副本，异步复制
- Topic-03：3分区3副本，同步复制

步骤三：获取测试工具

获取**Kafka命令行工具2.3.0版本**。

步骤四：购买客户端服务器

购买如下参数的2台ECS服务器，具体步骤请参考**购买弹性云服务器**。

- 区域、可用区、虚拟私有云、子网、安全组与**步骤一：购买Kafka专享版实例**保持一致，规格为4U8G，Linux系统的ECS。
- 区域、虚拟私有云、子网、安全组与**Kafka专享版实例**保持一致，“可用区”为“可用区2”，规格为4U8G，Linux系统的ECS。

购买完成ECS后，需要在ECS中完成以下配置：

- 安装**Java JDK**，并配置JAVA_HOME与PATH环境变量。
```
export JAVA_HOME=/root/jdk1.8.0_231
export PATH=$JAVA_HOME/bin:$PATH
```
- 下载**Kafka命令行工具2.3.0版本**，并解压。
```
tar -zxf kafka_2.11-2.3.0.tgz
```

3 测试步骤

测试以下四种场景下，客户端消息生产速率和服务端CPU消耗。

- 测试场景一（批处理大小）：相同的Kafka专享版实例和Topic，不同的batch.size
- 测试场景二（是否跨AZ生产）：相同的Kafka专享版实例和Topic，生产客户端和服务端在不同的AZ中
- 测试场景三（副本数）：相同的Kafka专享版实例，不同的Topic副本数
- 测试场景四（同步/异步复制）：相同的Kafka专享版实例，不同复制机制的Topic

表 3-1 测试参数

分区数	副本数	是否同步复制	batch.size	是否跨AZ生产
3	1	否	1KB	否
3	1	否	16KB	否
3	1	否	1KB	是
3	3	是	1KB	否
3	3	否	1KB	否

测试脚本如下：

```
./kafka-producer-perf-test.sh --producer-props bootstrap.servers=${连接地址} acks=1 batch.size=${batch.size} linger.ms=0 --topic ${Topic名称} --num-records ${num-records} --record-size 1024 --throughput -102400
```

- bootstrap.servers：**购买Kafka专享版实例**中获取的Kafka专享版实例的地址。
- acks：消息主从同步策略，acks=1表示异步复制消息，acks=-1表示同步复制消息。
- batch.size：每次批量发送消息的大小（单位为字节）。
- linger.ms：两次发送时间间隔。
- topic：**创建Topic**中设置的Topic名称。
- num-records：总共需要发送的消息数。

- record-size：每条消息的大小。
- throughput：每秒发送的消息数。

测试场景一：批处理大小

步骤1 登录客户端服务器，进入“kafka_2.11-2.3.0/bin”目录下，执行以下脚本。

batch.size=1KB，执行脚本如下：

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-01 --num-records 8000000 --record-size 1024 --throughput 102400
```

执行结果如下：

```
8000000 records sent, 34128.673632 records/sec (33.33 MB/sec), 879.91 ms avg latency, 4102.00 ms max
latency, 697 ms 50th, 2524 ms 95th, 2888 ms 99th, 4012 ms 99.9th.
```

客户端消息生产速率=34128

batch.size=16KB，执行脚本如下：

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=16384
linger.ms=0 --topic Topic-01 --num-records 100000000 --record-size 1024 --throughput 10240
```

执行结果如下：

```
100000000 records sent, 102399.318430 records/sec (100.00 MB/sec), 4.72 ms avg latency, 914.00 ms max
latency, 1 ms 50th, 5 ms 95th, 162 ms 99th, 398 ms 99.9th.
```

客户端消息生产速率=102399

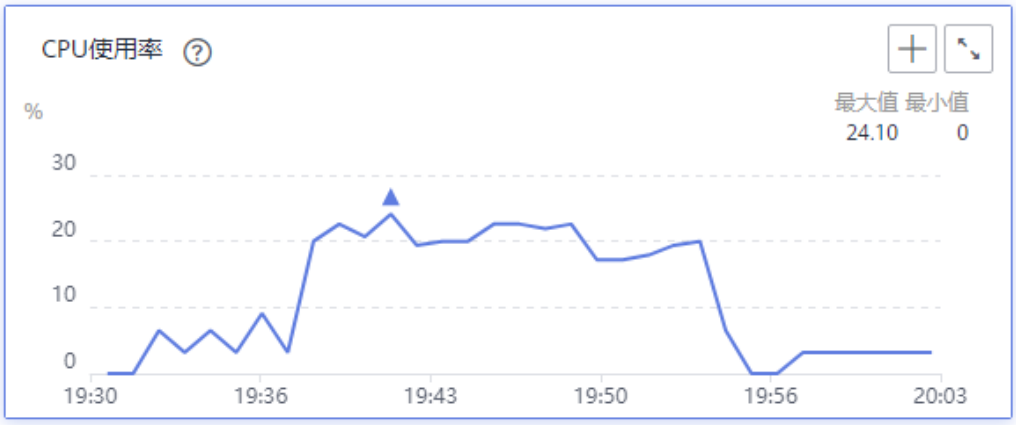
步骤2 登录Kafka专享版实例控制台，在测试实例所在行，选择“查看监控数据”，跳转到云监控界面。

步骤3 在“节点”页签，查看服务端leader节点的CPU使用率。

图 3-1 broker-0 的 CPU 使用率（batch.size=1KB）

CPU消耗=58.10%

图 3-2 broker-0 的 CPU 使用率（batch.size=16KB）

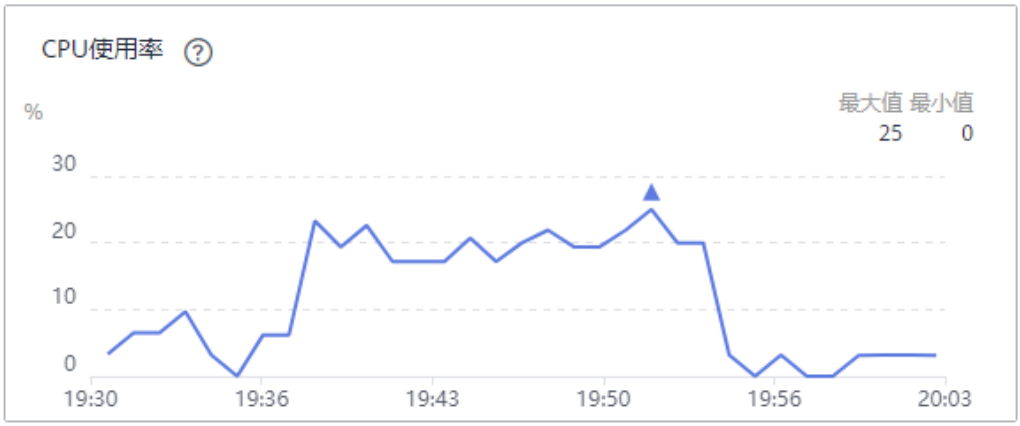


CPU消耗=24.10%

图 3-3 broker-1 的 CPU 使用率（batch.size=1KB）

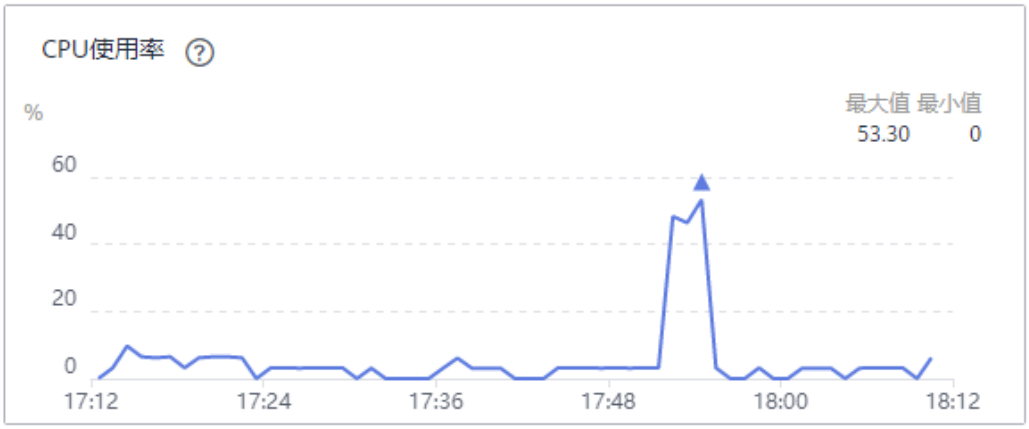
CPU消耗=56.70%

图 3-4 broker-1 的 CPU 使用率（batch.size=16KB）



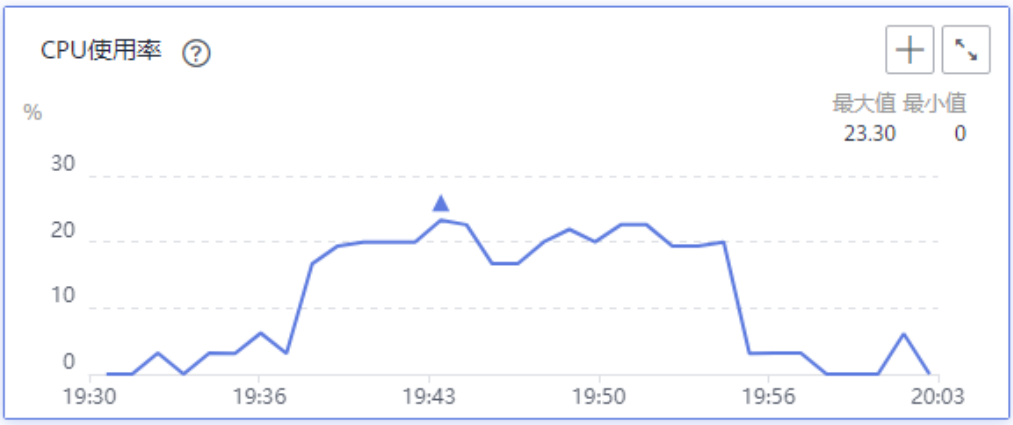
CPU消耗=25%

图 3-5 broker-2 的 CPU 使用率（batch.size=1KB）



CPU消耗=53.30%

图 3-6 broker-2 的 CPU 使用率（batch.size=16KB）



CPU消耗=23.30%

----结束

测试场景二：是否跨 AZ 生产

步骤1 登录客户端服务器，进入“kafka_2.11-2.3.0/bin”目录下，执行以下脚本。

客户端服务器和实例在相同的AZ中，执行脚本如下：

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-01 --num-records 8000000 --record-size 1024 --throughput 102400
```

执行结果如下：

```
8000000 records sent, 34128.673632 records/sec (33.33 MB/sec), 879.91 ms avg latency, 4102.00 ms max
latency, 697 ms 50th, 2524 ms 95th, 2888 ms 99th, 4012 ms 99.9th.
```

客户端消息生产速率=34128

客户端服务器和实例在不同的AZ中，执行脚本如下：

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-01 --num-records 4000000 --record-size 1024 --throughput 102400
```

执行结果如下：

```
4000000 records sent, 8523.042044 records/sec (8.32 MB/sec), 3506.20 ms avg latency, 11883.00 ms max
latency, 1817 ms 50th, 10621 ms 95th, 11177 ms 99th, 11860 ms 99.9th.
```

客户端消息生产速率=8523

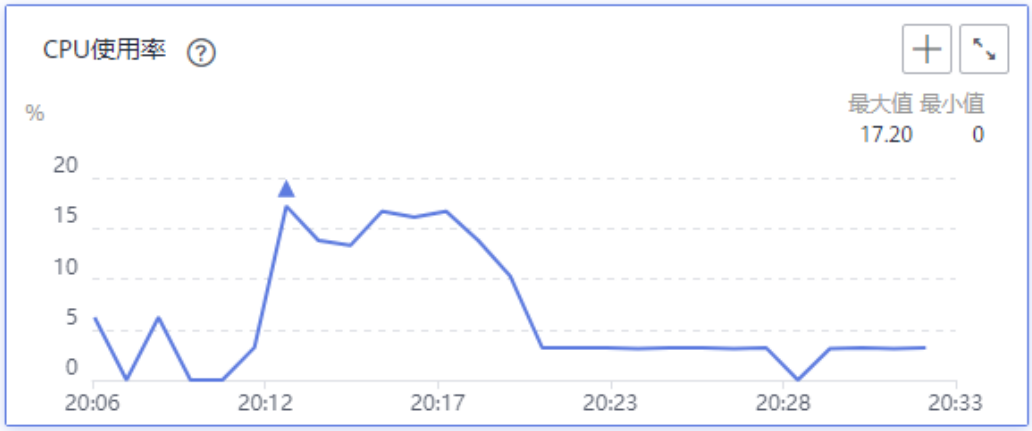
步骤2 登录Kafka专享版实例控制台，在测试实例所在行，选择“更多 > 查看监控数据”，跳转到云监控界面。

步骤3 在“节点”页签，查看服务端leader节点的CPU使用率。

图 3-7 broker-0 的 CPU 使用率（客户端服务器和实例在相同的 AZ 中）

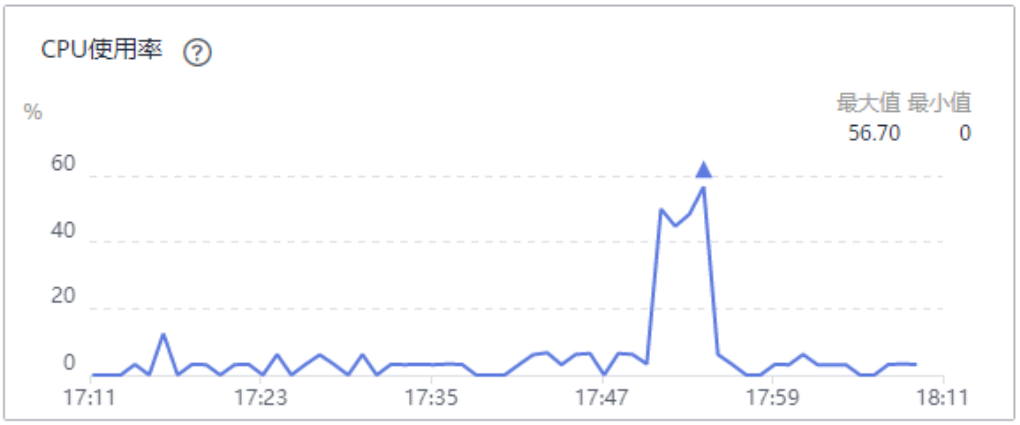
CPU消耗=58.10%

图 3-8 broker-0 的 CPU 使用率（客户端服务器和实例在不同的 AZ 中）



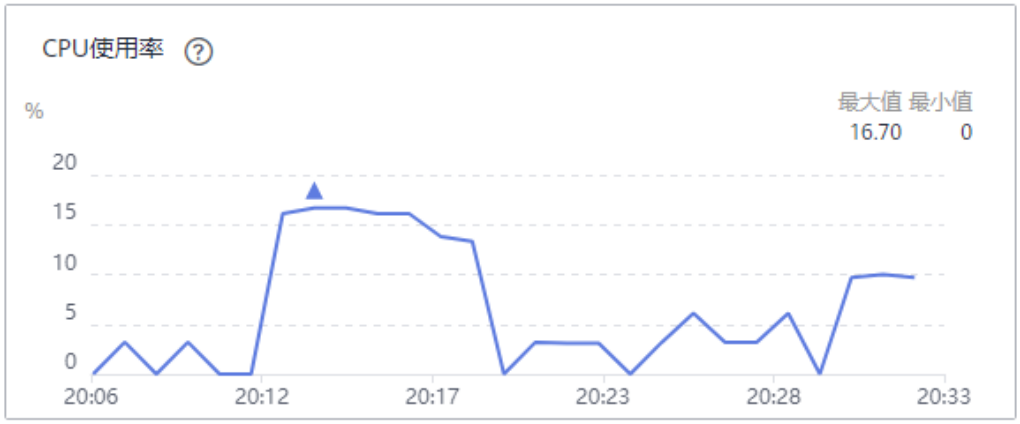
CPU消耗=17.20%

图 3-9 broker-1 的 CPU 使用率（客户端服务器和实例在相同的 AZ 中）



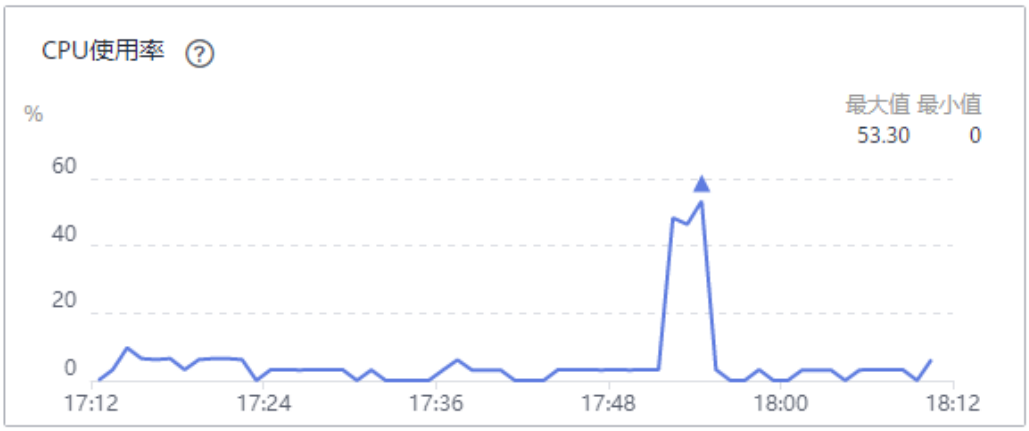
CPU消耗=56.70%

图 3-10 broker-1 的 CPU 使用率（客户端服务器和实例在不同的 AZ 中）



CPU消耗=16.70%

图 3-11 broker-2 的 CPU 使用率（客户端服务器和实例在相同的 AZ 中）



CPU消耗=53.30%

图 3-12 broker-2 的 CPU 使用率（客户端服务器和实例在不同的 AZ 中）

CPU消耗=18.80%

----结束

测试场景三：副本数

步骤1 登录客户端服务器，进入“kafka_2.11-2.3.0/bin”目录下，执行以下脚本。

1副本，执行脚本如下：

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-01 --num-records 8000000 --record-size 1024 --throughput 102400
```

执行结果如下：

```
8000000 records sent, 34128.673632 records/sec (33.33 MB/sec), 879.91 ms avg latency, 4102.00 ms max
latency, 697 ms 50th, 2524 ms 95th, 2888 ms 99th, 4012 ms 99.9th.
```

客户端消息生产速率=34128

3副本，执行脚本如下：

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-02 --num-records 4000000 --record-size 1024 --throughput 102400
```

执行结果如下：

```
4000000 records sent, 14468.325219 records/sec (14.13 MB/sec), 2069.99 ms avg latency, 7911.00 ms max
latency, 846 ms 50th, 6190 ms 95th, 6935 ms 99th, 7879 ms 99.9th.
```

客户端消息生产速率=14468

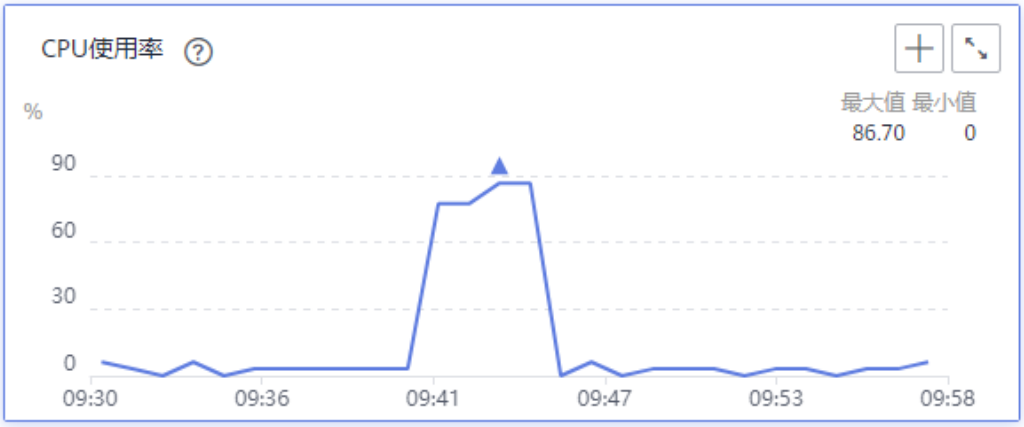
步骤2 登录Kafka专享版实例控制台，在测试实例所在行，选择“更多 > 查看监控数据”，跳转到云监控界面。

步骤3 在“节点”页签，查看服务端leader节点的CPU使用率。

图 3-13 broker-0 的 CPU 使用率（1 副本）

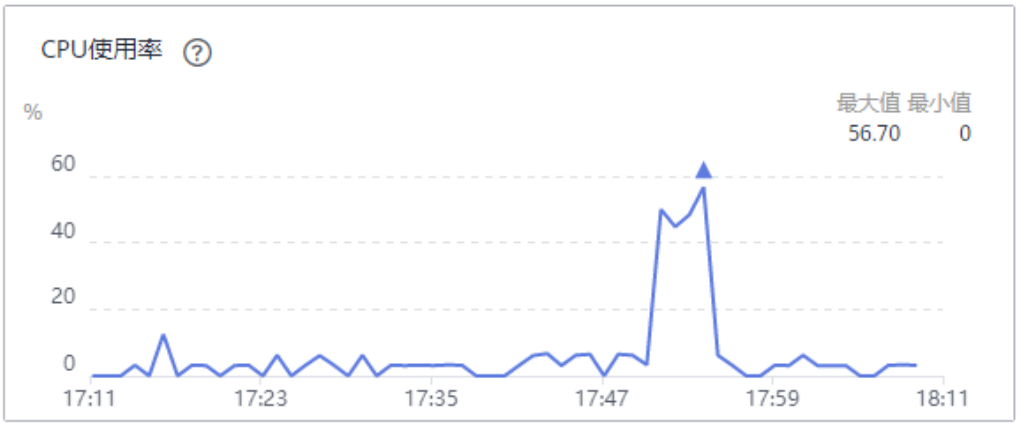
CPU消耗=58.10%

图 3-14 broker-0 的 CPU 使用率（3 副本）



CPU消耗=86.70%

图 3-15 broker-1 的 CPU 使用率（1 副本）

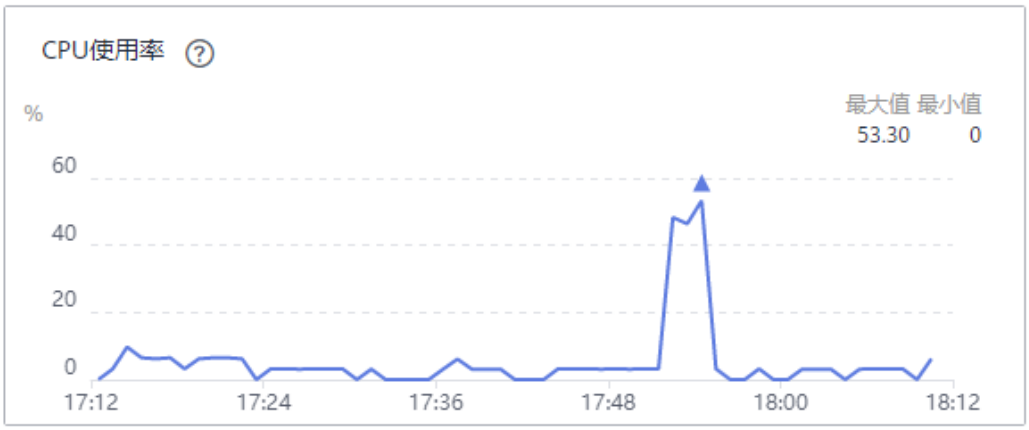


CPU消耗=56.70%

图 3-16 broker-1 的 CPU 使用率（3 副本）

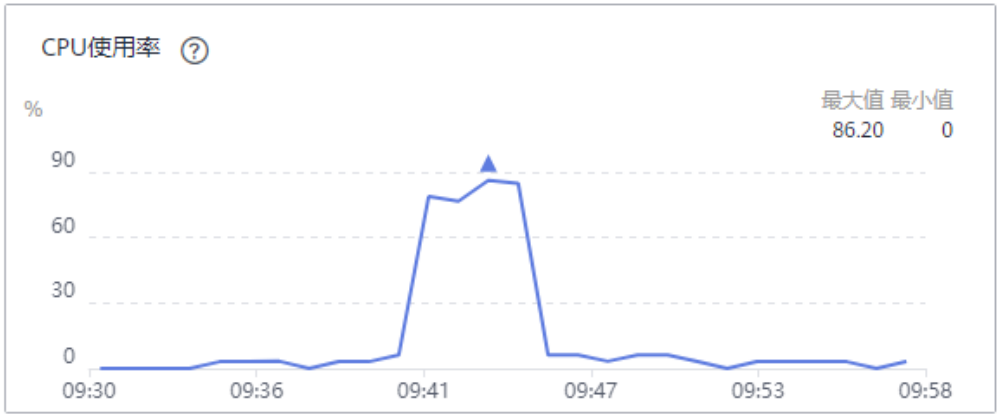
CPU消耗=80.60%

图 3-17 broker-2 的 CPU 使用率（1 副本）



CPU消耗=53.30%

图 3-18 broker-2 的 CPU 使用率（3 副本）



CPU消耗=86.20%

----结束

测试场景四：同步/异步复制

步骤1 登录客户端服务器，进入“kafka_2.11-2.3.0/bin”目录下，执行以下脚本。

异步复制，执行脚本如下：

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=1 batch.size=1024
linger.ms=0 --topic Topic-02 --num-records 4000000 --record-size 1024 --throughput 102400
```

执行结果如下：

```
4000000 records sent, 14468.325219 records/sec (14.13 MB/sec), 2069.99 ms avg latency, 7911.00 ms max
latency, 846 ms 50th, 6190 ms 95th, 6935 ms 99th, 7879 ms 99.9th.
```

客户端消息生产速率=14468

同步复制，执行脚本如下：

```
./kafka-producer-perf-test.sh --producer-props
bootstrap.servers=192.168.0.69:9092,192.168.0.42:9092,192.168.0.66:9092 acks=-1 batch.size=1024
linger.ms=0 --topic Topic-03 --num-records 1000000 --record-size 1024 --throughput 102400
```

执行结果如下：

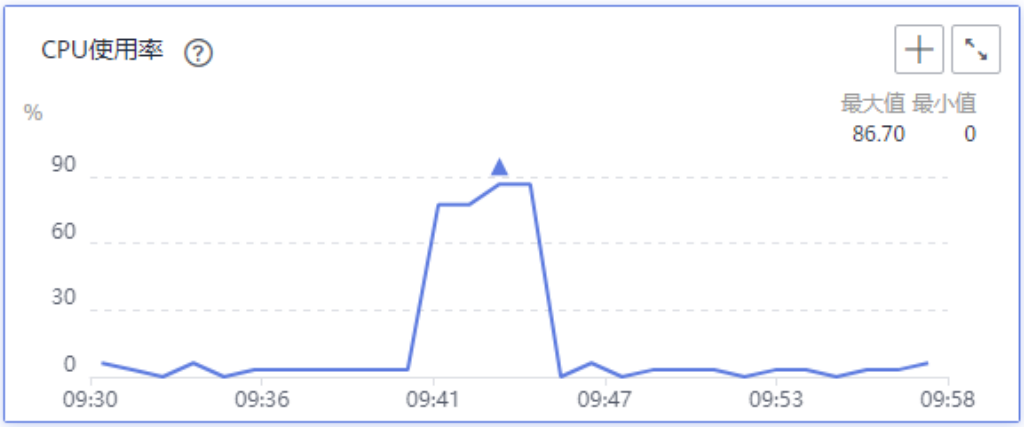
```
1000000 records sent, 3981.937930 records/sec (3.89 MB/sec), 7356.98 ms avg latency, 19013.00 ms max
latency, 6423 ms 50th, 14381 ms 95th, 18460 ms 99th, 18975 ms 99.9th.
```

客户端消息生产速率=3981

步骤2 登录Kafka专享版实例控制台，在测试实例所在行，选择“更多 > 查看监控数据”，跳转到云监控界面。

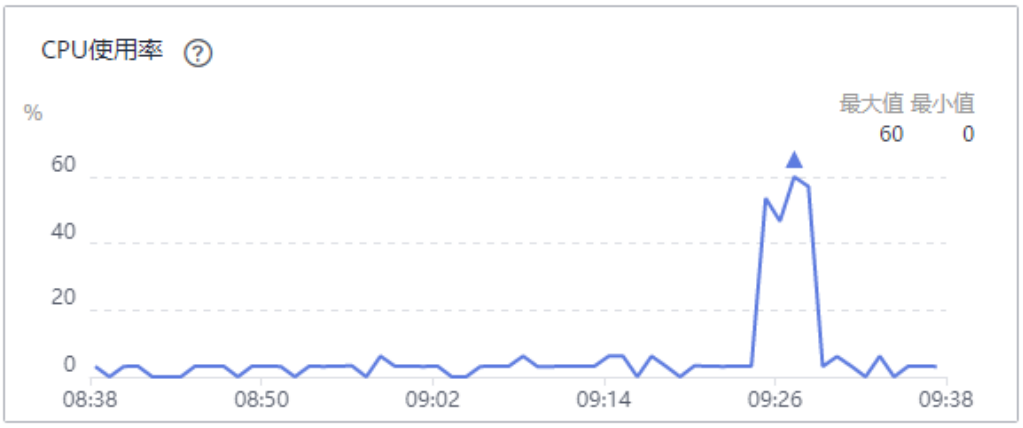
步骤3 在“节点”页面，查看服务端leader节点的CPU使用率。

图 3-19 broker-0 的 CPU 使用率（异步复制）



CPU消耗=86.70%

图 3-20 broker-0 的 CPU 使用率（同步复制）

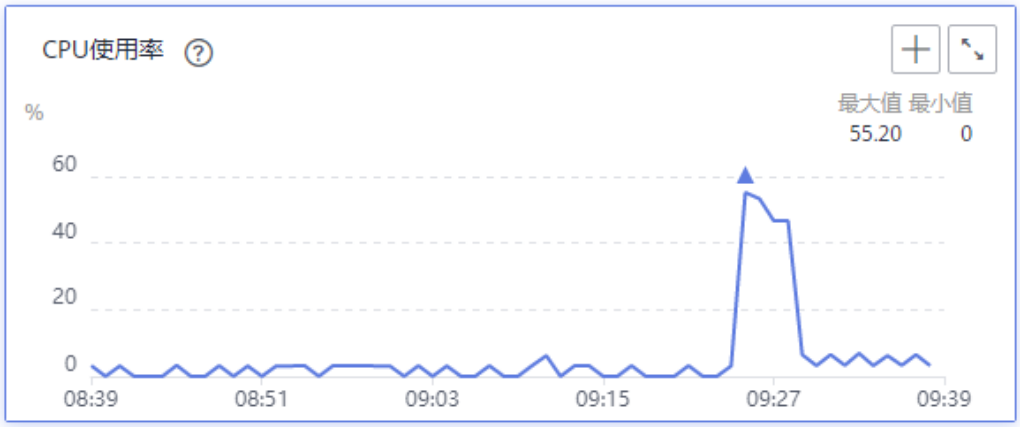


CPU消耗=60%

图 3-21 broker-1 的 CPU 使用率（异步复制）

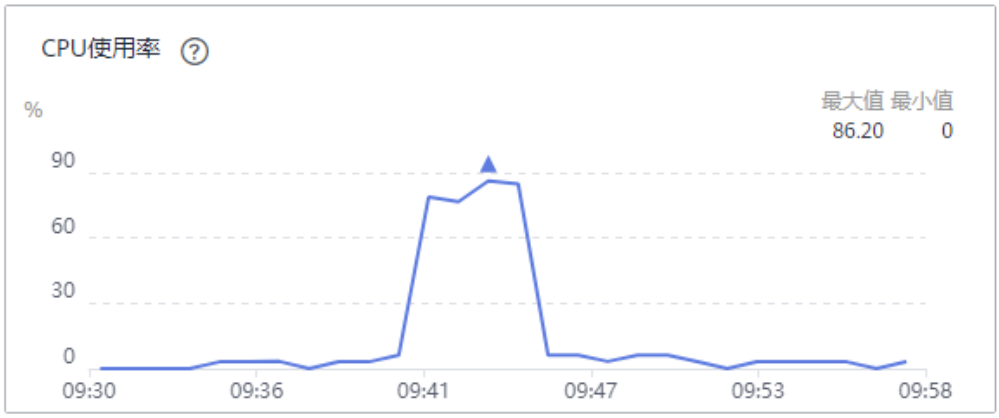
CPU消耗=80.60%

图 3-22 broker-1 的 CPU 使用率（同步复制）



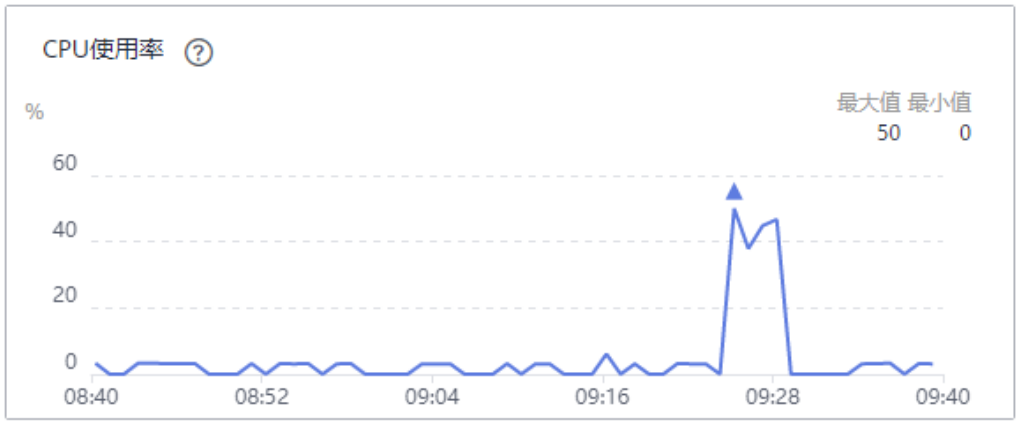
CPU消耗=55.20%

图 3-23 broker-2 的 CPU 使用率（异步复制）



CPU消耗=86.20%

图 3-24 broker-2 的 CPU 使用率（同步复制）



CPU消耗=50%

----结束

4 测试结果

表 4-1 测试结果

分区数	副本数	是否同步复制	batch.size	是否跨AZ生产	客户端消息生产速率	服务端CPU消耗（broker-0）	服务端CPU消耗（broker-1）	服务端CPU消耗（broker-2）
3	1	否	1KB	否	34128	58.10%	56.70%	53.30%
3	1	否	16KB	否	102399	24.10%	25.00%	23.30%
3	1	否	1KB	是	8523	17.20%	16.70%	18.80%
3	3	是	1KB	否	3981	60.00%	55.20%	50.00%
3	3	否	1KB	否	14468	86.70%	80.60%	86.20%

通过上表的测试结果，得出以下结论，仅供参考：

- 生产请求的batch.size变大16倍时，客户端消息生产速率增加，服务端CPU消耗减少。
- 同AZ生产和跨AZ生产相比，客户端消息生产速率增加，服务端CPU消耗也随之增加。
- 副本从1变成3时，客户端消息生产速率下降较多，服务端CPU消耗增加。
- 异步复制的Topic和同步复制的Topic相比，客户端消息生产速率增加，服务端CPU消耗也随之增加。