

# Kafka 性能测试实例

## 1. 概述

在分布式实时数据流场景下，随着数据量的增长，对 Kafka 集群的性能和稳定性的要求也很高。本篇博客将从生产者和消费者两方面来做性能测试，针对具体的业务和数据量，来调优 Kafka 集群。

## 2. 内容

### 2.1 测试环境

本次测试的环境信息由三台物理机组成，具体信息如下所示：

主机名	Kafka 版本	CPU	内存	磁盘	网卡
dn1	0.10.2.0	32 核	64GB	12*4T	千兆
dn2	0.10.2.0	32 核	64GB	12*4T	千兆
dn3	0.10.2.0	32 核	64GB	12*4T	千兆

### 2.2 测试工具

Kafka 系统提供了测试工具 `kafka-producer-perf-test.sh` 和 `kafka-consumer-perf-test.sh`，通过该工具可以对生产者性能和消费者性能进行测试，获取一组最佳的参数值，进而提升生产者的发送效率和消费者的读取效率。这里如果需要实现带有线程参数功能的工具，可以修改工具源代码，新建一个 `kafka-producer-perf-test-0.8.sh` 脚本，实现内容如下：

```
# 使用老版本的 ProducerPerformance 工具类
exec $(dirname $0)/kafka-run-class.sh kafka.tools.ProducerPerformance "$@"
```

### 2.2.1 生产者测试参数

参 数	说 明
--topic	指定生产者发送消息的主题
--num-records	测试时发送消息的总记录数
--throughput	最大消息吞吐量
--producer-props	通过键值对的方式指定配置属性，多组配置属性用空格分隔
--producer.config	加载生产者配置文件
--record-size	每条消息字节大小

### 2.2.2 消费者测试参数

参 数	说 明
--topic	指定消费者读取消息的主题
--zookeeper	指定字符串连接 Zookeeper 集群来获取 Kafka 集群元数据信息
--threads	指定线程数
--messages	读取消息记录数
--group	指定消费者组
--batch-size	执行批处理大小

## 3. 生产者测试

生产者测试，分别从线程数、分区数、副本数、Broker 数、同步与异步模式、批处理大小、消息长度大小、数据压缩等维度来进行。

### 3.1 线程数

创建一个拥有 6 个分区、1 个副本的 Topic，设置不同的线程数并发送相同的数据量，查看性能变化。测试脚本如下：

```
# 创建主题
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 --topic test_producer_perf --partitions 6 --replication-
factor 1

# 设置 1 个线程数
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf --threads 1 --broker-list dn1:9092,
dn2:9092,
```

dn3:9092

# 设置 10 个线程数

```
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000  
--topics test_producer_perf --threads 10 --broker-list dn1:9092,  
dn2:9092, dn3:9092
```

# 设置 20 个线程数

```
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000  
--topics test_producer_perf --threads 20 --broker-list dn1:9092,  
dn2:9092, dn3:9092
```

# 设置 25 个线程数

```
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000  
--topics test_producer_perf --threads 25 --broker-list dn1:9092,  
dn2:9092, dn3:9092
```

# 设置 30 个线程数

```
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000  
--topics test_producer_perf --threads 30 --broker-list dn1:9092,  
dn2:9092, dn3:9092
```



### 3.1.1 测试结果

Kafka 节点数	线程数	分区数	副本数	每秒发送消息	消息大小 (MB/s)
3	1	6	1	44287.7642	4.2236
3	10	6	1	349113.2523	33.2940
3	20	6	1	523889.3546	49.9620
3	25	6	1	526094.2761	50.1723
3	30	6	1	493435.3104	47.0577

### 3.1.2 结论

向一个拥有 6 个分区、1 个副本的 Topic 中，发送 500 万条消息记录时，随着线程数的增加，每秒发送的消息记录会逐渐增加。在线程数为 25 时，每秒发送的消息记录达到最佳值，随后再增加线程数，每秒发送的消息记录数反而会减少。

## 3.2 分区数

- (1) 新建一个拥有 12 个分区、1 个副本的主题；
- (2) 新建一个拥有 24 个分区、1 个副本的主题；
- (3) 向拥有 12 个分区、1 个副本的主题中发送相同数量的消息记录，查看性能变化；
- (4) 向拥有 24 个分区、1 个副本的主题中发送相同数量的消息记录，查看性能变化。

执行命令如下：

```
# 创建一个拥有 12 个分区主题
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 --topic test_producer_perf_p12 --partitions 12
--replication-factor 1
# 创建一个拥有 24 个分区主题
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 --topic test_producer_perf_p24 --partitions 24
--replication-factor 1

# 用一个线程发送数据到拥有 12 个分区主题中
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_p12 --threads 1 --broker-list
dn1:9092,
dn2:9092, dn3:9092

# 用一个线程发送数据到拥有 24 个分区主题中
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_p24 --threads 1 --broker-list
dn1:9092,
dn2:9092, dn3:9092
```

### 3.2.1 测试结果

Kafka 节点数	线程数	分区数	副本数	每秒发送消息	消息大小 (MB/s)
3	1	6	1	44287.7642	4.2236
3	1	12	1	43373.7866	4.1364
3	1	24	1	33497.0900	3.1946

### 3.2.2 结论

从测试结果来看，分区数越多，单线程生产者的吞吐量越小。

## 3.3 副本数

- (1) 创建一个拥有两个副本、6 个分区的主题；
- (2) 创建一个拥有 3 个副本、6 个分区的主题；
- (3) 向拥有两个副本、6 个分区的主题中发送相同数量的消息记录，查看性能变化；
- (4) 向拥有 3 个副本、6 个分区的主题中发送相同数量的消息记录，查看性能变化；

执行命令如下：

```
# 创建一个拥有两个副本、6 个分区的话题
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 --topic test_producer_perf_r2 --partitions 6
--replication-factor 2

# 创建一个拥有 3 个副本、6 个分区的话题
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 --topic test_producer_perf_r3 --partitions 6
--replication-factor 3

# 用 3 个线程发送数据到拥有两个副本的主题中
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_r2 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092

# 用 3 个线程发送数据到拥有 3 个副本的主题中
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_r3 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092
```

### 3.3.1 测试结果

Kafka 节点数	线程数	分区数	副本数	每秒发送消息	消息大小 (MB/s)
3	3	6	1	131309.3650	12.5226
3	3	6	2	83944.6972	8.0056
3	3	6	3	54205.8087	5.1696

### 3.3.2 结论

从测试结果来看，副本数越多，吞吐量越小。

## 3.4 Broker 数量

通过增加 Broker 节点数量来查看性能变化，脚本如下：

```
# Kafka 节点数为 4 个时，异步发送消息记录
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_b3 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092, dn4:9092 --batch-size 3000 --request-timeout-ms
100000
```

### 3.4.1 测试结果

Kafka 节点数	线程数	分区数	副本数	每秒发送消息	消息大小 (MB/s)
3	3	6	3	133865.1709	12.7664
4	3	6	3	157878.0549	15.0564

### 3.4.2 结论

从测试结果来看，增加 Kafka Broker 数量，吞吐量会增加。

## 3.5 同步与异步模式

分别使用同步和异步模式发送相同数量的消息记录，查看性能变化。执行脚本如下：

```
# 创建一个有用 3 个副本、6 个分区的主题
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 --topic test_producer_perf_s2 --partitions 6
--replication-factor 3

# 使用同步模式发送消息数据
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_s2 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092 --sync

# 使用异步模式发送消息记录
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_s2 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092
```

### 3.5.1 测试结果

发送方式	Kafka 节点数	线程数	分区数	副本数	每秒发送消息	消息大小 (MB/s)
同步	3	3	6	3	17870.8838	1.6801
异步	3	3	6	3	51782.3277	4.8383

### 3.5.2 结论

从测试结果来看，使用异步模式发送消息数据，比使用同步模式发送消息数据，吞吐量是同步模式的 3 倍左右。

## 3.6 批处理大小

使用异步模式发送相同数量的消息数据，改变批处理量的大小，查看性能变化，执行脚本如下：

```
# 以批处理模式发送，大小为 1000 条
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_s2 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092 --batch-size 1000 --request-timeout-ms 100000

# 以批处理模式发送，大小为 3000 条
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_s2 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092 --batch-size 3000 --request-timeout-ms 100000

# 以批处理模式发送，大小为 5000 条
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_s2 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092 --batch-size 5000 --request-timeout-ms 100000

# 以批处理模式发送，大小为 7000 条
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_s2 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092 --batch-size 7000 --request-timeout-ms 100000
```

### 3.6.1 测试结果

批处理大小	Kafka 节点数	线程数	分区数	副本数	每秒发送消息	消息大小 (MB/s)
1000	3	3	6	1	191204.5124	18.2347
3000	3	3	6	1	234796.8068	22.3920
5000	3	3	6	1	238015.8043	22.6990
7000	3	3	6	1	219076.5605	20.8919

### 3.6.2 结论

从测试的结果来看，发送的消息随着批处理大小增加而增加。当批处理大小增加到 3000~5000 时，吞吐量达到最佳值。而后再增加批处理大小，吞吐量的性能会下降。

## 3.7 消息长度的大小

改变消息的长度大小，查看性能变化，执行脚本如下：



```
# 发送消息，长度为 100 字节
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_s2 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092 --batch-size 3000 --request-timeout-ms 100000
--message-size 100

# 发送消息，长度为 200 字节
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_s2 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092 --batch-size 3000 --request-timeout-ms 100000
--message-size 200

# 发送消息，长度为 500 字节
[hadoop@dn1 ~]$ kafka-producer-perf-test-0.8.sh --messages 5000000
--topics test_producer_perf_s2 --threads 3 --broker-list dn1:9092,
dn2:9092, dn3:9092 --batch-size 3000 --request-timeout-ms 100000
--message-size 500
```



### 3.7.1 测试结果

消息长度	Kafka 节点数	线程数	分区数	副本数	每秒发送消息	消息大小 (MB/s)
100	3	3	6	1	234796.8086	22.3920
200	3	3	6	1	202765.6434	38.6745
500	3	3	6	1	117602.7378	56.0774

### 3.7.2 结论

从测试结果来看，随着消息长度的增加，每秒所能发送的消息数量逐渐减少（nMsg/sec）。但是，每秒发送的消息的总大小（MB/sec），会随着消息长度的增加而增加。

## 4. 消费者测试

消费者测试，可以从线程数、分区数、副本数等维度来进行测试。

### 4.1 线程数

创建一个拥有 6 个分区、1 个备份的 Topic，用不同的线程数读取相同的数据量，查看性能变化。测试脚本如下：



```
# 创建主题
```



```
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 --topic test_consumer_perf --partitions 6 --replication-
factor 1
```

# 设置 1 个线程数

```
[hadoop@dn1 ~]$ kafka-consumer-perf-test.sh - zookeeper
dn1:2181, dn2:2181, dn3:2181 --messages 5000000 --topic
test_consumer_perf
--group g1 --threads 1
```

# 设置 3 个线程数

```
[hadoop@dn1 ~]$ kafka-consumer-perf-test.sh - zookeeper
dn1:2181, dn2:2181, dn3:2181 --messages 5000000 --topic
test_consumer_perf
--group g2 --threads 3
```

# 设置 6 个线程数

```
[hadoop@dn1 ~]$ kafka-consumer-perf-test.sh - zookeeper
dn1:2181, dn2:2181, dn3:2181 --messages 5000000 --topic
test_consumer_perf
--group g3 --threads 6
```



### 4.1.1 测试结果

Kafka 节点数	线程数	分区数	副本数	每秒读取消息	消息大小 (MB/s)
3	1	6	1	934404.4104	89.1117
3	3	6	1	974229.8206	93.0951
3	6	6	1	1043623.043	99.5276

### 4.1.2 结论

随着线程数的增加，每秒读取的消息记录会逐渐增加。在线程数与消费主题的分区相等时，吞吐量达到最佳值。随后，再增加线程数，新增的线程数将会处于空闲状态，对提升消费者程序的吞吐量没有帮助。

## 4.2 分区数

新建一个 Topic，改变它的分区数，读取相同数量的消息记录，查看性能变化，执行脚本如下：



```
# 创建一个拥有 12 个分区的主题
```

```
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 --topic test_consumer_perf_p12 --partitions 12
--replication-factor 1
# 创建一个拥有 24 个分区的主题
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 --topic test_consumer_perf_p24 --partitions 24
--replication-factor 1

# 用一个线程读取数据到拥有 12 个分区的主题中
[hadoop@dn1 ~]$ kafka-consumer-perf-test.sh - zookeeper
dn1:2181, dn2:2181, dn3:2181 --messages 5000000 - topic
test_consumer_perf_p12 --group g2 --threads 1

# 用一个线程读取数据到拥有 12 个分区的主题中
[hadoop@dn1 ~]$ kafka-consumer-perf-test.sh - zookeeper
dn1:2181, dn2:2181, dn3:2181 --messages 5000000 - topic
test_consumer_perf_p24 --group g3 --threads 1
```

### 4.2.1 测试结果

Kafka 节点数	线程数	分区数	副本数	每秒读取消息	消息大小 (MB/s)
3	1	6	1	934404.4104	89.1117
3	1	12	1	905354.5725	85.7119
3	1	24	1	887605.3472	82.1391

### 4.2.2 结论

当分区数增加时，如果线程数保持不变，则消费者程序的吞吐量性能会下降。

## 4.3 副本数

新建 Topic，改变 Topic 的副本数，读取相同数量的消息记录，查看性能变化，执行脚本如下：

```
# 创建一个有用两个副本、6 个分区的主题
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 - topic test_consumer_perf_r2 --partitions 6
--replication-factor 2
```

```
# 创建一个有 3 个副本、6 个分区的主题
[hadoop@dn1 ~]$ kafka-topics.sh --create --zookeeper dn1:2181,
dn2:2181,
dn3:2181 -topic test_consumer_perf_r3 --partitions 6
--replication-factor 3

# 用 3 个线程读取数据到拥有两个副本的主题中
[hadoop@dn1 ~]$ kafka-consumer-perf-test.sh - zookeeper dn1:2181
, dn2:2181, dn3:2181 --messages 5000000 - topic
test_consumer_perf_r2 --group g2 --threads 3

# 用 3 个线程读取数据到拥有 3 个副本的主题中
[hadoop@dn1 ~]$ kafka-consumer-perf-test.sh --zookeeper dn1:2181
, dn2:2181, dn3:2181 --messages 5000000 - topic
test_consumer_perf_r3 --group g3 --threads 3
```

### 4.3.1 测试结果

Kafka 节点数	线程数	分区数	副本数	每秒读取消息	消息大小 (MB/s)
3	3	6	1	963390.7514	91.8761
3	3	6	2	963481.6523	91.9761
3	3	6	3	963241.8404	91.7727

### 4.3.2 结论

副本数对消费者程序的吞吐量影响较小，消费者程序是从 Topic 的每个分区的 Leader 上读取数据的，而与副本数无关。

## 5. 总结

Kafka 性能测试步骤并不复杂，大家可以根据实际的测试环境、数据量，通过对生产者和消费者不同维度的测试，来获取一组最佳的调优参数值。