

RocketMQ 性能测试报告

目录

RocketMQ 性能测试报告 1

1 概述..... 3

 1.1 编写目的..... 3

 1.2 适用范围..... 3

 1.3 术语表..... 3

 1.4 参考资料..... 4

2 性能测试目的 4

3 测试环境..... 4

 3.1 硬件环境（物理架构） 4

 3.2 软件环境（逻辑架构） 5

4 软件整体性能评价..... 5

 4.1 预期指标..... 5

 4.2 实际表现..... 5

5 测试过程与具体结论 5

1 概述

1.1 编写目的

测试 RocketMQ 的写数据、读数据性能及稳定性。

1.2 适用范围

对 RocketMQ 感兴趣的读者。

1.3 术语表

nameserver:

专为 RocketMQ 设计的轻量级名称服务。

producer:

消息生产者，负责生产消息，一般由业务系统负责生成消息。

consumer:

消息消费者，负责消费消息，一般是后台系统负责异步消费。

broker:

消息中转角色，负责存储消息，转发消息。

master:

broker 中的主节点。

slave:

broker 中的副节点。

异步复制:

消息先写入 master 节点,再由 master 节点异步复制到 slave 节点,类似 mysql 中的 master-slave 机制。

同步双写:

消息同时写入 master 节点和 slave 节点。

异步刷盘:

broker 的一种持久化策略，消息写入 pagecache 后，直接返回。由异步线程负责将 pagecache 写入硬盘。

同步刷盘:

Broker 的一种持久化策略，消息写入 pagecache 后，由同步线程将 pagecache 写入硬盘后，再返回。

TPS:

每秒钟发送的消息个数。

1.4 参考资料

官方网站上的：

《RocketMQ 原理简介》

《RocketMQ 用户指南》

《RocketMQ 最佳实践》

2 性能测试目的

测试 RocketMQ 的写入、读取性能及稳定性。

3 测试环境

3.1 硬件环境（物理架构）

8 台服务器（编号 1~8）

nameserver 部署在 1 上；

1~2 用于部署 producer；

3~6 用于部署 broker；

7~8 用于部署 consumer。

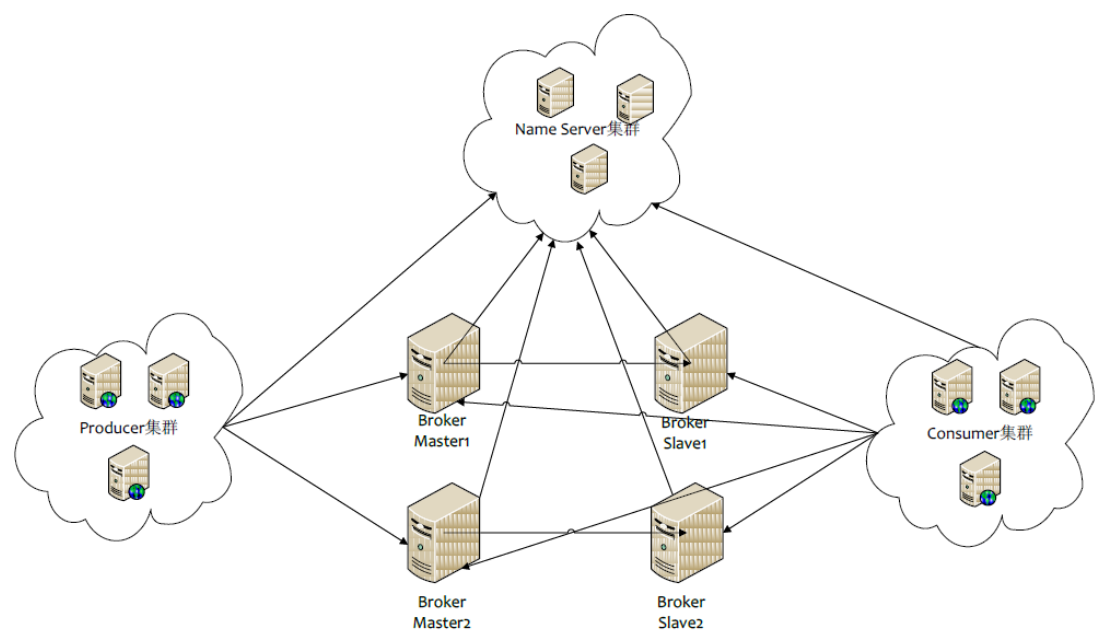
每台服务器均为以下硬件配置

CPU：E5-2420、8 核*2、1.9G 主频

内存：32G

硬盘：1T、RAID10

3.2 软件环境（逻辑架构）



每台服务器均为以下软件配置
操作系统：centos6（内核 linux 2.6.32-358.el6.x86_64）
JDK：1.6（Java HotSpot 64-Bit Server VM）

4 软件整体性能评价

4.1 预期指标

预期 RocketMQ 可以满足日常业务性能需求。

4.2 实际表现

经过测试，RocketMQ 稳定性高，未出现无法写入或读取的情况，未出现内存占用过高的情况。2 台 master、2 台 slave 组成的 broker 集群，消息大小为 2KB 时，写入速度可达每秒 60 多 MB，日可处理数据量约 7.2 亿条，性能可以满足需求。

5 测试过程与具体结论

1) 测试场景一
1 台生产者、1 台 broker、1 台消费者，broker 采用异步刷盘。消费者为 1 个线程，push 消费模式。测试线程数和消息大小对 TPS 和吞吐量的影响。
测试结果 1:

生产者线程数(个)	消息大小（Byte）	TPS（个）	吞吐量（MB）
-----------	------------	--------	---------

1	2048	1185	2.3
10	2048	6783	13.2
32	2048	12462	24.3
64	2048	14806	28.9
128	2048	13464	26.3

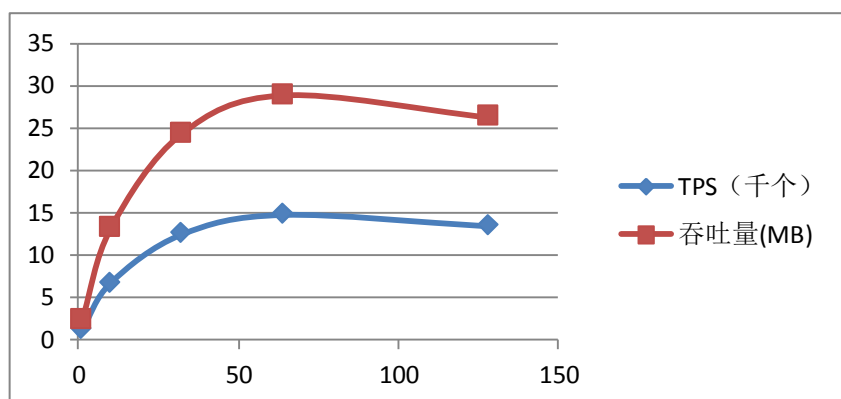
测试结果 2:

生产者线程数(个)	消息大小 (Byte)	TPS (个)	吞吐量 (MB)
32	1024	9848	9.6
32	2048	12462	24.3
32	4096	12545	49.0
32	8192	9215	72.0

以上测试情况消费者均能及时消费（延迟很小）。

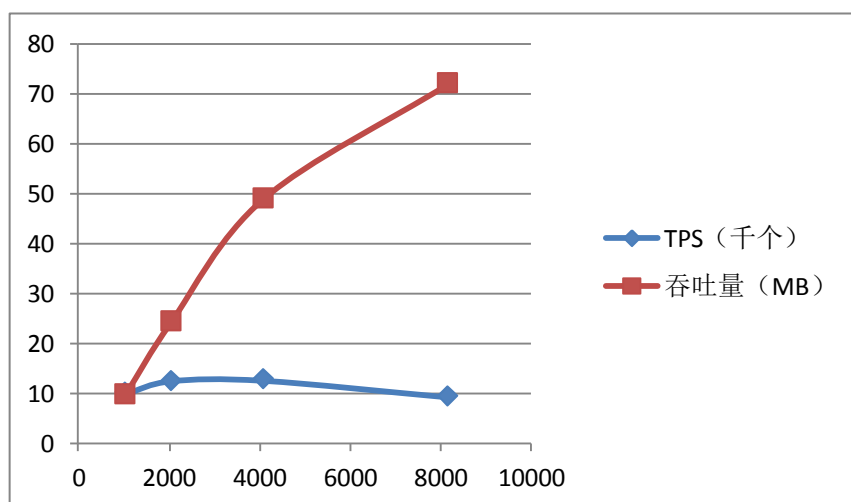
结论 1:

消息大小固定（2KB）时，随着生产者线程数的增加，TPS 和吞吐量均会上升，但当线程增加到一定数量后，TPS 和吞吐量会有所下降。如下图：



结论 2:

生产者线程数（32 个）固定时，随着消息大小的增加，TPS 会先升后降，吞吐量会逐步提升。如下图：



2) 测试场景二

2 台生产者，4 台 broker（2 台 master、2 台 slave 构成 2 个 broker 集群），2 台消费者。每台生产者开启 10 个线程，消息大小均为 2KB。每台消费者为 1 个线程，push 消费模

式。测试 broker 主从模式和写盘模式对 TPS 和吞吐量的影响。

1.异步复制、异步刷盘的测试结果：

	线程数(个)	消息大小 (Byte)	TPS (个)	吞吐量 (MB)
第 1 台生产者	10	2048	6457	12.6
第 2 台生产者	10	2048	6294	12.3
合计	20	2048	12751	24.9

以上测试情况消费者均能及时消费（延迟很小）。

2.异步复制、同步刷盘的测试结果：

	线程数(个)	消息大小 (Byte)	TPS (个)	吞吐量 (MB)
第 1 台生产者	10	2048	1835	3.6
第 2 台生产者	10	2048	1865	3.6
合计	20	2048	3700	7.2

以上测试情况消费者均能及时消费（延迟很小）。

3.同步双写、异步刷盘的测试结果：

	线程数(个)	消息大小 (Byte)	TPS (个)	吞吐量 (MB)
第 1 台生产者	10	2048	6268	12.2
第 2 台生产者	10	2048	6217	12.1
合计	20	2048	12485	24.3

以上测试情况消费者均能及时消费（延迟很小）。

4.同步双写、同步刷盘的测试结果：

	线程数(个)	消息大小 (Byte)	TPS (个)	吞吐量 (MB)
第 1 台生产者	10	2048	1898	3.7
第 2 台生产者	10	2048	1875	3.7
合计	20	2048	3773	7.4

以上测试情况消费者均能及时消费（延迟很小）。

结论：

同步双写模式比异步复制模式性能稍低，但不明显。

同步刷盘模式比异步刷盘模式性能低很多。

3) 测试场景三

2 台生产者，4 台 broker（2 台 master、2 台 slave 构成 2 个 broker 集群），2 台消费者。每台生产者开启 10 个线程，消息大小均为 2KB。每台消费者为 1 个线程，push 消费模式。Broker 采用异步复制、异步刷盘模式。连续运行 8 小时，测试稳定性。

测试结果：

所有消息均成功发送。具体结果可查看“rocketmq 测试结果记录/2p-2m(异步复制、异步刷盘)-2s-2c-8 小时稳定性测试”。

4) 测试场景四

2 台生产者，4 台 broker（2 台 master、2 台 slave 构成 2 个 broker 集群），2 台消费者。每台消费者为 1 个线程，push 消费模式。Broker 采用异步复制、异步刷盘模式。测试当前软硬件配置下 broker 集群的极限写入速度。

2 台生产者合计线程数(个)	消息大小 (Byte)	TPS (个)	吞吐量 (MB)
128	4096	9368	36.6

256	8192	8600	67.2
400	8192	8343	65.2

以上测试情况消费者均能及时消费（延迟很小）。

结论：

当前软硬件配置下，**broker** 集群可以达到六十多兆的写入速度（如果继续增加消息大小，还可以提高写入速度）。