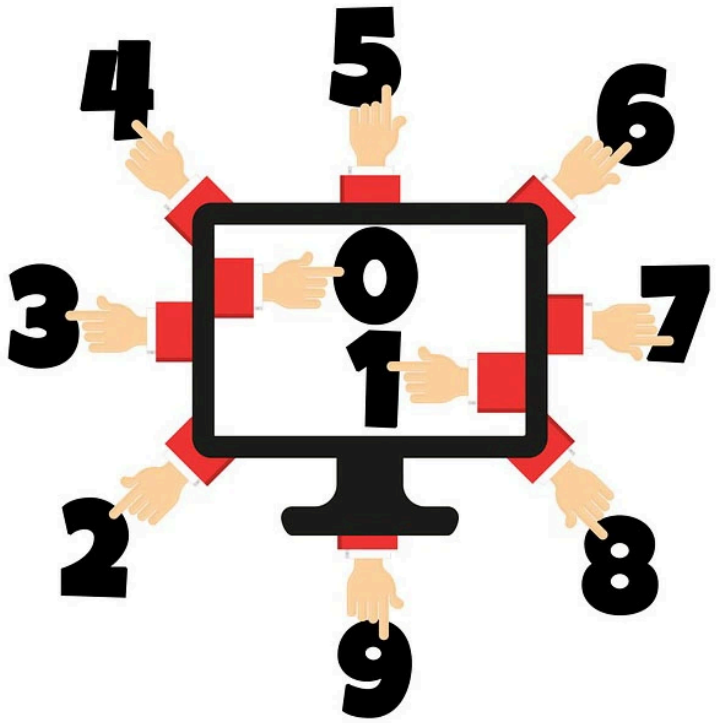
06 | Kafka线上集群部署方案怎么做?

胡夕 · Kafka核心技术与实战



专栏前面几期内容，我分别从 Kafka 的定位、版本的变迁以及功能的演进等几个方面循序渐进地梳理了 Apache Kafka 的发展脉络。通过这些内容，我希望你能清晰地了解 Kafka 是用来做什么的，以及在实际生产环境中该如何选择 Kafka 版本，更快地帮助你入门 Kafka。

现在我们就来看看在生产环境中的 Kafka 集群方案该怎么做。既然是集群，那必然就要有多个 Kafka 节点机器，因为只有单台机器构成的 Kafka 伪集群只能用于日常测试之用，根本无法满足实际的线上生产需求。而真正的线上环境需要仔细地考量各种因素，结合自身的业务需求而制定。下面我就分别从操作系统、磁盘、磁盘容量和带宽等方面来讨论一下。

操作系统

首先我们先看看要把 Kafka 安装到什么操作系统上。说起操作系统，可能你会问 Kafka 不是 JVM 系的大数据框架吗？Java 又是跨平台的语言，把 Kafka 安装到不同的操作系统上会有什么区别吗？其实区别相当大！

的确，如你所知，Kafka 由 Scala 语言和 Java 语言编写而成，编译之后的源代码就是普通的“.class”文件。本来部署到哪个操作系统应该都是一样的，但是不同操作系统的差异还是给 Kafka 集群带来了相当大的影响。目前常见的操作系统有 3 种：Linux、Windows 和 macOS。应该说部署在 Linux 上的生产环境是最多的，也有一些 Kafka 集群部署在 Windows 服务器上。Mac 虽然也有 macOS Server，但是我怀疑是否有人（特别是国内用户）真的把生产环境部署在 Mac 服务器上。

如果考虑操作系统与 Kafka 的适配性，Linux 系统显然要比其他两个特别是 Windows 系统更加适合部署 Kafka。虽然这个结论可能你不感到意外，但其中具体的原因你也一定要了解。主要是在下面这三个方面上，Linux 的表现更胜一筹。

I/O 模型的使用

数据网络传输效率

社区支持度

我分别来解释一下，首先来看 I/O 模型。什么是 I/O 模型呢？你可以近似地认为 I/O 模型就是操作系统执行 I/O 指令的方法。

主流的 I/O 模型通常有 5 种类型：阻塞式 I/O、非阻塞式 I/O、I/O 多路复用、信号驱动 I/O 和异步 I/O。每种 I/O 模型都有各自典型的使用场景，比如 Java 中 Socket 对象的阻塞模式和非阻塞模式就对应于前两种模型；而 Linux 中的系统调用 select 函数就属于 I/O 多路复用模型；大名鼎鼎的 epoll 系统调用则介于第三种和第四种模型之间；至于第五种模型，其实很少有 Linux 系统支持，反而是 Windows 系统提供了一个叫 IOCP 线程模型属于这一种。

你不必详细了解每一种模型的实现细节，通常情况下我们认为后一种模型会比前一种模型要高级，比如 epoll 就比 select 要好，了解到这一程度应该足以应付我们下面的内容了。

说了这么多，I/O 模型与 Kafka 的关系又是什么呢？实际上 Kafka 客户端底层使用了 Java 的 selector，selector 在 Linux 上的实现机制是 epoll，而在 Windows 平台上的实现机制是 select。**因此在这一点上将 Kafka 部署在 Linux 上是有优势的，因为能够获得更高效的 I/O 性能。**

其次是网络传输效率的差别。你知道的，Kafka 生产和消费的消息都是通过网络传输的，而消息保存在哪里呢？肯定是磁盘。故 Kafka 需要在磁盘和网络间进行大量数据传输。如果你熟悉 Linux，你肯定听过零拷贝（Zero Copy）技术，就是当数据在磁盘和网络进行传输时避免昂贵的内核态数据拷贝从而实现快速的数据传输。Linux 平台实现了这样的零拷贝机制，但有些令人遗憾的是在 Windows 平台上必须要等到 Java 8 的 60 更新版本才能“享受”到这个福利。**一句话总结一下，在 Linux 部署 Kafka 能够享受到零拷贝技术所带来的快速数据传输特性。**

最后是社区的支持度。这一点虽然不是什么明显的差别，但如果不了解的话可能比前两个因素对你的影响更大。简单来说就是，社区目前对 Windows 平台上发现的 Kafka Bug 不做任何承诺。虽然口头上依然保证尽力去解决，但根据我的经验，Windows 上的 Bug 一般是不会修复的。**因此，Windows 平台上部署 Kafka 只适合于个人测试或用于功能验证，千万不要应用于生产环境。**

磁盘

如果问哪种资源对 Kafka 性能最重要，磁盘无疑是要排名靠前的。在对 Kafka 集群进行磁盘规划时经常面对的问题是，我应该选择普通的机械磁盘还是固态硬盘？前者成本低且容量大，但易损坏；后者性能优势大，不过单价高。我给出的建议是使用普通机械硬盘即可。

Kafka 大量使用磁盘不假，可它使用的方式多是顺序读写操作，一定程度上规避了机械磁盘最大的劣势，即随机读写操作慢。从这一点上来说，使用 SSD 似乎并没有太大的性能优势，毕竟从性价比上来说，机械磁盘物美价廉，而它因易损坏而造成的可靠性差等缺陷，又由 Kafka 在软件层面提供机制来保证，故使用普通机械磁盘是很划算的。

关于磁盘选择另一个经常讨论的话题就是到底是否应该使用磁盘阵列（RAID）。使用 RAID 的两个主要优势在于：

- 提供冗余的磁盘存储空间
- 提供负载均衡

以上两个优势对于任何一个分布式系统都很有吸引力。不过就 Kafka 而言，一方面 Kafka 自己实现了冗余机制来提供高可靠性；另一方面通过分区的概念，Kafka 也能在软件层面自行实现负载均衡。如此说来 RAID 的优势就没有那么明显了。当然，我并不是说 RAID 不好，实际上依然有很多大厂确实是把 Kafka 底层的存储交由 RAID 的，只是目前 Kafka 在存储这方面提供了越来越便捷的高可靠性方案，因此在线上环境使用 RAID 似乎变得不是那么重要了。综合以上的考量，我给出的建议是：

追求性价比的公司可以不搭建 RAID，使用普通磁盘组成存储空间即可。

使用机械磁盘完全能够胜任 Kafka 线上环境。

磁盘容量

Kafka 集群到底需要多大的存储空间？这是一个非常经典的规划问题。Kafka 需要将消息保存在底层的磁盘上，这些消息默认会被保存一段时间然后自动被删除。虽然这段时间是可以配置的，但你应该如何结合自身业务场景和存储需求来规划 Kafka 集群的存储容量呢？

我举一个简单的例子来说明该如何思考这个问题。假设你所在公司有个业务每天需要向 Kafka 集群发送 1 亿条消息，每条消息保存两份以防止数据丢失，另外消息默认保存两周时间。现在假设消息的平均大小是 1KB，那么你能说出你的 Kafka 集群需要为这个业务预留多少磁盘空间吗？

我们来计算一下：每天 1 亿条 1KB 大小的消息，保存两份且留存两周的时间，那么总的空间大小就等于 1 亿 * 1KB * 2 / 1000 / 1000 = 200GB。一般情况下 Kafka 集群除了消息数据还有其他类型的数据，比如索引数据等，故我们再为这些数据预留出 10% 的磁盘空间，因此总的存储容量就是 220GB。既然要保存两周，那么整体容量即为 220GB * 14，大约 3TB 左右。Kafka 支持数据的压缩，假设压缩比是 0.75，那么最后你需要规划的存储空间就是 0.75 * 3 = 2.25TB。

总之在规划磁盘容量时你需要考虑下面这几个元素：

新增消息数

消息留存时间

平均消息大小

备份数

是否启用压缩

带宽

对于 Kafka 这种通过网络大量进行数据传输的框架而言，带宽特别容易成为瓶颈。事实上，在我接触的真实案例当中，带宽资源不足导致 Kafka 出现性能问题的比例至少占 60% 以上。如果你的环境中还涉及跨机房传输，那么情况可能就更糟了。

如果你不是超级土豪的话，我会认为你和我平时使用的都是普通的以太网络，带宽也主要有两种：1Gbps 的千兆网络和 10Gbps 的万兆网络，特别是千兆网络应该是一般公司网络的标准配置了。下面我就以千兆网络举一个实际的例子，来说明一下如何进行带宽资源的规划。

与其说是带宽资源的规划，其实真正要规划的是所需的 Kafka 服务器的数量。假设你公司的机房环境是千兆网络，即 1Gbps，现在你有个业务，其业务目标或 SLA 是在 1 小时内处理 1TB 的业务数据。那么问题来了，你到底需要多少台 Kafka 服务器来完成这个业务呢?

让我们来计算一下，由于带宽是 1Gbps，即每秒处理 1Gb 的数据，假设每台 Kafka 服务器都是安装在专属的机器上，也就是说每台 Kafka 机器上没有混部其他服务，毕竟真实环境中不建议这么做。通常情况下你只能假设 Kafka 会用到 70% 的带宽资源，因为总要为其他应用或进程留一些资源。

根据实际使用经验，超过 70% 的阈值就有网络丢包的可能性了，故 70% 的设定是一个比较合理的值，也就是说单台 Kafka 服务器最多也就能使用大约 700Mb 的带宽资源。

稍等，这只是它能使用的最大带宽资源，你不能让 Kafka 服务器常规性使用这么多资源，故通常要再额外预留出 2/3 的资源，即单台服务器使用带宽 700Mb / 3 ≈ 240Mbps。需要提示的是，这里的 2/3 其实是相当保守的，你可以结合你自己机器的使用情况酌情减少此值。

好了，有了 240Mbps，我们就可以计算 1 小时内处理 1TB 数据所需的服务器数量了。根据这个目标，我们每秒需要处理 2336Mb 的数据，除以 240，约等于 10 台服务器。如果消息还

需要额外复制两份，那么总的服务器台数还要乘以 3，即 30 台。

怎么样，还是很简单的吧。用这种方法评估线上环境的服务器台数是比较合理的，而且这个方法能够随着你业务需求的变化而动态调整。

小结

所谓“兵马未动，粮草先行”。与其盲目上马一套 Kafka 环境然后事后费力调整，不如在一开始就思考好实际场景下业务所需的集群环境。在考量部署方案时需要通盘考虑，不能仅从单个维度上进行评估。相信今天我们聊完之后，你对如何规划 Kafka 生产环境一定有了一个清晰的认识。现在我来总结一下今天的重点：

因素	考量点	建议
操作系统	操作系统I/O模型	将Kafka部署在Linux系统上
磁盘	磁盘I/O性能	普通环境使用机械磁盘，不需要搭建RAID
磁盘容量	根据消息数、留存时间预估磁盘容量	实际使用中建议预留20% ~ 30%的磁盘空间
带宽	根据实际带宽资源和业务SLA预估服务器数量	对于千兆网络，建议每台服务器按照700Mbps来计算，避免大流量下的丢包

开放讨论

对于今天我所讲的这套评估方法，你有什么问题吗？你还能想出什么改进的方法吗？

欢迎你写下自己的思考或疑问，我们一起讨论 。如果你觉得有所收获，也欢迎把文章分享给你的朋友。

AI智能总结

本文深入探讨了在生产环境中部署Kafka集群的方案选择和规划。首先，针对操作系统的选择，Linux系统被认为更适合部署Kafka，因其在I/O模型、网络传输效率和社区支持度方面具有优势。其次，对于磁盘的选择，建议使用普通机械硬盘，因为Kafka大量使用的是顺序读写操作，规避了机械磁盘的劣势。此外，对于磁盘容量的规划，文章提供了详细的计算方法和考量因素，帮助读者根据业务需求合理规划Kafka集群的存储空间。另外，文章还深入讨论了带宽资源的规划，指出带宽特别容易成为Kafka性能问题的瓶颈，并提供了实际例子和

计算方法，帮助读者评估线上环境的服务器台数。总的来说，本文通过深入浅出的方式，为读者提供了在生产环境中部署Kafka集群的实用指南，涵盖了技术细节和实际操作，对于需要了解Kafka集群部署方案的读者具有很高的参考价值。

全部留言 (110)

最新　精选

mickle 置顶
2019-06-15

1000*1000/(60*60)=277,这个2336MB是怎么换算来的，还有什么要考虑的吗?

作者回复: 277MB，乘以8，大致等于2300+Mb（小b）。带宽资源一般用Mbps而不是MBps衡量

共 10 条评论>　106

A_NATE_👻
2019-06-15

我们曾经也认为用普通硬盘就行，换成普通硬盘导致生产者堵塞写入负载偏高，换成SSD就没事了，我们每天消息数大概50亿。

作者回复: 嗯嗯，专栏里面只是给出一个评估的方法。具体还要结合自己的实际情况来调整。通常我们认为SSD的顺序写TPS大约是HDD的4倍。除了纵向扩展使用SSD之外，也可以尝试一下横向扩展，增加更多的broker或HDD分散负载：）

共 8 条评论>　67

蒙开强
2019-06-17

老师，你好，你讲的这几个纬度很好，之前我们搭建一套kafka集群就不知道怎么去衡量，我再问一个相关问题，我个人觉得kafka会出现丢数据情况，比如某个分区的leader挂了，在切换选举到另外副本为leader时，这个副本还没同步之前的leader数据，这样数据就丢了

作者回复: 嗯嗯，对于producer而言，如果在乎数据持久性，那么应该设置acks=all，这样当出现你说的这个情况时，producer会被显式通知消息发送失败，从而可以重试。

共 4 条评论>

51

WL

2019-06-15

有三个问题请教一下老师:
1. 上文提到对于千兆网卡kafka服务器最多使用700M的带宽资源, 这700M的资源是单机使用的还是集群共用的, 为什么不能作为常规使用呢?
2. 文章举例是1小时1T的数据处理目标, 那一秒中是不是1024/3600 = 0.284G = 285M, 请问下文章中的2336M是咋算出来的.
3. 文章中的例子kafka单机要达到240M的读写能力, CPU应该配几核的?

作者回复: 1. 这个700Mb只是经验值罢了。另外预留buffer的意思是即使你最好不要让broker常规占用700Mb的资源。一旦碰到峰值流量，很容易将带宽打满。故做了一些资源预留
2. 285M是大B，即字节啊，乘以8之后就是2336Mb。带宽资源一般用Mbps而非MBps衡量
3. 我没有谈及CPU，是因为通常情况下Kafka不太占用CPU，因此没有这方面的最佳实践出来。但有些情况下Kafka broker是很耗CPU的：1. server和client使用了不同的压缩算法；2. server和client版本不一致造成消息格式转换；3. broker端解压缩校验

其中前两个都能规避，第三个目前无法规避。不过相比带宽资源，CPU通常都不是瓶颈

33

李跃爱学习

2019-07-16

老师希望解答一下，之前也说明了Kafka 机器上没有混布其他服务，为什么常规需要预留2/3，只能跑240Mbps，

作者回复: 为follower拉取留一些带宽

共 16 条评论>

30

墙角儿的花

2019-06-15

弱弱的问一句老师，“根据这个目标，我们每秒需要处理 2336Mb 的数据，除以 240，约等于 10 台服务器”，机房入口带宽1Gbps,怎么能做到1秒处理2336Mb的数据的

作者回复: 这里是指单机带宽，机房总带宽不可能这么小的。。。

共 4 条评论>

20

Geek_Sue
2019-06-17

胡老师，您好，我想请问下，我们公司的环境是基于Docker这种微服务架构，那么kafka部署在Docker容器中部署方案是否会有一些不同呢?

作者回复: 目前社区对Docker方案支持的并不是太好，主要都是一些第三方公司还有Confluent公司在提供解决方案。在Docker上部署我个人觉得没有太大的不同，只是注意带宽资源吧，因为常见的做法都是买一台性能超强的服务器然后在上面启动多个Docker容器，虽然CPU、RAM、磁盘都能承受，但单机还是受限于带宽的。

共 2 条评论>

18

疯琴
2019-06-17

老师，partitons的数量和硬盘的数量有匹配关系么？一块盘一个partiton比一块盘多个partiton要快么？是线性的关系么?

作者回复: 没有具体的关系。

“一块盘一个partiton比一块盘多个partiton要快么？”没有实验数据支撑，单纯从分析角度来看我是认同的。当某块磁盘上有太多的分区时引入了很多随机IO拖慢了性能。事实上，阿里的RocketMQ一直宣称当单磁盘超过256分区时Kafka性能不如RocketMQ，原因也在于此。
数据来源：http://jm.taobao.org/2016/04/07/kafka-vs-rocketmq-topic-amout/

共 3 条评论>

16

南辕北辙
2019-06-17

这个假设是：follower与leader处于不同的broker而实际环境中不推荐单机多broker的架构
摘自老师回复其他同学。
老师这个的意思是不是生产上的架构通常一台服务器上只会有leader或者follow的分区，而不会二者存在一台服务器上，所以根据带宽计算服务器数量时，根据备份数为2，所以就直接✖3了。

作者回复: Leader副本和Follower副本必然在不同的Broker上，而生产环境一般也不推荐将多台Brok

er混布到同一台服务器上。当然服务器性能强劲的话也未尝不可：）

11

Royal

2019-06-15

您好，我想请教下kafka metric相关的知识，比如kafka produce速率获取等

作者回复: kafka producer速率可以监控这个JMX指标：

kafka.producer:type=[consumer|producer|connect]-node-metrics,client-id=([-.\w]+),node-id=([0-9]+)

共 2 条评论>

11