Apache Pulsar：雅虎开发的企业级发布订阅消息系统

Apache Pulsar（孵化器项目）是一个企业级的发布订阅（pub-sub）消息系统，最初由Yahoo开发，并于2016年底开源，现在是Apache软件基金会的一个孵化器项目。Pulsar在Yahoo的生产环境运行了三年多，助力Yahoo的主要应用，如Yahoo Mail、Yahoo Finance、Yahoo Sports、Flickr、Gemini广告平台和Yahoo分布式键值存储系统Sherpa。

Apache Pulsar is an open-source distributed pub-sub messaging system originally created at Yahoo and now part of the Apache Software Foundation.

如果想及时了
解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

Pulsar相关概念和术语

向Pulsar发送数据的应用程序叫作生产者（producer），而从Pulsar读取数据的应用程序叫作消费者（consumer）。有时候消费者也被叫作订阅者。主题（topic）是Pulsar的核心资源，一个主题可以被看成是一个通道，消费者向这个通道发送数据，消费者从这个通道拉取数据。

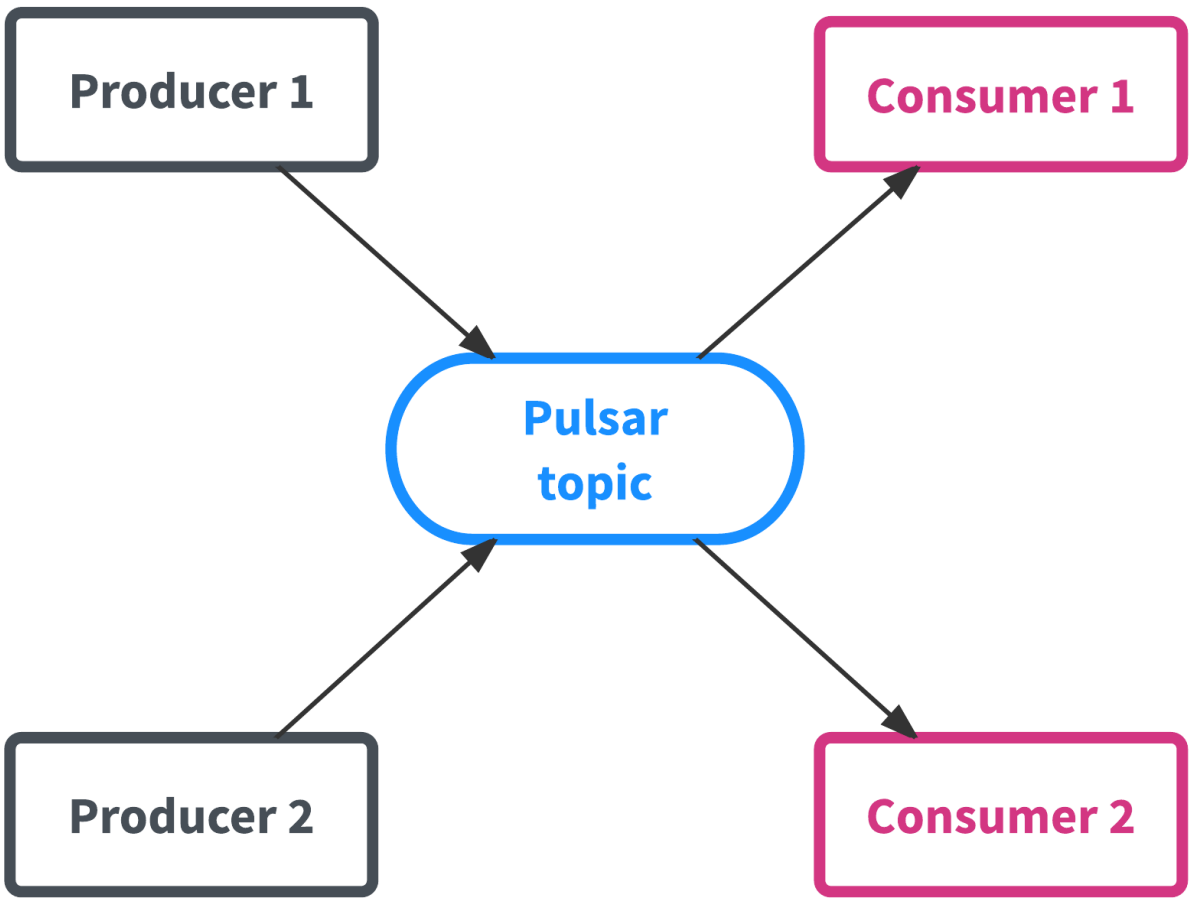

图1：生产者、消费者和主题

构建Pulsar的目的是为了支持多租户（multi-tenant）应用场景。Pulsar的多租户机制包含了两种资源：资产（property）和命名空间（namespace）。资产代表系统里的租户。假设有一个Pulsar集群用于支持多个应用程序（就像Yahoo那样），集群里的每个资产可以代表一个组织的团队、一个核心的功能或一个产品线。一个资产可以包含多个命名空间，一个命名空间可以包含任意个主题。

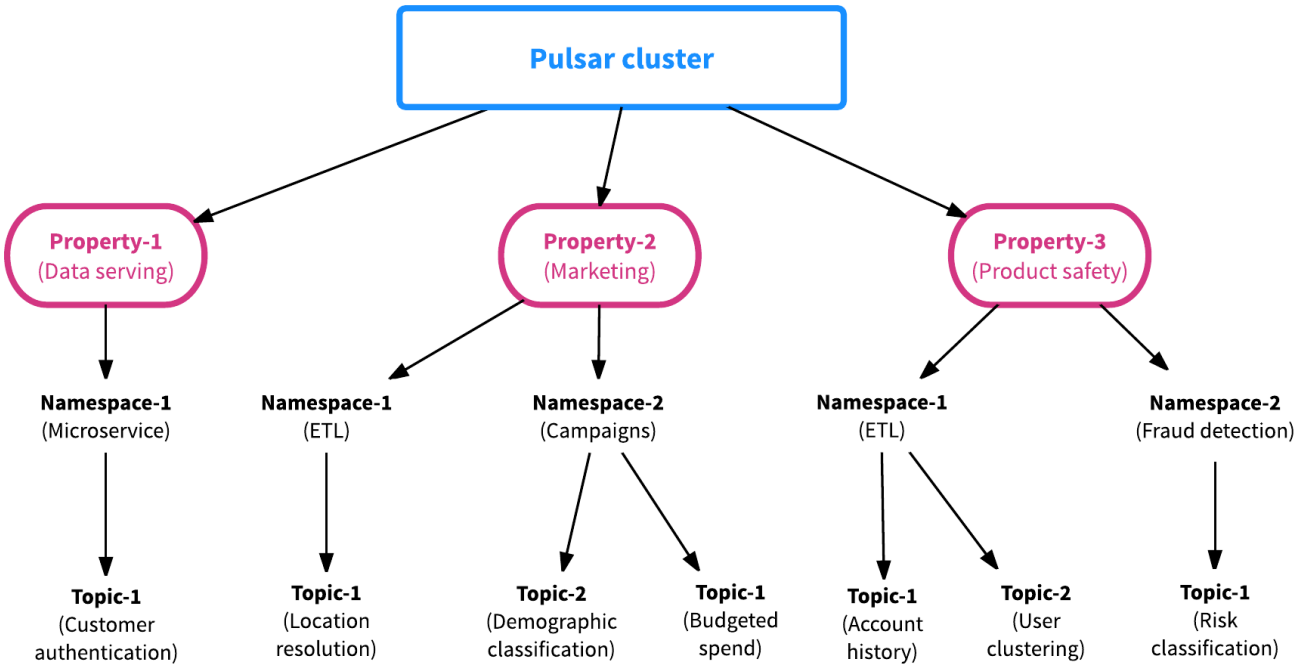

图2：Pulsar各个组件间的关系

命名空间是Pulsar最基本的管理单元。在命名空间层面，我们可以设置权限、调整复制选项、管理跨集群的数据复制、控制消息的过期时间或执行其他关键任务。命名空间里的主题会继承命名空间的配置，所以我们可以一次性对同一个命名空间内的所有主题进行配置。命名空间可以分为两种：

- 本地（local）——本地命名空间只在集群内可见。
- 全局（global）——命名空间对多个集群可见，可以是同一个数据中心内的集群，也可以是跨地域数据中心的集群。该功能取决于是否启用了集群复制功能。

虽然本地命名空间和全局命名空间的作用域不同，但它们都可以在不同的团队或不同的组织内共享。如果应用程序获得了命名空间的写入权限，就可以往该命名空间内的所有主题写入数据。如果写入的主题不存在，就会创建该主题。

每个命名空间可以包含一到多个主题，每个主题可以有多个订阅者，每个订阅者可以接收所有发布到该主题的消息。为了给应用程序提供更大的灵活性，Pulsar提供了三种订阅类型，它们可以共存在同一个主题上：

- 独享（exclusive）订阅——同时只能有一个消费者。
- 共享（shared）订阅——可以由多个消费者订阅，每个消费者接收其中的一部分消息。
- 失效备援（failover）订阅
  ——允许多个消费者连接到同一个主题上，但只有一个消费者能够接收消息。只有在当前消费者发生失效时，其他消费者才开始接收消息。

图3展示了这三种类型的订阅。Pulsar的订阅机制解耦了消息的生产者和消费者，在不增加复杂性和开发工作量的情况下为应用程序提供了更大的弹性。

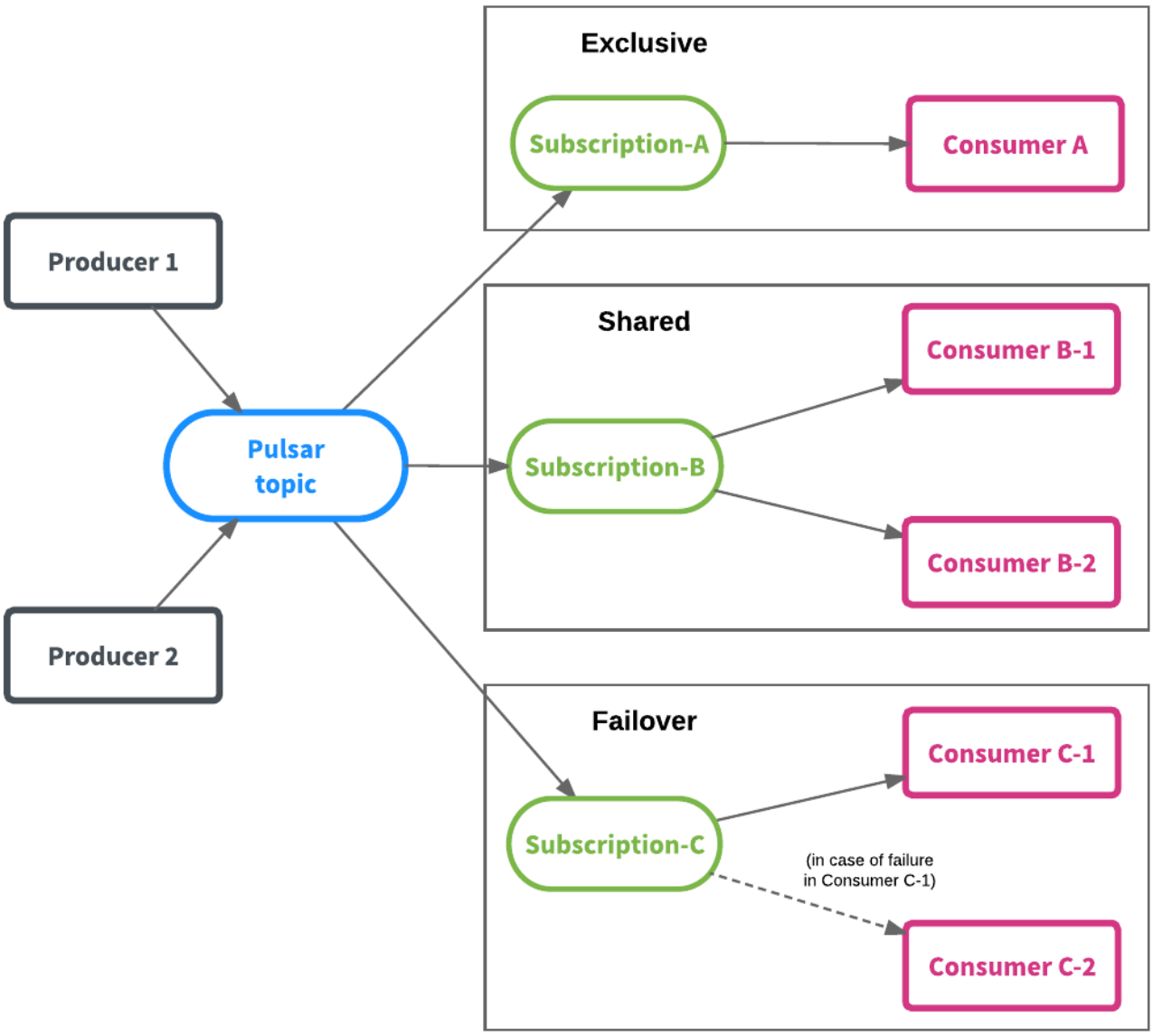

图3：不同类型的Pulsar订阅

数据分区

写入主题的数据可能只有几个MB，也有可能是几个TB。所以，在某些情况下主题的吞吐量很低，有时候又很高，完全取决于消费者的数量。那么碰到有些主题吞吐量很高而有些又很低的情况该怎么处理？为了解决这个问题，Pulsar将一个主题的数据分布到多台机器上，也就是所谓的分区。

在处理海量数据时，为了保证高吞吐量，分区是一种很常见的手段。默认情况下，Pulsar的主题是不进行分区的，但通过命令行工具或API可以很容易地创建分区主题，并指定分区的数量。

在创建好分区主题之后，Pulsar可以自动对数据进行分区，不会影响到生产者和消费者。也就是说，一个应用程序向一个主题写入数据，对主题分区之后，不需要修改应用程序的代码。分区只是一个运维操作，应用程序不需要关心分区是如何进行的。

主题的分区操作由一个叫作broker的进程来处理，Pulsar集群里的每个节点都会运行自己的broker。

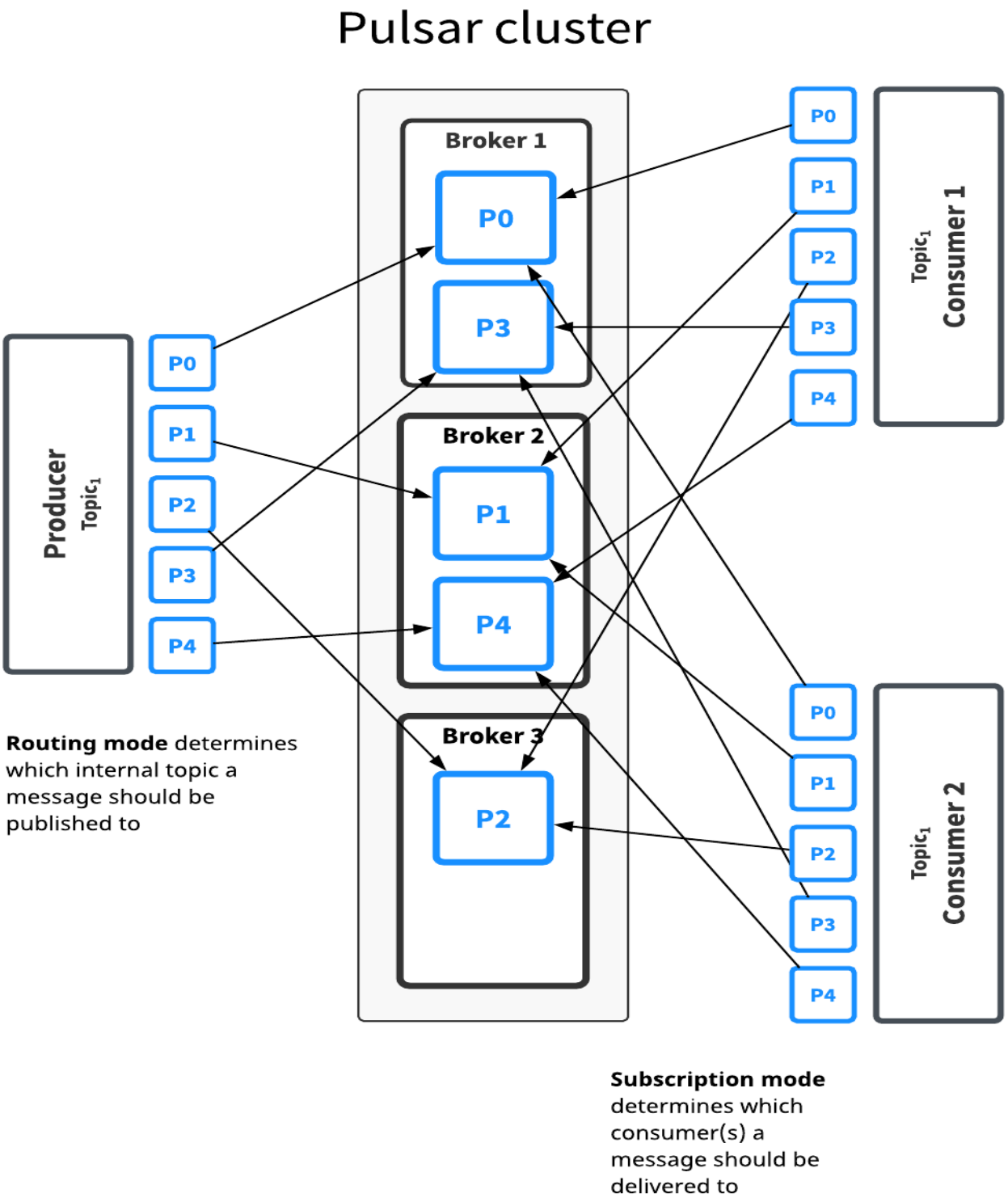

图4：将一个主题分到多个broker上

主题分区不会影响到应用程序，除此之外，Pulsar还提供了几种消息路由策略，帮助我们更好地跨分区、跨消费者分布数据。

- 单个分区
  ——生产者随机挑选一个分区，并将数据写入该分区。该策略与非分区主题提供的保证是一样的，不过如果有多个生产者向同一个主题写入数据，该策略就会很有用。
- 轮询（round
  robin）分区
  ——生产者通过轮询的方式将数据平均地分布到各个分区上。比如，第一个消息写入第一个分区，第二个消息写入第二个分区，并以此类推。
- 哈希（hash）分区
  ——每个消息会带上一个键，要写入哪个分区取决于它所带的键。这种分区方式可以保证次序。
- 自定义分区
  ——生产者使用自定义函数生成分区对应的数值，然后根据这个数值将消息写入对应的分区。

  持久性

Pulsar broker在收到消息并进行确认之后，就必须确保消息在任何情况下都不会丢失。与其他消息系统不同的是，Pulsar使用Apache BookKeeper来保证持久性。BookKeeper提供了低延迟的持久化存储。Pulsar在收到消息之后，将消息发送给多个BookKeeper节点（具体由复制系数来定），节点将数据写入预写式日志（write ahead log），同时在内存里也保存一份。节点在对消息进行确认之前，强制将日志写入到持久化的存储上，因此即使出现电力故障，数据也不会丢失。因为Pulsar broker将数据发给了多个节点，所以只会在大多数节点（quorum）确认写入成功之后它才会将确认消息发给生产者。Pulsar就是通过这种方式来保证即使在出现了硬件故障、网络故障或其他故障的情况下仍然能够保证数据不丢失。在后续的文章中，我们将深入探讨这方面的细节。

生产环境实践

Pulsar目前在助力Yahoo的主要应用，如Yahoo Mail、Yahoo Finance、Yahoo Sports、Gemini广告平台和Yahoo分布式键值存储系统Sherpa。很多场景都要求很强的持久性保证，比如零数据丢失，同时又要求很高的性能。Pulsar从2015年开始部署到生产环境，现在在Yahoo的生产环境里大规模地运行。

- Pulsar被部署在10多个数据中心里，具备了全网格复制能力
- 每天处理超过1000亿个消息
- 支持着140万个主题
- 整体的消息发布延迟小于5毫秒

总结

在这篇文章里，我们简单介绍了Apache Pulsar的一些概念，并解释了Pulsar是如何通过在发送确认消息前提交数据来保证持久性的，以及通过分区来提高吞吐量，等等。在后续的文章中，我们

将深入探讨Pulsar的整体架构和特性细节，我们也将提供一些指南教大家如何更好地使用Pulsar。

本文原文：发布订阅消息系统Apache Pulsar简介
英文原文：Introduction to the Apache Pulsar pub-sub messaging platform