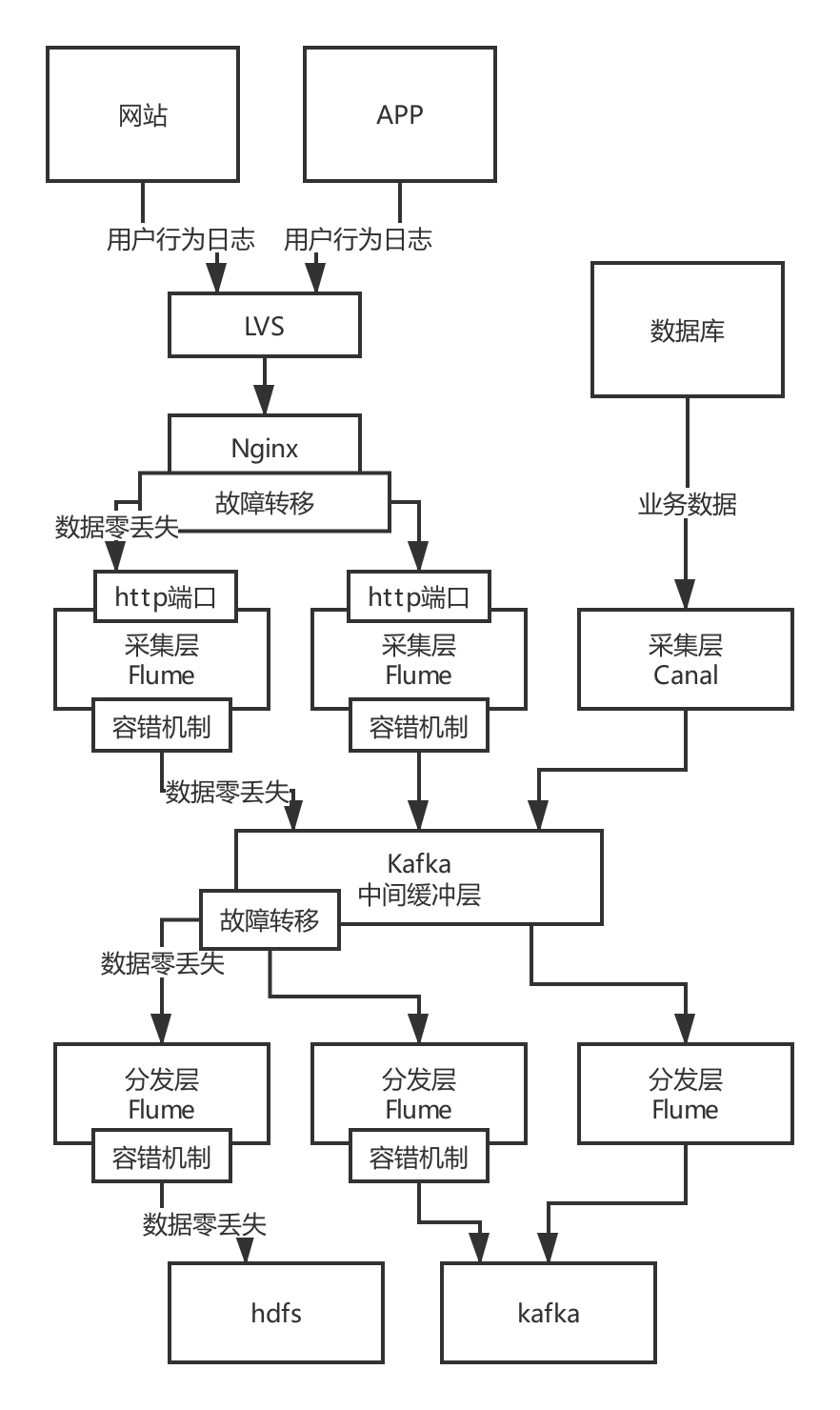
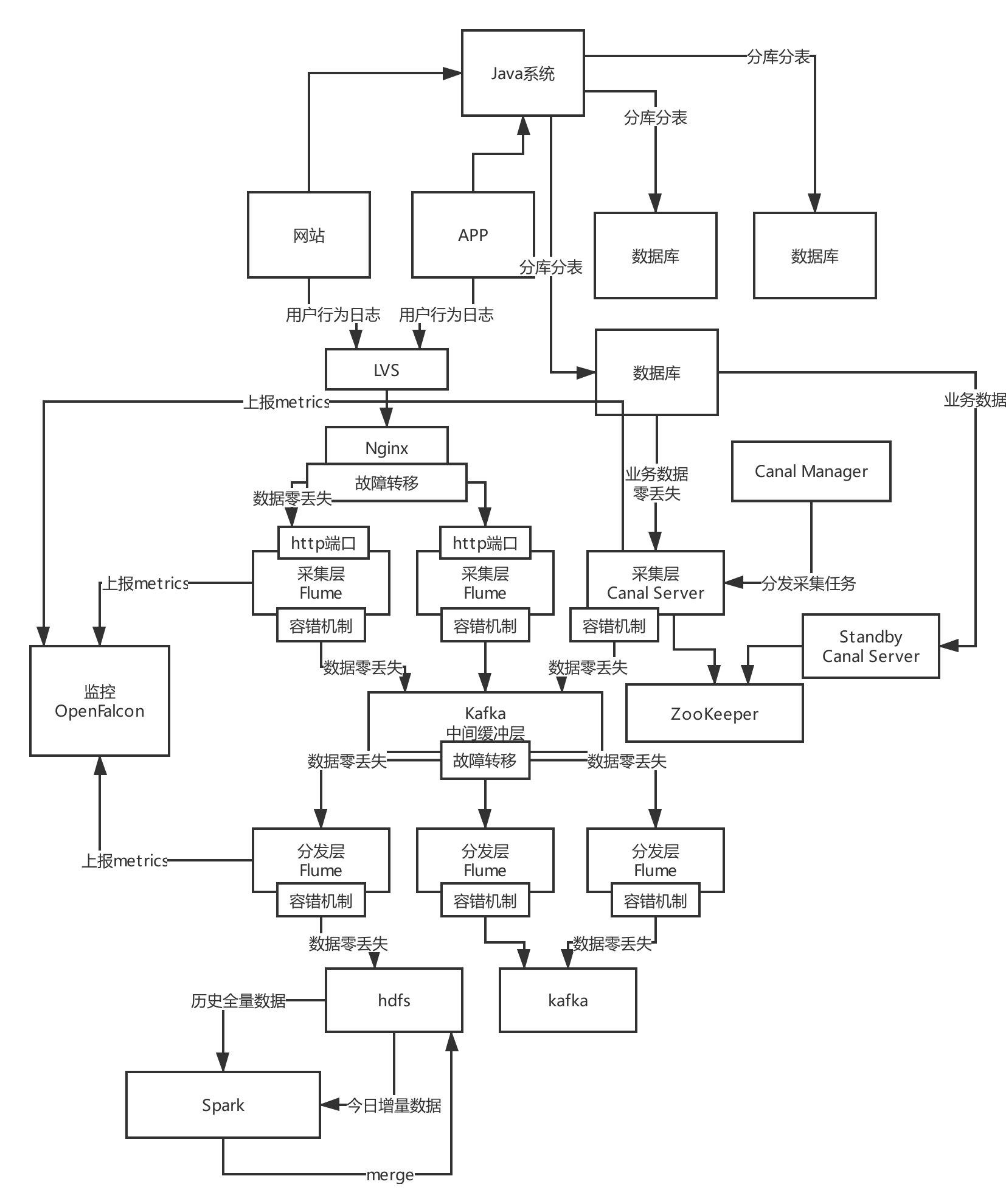
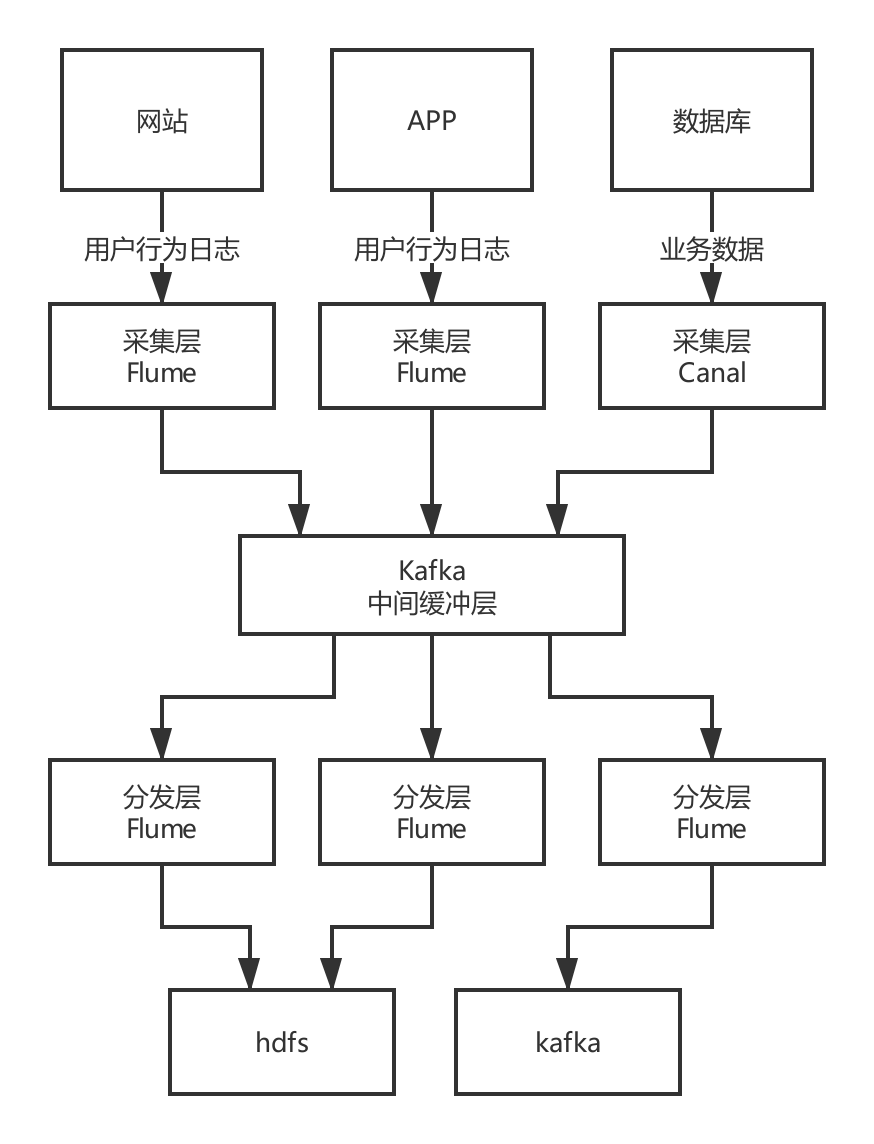
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/05\_如何借助Flume的事务机制实现全链路数据流转零丢失？/01\_数据采集平台架构设计 (1).png**



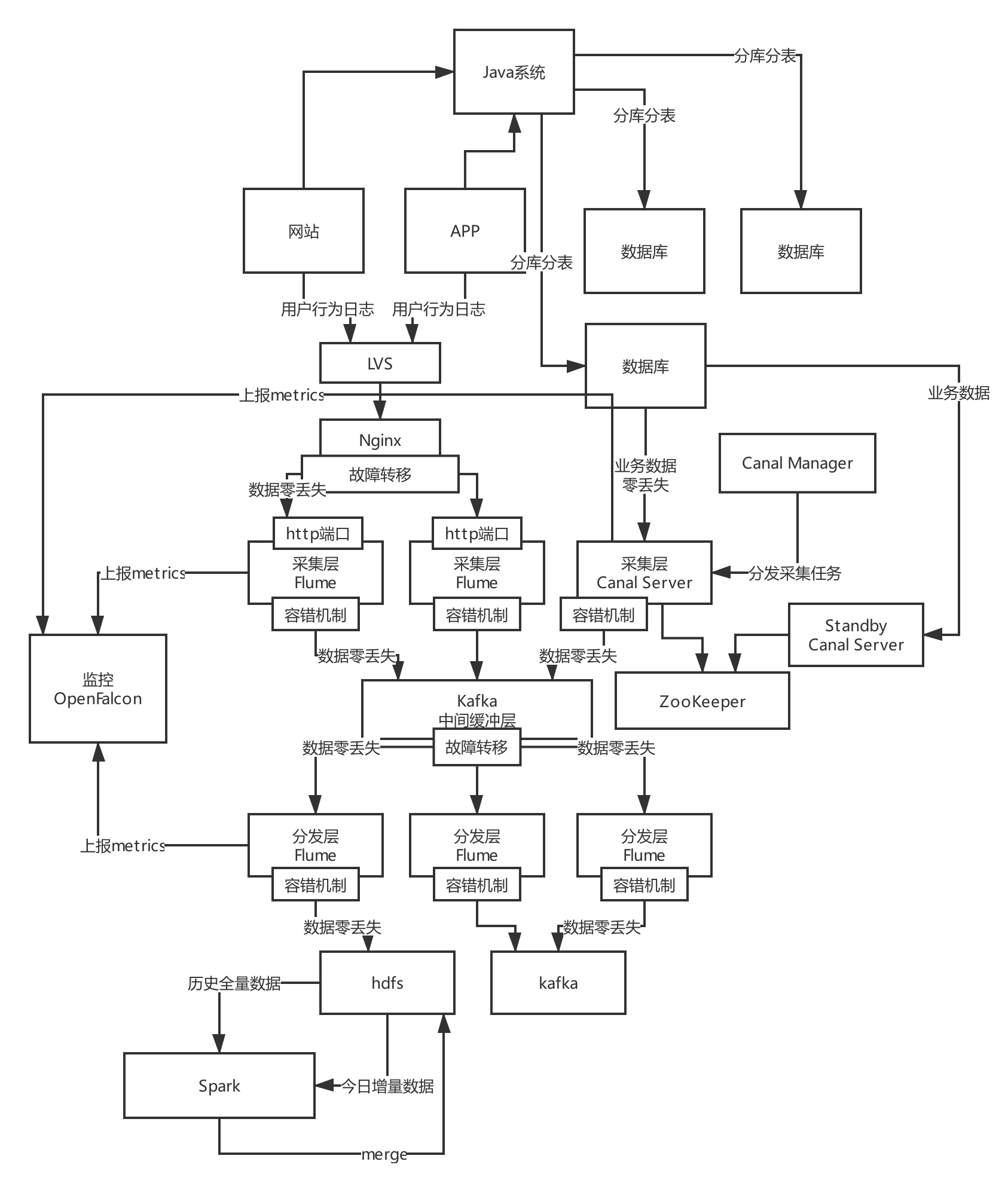
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/18\_高级大数据人才应该如何学习：生产、源码、定制、优化/01\_数据采集平台架构设计 (2).png**



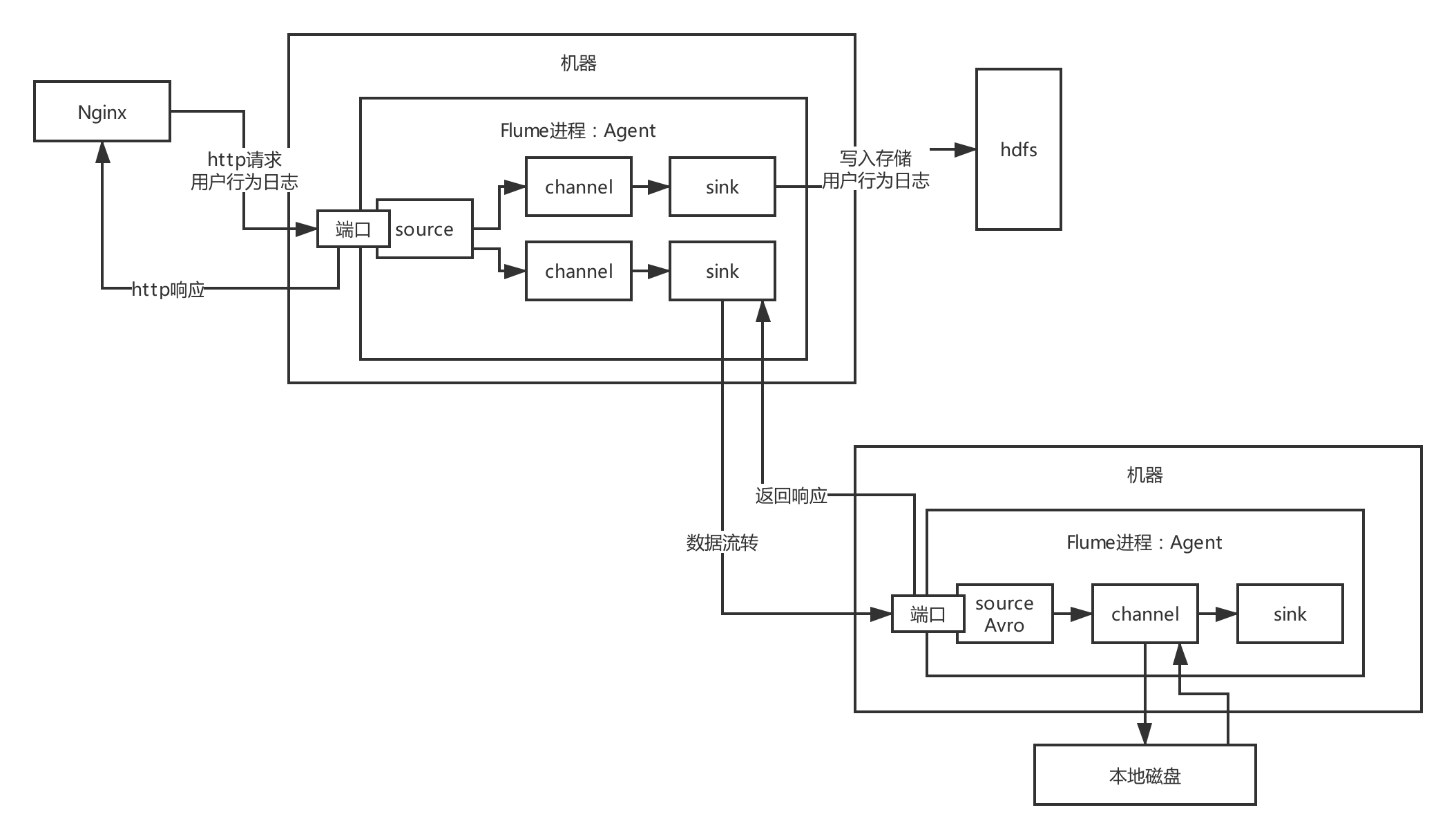
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/03\_如何基于Flume和Canal两种技术来实现数据采集平台/01\_数据采集平台架构设计.png**



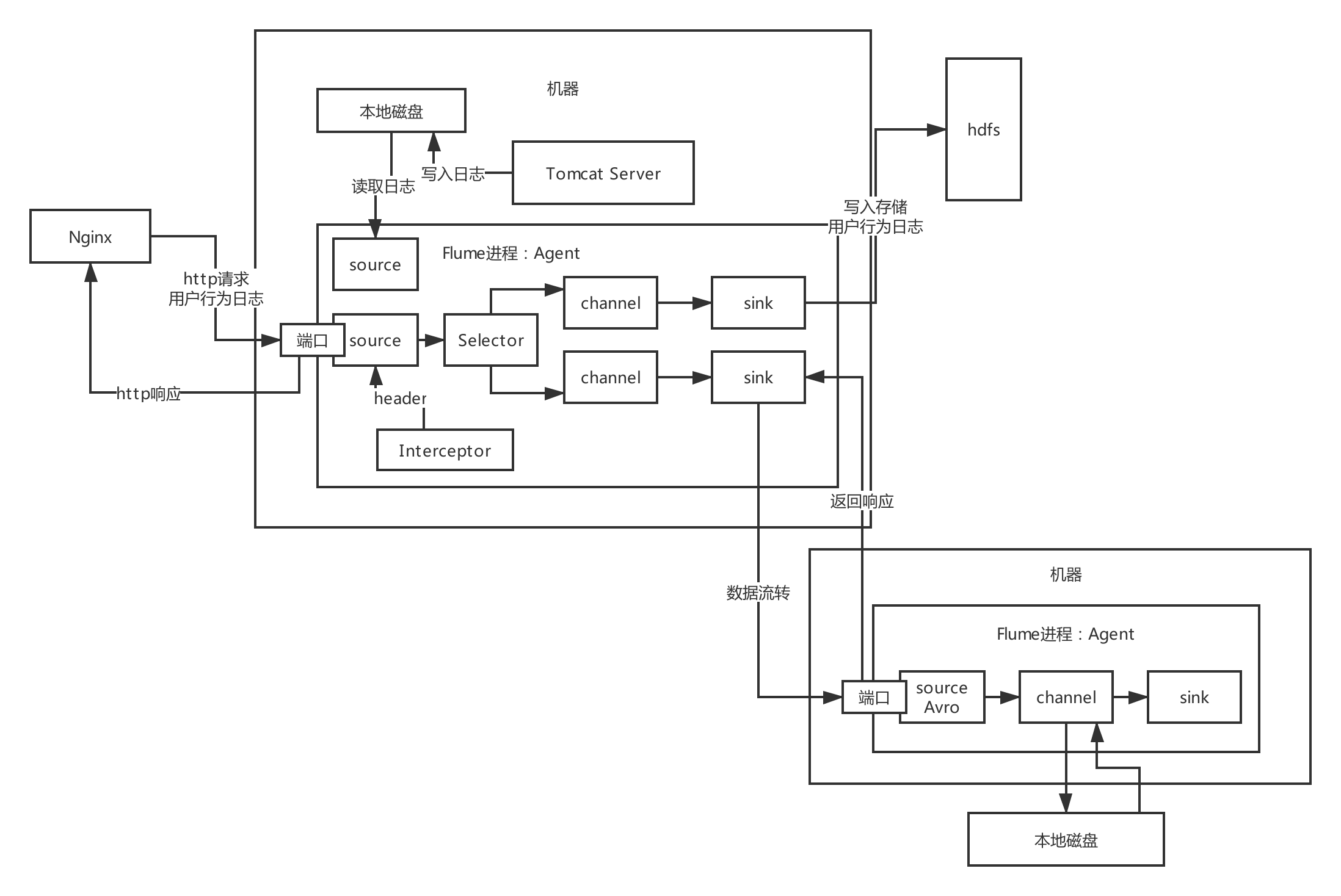
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/045\_基于Flume架构原理和核心功能实现数据采集平台的回顾/01\_数据采集平台架构设计.png**



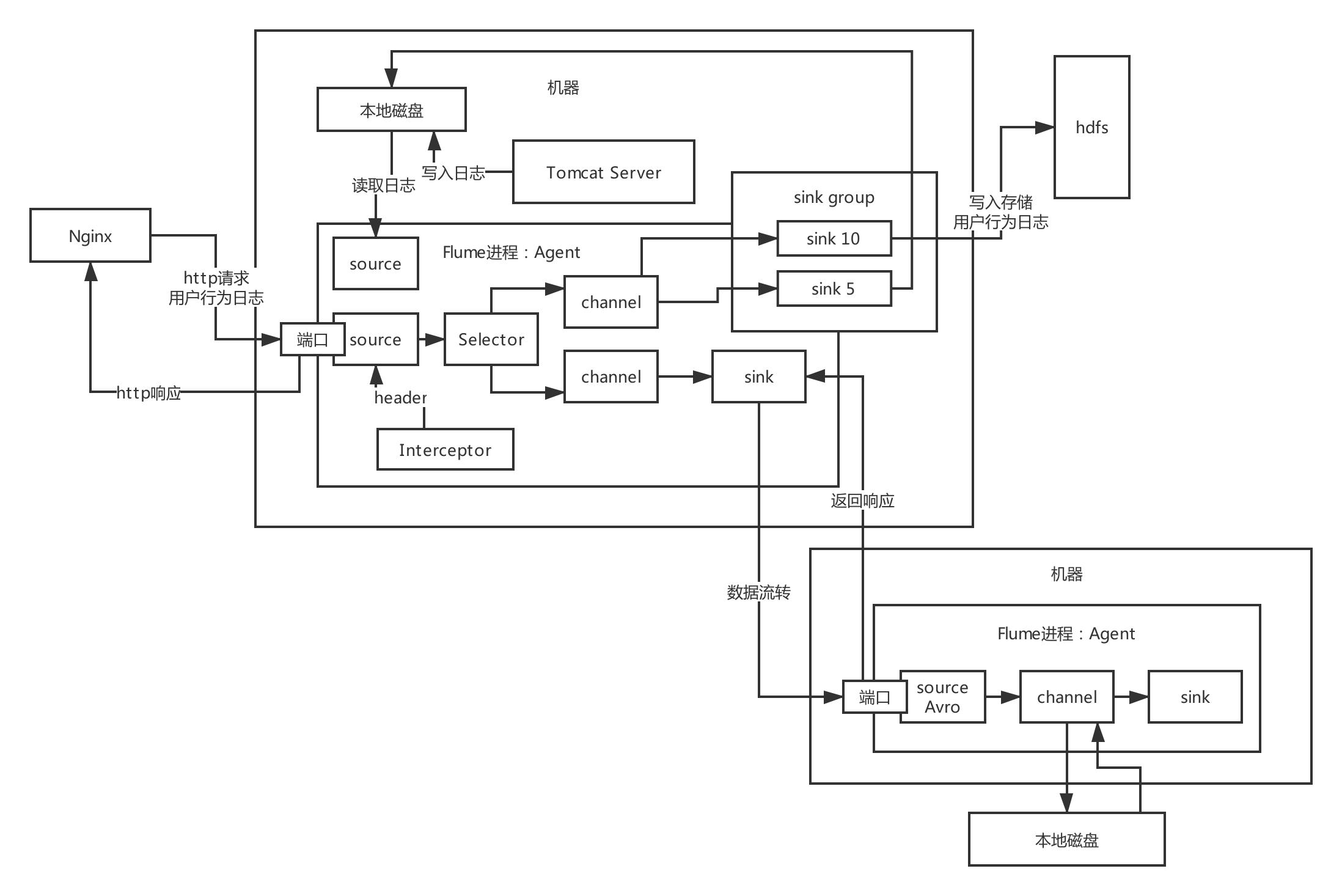
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/023\_将一条数据分发到多个数据存储的复杂数据流模型/02\_Flume架构原理 (1).png**



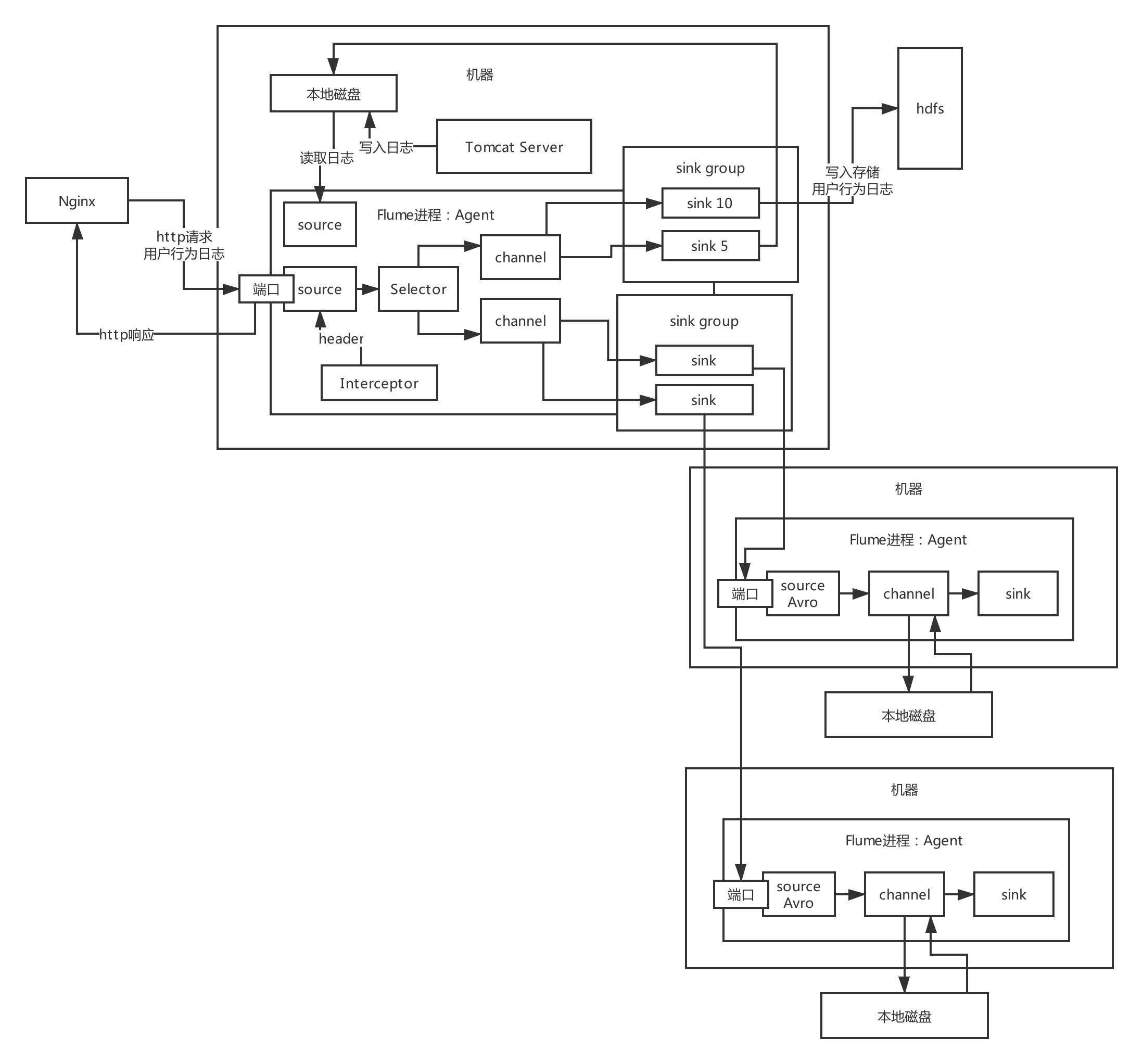
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/029\_Flume的常用Source组件的功能介绍和说明/02\_Flume架构原理 (2).png**



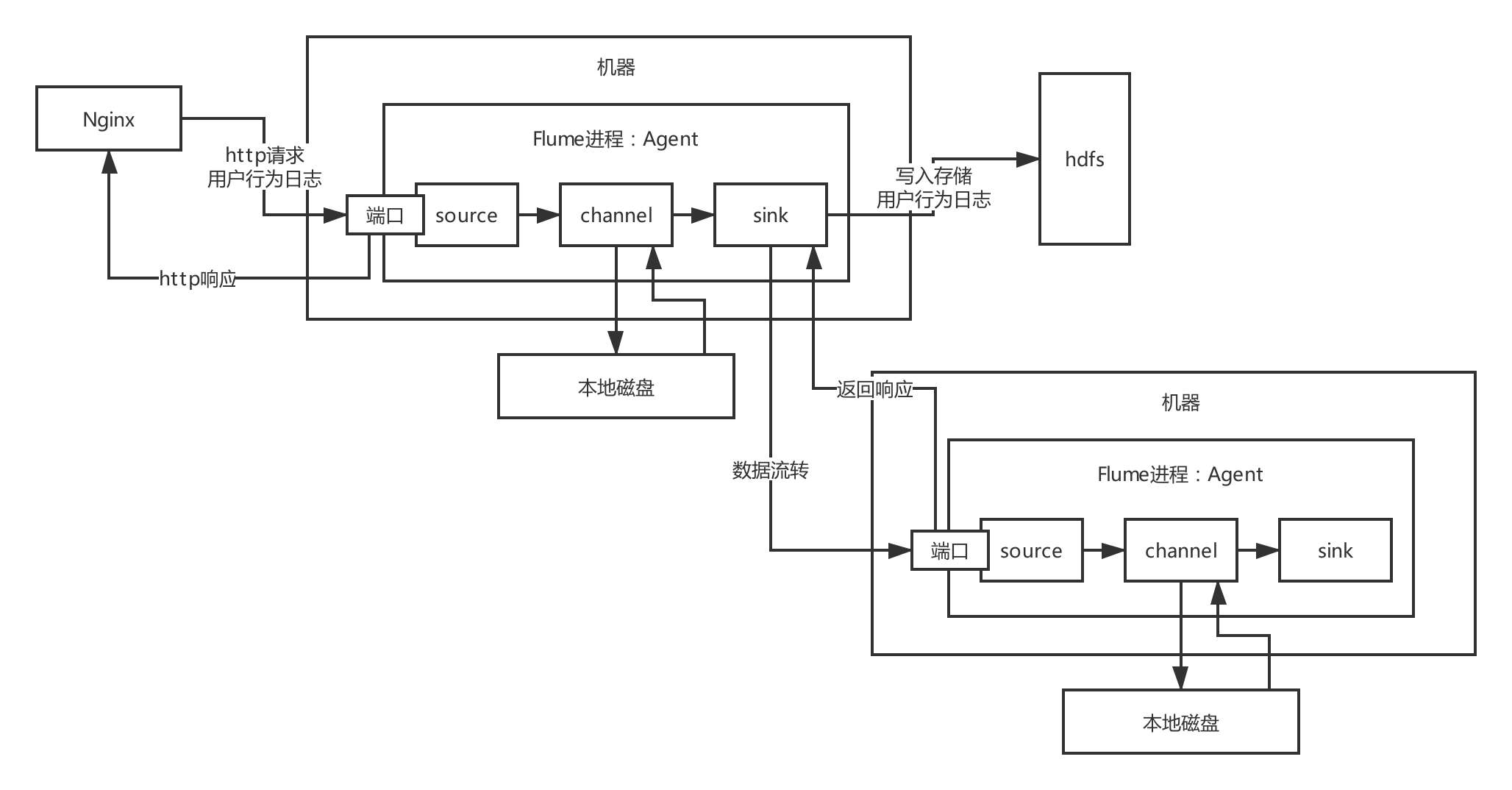
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/032\_如何基于sinkgroup机制实现存储层的failover ？/02\_Flume架构原理 (3).png**



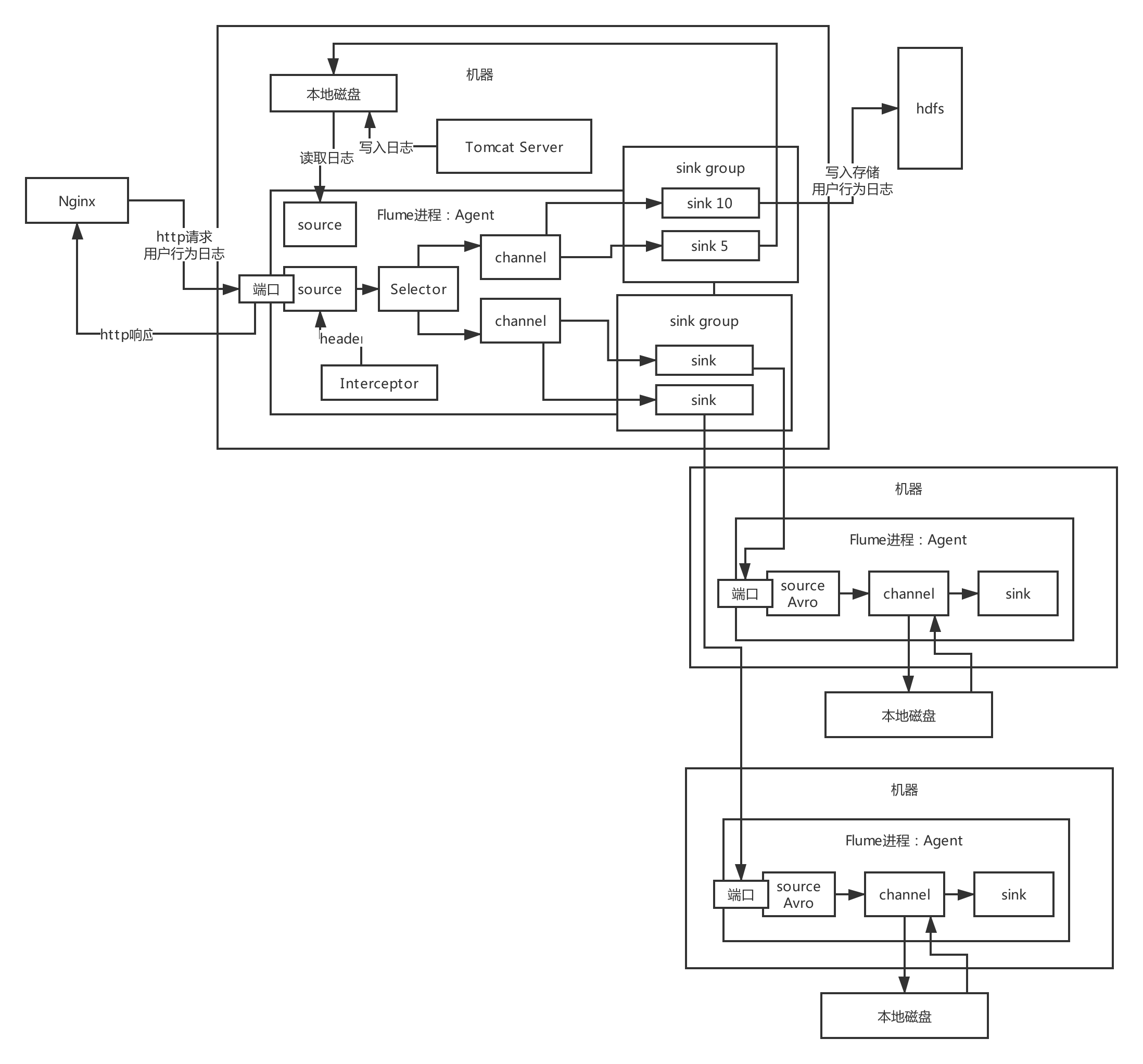
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/033\_如果设计了多Agent直连架构，如何实现负载均衡？/02\_Flume架构原理 (4).png**



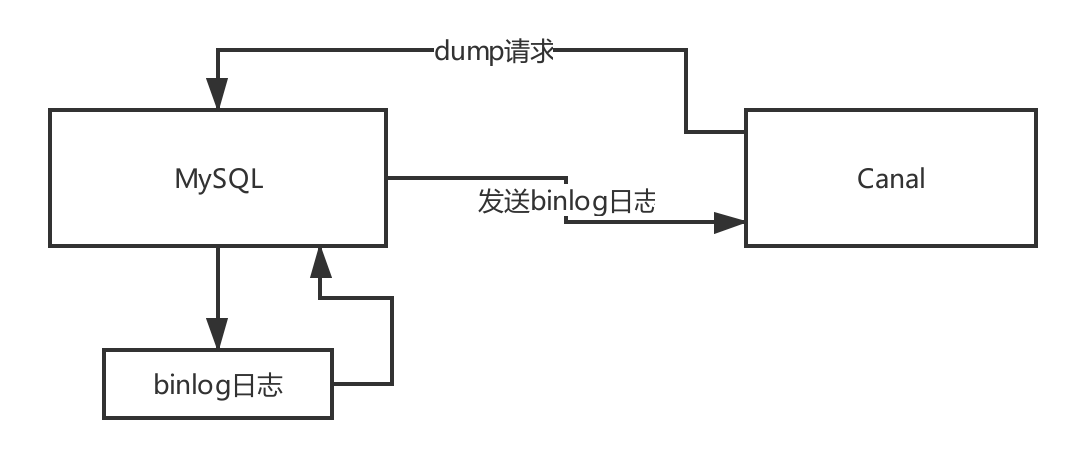
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/020\_Flume是怎么以事务性的方式保证消息传递不丢失的？/02\_Flume架构原理.png**



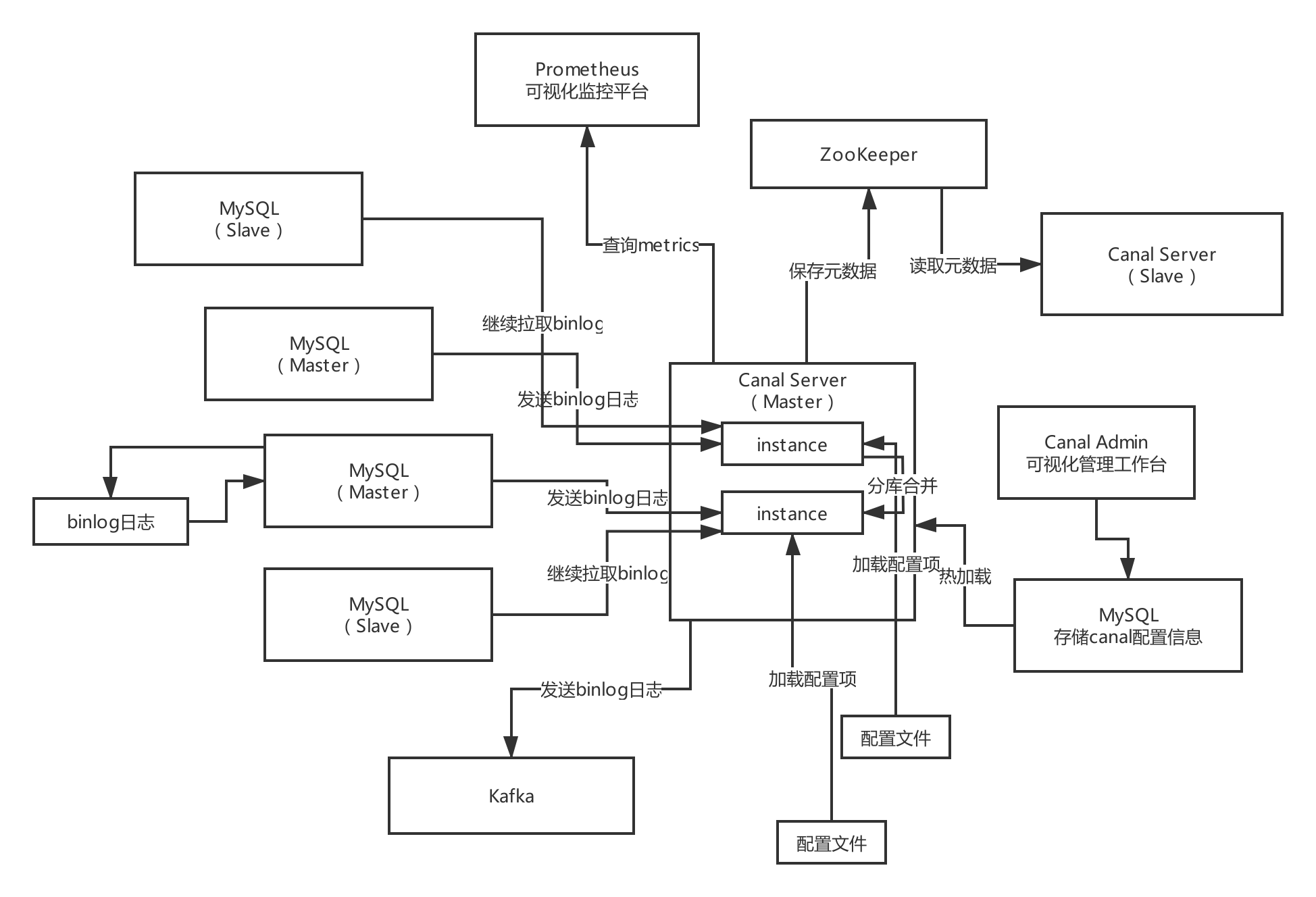
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/044\_Flume整体架构原理的回顾/02\_Flume架构原理.png**



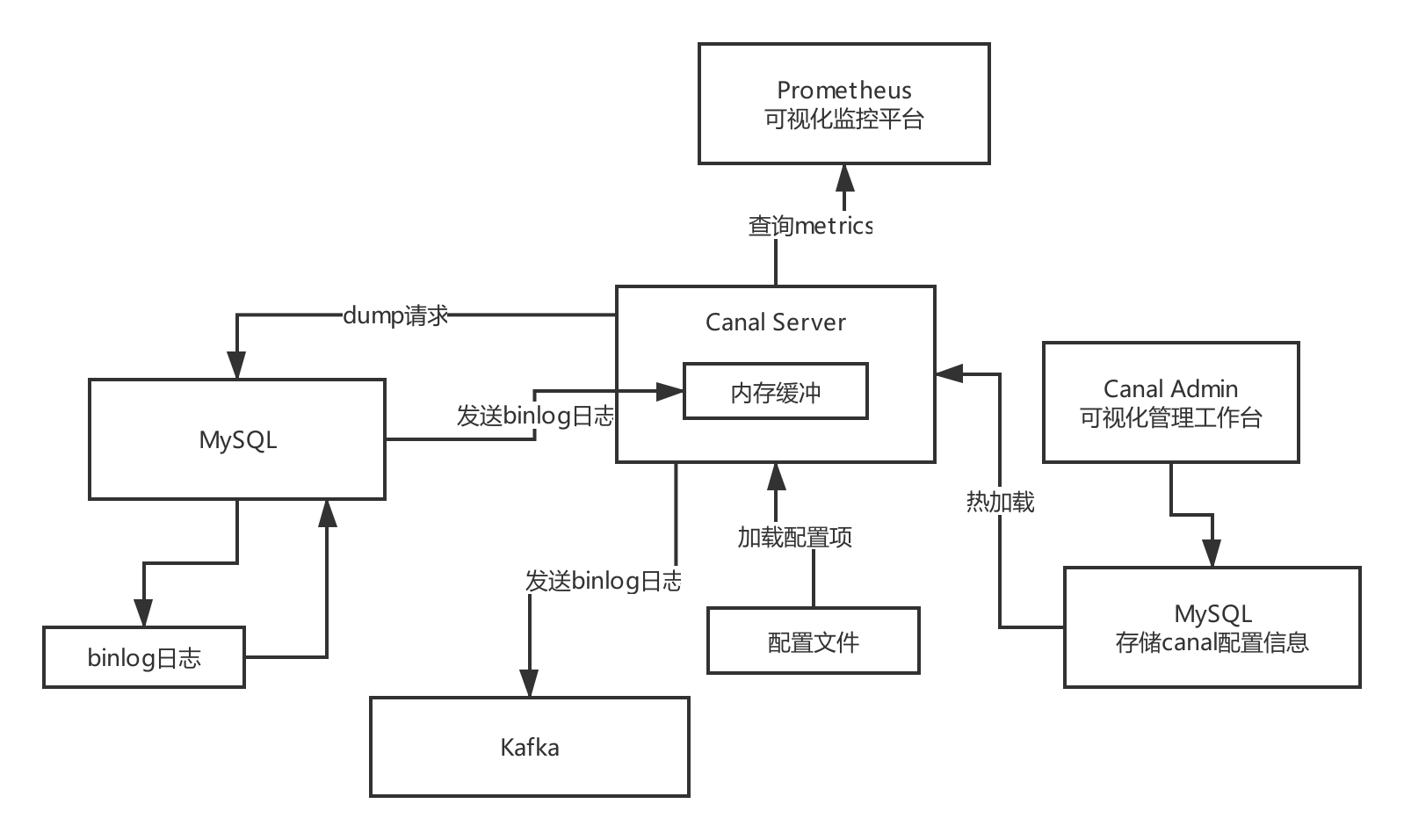
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/047\_初步对Canal的产生背景和工作原理做一个介绍/03\_Canal架构原理.png**



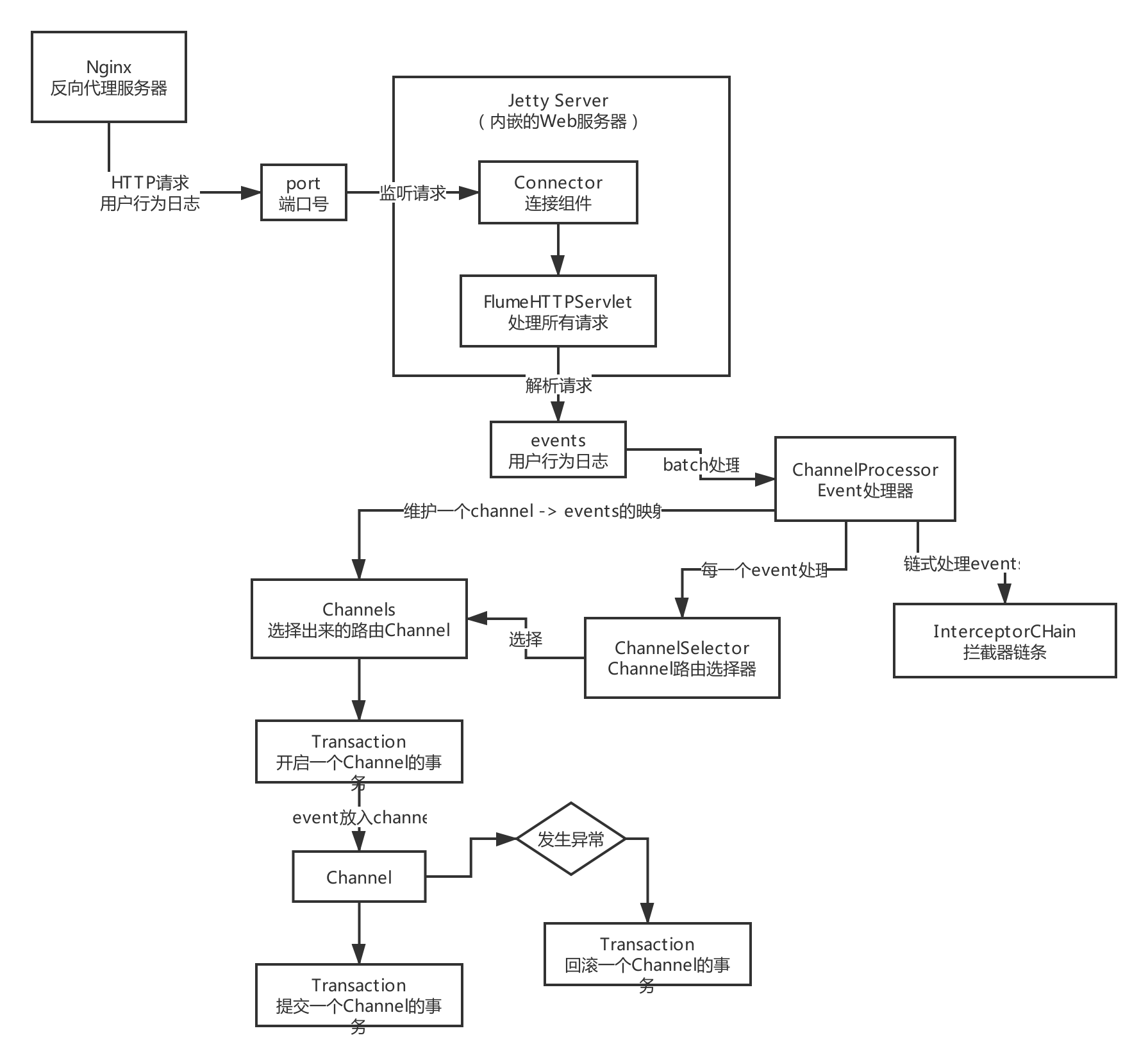
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/052\_Canal HA双机高可用架构是怎么做到的？/03\_Canal架构原理.png**



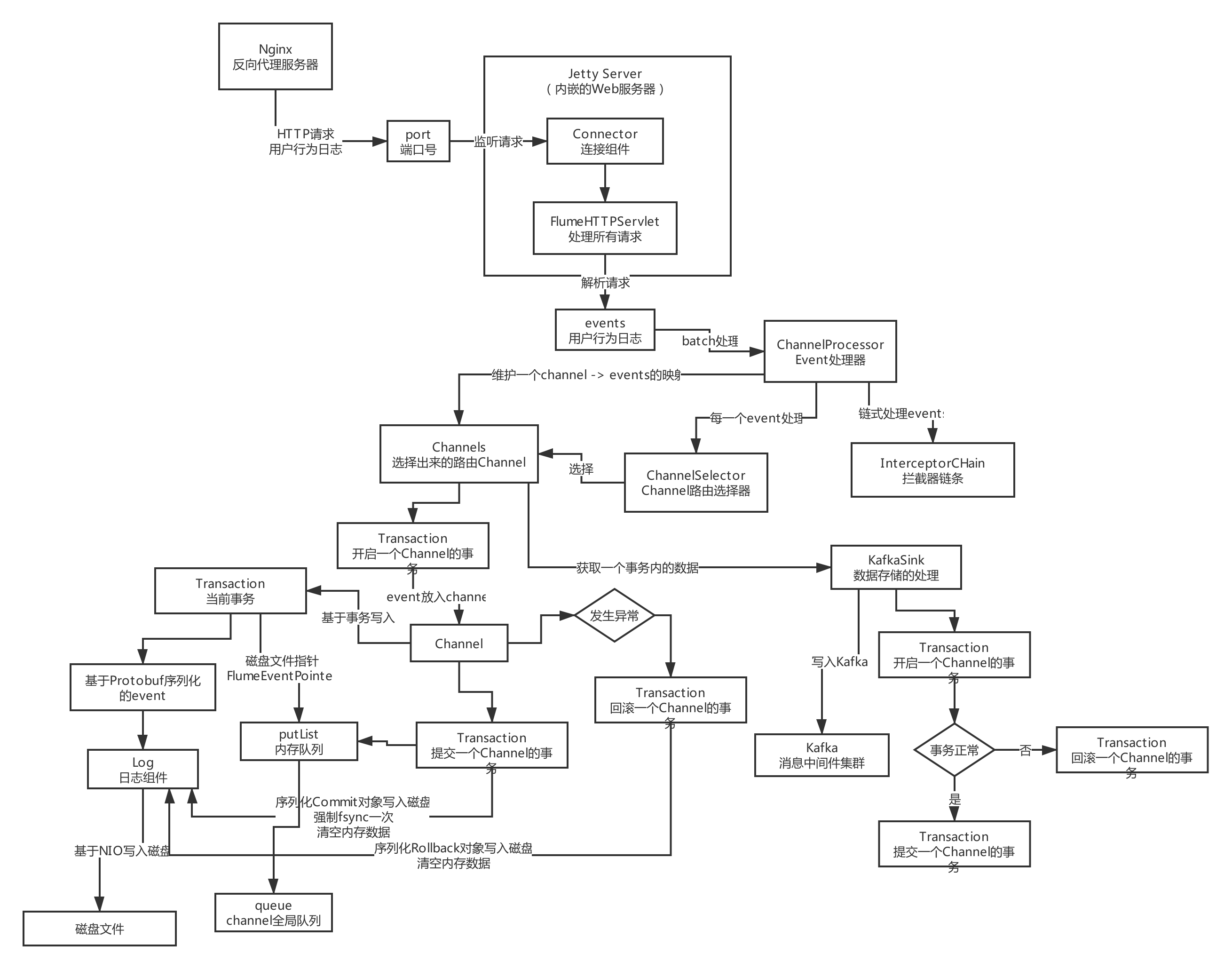
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/051\_了解一下Canal官方文档中的各种配置项的详细解释/03\_Canal架构原理.png**



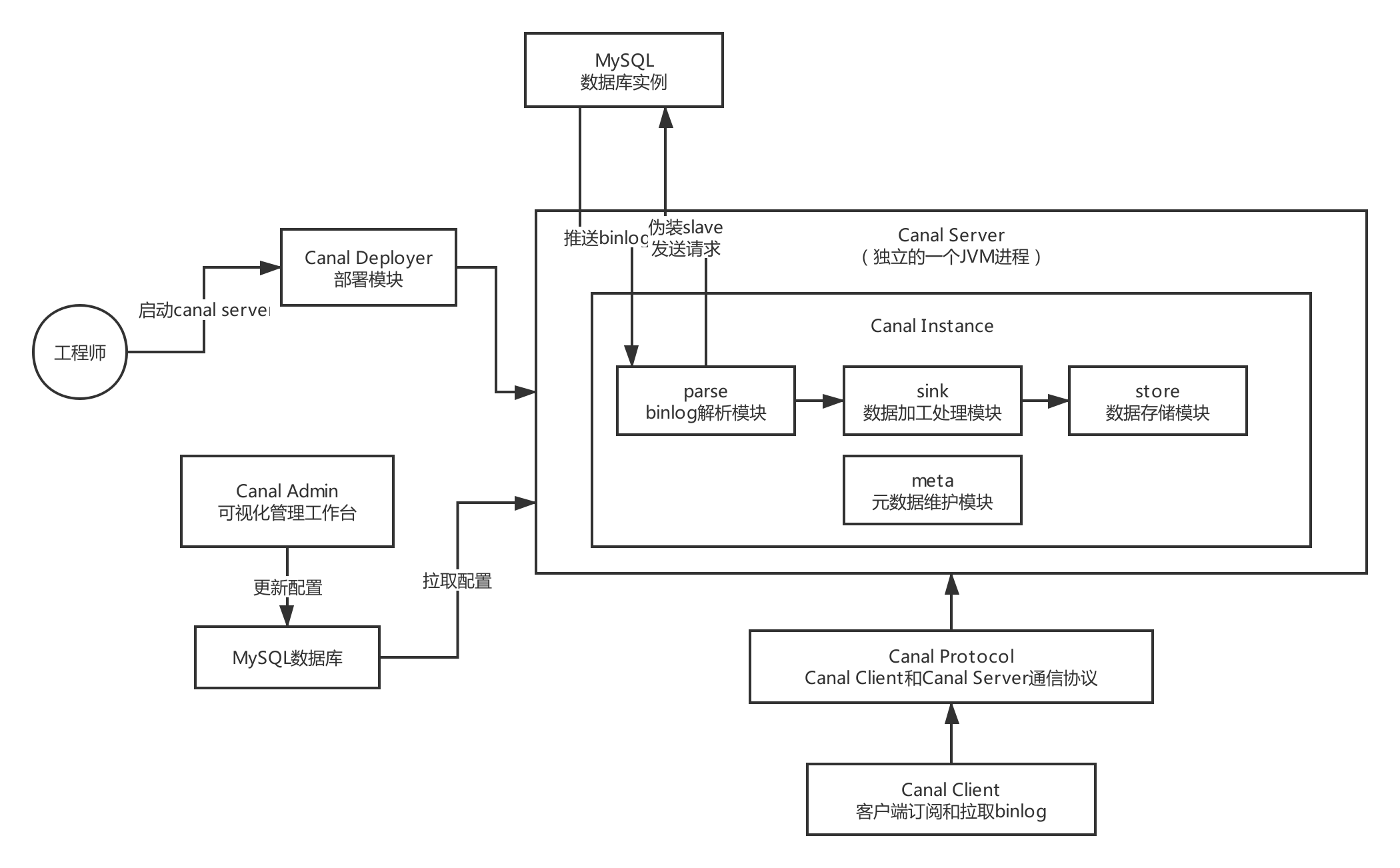
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/059\_Source是如何将一批events放入Channel中的？/04\_Flume HTTP Source的工作原理.png**



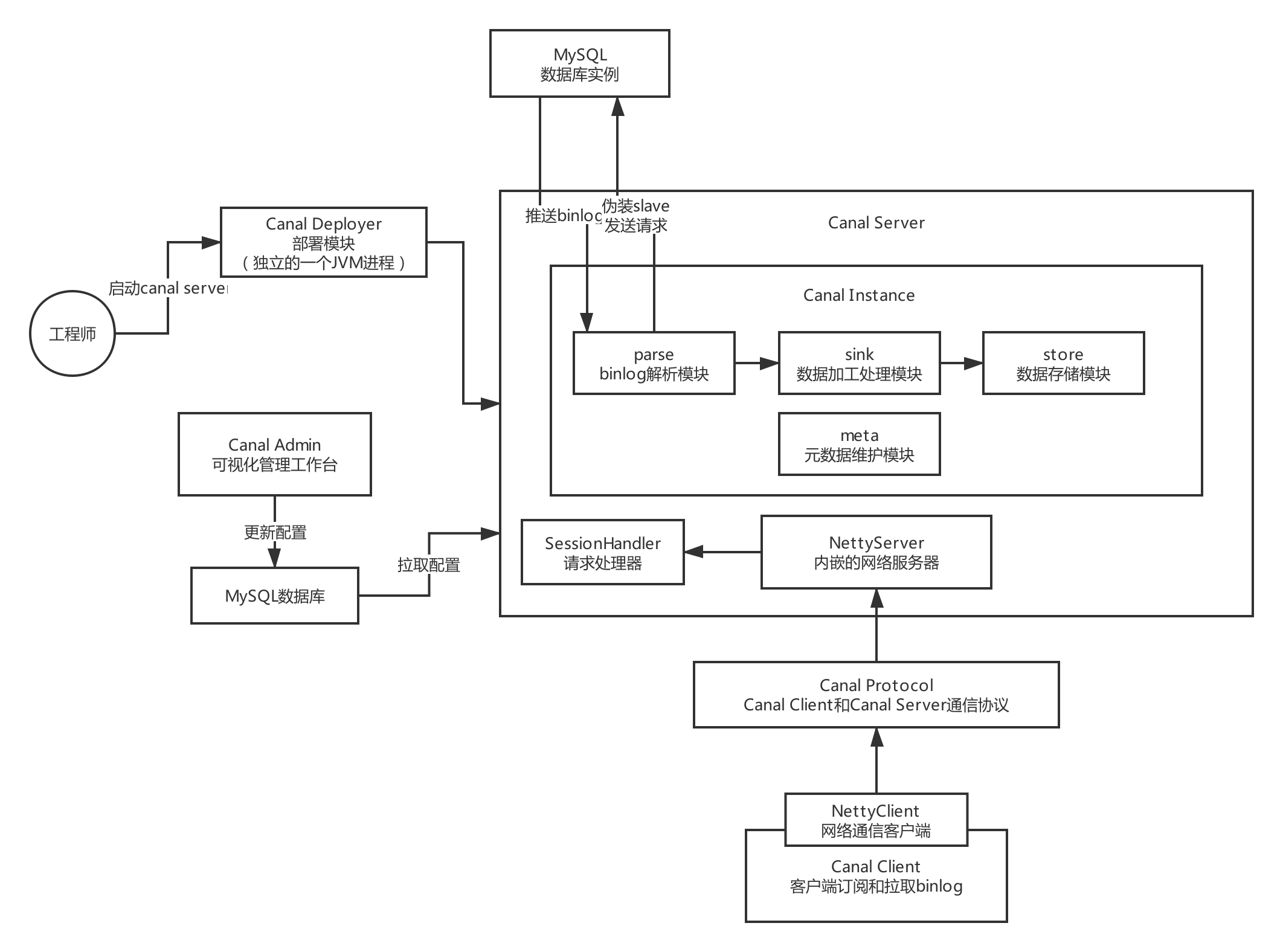
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/070\_HDFSSink的分析以及SinkGroup的工作原理/04\_HTTPSource、FileChannel、KafkaSink的工作原理.png**



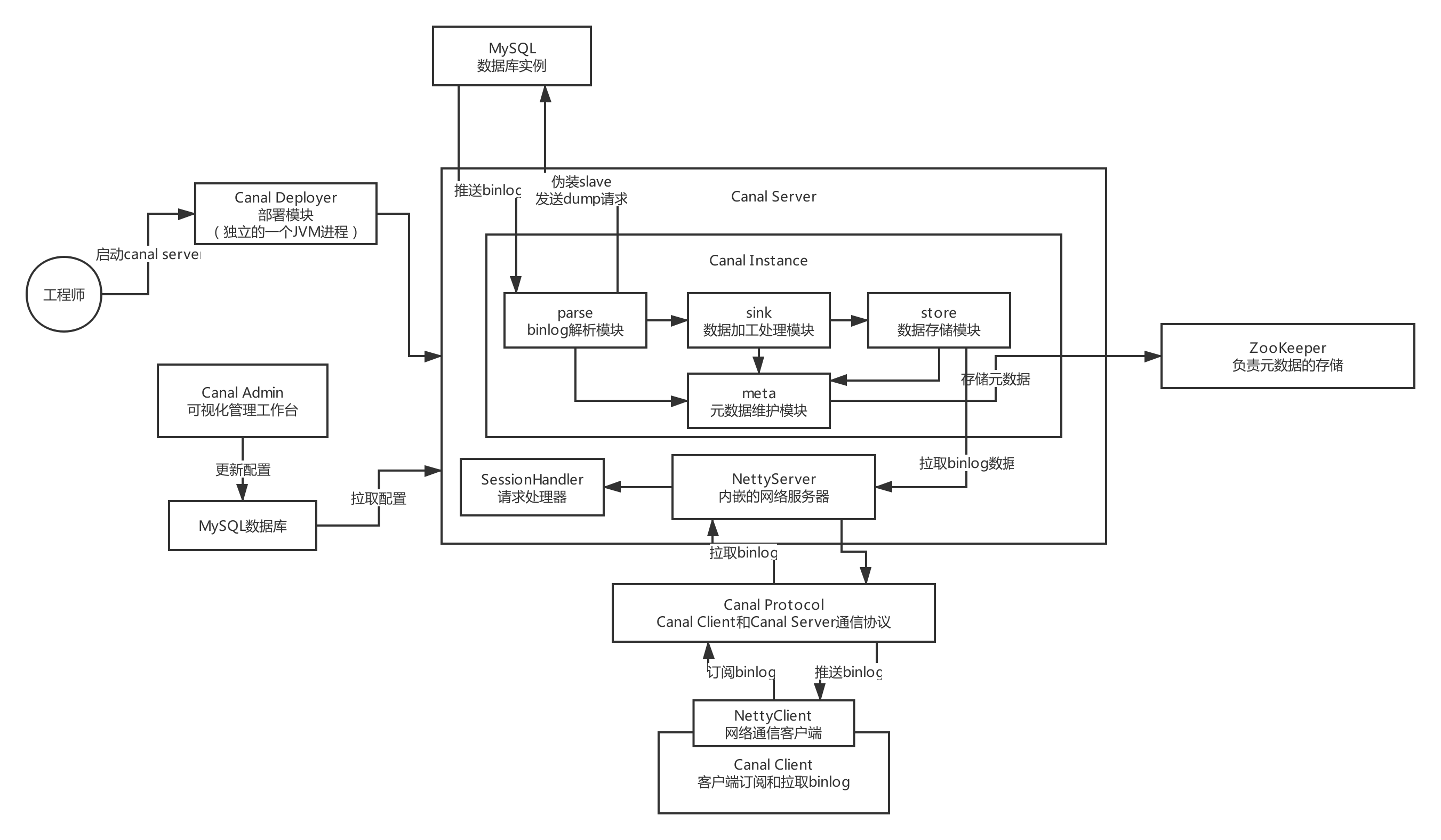
# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/072\_画图说明Canal各个组件的协作和交互关系以及整体架构/05\_Canal架构原理.png**



# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/074\_Deployer模块是如何启动Canal Server的？/05\_Canal架构原理.png**



# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/080\_Canal会丢失数据吗？如何保证数据的零丢失？/05\_Canal架构原理.png**



# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/033\_如果设计了多Agent直连架构，如何实现负载均衡？/笔记.doc**

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/032\_如何基于sinkgroup机制实现存储层的failover ？/笔记.doc**

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/040\_如何基于Flume实现数据采集平台的高可用架构？/笔记.doc**

采集层Flume的高可用

sinkgroup做故障转移，kafka故障写入本地磁盘；二次开发，自己实现一个定制版的source，专门监控kafka是否恢复，把本地磁盘缓冲的一些消息重新读取出来，走channel和sink，写入到kafka里去即可

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/035\_生产环境启动Flume时如何设置JVM参数？/笔记.doc**

设置JVM参数，尤其是《从0开始带你成为JVM实战高手》，优化一下他的参数

flume，conf，flume-env.sh，里面可以配置对应的JVM参数

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/036\_Flume运行时的日志在哪里可以看到？如何配置？/笔记.doc**

日志，log4j，conf，log4j配置文件

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/039\_Flume集群监控的几个层次：机器、JVM、Metric/笔记.doc**

每台机器的资源使用量：CPU、内存、网络、磁盘

JVM：GC频率

Metrics：每个Flume，他的source、channel、sink

HDFS，JVM参数是不是要优化？对HDFS进行监控，HDFS集群所在机器的负载、JVM、Metrics，HDFS平时运行打印出来的日志，我们还得对HDFS集群做一套运维的方案，资源使用量的监控，到什么程度就得扩容

平时检查一下损坏的数据和磁盘

HDFS：监控方案、扩容方案、升级方案、日志检查、报警方案、优化方案

Kafka：监控方案、扩容方案、升级方案、日志检查、报警方案、优化方案

Flume：监控方案、扩容方案、升级方案、日志检查、报警方案、优化方案

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/043\_如何基于Flume实现数据采集平台的可伸缩性？/笔记.doc**

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/041\_如何基于Flume实现数据采集平台的数据零丢失架构？/笔记.doc**

数据真的是必须得0丢失？

用户行为日志，产出的一个数据报表，无非就是看一个趋势，折线图，精准的数字是没人关心的，每天用10984561次点击，每天有10984510次点击。但是量大，100万日活用户，用户行为日志上亿，量很大，吞吐量很大

业务数据库里的数据，截止昨天为止，APP上的营收是1030万，APP上的营收是980万，50万，这个就不行了。每天100万日活用户，大量的都是一些业务库的查询，查商品，查订单，查评论。订单10万单。

Spillable Memory Channel，优先是写数据到内存

对于分发层的flume，做一点二次开发，定制一个interceptor拦截器，对日志数据，打入一种header，type=log；对业务库的数据，打入一种header，type=db

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/034\_Flume支持哪些Intercepor给数据加入Header？/笔记.doc**

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/032~043资料/042\_如何基于Flume实现数据采集平台的高吞吐量？/笔记.doc**

对Flume的FileChannel做一点二次开发，尽量让数据落地到os cache里，然后从os cache里

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/04\_基于Flume的日志采集架构如何设计实现高可用性？/笔记.doc**

collector死、dispacher死或慢、hdfs/kafka死或慢

采集层部署的Flume，一般来说，有两种采集日志的方式，第一种方式，你可以自己开发一个Java系统部署了，用他们来接收人家APP和网站发送过来的日志，也可以是用Nginx来接收那些日志，写入磁盘里去

Flume直接从磁盘目录里读取人家不断写入的日志

APP、网站，日志发送回来，基于http协议发送回来的，都会走到前置LVS、Nginx上去，Nginx会做一个反向代理，他可以把你的请求负载均衡打到后端你的多台机器上去，这些机器上你就可以部署Flume

采集层的Flume高可用保障，Nginx做一个故障转移

很多课程，Nginx、Tomcat采集日志，写入磁盘，通过flume直接从磁盘里读取日志，如果说这种日志还需要落地磁盘的话，那岂不是会导致采集的性能和吞吐量都会降低

Kafka死掉了，我们要对flume做一个二次开发，一旦说发现kafka死掉了，你就必须把日志写入本地磁盘，暂存起来，不停的判断，如果kafka后续又恢复过来了，此时就可以重新把本地磁盘暂存的日志再次写入kafka

这批flume应该走的是相同的consumer group，分发层的flume宕机，基于kafka consumer group的语义，自动进行顾长官转移

存储层的hdfs和kafka如果死掉了，对flume做一点二次开发，如果发现hdfs或者kafka死了，此时应该把数据暂存起来，后续发现hdfs和kafka活过来之后，再次从磁盘里读取数据发送过去

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/10\_基于Canal的数据库binlog采集架构该如何设计？/笔记.doc**

canal manager

canal server，两台，基于zk做高可用

一个mysql实例一份binlog，包含多个db的，canal选择其中订阅的db binlog，发送到kafka

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/01\_课程思路：架构设计、技术学习、实战部署、源码剖析/笔记.doc**

初步分析完了我们的数据化运营平台的需求

第一步，你针对你要做的这些需求，首先得把你需要的数据采集过来，分成两种，第一种是用户行为日志，第二种是业务系统里的数据库的数据

HDFS、Kafka，这两个技术是最最关键的，一个是离线数据的存储，一个是实时数据的存储，源码级别搞定了两个技术，包括一些实战性的东西

数据采集平台 -> 数据仓库建模规范 -> 数据仓库 -> 数据开发工具链平台（即席查询平台、ETL开发平台、作业调度平台） + 数据治理平台（元数据、数据质量） -> ETL开发规范 -> 动手开发对应的ETL -> 可视化BI平台 -> 针对YARN、Hive和Spark的源码深入探究 -> ETL性能优化 -> 数据化运营平台

明年实时（flink） -> 分析（druid、kylin、ELK）-> 数据中台

把大数据领域所有通用的一套东西，包含了通用的一套数据仓库模型，通用的工具平台，中台底层是什么东西？集群，YARN集群，HDFS集群，HBase集群，ES集群

大中台，小前台，各种业务方向的数据团队，外卖、单车、团购，到家、找工作、生鲜、商超，这些业务的数据团队，都不用自己做什么开发了，中台提供的整体的基础数据和工具链平台，做一些数据的建模、开发，基于你的平台来做，产出一些业务关注的一些东西，离线、实时、分析

机器学习算法、spark mllib、深度学习算法，tensorflow，讲一些跟大数据相结合的一些智能化的项目，智能物流，智能车辆调度，智能配送调度

搜索、推荐、商家

把整体的数据采集平台的架构给设计出来，学习对应的技术，flume、canal，实战部署，源码，做一些二次开发，核心源码会走读一下，生产环境机器部署

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/05\_如何借助Flume的事务机制实现全链路数据流转零丢失？/笔记.doc**

把Java给我学好

数据零丢失，采集层的Flume宕机可能导致数据丢失

第一个办法，把每条数据都写入磁盘，从磁盘读取出来发到kafka上去

结合nginx故障转移的机制，flume源码，接收到一条数据之后，后续是如何可以判定数据已经写入kafka了，此时才能告诉nginx说本次请求成功。但凡数据没成功发送给kafka，都需要返回http异常码给nginx

kafka数据零丢失

必须保证说数据写入了多个副本了，此时才能认为是写成功了，否则就是写失败，让nginx重发消息

分发层flume，结合源码，做点二次开发，去保证他必须是把数据写入到hdfs和kafka了，此时才能commit offset

结合flume源码做很多的代码改造，保证他全链路的数据零丢失

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/08\_生产环境下对Flume日志采集流程如何进行优化？/笔记.doc**

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/07\_如何基于OpenFalcon进行Flume日志采集系统的监控/笔记.doc**

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/06\_Flume日志采集架构能支持按需线性扩容抗更大数据量吗？/笔记.doc**

高可用、全链路数据0丢失、高并发、高性能

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/09\_按每天TB级数据量来计算Flume的生产环境部署机器需求/笔记.doc**

考虑生产环境机器部署需求，每天大概多少条日志，多少TB，每秒并发量多少，考虑部署多少台采集层flume，kafka需要部署多少台机器，分发层flume需要部署多少台机器，存储层的hdfs和kafka需要部署多少台机器，抗下日增量的数据，而且还抗住高并发写入

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/02\_数据采集平台的三层架构设计：采集层、分发层、存储层/笔记.doc**

点击一个商品，搜索一个关键词，点击一个按钮，下了一个订单，支付了一笔订单，用户行为，互联网公司一般都会采集用户行为

网站+APP，采集下来，网站上嵌入JS脚本，APP嵌入一些日志，他们把用户行为日志，发送到后台，让我们的数据采集系统来采集进来

数据库，订单数据，用户数据，优惠数据，活动数据

hdfs，kafka，两种存储，离线，实时

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/11\_MySQL binlog数据的采集有一个巨大的问题/笔记.doc**

binlog

id = 1的数据

insert binlog，日志，在某个表里插入了一条数据

update binlog，日志，在某个表里修改了一条数据

delete binlog，日志，在某个表里删除了一条数据

代表了数据库里完整的一份数据

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/001~018资料/03\_如何基于Flume和Canal两种技术来实现数据采集平台/笔记.doc**

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/021\_Flume是配置数据流以及启动Agent进程的？/笔记.doc**

部署flume到一台机器上去

下载一下flume二进制压缩包，tar.gz，放到一台linux机器上去，解压缩

对他做一些配置，配置好对应的数据流，再启动这台flume就可以搞定了，他就开始运行起来了，不在这里做一些demo级别的演示

JVM参数优化，核心参数优化，日志，监控（系统监控，JVM监控，机器监控）

狸猫技术窝，知识店铺，专栏，《从0开始带你成为JVM实战高手》

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/030\_Flume的常用Channel组件的功能介绍和说明/笔记.doc**

Memory Channel：直接数据流转走内存队列，速度快，但是可能丢数据

File Channel：写本地磁盘，数据不会丢，但是速度会比较慢，数据流转中间落地一次磁盘，会对性能有一定的影响

Spillable Memory Channel：优先写内存队列，从内存队列里读取数据，速度比较快；但是如果内存队列满了，此时就写磁盘，从磁盘读取数据；宕机，会导致内存队列里的数据会丢失掉的

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/019\_Flume的核心架构原理以及数据流动模型/笔记.doc**

分布式存储系统，超大的大文件拆分成很多小块，放在多台机器上

Kafka，玩儿Java系统的同学也是经常会用的

Flume，Canal，本身都是Java开发的，数据采集而言

读读核心源码，基于核心源码做点二次开发

Canal也同样是比较简单的这样的一个系统，都不难，读读核心源码，做点二次开发

关于Flume功能的讲解，直接打开官方文档，给大家一边讲解官方文档的功能说明，一边来画图，让大家了解清楚Flume完整、复杂的一套架构是如何来运行的

你了解任何一门技术，第一途径就是官方文档，不是看任何的书，版本老旧，根本没有官方文档好，英语，教你看官方文档的方式，Canal

数据采集平台，读源码，二次开发，部署，测试，最后把整个架构和流程跑通，生产环境的机器配置、JVM参数调优、核心参数优化、监控架构

日志埋点，业务库表结构，代表着数据的来源 -> 数据需求

结合一个一个细化的数据报表需求，去做数据仓库的建模，规范，大公司流程走一遍，完全胜任国内任何一个公司的数据仓库架构师这块的职责

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/025\_Flume数据流模型中的各个组件的属性如何配置？/笔记.doc**

# list the sources, sinks and channels for the agent

a1.sources = r1

a1.sinks = k1,k2

a1.channels = c1,c2

a1.sources.r1.type = http

a1.sources.r1.bind = localhost

a1.sources.r1.port = 9000

a1.channels.c1.type = memory

a1.channels.c1.capacity = 1000

a1.channels.c1.transactionCapacity = 100

a1.channels.c2.type = memory

a1.channels.c2.capacity = 10000

a1.channels.c2.transactionCapacity = 100

a1.sinks.k1.type = hdfs

a1.sinks.k1.hdfs.path = hdfs://namenode/flume/webdata

a1.sinks.k2.type = avro

a1.sinks.k2.hostname = localhost

a1.sinks.k2.port = 44444

# set channel for source

a1.sources.r1.channels = c1,c2

# set channel for sink

a1.sinks.k1.channel = c1

a1.sinks.k2.channel = c2

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/024\_Flume中复杂的数据流模型是如何配置出来的？/笔记.doc**

# list the sources, sinks and channels for the agent

a1.sources = r1

a1.sinks = k1,k2

a1.channels = c1,c2

# set channel for source

a1.sources.r1.channels = c1,c2

# set channel for sink

a1.sinks.k1.channel = c1

a1.sinks.k2.channel = c2

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/022\_多个agent串联起来的较为复杂的数据流模型/笔记.doc**

经验，如果你要玩儿多个Agent串联起来的数据流

avro sink -> avro source

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/029\_Flume的常用Source组件的功能介绍和说明/笔记.doc**

Avro Source：两个Agent之间的数据流转

Spooling Directory Source：比如你要获取的用户行为日志，都是部署在Tomcat里的Web系统直接写入本地磁盘的话，此时Agent就得用这种方式去获取本地磁盘日志

Kafka Source：flume从Kafka里消费数据

HTTP Source：按照HTTP协议来走了，Nginx把HTTP请求转发给我就可以了

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/020\_Flume是怎么以事务性的方式保证消息传递不丢失的？/笔记.doc**

Flume在数据流转的时候，他的一些消息传递的语义

event保留在channel里，必须进入最终存储，或者进入下一个agent的channel，才算ok，这是事务性的

agent内部有事务，两个agent之间，sink和source是一个事务

channel用file模式，可以持久化event，如果宕机重启会恢复消息投递，不会丢；但是如果用纯内存的模式，那么宕机就会丢数据

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/031\_Flume的常用Sink组件的功能介绍和说明/笔记.doc**

HDFS和Hive Sink：让他直接把数据写入都hive表里去

库和表，往hive里去导入的时候，建议mysql里有一个库+表，在hive里也应该是一个库+表，header，每个数据都有db和tb，multiplexing，不同的db和tb流入不同的channel和sink，对于一个sink就配置一个hive里的表

修改一些HiveSink的源码，他判断一下，如果是对应于不同的db的表，写入到hive不同的表去

Avro Sink：Agent之间的数据流转

HBase Sink和Elasticsearch Sink：也有用，常用，但是不用在我们这里

Kafka Sink：把数据写入kafka去

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/026\_在一个Flume Agent中配置多个数据流模型怎么做？/笔记.doc**

一个agent，可以支持从source -> channel -> sink，多条数据流

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/023\_将一条数据分发到多个数据存储的复杂数据流模型/笔记.doc**

replicating模式，可以把消息分发给多个channel

multiplexing模式，可以按照路由规则把消息路由给不同的channel

不同的channel把数据发给不同的sink，落地到不同的存储，这个模式是很有用的

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/028\_fan-out multiplexing模式的数据流模型如何配置？/笔记.doc**

replicating，就是source把自己的一条数据分发给所有的channel

高阶功能，interceptor，拦截器，是可以给你的每条数据加入不同的header

有这么一个高阶的功能，配合后面interceptor给每个数据打上heade，multiplexing模式，可以根据不同的header的值，分发到不同的channel和sink里去

# list the sources, sinks and channels in the agent

agent\_foo.sources = avro-AppSrv-source1

agent\_foo.sinks = hdfs-Cluster1-sink1 avro-forward-sink2

agent\_foo.channels = mem-channel-1 file-channel-2

# set channels for source

agent\_foo.sources.avro-AppSrv-source1.channels = mem-channel-1 file-channel-2

# set channel for sinks

agent\_foo.sinks.hdfs-Cluster1-sink1.channel = mem-channel-1

agent\_foo.sinks.avro-forward-sink2.channel = file-channel-2

# channel selector configuration

agent\_foo.sources.avro-AppSrv-source1.selector.type = multiplexing

agent\_foo.sources.avro-AppSrv-source1.selector.header = State

agent\_foo.sources.avro-AppSrv-source1.selector.mapping.CA = mem-channel-1

agent\_foo.sources.avro-AppSrv-source1.selector.mapping.AZ = file-channel-2

agent\_foo.sources.avro-AppSrv-source1.selector.mapping.NY = mem-channel-1 file-channel-2

agent\_foo.sources.avro-AppSrv-source1.selector.default = mem-channel-1

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/019~031资料/027\_将多个agent串联起来的数据里模型怎么配置？/笔记.doc**

# list the sources, sinks and channels for the agent

agent-collector.sources = http-source

agent-collector.sinks = hdfs-sink avro-sink

agent-collector.channels = memory-channel file-channel

agent-collector.sources.http-source.type = http

agent-collector.sources.http-source.bind = localhost

agent-collector.sources.http-source.port = 9000

agent-collector.channels.memory-channel.type = memory

agent-collector.channels.memory-channel.capacity = 1000

agent-collector.channels.memory-channel.transactionCapacity = 100

agent-collector.channels.file-channel.type = file

agent-collector.channels.file-channel.checkpointDir = /usr/local/flume/checkpoint

agent-collector.channels.file-channel.dataDirs = /usr/local/flume/data

agent-collector.sinks.hdfs-sink.type = hdfs

agent-collector.sinks.hdfs-sink.hdfs.path = hdfs://namenode/flume/webdata

agent-collector.sinks.avro-sink.type = avro

agent-collector.sinks.avro-sink.hostname = 192.168.31.111

agent-collector.sinks.avro-sink.port = 44444

# set channel for source

agent-collector.sources.http-source.channels = memory-channel file-channel

# set channel for sink

agent-collector.sinks.hdfs-sink.channel = memory-channel

agent-collector.sinks.avro-sink.channel = file-channel

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/075\_看一下Canal Instance是如何被创建和启动的？/笔记.docx**

《066\_看一下Canal Instance是如何被创建和启动的？》

HDFS HA，NameNode的高可用模式，也是依赖于ZooKeeper来实现的

核心源码和算法可以看一下，Kafka对ZK的使用，Canal对ZK的使用，HDFDS对ZK的使用，从他们的源码层面去看看他们是怎么使用ZK的

高可用，数据零丢失，需要在哪些地方做一些事情

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/076\_Parser如何伪装成MySQL Slave拉取binlog日志的？/笔记.docx**

《067\_Parser如何伪装成MySQL Slave拉取binlog日志的？》

之前踏踏实实的跟着我学习，积累了很深厚的技术功底了，如果你有兴趣，完全可以自己慢慢的一行一行的区看canal的源码

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/073\_初步看看Deployer模块是如何运行起来去部署系统的？/笔记.docx**

《064\_初步看看Deployer模块是如何运行起来去部署系统的？》

mvn clean install -Dmaven.test.skip -Denv=release

canal这个开源项目，他其实是中国人自己写的，是一个非常适合我们去参考的一个开源项目，如果我们是写一个分布式的业务系统，web系统，其实倒是没什么特别的，可以参考自己的maven的课程

把多个项目变成一个maven项目的多个module，然后一起提交到git仓库里去

但是如果我们是要写一个中间件的系统，甚至是一个开源的系统，完全是可以参考canal的源码结构，包括他的maven配置，可以把一个开源中间件系统的多个module放在一起，然后可以参考他的源码目录的结构

可以做到一个中间件系统，一旦打包之后，直接会把所有的源码打包在一起，然后把deployer模块的一些bin目录下的命令和conf目录下的配置文件，打包放在最终的执行包的外层

会给你打成类似于canal-xx-xx-x.tar.gz，解压缩以后就是一堆编译后的class代码在里面，还有就是他的bin目录，conf目录

你直接执行bin目录中的一些shell脚本，就可以直接启动canal，无非也是启动一个Java进程而已

修改源码之后可以打包，然后就可以去部署了，走deployer下面的bin命令，那是shell脚本，他会启动Deployer里的main类

解析配置文件

启动canal server，监听canal client请求

启动canal instance，抓取binlog

监听配置文件的变化

狸猫技术窝上面，现在我也有一些合作，推荐过上面的我一个好朋友出的JVM专栏

Deployer本身就是一个main方法，带有main方法的类被执行之后，直接启动起来的就是JVM进程，在这个JVM进程中可以启动或者执行各种代码模块

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/071\_Canal源码结构说明以及整体架构分析/笔记.docx**

《062\_Canal源码结构说明以及整体架构分析》

讲解一下Canal的源码，剖析一下他的核心运行原理

canal-admin：可视化的有前端界面的，canal管理工作台，上面可以对canal做一些管理，对canal做很多配置和部署的修改，不需要重启系统，直接前端界面做一下修改就可以了，还是比较方便的

client：就是说canal负载去拉取mysql binlog，但是这个拉取到的binlog给谁去消费呢？其实是给canal client去消费，就是你可以写一个java系统去从canal中订阅和拉取mysql binlog

deployer：负责部署一个canal server

instance：代表了一个canal instance，一个canal instance负责伪装成一个mysql的slave，跟一个mysql去进行交互获取其中的binlog

parse（canal.parse.driver、canal.parse.dbsync） + sink + store + meta：他们四个组件构成了一个canal instance：一个canal instance通过driver去跟一个mysql进行连接和通信，获取到的binlog交给dbsync去解析，通过sink去过滤和处理数据，store最终把数据进行存储，比如说放入kafka中，meta负责维护一些元数据这样子

promtheus：负责让canal可以发送统计信息给prometheus监控系统，让我们可以通过prometheus进行可视化的canal集群的监控

protocol：代表的是协议，Java架构中现在最新正在讲解的基于Netty的工业级IM系统的项目实战，如何基于Protobuf去制定各个系统之间的通信协议，也是有一个protocol模块的，负责client和server就是基于protocol里定义的协议进行通信的

server：代表了一个canal server，一般来说是用deployer在一台机器上部署一个canal servre，一个canal server中会包含多个canal instance，每个instance负责从一个mysql实例中拉取binlog过来

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/078\_默认的Store是用来干什么的以及client订阅binlog是如何实现的？/笔记.docx**

《069\_默认的Store是用来干什么的以及client订阅binlog是如何实现的？》

为什么他这里的client要用订阅来形容呢？client和server之间是基于netty实现的长连接，基于这个长连接，client可以向server发送请求，server本身也是可以反向把client订阅的binlog数据给推送回client的

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/072\_画图说明Canal各个组件的协作和交互关系以及整体架构/笔记.docx**

《063\_画图说明Canal各个组件的协作和交互关系以及整体架构》

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/074\_Deployer模块是如何启动Canal Server的？/笔记.docx**

《065\_Deployer模块是如何启动Canal Server的？》

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/077\_Sink到底是用来干什么的以及如何加工数据？/笔记.docx**

《068\_Sink到底是用来干什么的以及如何加工数据？》

sink，主要是用来提供一些扩展点，让你对数据做一些扩展，或者是过滤掉一些数据

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/080\_Canal会丢失数据吗？如何保证数据的零丢失？/笔记.docx**

《071\_Canal会丢失数据吗？如何保证数据的零丢失？》

如果说Canal一旦宕机的话，就会导致数据的丢失，内存级别的MemoryStore，我觉得我们得参考Flume的实现，我们得在Canal中加入一个FileStore，让他把拉取到的binlog数据，都写入本地磁盘文件里去

另外一种实现思路，就是不用FileStore，而是把获取到的每一条binlog都给写入到Kafka里去，我们只要保证说Kafka宕机，直接写入本地磁盘

必须得是把数据写入到kafka之后，才能更新一下对binlog拉取和消费的一个位置，假设canal拉取到了一些binlog数据之后，然后还没来得及写入kafak，直接宕机了，导致一部分binlog丢失了

另外一个备用的canal server接管过来之后必须从zk里加载出来之前最后一条写入kafka的mysql binlog的位置，重新拉取binlog写入kafka里去

Flume、Canal的核心源码的研究，还是有收获的

Flume：SpillableMemoryChannel，改造一些KafkaSource、KafkaSink，SinkGroup去实现一个failover，hdfs、kafka故障了之后，可以把数据写入本地磁盘文件里去暂存起来

Canal：数据直接写Kafka，而且要配合着更新zk里的binlog位置，一旦有数据消费到但是还没写入kafka中，此时canal server宕机了，必须从上一次binlog继续读取发送数据到kafka去；kafka如果宕机，此时可以把数据线写入磁盘文件里去

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/071~080资料/079\_要实现高可用的话，Canal架构上要如何设计？/笔记.docx**

《070\_要实现高可用的话，Canal架构上要如何设计？》

Canal的架构设计，分成两块，第一个的话呢，实际上来说就是高可用，第二块是数据零丢失，高可用

Canal Server自己挂掉，Kafka自己挂掉

Canal Server高可用，在我们学习完ZooKeeper之后，我们会结合ZK来研究一下Canal是如何实现主备自动切换和高可用的

在他进行binlog拉取和解析的过程中所有的元数据都必须存储在zk中，一旦canal server master挂掉了，此时canal server backup自动开始运行，基于zk中存储的元数据，继续拉取binlog其实就可以了

Kafka如果挂掉，你必须得保证Canal可以在磁盘文件里先暂存数据

Canal自己是没有做到的，所以说对我们而言，到时候我们要对Canal做一些二次开发，要让他的写Kafka这块的逻辑，可以实现一个效果，发现Kafka故障宕机之后，他可以自动把数据写入磁盘文件里去

Canal要直接把数据写入Kafka是如何来配置的，看一下那块的源码，改动一下那块的源码就可以了

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/068\_KafkaSink是如何从FileChannel中获取数据以及管理事务的？/笔记.docx**

《059\_KafkaSink是如何从FileChannel中获取数据以及管理事务的？》

其实如果把Channel给做好了之后，那么结合事务机制，我们自己改动一些源码，就可以完全实现就是如果内存放不下了，那么就写磁盘

Sink，优先读内存，然后再读磁盘，磁盘都读完了之后，再回过头来读内存

Sink一次事务，写kafka或者hdfs如果失败了之后，这个时候他本次读取的数据会全部都回到之前的channel的queue里去，以供下次再次读取

KafkaSink，完全可以自己来实现，基于他 的代码来写一套，Kafka如果他故障了，我们可以把数据暂时先写放磁盘里去，后续再次从磁盘里读取再次发送这样子

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/062\_FileChannel的事务机制在写入的时候是如何先暂存内存队列的？/笔记.docx**

《053\_FileChannel的事务机制在写入的时候是如何先暂存内存队列的？》

很多代码细节没仔细去看，没那么多时间，没那么大必要，因为这些底层和基础的技术，在Java架构中已经全部都打通了

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/054\_关于大数据平台部署采用CDH的说明以及Flume源码导入IDEA的说明/笔记.docx**

《045\_关于大数据平台部署采用CDH的说明以及Flume源码导入IDEA的说明》

正儿八经开始做项目之后，大数据平台的部署，hdfs、flume、hbase之类的东西，都是跟业内统一采用cdh来进行部署

hadoop版本，一般来说分成两种，第一种是apache hadoop，Apache下面的开源Hadoop版本，有会一些bug，不稳定，而且他没有提供任何企业级的生产的部署、监控、管理和运维的一些可视化平台

另外一种hadoop，cdh，cloudera hadoop，cloudera，专门做就是把apache hadoop给做一些稳定性的增强，修复一些bug，提升一些性能，增加少量的功能，cloudera manager，cm，非常好用，直接就可以一键部署集群，做版本升级，运维和管理

可视化的集群监控，磁盘、内存、jvm、网络、CPU的负载，还可以看到很多其他的监控信息

cdh hadoop，偏向于企业级一些

hdfs源码，我们直接就是用的apache hadoop的版本，无所谓的，cdh的源码和apache的源码，大体上是一样的，cdh直接就是用apache hadoop的版本来进行封装的。所以我们之之前用apache hadoop来讲hdfs源码是没有问题的

正式进入做项目以后，hdfs和kafka，漏了一个技术，zookeeper，kafka大量的使用了，canal也大量的使用了zookeeper，十一之前，Flume源码和Canal源码给核心源码讲解完毕，十一过后我们可以再用两周的时间，两三周的时间，把ZooKeeper这块先讲一下

核心原理、架构、底层的核心源码、核心算法、分布式系统的一些概念

正式动手来搞一堆虚拟机，来进行我们的大数据平台的部署，Kafka、Canal -> ZooKeeper，技术不讲的话，就导致大家一致对ZooKeeper这个东西，感觉到非常的陌生，重点结合Canal和Kafka对ZooKeeper的一些操作的源码，去分析一下

ZooKeeper在十月份彻底打通，总感觉缺了一块东西

11月份，一个月的时间，两周到三周的时间，就可以把数据采集平台的架构全部给他部署出来，做一定的二次开发，Flume，稍微改动点源码，如何重新编译和打包，版本的flume，xxx-tar.gz，放到cdh里去替换cdh原生带过来的flume的包，就可以掌握如何对cdh中的系统进行源码级的二次开发

canal我们也会做一点二次开发

2019年，快到尾声了，2019年才开始的，2018年底起了个头，一直讲的进度控制的不会太快，比较慢一些，学大数据的同学很少，大部分学大数据的同学都是2019年才入班的，提供了前大半年的课程都是Java的

并不是跟培训机构一样，只是讲怎么用，做一些demo项目，直接从hdfs源码

hdfs源码，二次开发，kafka实战和源码 => 2遍，3遍

打下一个基础，大数据方向的学员而言，2019年跟着我学一年，打基础，Java底层学精通，HDFS、Kafka、数据采集平台、Flume、Canal、ZooKeeper，全部都是到源码级的，进入数据仓库的大项目

数据仓库+YARN+Hive+Spark+作业调度平台+BI平台 + 数据治理 -> 大型的数据仓库的项目给讲完，YARN、Spark，都是到源码级别的

实时这块 + Kylin、Druid、ELK（分析） + 数据中台，大数据平台架构，全部搭建起来，搭建通透

把flume-ng的源码导入到intellij idea里去，同时还会通过maven给你下载很多的依赖，这个时间很长，大概需要一两个小时左右，intellij idea，配置好maven、jdk

canal，很多其他的源码都是这样子的

github地址，intellij idea自动拉取源码，自动下载依赖

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/070\_HDFSSink的分析以及SinkGroup的工作原理/笔记.docx**

《061\_HDFSSink的分析以及SinkGroup的工作原理》

channel，我们会自己写一个SpillableMemoryChannel

改造一些KafkaSink，改造一些KafkaSource，我们会基于SinkGroup的机制来做failover，也就是说一个Sink发现了问题之后，就会换一个Sink来处理，会把数据写入本地磁盘文件里去这样子

自定义Source，从本地磁盘文件里读取数据出来再次进行处理

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/058\_FlumeHTTPServlet是如何处理接收到的HTTP请求的？/笔记.docx**

s

《049\_FlumeHTTPServlet是如何处理接收到的HTTP请求的？》

连jetty server，web服务器，servlet都不懂的话，在这里听这块代码一定会懵，搞大数据的，精通Java技术是一个基础，Java Web技术，分布式Java系统的一些技术，Java底层的一些技术，并发、集合、网络、NIO、Netty

搞大数据的不一定要去懂高并发、海量数据、高性能、高可用、可扩展，Java复杂架构。很多大数据架构师，都是要求说，懂上述复杂Java架构的人选是优先

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/066\_看一下KafkaSink是如何从Channel中获取数据的？/笔记.docx**

《057\_看一下KafkaSink是如何从Channel中获取数据的？》

可以FileChannel和MemoryChannel都可以看一下

在我们的日志采集系统里，SpillableMemoryChannel，是自己研发的，直接就是基于MemoryChannel和FileChannel封装起来来写的

KafkaSink也是我们自己研发的，他都不支持kafka的一些新版本，深入研究过Kafka架构和技术，源码，现在Kafka最新的稳定版本已经是Kafka 2.3.0版本，我们后面也打算直接部署最新的版本

起码也得是Kafka 1.x系列的版本

参考一下Kafka Sink的源码，后续呢自己来写一个Kafka Sink

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/063\_FileChannel的事务提交和回滚机制是如何运行的？/笔记.docx**

《054\_FileChannel的事务提交和回滚机制是如何运行的？》

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/059\_Source是如何将一批events放入Channel中的？/笔记.docx**

《050\_Source是如何将一批events放入Channel中的？》

一种方式是可以去根据条件，根据header去把event路由到指定的几个channel中去；另外一种就是把这个event路由到所有的channel中去

根据source -> channel的事务机制，如果说收到一个请求，全部正常放入channel，事务就会提交，返回一个正常的HTTP Status给Nginx；但是一旦说发生了异常，请求没有全部写入channel中，此时回滚事务，把之前写入channel的events都删除掉，返回一个异常HTTP Status，503，给Nginx

Nginx收到异常之后，就会自动去进行一个重试，把这个请求发送给其他的Flume所在的机器上去

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/069\_KafkaSource是如何从Kafka消费数据然后写入Channel的？/笔记.docx**

《060\_KafkaSource是如何从Kafka消费数据然后写入Channel的？》

cdh上部署的kafka版本就可以了，另外主要是加入一些kafka source

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/067\_KafkaSink是如何在一个事务中从Channel获取数据的？/笔记.docx**

《058\_KafkaSink是如何在一个事务中从Channel获取数据的？》

如果是基于内存的MemoryChannel，正常情况下是往Transaction里的内存队列中写，一旦Commit的话，就是提交到Channel层面的一个内存队列里去，只要说是一个transaction完毕之后，数据就会进入到channel全局的一个内存队列里去

take的时候，直接就是从全局的内存队列里去获取的，poll方法，获取一条数据暂存在事务的take list中，接着就可以去处理这条数据了

可能会导致有部分数据已经写入kafka了，但是这些数据可能会重新放到queue里去，再次会发送一遍这样子，所以说呢，flume他本身是没办法正常数据没有重复，所以我们自己要注意数据重复的一些问题

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/065\_回过头看一下MemoryChannel的实现原理为二次开发自研Channel做准备/笔记.docx**

《056\_回过头看一下MemoryChannel的实现原理为二次开发自研Channel做准备》

写内存队列是非常简单的，其实就是一个简单的内存队列这样子，每个事务都是直接写内存队列而已，直接拿原生的内存队列来改造就可以了

一次transaction会有多条数据写入到Transaction内部的内存队列里去，然后commit的时候，就是把这个Transaciton里的数据都加入到MemoryChannel他的一个大的内存队列里去

如果拿两个channel来封装，应该是优先写入内存队列里去，如果写满了就写FileChannel里去，Sink在获取数据的时候，应该是从MemoryChannel开始来获取，如果内存里的数据后去完毕了之后

就可以从FileChannel里去获取数据，直到磁盘里的数据都获取完毕了，接着再继续从内存里来获取数据

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/060\_Flume的FileChannel是如何基于纯磁盘存储数据的？/笔记.docx**

《051\_Flume的FileChannel是如何基于纯磁盘存储数据的？》

HTTP Source跟其他的Source，唯一不一样的一点，其实就是获取数据方式的不同，他只不过是利用了内嵌的jetty server去监听端口号，接收Http请求，用servlet去处理，把每个请求搞成一堆event，每个event就是一个json格式的用户行为日志

一般是不会直接用纯内存的，如果纯内存队列要是满了，就会直接报错了

FileChannel，SpillableMemoryChannel

但凡你的数据写入FileChannel对应的磁盘文件之后，此时一旦说这个Flume Agent他宕机了，接下来Flume如果说重启的话，FileChannel一定会做一个replay，把磁盘里的数据重新恢复出来，继续提供你的Sink去获取数据写入存储

初步的看了一下FileChannel的一些源码，他的使用是完全跟Transaction机制配套起来的，事务的开启在干什么，如何通过事务去写入数据，事务如何提交，事务如何回滚，要结合事务机制来看看FileChannel的工作原理

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/064\_SpillableMemoryChannel是如何在内存队列满了之后写磁盘的？/笔记.docx**

《055\_SpillableMemoryChannel是如何在内存队列满了之后写磁盘的？》

他的意思就是说，这个Channel作为官方而言，仅仅建议用在实验环境，不建议用在生产环境上，我们对源码的阅读都没太大的意义了，flume 1.6.0里面，其实他的源码层面明显是也没把这块东西写的太完善

这个东西，我们基本上得自己去写了，我们得基于MemoryChannel和FileChannel，自己写一个生产级可以使用的SpillableMemoryChannel出来，基于MemoryChannel和FileChannel进行封装

这块，从源码里就可以看到，这个东西是第一个要进行源码级二次开发的地方，我们下一讲，应该是把MemoryChannel给看一下源码

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/057\_Flume的HTTP Source是如何启动Jetty Server以及绑定Servlet的？/笔记.docx**

《048\_Flume的HTTP Source是如何启动Jetty Server以及绑定Servlet的？》

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/056\_Flume的HTTP Source是如何监听端口以及接收HTTP请求的？/笔记.docx**

《047\_Flume的HTTP Source是如何监听端口以及接收HTTP请求的？》

我们这套架构里，会把最最常用的一些source、channel、sink都用到，源码都会讲一遍的，http source，kafka source，spooldir source

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/055\_Flume整体源码结构说明以及源码分析思路的说明/笔记.docx**

《046\_Flume整体源码结构说明以及源码分析思路的说明》

你在启动flume的时候，说白了就是在启动flume-ng-node里的一个带main方法的Application类，包括HDFS、Kafka、Flume、HBase，但凡是用JVM类的语言编写的，其实系统启动就是一个main方法

java命令，启动一个JVM进程，从main方法开始运行

Application启动，一定会干几件事情，main方法执行的时候，一定会解析我们配置的一些数据流，解析完了以后放到内存里来，根据我们的配置，去创建对应的组件，后面就是坐等source组件去获取数据

获取到数据之后就交给channel，sink再从channel里获取数据写入对应的存储里去

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/054~070资料/061\_结合Channel的事务机制来研究FileChannel写磁盘的工作原理/笔记.docx**

《052\_结合Channel的事务机制来研究FileChannel写磁盘的工作原理》

FileChannel对应的事务机制，就是FileBackedTransaction，每次收到一个请求对应了一个事务，然后一个事务默认最多是1万条数据，一个请求最多带1万条用户行为日志，否则再多就不支持了

事务机制写入的时候，是如何写入到内存队列里去的

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/049\_对Canal原生支持的可视化监控平台了解一下/笔记.docx**

《049\_对Canal原生支持的可视化监控平台了解一下》

可视化监控平台：Zabbix、Ganglia、Open-Falcon、Prometheus，现在包括微服务架构都推荐在引入Prometheus

干脆就基于Prometheus来进行可视化监控得了

监控平台，他都是会自己发送请求给对应的要监控的系统去采集他的metrics的数据，或者是尝试从指定的机器上采集对应的数据指标

直接基于Prometheus去进行Flume、Canal、Kafka、HDFS的可视化的监控，因为那些系统内部都有一些Metrics，只要把Metrics采集到Prometheus里面去，就可以可视化的看到各个系统运行的情况

基于他去配置一些报警，对一些关键指标超过一定阈值之后，发送短信、邮件给到你

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/053\_初步分析一下Canal内部的运行原理/笔记.docx**

《053\_初步分析一下Canal内部的运行原理》

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/045\_基于Flume架构原理和核心功能实现数据采集平台的回顾/笔记.docx**

《045\_基于Flume架构原理和核心功能实现数据采集平台的回顾》

Flume核心架构原理搞定了，心里就完全有数了，到底如何来实现数据采集平台，用户行为日志采集这块到底怎么来做

做一些简单的日志采集，非常简单的，很多大数据培训机构，基本上来说都会简单带一带Flume使用，讲的不会太深，做一个demo级别的日志采集

中大型公司的日志采集平台，BAT大厂，架构都是非常工业级的，考虑到很多的因素。中型互联网公司，比如说一些中型的视频网站，千万级别，日活百万级别，人家的日志采集平台设计的都是工业级的

高可用、数据零丢失、高吞吐、可扩展

LVS，多机器部署，流量均匀分发到各个机器的呢？我们对外提供的公司的一个后台系统，都是走一个域名的，DNS里配置一个域名的解析可以让他来做负载均衡，一个域名的请求，均匀的分发给多个IP地址

多台LVS机器部署的地方

二次开发：改造KafkaSink的源码（控制Kafka故障的开关），改造HDFSSink的源码（控制HDFS故障的开关），自定义容灾备用Source

二次开发：改造Canal写Kafka的源码，支持故障的时候写本地磁盘

Flume，Channel采用基于磁盘文件的，让数据固化在磁盘里，不至于Agent进程宕机丢失内存里的数据

二次开发：KafkaSource+KafkaSink/HDFSSink，关闭自动提交offset，KafkaSink/HDFSSink投递成功之后通知KafkaSource手动提交offset；另外一种方式，KafkaSource消费数据之后成功写入FileChannel，手动提交offset

高吞吐：数据流转的性能要高；尽量采用Batch的方式；日志采集，可以允许丢失部分数据，丢一点数据影响并不是太大

可扩展：基于Kafka中间做一个分层和解耦

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/046\_Canal到底是个什么东西？/笔记.docx**

《046\_Canal到底是个什么东西？》

MySQL binlog

玩儿大数据的时候，从MySQL里采集数据，其实是怎么采集的呢？采集的是MySQL的binlog，日志

Id name age

leo 21

INSERT INTO xxx VALUES xxx

Binlog，日志：insert语句，（1，leo，20）

UPDATE xxx SET xxx ->

Binlog，日志：update语句，（1，leo，21）

采集过来的是MySQL的增删改的日志，如何把这些日志回放，还原成类似MySQL里的一个数据情况，那就是你自己要处理的事情了

[https://github.com/alibaba/canal/wiki](media/image4.png)

MySQL支持主从架构，MySQL主库，MysQL从库，从库就是一个Slave，从库会不停的从主库去拉取数据，把主库的数据同步过来

读写分离，高可用架构

写请求往主库去写，读请求从从库去读

主库宕机，从库直接转换成主库，对外提供服务

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/047\_初步对Canal的产生背景和工作原理做一个介绍/笔记.docx**

《047\_初步对Canal的产生背景和工作原理做一个介绍》

MySQL binlog，采集，增量采集

MySQL增量数据，不断新产生的增删改的日志，canal，增量采集

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/048\_初步看一看Canal是怎么跟Kafka配合起来使用的/笔记.docx**

《048\_初步看一看Canal是怎么跟Kafka配合起来使用的》

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/050\_不能漏过的一个点：Canal的管理工作台/笔记.docx**

《050\_不能漏过的一个点：Canal的管理工作台》

Canal管理工作台，需要在数据库里保存一些你对canal管理的一些配置信息之类的东西，所以他是需要数据库来存储的

先用他的sql脚本在一个mysql库里初始化表结构

启动一个canal admin，他会连接到一个mysql库，基于数据库管理你的canal的管理配置信息，可以通过访问8089端口连接到canal可视化管理工作台

如果你在canal admin里修改了一些canal的配置，canal如何感知到呢？

同时在canal中配置一下canal admin的数据库的地址，canal admin如果修改了一些配置会同步到数据库里去，canal自己本身就会从数据库里加载修改后的配置

未必说你必须立马就把所有的功能都亲手实验一遍

学习任何一门新技术，推荐的是什么，先把官方文档过两遍，第一遍过的时候就可以跟着画一些图，写一份笔记，在这个笔记里，就把这个技术架构原理，核心原理，核心功能，都简单的写一下

很多官方文档的东西第一遍是看不懂的

第二遍，你已经对这个技术有一个比较高屋建瓴的了解了，就可以把每个细节都看明白，主要就是画图+写一些笔记

技术学习的境界

了解一个技术：随便看几篇博客，或者公众号上发的文章，有一个简单的认识

熟悉一个技术：官方文档过两遍，稍微搭建和实操几个简单的demo，整体架构，核心原理，主要功能和特性，都搞清楚，算做是熟悉

精通一个技术：就是把技术在生产环境和项目里大量的运用，积累很多的实践经验，同时还阅读过其核心源码，能基于源码做一些问题的排查，甚至可以改造源码。这个技术在国内的一个专家

技术的开源贡献者：就是你直接可以对开源技术贡献一些核心的代码，contributor甚至committer。就是这个技术在国内的一个顶级专家

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/044\_Flume整体架构原理的回顾/笔记.docx**

《044\_Flume整体架构原理的回顾》

两周的时间，把Flume整体架构的原理，他对应的核心组件和用法，过了一遍，看了一下Flume的架构和功能，对我们设计出来的数据采集平台的架构，落地的时候，大致应该如何落地

承上启下，Flume回顾，Canal

不知道怎么看官网，实际上，我们带着大家在过官网的时候，做的一些总结，画的一些图，好好去学习我带大家看官网的技巧，授人以鱼不如授人以渔

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/052\_Canal HA双机高可用架构是怎么做到的？/笔记.docx**

《052\_Canal HA双机高可用架构是怎么做到的？》

# **File Path: /Users/xiaotingting/Downloads/学习资料/07\_每日TB量级的数据采集平台架构实战unzip/044~053资料/051\_了解一下Canal官方文档中的各种配置项的详细解释/笔记.docx**

《051\_了解一下Canal官方文档中的各种配置项的详细解释》

Canal会基于内存来缓存一些数据，这里面是不是有一个数据丢失的风险呢？如果说有一些数据挤压在canal的内存里，还没来得及写入kafka中，此时canal宕机了，是不是数据就会丢失了呢？

针对数据零丢失这个问题，到时候我们需要阅读canal的核心源码，可能要对核心源码做一些二次开发和改造，保证说他的数据是零丢失的

二次开发：让canal支持分库+分表的环境，原生是支持分库的环境