

[架构](#) [前端](#) [编程语言](#) [云计算](#) [AI](#) [开源](#) [技术管理](#) [运维](#) [区块链](#) [新基建](#) [云原生](#) [产品](#) [安全](#) [Google Cloud](#) [全部频道](#)

# 腾讯云 Serverless 衔接 Kafka 上下游数据流转实战

作者：许文强

发布于：2020 年 8 月 22 日 10:00

[七牛云 CEO 许式伟，首次分享完整架构经验，带你解读常见业务场景的架构范式，戳此学习 >>](#)

## 01 Tencent Cloud Kafka 介绍

Tencent Cloud Kafka 是基于开源 Kafka 引擎研发的适合大规模公有云部署的 Cloud Kafka。是一款适合公有云部署，运维的分布式的、高可靠、高吞吐和高可扩展的消息队列系统。它 100% 兼容开源的 Kafka API，目前主要支持开源的 0.9, 0.10, 1.1.1, 2.4.2 四个大版本，并提供向下兼容的能力。

目前 Tencent Cloud Kafka 维护了近万节点的集群，堆积数据达到了 PB 级。是一款集成了租户隔离、限流、鉴权、安全、数据监控告警、故障快速切换、跨可用区容灾等等一系列特性的，历经大流量检验的、可靠的公有云上 Kafka 集群。

## 02 什么是数据流转

CKafka 作为一款高吞吐，高可靠的消息队列引擎。需要承接大量数据的流入和流出，数据流动的这一过程我们称之它为数据流转。而在处理数据的流入和流出过程中，会有很多成熟丰富的开源的解决方案，如 Logstash, Spark, Flink 等。从简单的数据转储，到复杂的数据清洗，过滤，聚合等，都有现成的解决方案。

如图所示，在 Kafka 上下游生态图中，CKafka 处于中间层，起到数据聚合，流量削峰，消息管道的作用。图左和图上是数据写入的组件概览，图右和图下是下游流式数据处理方案和持久化存储引擎。这些构成了 Kafka 周边的数据流动的生态。



QCon+案例研习社限时7.5折特  
365天畅学200+大厂前沿实战案

### 推荐阅读

分布式安全：上百个分布式节点  
现“内奸”吗？

2020 年 1 月 20 日

微博 Service Mesh 实践之路（上  
2018 年 11 月 10 日

对话 Confluent CTO：针对 Kaf  
式流数据查询引擎 KSQL 技术解  
2017 年 11 月 1 日

Lambda plus: 云上大数据解决方  
2019 年 6 月 10 日

分布式体系结构之集中式结构  
上，万人在下  
2019 年 10 月 11 日

重磅开源 KSQL：用于 Apache H  
数据 SQL 引擎  
2017 年 8 月 29 日

阿里、百度、腾讯都选择 Flink，  
什么魔力？  
2019 年 4 月 24 日

### 相关会议



240个大厂实战案例分享  
[立即参与 >>](#)

数据中台建设

大前端趋势

人工智能应用

高可用架构实践



图 1: Kafka 上下游生态图

技术大会极客时间极客大学团队学习高端会员研究报告

InfoQ

首页直播HOT专题迷你书HOT话题免费视频技术博客

请输入关键词搜索

成为作者



图 2: 流式计算典型数据流动示意图

而从学习成本，维护成本，金钱成本，扩缩容能力等角度来看，这些开源方案还是有欠缺的。怎么说呢？开源方案的缺点主要在于如下三点：

- 学习成本
- 调优、维护、解决问题的成本
- 扩缩容能力

以 Logstash 为例，它的入门使用学习门槛不高，进阶使用有一定的成本，主要包括众多 release 版本的使用成本，参数

服务治理架构演化

特惠预售中>>

## 订阅每周精

你将获得

- 资深编辑编译的全球 IT 要闻
- 一线技术专家撰写的实操技术干货
- InfoQ 出品的课程和线下活动折扣

请输入邮箱

立即订阅>

## 相关厂商

在这里，探索如何创建出色的游戏体验  
[立即参与 >>](#)

2020 Google游戏开发者峰会

Google jeRank算法 研报

解决方案，助力开发者打造出色的游戏体验

交流分享，更多GoogleCloud资讯

## 大厂实战案例

华为云区块链服务技术决策和实践经验

刘再耀 | 华为云区块链服务产品售前

[立即下载](#)

从重新认识前端渲染开始，聊聊前端性能监控及优化实践

李季骏 | 小红书 社区前端工程师

[立即下载](#)

嵌入式深度学习框架研发与部署

李永会 | 百度 多模搜索部资深工程师

[立即下载](#)

调优和故障处理成本，后续的维护成本（进程可用性，单机的负载处理）等。如果用流式计算引擎，如 spark 和 flink，其虽然具有分布式调度能力和即时的数据处理能力，但是其学习门槛和后期的集群维护成本，将大大提高。

我们来看 Serverless Function 是怎么处理数据流转的。如图所示，Serverless Function 运行在数据的流入和流出的处理层的位置，代替了开源的解决方案。Serverless Function 是以自定义代码的形式来实现数据清洗，过滤，聚合，转储等能力的。它具有学习成本低，无维护成本，自动扩缩容，按量计费等优点。



图 3: Serverless Function 实现低成本数据流转

接下来我们来看一下 Serverless Function 是怎么实现数据流转的，并且了解一下其底层的运行机制及其优势。

## 04 Serverless Function 实现数据流转

首先来看一下怎么使用 Serverless Function 实现 Kafka To Elasticsearch 的数据流转。下面以 Function 事件触发的方式来原因 Function 是怎么实现低成本的数据清洗，过滤，格式化，转储的：

在业务错误日志采集分析的场景中，会将机器上的日志信息采集并发送到服务端。服务端选择 Kafka 作为消息中间件，起到数据可靠存储，流量削峰的作用。为了保存长时间的数据（月，年），一般会将数据清洗，格式化，过滤，聚合后，存储到后端的分布式存储系统，如 HDFS，HBASE，Elasticsearch 中。

以下代码段分为三部分：数据源的消息格式，处理后的目标消息格式，功能实现的 Function 代码段

- 源数据格式：

```
1  {
2      "version": 1,
3      "componentName": "trade",
4      "timestamp": 1595944295,
5      "eventId": 9128499,
6      "returnValue": -1,
7      "returnCode": 101103,
8      "returnMessage": "return has no deal return error[错误: 缺少 **c 参数][seqId:u3Becr8iz*]",
9      "data": [],
10     "seqId": "@kibana-highlighted-field@u3Becr8iz@/kibana-highlighted-field@*"
11 }
```

复制代码

- 目标数据格式：

 复制代码

```

1  {
2      "timestamp": "2020-07-28 21:51:35",
3      "returnCode": 101103,
4      "returnError": "return has no deal return error",
5      "returnMessage": " 错误: 缺少 **c 参数 ",
6      "requestId": "u3Becr8iz*"
7  }

```

- Function 代码

Function 实现的功能是将数据从源格式，通过清洗，过滤，格式化转化为目标数据格式，并转储到 Elasticsearch。代码的逻辑很简单：CKafka 收到消息后，触发了函数的执行，函数接收到信息后会执行 convertAndFilter 函数的过滤，重组，格式化操作，将源数据转化为目标格式，最后数据会被存储到 Elasticsearch。

 复制代码

```

1  #!/usr/bin/python
2  # -*- coding: UTF-8 -*-
3  from datetime import datetime
4  from elasticsearch import Elasticsearch
5  from elasticsearch import helpers
6
7  esServer = "http://172.16.16.53:9200" # 修改为 es server 地址 + 端口 E.g. http://172.16.16.53:9200
8  esUsr = "elastic" # 修改为 es 用户名 E.g. elastic
9  esPw = "PW123" # 修改为 es 密码 E.g. PW2312321321
10 esIndex = "pre1" # es 的 index 设置
11
12 # ... or specify common parameters as kwargs
13 es = Elasticsearch([esServer],
14                     http_auth=(esUsr, esPw),
15                     sniff_on_start=False,
16                     sniff_on_connection_fail=False,
17                     sniffer_timeout=None)
18
19 def convertAndFilter(sourceStr):
20     target = {}
21     source = json.loads(sourceStr)
22     # 过滤掉 returnCode=0 的日志
23     if source["returnCode"] == 0:
24         return
25     dateArray = datetime.datetime.fromtimestamp(source["timestamp"])
26     target["timestamp"] = dateArray.strftime("%Y-%m-%d %H:%M:%S")
27     target["returnCode"] = source["returnCode"]
28     message = source["returnMessage"]
29     message = message.split(" ")[0]
30     errorInfo = message[0].split("(")
31     target["returnError"] = errorInfo[0]
32     target["returnMessage"] = errorInfo[1]
33     target["requestId"] = message[1].replace("]", "").replace("seqId:", "")
34     return target
35
36
37 def main_handler(event, context):
38     # 获取 event Records 字段并做转化操作 数据结构 https://cloud.tencent.com/document/product/583/17530
39     for record in event["Records"]:
40         target = convertAndFilter(record)
41         action = {
42             "_index": esIndex,
43             "_source": {
44                 "msgBody": target # 获取 Ckafka 触发器 msgBody

```

```
45         }
46     }
47     helpers.bulk(es, action)
48     return ("successful!")
```

看到这里，大家可能会发现，这个代码段平时是处理单机的少量数据的脚本是一样的，就是做转化，转储，很简单。其实很多分布式的系统做的系统从微观的角度看，其实就是做的这么简单的事情。分布式框架本身做的更多的是分布式调度，分布式运行，可靠性，可用性等等工作，细化到执行单元，功能其实和上面的代码段是一样的。

从宏观来看，Serverless Function 做的事情和分布式计算框架 Spark, Flink 等做的事情是一样的，都是调度，执行基本的执行单元，处理业务逻辑。区别在于用开源的方案，需要使用方去学习，使用，维护运行引擎，而 Serverless Function 则是平台来帮用户做这些事情。

接下来我们来看 Serverless Function 在底层是怎么去支持这些功能的，来看一下其底层的运行机制。如图所示：

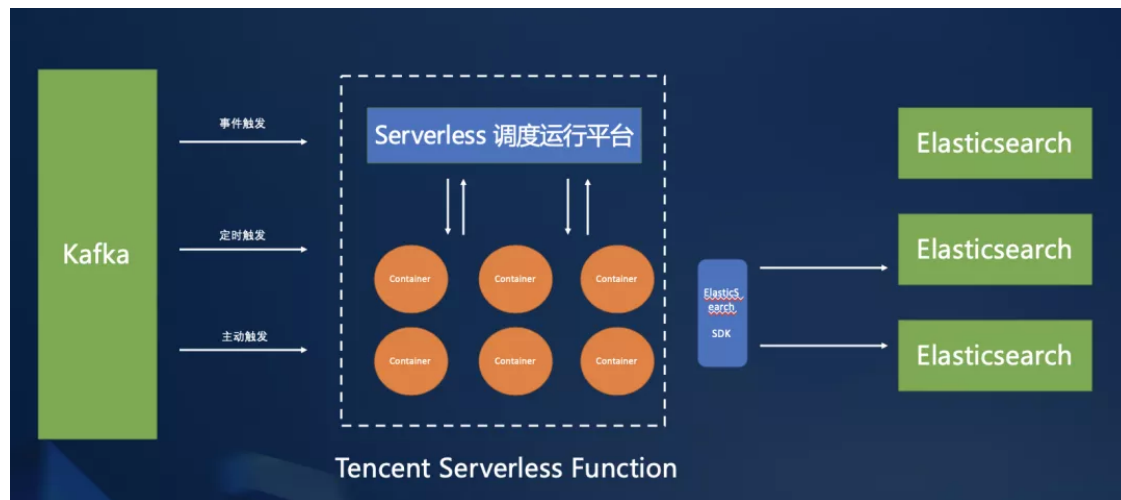


图 4: Serverless Function 实现数据流转原理解析

Function 作为一个代码片段，提交给平台以后。需要有一种触发函数运行的方式，目前主要有如下三种：事件触发，定时触发，主动触发。在上面的例子中，我们是以事件触发为例的。当消息提交到 Kafka，就会触发函数的运行。此时 Serverless 调度运行平台就会调度底层的 Container 并发去执行函数，并执行函数的逻辑。此时关于 Container 的并发度是由系统自动调度，自动计算的，当 Kafka 的源数据多的时候，并发量就大，当数据少的时候，相应的就会较少并发数。因为函数是以运行时长计费的，当源消息数据量少的时候，并发量小，自然运行时长就少，自然所需付出的资金成本就降下来。

在函数执行过程当中，函数的可靠性运行，自动扩缩容调度，并发度等都是用户不需要关心的。用户需要 Cover 的只是函数代码段的可运行，无 BUG。这对于研发人员的精力投入成本就降低很多。

值得一谈的是，在开发语言方面，开源方案只支持其相对应的语言，如 Logstash 的嵌入脚本用的是 ruby，spark 主要支持 java，scala，python 等。而 Serverless Function 支持的是几乎业界常见到的开发语言，包括不限于 java，golang，python，node JS，php 等等。这点就可以让研发人员用其熟悉的语言去解决数据流转问题，这在无形中就减少了很多代码出错和出问题的机会。

## 05 Serverless Function 在数据流转场景的优势

下面我们来统一看一下 Serverless Function 和开源的方案的主要区别及优势。如图 5 所示，和开源方案相比。在非实时的数据流转场景中，Serverless Function 相对现有的开源方案，它具有的优势几乎是压倒性的。从功能和性能的角度，

它在批式计算 (非实时) 的场景中是完全可以满足的。但是它相对开源方案在学习成本，运维成本几乎可以忽略，其动态扩缩容，按需付费，毫秒级付费对于资金成本的投入也是非常友好的。

	开源方案	Serverless Function
语言成本	开源方案对应的语言	不限语言
学习成本	学习开源方案	无
分布式调度	需要学习了解开源方案的分布式调度方案	不需要
维护成本	需要持续维护集群稳定性及可用性	极低，几乎没有
成本	人工评估集群规模	按需付费
动态扩缩容	无	有

图 5: Serverless Function 对比现有开源方案的优势

用一句话总结就是：Serverless Function 能用一段熟悉的语言编写一小段代码去衔接契合流式计算中的数据流转。

## 06 Serverless Function 在批式计算场景的展望

随着流式计算的发展，慢慢演化出了批量计算 (batch computing)、流式计算 (stream computing)、交互计算 (interactive computing)、图计算 (graph computing) 等方向。而架构师在业务中选择批式计算或者流式计算，其核心是希望按需使用批式计算或流式计算，以取得在延时，吞吐，容错，成本投入等方面的平衡。在使用者看来，批式处理可以提供精确的批式数据视图，流式处理可以提供近实时的数据视图。而在批式处理当中，或者说在未来的批式处理和流式处理的底层技术的合流过程中，Lambda 架构是其发展的必然路径。

Serverless Function 以其按需使用，自动扩缩容及近乎无限的横向扩容能力给现阶段的批式处理提供了一种选择，并且在未来批流一体化的过程中，未来可期。

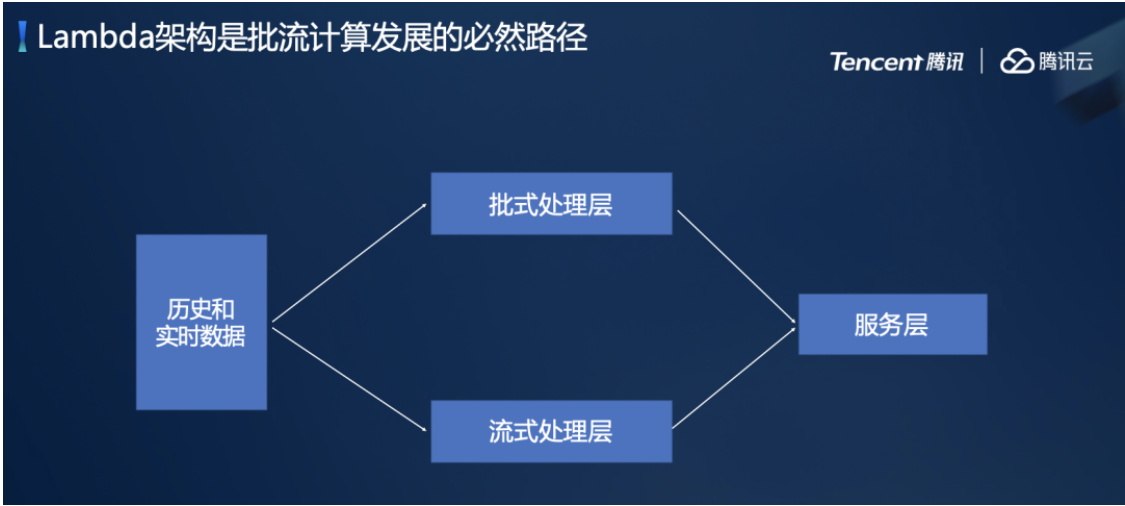


图 6: 批式处理和流式处理

作者介绍：

许文强，腾讯云 Ckafka 核心研发，精通 Kafka 及其周边生态。对 Serverless、消息队列等领域有较深的理解。专注于 Kafka 在公有云多租户和大规模集群场景下的性能分析和优化、及云上消息队列 serverless 化的相关探索。



本文转载自公众号腾讯云中间件（ID：gh\_6ea1bc2dd5fd）。

原文链接：

腾讯云 Serverless 衔接 Kafka 上下游数据流转实战



阅读数：924      发布于：2020 年 8 月 22 日 10:00  
文章版权归极客邦科技 InfoQ 所有，未经许可不得转载。

 [开源](#)，[运维](#)，[架构](#)，[腾讯](#)，[腾讯云](#)，[Serverless](#)

更多 开源、运维、架构 相关课程，可下载【极客时间】App 免费领取 >

## 评论

快抢沙发！虚位以待

发布

暂无评论

## 更多内容推荐

### 我们为什么从 Kafka 迁移至 Pulsar？

Apache Pulsar迁移实践。

2020 年 4 月 29 日

### 携程基于 Storm 的实时大数据平台实践

本文讲解了携程在实时数据平台的一些实践，按照时间顺序来说明我们是怎么一步一步构建起这个实时数据平台的，目前有一些什么...

2016 年 10 月 9 日

### 从 Chukwa 到 Keystone：Netflix 的数据流水线演进

2015 年 12 月，Netflix 新的数据流水线 Keystone 上线。本文将介绍近年来 Netflix 数据流水线演进的 3 个阶段。

2016 年 2 月 18 日

### 下一代微服务架构 Service Mesh

说到Service Mesh，在如今的微服务领域可谓是无人不晓。那么究竟什么是Service Mesh？Service Mesh是如何实现的？

2018 年 11 月 6 日

### 云上大数据：云计算遇上大数据，为什么堪称天作之合？

云上大数据服务的出现和发展，使得在云上存储、处理和查询大数据变得简单而高效，也把云计算的计算存储分离特性体现得淋漓尽...

2020 年 4 月 1 日

### 如何基于 Apache Pulsar 和 Spark 进行批流一体的弹性数据处理？

本文介绍了以 Pulsar 做流数据平台，使用 Spark 进行批流一体数据处理的编程实践。

2019 年 6 月 20 日

## 大数据周报第 8 期：Apache Beam 将统一大数据平台的开发

Apache Beam将统一大数据平台的开发

2016 年 5 月 9 日

## MapReduce 后谁主沉浮：怎样设计下一代数据处理技术？

十年前，如果你是在主持大规模数据处理技术的优化，你会怎么设计它，从而带领下一个十年的技术革新？

2019 年 4 月 19 日

## 分布式计算模式之 Stream：一门背锅的艺术

我与你介绍了流计算的工作原理，并分析了Storm的核心组件，以加深你对流计算的理解。

2019 年 10 月 28 日

## Apache Beam 实战指南 I 玩转 KafkaIO 与 Flink

本文是Apache Beam实战指南系列文章的第二篇内容，将重点介绍 Apache Beam与Flink的关系，对Beam框架中的KafkaIO和Flink源...

2018 年 9 月 12 日



促进软件开发及相关领域知识与创新的传播

### 活动大本营

更多精彩活动持续更新

### InfoQ

关于我们

我要投稿

合作伙伴

加入我们

关注我们

### 联系我们

内容投稿：editors@geekbang.com

业务合作：hezuo@geekbang.com

反馈投诉：feedback@geekbang.com

加入我们：zhaopin@geekbang.com

联系电话：010-64738142

地址：北京市朝阳区叶青大厦北园

### InfoQ 近期会议

深圳 全球架构师峰会 09月11-12日上海 全球人工智能与机器学习技术大会 09月24-25日北京 全球软件开发大会 10月15-17日北京 全球大前端技术大会 11月24-25日