

# 基于OpenResty/Kong构 建边缘计算平台

刘洋([k@polaristech.io](mailto:k@polaristech.io))

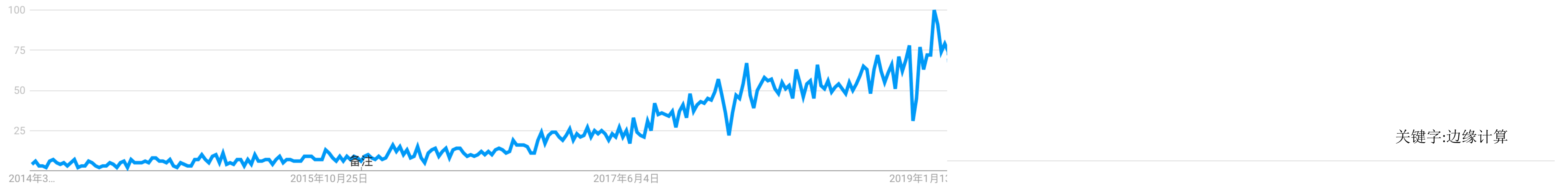
2019-03

# 边缘计算与边缘节点

边缘计算是一种将主要处理和数据存储放在网络的边缘节点的分布式计算形式。-ISO/IEC JTC1/SC38

“边缘”相对概念，指从数据源到计算中心路径之间的任意计算、存储和网络资源。

相对传统“中心”计算：数据上传到数据中心，通过中心的资源完成存储和计算，但随着万物互联时代的到来，智慧城市、智能制造、智能交通、智能家居，5G时代、宽带提速、IPv6的不断普及，导致数百亿的设备接入网络，在网络边缘产生ZB级数据，传统计算难以满足物联网时代大带宽、低时延、大连接的诉求



关键字:edgecomputing



# 边缘计算与边缘节点

关于边缘计算，有很多定义的方式。我们首先定义**边缘节点**：非数据中心的计算资源，都是边缘节点。需要注意的是，作为边缘计算平台，不仅仅由边缘计算节点组成，也包含数据中心的节点。在实际中，我们遇到了如下的边缘节点**场景**：

- 部署在楼宇、园区等
- 部署在基站接入位置
- 部署在监控设备附近
- 家庭的数据聚集位置
- 移动设备所装载的PC，比如车、船等
- 应用交付与加速，比如CDN

# 边缘节点的环境和特点

相比于基于数据中心的“云计算”，边缘计算所在的**环境是边缘计算的**最大前提****。数据中心通常是可靠环境，包括电力、网络、温度等，是**可靠**计算环境；边缘节点处于**不可靠**计算环境：

- ✓ 电力并非总有可靠保障
- ✓ 网络并非高带宽、低延时；网络是不稳定的，带宽没有保障，延迟没有保障
- ✓ 物理环境缺乏统一的条件，包括温度、湿度、气压等
- ✓ 算力不统一。数据中心的计算资源，通常在处理器、内存、网络、磁盘等，有明确的标准和等级；但是边缘计算的算力，可能是基于X86的PC计算平台，也可能是一些基于arm的计算平台
- ✓ 连接方式以TCP/IP为主，包括基于移动通信的4G和WIFI

# 为什么基于Resty/Kong

基于数据中心的“**云计算**”是以“**计算资源**”为中心的；边缘计算本质是以“**网络资源**”为中心的，是**面向网络**的计算。边缘节点的特征需要边缘节点既可以适应高能力的计算环境（可靠的服务器），也需要可以适应其他条件，比如低功耗PC等。那么为什么我们选择Resty和Kong为核心来构建边缘计算平台？

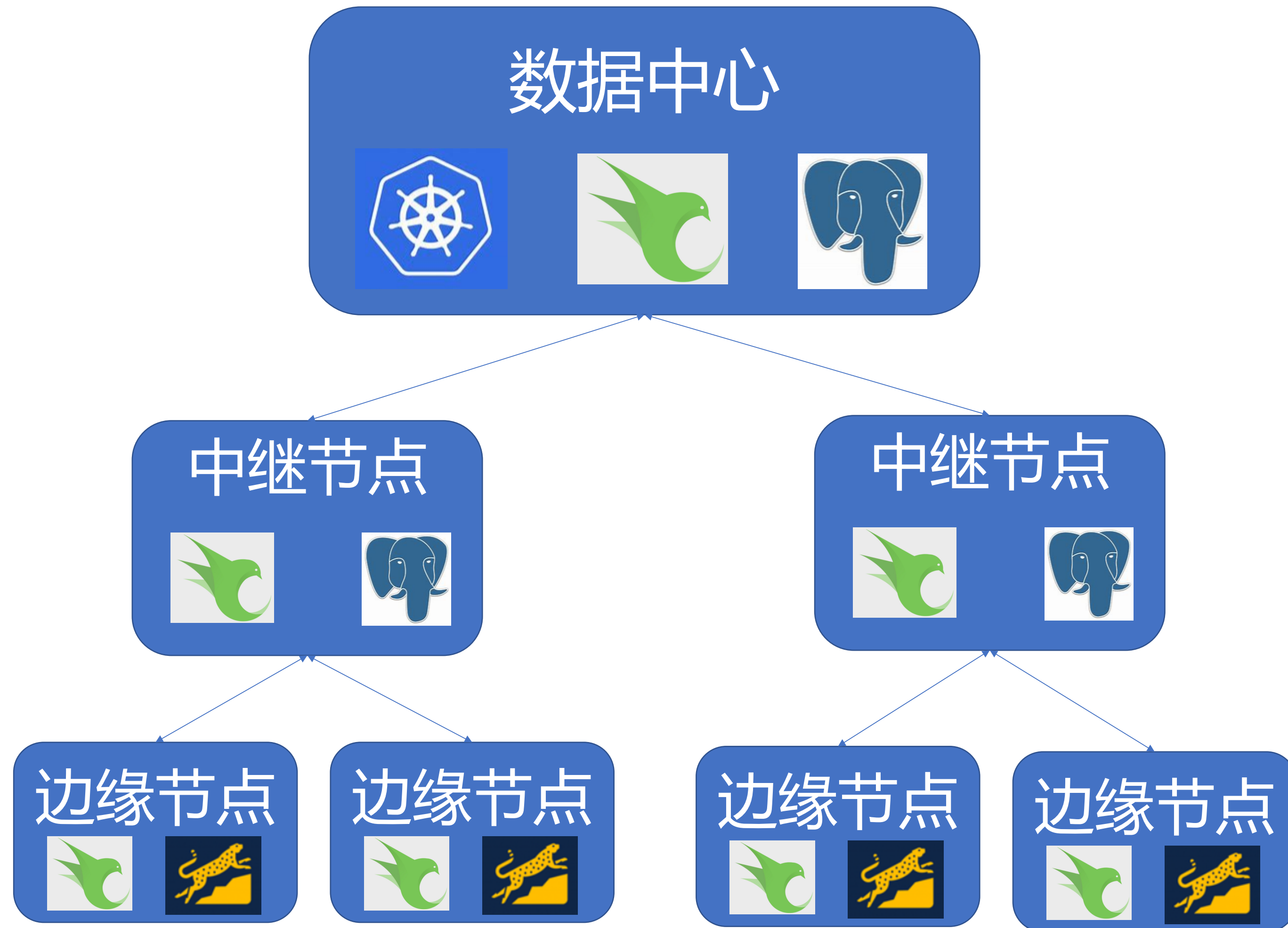
- Resty是目前最适合的**面向应用(Application)网络的计算框架**。小巧、灵活、可靠、高效稳定的网络处理能力。几乎所有可以运行Linux服务器程序的硬件，都可以运行Resty，而且都是高效和稳定的
- Resty是可以扩展的。数据到达边缘节点后，Resty可以把数据和各种常见的计算框架结合起来
- Kong可以看成是Resty领域里边的Struts(类比Java)或者Rails(类比Ruby)，并且提供了简单可靠的多节点配置同步能力。进一步提供了简化的开发框架，可以快速针对应用网络数据做各种处理；高效的开发效率和可靠的执行效率及稳定性
- Resty/Kong除了适应边缘节点环境，同样适用各种级别数据中心环境。在数据被最终存储之前，Resty/Kong可以构建**整个应用数据的传输链路**，并完成各个环节的计算任务
- Resty/Kong很“皮实”，基于Resty/Kong组成的计算节点，可以做到“**RESET即恢复功能**”

# 除了Resty/Kong还用了什么

在Resty/Kong之外，我们主要采用了如下几种开源软件构建边缘计算平台：

- **CentOS**。包括在x86平台和ARM平台
- **PostgreSQL**。包括数据中心节点和中继节点。PostgreSQL丰富的扩展能力，标准的接口，很小的footprint是我们选择它的根本原因：可以存储时序数据；可以存储和检索json；可以用SQL；在给定的硬件条件下，足够的快
- **RocksDB**。主要部署在边缘节点，用于以k/v形式缓存数据，包括上行和下行数据
- 基于**Nodejs**的调度程序。**调度程序是边缘计算的核心能力**。Nodejs丰富的SDK，很小的启动资源，灵活的编程模型，高效的网络处理能力，以及脚本语言适合调试的特点，是我们选择Nodejs的原因
- **Kubernetes**。用于数据中心的计算资源管理，提供多租户的数据中心计算能力

# 整体架构



➤ 数据中心。提供整体管理系统(BSS/OSS)；多租户的运算能力。数据中心节点采用标准X86服务器

➤ 中继节点。在边缘节点和数据中心之间，上行和下行数据通过一层或者多层的中继节点中转和处理。中继节点采用标准X86服务器

➤ 边缘节点。边缘节点用于对终端做下行流量发放，以及上行流量第一次汇聚和处理

# 我们做了什么

整体方案核心组件都是开源软件，在此基础上，我们做了如下一些工作：

- 集成与管理系统
- 设备管理
- 基于HTTP的异步处理框架
- 非HTTP协议的接入，比如MQTT
- 基于FastCGI的多编程语言支持
- 监控
- 调度
- 流量分发



# Key Takeaways

- 边缘计算平台 = 云计算 + 边缘计算
- 边缘计算是**面向流动数据**的
- 常用的数据中心软件在边缘环境并不适用
- 面对不确定、不可靠的计算环境，**异步**很重要
- Reset-To-Work是核心的设计观念。数据中心之外，**没有运维，只有重启**
- 和云计算一样，难点和重点是：**监控 + 调度**。我们采取以网络监控为主的方式
- 云计算是面向SLA的；边缘计算是**面向流量**的：把**流量交付**出去，和把**流量收集**上来，是边缘计算的核心诉求

# 云计算



# 边缘计算



# Q&A