边缘函数

产品简介

产品文档

腾讯云

文档目录

产品简介
产品介绍

最近更新时间：2019-08-01 11:51:33

腾讯云边缘函数（Edge Function）帮助您在自有的物联网设备、虚拟服务器或物理服务器上运行应用，管理、转发和缓存消息，实现边缘设备与云的结合，实现边缘计算。您只需要进行核心代码的编写、设备的接入配置和管理，代码即可在您的自有设备上按需、安全地运行，实现边缘设备的能力扩展及智能化升级。

从云至边缘的发展和变迁

随着入网设备的越来越多，数据产生的速度越来越快，体量越来越大，一方面，云计算在不断的持续发展，以提供强大的计算资源，充足的带宽及无限量的存储能力；另一方面，边缘计算也可以开始发挥自身所拥有的特性，以靠近数据源的位置，实现快速实时的响应，有效的数据过滤及初步处理。

边缘函数简介

腾讯云边缘函数是腾讯云利用已有的无服务器云函数，提供将无服务器云函数调度到自有设备上按需运行的能力。在这个过程中，除了管理函数及所运行的对应设备，还提供了消息能力，能够通过云与设备间建立的安全消息通道，实现云 - 端的消息交互、设备端的消息转发管理、断网时的缓存。

配置灵活

边缘函数利用了无服务器云函数可按需调动、自动扩缩容的特性，根据用户配置情况，将函数代码下发至边缘设备，并根据用户需求，以触发方式或持续运行方式启动并运行，并同样可以根据配置和触发情况，自动扩缩容以满足请求量。

开发高效

使用边缘函数，您可以很方便地开发、测试并配置边缘设备上的应用，例如实时更新物联网边缘设备中的数据清洗程序，或自有机房内的服务器上的数据分析代码。通过在云上编写无服务器云函数，可以在线预先进行模拟请求测试，提高应用开发效率。

简化管理

使用边缘函数，通过将云函数绑定至设备，您可以将应用轻松发布至您管理的设备上，无论设备是两三台，还是成百上千台，均可以轻松完成配置和更新，协助您完成设备更新的 OTA 过程。设备将自动完成代码更新及配置更新，并无缝切换至最新版本的代码开始运行。

自定义代码运行方式

同时，您还可以自定义您的代码运行的方式。边缘函数提供在设备上触发运行、持续运行两种运行方式。

- 触发运行类似云上的无服务器云函数，由消息事件触发，在有指定消息产生时，函数代码会被拉起运行，通常这种模式用于数据处理和响应。
- 持续运行是在边缘节点上提供的一种新的运行方式，函数代码会以类似守护进程的方式运行，您可以自行在代码内使用死循环来保持进程，或利用我们的守护进程来持续无条件的拉起函数运行，通常这种模式用于数据采集、文件读取分析。

产品优势

最近更新时间：2019-08-01 11:52:55

简单易用

自动更新

云上的函数更新、配置更新，通过简单的一键部署，就可以完成下发到设备上的操作。设备自动进行函数及配置更新，自动使用新的代码及配置开始运行。

高效开发

简化开发

边缘函数可以改变原有嵌入式物联网设备的应用开发流程，支持使用更现代化的开发语言来进行嵌入式或物联网设备的应用开发。无需采用原有的应用开发、编译打包、远程升级的繁琐工作流程，将物联网开发简化为在线应用开发及测试、一键下发、自动更新升级。

稳定可靠

离线运行

边缘函数在同步至设备后，在设备断网情况下仍然可以按配置持续运行，不受网络条件影响。

消息缓存

在设备与云端的连接中断的情况下，云端和设备端的消息传递将会缓存下来，并在连接重新建立后发送，避免消息丢失。

简化管理

可视化管理

用户可直接在控制台管理函数代码及边缘设备，简单单击即可完成函数下发到设备的配置。

简化设备配置

通过云上提供的 Agent 包及配置文件，下载、解压、运行即可将设备接入云网络，实现设备入网。

价值提升

设备高效利用

通过持续更新函数，为设备带来更多功能，增强设备能力，为硬件带来更多的价值提升。

AI in all

通过使用边缘函数的 AI 推理能力，将 AI 能力赋予到设备中，真正实现 AI in all。

工作原理

最近更新时间：2019-08-01 11:53:55

函数运行时的容器模型

边缘函数运行机制仍然利用了容器机制。在设备端 Agent 内，我们内置了 runc 轻量级容器。通过容器，我们仍然可以有效地控制代码程序对设备 CPU、内存的使用，避免函数代码对设备内已有进程或系统的影响。

同时，由于边缘设备的种类多样，硬件、系统环境有众多差异，因此，我们也会将更细粒度的函数运行环境控制逐步暴露出来，便于用户针对硬件差异进行调整。

边缘函数的设备端 Agent 将在事件触发时执行 SCF 函数，根据您的配置信息（如内存等）进行资源分配，并启动和管理容器（即函数的执行环境）。Agent 负责设备上所有函数运行容器的创建、管理和删除操作。

同云上的无服务器云函数相同，容器启动时需要一些时间，这会使得每次调用函数时增加一些延迟。但是，通常仅在首次调用函数、更新函数、或长时间未调用时重新调用时会察觉到此延迟，因为平台为了尽量减少此启动延时，会尝试对后续调用重用容器，在调用函数后容器仍会存留一段时间，预期用于下次调用。在此段时间内的调用会直接重用该存留的容器。

> ⚠ **注意：**
>
> 请勿在函数代码中假定始终重用容器，因为是否重用和单次实际调用相关，无法保证是创建新容器还是重用现有容器。

运行模式

边缘函数支持在与设备绑定时设置为 **触发运行** 或 **持续运行** 两种运行方式。

触发运行 与函数在云上的运行模式相同，在被触发时才会拉起运行。

持续运行 是在边缘函数中才有的模式。配置为持续运行的函数，在更新到设备上之后，就会拉起并开始运行。持续运行的函数，会以单例的方式运行，也就是仅会启动一个容器。持续运行的函数，如果由于代码退出或异常而终止，将会被重新启动运行。因此，用户可以通过在代码内置死循环的方式防止退出，或单次执行完后正常退出，由系统再次拉起执行。