

— 2017 Tencent Cloud Summit —

云⁺未来 峰会 | 北京站

连接 · 智能 · 未来

无服务器架构-让云端开发更纯粹

腾讯-架构平台部-陈杰

我们一直在去服务器化

物理机托管

无须关心机房运维

无人值守

开发（天）/部署（小时）

云主机

无须关心物理服务器

调度自动化

开发（天）/部署（分钟）

容器

无须关心服务部署

标准持续集成

开发（天）/部署（秒）

云函数

无须关心服务运维

计算服务化

开发（分钟）/部署（毫秒）

降低开发运维负担，更聚焦业务创新，解放生产力

无服务器云函数是真正的计算服务

	COS对象存储	SCF无服务器云函数
目标	数据托管	计算托管
单位	文件	函数
接口	上传/下载/修改文件	上传/执行/修改函数
计费	文件容量大小/存储时长/下载上传流量	函数资源大小/执行时长/计算流量

无服务器架构是FAAS+BAAS

FAAS
Function as a Service
函数即服务

SCF 无服务器云函数

+

BaaS
Backend as a Service
后端即服务

COS 对象存储

CDB 云数据库

CMQ 消息队列

REDIS 弹性缓存

CDN内容分发

....

无服务器云函数的核心价值



简化架构

函数粒度的微服务架构，使得系统的各个功能天然解耦，能像搭积木一样组合自有及外部服务。



简化开发

无需关注底层硬件配置、OS，服务启停、网络收发，故障容灾，服务扩缩容等，只需写最核心的业务逻辑，实现代码即服务。



简化运维

无须关注服务部署，服务器运维，安全管控，扩缩容配置等，且应用能无缝升级，天生就是DevOps模式。



减少支出

无闲置成本，仅对函数资源大小，执行时间，执行次数按需计费。相对云主机平均5%~15%的使用率，更绿色节能省成本。

无服务器云函数的典型开发过程



创建函数

选择运行时环境
配置内存资源
配置超时时间



代码编写

Web在线编写代码
或 上传代码zip包
或 配置代码cos地址



触发配置

定时触发
cos事件触发
cmq事件触发
...



运行监控

函数执行结果
函数执行延时
函数执行日志
...

实现一个简单的缩略图生成服务

对上传至COS目录的图片自动生成缩略图



实现一个简单的缩略图生成服务

函数名称

CreateThumbnail

最长60个字符，字母开头，支持 a-z, A-Z, 0-9, -, _

运行环境

Python 2.7

内存 ①

128MB

超时时间 ①

3

秒

1-300秒

描述 (选填)

请输入函数的描述

最大支持1000个英文字母、数字、空格、逗号和英文句号，不支持中文

下一步

代码输入种类

☒ 在线编辑☐ 本地上传zip包

执行方法 ①

CreateThumbnail.main_handl

```
53
54     # download image from cos
55     request = DownloadFileRequest(bucket, "/" + key.split('/')[1], download_path)
56     download_file_ret = cos_client.download_file(request)
57     if download_file_ret['code'] == 0:
58         logger.info("Download file [%s] Success" % key)
59         logger.info("Image compress function start")
60         starttime = datetime.datetime.now()
61
62         #compress image here
63         resize_image(download_path, upload_path)
64         endtime = datetime.datetime.now()
65         logger.info("compress image take " + str((endtime-starttime).microseconds/1000) +
66
67         #upload the compressed image to resized bucket
68         request = UploadFileRequest(u'%sresized' % bucket, key.decode('utf-8'), upload_path)
69         upload_file_ret = cos_client.upload_file(request)
70         logger.info("upload image, return message: " + str(upload_file_ret))
71
```

实现一个简单的缩略图生成服务

添加触发方式

触发方式

COS触发

COS Bucket ⓘ

scfsource

事件类型 ⓘ

文件上传

保存 取消

调用时间



日志内容 2017-08-08 17:14:55

运行时间:80.709ms 计费时间:100ms 占用内存:0.242MB

2017-08-08 17:14:55

调用成功

2017-08-08 17:14:54

调用成功

2017-08-08 17:14:54

调用成功

2017-08-08 17:14:53

调用成功

2017-08-08 17:14:53

调用成功

2017-08-08 17:14:53

调用成功

2017-08-08 17:14:52

调用成功

"ret_msg":

None

"log":

start main handler

Get from [scfsource] to download file [/1251350908/scfsource/n10405540_3319.jpg]

Uri is http://scfsource-1251350908.cosgz.myqcloud.com/n10405540_3319.jpg?

sign=oJm37ITKq1iNnzg07U2w8HyN4JhPTEyNTEzNTA5MDgmaz1BS0IEUGhNd1ozR2dKa0Raamg3M1JjYIAyV

http://scfsource-1251350908.cosgz.myqcloud.com:80 "GET /n10405540_3319.jpg?

sign=oJm37ITKq1iNnzg07U2w8HyN4JhPTEyNTEzNTA5MDgmaz1BS0IEUGhNd1ozR2dKa0Raamg3M1JjYIAyV

HTTP/1.1" 200 149198

Download file [/1251350908/scfsource/n10405540_3319.jpg] Success

Image compress function start

Error closing: 'NoneType' object has no attribute 'close'

compress image take 5ms

sending request, method: POST, bucket: scfsourceresized, cos_path:

/1251350908/scfsource/n10405540_3319.jpg

http://gz.file.myqcloud.com:80 "POST

/files/v2/1251350908/scfsourceresized/1251350908/scfsource/n10405540_3319.jpg HTTP/1.1" 200 563

upload image, return message: {u'message': u'SUCCESS', u'code': 0, u'data': {u'url':

u'http://gz.file.myqcloud.com/files/v2/1251350908/scfsourceresized/1251350908/scfsource/n10405540_3319.ji

对比传统开发模式的收益或不足

	传统开发模式	无服务器架构开发模式	收益或不足
架构设计	架构分层，功能划分，模块通信，并发，容灾，安全，成本设计...	函数功能划分，函数调用流程/组合，事件触发关系定义等	明显降低架构设计的工作量，上述例子降低90%以上
代码开发	网络服务开发，业务逻辑开发，容灾/并发/扩缩容等非功能性需求开发	针对事件的处理函数开发	明显降低代码开发工作量，上述例子降低90%以上
代码调试	登陆服务器使用熟悉的工具本地调试	主要依赖日志调试	代码调试体验暂不如传统开发模式
发布部署	传统软件分发，或使用容器标准化部署，多版本管理，热升级实现较为困难	通过API或页面一键部署，支持函数多版本，可通过别名选择正式环境版本，支持热升级	明显提升发布部署效率，减少发布部署对业务的影响
服务运维	服务器运维负担重，业务扩缩容，故障处理等成本较高，耗时长	依据负载自动扩缩容，故障自愈	免去服务器运维，免去扩缩容/故障处理工作量，减少对业务的影响
服务成本	按使用时间付费，至少保留1台服务器	按照实际使用付费，无须保留服务器	彻底的按需付费，无闲置成本

无服务器云函数的架构原理

调用源

函数调用来源于用户或内部云产品事件触发

接入层

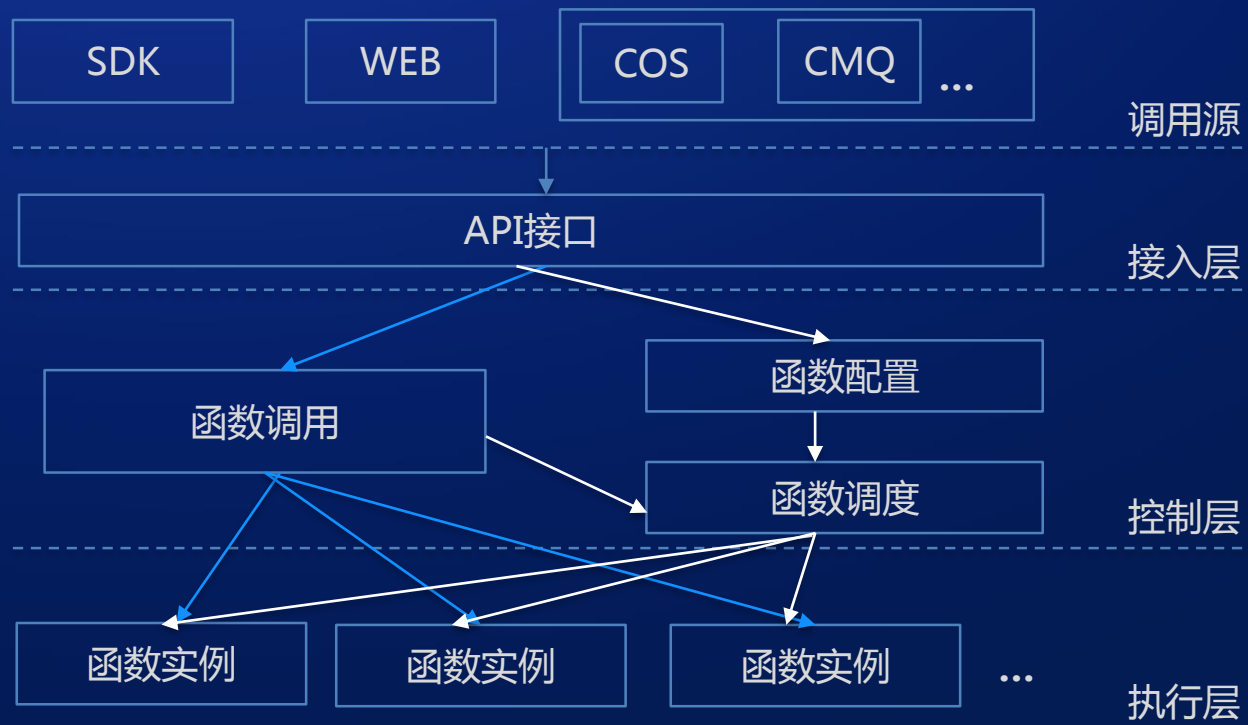
负责用户鉴权等问题

控制层

负责函数调用的分发，函数配置，函数实例生命周期管理，动态扩缩容，容灾调度，负载均衡等

执行层

负责运行时环境，执行用户函数



无服务器云函数的适用场景



无服务器云函数的关键特点

按需分配函数资源，函数实例不常驻
单次调用有超时限制，默认3s，最长5分钟
依据负载自动扩缩容，扩容可在1~2s内完成
第一次调用时间较长：1~2s
常态调用有稍许损耗：3~5ms
按照实际调用次数/时长/资源计费



无服务器云函数适用场景

内存无状态微服务，如有状态，需借助其它组件，
比如CMQ，CDB等实现状态保存
单次调用时长不超过5分钟
有明显的波峰波谷效应，资源经常闲置
对延时不敏感，能够容忍额外3~5ms延时
低频服务，比如月调用次数小于1百万

无服务器云函数典型应用场景

音视频转码

视频应用、社交应用等场景下，用户上传的图片、音视频的总量大，频率高，对移动应用的实时性和并发能力都有较高的要求。

使用多个SCF无服务器云函数，您可以轻松对上传的视频短片进行实时并发转码，以满足不同场景下用户的需求。

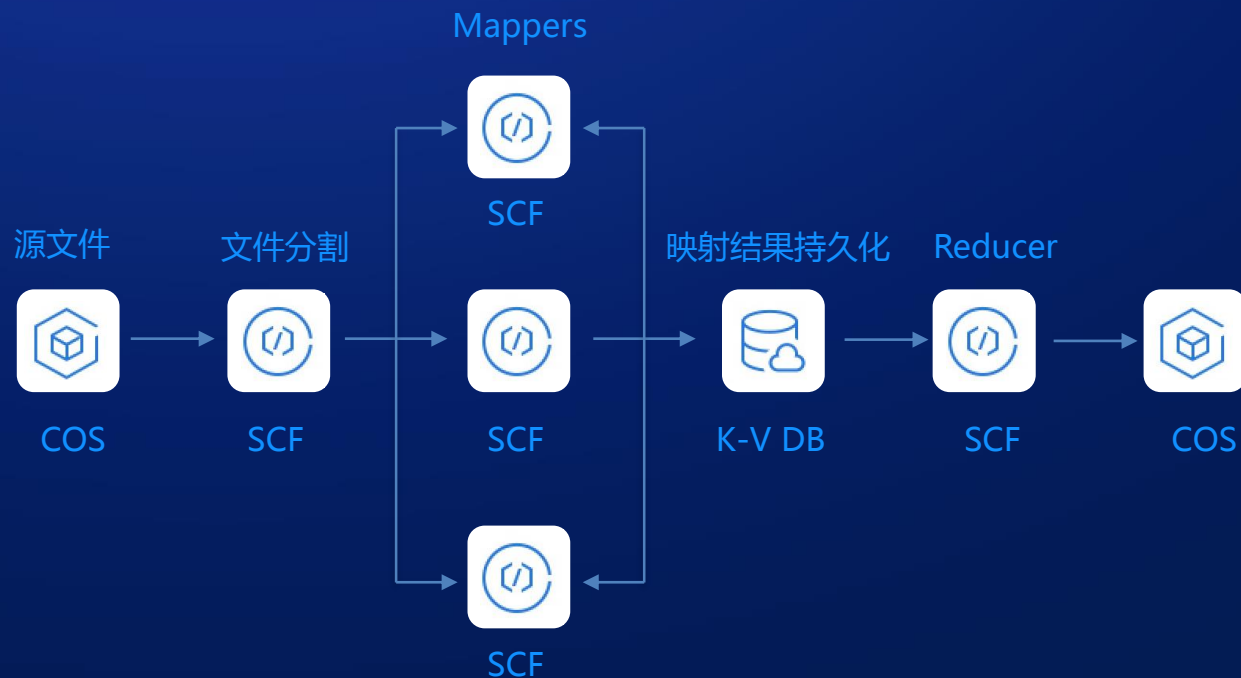


无服务器云函数典型应用场景

数据ETL处理

电商、金融等应用经常需要实时或周期性处理大量数据，例如：电商每天处理一遍秒杀网站的交易流日志获取因售罄而导致的错误从而分析商品热度和趋势等。

使用SCF 无服务器云函数，您可以轻松地进行大容量数据的计算，比如可以对源数据并发执行多个 mapper 函数，并通过reducer函数汇总执行结果。



无服务器云函数典型应用场景

Web或移动端后台

微信小程序或其它长尾/低频型业务场景，维护服务器后台代价较大。

使用SCF无服务器云函数及其他云服务，您可以轻松构建可弹性扩展并在多个数据中心高可用运行的移动或 Web 应用程序后台。



SCF应用实践-Web服务拨测

Web服务拨测

在云函数里使用curl主动探测web服务健康状态，尝试在用户前发现并解决问题。

收益：快速实现，几乎无成本。

```
import pycurl
import apdwarn
c = pycurl.Curl()
c.setopt(pycurl.TIMEOUT, 5)
w = apdwarn.Init()

URL = "172.xxx.xxx.xx" #the url to test

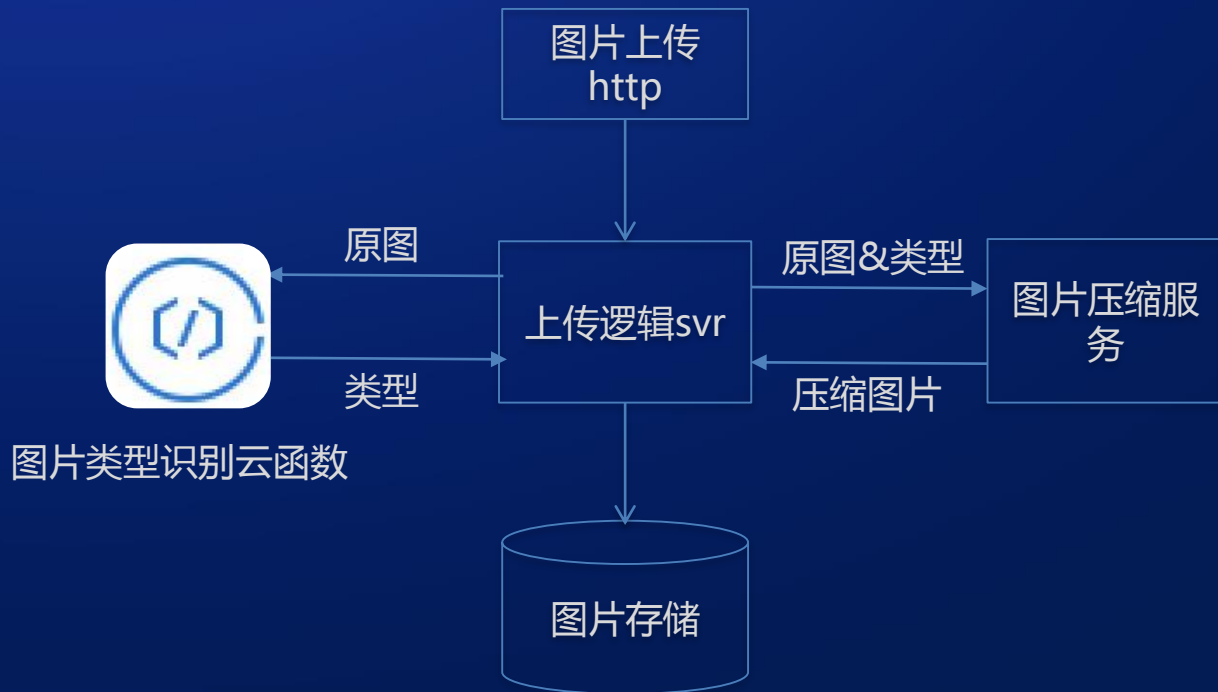
def main_handler(event, context):
    try:
        c.setopt(pycurl.URL, URL)
        c.perform()
        ret = c.getinfo(c.HTTP_CODE)
        if ret == 200:
            print "INFO, the service is ok"
        else:
            warnmsg = "WARN: the service %s return %d" %(URL, ret)
            print warnmsg
            w.warn(warnmsg)
    except Exception,e:
        ret = "conneccion error:"+str(e)
        warnmsg = "ERROR: %s" %(ret)
        print warnmsg
        w.warn(warnmsg)
        c.close()
    return ret
```

SCF无应用实践-图片类型识别

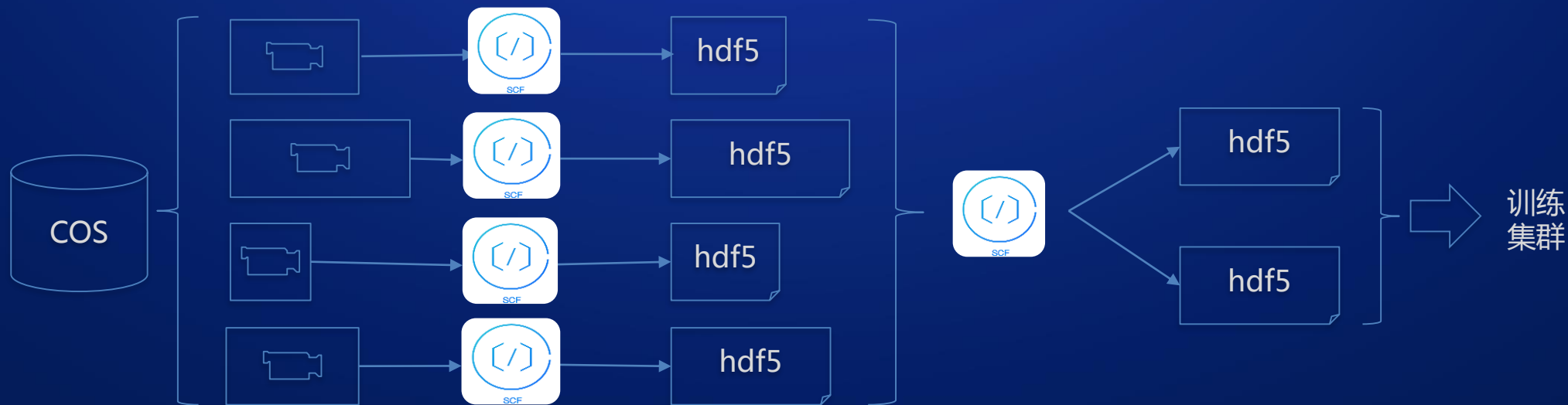
嵌入图片类型识别功能

在已有的图片上传压缩流程里嵌入图片类型识别功能，识别广告图片使用更高的压缩比，节省下载流量与存储容量。

收益：低成本快速实现试错



SCF应用实践-AI训练数据预处理



AI数据预处理服务

将游戏录像文件特征提取，转化为可供训练的hdf5文件，再将这些文件shuffle随机化按照固定帧数规整输出。

收益：快速实现数据预处理，减轻AI工程师平台型事项的工作量，能更关注算法创新；同时避免保留大量计算资源，节省成本。

无服务器云函数的注意事项

使用无状态风格编写代码

以无状态的风格编写函数代码，确保您的代码不依赖本地内存及文件系统状态，如需保留状态，应当使用COS、Redis/Memcached等服务缓存中间信息并落地存储最终计算结果。

尽量复用全局资源

在主执行方法外实例化任何可能复用的对象，静态数据初始化，数据库连接初始化等。

防止死循环式事件触发

配置触发事件时，避免死循环式事件触发，比如配置某cos bucket写文件事件触发，但在处理函数中写入了同一个bucket。

注意代码打包规范

打包时附加依赖库，比如对python运行时环境，可使用pip打包依赖库；已在上传的 ZIP 中设置对您的文件的可读及执行权限，以确保代码能够执行；针对代码内容打包，而非代码目录。

无服务器云函数近期新特性



Node.js支持

可选择Node.js 6.10或4.3运行时环境，使用API或页面直接提交node.js源代码或zip包，快速上线您的Node.js服务。



Java支持

可选择Java8运行时环境，支持通过API提交Java bytecode压缩包，快速上线您的Java服务。

SCF用户交流群



Tencent 腾讯 |  腾讯云