



无微不虑 净护全家

下单立享惊喜价
50元优惠券

感恩12月

A股暴跌后即将反弹
立即诊断手中个股是否有反弹潜力
准确率高达92%【散户必看】

首页 > 程序人生

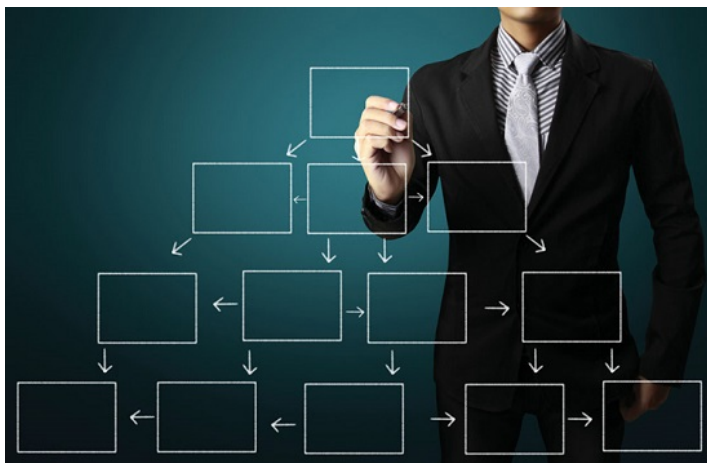
短视频应用应该如何打造技术架构?

2016-01-20 10:36 编辑: suiling 分类: 程序人生 来源: 创业邦

1 118

美拍 短视频应用

招聘信息: iOS研发工程师



本文系美图架构师麦俊生,在Boss直聘主办的直聘学院「对话架构师」活动上的分享整理,介绍短视频社交“美拍”架构实践的总结。

麦俊生, Boss直聘「直聘学院」特邀分享嘉宾。美图架构平台深圳技术总监,曾担任新浪微博、奇虎360技术专家,从事高性能高可用架构设计开发工作,参与建设微博的feed和私信im系统、负责rpc框架motan、cache service、counter service、公用类库等基础建设,以及奇虎360存储服务和基础框架方面的建设。个人擅长性能调优、高可用中间件、分布式存储、IM等相关领域。

以下是分享实录整理:《亿级短视频社交美拍架构实践》

热门资讯

宅男福音! Xcode 程序员鼓励师插件 Miku
点击量 20248看了一天简历的感悟~ 心真累
点击量 162892016年让开发工作变得更简单的小事
点击量 12176最轻巧的自动布局 --ZXPAutoLayout框架
点击量 11494iOS 架构模式--解密 MVC, MVP, MVVM
点击量 11447实践干货! 猿题库 iOS 客户端架构设计
点击量 11176如何做优化, UITableView才能
点击量 9439我从55个Swift标准库协议中学到了什么?
点击量 9270iOS面试一般性问题
点击量 8041基于彻底解耦合的实验性iOS架构
点击量 6172

综合评论

图片视频, 这个是怎么做到的?

kaly 评论了 2016年让开发工作变得更简单的小事...

讲的好像是有些简单 这里主要就是减少图片资源占用

nemoCH 评论了 iOS 滑动性能优化

晕了, 我。。。

uiformchao 评论了 单人研发2个月苹果年度最佳【Downwell】开发...

这广告做的, 什么都没讲一样

梦儿飘 评论了 短视频应用应该如何打造技术架构? ...

要看着原文对着看- -, 翻译得不太好
sharlier 评论了 编写高性能的 Swift 代码...

刚试了下 还行, 美化的只是脸部,



一、短视频市场的发展

近几年来，短视频应用在国内应用市场引爆，美图公司推出了美拍，GIF快手公司也推出了短视频产品，以及微博投资的秒拍，还有微视、逗拍、玩拍等一系列短视频产品也丰富了短视频应用市场。

短视频的相继爆发，与几个因素有关：

1.带宽，随着中国基础网络环境的发展，越来越多的2G移动网民开始转向3G/4G网络，从而提供更好的上传下载带宽和更稳定的体验，目前3G/4G的移动用户比例大概占比85%以上；同时随着资费的进一步降低，月户平均流量也达到了360M，甚至不少部分是GB甚至几十GB的级别的流量，所以就一个普通的10s视频而言大概不到1~2M甚至几百K，带宽流量的提升无疑会逐步降低用户使用的门槛；此外，家用带宽也随之增加，目前10M甚至100M已经成为家用带宽的主流，从而为短视频的发展提供了必要条件。

2.手机硬件配置的极大改进，随着像素的增加、手机硬件配置CPU、GPU、内存等的升级，能够让手机更快的来处理和优化视频效果，从而能够给手机视频的处理带来更多的创意空间。

3.传统的文字和图片的表达能力不够丰富，所以无法满足网民的需求，而短视频带来了足够大的表现空间。

4.产品本身，提供了各种方式来降低用户视频的制作门槛，比如美拍提供了MV特效等，来提升了制作视频的趣味性的同时，降低使用的门槛。同时产品的多样化，也满足了各种用户的差异化需求，激发用户自传播。

二、美拍的发展

美拍在2014.05月发布，上线仅1天，即登入App Store免费总榜第一，当月下载量排名第一。在发布9个月的时候，用户就突破1个亿。目前每天美拍视频日播放量在2.7亿以上，日视频播放时长达到183万小时。

面临这种用户量爆发的增长，美拍也遇到了很多应用起步之初那些年所遇到的甜蜜和苦涩的回忆，经过1年多的架构的演进，美拍也积累了一定的经验，形成了一套高可用性的架构实践。虽无法做到很华丽，却会随着架构的不断的演进而不断的完善。

相比于普通的文本社交类App，做这么一个短视频产品在技术架构层面，会面临哪些问题呢？

uiformchao 评论了 iOS版微软自拍App上架：自然美颜 上手简单...

请问下这个banner只能设置 默认大小吗 我想设置一个自定义的banner 可以YaSuoo 评论了 在APP中集成iAd Banner展示广告盈利...

赞一个。终于找着command + 0 command + option + 0 了 昨天老师还神二哥 评论了 14个Xcode中常用的快捷键操作...

这个东西我早在atom看见无数次了，是个国人制作的，但是这个东西下载数量sharlier 评论了 宅男福音！Xcode 程序员鼓励师插件 Miku...

居然有评论

OOC 评论了 如何创造一款带有喜剧元素的游戏...

相关帖子

急求解决，为什么打包H5安卓手机不清晰？

阳光保险股权，保险业迎来“钻石期”亟待转型升级

微信支付预处理body中文编码问题

cocos3.9支持的cocostudio版本是多少

葛洲坝电力集团有限责任公司深化改革实现转型升级

cocos2dx + lua 接入sdk切换黑屏，求大神解答

ios中引用计数器为什么会成-1

tableView上textField的显示问题

Cocos Creator 论坛专区已开放

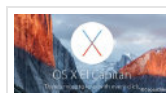
微博



CocoaChina

加关注

【苹果发布 OS X 10.11.3 正式版，修复 bug】19日，苹果发布了 OS X 10.11.3 正式版。这是 El Capitan 操作系统自去年9月30日发布以来第三次更新。OS X 10.11.3 在去年11月16日开始测试，而 OS X 10.11.2 正式版也在去年12月发布。<http://t.cn/R4QmNLs>



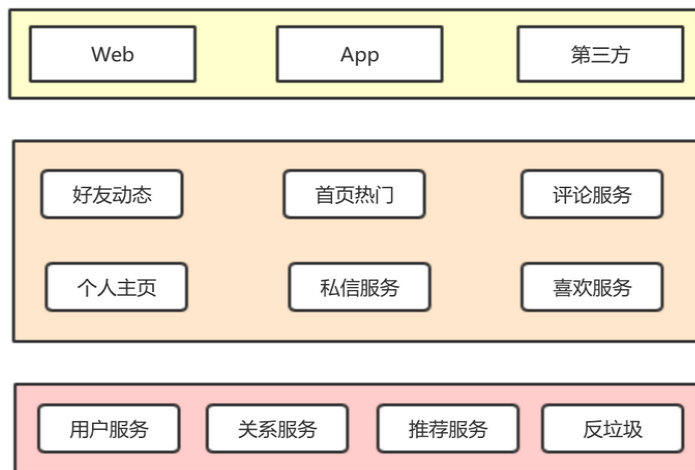
29分钟前

转发(1) | 评论

转发微博



王芋头先森：受阳仔启发，搞了个这个<http://t.cn/R4YgFd3>



和通用的文本社交类产品一样，美拍有首页热门、好友动态(feed流)、评论服务、私信服务等基础功能；所以在用户爆发增长后，同样会面临应用层、数据库、缓存、接入层等方面的挑战，那么如何做到低延迟、高可用呢。同时因为是短视频本身，也会面临一些特定的领域性相关的问题。

三、短视频所面临的架构问题

短视频相比于文本数据而言，有着一些差异：

1.数据大小的差异。

比如一条美拍，经过视频压缩和清晰度的权衡，10s的视频大小1MB多，而一条5分钟视频的美拍甚至要达到几十M，相比与几十字节或者几百字节的文本要大得多。因为数据量要大得多，所以也会面临一些问题：如何上传、如何存放、以及如何播放的问题。

关于上传，要在手机上传这么一个视频，特别是弱网环境要上传这么一个文件，上传的成功率会比较低，晚高峰的时候，省际网络的拥塞情况下，要更为明显得多。所以针对上传，需要基于CDN走动态加速来优化网络链路（通过基调实测过对于提升稳定性和速度有一定帮助），同时对于比较大的视频需要做好分片上传，减少失败重传的成本和失败概率等来提升可用性。同时不同CDN厂商的链路状况在不同的运营商不同地区可能表现不一，所以也需要结合基调测试，选择一些比较适合自己的CDN厂商链路。

同时因为数据相对比较大，当数据量达到一定规模，存储容量会面临一些挑战，目前美拍的视频容量级别也达到PB级别的规模，所以要求存储本身能够具备比较强的线性扩展能力，并且有足够的资源冗余，而传统的Mysql等数据库比较难来支持这个场景，所以往往借助于专用的分布式对象存储，通过自建的服务或者云存储服务能够解决，得益于近几年云存储的发展，目前美拍主要还是使用云存储服务来解决。自身的分布式对象存储主要用于解决一些内部场景，比如对于数据隐私性和安全性要求比较高的场景。

关于对于播放，因为文件比较大，也容易受到网络的影响，所以为了规避卡顿，一些细节也需要处理。比如对于60s，300s的视频，需要考虑到文件比较大，同时有拖动的需求，所以一般使用http range的方式，或者基于HLS的点播播放方式，基于前者比较简单粗暴，不过基于播放器的机制，也能够满足需求，也能实现点播拖动。而直接基于HLS的方式会更友好，特别是更长的一些视频，比如5分钟甚至更大的视频，不过这种需要单独的转码支持。之前美拍主要是短视频为主，所以更多使用http range的方式。而后续随着5分钟或者更大视频的场景，也在逐步做一些尝试。同时对于播放而言，在弱化环境下，可能也会面临一些问题，比如播放时长卡顿的问题，这种一般通过网络链路优化；或者通过多码率的自适应优化，比如多路转码，然后根据特定算法模型量化用户网络情况进行选码率，网络差的用低码率的方式。

2.数据的格式标准差异

相比与文本数据，短视频本身是二进制数据，有比较固定的编码标准，比如H.264、H.265等，有着比较固定和通用的一些格式标准。



3.数据的处理需求

视频本身能够承载的信息比较多，所以会面临有大量的数据处理需求，比如水印、帧缩略图、转码等，以及短视频鉴黄等。而视频处理的操作是非常慢的，会带来巨大的资源开销。

美拍对于视频的处理，主要分为两块：

- 客户端处理，视频处理尽量往客户端靠，利用现有强大的手机处理性能来规避减少服务器压力，同时这也会面临一些低端机型的处理效率问题，不过特别低端的机型用于上传美拍本身比较少，所以问题不算明显。客户端主要是对于视频的效果叠加、人脸识别和各种美颜美化算法的处理，我们这边客户端有实验室团队，在专门做这种效果算法的优化工作。同时客户端处理还会增加一些必要的转码和水印的视频处理。目前客户端的视频编解码方式，会有软编码和硬编码的方式，软编码主要是兼容性比较好，编码效果好些，不过缺点就是能耗高且慢些。而硬编码借助于显卡等，能够得到比较低的能耗并且更快，不过兼容和效果要差一些，特别是对于一些低配的机型。所以目前往往采用结合的方式。
- 服务端的处理，主要是进行视频的一些审核转码工作，也有一些抽帧生成截图的工作等，目前使用ffmpeg进行一些处理。服务端本身需要考虑的一些点，就是因为资源消耗比较高，所以需要机器数会多，所以在服务端做的视频处理操作，会尽量控制在一个合理的范围。同时因为可能美拍这种场景，也会遇到这些热点事件的突变峰值，所以转码服务集群本身需要具备弹性伸缩和异步化消峰机制，以便来适应这种突增请求的场景。

4. 审核问题

视频内容本身可以有任意多样的表现形式，所以也是一个涉黄涉恐的多发地带，而这是一个无法规避掉的需求，因为没有处理好，可能分分钟被封站。

审核的最大的问题，主要是会面临视频时长过长，会带来人力审核成本的提升。比如100万个视频，每个平均是30s的话，那么就3000W 秒，大概需要347人日。

通过技术手段可以做一些工作，比如：

- 接入一些比较好的第三方的视频识别模块，如果能够过滤掉85%保证没有问题的视频的话，那么工作量会缩减到15%。不过之前在接入使用的时候，发现效果没有达到预期，目前也在逐步尝试些其他方案。
- 通过抽帧的方式，比如只抽取某几帧的方式进行检查。
- 通过转码的方式，比如一个60s的美拍视频，通过2倍速的方式，无声，140 * 140的分辨率转换，大概大小能够在650kB左右，这样加速了播放的过程的同时，还能够减少审核带宽的消耗，减少了下载过程。
- 基于大数据分析，分析一些高危地带、用户画像等，然后通过一些黑名单进行一些处理，或者对于某些潜在高危用户进行完整视频的审核，而对于低危用户进行抽帧的方式等等。

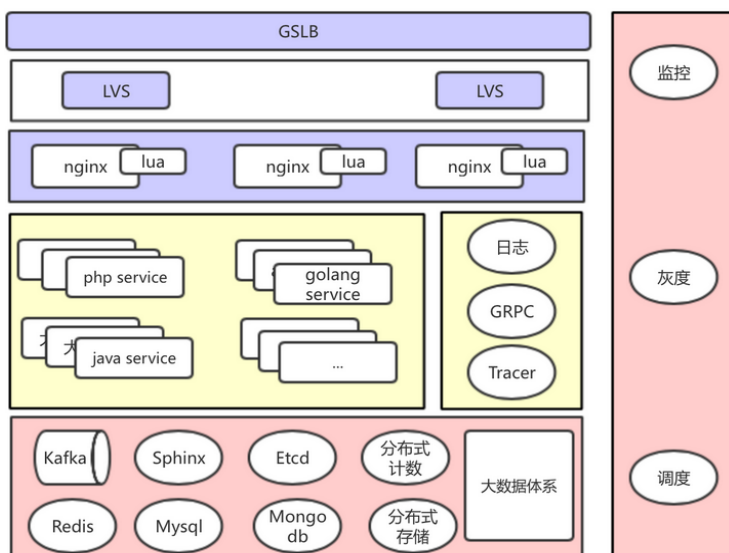
四、为支持亿级用户，美拍架构所做的一些改进

随着用户和访问量的快速增长，美拍遇到不少的挑战：

- 性能的挑战
- 可用性的挑战
- 突发热点的挑战
- 业务频繁迭代的挑战

在频繁的业务迭代的情况下，如何能够在海量请求下需要保证足够高的可用性，同时以一个比较好的用户体验和比较低的成本的方式来提供服务成为我们努力的方向。

这个是目前美拍的整体架构全貌：



这一个架构目前也正在不断的持续演进的过程，同时除了一些基础服务组件的建设外，我们还着重在服务治理做一些相关工作，来保证整体服务的可用和稳定。



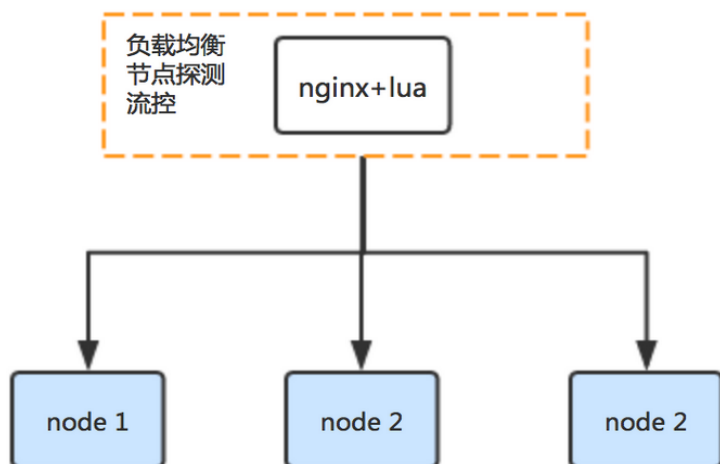
分而治之、化繁为简

规划整体架构，明确服务模块的单一职责，尽量保持足够内聚，而服务模块之间做到解耦，这样就能够针对单一模块进行更精细化的优化工作，同时能够适合合适技术解决合适的场景问题。

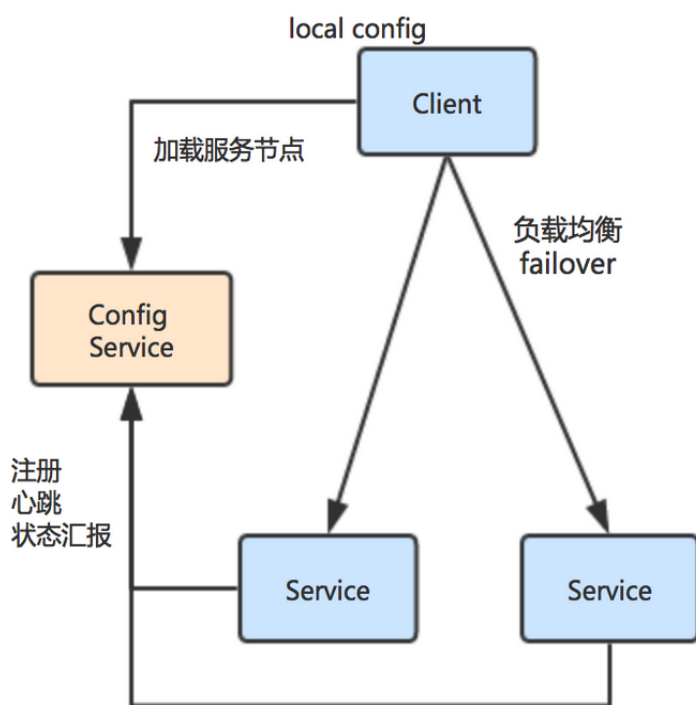
服务之间的交互和通讯，我们主要走了两种方式：

- 基于http的方式
- 基于config service + rpc的方式

前者使用的方式比较简单，目前我们主要在跨团队、跨语言（比如php和golang之类的）会使用，主要会在七层nginx层做一些工作，如负载均衡、节点探测、并发保护等。



而第二种方式，我们主要用于内部系统之间的一些交互。目前我们主要基于etcd来实现我们的动态服务发现和配置服务，在client层面扩展实现了包含负载均衡、心跳、节点健康状态探测、etcd节点挂掉的灾备等基础功能，同时会通过一些metrics埋点，以便跟踪内部的状态，用统一的trace_id来跟踪服务的链路调用情况。



开放扩展

主要针对下面几个点：

- 代码功能的可扩展性
- 交互协议的扩展性
- 数据存储格式的可扩展性
- 应用的可扩展性
- 资源的可扩展性

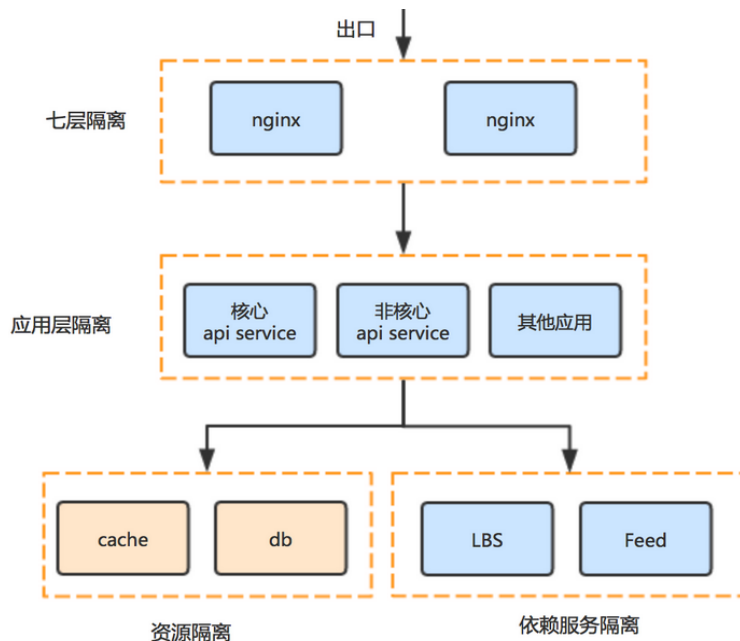
比如，交互协议，既针对交互接口，也针对app客户端和服务端的交互协议。特点是app客户端和服务端的交互协议，因为app的升级较之服务端升级的时间久得多，比如你发布了一个客户端版本V0.1，如果用户后面一直不升级，这个时间可能是几个月、半年甚至一年，那么就会引入一些兼容问题，所以在协议层面设计的关键点需要考虑这种情

况的存在，需要保证协议能够向前兼容，预留好扩展点。

分级隔离

目前我们主要通过这几个维度进行一些隔离：

- 核心和非核心的隔离
- 单一集群的内部隔离
- 不同集群的外部物理资源隔离
- 不同集群的外部依赖资源的隔离



美拍在发展早期，跟多数发展早期的系统一样，也是多数接口部署在同一个集群中，包括也共用了一些资源（比如 memcached），这样的好处是早期部署上足够的简单。在发展的过程中，业务在快速发展，业务复杂度也在逐步提升，接口调用量也急剧增加，逐步就暴露出一些问题。美拍的发展过程也是实际的去验证了前面提到的分级隔离机制。

在发展早期，曾经有个调用量不小的非核心的业务，在对存储数据结构做了调整后的上线过程中出现性能问题，导致整个集群服务都受到一定的影响。虽然通过降级策略和运维配套设施快速的解决了问题，但是也引发了我们的进一步思考。在架构上我们会尽量保证在开发效率、系统架构、部署和运维成本等方面达到一定的平衡，以避免过度设计或者架构支撑不了业务。这到了需要做一些事情的时候，我们把核心业务和非核心业务在七层和应用层做了部署上的隔离。

做完上面的核心业务和非核心业务拆分之后，接口互相之间的依赖影响降低很多。但是还没有解决核心业务或者非核心业务内部接口之间的依赖影响问题。所以接下来也更进一步，针对部分场景也做了内部隔离，通过限定每个接口最多只能使用的固定处理线程数方式，来避免因单个集群内某个接口的问题导致整个集群出问题的情况发生。

以上主要是在接口层面做隔离，而在依赖的资源及其外部服务方面，如果没有相应的隔离机制，也会有互相依赖影响的问题，比较典型的有 memcached slab calcification 问题等。所以我们也 memcached、mysql 等核心资源层面做了拆分。

综合来看，分级隔离本质上也是在解决服务之间依赖影响问题。

资源冗余

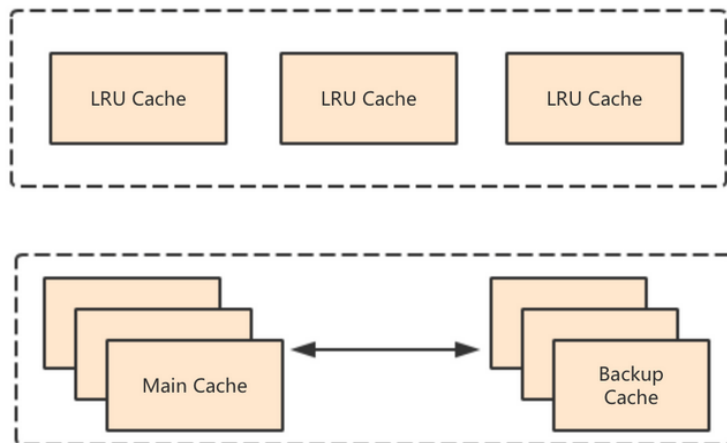
因为短视频是一个比较耗带宽的服务，因此在通用的应用自身资源冗余的情况下，还需要考虑到服务所依赖的外部资源，比如 CDN 和云存储服务本身的情况。对于 CDN 层面，可能还要考虑不同厂商在不同区域不同运营商下的资源冗余

情况。而依赖的云服务等, 这些服务本身从对外角度看是一个可无限扩展的服务, 理论上通过扩展就能够满足性能需求, 但是在使用往往会受限于实现, 因为内部不一定是一个完全隔离的场景, 比如说和别的企业服务混跑, 同时可能只会分配对应的资源池, 但这个资源超过性能预期的时候, 不是一个可自动动态伸缩调度的场景。

容灾

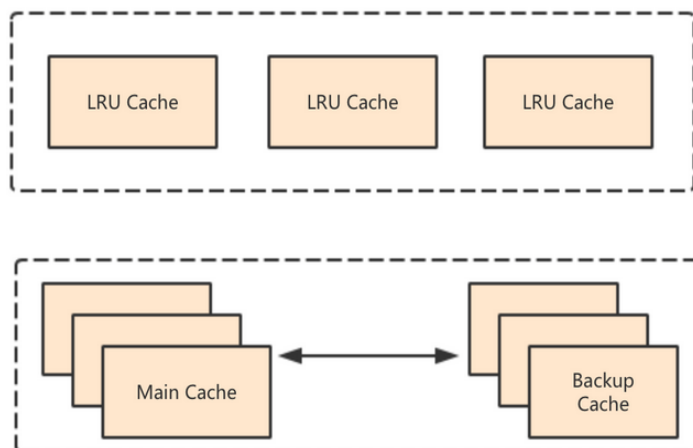
美拍的容灾主要分为自身服务容灾、CDN容灾、云存储容灾等。

自身服务容灾主要包含一些典型的容灾场景, 比如cache容灾, 通过多级cache、cache的分片hash的方式、以及本地cache的方式来解决。目前我们这边的容灾也借鉴了微博的多级cache机制的机制, 针对核心的cache资源会有主备节点, 避免单一节点挂掉后, 穿透会压垮后端DB, 同时对于请求量特别大的场景, 比如对于某个热点资源访问量很大的情况下, 也会在之前增加一层L1的LRU cache来规避和缓解这一问题。



CDN容灾主要通过接入多家供应商进行互备, 然后通过一些基调检测不同服务厂商的链路和服务状态, 当发现服务有问题的时候, 通过DNS进行区域的切换。不过不同CDN厂商的服务表现不对等, 所以在选型CDN厂商的话, 需要侧重关注可用性、节点布点和链路状况、回源量、资源冗余量、晚高峰的链路状况、以及对于多媒体是否有单独优化等等来评估靠谱性。

云存储容灾, 目前美拍也主要使用两家互备的方式, 因为国内的网络链路状况容易出问题容易导致个别上传服务失败, 以及云服务厂商服务挂掉的情况我们需要保证我们的服务可用。目前的做法是上传优先走主的云服务, 如果上传失败的话, 那么就会启用备的云服务。然后服务端层面也可能控制整体降级的方式, 可以直接从主云服务直接降级读些备云服务。基于每天的统计来看, 通过这种方式至少提升上传的0.1%以上的可用性, 在某些极端情况下, 可能达到1%的可用性, 当然这一块通过网络链路优化可能使得可用性情况没有数据中那么差。不过他的主要作用是在当某个云服务厂商节点服务出现短暂不可用或者长时间不可用的时候, 我们也不会受太大影响。



五、后续的一些发展

随着短视频的不断发展,以及实时直播的崛起,带宽的压力会越来越大,所以能够结合着P2P+CDN的方式来缓解服务端的带宽压力,不过P2P主要会面临着防火墙的问题、以及节点网络质量的影响,同时也依赖与视频播放的热度,这种对于效果都会有一些影响,同时为了更好的播放流畅度,单一的P2P无法满足需求,需要基于P2P和CDN的辅助进行。而带宽的另外一个节省之道,就是通过更好的编码标准来进行优化,比如H.265的编码标准,通过这个能够节省1半的流量,只不过目前H.265在硬编支持不是很好,只有个别手机机型支持,而软编码的方式相比与H.264,编解码速度要慢个几倍,这种对于能耗消耗比较高,处理也比较慢,不过随着硬件的不断升级,H.265将会是后续的一个趋势,同时依托于这个之上的一些图片编码标准也能够得到有效的应用,从而来节省带宽。

同时美拍会越来越多的把一些客户端的图片视频美化算法云端化,以服务的形式暴露给内部其他服务使用,以便能够支撑更多围绕“美”体系建设的产品生态链。这主要会面临的架构难点,就是资源消耗高。而这个的解决会依赖与两种方式,一种通过硬件GPU、协处理器、CPU SIMD指令等来优化性能,同时还需要解决架构的视频处理集群的自动弹性调度的问题,同时对于一些场景,比如类似与H5的推广页面,会逐步通过结合公有云调度的方式来解决。



微信扫一扫

订阅每日移动开发及APP推广热点资讯

公众号: CocoaChina

我要投稿

收藏文章

分享到:

0

上一篇: 月薪2W在iOS开发界算什么水平?

相关资讯



我来说两句



你怎么看? 快来评论一下吧!

发表评论

所有评论 (1)



梦儿飘

刚刚

这广告做的,什么都没讲一样

0 0 回复

[关于我们](#) [广告合作](#) [联系我们](#) [Cocos商店](#)

©2015 Chukong Technologies, Inc.

