一个互联网公司的云实践

—— 新浪云平台的经验和教训

@Easy

很高兴有机会和大家分享新浪在云计算实践上的一些经验和教训.正如标题所写,新浪是一个典型的互联网公司,云计算又是一个很大的领域,我们在什么时候采用云计算?采用哪些云计算技术?这些都是我们要认真思考的问题.经过一段时间的摸索,我们确定了两个根本出发点,一是要能给公司带来价值;二是要能让我们的网友看得见,摸得着,用得上.能真正改变他们的网络生活.

实践阶段

- 物理机集群→IaaS平台 (2008~)
- IaaS平台→PaaS平台 (2009~)

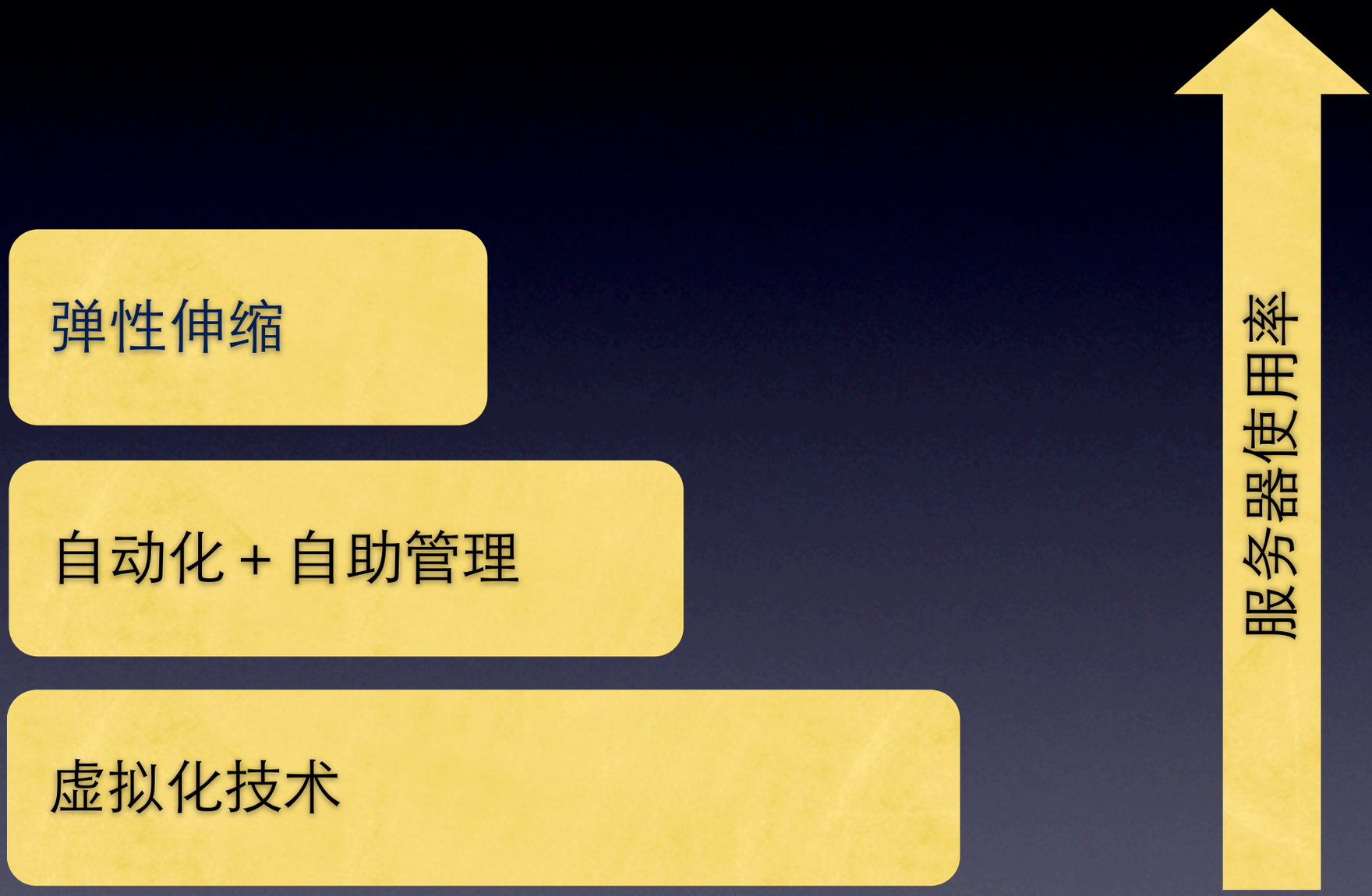
- 云存储(2009~)

新浪的云计算大体从08年开始,分别覆盖了iaas(基础设施即服务),paas(平台即服务)和saas(软件即服务)三个层次.其中我们有两个对外的产品,一个是面向开发者的paas平台--sae,一个是面向最终网友的saas产品微盘.我们先从第一阶段讲起.

物理机集群→IaaS平台

由于虚拟化技术的兴起和日益成熟,我们开始考虑将它用到我们的内部平台中.于是便有了第一阶段的实践.

实践思路



我们的基本思路是,将物理机虚拟化,将不同的虚拟机分配给不同的项目,从而提高服务器的使用率,帮公司节省服务器.在虚拟化技术的基础上,我们还加强了自动化和自助管理功能,使整个平台可以在没有系统工程师参与的情况下,实现弹性伸缩.

实践效果

同一项目节省机器80%*

实践的结果很不错,理论上来讲,一个中型web项目能节省80的机器.但是事实上,我们节省的机器可能40%不到.这是因为存在一些问题.

存在的问题

- 虚拟化技术不稳定
- 资源滥用

最主要的问题有两个.一个是虚拟化技术不稳定,虚拟机经常挂掉;一个是大家都不把虚拟机当服务器,觉得很廉价,不像使用物理机那么节省.

解决方案

- Q：如何处理虚拟化技术不稳定?
- A：无单点设计，弹性伸缩，动态迁移

对开发者的架构能力要变高了

对第一个问题我们的解决方案包括两方面.一方面我们选用更可靠的虚拟化技术,但这个依赖于相关项目的成熟度.另一方面,我们开始从架构层去解决问题.首先是,要求我们的项目采用无单点设计,这样即使其中某一台虚拟机挂了,应用不会受影响;然后我们有一个自动化的监控系统,能及时发现宕机和服务受限的情况.发现后,就立即启动一个新的虚拟机,初始化为对应的角色,然后加入进去.这样,虽然虚拟机没有物理机靠谱,但是我们的服务可用性反而更高了.不过这个做法有一个问题,它对开发者的架构能力要求很高.

解决方案

- Q：如何处理资源滥用?
- A：资源精确计费

从虚拟机粒度到应用粒度

第二个问题,我们就想到了计费.其实计费一直都有,但通常都因为不够精确不了了之.最大的问题就是跨项目的虚拟机复用带来的计费难度.开发人员通常都很懒,比如数据库服务,他们往往不会为新项目搭建新的数据库,而是在以前项目中用于数据库服务的虚拟机上添加.有些时候由于业务需要,一台虚拟机很多个开发者都有帐号,这让情况更加混乱.我们需要更精确的计费.

IaaS平台→PaaS平台

私有云→公有云

为了更好的解决前边提到的问题,我们开始了第二阶段的实践.主要的变化是,我们从iaas转向paas,从私有云转向公有云了.

为什么要做PaaS平台

- 应用(项目)粒度的精确计费
- 节省开发成本

做paas平台的原因有两个,一个是为了实现之前提到的精确计费,另一个其实也是之前提到的,为了解决iaas对开发者能力要求提高而带来的人力成本增加.这个稍后会详细解释.

为什么要做公有云

- 开放和合作成为主流
  - 类似新浪乐居的合资公司
  - 类似新浪玩玩的联盟平台
- 开放平台与众包模式兴起

另外一个变化就是,我们开始做公有云了.最主要的原因,还是公司业务和行业趋势的要求.新浪有了乐居这样的合资公司,有了玩玩这样的联盟平台,后来还有了微博开放平台.这些都要求我们能安全的向合作伙伴和第三方提供我们的服务.

实践思路

- 以应用为粒度
- 以虚拟货币“云豆”为核心的计费体系
- 面向公有云设计，其他部门也是“客户”
- 平台+服务的思路

然后我们开始了第二阶段的实践.有三个重点,一是以应用为粒度的,以虚拟货币云豆为核心的计费体系;二是完全面向公有云设计,把公司其他部门看作内部客户,一样受云豆和权限的控制,第三是采用平台+服务的思路,使技术团队分工更加明确,更加高效.

平台+服务

先说下平台+服务的思路.以往我们开发应用,通常是按业务逻辑分功能模块,每个工程师负责不同的部分.而平台+服务的思路则是将一些通用的,难度比较高的功能服务化,交由专门的服务开发团队来开发,而工程师只负责将服务和业务逻辑通过代码粘合起来.就像一个电视机的流水线,工程师只需要负责将芯片安装到主板上就可以了,而无需关心芯片内部的实现.这一方面大大降低了对开发者的要求,另一方面也使第三方为sae提供服务成为了可能.

节省开发成本

使用SAE前的开发团队

初级工程师 8万/年 Web应用开发 X4

高级工程师 15万/年 Web应用开发 X2

系统架构师 15万/年 系统架构和运维 X1

使用SAE后的开发团队

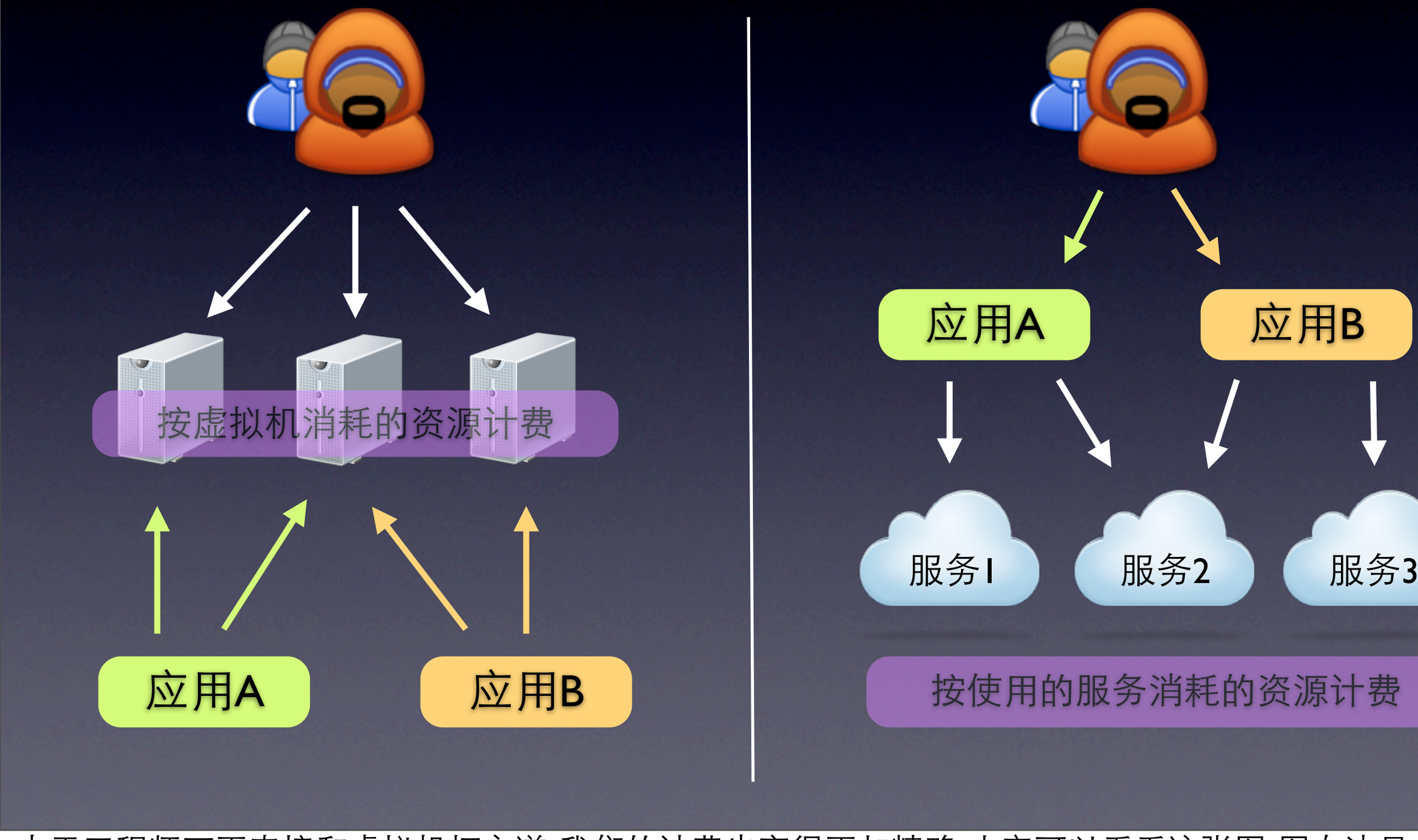
初级工程师 8万/年 Web应用开发 X4

高级工程师 15万/年 服务开发 X0.5

系统架构师 15万/年 系统架构和运维 X0

按照这个思路,我们来对比下开发成本.在不使用sae的情况下,一个中型的web项目,需要4个初级工程师来完成业务逻辑,需要2个高级工程师来进行应用架构,比如无单点设计等,需要1个系统架构师来负责机器的正常运行.而如果使用sae的话,机器完全由sae的运维团队负责,无需系统架构师;工程师进行分工,难度大的功能由高级工程师实现成服务,初级工程师将服务和业务逻辑通过代码粘合在一起.服务开发的工程师很可能工作量不会饱和,开发出来的服务会被其他项目公用,所以算0.5人.左右比较,我们发现在sae上分工更细.工程师能专注自己的领域.工作效率更高.

应用粒度的精确计费



由于工程师不再直接和虚拟机打交道,我们的计费也变得更加精确.大家可以看看这张图,图左边是之前我们描述过的混乱状况,而图右边,是在sae平台上计费的情况.工程师直接面对应用,应用使用多种服务,我们可以给每个应用出对应的资源使用报表.

抱歉,插播一页广告~

- 微盘团队正招聘PHP和Android高手
- SAE正在招聘windows客户端高手
- 简历投送邮箱 easychen@gmail.com

加入我们

成为社交网络+云计算+移动互联网领域专家

如果你想要更好用的SAE,如果你希望在任何能上网的设备上都能使用微盘,请转告你认识的技术高手,我们很想他/她 >///<

实践效果

- 服务器利用率再度提升约40%
- Web项目开发速度提升约100%
- 可提供每个应用每分钟的资源消耗数据

最终的实践效果如下,服务器利用率再提高40%;项目开发成本降低的同时,开发速度却提升了100%.我们可以提供每分钟每个项目的资源消耗报表了.结果是相当满意的,不过又开始遇到了新的问题.

存在的问题

- 现有项目迁移成本高
- 如何吸引并留住更多的开发者

在平台逐渐成熟后,越来越多的老项目也开始希望使用我们的新平台;当我们开始对外推广我们的公有云时,也有和多客户希望能直接使用已有的开源项目, 项目的迁移开始成为一个大问题.同时,当我们从iaas平台变成paas平台后,如何吸引和留住开发者,如何培养起完善的生态链条,成为决定我们未来的重要因素.

解决方案

- Q：如何解决项目迁移成本高的问题?
- A：尽量兼容原有接口，提供迁移工具

其实当时设计paas平台的时候,我们有一些冒进,比如我们当时决定不支持迁移项目.你要用我的平台,你就得按照我的规则,相当的霸气外露啊 (笑).然而,当需求扑面而来时,我们发现改变用户习惯,是一件非常非常困难的事情.在客户日复一日的抱怨中,我们开始反省自己.云计算本来是为了让开发更方便,为什么反而增加了开发者的成本呢?有没有一种方式,可以在新的架构和旧的接口之间,做到一个平衡呢?于是后来就有了代号为no sae的行动.它的目的就是尽可能的兼容php的原生接口,让开发者感觉不到在sae上开发的不同.

解决项目迁移成本高的问题

- 数据抓取，REST→ Curl
- 云存储，REST→ PHP文件操作函数
- 图像处理，REST→GD函数
- 数据库，MySQL→ RDC，接口不变

  亚马逊提供关系数据库服务RDS

  Google的数据库服务开始提供SQL接口

在这个思路下,我们开始替换php的原生接口.比如我们修改了curl,将其转向到我们的数据抓取服务,开发者不用修改任何curl的代码就能使用;我们通过wrapper技术使开发者可以只修改文件路径就可以通过原有的文件函数操作云存储;我们把图像处理服务换回了GD。可能有朋友会问,你这样还算云么,和原生环境有什么区别?我们的服务虽然接口和原生接口保持一致,但是背后的实现却都是我们自己重新实现的,具有分布式和可扩展能力,这些原生环境是不具备的.

解决项目迁移成本高的问题

- 迁移工具*
  - 分析原有PHP代码，提供转换建议
  - 原有数据的分析和批量导入

* 开发中

可能还有人问,你们的接口可以和原生接口完全兼容么?我们的愿望是如此,不过需要时间和技术实力去实现,现在有一些接口是不完全兼容的,所以我们正在开发迁移工具,它可以自动分析你的代码,然后提示你哪些地方需要修改,让开发者迅速的完成迁移工作.

解决方案

- Q：如何吸引并留住更多的开发者?
- A：降低开发者投入，增加开发者收益

硬件投入，
开发时间，
学习成本。

所有这一切,都是为了降低开发者的投入.包括硬件投入,开发时间,学习成本等等.我们觉得要吸引和留住开发者,最终还是靠平台本身的”性价比”,我投入多少,你回报多少.将开发者的投入降到最低,将开发者的收益提到最高,这样的平台想不火都不行.在降低开发者投入上,我们已经进行了很多实践,也取得了不错的效果,但在增加开发者收益上,我们还没有来得及去实践.

如何增加开发者收益

- 和开放平台深度整合
- 应用商店模式
- 项目外包平台

→ SaaS

一些尚未实践的思路

这里有一些尚未实践的思路,主要着眼点是帮助开发者将技术变现.比如我们可以和开放平台整合,帮助开发者通过社交应用挣钱;我们可以借鉴应用商店的模式,为开发者提供销售渠道;我们也可以建立基于sae的项目外包平台,帮助开发者将技术换成钱.具体靠不靠谱,需要实践后才知道了.

云存储

另外再说下新浪的云存储.新浪云存储主要包括两部分.一部分是位于iaas层次的,类似亚马逊s3的sina storage服务,另一部分是位于saas层次的,提供多终端数据同步的微盘.

云存储

Web网站 | PC客户端 | iPhone/iPad | Android | 第三方应用…

微盘 API

微博API | Sina Storage

Sina App Engine

新浪邮箱
共享资料
科技下载
……

sina storage目前已经逐渐成为新浪的统一存储,被广泛使用于邮箱,共享资料和科技下载.云存储可以通过文件排重,实现重复文件的瞬间”上传”,带来惊艳的用户体验;同时多个项目之间也可以因此节约存储空间.运用类似的机制,还能实现对病毒文件和敏感内容的云标记功能,任何一个项目的监管人员将一个文件标识为病毒,其他项目将能自动得到标记后的结果.我们在sina storage和sae上,建立了微盘项目.其目的是让每一个新浪网友都能享受到云存储的便利.微盘以open api为统一入口,支持pc,iphone和android等终端的数据同步.

微盘的经验教训

- 安全和备份是一切的基础

  用户的错误就是我们的错误

- 云存储本质是随时随地访问数据的能力

在微盘项目上,我们有一个教训和一个小小的心得.首先是安全和备份是一切的基础,不但要备份用户现在的数据,即使用户自行删除了数据,我们也必须提供一定时间的备份,这样才能在用户犯错的时候,帮他找回数据.另外的心得是,云存储的本质就是随时随地访问数据的能力.什么安全备份,其他的存储也能做到.而我们眼中的云存储是,你可以在任何能上网的设备上访问到你的数据,电脑可以,手机可以,电视可以,游戏机也可以,甚至将来,你家微波炉都能往微盘里边存你刚烤好的鸡翅照片.这是我们的愿景,希望通过我们和各位的共同努力,将这种看起来很科幻的场景搬进现实.

More? Try it!

http://sae.sina.com.cn

http://vdisk.me

我的分享就到这里,关于sae和微盘各位可以到以上网址了解更多.

Thanks

反馈和交流

新浪微博@Easy
Easychen@gmail.com

对我的分享有疑问和建议的朋友,希望加入sae和我们一起并肩战斗的朋友,还有希望和sae进行合作的朋友可以通过我的新浪微博和电子邮件地址和我沟通.谢谢大家.