



我们听过许多分享，各大公司，巨头，用数以万计的服务器、上百条产品线、几千名研发、几百T的数据、几个亿的订单。这些公司在这样的业务催动下，已经有相当强大、深入的技术体系：)

但现实中，除了这些巨头，我们还有许多公司，服务器级别在千台、产品线相对集中，核心业务就那么一两个、研发人员几百人规模。这类网站是很大一段的长尾，在我们互联网生态圈中占据相当大比例。



艺龙-OTA电商技术路上小跑步



@Syupe

所以今天，非常高兴能和大家一起分享一下像艺龙这个规模不算海量、业务相对集中、以在线旅游为垂直方向的典型电商，这两年来在技术上做的一些小步快跑的事情，以及我们为什么要这样做，它的背景、原因是什么、又对艺龙带来了什么样的改变和收益。



1x products

2 core business

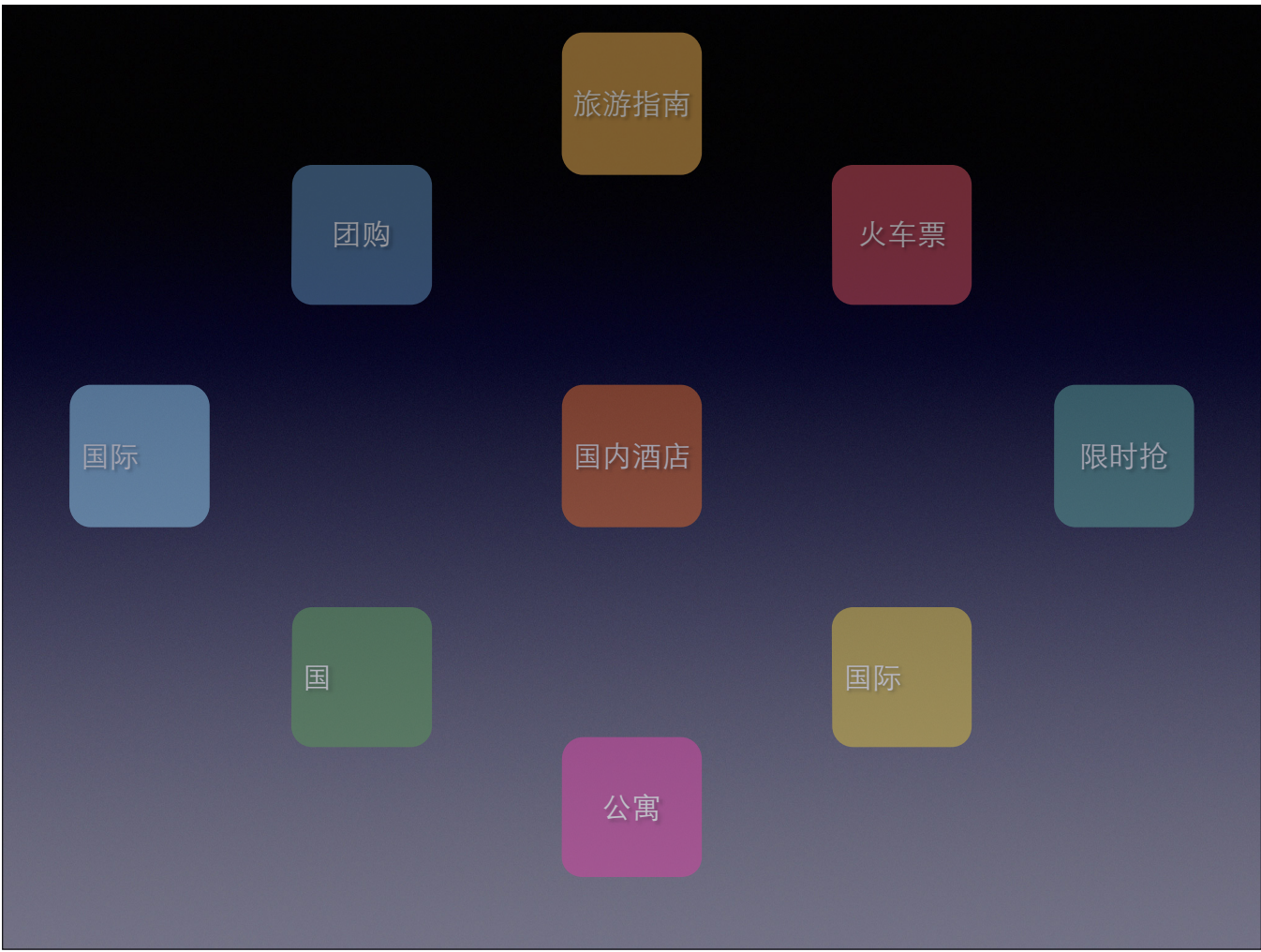
2x ' 000 + orders

4K+ Servers

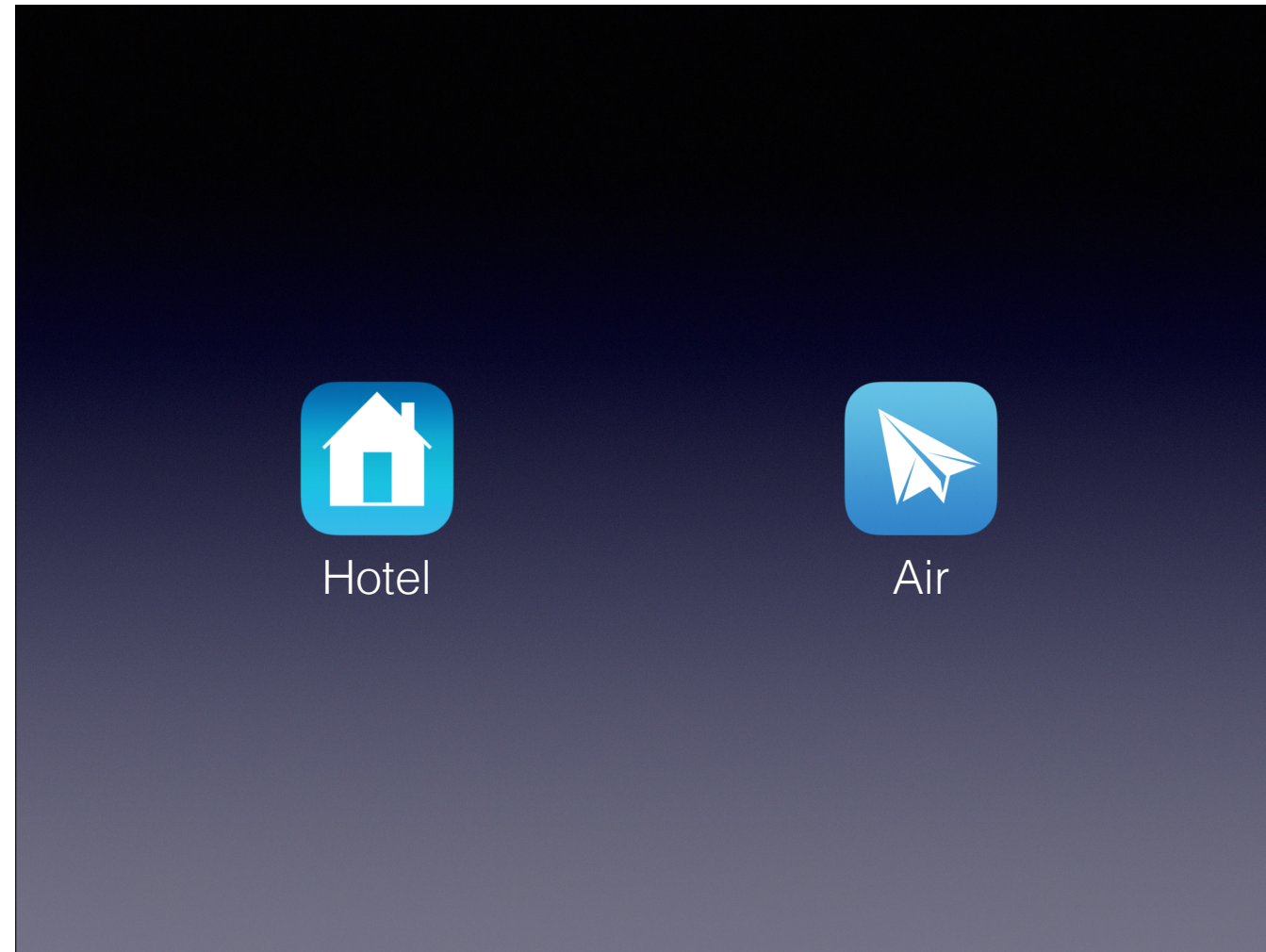
300+ Developers

服务器是包括虚拟机的数量。艺龙目前大量使用VM虚拟机。

研发包括了所有技术团队，包括运维，包括测试，包括数据分析



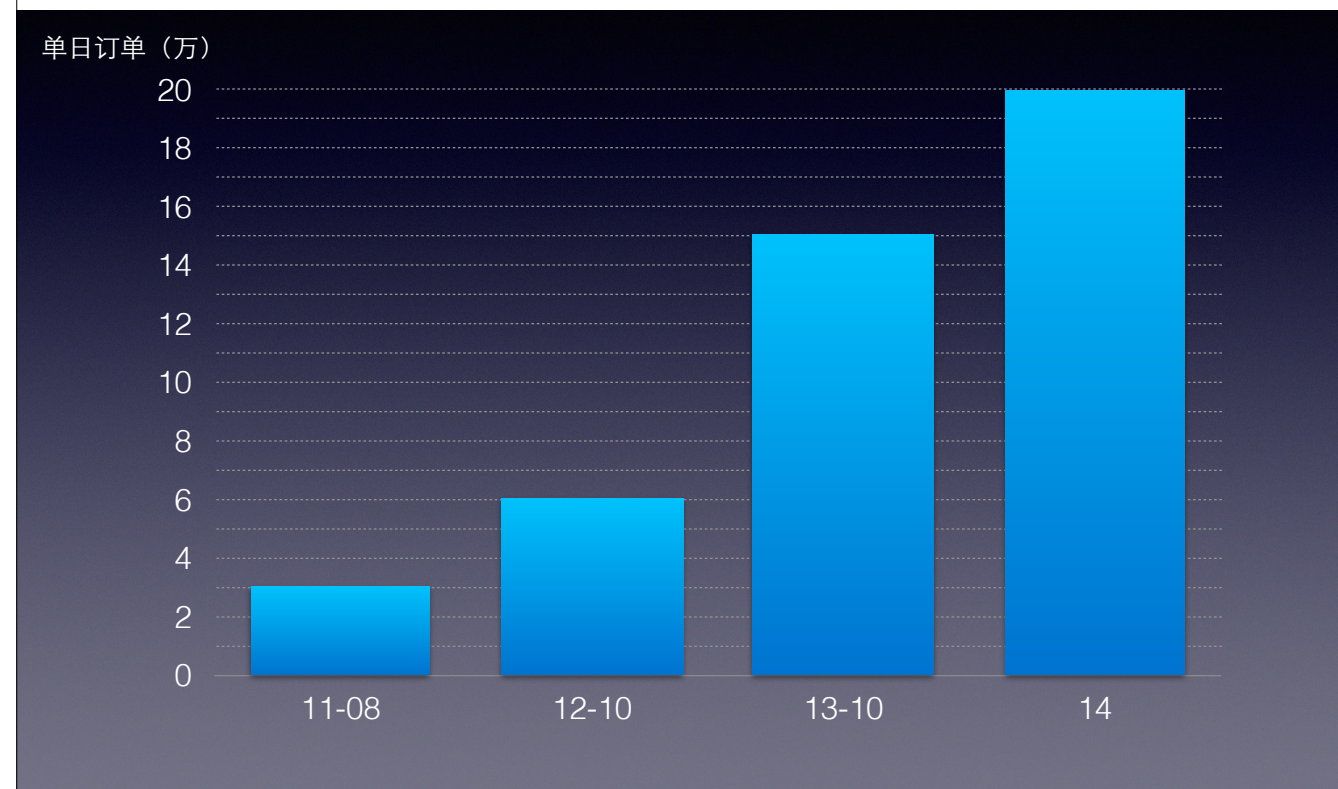
艺龙的业务主要是以在线旅游服务方向为主，到今天为止，一共拥有10余条产品线。涵盖国内、国外出行住宿的各个方面。



虽然这两年扩张了许多新的业务，但是这些扩展业务的核心，仍然是围绕在酒店和机票两大主题市场上。同时艺龙也搭乘着全球互联网发展的东风，业务规模的发展也非常迅速。下面是我们近三年来的业务发展状态。

2011 2012 2013 **2014**

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec



从06年到11年，艺龙用了五年时间使得在线间夜量突破3W，到6万时只再花了1年，13年再翻一番，到今天为止仍然在保持着较高的成长。



在高速的业务增长同时，业务形态也在持续地发生着变化、要求技术脚步需要不停的往前扑，从11年到现在，我们经历了例如团购模式、移动的兴起（从11年底的不到10%，到现在的早已超过50%）、支付方式的变化（快捷支付、各类移动支付）、云和大数据。这些业务、技术方向的不断更新调整进化，使得公司的技术体系面临很大的挑战：如何才能更快？



原始阶段，更像是一个传统软件公司而不是一家互联网公司。

基于Win平台，开源系统使用较少。

测试流程、运维流程以人为主。（手工传FTP上线，缺乏工具）

没有敏捷，没有自动化运维，人员技能单一（黑盒测试、手工运维）

甚至还有一些跑了十年以上的ASP老程序

问题：

1、技术交流不顺畅

2、解决方案单一，无法与社区知识共享

3、招人困难



从12年开始，技术体系开始发生全面的转型。开始了漫长且阵痛的过程。

在两年的过程中，经过所有工程师们的努力，系统平台绝大部份已经迁移到Linux

底层开发转向C/C++、Erlang，

业务开发转向Java、PHP

数据库迁移往MySQL

大量使用各类成熟简单的开源软件

实施敏捷、推广自动化运维

构建一系列基础技术平台

具体做了些什么呢？挑几个典型的实际项目与大家分享。

Skymap

Fast-Env Building

Distributed Library

Search

Pavo

会用五个系统实例来介绍，保证干货。

- Deploy
- Task Controller
- Monitor
- Log Collection
- Login Security
- Resource Location

定时关联任务 3,000 次
执行部署服务 5,500 次
收集日志 300 GB
采集状态数据 230G
分析 2,260,000,000 次

Everyday

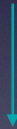
Skymap是我们的自动化运维体系

What's the difference ?

它本身不是一个服务，而是一套标准化体系，在这个体系里面，是多个互相关联打通的应用组合，这套体系是根据艺龙的规模、特点，定身打造，并且同时支持Win和Linux两套操作系统。

没有使用开源运维系统的原因是因为当时它们对Win的支持还不友好，另外它们虽然在各自的领域做得很好，但是要打通成体系很困难。

Leo: Controller



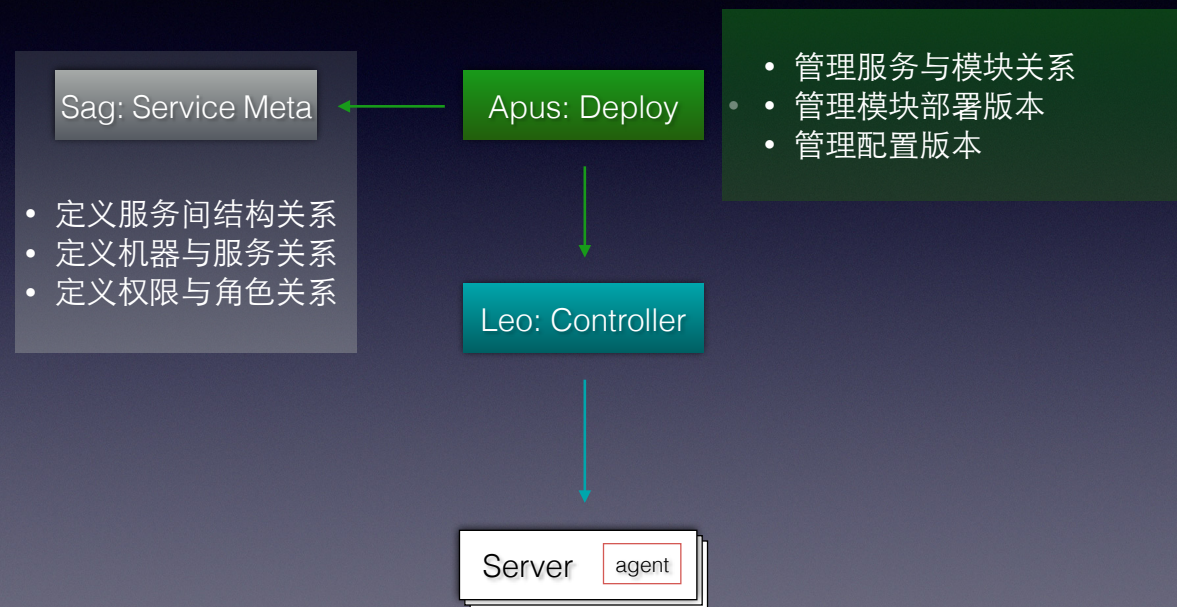
Server agent

- 操控机器入口
- 并串行、暂停点等调度逻辑控制
- 结果收集反馈、回调
- 权限控制
- Erlang OTP 构建
- RESTful API

4'000+ Servers Installed

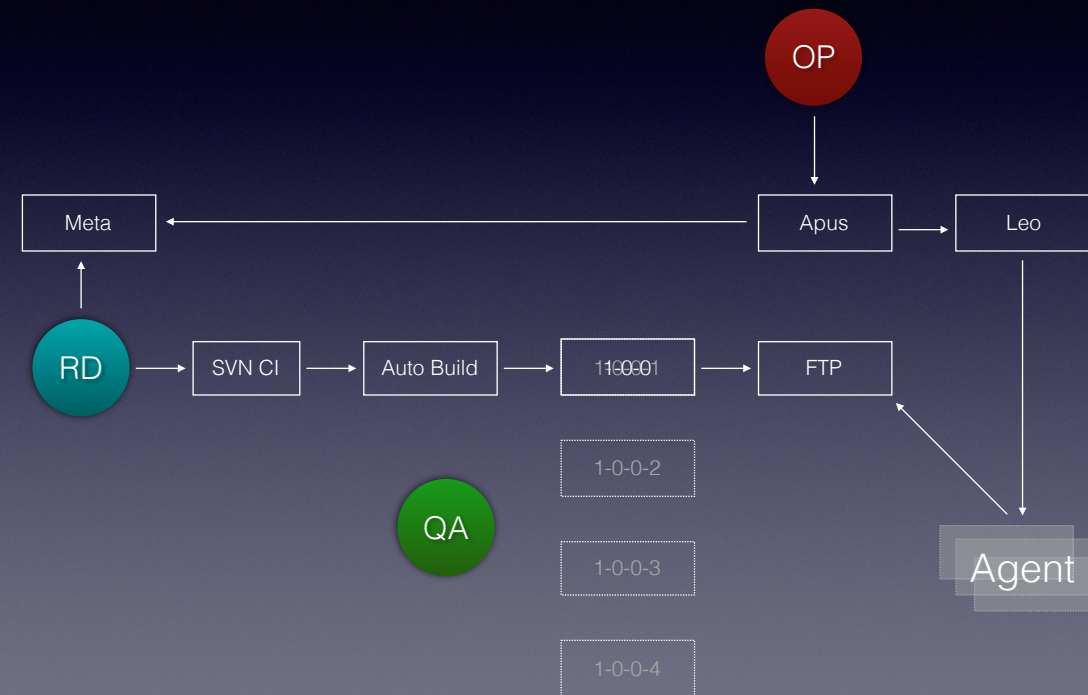
这套系统的核心，是由一个名为Leo的控制系统组成，这套系统在艺龙的全部机器上部署上代理客户端，以及一个高可用的总控调度中心来完成。其主要实现PPT上所描述的那些功能。

得益于Erlang-OTP，这套系统从12年至今，已经连续运行近3年无故障。



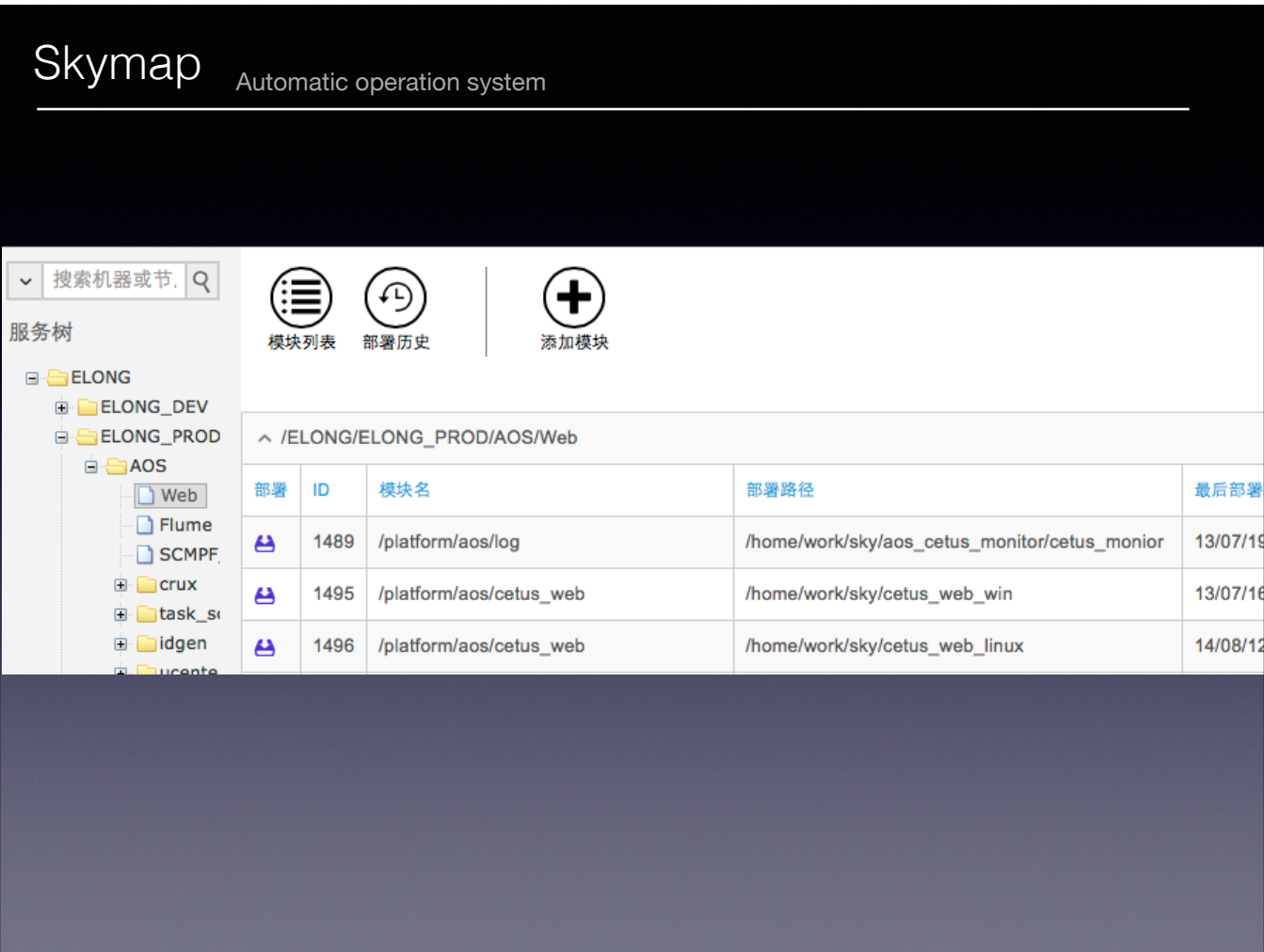
利用这套控制系统，我们在外围做了一些扩展模块来实现各种功能，先是梳理和抽象了艺龙的各种服务、模块之间的组成、依赖关系，开发了一套名叫射手座的系统，用来存放上述各类关系的元信息。再来有一套名为雨燕座的部署系统，用来执行自动化部署。我们期望整个部署过程像燕子一样轻快灵活。我们接下来看看这套系统。

Apus: Deploy



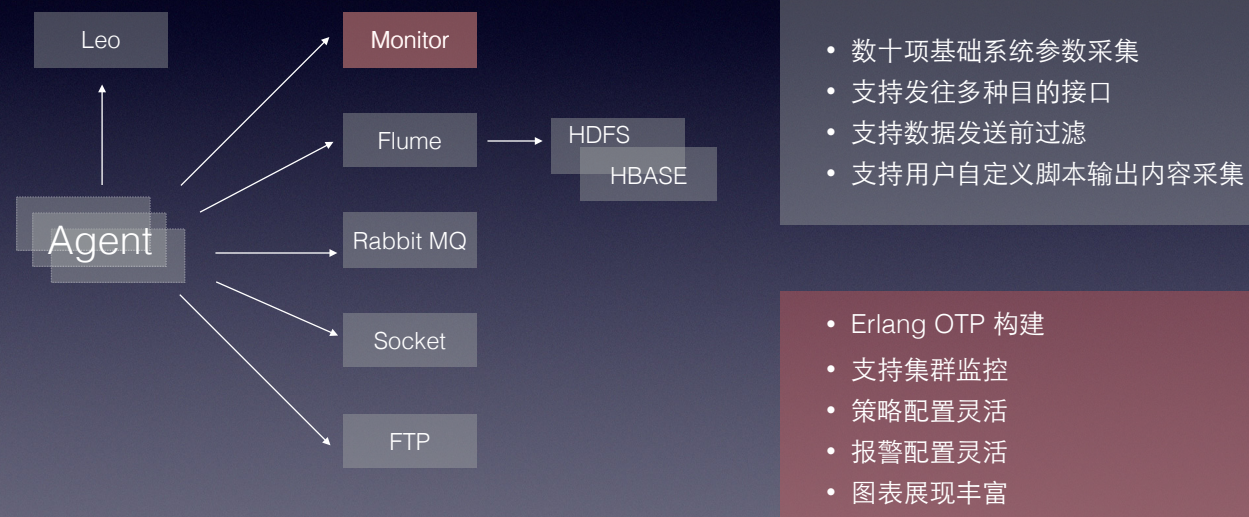
与Puppet不同：控制部署节奏、并串行逻辑、灰度部署、前后置等。并回报状态供业务端操作。

通过这套系统，我们实现了从CI、打包、发布、部署、异常回滚、部署历史记录的全过程。上线过程基本上喝咖啡。



这是我们Web端的实际截图。

目前这套部署系统已经覆盖我们全部的产品线。运维同学再也不用上线上线半夜哭，妈妈叫吃饭也不用等了。



再回到单台机器上：

在单台机器上有我们的执行代理。这个代理除了执行调度机器上的任务外，还以插件形式开放任何可执行应用。我们在实际工作中，还利用它采集机器上的各类数据。这些数据支持以多种协议方向发往各个不同的目的端。（见图）

Scutum: SSH Security

Pyxis: Resource Location

Hor: Task Scheduler

.....

除了监控、上线以外，我们还有一些其它的组件，例如

Scutum：根据权限角色控制登录线上机器、su work root，以及对操作内容做审计、报警。

Pyxis：类似ZK，除了用于传统的配置管理，也帮助我们管理上下游之间的关系。现在用于我们的资源定位和配置中心。

Hor：分布式任务调度中心，基于Erlang构建，最底层会调用Leo控制系统（多个系统是打通的），它自己用来处理任务的执行时机、执行范围、上下游执行关系，以及异常处理及报警。

30' Minutes Building Full Sites

有了这些基础平台以后，我们开始着手解决下一个目标：背景

电商类项目，从上至下环节众多，有时一个需求下来，很可能会涉及到多个跨部门之间服务改动、调用、联调。

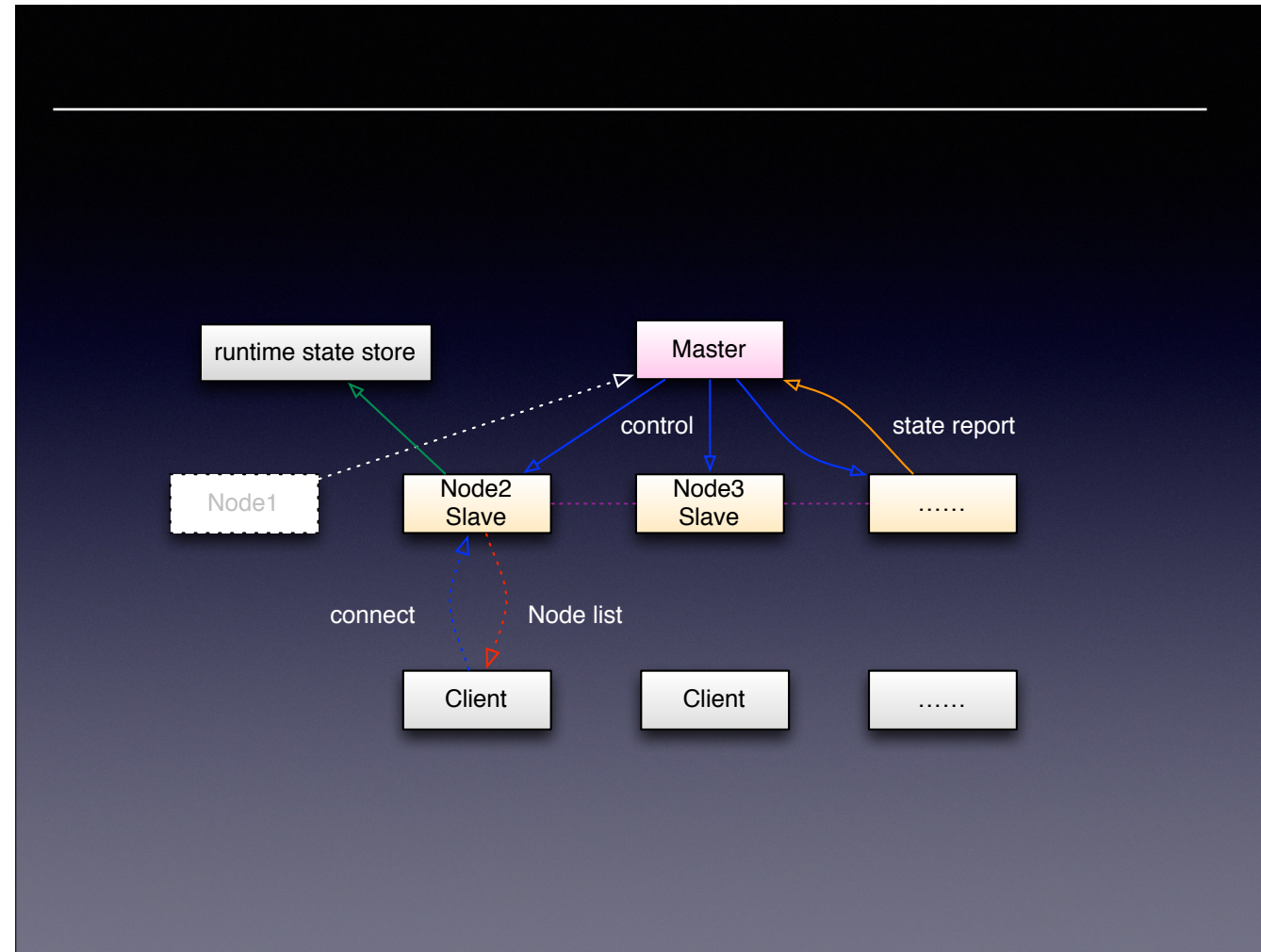
我们为了支持这类复杂联调项目的环境快速搭建、甚至多套环境并行，要求实现在任意机器群组上，都能实现30分钟内的全站的部署。得益于Skymap对各种服务做了抽象、定义，加上服务树树型节点隔离不同环境的配置、数据差异，借助于控制系统、初始化系统的开放API，使得这个复杂工程得以实现。

-
- Why Not ZK ?
 - Based on Raft
 - C/Java/Go/Erlang/Python library
 - How to use it ?
 - Who used ?

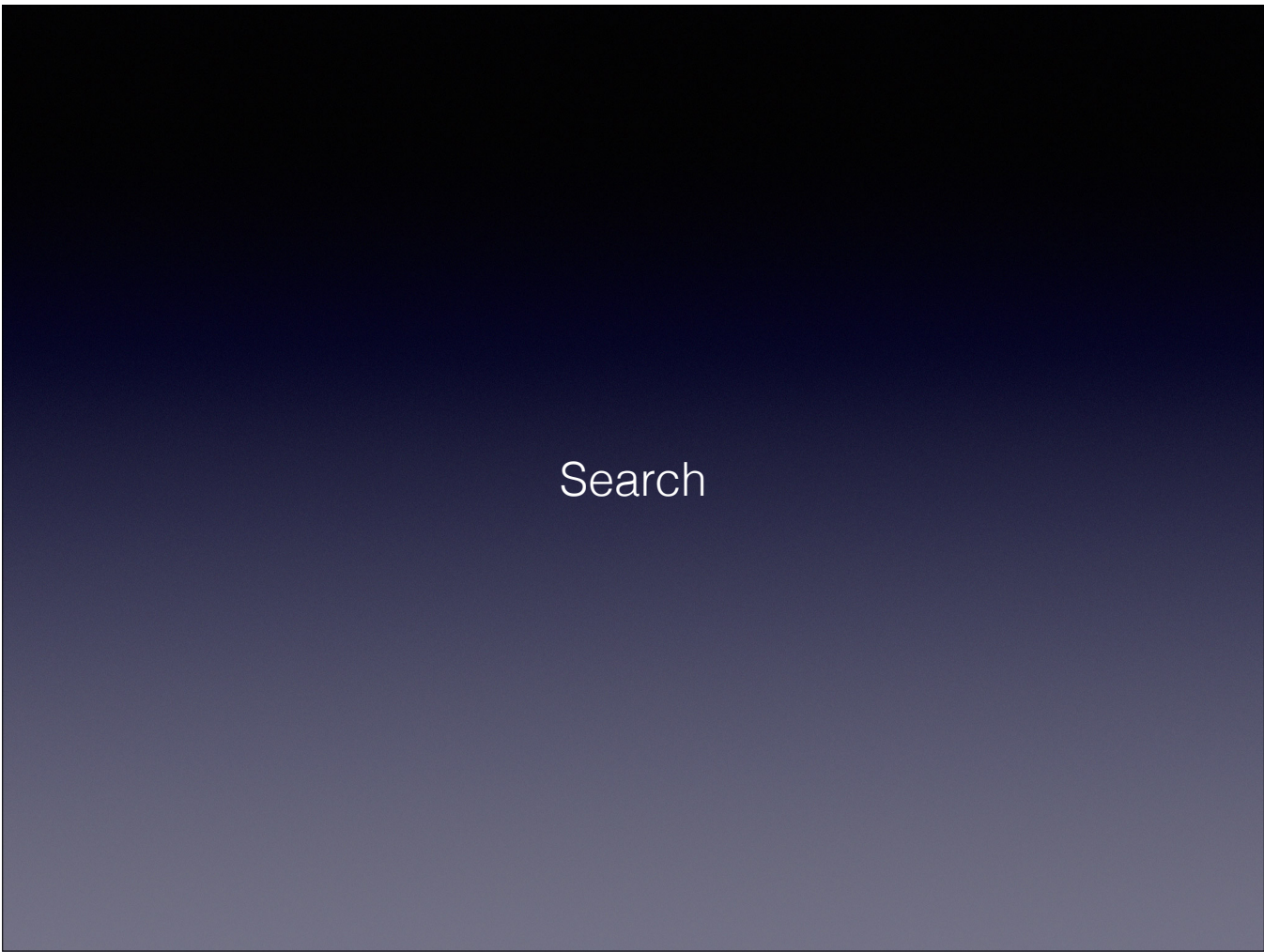
之后介绍一下我们定制的一些基础库，我们大量利用开源，除了以服务形式提供的开源软件外，我们也使用了许多业界广泛使用、成熟的库或者框架。但是仍然有一些是我们自己开发定制。11年艺龙出现过一次IDC级别的重大事故，导致全站服务中断26小时不可用。从那个时候开始，我们对高可用及分布式的需求提上日程。

为什么不是用ZK？ ZK提供选主服务，以及1M以下的容量存储，不能满足全部场景。我们更多的将ZK用于一些读写不是那么密集的配置元信息存储。最初我们的库是基于paxos的，但将paxos应用在工程上，需要挑战的细节还是比较多，2013年5月raft paper发布后，我们也及时跟进，发现这个协议在工程上处理起来会更容易，于是基于raft协议编写了包括选主在内的强一致性库。并且为了解决raft无条件选主在网络抖动时可能带来的不必要选举，又做了一些工程上的优化。

该套库提供了写、读、日志、快照等少数几个回调接口，应用只需按要求实现这几个接口，背后实现自己的逻辑，即可实现全局强一致性的写入要求。得益于这套基础库，我们很轻松的实现了类似全局ID生成器这样的系统。代价仅仅是一个工程师一天的时间。也使得之前提过的控制系统、任务调度这样的高可用要求的应用，以很低的成本改造成了分布式应用。



例如前面讲过的控制系统，这是最新的结构，原来很简单，直接由数几千机器连往一台控制主机，后期因流量增大，做了HASH，然后是主备的方式来做灾备。后期利用这套库很方便的转换成了高可用结构。

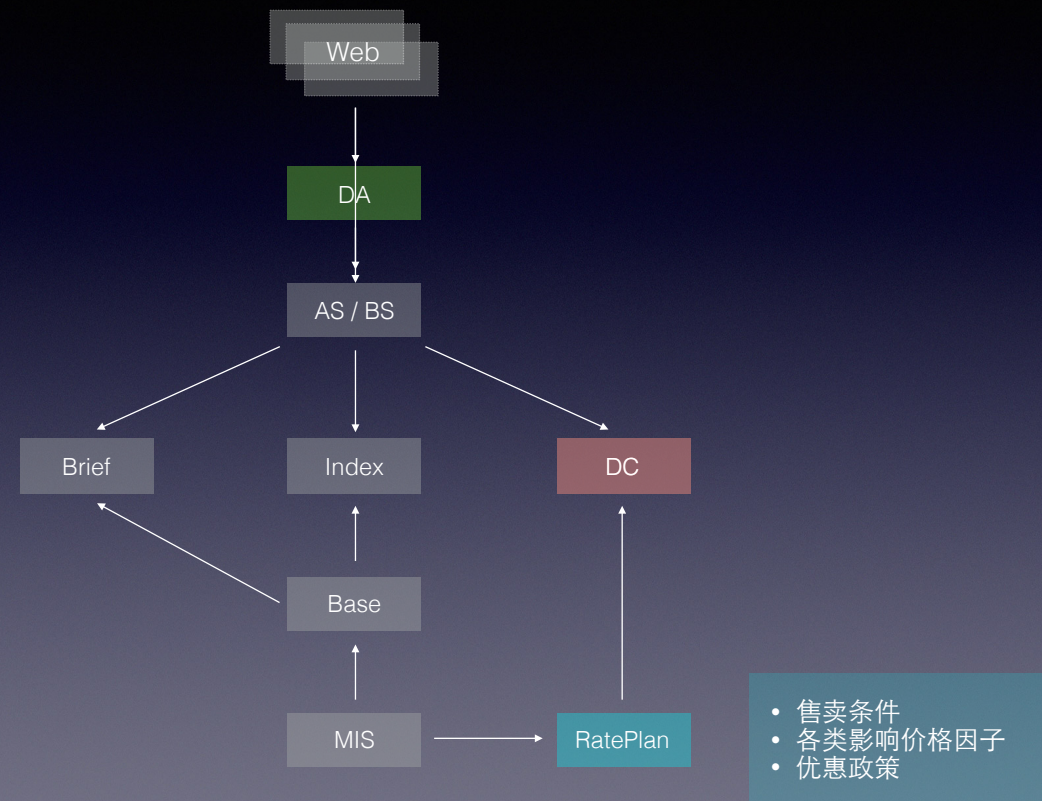


接下来讲讲搜索。

电商类的搜索有它自己独有的场景、问题。

艺龙原来的方式：没有太复杂， 程序读数据库算， 程序做结果缓存、业务规则复杂后， 程序预跑结果放缓存、再后来利用一些开源的分词、搜索系统做检索。

Search



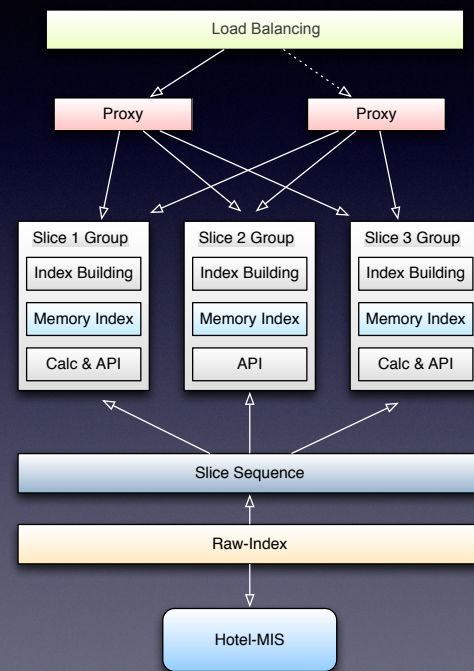
13年底开始，我们尝试构造自己新的检索架构。大体结构跟通用搜索很类似。 MIS—>BASE—>建库，索引—>摘要—>BS(拉链) AS（归并）

Brief库后面是MangoDB， Base到Index之间的分词是定制的；

和传统搜索不同的是，摘要服务的结构化（字段）要求比较繁杂，各个场合用的不一样，我们做了动态字段提取。

另外，价格计算在搜索中，也是电商系统面临的很特殊的问题。RatePlan—DC

对于移动兴起后的语音检索，第三方做了语音转文字，但对于Query的意图识别，在收敛的领域上可以做得更好。



- 全内存索引 拉链类型: HashMap + SortedSet
- 索引读写双Buf切换, 解决读写互斥, 保证数据原子性
- 分组、分片, 解决压力、高可用、并行性能

10'000 产品价格计算 / 单组

tps
50ms

qps
600

DC是一套全内存的实时计算引擎, 用于强力快速的计算库存和产品。

搜索系统做了这样的改造后, 性能得到了极大的提升,

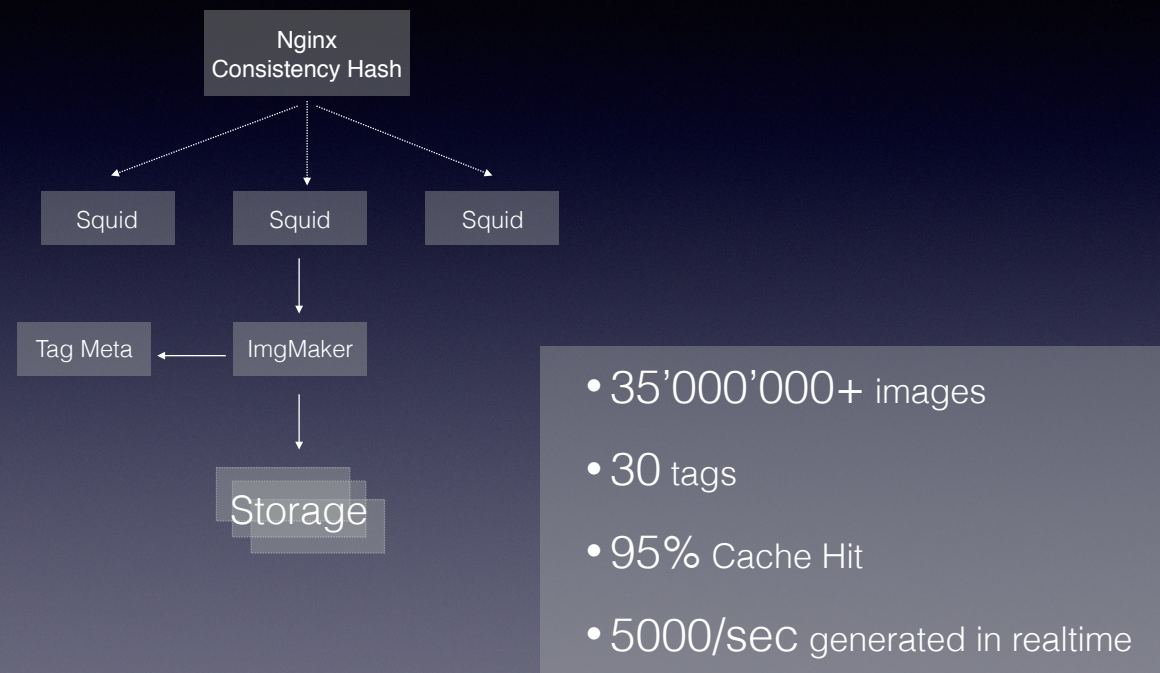
我们已经在国际酒店 (试用非常快)、H5全流量启用, 部份Mobile渠道小流量测试, 并且将在Q4以前将所有的搜索都迁移到新的平台, 欢迎大家来艺龙, 用更快的速度、预订心仪的酒店。

我们的原则: 能快就快, 少做缓存。有实时性要求的地方完全实时算。



按下来是我们的Pavo系统,Pavo是Skymap中的一员，美丽的孔雀座，我们用这个名字来命名图片系统。主要用于产品的图片展示。

诞生背景：各种页面场合，对图片要求不同。移动设备增多，各种屏幕适配。最初我们的MIS图片管理是为每一次上传生成各种规格的缩略图片，后来随着网站的多次改版，图片设计的大小发生调整。我们又重新生许多各种批次大小的缩略图片。后来进入移动后，屏幕尺寸种类繁多、在流量与适配下找到平衡，我们又生成了更多不同的缩略图。实在太累了。



Pavo是一套图片规格动态生成、图片缓存、存储的高可用系统。

为什么实时：计算流量、缓存命中率、生成时间、冷热分布

最终认为是划算的

容灾处理：集群放在多个CPU低消耗型的机器上，生成不过来时，自动选用最接近的尺寸返回。

squid缓存时间设置在15分钟，超时后会去询问Meta信息，如果meta未做修改，则直接以304返回，不会重新加载。

<http://pavo.elong.com/ipad/0ae9f.jpg>

```
ipad: {  
  width:800px; height:600px;  
  align:center; vertical:middle;  
  wm: true; dm: true;  
}
```

Tag: 用于定义图片的规格，动态生成图片，当修改Tag后自动生效

另外，Tag除了显示的定义外，也可以跟HTTP-Header信息做动态绑定，根据请求头来决定大小。而这一切都是可编程接口的。

这套系统已经在我们的国际酒店频道运行，预计下个季度将会全面替代全站的图片系统。

Zhizi: Docker

No Cache

Future 2015

No Windows

Personalised recommendations

还有许多系统做得不好，需要慢慢演进，这是一个长期且还将痛苦的过程

利用Docker，逐步取代VM；再与监控、调度等系统结合，完成混布，提高机器利用率，实现15年服务器零增长。（有过期机器置换）

Nocache：尽量减少cache服务，cache存在的必要性在于接口速度不一致、调用频率不一致，因为第三方、业务特性原因必须使用Cache的该用就用，但是对于内部系统调用，能不用就少去用，压榨开发者把单体速度提上来、或者把程把架构做得可以方便的横向扩展以增加性能。

旧有系统全部迁移，从Win转向开源平台。数据库还有一部份SQL Server也要转。

在新的搜索平台上，做更好的个性化推荐。

Summary

- 1、垂直类网站，尤其是电商，因为领域收敛，很多大公司在通用平台上难以做好的事情，可以在自己擅长的地盘做得更有效果（例如意图识别、例如个性化结果）。所以事事不必尽学大公司，一定要有自己的独特解决之道。不要重复造轮子，但小轿车总不能直接用大卡车的轮子。
- 2、开源系统，尽量选择成熟、普及、简单、结构简单的，思考下自己有没有必要用，不要为了迎合开源而选择开源，Hold不住的就别用。思考下业务的依赖度、投入的人力成本，权衡下收益。
- 3、尽量减少服务层次，把每一个服务做小、做简单，这不仅仅是为了维护方便，而且在业务变化要求调整时也船小好调头。
- 4、把每一个责任模块做到极致，大家不要觉得架构图上越多越美，实际上架构上每加一层都是不必要之恶。如果能把每个模块职责单一，做到极致，架构上会减少很多不必要的地方。先练内功，再做架构。拿把菜刀都砍不死人的话，就不要着急去练北斗七星阵。



欢迎加入艺龙技术团队

艺龙网-酒店预订更轻松

我今天要分享的内容就到此为止，谢谢大家还没有走。

顺便打下广告：

艺龙技术团队求各种英雄，QA.运维.自动化.分布式.Erlang.Golang.搜索方向。

有兴趣的请发邮件到 pei.yu@corp.elong.com