## 上云最佳实践——电商行业

# 一、故事的开端

那是2014年的12月中旬的某日，客户联系到我们。由于客户2015年2月机房机柜到期，所以想把资源迁移上云。客户联系到我们时，表示需要我们的支持。客户需求很明确，希望根据平台目前数据及特性，兼顾成本与性能给予最佳云架构/资源方案。

**客户背景：2006年成立，杭州某知名网上私人订制礼品购物平台。**

# 二、前期调研

**运维团队规模（4人）：**运维1人、运维架构师1人、网络工程师1人

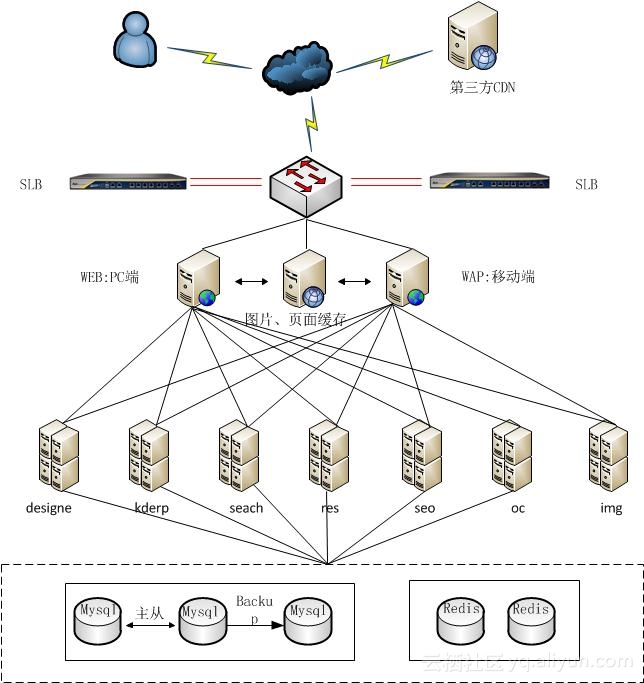
**客户研发/测试团队规模：**30人

**客户电信机房资源：**

* 3个机柜
* 40台左右硬件服务器（dell r410为主，其实两台16核/96G内存用于xen虚拟化）
* 200Mbps（独享）

**客户环境简介：**

所以从上图可以看出传统电商的架构环境



由于电商环境存在大量商品图片，所以CDN是必不可少。

服务器端，前端采用nginx+varnish作为二级缓存，主要减少CDN回源访问的压力。

后端业务系统名称designe/kderp/seach/res/seo/oc/img等十余项，采用的开发语言主要为Java、php、python。操作系统主要为centos为主，少量windows环境。

图片源文件等，主要通过nfs进行磁盘挂载共享，图片数据量2T+。

数据库缓存端主要采用redis作为数据库缓存，减少数据库压力。

数据库端主要为mysql，采用硬件服务器上面部署mysql主从。

# 三、确认合作意向

从客户角度来讲，有三大痛点：

1. 虽然云的确在成本、扩展、灵活性、快捷等方面有很大优势。但是，对云产品、云架构的灵活运用，是有一定技术门槛的。怎么样利用云资源设计出低成本高性能的架构，这是个经验性的技术活。
2. 客户没有7\*24监控响应中心，导致出现报警往往不能及时马上联系上运维，及立即响应解决，运维的7\*24无法得到保障。
3. 客户有四个运维人员，成本高昂也是最实质性的痛点。

所以通过洽谈，最终在12月底确定了合作意向（具体商务方面的细节不再这里概述）。我们为客户提供上云架构方案 + 上云迁移 + 7\*24监控 +7\*24运维服务（我方运维为主，客户运维为辅）来解决客户痛点。

# 四、上云迁移的挑战性

* **挑战1：时间短：**客户机房2月到期，并且在每年2月14日情人节是一年中的业务高峰期。由于确定商务合作的时间点已经12月底了，所以项目排期，我们需要在1月中旬（仅两周）完成项目的上云迁移、测试、及正式上线，预留两周作为观察过渡期。
* **挑战2：业务系统较多、技术环境较多：**通过梳理，客户有十余个业务系统。Nginx、varnish、tomcat、php、python、redis、mysql等技术环境较多，这远远增加了迁移难度。
* **挑战3：零配置文档、零规范：**其实说到这点挑战，我是很想吐槽的。很难想象，一个做了八年运维的系统，居然在运维配置文档、运维手册方面没有一份文档，仅仅有几张零碎的架构图。另外，在主机名、防火墙、配置文件规范方面更是杂乱无章。在迁移期间还遇到件比较搞笑的事情，忘记机房交换机密码，然后网络工程师亲自破解获取最新密码。这跟我们带来的迁移难度及挑战可想而知。

# 五、上云迁移

过完元旦后，2015年1月4号正式上班。来到公司(上海)，做了些简单准备，收拾好行李。我作为运维负责人，和两名架构师、1名DBA、1名高级运维、两名中级运维，在下午开车前往杭州进行项目周期为期两周的上云迁移。

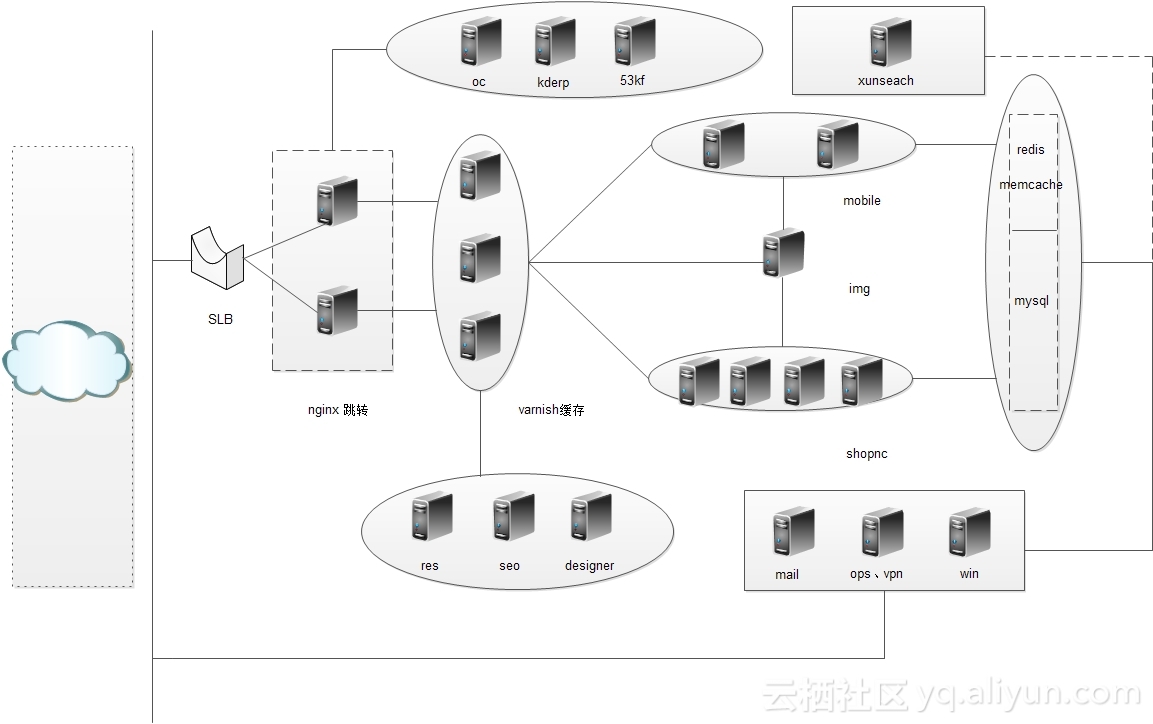
# 5.1、项目启动：2015年1月5日

这是来到客户公司正式开展工作的第一天。这一天中，我们主要确定双方参与项目人员的职责，制定项目通讯录。并且确定了项目实施计划，项目周期为12天。

# 5.2、系统架构梳理及评估：2016年1月6日—2016年1月7日

接下来进入是项目迁移实施期间，首先我们需要对原系统进行评估、并且制定云上架构。原系统评估的内容涉及到：系统架构、软件模块架构、业务架构、接口以及调用依赖关系、性能评估、上云迁移目标

云上架构涉及到的内容：上云后系统架构、软件架构、业务架构、性能目标、上云难点等等。其中云上架构图如下：



与IDC架构不同的是：

* 上云实践1：加入SLB保障架构灵活扩展性  在前端我们加入了SLB负载均衡。在原IDC架构中，域名解析到不同nginx+varnish上，然后进过前段静态缓存，然后转发到后端对应的业务服务器。加入SLB，将此架构变得更加灵活。即将所有域名绑定到SLB，然后转到后端nginx，通过nginx做虚拟主机等七层更灵活的控制。
* 上云实践2：采用TCP层SLB保障性能  在实践中，在面对高并发性能要求的场景。我们发现HTTP层的负载均衡相比TCP层的负载均衡，性能上面有很大差距。HTTP层负载均衡只能达到万级别并发，而TCP层负载均衡能达到几十万级、甚至上百万的并发量。所以在电商等网站应用中，SLB我们优先选择TCP层。
* 上云实践3：低成本高效率的按量带宽  在IDC机房，200Mbps的独享电信带宽，一年成本大概1Mpsb/100元/月 \* 12个月 \* 200 =24w 。而在云端，采用1Gpbs峰值的BGP多线SLB带宽，在带宽质量上面提升了几个量级。另外，带宽费用采用按量付费，这给我们大大降低了成本。
* 上云实践4：数据库优先采用rds，低成本高效率  在IDC硬件上采用mysql主从手动部署并维护的模式，给我们带来了后期很大维护管理成本。即我们要监控及维护主从状态，并且出现问题需要及时处理，保障业务对数据库读写的连续性。采用rds后，这一切问题都自动化解决了。即，对数据库主从的监控、备份、后期维护、故障切换等等，都是全自动的。

# 5.3、迁移方案：2016年1月6日—2016年1月7日

在进行系统架构梳理及评估的同时，我们同时开展了迁移方案的确认。即，如何将应用、数据迁移至云端。同时我们还确认系统割接上线的流程及对应的时间节点。在迁移方案中，我们确认了客户云上资源清单（23台ECS、两台RDS、一台SLB）及具体的服务器配置。

* 上云实践5：云计算的优势在于分布式  很多用户喜欢把单台云主机跟同等配置的传统物理服务器相比较，结果往往是吐槽云主机的性能是如何的糟糕。传统物理服务器，多核高频CPU等方面的性能，真的能把云主机甩上几条街。何为云计算，关键字在于“云”，即分布式是云计算最大的优势。所以在实践中，我们不要追求单台机器的性能要如何高，而是我们要通过分布式的设计思想保障业务的高性能。所以在此项目中，我们服务器的标准配置都是4核8G，也有大数服务器采用2核4G的配置。我们通过分布式充分压榨了单台服务器的资源使用，从而最大限度保障了最终的低成本（在后面详细明细了一下这块费用）。

在迁移方案中，图片文件迁移的方案有一定难度。一方面，线下图片数据目录的数据量有2T多，而线上单块磁盘只能最多支持1T的容量（当前官网单块磁盘支持32T）。另外一方面，2T的图片主要是小文件，数量特别多。怎么样把这些文件迁移到云端？

* 上云实践6：lvm在磁盘管理方面的应用  在云端迁移方案中，我们购买了四块1T数据盘（每台ECS最多只能挂四块数据盘），通过LVM逻辑卷虚拟成一块4T磁盘，这样我们在云端保障了大于2T存储数据量的冗余空间。官方是不推荐使用LVM。因为阿里云的快照主要针对的单块磁盘，不能针对几块磁盘同时快照。而LVM主要针对的多块磁盘（物理卷）的基础之上，抽象成为逻辑卷。LVM的读写针对的是逻辑卷，数据被分散存在至底层的物理卷（磁盘）上。如果某块磁盘数据损坏，如果通过快照恢复这块磁盘的数据，我们无法保障LVM逻辑卷整体数据的完整性。通过LVM主要是能够提升磁盘IO，比如需要购买100G的数据盘。常规配置，买一块100G的数据盘即可。但是我们也可以购买4块25G的数据盘，通过LVM虚拟化成为一块100G的磁盘。在功能性上面都能满足需求，但在磁盘IO性能上面，LVM至少能提升20%-40%的IO性能。
* 上云实践7：rsync在云端应用  怎么样将线下数据不停机实时的迁移到云端，rsync是文件增量同步迁移最优方案。只不过在此项目中，一方面数据传输要走公网，另外一方面数据量较大。所以我们初步统计下来，完成数据增量迁移至少需要一周多。所以这方面的数据迁移由于时间周期较长，为了避免影响整体迁移进度，我们需要提前进行。

# 5.4、迁移实施：2016年1月6日—2016年1月7日

二十多台云主机牵扯nginx、php、tomcat、redis、varnish等环境部署，我们通过自动化的部署手段来保障部署的最大效率。线上23台服务器环境的部署，我们半个小时内搞定。

* 上云实践8：域名备案要先行  上云的最后一步，是需要将域名的ip解析到SLB公网IP（或ECS公网IP）上。但前提是域名备案需要在阿里云上，不要到最后域名解析到阿里云上后发现域名被拉黑，业务访问被拒绝，才发现未备案到阿里云，这将会变得非常麻烦。所以我们需要提前通过阿里云进行域名备案，或者已经在其他供应商备案过，需要将域名备案转接入阿里云。
* 上云实践9：通过镜像提升云端部署效率  刚开始我们开了一台ECS，针对这台ECS我们做了运维规范方面的系统调优、安全加固等。然后我们把这台ECS做成了一个基础镜像，批量开通了22台同样环境服务器，在部署效率上面大大提升。
* 上云实践10：自动化运维工具的应用  对应软件的安装脚本，我们内部团队都统一存在在内部的gitlab中。我们通过ansible工具，定制对应playbook，推送对应的安装脚本到目标机器上。5分钟内搞定了对应Java、php、python等环境的安装。

至此，我们引来了迁移最为痛苦的时期。由于运维配置手册、运维文档的缺失，所以我们将应用代码部署到我们已经搭建好的环境中后，我们需要对每一项参数、每一个配置都要仔细调试。我们三名运维同学拉着客户运维人员、研发团队不眠不休整整一天一夜，完成了所有代码的调试、对应配置的文件的调试。至此，我们迁移工作完成了大半。后续核心工作主要集中在功能测试、性能测试及上线割接了。

# 5.5、迁移测试：2016年1月9日—2016年1月11日

此阶段主要为功能测试、性能测试，主要集中在客户的测试团队。

# 5.6、上线割接：2016年1月13日—2016年1月15日

上线割接前，需要做好客户及公司内部的维护通告。正式迁移的时候，由于系统、代码、文件都已迁移过去。加上客户数据库较多，无法做到实时迁移，所以我们采用了保守做法，停机迁移。迁移的最后一步是将域名解析至阿里云，这里在前面也提过，域名需要提前备案的。

到此是不是完成了最终迁移呢？其实还是没有的，虽然域名已经解析到最新的ip，当前万网的刷新最新的解析记录的时间周期最短也仅仅10分钟。但是我们没法把控的客户端本地的DNS缓存，即还会有部分客户还是访问到老的站点。所以完成最后迁移，我们还差最后一步：

* 上云实践10：nginx反向代理将老用户请求引流至阿里云  还依旧访问IDC机房应用的客户，我们在IDC机房前端nginx上做302重定向跳转，将依旧还访问IDC的客户引流到阿里云。值得注意的是，由于nginx是七层负载均衡，需要匹配域名。这里nginx的server\_name和跳转的链接配置的域名都是同一个，为了确保跳转的域名解析的地址为阿里云，我们可以在nginx所在服务器的hosts配置中强制将域名的解析ip设置为阿里云对应的IP。

# 5.7、项目交付及后期监控运维

后续便是项目交付，主要为文档的编写总结。此项目我们总共汇总了三十余个文档，主要包含系统软件架构、系统架构、迁移方案、运维实施配置文档、运维维护手册、故障处理文档、资源清单等等。

文档交付后，进入后续7\*24日常监控及运维阶段，这里不再过多概述。

# 六、上云前后的对比

写这篇文章的时候，我一直在脑海中搜索有没有一个上云的实践对得起“最佳”二字。对我本身而言，在面对成百上千的客户实践案例中。这个项目无非是我体会最深刻，总感觉千言万语总嫌少。一切尽在以下对比图中：

|  |  |  |
| --- | --- | --- |
|  | IDC | 阿里云 |
| 配置 | 3个机柜  15台硬件服务器（包含两台96G内存配置） | 23台ECS（4核8G、2核4G） 1台按量SLB 2台RDS(6000M/200G、2400M/200G) |
| 带宽 | 200Mbps/电信独享 | 1Gbps/BGP网络 |
| 成本 | 人员成本：15w/人 \*4人= 60w 资源成本：8w/年\*3个机柜=24w      100元/Mbps\*1个月\*12个月\* 200 = 24w                                         合计：100w/年 | 资源成本：15000元/月\*10= 15w 第三方运维服务费用：12w                              合计：27w/年 |

* 上云实践11：IT正在向DT变革随着云计算的到来，传统IT已经向大数据（DT）时代变革。云计算低成本、高效率、灵活扩展等诸多优点，已经在逐渐淘汰传统IDC的IT模式。正如上面迁云的对比表格中，可以看到在成本里面。迁云前，是有四个运维人员的。迁云后，客户是没有运维人员的。在上云的第一年，客户仅保留了一名运维来处理日常琐事。在第二年中，客户公司已经将剩下的一名运维人员也裁掉。从某方面来讲，云时代对运维行业是冲击，很多运维人员面临失业。因为传统中小型互联网公司不再需要运维人员来做一些琐事，这些问题在云平台中都得以解决。从另外一方面来讲，这也将跟我们带来新的机遇及挑战，要求技术人员的技术更加全面。这也是为什么很多人说devops是未来之路的根本原因！