## 一）应用框架示意



### 1.1日常业务支撑：

活动:建立对应的模板支撑，建立事前预估、事中跟踪和事后总结的评估闭环

宣传:线上线下的广告引流标示

业务指标:（实时和延迟指标（中间表））层级维度规划，数据收集分析呈现

模型设置：用户（新老/额度/积分/接入方式）、渠道(线上（pc、微信、app）/线下)、定价（优惠/返利/原价）、时限(促销)、地域（甲乙双方） 金币/积分/返现/余额

### 1.2 性能：

并发连接数：nginx性能基数如下

参考：DELL 2850 两颗 Intel(R) Xeon(TM) CPU 2.80GHz，OS认作4个CPU，4GB内存 单页123K大小，1000~3000的并发数可以控制在3秒以内（1万并发用户数，每个网页3~10个连接）

IO带宽：浏览静态宣传内容（网页体积\*并发）

运算性能：主要应对用户登录后的查询和交易，并发数与网站浏览比对，应该是指数级下降，需按照用户规模、活动因素、时间因素进行估算

伸缩性：

**部署层面**：考虑nginx和tomcat集群进行hash路由设置，每个应用服务器50~100的并发考量，理论上可以随着应用服务器的增长线性增长的，业务再度扩大可考虑进行二级nginx代理分发。

如果不进行hash路由设置，则需对整个体系结构进行重新梳理，协议层面完全按照无状态进行交互，技术实现复杂度上升

不推荐进行集群session复制，伸缩性较差，管理难度高

**开发设计层面**：可以按照业务按照SOA理念进行切分，协议按照无状态连接进行切分，淘宝的无状态保持是基于cookie的解决方案，对于cookie数量有限制的解决方式是采用组合键值，控制在20个以内

参考：罗兰家纺[www.luolan.cn](http://www.luolan.cn) 50W U/V（日）DNS到两台Ngnix， 8台HP web server（每台上面起两到三个tomcat ），通过memcached进行单点session存储，每台机器均无状态访问， 通过URLrewrite伪静态化，CDN每个月五六千块（需要静态化的规则提交给CDN） 营销活动通过模板来自行CMS，对业务的报表每日生成，自己监控两张报表。

### 1.3 安全

网络安全：攻击和篡改

业务安全：伪造和泄密

数据安全：月（全量）/周（全量）/日（增量）备份

内部风险：部署和数据等敏感信息管理流程

## 二）网络结构

|  |  |  |  |
| --- | --- | --- | --- |
|  | **用途** | **部署** | **备注** |
| 1 | 网关 | Nginx代理分发 | 可选 |
| 2 | 应用前置机1 | Nginx+tomcat(http) |  |
| 3 | 应用前置机2 | Nginx+tomcat(http) | 可选 |
| 4 | 安全交易机 | tomcat(https) |  |
| 5 | 内部管理应用服务器 | Tomcat |  |
| 6 | 数据库服务 | Db |  |
| 7 | 文件服务器 |  | 可选 |
| 8 | 内部管理服务器 | Tomcat |  |

## 三）安全影响

使用HTTPS方式的技术变化

证书生成

<http://www.linuxde.net/2012/03/8316.html>

部署配置

<http://blog.csdn.net/techchan/article/details/5484253>

server to server

<http://stevenjohn.iteye.com/blog/1686216>

android to server

<http://my.oschina.net/blackylin/blog/144136>

IOS to server

<http://blog.csdn.net/woaifen3344/article/details/41145729>

调用不可信证书rest（开发环境使用）

<http://blog.csdn.net/chszs/article/details/8206607>

## 四）部署技术

Tomcat 8 支持完整版websocket 1.0(全双工双向通讯

Ngnix)

1）动静态分离

<http://my.oschina.net/u/878259/blog/172491>

2）tomcat集群

<http://www.tuicool.com/articles/UNZnqiU> ip\_hash实现同一用户持续访问为同一tomcat，不用考虑session复制问题

3）nginx禁止IP访问配置

<http://www.cnblogs.com/fangbo/archive/2011/02/21/1959855.html>

4）网站安全狗

<http://www.safedog.cn/install_desc_website.html>

5）性能监测

<http://ihuangweiwei.iteye.com/blog/1219302>

6）Ngnix性能调优

<http://blog.csdn.net/xifeijian/article/details/20956605>

7)架构皮毛

<http://blog.csdn.net/jhzyz/article/details/33737111>

8)memcached java客户端

<http://blog.csdn.net/seelye/article/details/8511073>

9) 用Redis存储Tomcat集群的Session

<http://blog.csdn.net/chszs/article/details/42610365>

10）用户安全鉴权

http://www.v2ex.com/t/123346

## 五）开发框架及组件选型

### 5.1 整体

|  |  |  |  |
| --- | --- | --- | --- |
|  | **网站宣传** | **在线业务** | **内部管理** |
| 呈现 | 静态（PC+手机） | Web（Jquery bootstrap requires ）  App：（H5（活动）+原生）混合式  微信： | 目前即可 |
| 业务 | 来自内部管理分配的固定标示 | 服务接入分散  用户服务和运营服务接口隔离  通道等标示扩展性加强  严格业务事务逻辑控制  Session集中  安全权限集中  调度任务集中  交易模块集中  公共服务（短信）集中  产品上下架服务集中  社区集中 | 工作台  工作流  权限管理端 |
| 持久 |  | 业务数据和监控（浏览器属性、地域访问IP、服务器负载等）数据隔离  第三方接口调用痕迹凭证保留完整 | 公司用户数据和客户数据分库 |

在线业务预计要负责定时生成静态页面（或者json文本数据）网站宣传

在线业务数据要推送一份前天部分业务数据至内部管理以备数据沉淀和统计分析

内部管理预计要发送指令到在线业务更新活动状态或者产品状态

集中的原则是soa和启停不影响全局

### 5.2在线业务考虑

## 六）展开计划

|  |  |  |
| --- | --- | --- |
| 一级 | 二级 |  |
| 原型验证 | Ngnix+tomcat集群 |  |
|  | 应用热部署 |  |
|  | Tomcat https部署 |  |
|  | https简单api的各种终端接入 |  |
|  | Session集中共享 |  |
|  | 事务控制原型 |  |
| 项目业务识别 | 模块分割及子系统分离 |  |
| 项目技术约定 |  |  |
| 实现技术定型 |  |  |
| 管理流程约定 |  |  |

## 讨论

技术线路大众化些，新进人比较快掌握

我个人想法是： 快（开发实现部署）前台（业务交互系统） 重（全准）后台（数据层）

Api version要不要考虑

停机维护场景？部署期间的异常？

极致化的热插拔 osgi？

代码目录结构

