爱旅行-系统架构设计及部署策略分析

爱旅行项目是一个综合性的旅游电商平台，并具备跨平台性。那么基于PC、移动端（Andriod、Ios）的考虑，以及该项目的非功能性需求（详见《爱旅行系统需求分析说明书.doc》）:安全性、可靠性、互操作性、健壮性、易用性、可维护性、可移植性、可重用性、可扩充性等等都是基于性能、安全等架构目标的，因此爱旅行项目系统架构的设计目标有以下7个：

1. 可靠性（reliable）
2. 安全性（secure）
3. 可扩展性（scalable/extension）
4. 可定制化（customizable）
5. 可维护性（maintainable）
6. 客户体验（customer experience）
7. 市场时机（time to market）

## 系统架构设计

爱旅行架构分为：前端架构和后端架构，将前端和后端完全分离，并采用分布式集群的架构设计，通过定义API，与前端进行数据交互，前端使用html进行数据展现，并可加入移动端（Andriod、Ios）实现，直接调用API即可。爱旅行系统的整体技术架构如图1所示：

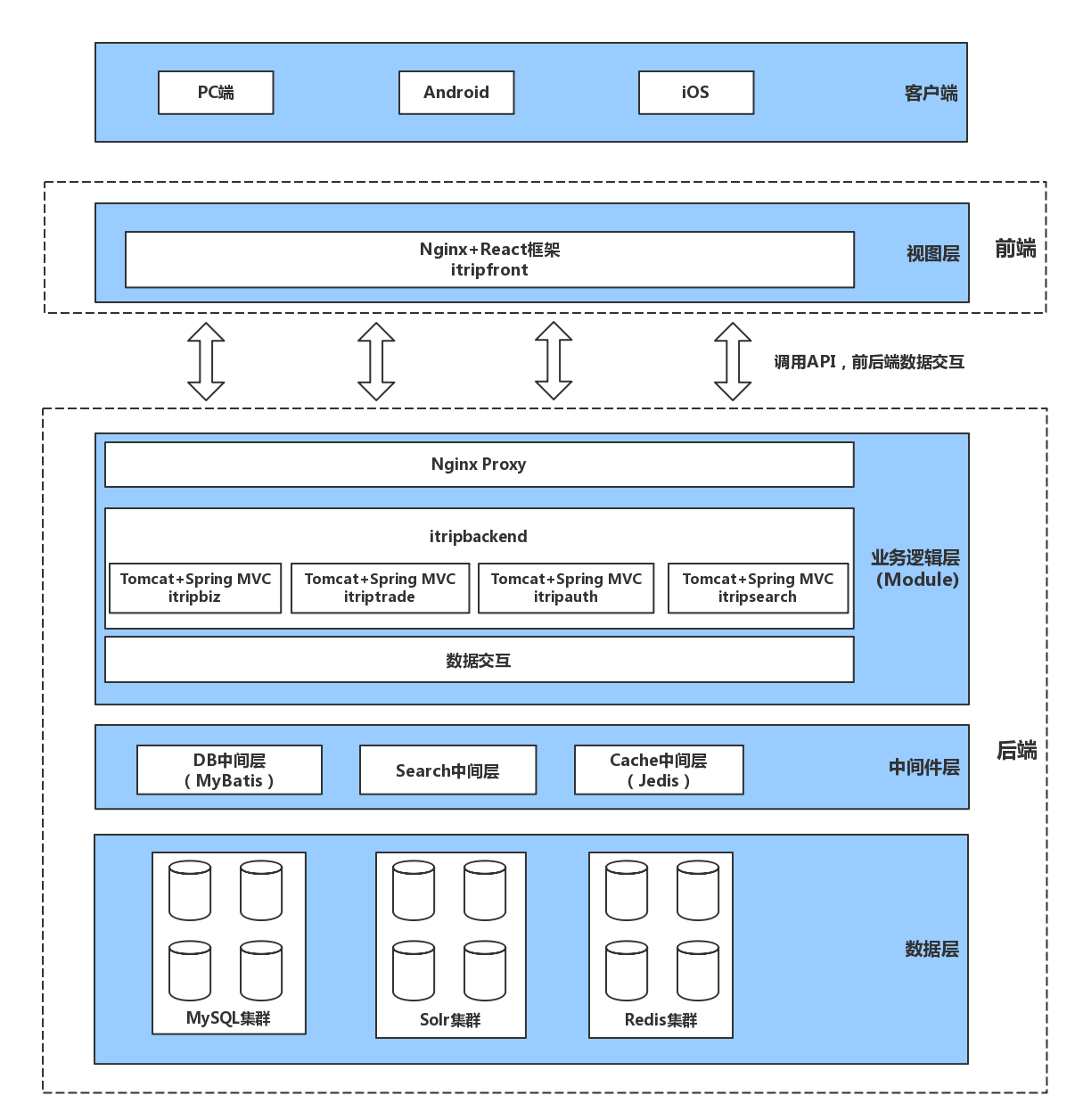


图1：爱旅行系统架构图

前端架构采用React JavaScript 库实现组件化开发，并通过Nginx反向代理到后端项目，实现API的调用，进行前后端数据交互。

后端架构采用模块化开发，采用Maven进行多Module管理。根据业务进行模块划分，itripbackend分为以下四个Web模块：

1. 主业务模块（itripbiz）：包括酒店业务、旅游业务、机票业务、攻略业务
2. 搜索模块（itripsearch）：爱旅行项目中所有搜索功能（solr）
3. 认证模块（itripauth）：负责用户身份验证，生成并维护Token
4. 支付模块（itrptrade）：包括支付宝支付、微信支付

以及为Web模块提供服务的三个基础模块：itripdao、itriputils、itripbeans。

每个web模块均独立部署在单独的web服务器上，并通过Nginx反向代理到相应的应用服务器（tomcat）上：

|  |
| --- |
| upstream itripbiz\_server  {  server IP:port; # itripbiz模块部署的服务器IP及端口号  server IP:port backup; # itripbiz模块部署的服务器IP及端口号（备份服务器）  }  upstream itripsearch\_server  {  server IP:port; # itripsearch模块部署的服务器IP及端口号  }  upstream itripauth\_server  {  server IP:port; # itripauth模块部署的服务器IP及端口号  }  upstream itriptrade\_server  {  server IP:port; # itriptrade模块部署的服务器IP及端口号  }  …  server {  listen 80;  server\_name itrip.project.bdqn.cn;  root /data/itrip/itripfront; #前端静态工程  index index.html;  location /biz {  …  proxy\_pass http://itripbiz\_server;  }  location /search{  …  proxy\_pass http://itripsearch\_server;  }  location /auth {  …  proxy\_pass http://itripauth\_server;  }  location /trade{  …  proxy\_pass http://itriptrade\_server;  }  …  } |

## 1.2 系统部署架构设计

根据不同用生产环境要求（并发量，访问量等等）以及成本控制等综合考虑，我们可以采用不同部署策略，可根据具体情况选择合适的部署架构。

1. 第一种部署架构模式

基于成本考虑，采用一台阿里云服务器运行前后端应用程序及数据库、缓存服务等，具体部署架构图如下图2所示：

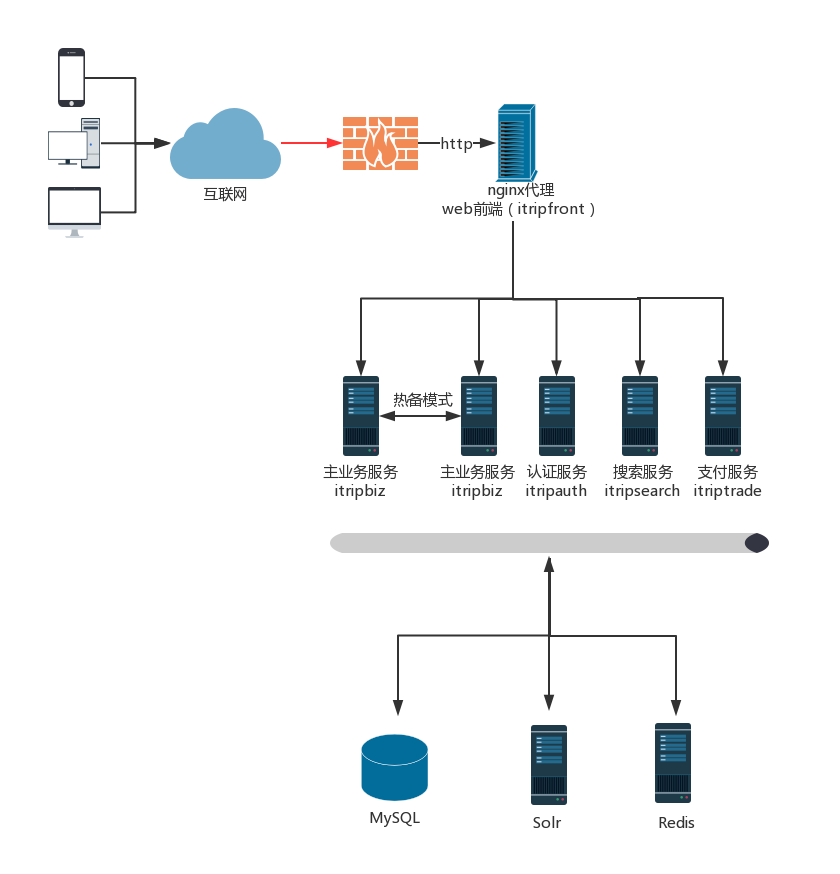


图2：爱旅行系统部署架构-策略1

1. 第二种部署架构模式

采用两台阿里云服务器分别运行前、后端应用程序。前端、后端完全分离部署，并使用Nginx做代理。即：一个前端服务器（nginx）和一个后端服务器（nginx、tomcats）具体部署架构图3如下所示：

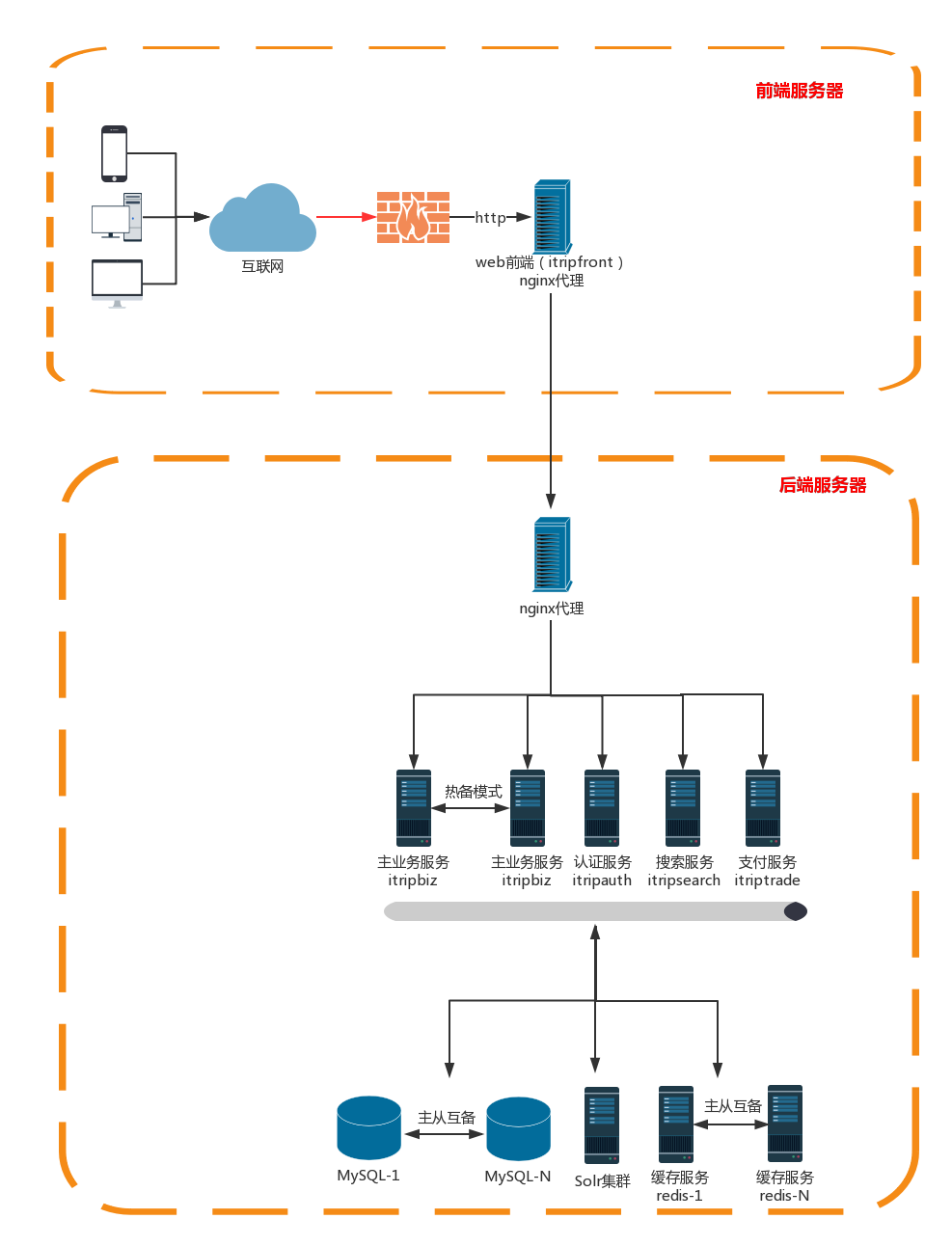


图3：爱旅行系统部署架构-策略2

1. 第三种部署架构模式

采用N台服务器，第三种是最为理想的是部署架构，所谓的理想状态，是指在面临高并发和高稳定性的目标状态下，也为以后更高的扩展能力。具体的部署架构如下图4所示：

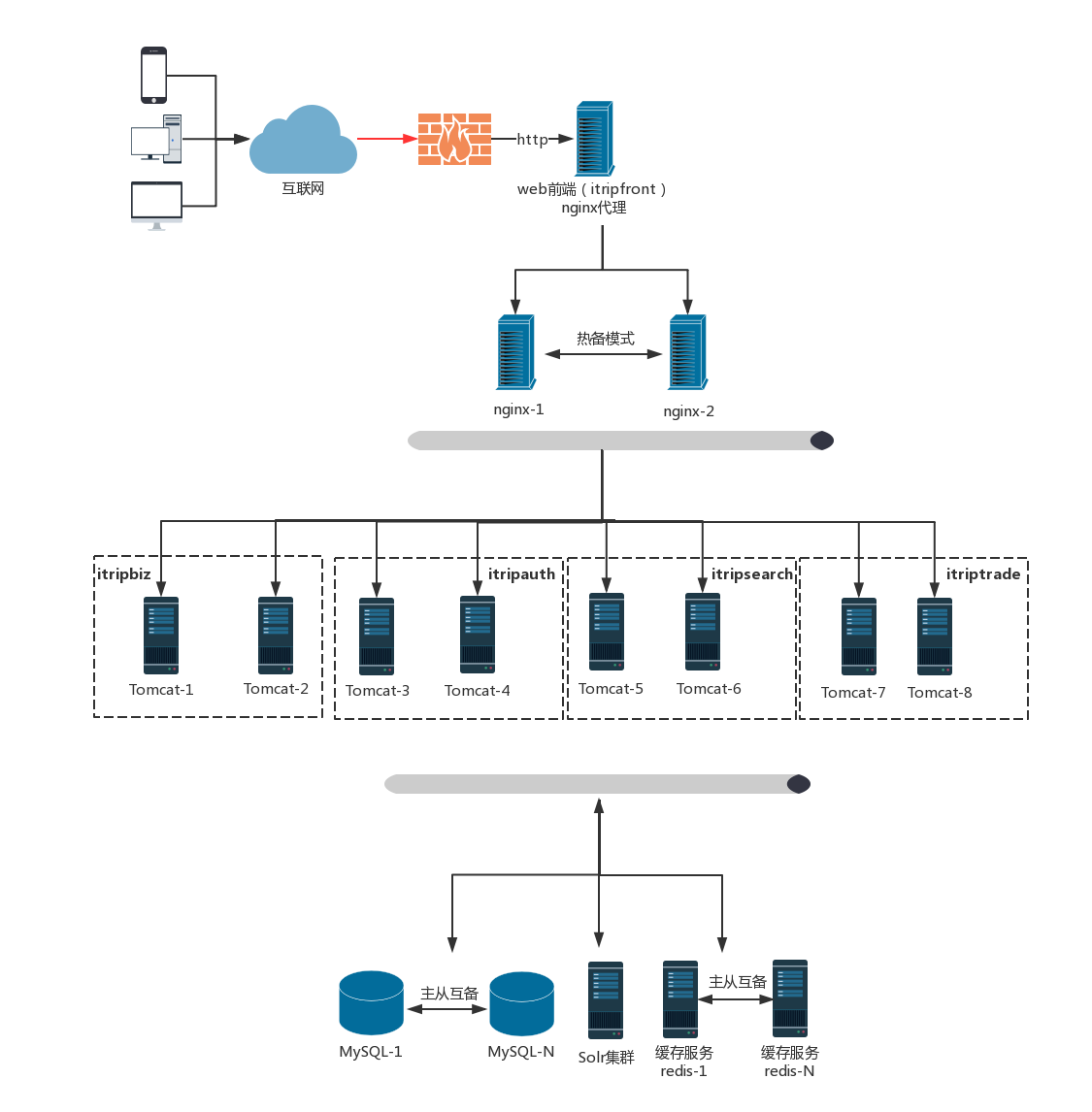


图4：爱旅行系统部署架构-策略3

在上述部署架构模式中，前端项目单独运行在一个Nginx服务器上，并通过代理到后端服务器，可根据并发量不同，可以配置不同数量的负载，Nginx之间的关系为互备，通过交替轮询的方式或者基于权重轮询的方式将请求转发给Tomcat，根据业务模块（itripbiz、itripauth、itriptrade、itripsearch）分为4组Tomcat服务器，每组Tomcat可配置为两个，采用双机热备的模式，Nginx每次只将请求发送至一个Tomcat上，直接进行缓存和数据的操作，若其中一个宕机的话，Nginx将自动转发请求至热备的Tomcat上，以达到系统的稳定性。即：所有的服务器软件都可以单独的启用并使用，比如：tomcat、redis、mysql等，单独使用一个也可以，这就体现了此架构的高扩展性。