

---

## 爱旅行项目-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.docx》）：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）

### 1.1 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：

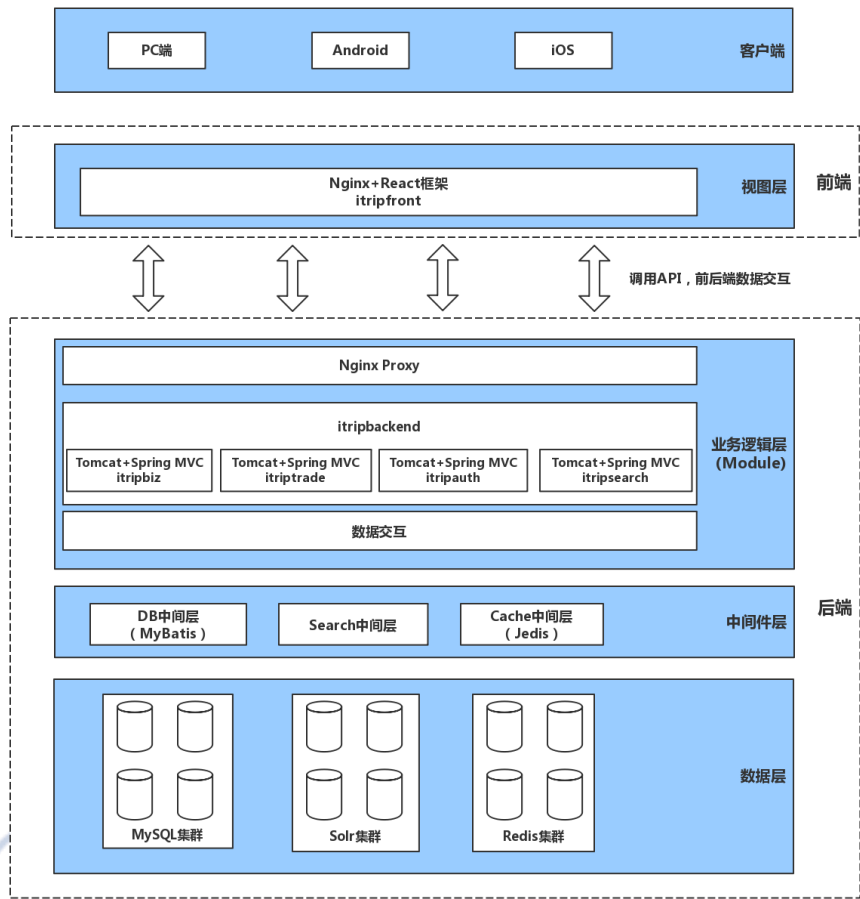


图 1：爱旅行系统架构图

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（itriptrade）：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.bdsn.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:

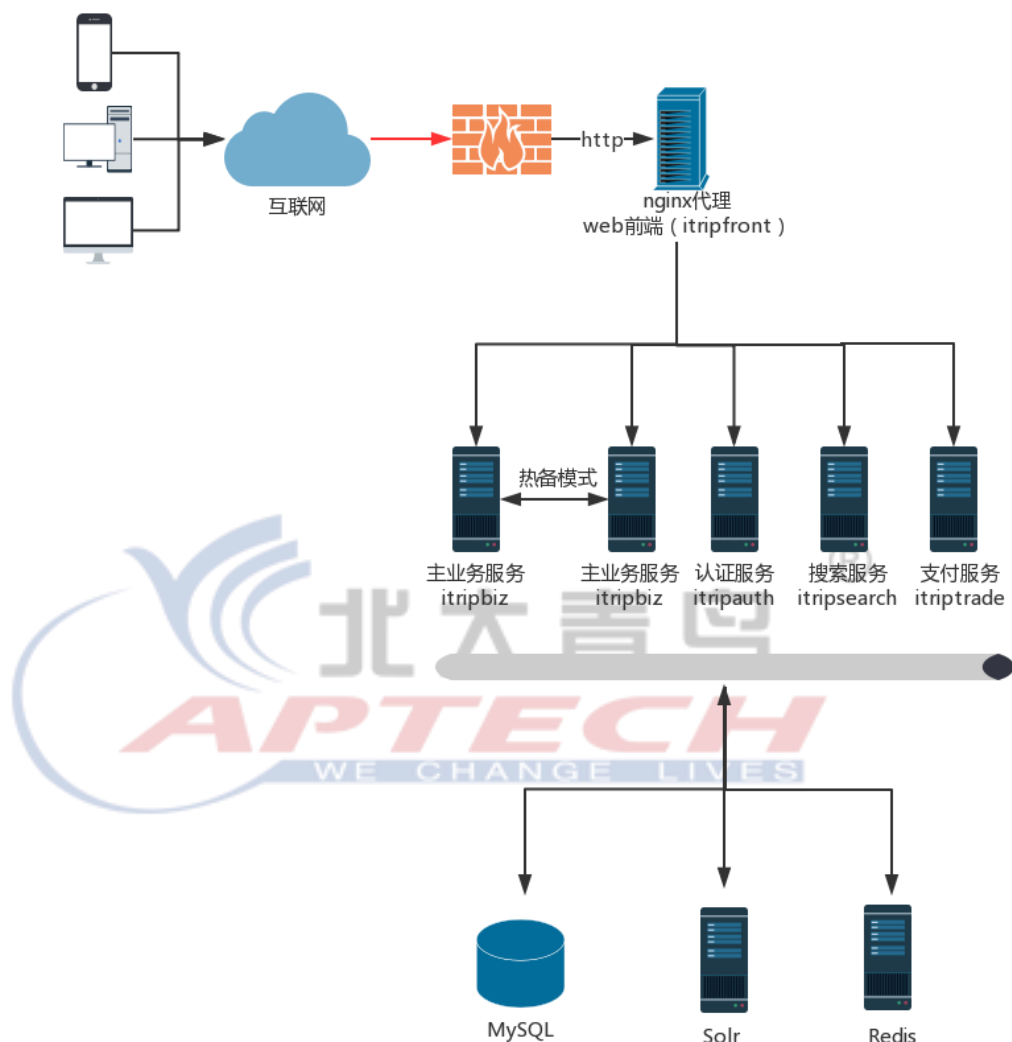


图 2: 爱旅行系统部署架构-策略 1

## 2. 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:

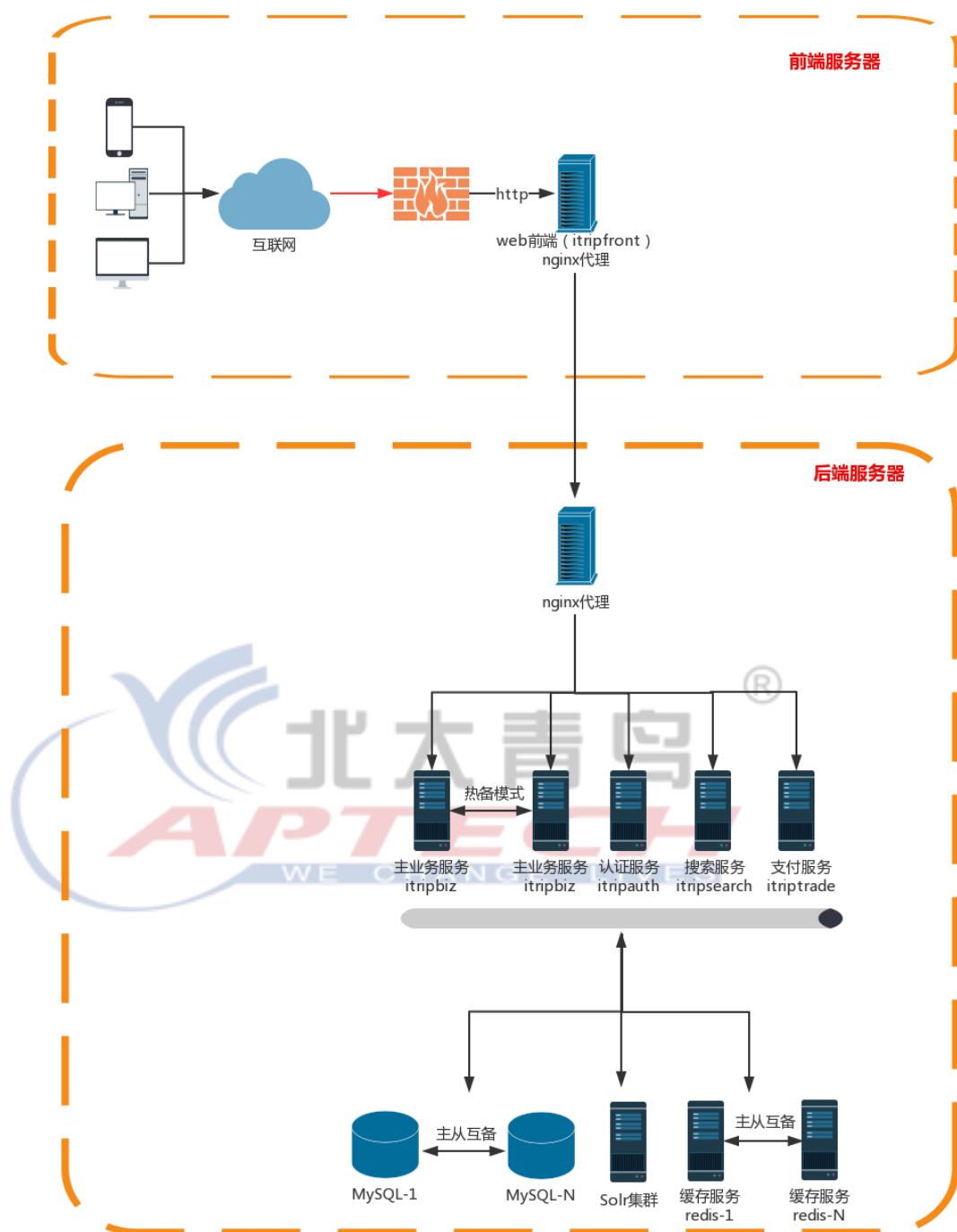


图 3: 爱旅行系统部署架构-策略 2

### 3. 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：

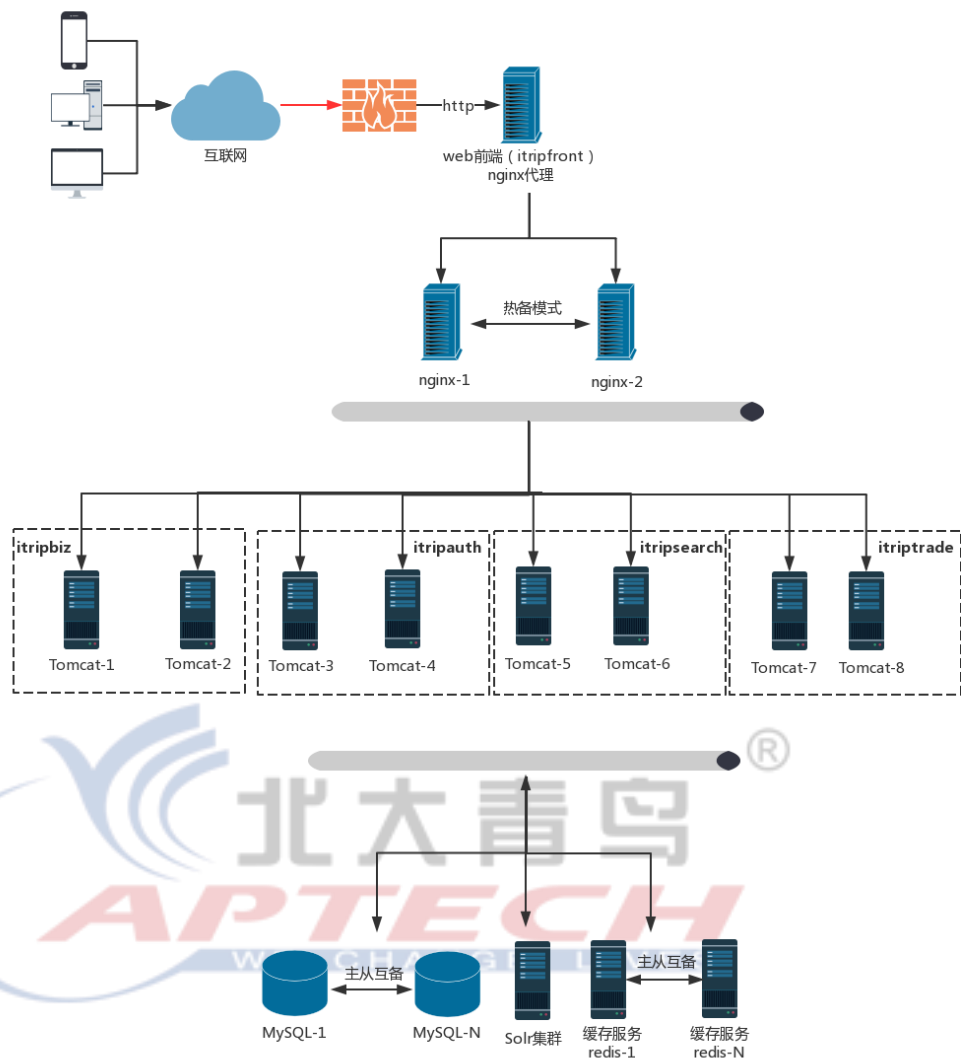


图 4：爱旅行系统部署架构-策略 3

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。