

# 一 . 性能策略

目标：提升访问速度

## 1 数据库策略：分割数据，减少数据查找量，加速操作速度

点餐平台主要由用户（点餐者和商家）的行为提供数据库操作内容，写操作大于读操作，将读操作与写操作分离，使用分布式数据库，可以大大提升数据库性能。垂直分表，水平分区，根据哈希算法，采用 `user_id%` 的方法分区后用户操作时数据库先去扫描小的分区内容，工作量大大减少。

## 2 服务器策略：缓存服务器，

使用本地缓存: 缓存将需要频繁访问的网络内容存放在离用户较近、访问速度更快的系统中，以提高内容访问速度。将 Web 对象缓存的拷贝保存在本地计算机中。Web 浏览器保留一个先前访问对象的缓存。如 IE 的“临时 Internet 文件”。我们的一些用户（商户）经常使用同一台设备访问我们的网站，所以比较适合。

## 3 文件系统策略：分布式文件系统，本地存储文件，加速查找文件

分布式文件系统（Distributed File System，DFS）是指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络与用户计算机相连，将网站资源与用户的计算机相连，提升系统性能，加快访问速度。我们可以使用 Hadoop 实现分布式文件系统

# 二 . 可用性策略

目标：易学易用

## 1 不浪费用户耐心

为了确保用户在第一时间查找到自己想使用的功能，我们使用了 Ajax 技术，将所有功能列于左侧功能栏，直接浏览和选择二级菜单即可

## 2 抓住用户注意

我们的点餐系统为了确保用户不会浪费浏览时间, 在注册和登录页面以较为简洁的背景与处于页面中央的操作界面减少信息干扰, 快速实现注册和登录。

## 3 规范和可视化

所有菜品使用列表形式整齐的列出, 并配有图片以供用户选择。