Atitit 提升响应速度 提升性能 简化 缩短调用链

目录

[1. 代码扁平化策略 1](#_Toc26334)

[1.1. 扁平化策略 减少层次 单层》双层》三层》多层 尽可能的简化 1](#_Toc32168)

[1.2. 减少跨文件代码跳转 1](#_Toc29260)

[1.3. 优先使用static静态代码块 1](#_Toc18314)

[1.4. 减少代码数量 加载类数量 2](#_Toc21440)

[1.5. 大力减少代码防止太大 2](#_Toc23293)

[2. 多功能扁平化接口设计 2](#_Toc4759)

[2.1. 多重sql往返场景，适当使用服务端sp触发器 存储过程等 2](#_Toc20741)

[3. 慎用重量级框架 臃肿不堪 2](#_Toc15994)

[3.1. 多使用更加轻量级技术 2](#_Toc12943)

[3.2. 使用简单技术 组件 框架 类库 2](#_Toc15582)

[3.3. 减少复杂组件 工具 类库的使用 3](#_Toc22137)

[4. 简化调用架构 简化部署 3](#_Toc23423)

[4.1. 应用和存储缩短调用架构 3](#_Toc9367)

[5. 使用成熟的技术代替自我实现 3](#_Toc6084)

[5.1. Mq代替im 长连接开发 通知等 3](#_Toc3599)

[5.2. 上云 提升性能 3](#_Toc18987)

[5.3. 适当使用大型中间件提升性能 大型数据库 cache等 3](#_Toc19853)

# 代码扁平化策略

## 扁平化策略 减少层次 单层》双层》三层》多层 尽可能的简化

## 减少跨文件代码跳转

代码跳转意味着加载另外一个文件资源。。减少跨文件跳转

也对可读性很大提升。。对ide也减少资源消耗，加快编译

## 优先使用static静态代码块

减少不必要的过度oo

## 减少代码数量 加载类数量

能用动态类处类就用动态类减少大量的样板代码

更少的代码量，更快的编译速度，更快的性能启动

## 大力减少代码防止太大 加快启动与编译

# 多功能扁平化接口设计

减少大量重复建设的也业务代码，，配置参数动态化。。

## 多重sql往返场景，适当使用服务端sp触发器 存储过程等

减少大量网络io

# 慎用重量级框架 臃肿不堪

## 多使用更加轻量级技术

## 使用简单技术 组件 框架 类库

往往更简单更稳定

## 减少复杂组件 工具 类库的使用

# 简化调用架构 简化部署

## 应用和存储缩短调用架构

做到上传，运行即可。。热部署更加方便

# 使用成熟的技术代替自我实现

## Mq代替im 长连接开发 通知等

## 上云 提升性能

## 适当使用大型中间件提升性能 大型数据库 cache等

大型mq等 大型缓存中间件