Atitit 性能问题分析与解决

目录

[1. 使用类orm等重型消耗cpu对框架 1](#_Toc25732)

[1.1. 不使用外挂框架 1](#_Toc13893)

[1.2. 双重嵌套循环 sql 1](#_Toc185)

[1.3. 没有开启query cache 等各种cache 2](#_Toc19622)

[1.4. 开启code cache 2](#_Toc26547)

[1.5. 语言缓慢，替换为java sql sp 2](#_Toc22520)

[2. ref 2](#_Toc18710)

[2.1. Atitit 系统性能优化流程 2](#_Toc23911)

# 使用类orm等重型消耗cpu对框架

## 不使用外挂框架

的 PHP 的运算性能表现都很优异，但是受到外加框架的影响，原本可以在数秒内处理上千个网页要求的 PHP，性能大幅下降，变为只能处理数十 个要求。

😂orm框架很影响性能啊，在这类复杂接口里面直接拼sql吧。。框架加载很耗时

## 双重嵌套循环 sql

那个复杂接口，可以 打个日志统计下 循环调用sql的数量 ，看是不是太多？

即使是双重嵌套循环，外部100个，里面100个，乘起来也是一万条类

尽可能前置运算到前端。。。

## 没有开启query cache 等各种cache

## 开启code cache

## 语言缓慢，替换为java sql sp

# ref

## Atitit 系统性能优化流程

目录

1. 分析性能 在代码还是db 1

1.1. 要求立竿见影速度快，，配置式样优先 1

2. 配置模式调优 根据 性能优化 top5 大方法 1

2.1. 负载均衡架构 数据库集群读写分离 分库 2

2.2. 前端cache cdn，后端db中间件cache（一般集成redis） ，db query cache 2

2.3. 多利用数据库对性能机制 2

2.4. 代码cache opcache一类对 2

2.5. Vm调优 gc ，， jit c1 c2等 2

2.6. 调整的内容包括更换服务器，从websphere改为tomcat， 2

2.7. 更改启动参数 2

2.8. 代码缓存 opcache一类的 2

2.9. 热部署配置调整，加快启动速度，不要开始就加载所有 2

3. 代码微调类 2

3.1. 针对每个模块可使用性能更高的组件 3

4. 代码模式改动调优，较为麻烦 3

4.1. 换用高性能语言重写 3

5. 性能优化几大方法 3

5.1. 配置化解决和vm优化最简单 3

5.2. 架构配置优化 vm优化 代码语言优化 3