Atitit 稳定性提升的艺术 之技术解决之道

目录

[1. 2. 为什么会发生稳定性问题 1 1](#_Toc12676)

[2. 大原则 1](#_Toc20067)

[2.1. 尽快释放资源类似php最好的稳定性 1](#_Toc1054)

[2.2. Nginx 负载均衡 排除单点故障 1](#_Toc980)

[2.3. 分离 隔离故障 1](#_Toc4144)

[2.4. Crash-only 模式 自动重启服务 守护 2](#_Toc15888)

[3. 语言与架构级别提升 2](#_Toc6979)

[3.1. 彻底的前后端分离 2](#_Toc28705)

[3.2. 多进程 前端使用php ngihx做加载 2](#_Toc2391)

[3.3. . Atitit 提升稳定性 错误处理 全局错误捕获 2](#_Toc7861)

[4. 技术管理上的举措 2](#_Toc581)

[4.1. 压力测试 2](#_Toc22662)

[4.2. 预生产环节 2](#_Toc20255)

[4.3. 4. 稳定性测试 3 2](#_Toc19120)

# 2. 为什么会发生稳定性问题 1

2.1. 单点故障(单点故障率较高) 1

2.2. 复杂 1

2.3. 资源耗尽 1

2.4. 死锁与等待 1

2.5. 崩溃 1

# 大原则

## 尽快释放资源类似php最好的稳定性

## Nginx 负载均衡 排除单点故障

## 分离 隔离故障

## Crash-only 模式 自动重启服务 守护

Crash-only software refers to computer programs that handle failures by simply restarting, without attempting any sophisticated recover

# 语言与架构级别提升

## 彻底的前后端分离

## 多进程 前端使用php ngihx做加载

## . Atitit 提升稳定性 错误处理 全局错误捕获

# 技术管理上的举措

## 压力测试

## 预生产环节

## 4. 稳定性测试 3

4.1. Throw ex测试 3

4.2. 断开连接测试 3

4.3. 多线程测试并发 3

单元测试

atitti 提升稳定性的艺术之程序代码级别稳定性的艺术 attialx著 艾提拉著