提升稳定性 总结

Catalog

[1. 参考金融 os系统 2](#_Toc11243)

[ 内存mng 进程隔离 内核态 核心态 用户层 2](#_Toc9889)

[2. 减少类库减少进程占用 2](#_Toc4552)

[3. 多京城模式 2](#_Toc20845)

[4. Webapi简化，通过cmd调用业务url 2](#_Toc5734)

[ 网络组件替换为本地模式 2](#_Toc26162)

[5. Sqlite lucene 2](#_Toc23382)

[ 架构简化 简化调用环境 3](#_Toc19526)

[6. 尽可能减少调用的类库 3](#_Toc20114)

[7. 网络组件替换为本地模式 3](#_Toc13767)

[8. Webapi使用sdk web api即可 3](#_Toc11659)

[9. 存储也可以使用file api即可 3](#_Toc5899)

[10. Sqlite+lucene也可 3](#_Toc7511)

[ 存储管理 3](#_Toc25209)

[11. 虚拟内存技术 3](#_Toc27052)

[12. Share分片 与复制 3](#_Toc29756)

[13. 减少内存占用 简化 3](#_Toc18502)

[ 安全api3. 错误处理 4](#_Toc13418)

[ 负载均衡 4](#_Toc11497)

[ Other 4](#_Toc27546)

[14. 优先级调度 4](#_Toc23081)

[15. 资源管理 死锁检测与避免 4](#_Toc25859)

[4. ：通过死锁检测算法及时发现和处理死锁情况，确保系统资源有效利用。 4](#_Toc32144)

[16. 热替换技术 4](#_Toc4748)

[17. 哨兵模式 4](#_Toc23007)

## 参考金融 os系统

# 内存mng 进程隔离 内核态 核心态 用户层

## 减少类库减少进程占用

## 多京城模式

根据进程的重要性和资源需求动态调整其运行优先级，优化系统性能和响应能力。

**进程隔离**：通过虚拟内存技术确保进程间的隔离，防止一个进程的崩溃影响到其他进程。

## Webapi简化，通过cmd调用业务url

# 网络组件替换为本地模式

## Sqlite lucene

# 架构简化 简化调用环境

## 尽可能减少调用的类库

## 网络组件替换为本地模式

## Webapi使用sdk web api即可

## 存储也可以使用file api即可

## Sqlite+lucene也可

# 存储管理

## 虚拟内存技术

：允许系统使用硬盘空间作为扩展内存，减少内存不足导致的崩溃。

## Share分片 与复制

## 减少内存占用 简化

# 安全api3. ****错误处理****

* **异常处理机制**：提供有效的错误检测和处理机制，捕获异常并进行相应的恢复操作。
* **日志记录**：系统记录错误和事件的日志，便于后续分析和故障排查。

# ****负载均衡****

：通过分布式系统和负载均衡技术提高系统的可靠性和可用性。

# Other

## 优先级调度

## ****资源管理 死锁检测与避免****

### ：通过死锁检测算法及时发现和处理死锁情况，确保系统资源有效利用。

* **配额管理**：为用户和进程设置资源使用限额，防止资源耗尽导致系统崩溃。

**负载均衡**：通过分布式系统和负载均衡技术提高系统的可靠性和可用性。

## ****热替换技术****

：支持在系统运行时替换故障组件，提高系统可用性。

## 哨兵模式