

高性能服务器程序设计

肖彬



设计原则



服务器设计



测试/维护/监控



案例分析

设计原则

- ◆ 可用性
- ◆ 稳定压倒一切
- ◆ 安全
- ◆ 性能，今天的主题
- ◆ 易于扩展
- ◆ 易于升级和维护

服务器设计

- ◆ 性能瓶颈
- ◆ OS选择
- ◆ I/O策略
- ◆ 服务器模型
- ◆ 连接池与缓存
- ◆ 数据库方案
- ◆ 小技巧
- ◆ 分布式

服务器设计

性能瓶颈



数据copy



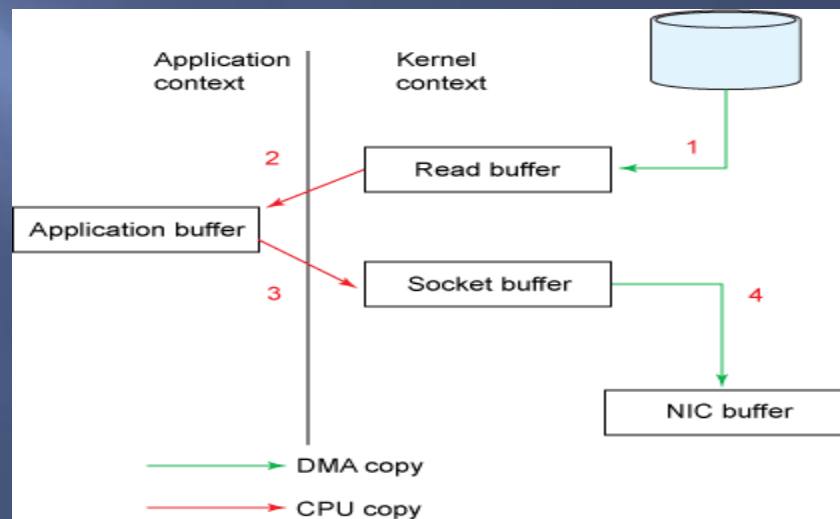
上下文切换



内存分配



锁竞争



性能瓶颈

上下文切换



产生场景

- ◆ 用户层调用跟内核层调用切换
- ◆ 线程之间切换



如何优化

- ◆ 使用 `tmpfs`、`sendfile`、`writenv`、`sendv` 等函数
- ◆ 合理分配线程

性能瓶颈

内存分配



产生场景

- ◆ 按需动态分配、动态释放



如何优化

- ◆ 预先分配
- ◆ 延时释放
- ◆ 避免竞争

服务器设计

OS选择



Socket API效率 <http://bulk.fefe.de/scalability/>

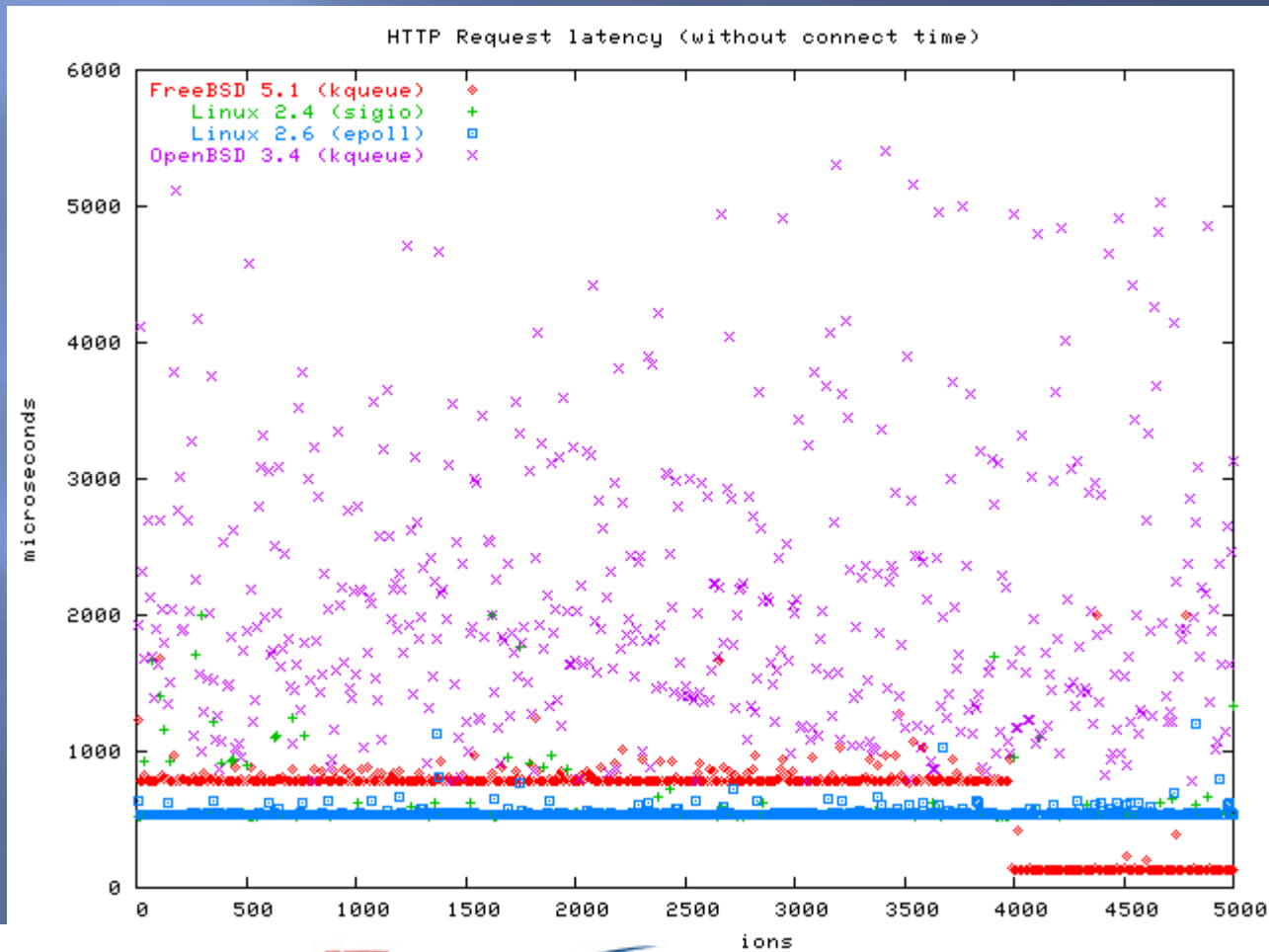


线程创建和销毁



I/O支持

服务器设计



服务器设计

I/O策略



Select



Poll



Kqueue



/dev/poll



Epoll

服务器设计

服务器模型

◆ 协议选择

TCP

UDP

◆ 工作模式

阻塞

非阻塞

◆ 多进程

◆ 多线程

◆ 多进程/多线程

◆ prefork

服务器设计

连接池与缓存



数据库连接池

- ◆ 限制最大连接数
- ◆ 避免短连接



缓存

- ◆ 使用嵌入式数据库替代，如Berkeley DB
- ◆ 使用纯内存数据库memcache
- ◆ 自主实现

服务器设计

数据库方案



oracle



mysql

性能瓶颈

分布式



负载均衡

- ◆DNA轮询
- ◆NAT
- ◆LVS
- ◆F5硬件设备



同步

- ◆缓存同步
- ◆代码同步

服务器设计

小技巧

- TCP选项
 - 立即发送
 - ✓ TCP_NODELAY
 - ✓ TCP_CORK
 - 缓存区大小设置
 - ✓ SO_SNDBUF
 - ✓ SO_RCVBUF
- Readv
- Writev

测试/维护/升级



测试

- ◆ 功能测试
- ◆ 压力测试工具，如ab、LoadRunner等
- ◆ 内存泄漏检测工具
 - ◆ 静态检测工具，如splint、PC-LINT等
 - ◆ 动态检测工具，如valgrind, Rational purify等
- ◆ 性能分析工具，如gprof

测试/维护/升级



维护

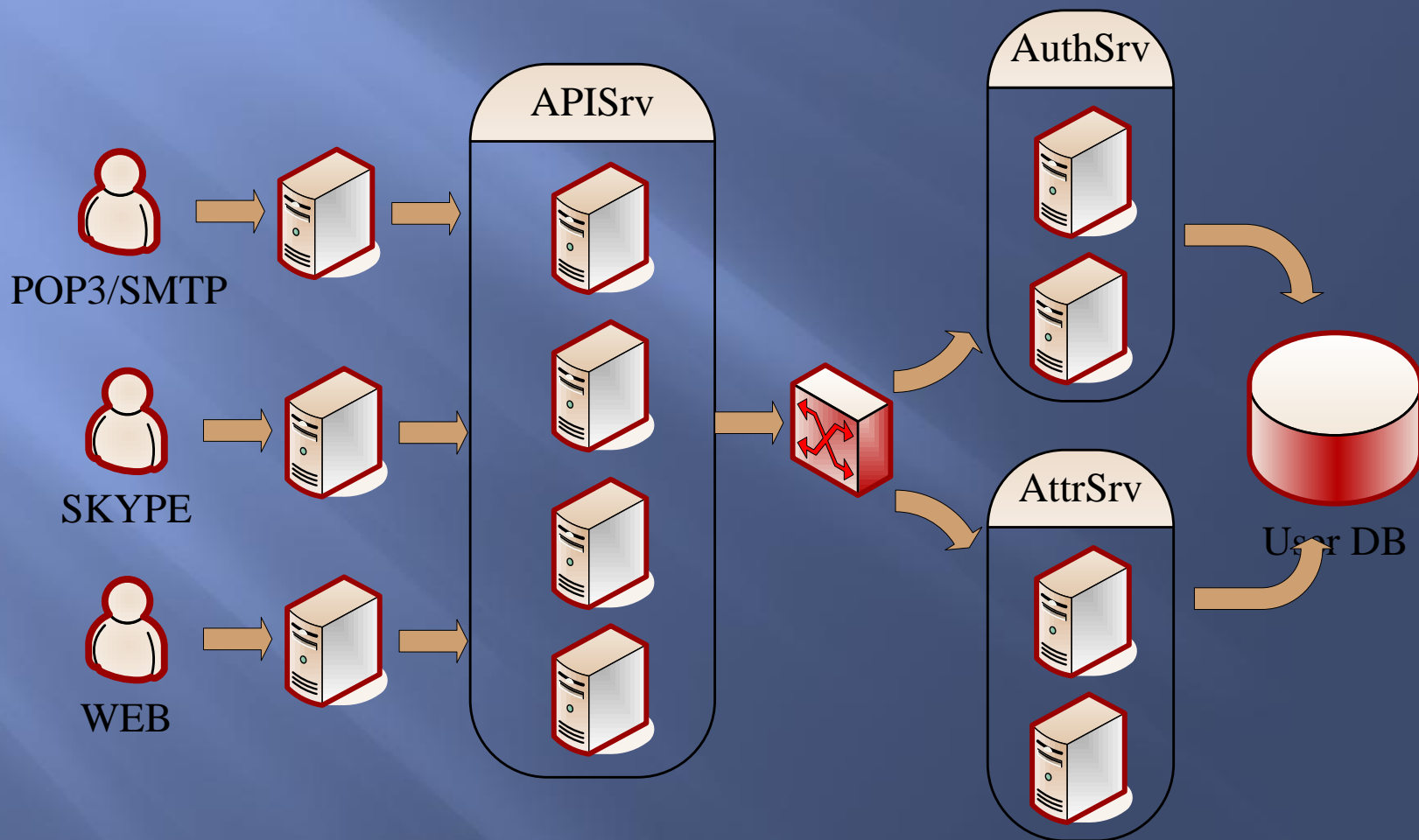
- ◆ 详细的文档
- ◆ 完善的日志记录以及日志分析系统
- ◆ 监控平台
- ◆ 一致的测试平台



升级

- ◆ 升级前的详尽测试
- ◆ 自动化升级脚本

案例分析



高性能服务器设计

谢 谢