

高并发之路

性能优化

锁竞争

- 并发级别
- 锁粗化
- ThreadLocal
- ReadWriteLock
- 锁分离
- 锁消除
- 自旋锁
- 乐观锁
- 轻量锁
- 偏向锁
- 锁范围
- 锁粒度
- 去热点
- 锁分段

非阻塞

- ConcurrentHashMap
- 非阻塞栈
- 非阻塞链表
- 非阻塞计数
- Atomic\*
- RCU
- CopyOnWriteArrayList

内存同步

上下文切换

Thread模型

- 线程池
  - 传统池
  - fork/join pool
- 并行定律
  - Amdahl/Gustafson's law
  - CPU/IO密集型

Actor模型

STM模型

Future/CompletableFuture

设计模式

- disruptor
- Producer/consumer
  - Read/Write
- Barrier
- Latch
- Guarded suspension
- Scheduler
- Active Object
- Thread-Specific Storage
- Leader/Follower
- Reactor
- Preactor
- Map reduce

Fiber/Coroutine

并发模型

底层实现

同步

- 内置锁
- 显式锁
- LockSupport

通信

- 内置条件
- 显式条件
- 高级工具
  - CountDownLatch
  - CyclicBarrier
  - Semaphore

线程

- interrupt
- suspend/resume
- join/yield

安全问题

争取不变性

- final
- 事实不可变对象

隔离可变性

- 并发容器
  - \*Queue

保证可见性

- 锁
- volatile
- 其它happens-before

保证原子性

- 锁
- CAS
- 64位操作

安全发布

- 逸出
- 栈封闭

JMM

- 原子性
- 可见性
- 有序性
- Happens-before

活跃问题

死锁

- 重入锁
- 超时
- 试获取

活锁

饥饿

应用架构

手段

- 拆分/分层
- 异步化
- 缓存

原则

- 无状态
- 高层次上实现并发

典型场景

- C1000K问题
- 计数问题
- 秒杀

Load Balancing

- 线程池的lb
- 常见lb算法

BackPressure