

Writing Concurrent Code with Lock-Free Algorithms

Course Summary

Description

This is a truly inspiring and challenging course for Java and C++ programmers interested in building low-latency, high-throughput systems.

Taking a unique hardware-up approach, you will be taught how to build code that reaches world-class performance levels.

Objectives

By the end of this course candidates will have acquired a new in-depth knowledge of concurrent programming that allows them to develop data structures such as queues, executors, and shared memory IPC transports that significantly out perform any products currently available.

In addition to learning the basic techniques of building high-performance data structures, a performance testing and profiling discipline will be introduced that is fundamental to working at the extremes of systems performance.

Topics

- Understanding Mechanical Sympathy
- Performance considerations on modern hardware
- How modern x86-64 hardware informs concurrent and high performance programming techniques
- Memory models for software and hardware
- Why contention of any type is the enemy and how to avoid it
- How we manage contention when we absolutely must
- How the JVM provides support for concurrent programs
- Profiling and performance testing
- Concurrent and High-performance Algorithm Design
- Lock-free concurrency primitives
- Signaling state change
- Ensuring order
- Concurrent data structures – Queues, Dynamic Lists, Executors, Shared memory IPC, Locks
- Preventing speculative execution during critical data exchanges
- Managing contended state
- Wait-free techniques
- API design to avoid the latency J-curve
- Efficient back-off strategies
- Discovering hidden contention and how to avoid it

Prerequisites

There are no prerequisites for this course.

Duration

Three days