STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

STAC Application Performance Boot Camp

Minimizing latency in trading applications is a science, not an art. So is maximizing the efficiency of parallel and floating-point intensive applications. But the techniques are rarely taught in a systematic way. The STAC Application Performance Boot Camp has changed that.

Join us as we explain modern computing architectures and how to get maximum application performance from them in a trading environment. The Day 1 overview of modern hardware systems is suitable for developers, quants, system engineers, and managers. Day 2 is an intensive, code-driven course for application developers and quants who want to improve the performance engineering of their apps. Both are taught by finance-computing veteran and thought leader Niall Dalton.

Day 1 → Modern Systems Boot Camp

Day 2 → High-Speed Code Boot Camp

Day 1 only: USD 1600 | Both days: USD 3100
Includes breakfast, lunch, snacks, and all course materials.
Group discounts are available.

Details and registration:

New York
October 17 & 18, 2012
Downtown Conference Center
157 William Street

London
November 8 & 9, 2012
Bonhill House
1-3 Bonhill Street

Copyright © 2012 STAC. "STAC" and all STAC names are trademarks or registered trademarks of the Securities Technology Analysis Center LLC. All other names are the property of their respective owners.

Day 1: Modern Systems Boot Camp

Course motivation:

In order to write competitive low-latency or high-capacity applications, today's developer must understand how to exploit the target hardware and systems software environment. Likewise, to deploy and support high-performance applications, infrastructure engineers must understand what these apps demand from the platforms. This course will provide the foundation for such understandings. In addition to an in-depth explanation of modern computing building blocks, attendees will gain a conceptual framework for understanding the impact of technologies yet to emerge.

Who this is for:

Developers, systems engineers, managers, and anyone else with a desire to bolster their understanding of modern hardware systems and the systems software stack.

Prerequisites:

- None.

Course Description:

Attendees will develop an understanding of modern computer systems from the ground up, including microprocessor, network, and storage technologies. Starting from the general design principles and constraints facing hardware architects, we use the "numbers that everyone should know" to evaluate back-of-the-envelope designs at various scales, from a single rack unit to entire datacenters.

Topics include:

- Microprocessors, caches, the memory hierarchy, and NUMA. Dennard scaling and latency-optimized processors with examples from the Intel 8086 to Haswell (pipelined, superscalar, speculative out-of-order, transactional microarchitectures), with particular attention to the Intel "Sandy Bridge" and "Ivy Bridge" platforms.

- The end of Instruction Level Parallelism and the rise of multi- and many-core vector machines including threaded and SIMD architectures. Following in-depth coverage of Intel Advanced Vector Extensions (AVX) and the forthcoming AVX2 SIMD extensions from a hardware and software perspective, we cover architectural and programming implications of recent GPU designs and the Intel Xeon Phi processor. We briefly survey alternative architectures such as FPGAs and tiled network processors.

- Networks and interconnects, from chips to continents. After surveying the principles and models that apply to networks of all scales and technologies, we survey the most common, paying particular attention to PCIe, QPI, Ethernet, and InfiniBand. We explore the NICs and switches in modern high-speed networks optimized for latency-sensitive and "big data" applications, as well as specialized technologies such as FPGAs. We survey the leading techniques for clock distribution / time sync.

- Storage technologies for volatile and non-volatile data optimized for sequential and random access. We characterize both spinning and solid state media (SSD versus PCIe; DRAM and related memories; PCM and other near future non-volatile memories) and explore storage-system architectures (SAN, NAS, DAS).

- Software topics, including OS and application considerations, will be touched upon in preparation for in-depth treatment in Day 2.

Copyright © 2012 STAC. "STAC" and all STAC names are trademarks or registered trademarks of the Securities Technology Analysis Center LLC. All other names are the property of their respective owners.

Day 2: High-Speed Code Boot Camp

Course motivation:

Many financial applications that need to achieve low, deterministic latency or handle a large number of computations are unable to exploit the innovative multi- and many-core designs coming from the chip vendors. And too often, programmers and quants who try to exploit them get burned.

Course description:

The course will cover the principles and practice of building high-speed systems for handling trade execution or compute-intensive analytics. This includes applications that require low-latency I/O, those that require fast computation, and the growing number that require both. Going into more depth on key hardware discussed in the STAC Modern Systems Bootcamp, we will lay out the principles of low-latency application design, vector and multi-threaded, multi-process, and distributed applications. We will have a detailed discussion on the capabilities available to SIMD programmers on both CPUs and GPUs. Special emphasis is placed on reusable high-performance patterns, with examples from common libraries such as Intel TBB.

Who this is for:

- Developers who wish to learn or solidify hardware-aware programming techniques for low-latency.
- Quantitative researchers and developers who wish to learn the latest threading and SIMD techniques, particularly use of AVX on recent Intel systems, in addition to forward-scalable code for GPUs and future machines like Intel Xeon Phi.

Prerequisites:

- Intermediate level C/C++ skills. Lectures will make extensive use of code examples.
- Must have attended STAC Modern Systems Bootcamp.

Topics include:

- Demystifying the modern Ethernet NIC. Kernel and driver internals, and userspace interfaces including userspace protocol stacks, tradeoffs between bandwidth and latency (e.g., interrupt coalescing). Non-standard APIs for low-latency. Utilizing ring buffer designs in software to mimic NIC hardware and software designs.

- Operating system and application software implications as we follow the “lifecycle of a bit” from the network to user application and back to the network. Good design principles on Sandy Bridge, considering DDIO and cache algorithms. Understanding the OS and tuning it to optimize for low-latency and low-jitter operation. Interrupt processing, core binding, shielding, and other best practices

- An introduction to SIMD programming on a modern Intel processor. In particular the theory and practice of using AVX will be presented in depth, in addition to algorithmic principles and deterministic patterns for high-performance programming on multi-threaded, multi/many-core SIMD processors. We will explore successively lower levels of tool support for SIMD programming and profiling. With a grasp of AVX assembly language, attendees will learn how to understand what, if any, use is being made of AVX by standard compilers. We will then show how language extensions, automatic vectorization, intrinsics and assembly language increase the scope of code that can be profitably vectorized. We will review the roadmap for SIMD extensions on future x86 processors, as well as GPUs and Intel Xeon Phi. We will also cover some of the key points of OpenCL and CUDA.

- Threading do’s and don’ts including design for latency versus design for application throughput. Programming for cache locality and memory efficiency with native and managed code. Correctness and performance anti-patterns, with a survey of major APIs including TBB, OpenCL, CUDA, etc.

- Local, network, distributed, and clustered file-systems. Buffered, Direct IO, and memory mapping. Random versus sequential IO patterns and systems effects they cause. Tradeoffs of synchronous and asynchronous programming.

Copyright © 2012 STAC. “STAC” and all STAC names are trademarks or registered trademarks of the Securities Technology Analysis Center LLC. All other names are the property of their respective owners.

Instructor Biography

Lead instructor: Niall Dalton

Niall is a STAC Fellow with expertise in algorithms and technology for low-latency and compute-intensive systems in applications such as high-frequency trading. Currently doing software architecture at Calxeda, Niall's 17 years of experience also include working as Director of High-Frequency Trading at a Wall Street firm; as CTO of Kx Systems, a leading vendor of high-performance column-oriented database software widely used on Wall Street; as senior software engineer at NVIDIA; and as CTO at X.R.N.D, a European vendor of high performance parallel data analysis software. He has enjoyed a variety of engineering and research positions in Europe and the US in areas such as language design and compilers for parallel computing, data-intensive distributed systems, and non-traditional database internals. He currently serves as an advisor to MemSQL, creators of a realtime in-memory MySQL compatible database, 0xData, a stealth-mode big-data analytics startup, and Parallel Scientific, creators of novel parallel programming tools. Despite many publications and multiple degrees in Computer Science, Niall acquired the skills to swear fluently at multifarious hardware and software systems in a wide variety of common and obscure programming languages. He has never met an abstraction layer he didn't enjoy violating.

About STAC

STAC (www.STACresearch.com) is a technology-research firm that facilitates the STAC Benchmark Council™, an organization of leading trading organizations and vendors that specifies standard ways to test the performance of trading solutions. STAC Benchmarks cover workloads throughout the trading process, inculding market data, risk analytics, and execution.

STAC helps end-user firms relate the performance of new technologies to that of their existing systems by supplying them with STAC Benchmark reports as well as standards-based STAC Test Harnesses™ for rapid execution of STAC Benchmarks or customized tests in their own labs. End users do not disclose their results. Some STAC Benchmark results from vendor-driven projects are made available to the public, while those in the STAC Vault™ are reserved for qualified members of the Council.

Copyright © 2012 STAC. "STAC" and all STAC names are trademarks or registered trademarks of the Securities Technology Analysis Center LLC. All other names are the property of their respective owners.