

Christopher D. Metcalf

metcalf@lcs.mit.edu

Work Experience:

2003–present, VERITAS Software. Development Architect. Worked in the “VERITAS Application Saver” product group, building a production-focused application reliability management tool that provides low-overhead execution tracing information at the level of source lines for multithreaded applications on Windows, Solaris, Linux, Java, and .NET; provides a highly-sophisticated memory allocator for C/C++ applications; and reports a wide range of reliability metrics to a central enterprise console for actionable alerting and over-time analysis.

- Drove the overall architecture of the product, writing architectural documents and working closely with developers to oversee implementation and ensure the architecture stayed clean.
- Generated specifications for new product releases in consultation with product management.
- Improved the development process on many levels: defect tracking, peer review, source documentation, and cross-site visibility between development groups.
- Directly managed one engineer locally and served as overall technical lead for the other engineers in our group and remotely.
- Worked with senior developers in other VERITAS groups to share technology and enable our respective products to interoperate.
- Directly contributed substantial individual development to the code base, including most recently a port of our execution-tracing instrumentation to the Linux platform.
- Reworked our core technology to make it suitable for a forthcoming VERITAS .NET product, proving technical superiority over a pre-existing instrumentation effort underway in that product.
- Actively participated in sales calls, proof-of-concept visits and product betas.
- Taught training sessions to sales engineers and services representatives.

2002–2003, Geodesic Systems (acquired by VERITAS). Chief Architect. Played a similar role to that described above at VERITAS. After Geodesic’s acquisition of InCert (below), worked to integrate the InCert and Geodesic development groups and pull together the two technologies and products into a single enterprise-focused product that provided all the functionality of the two separate products. Implemented several substantial pieces of core functionality, including in-process runtime metrics infrastructure and metrics transfer to host agent. Part of the team involved in the acquisition of Geodesic by VERITAS.

1997–2002, InCert Software Corp (acquired by Geodesic Systems). Co-founder. Key architect and developer of: binary-translation technology to insert high-performance native probes into S/390 mainframe, Windows/NT, and Solaris applications; runtime library that interacts with and interposes on the host operating system to capture thread status, exception/signal status, and the like; and reconstruction layer that converts runtime-compressed control trace information into viewable source-level traces. Technical team leader and initial developer for both Windows/NT and Solaris products.

1993–1996, Consultant (greater Boston area). Worked at Virtual Machine Works (now part of Mentor’s emulation division); University of Massachusetts Medical Center; and BB&N, Administered Unix systems and wrote IT and development tools.

1989–1997, MIT. Key system administrator for my research group, with over fifty Unix workstations serving three groups; implemented comprehensive cross-platform tool environment.

1988–1989, Institut de Recherche et Coordination Acoustique/Musique, Paris, France. Systems administrator for Unix network. Wrote and ported both user- and kernel-level code.

Summer 1987, Thinking Machines Corporation, Cambridge, MA. Wrote a disk test suite for the Connection Machine parallel “self-healing” drive (the DataVault).

Summer 1983, Compute! Publications. Editorial programmer working with and porting code among a wide range of home computers.

Technical Skills:

- Operating Systems: long experience with Unix (primarily Solaris and Linux; other versions as well) and Windows. Familiarity with systems at the tools level, at the deeper user/kernel interface level, and in the kernel.
- Languages: Fluent in C++/STL and C; experienced in Java, C#/.NET, Perl, shell scripting, and a wide range of other languages. Very knowledgeable at the compiler/system interface level.
- Hardware: Broad experience with x86, SPARC, MIPS, S/390, and other instruction-set architectures; solid understanding of system architecture issues from the VLSI level up through the chip to multiprocessor, systems, and network levels.
- Development tools: GNU toolchain (make, configure, gcc), Windows visual studio, source control tools (CVS, Perforce), code-documentation tools (Javadoc, Doxygen)
- Process tools: Familiar with Microsoft Project, TeamTrack, TestTrack, Gnats, and the Microsoft office productivity tools
- Other expertise: TCP/IP, HTTP/HTML, XML.

Education:

Ph.D. in Computer Science, Massachusetts Institute of Technology, September 1997 [2]. Thesis under Professor Steve Ward in the Computer Architecture Group at the Laboratory for Computer Science, “Managing Scheduled Routing With A High-Level Communications Language.” The work examined the issues involved in scheduled (precomputed) routing on a multicomputer. The ability to precompute paths allows improved routing in the common case, and makes it possible to design extremely simple, high-speed routing elements. The thesis presents techniques for efficiently using such a substrate with a wide range of applications, and shows substantial speedups with suitable applications.

My research group worked to design, implement and explore the general area of scheduled routing [2-6]. We created an extremely high-speed routing element that benefited from run-time simplicity and compile-time routing analysis. In addition to my thesis work, and contributions to the chip architecture, I implemented and maintained an event-driven, modular simulator, ‘nsim’, using a Tcl/Tk graphical command and debugging interface. I also wrote a number of applications, e.g. a fault-tolerant diffusion-based parallel Traveling Salesman Problem solver.

Combined M.S. and B.S. in Computer Science, Yale University, May 1988. Awarded *summa cum laude* and distinction in the major. Wrote Master’s thesis (awarded Honors) [7] on the design and implementation of an asynchronous data-parallel virtual machine.

Teaching Experience:

Spring 1997, teaching assistant for a graduate course on parallel systems architecture and applications, with Professor Anant Agarwal.

Fall 1990, teaching assistant for an introductory graduate-level course in VLSI systems design, with Professor Jacob White and Professor Anant Agarwal.

Fall 1989, teaching assistant for an undergraduate overview course in computer systems engineering including a large number of papers from the literature, with Professor Barbara Liskov.

Patents:

“Method for determining the degree to which changed code has been exercised” (No. 6,748,584).

Publications:

- [1] “TraceBack: First Fault Diagnosis by Reconstruction of Distributed Control Flow.” Andrew Ayers, Richard Schooler, Anant Agarwal, Chris Metcalf, Junghwan Rhee, and Emmett Witchel. In the *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) 2005*.
- [2] “Managing Scheduled Routing With A High-Level Communications Language.” Chris Metcalf. Ph.D. Thesis, Massachusetts Institute of Technology. *Also available as MIT TR 725*. August 1997.
- [3] “A Unified System for Scheduled Communication.” David Shoemaker, Frank Honoré, Pat LoPresti, Chris Metcalf, and Steve Ward. *International Conference on Parallel and Distributed Processing Techniques and Applications*, 1997.
- [4] “NuMesh: An Architecture Optimized for Scheduled Communication.” David Shoemaker, Frank Honoré, Chris Metcalf, and Steve Ward. *Journal of Supercomputing*, 10(3):285-302 (1996)
- [5] “NuMesh: A Communication Architecture for Static Routing.” David Shoemaker, Chris Metcalf, and Steve Ward. *International Conference on Parallel and Distributed Processing Techniques and Applications*, November 1995. (Named one of best five papers at the conference.)
- [6] “The NuMesh: A Modular, Scalable Communications Substrate.” S. Ward *et al.* *Proceedings of the International Conference on Supercomputing*, July 1993.
- [7] “An Exploration of Asynchronous Data-Parallelism.” Michael Littman and Chris Metcalf (Master’s thesis). *Yale Technical Report 684*, October 1988, revised June 1990.
- [8] *Learning C: Programming Graphics on the Amiga and Atari ST*. Marc Sugiyama and Chris Metcalf. *Compute! Publications*, 1986.
- [9] *Compute!’s Beginner’s Guide to Machine Language on the IBM PC and PCjr*, Chris Metcalf and Marc Sugiyama. *Compute! Publications*, 1984.

Other works with *Compute! Publications* include an introductory chapter on C programming for the *Amiga Programmer’s Guide* and a number of articles in other *Compute!* books.