Kevin P Schoedel

540 Mustang Place, Waterloo, Ontario N2K 4B9
kps@datatravelandexperiments.com
+1 226 600 8898

Experience

Senior Software Engineer (L5), Google, 2014 – 2023

Notable projects I worked on, chronologically:

- A Chromium/ChromeOS platform abstraction ('Ozone'), focusing on text input. Improved internationalization, accessibility, and web-standards conformance, simplified by my chosen representation at the abstraction layer.
- An early smart-speaker project, which would have had Google provided a reference embedded-Linux-based controller and software for OEMs. My role was to develop a flexible but straightforward interface for OEM user controls.
- A prototype full-featured standalone VR headset, primarily on controller input and web content embedding; then, for Daydream phone-based VR, on controller input and Android content embedding.
- Home displays ('Nest Hub'), on the original embedded Linux/Chrome-based stack. I worked on touchscreen input, accessiblity, memory profiling, and IoT hardening.
- Matter, the emerging home IoT standard. In the open-source SDK, I worked mostly on hardening and streamlining inherited code, and memory use reduction, profiling, and tooling. Subsequently, I worked on bringing Matter support to Nest devices, focused on device discovery (DNS-SD) and IPv6 issues.

Software Engineer, Intel, 2010 – 2013

Intel's Waterloo, Ontario site was established at the end of 2009 from its acquisition of RapidMind, a startup selling a data-parallel JIT programming system.

- Worked on bringing the Intel Labs Ct project, with some additions from RapidMind, up to product quality as Intel Array Building Blocks (ArBB).
- Adapted Intel's internal OpenCL vectorizer for an LLVM-based successor to ArBB designed around high-level data parallel intrinsics in LLVM IR.
- Added Cilk Plus elemental functions to Clang, using an IR interface modelled on OpenCL SPIR, in support of a Clang-based front end for ICC.
- Addressed problems related to use of LLVM in Android, including fixing Renderscript-related bugs, and other issues building the x86 Android tree with LLVM.
- Planned and prototyped support for Intel MPX (Memory Protection Extensions) in LLVM. (MPX was a set of new instructions aimed at allowing pointer bounds checking in C/C++ without breaking existing ABIs and data layouts.)

Senior Software Engineer, Sandvine, 2005 – 2010

Sandvine developed carrier-grade network equipment providing flexible congestion management, network integrity, and operational support.

- Optimized the network processor's assembly-language software to 40% of its original length to meet the essential hard real-time target imposed by 10GigE wire speed.
- Enhanced the software to support IP-based load balancing, MPLS and L2TP tunneling protocols, and new network deployment variations, while maintaining the critical-path time constraint. Ported the assembly code to two new hardware platforms.
- Worked on FreeBSD kernel drivers for the network processor and associated components.
- Maintained and ultimately redesigned the controlling C++ host application.
- Wrote a debugging shell that was ultimately adopted as the system command-line interface.

Senior Software Designer, Archelon, 1989 – 2004

Archelon, previously Bit Slice Software, began developing compilers for microcode for high-performance graphics systems, and broadened to support other processors outside the general-purpose mainstream, including massively-parallel SIMD, VLIW, DSPs, and special-function controllers; clients have ranged from start-ups to well-known computer and semiconductor companies. Selected projects:

- Designed, wrote, maintained and enhanced the register allocator, tailored to handle idiosyncratic processors. This project led to Archelon's compiler outperforming GCC on customer benchmarks.
- Targeted the compiler, in whole or in part, to around 20 processors. These tasks typically included designing the compiler ABI, and hand-coding low-level library routines, as well as retargeting the code generator.
- Added various compiler features, including SIMD conditional locking, symbolic debugging support, and output rewriting to support arbitrary third-party assemblers.
- Designed and wrote a standalone instruction scheduler for the Intel i960 family of processors. (The i960 CA was the first commercial superscalar microprocessor.) Also worked on scheduling for other processors, and a generic microcode compaction tool.
- Wrote a FORTRAN 77 compiler front end, and FORTRAN standard intrinsic libraries. Interfaced the front end to three local code generators and to a third-party back end.
- Added to the in-house linker support for object libraries, arbitrary relocation expressions, user-defined debug table formats, and FORTRAN common blocks. To the assembler, added general code relaxing with user-level directives for conditional assembly by address range, and other additions including symbolic debugging support for hand-written assembly code. Wrote an m4 clone.
- Built a cross-platform floating-point compiler support library, derived from the contemporary Cephes suite.

Intern, Control Data Corporation, Fall 1985 and Summer 1986

At CDC, I primarily performed acceptance testing for VX/VE, a UNIX environment for the Cyber 180, and the associated C compiler. I wrote and ran tests and prepared bug reports for the vendor; I also filtered or handled customers' issues.

Education

Bachelor of Mathematics, University of Waterloo, 1989
in Joint Honours Pure Mathematics and Computer Science.
Graduated on the Dean's Honours List.