Nathan Holland

Personal Data

Address: 460 S Marion Pkwy Apt #1756B, Denver, CO 80209
Phone: (201)-317-1487
Email: nholland94@gmail.com
Github: https://github.com/nholland94

Work Experience

Current
Dec 2013

Software Developer
Granicus

Developed internal projects to support legacy software. Spearheaded development on new products. Taught other developers various programming languages, technologies, and techniques. Investigated new technologies for the company and provided analysis. Worked on automated deployment system and tools associated with it. Built build processes for frontend applications.

Projects

Prometheus

Distributed, automated test running tool
Closed source

Runs cucumber test suites across machines with different configurations. Gives QA access to livestreams of tests as well as live logs. Analyzes logs from test runs and uses that information to highlight tests that are more likely to fail, prioritizing those first.

Technologies: Go, AWS, Azure, Cucumber

Fudd

MySQL binlog deserializer and RabbitMQ publisher
MySQL library source: https://github.com/granicus/mysql-binlog-go
Other components are closed source

Built to solve specific problem with legacy software in which we could not easily replicate model transactions to RabbitMQ. No MySQL binlog deserializer would work for our version of MySQL, so one had to be built from scratch. Tails the binlog live on the database server and takes advantage of concurrency to keep up with the application's high throughput.

Technologies: Go, MySQL, RabbitMQ

Zeus

Easily configurable, highly concurrent, extensible service bus DSL
Source: https://github.com/nholland94/zeus

Designed to make wiring restful service oriented architectures easy. Still maintains a high level of configurability. Written and configured in Elixir, but requires little knowledge of the language to use.

Technologies: Elixir, Beam VM, OTP

PAL

Array programming language which compiles to run on GPUs
Source: https://github.com/nholland94/pal

A high level programming language based off of APL. Compiles down to SPIR-V shader modules and layer dependency graphs. Dynamically assigns tasks to queues across devices during runtime.

Technologies: OCaml, C, Prolog, CLPFD

Programming Languages

Highly Proficient:	C, OCaml, Ruby, Go, Javascript, PHP, Java, Lua, Elixir
Moderately Proficient:	Lisp (specifically Scheme), x86 assembly, Python, Prolog, SML
Mildly Proficient:	Factor, Ada, Nim, Rust, APL, Haskell, Red, ATS

Technologies

Web Development:	Rails, Phoenix, React.js, Backbone.js, Underscore.js, WebSockets, Socket.io, Apache, HTML, CSS
Other Technologies:	Linux, SQL, Chef, Makefile, Grunt, Xlib, Wayland, OpenGL

Awards and Activities

2011 and 2012	"Excellence in Computer Science" - Gill St. Bernard's School
2010	Receive a 5 in AP Computer Science
2010	Worked with a Computer Science graduate student at Drexel University *Worked on optimizing N-Body simulator by performing calculations in parallel on GPUs*
2011 - 2012	Designed and participated in a programming independent study during highschool *Included artificial neural networks and DirectX 3D work*
2008 - 2012	Programmed on First Robotics Competition (FRC) team 2458

Interests and Activities

Programming Languages, Compilers, Kernels, Music Visualization, Virtual Reality, Puzzles, Vulkan