4762 Oregon St
San Diego, CA 92116

Alden King

636.253.3650
apking@cs.ucsd.edu

Education

Ph.D. Candidate — UC San Diego — Sept. 2007 - Present
Graduate student in High Performance Computing group. Advisor: Scott Baden.

BS in Comp. Sci., BA in Philosophy — University of Washington — Aug. 2002 - June 2007

Research

Tettnang — with Scott Baden — May 2011 - Present
A translator for the Saaz query language. Reduces and eliminates overheads in libraries.

Saaz — with Scott Baden and Sutanu Sarkar — Sept. 2008 - Present
Building a query language which is more appropriate for scientific computations than conventional relational languages (e.q. SQL) and databases (e.g. Oracle, SQL Server, Postgres).

Fraser Network — with Sandy Fraser — June 2008 - Sept. 2008
Designed a new network infrastructure and protocols for a new more secure internet. Features include backwards compatibility, a similar programming model, and vast security improvements. Built a simulator for physical through application layer protocols. Simulator was able to tunnel videos from youtube.com.

SEMINAL — with Dan Grossman and Benjamin Lerner — Sept. 2005 - June 2006
Experimentation with reducing the extra state and overhead in compiler type checkers required to report errors. Instead, if type-checking fails, a second pass is run which uses search to find an alternate version of the program which will type-check. Predict improvements in type-checker operating speed and maintainability.

Publications

The Saaz Framework for Turbulent Flow Queries — 7th IEEE International Conference on e-Science
Alden King, Eric Arobone, Sutanu Sarkar, Scott Baden — November 2011

Reducing Library Overheads through Source-to-Source Translation — International Conference on Computational Science
Alden King, Scott Baden — June 2012

The Saaz Framework for Turbulent Flow Queries — Future Generation Computing Systems
Alden King, Eric Arobone, Sutanu Sarkar, Scott Baden — IN SUBMISSION

White-Box Optimization of a Domain Specific Library — OOPSLA '12
Alden King, Scott Baden — IN SUBMISSION

Work Experience

Lawrence Livermore National Laboratory — Intern — June 2012 - Dec. 2010
Developer for the Rose source-to-source compiler for C++ and Fortran.
- Corrected support for l-value and r-value distinctions in C++.
- Compiled the before and after pre-processing token streams, allowing for source-code rewriting instead of regenerating.

Fraser Research — Intern — June 2008 - Sept. 2008
Researched new internet design without using TCP or IP.
- Built a simulator of the network backbone pipes and new routers.
- Designed and implemented protocols for the new network simulator and emulator.

Google, Inc — Intern — June 2006 - Feb. 2007
Researched means of getting tools for university classrooms to teach students how to think about parallel computing, large datasets, and "big" problems.
- Setup of computing cluster and pilot class at University of Washington. — June 2006 - Dec. 2006
- TA position at the University of Washington for the pilot class. — Jan. 2007 - Feb. 2007
- TA position at UCSD for a similar undergrad distributed systems course. — Sept. 2007 - Dec. 2007

Oltis Software, LLC — Programmer — Jan. 2004 - Sept. 2005
Managed development and release of Finance Logix Visual Advisor, software for financial advisors at banks and other investing companies.
- Managed new feature set development and implementation for second edition.
- Assisted in draft of specifications for RFP proposal for project bid.

Projects

- Cell Snort. Speeding up Snort, the intrusion detection system, on the cell BE processor. Pattern matching on large numbers of packets very quickly on the cell BE processor.
- Rapid update of scenes to large-scale tiled displays. This involves some distributed consistency, workload, and management.
- Comics Library. Built a C# application to read and write metadata to digital comics files. Created Vista/Windows 7 extension to view thumbnails.

Skills

Languages: C, C++, C#, Java, OCaml, XML, XQuery.
Tools: bash, make, lex & yacc, LaTeX, Linux.
Active Experience: networking, simulation, distributed computing, parallel computing, databases, library optimization, compilers, and programming languages.