Lars Bergstrom

1255 S Michigan Ave #PH7
Chicago, IL 60605
(425) 891-6751
lars@lars.com

Skills Summary

Currently working on web browser engine design and implementation. My Ph.D. research in Computer Science focused on high-performance compilation techniques and runtime technology for functional languages that target parallel hardware. Before that, I spent seven years at Microsoft building high-quality software for developers on the Windows platform, both as a developer and manager. I have also built several iPhone applications.

Experience

Researcher **Mozilla Research**
Chicago, IL 6/13–present

I currently work on the Servo project, and have been both a manager and individual contributor.
I managed 4 staff, 2 contractors, and 5 graduate researchers, working on styling, layout, graphics, HTML parsing, parallelism, power usage, session types, testing, and string encoding.

I am on the Servo leadership team, where my focus is on our relationship with Rust, concurrency/parallelism in Servo, integration of Rust into Firefox/Gecko, support for mobile / embedded platforms (primarily ARM Linux and Android), and coordinating partner and research efforts.

Collaborated with multiple large electronics companies on partnership opportunities. Trained engineers at one and worked with two on shared browser engine goals and planning.
Contracted or negotiated for work with many smaller firms (GitHub, Travis CI, and several consulting firms).

Research Assistant on Manticore **University of Chicago**
Chicago, IL 9/08–6/13

Compiler internals for Manticore, a compiler for a parallel dialect of Standard ML. Implemented several compiler optimizations and did significant work on static analyses.
Ported and tuned the compiler and runtime to gain speedups on up to Intel 64 thread machines and AMD 48 core machines.
Mentored five different undergraduates through a variety of compilers-related projects, two of which resulted in publications.
Co-authored an NSF grant, which resulted in most of the funding for my graduate career.
Authored an NSF REU subgrant, which provided additional funding for undergraduates to work on Manticore.

Independent iPhone Development
Chicago, IL 8/08–8/09

Implemented WiFinder, a wireless access point scanner. Sold more than $100,000 in net revenues, and was downloaded more than one million times.
iDraft and SplitTheBill, applications for tracking the NFL draft and splitting up a restaurant bill

between multiple parties.
Won the 2009 University of Chicago New Venture Challenge as part of the Bump team.

Research Assistant on Standard ML of New Jersey **University of Chicago**
Chicago, IL 9/07–09/09
Implemented pieces of the Windows-specific portion of the Standard ML basis library in C with ML frontend hooks for Standard ML of New Jersey (SML/NJ) .
Implemented profiling APIs, an MSI-based setup project, and stabilized the Windows implementation of SML/NJ.

Teaching Assistant **University of Chicago**
Chicago, IL 9/07–9/11
Hands-on laboratory sessions for Introduction to Computer Systems (2009, 2010, and 2011). Developed and administered exercises to teach the use of development tools to undergraduates.

Senior Software Design Engineer **Microsoft, Developer Tools Platform**
Redmond, WA 3/06–9/07
Ported all software from the next-generation tools platform project back into Visual Studio.
Prototyping work in C# with WSE on MSBuild multi-process and multi-machine support.
Built up and managed automated checkin and build systems in JScript, Batch, MSBuild, and C#.
Authored setup projects in WIX.
Created and maintained Team System project management templates.

Software Design Engineering Lead **Microsoft, Visual Studio Nautilus**
Redmond, WA 10/02–3/06
Founded team with three other people, leading a team of 10 developers and architects and two test developers building the base of the next generation of development tools at Microsoft. We did work in compilers, data modeling, large application framework, and user interface tooling.
Managed both keeping our dependencies stable and up to date (WPF/Avalon, .NET Framework) and ensuring we met our deliverables to other teams. Involved handling pre-alpha quality software while keeping the team productive and meeting a schedule.
Restructured the division's development process to be an agile software product line, incorporated developer testing in our processes, and fully automated the checkin and build monitoring.
Directed the re-organization of the Visual Studio Environment team into the Nautilus team, handling people evalutation and restructuring of nearly 200 people across the development and test organizaitons.

Software Design Engineer **Microsoft, Visual Studio Environment**
Redmond, WA 6/00–10/02
Created and shipped the Visual Studio Academic toolset and Academic SKU of Visual Studio, written in C# using Windows Forms, COM interop to Visual Studio's Automation model, and HTML/JavaScript for start page integration.
Time-bomb, licensing, class view, and object browser features of Visual Studio .NET 7.0, written in C++, mostly using COM.
Settings auto-upgrade feature of Visual Studio .NET 2003. C++ Win32 code.
Developed, rolled out, and supported a new build and release infrastructure to unify all the Developer Division tools division, in use now by over 1200 developers for the .NET Framework, compilers, and all of the Visual Studio environment teams. Written in perl, CMD batch, nmake makefiles, custom build tools, and the entire .NET Framework tool stack. Solved technology issues for how to get reproducible builds, even though we were building many of the tools we needed to use to do the build and how to scale up to the large number of developers worldwide on this code base.

Undergraduate Research Assistant **Northwestern University**

Evanston, IL 1996–2000

Ported Scheme 48 to Windows and the Sony Aibo robots. Also extended Scheme 48 with an editing environment for use in the introduction Computer Science class.

Undergraduate teaching assistant for the introduction to Computer Science class.

Taught a not-for-credit course on Windows Systems programming.

Independent project: RAM-based file system driver for Windows.

Won the 1999 ACM Quest for Windows CE application contest for a programmers' text editor for small devices.

Senior thesis entitled *A Substrate for ActiveX Support in Research Languages.*

Presented thesis work as a poster at USENIX Windows Systems 2000.

Managed the Computer Science department's lab and servers for two years.

Awards

2012 Computer Science Department TA of the Year

5 Microsoft Gold Stars

Systems Proficiency

Programming Languages

C#, C++, CUDA, Scheme, Objective C, ML

Technologies

.NET Framework, Windows Forms, Avalon, COM, OLE, ActiveX, Win32, Cocoa

Languages

English (native)

German (beginner — 6 years of primary and secondary coursework)

Education

Ph.D. Computer Science, June 2013
University of Chicago, Chicago, IL
Title: Parallel Functional Programming with Mutable State

M.S., Computer Science, 2009
University of Chicago, Chicago, IL
Paper: Arity Raising and Control-Flow Analysis in Manticore

M.S., Computer Science, 2005
University of Washington, Seattle, WA
3.8/4.0 GPA

B.S., Computer Science, 2000
Northwestern University, Evanston, IL
4.0 Major GPA, 3.49 overall

Publications

Engineering the Servo Web Browser Engine using Rust. B. Anderson, L. Bergstrom, M. Goregaokar, J. Matthews, K. McAllister, J.Moffitt, S. Sapin. Proceedings of the International Conference on Software Engineering (ICSE 2016), May 2016.

Practical and Effective Higher-Order Optimizations. L. Bergstrom, M. Fluet, M. Le, J. Reppy, and N. Sandler. International Conference on Functional Programming (ICFP 2014), September 2014.

Data-Only Flattening for Nested Data Parallelism. L. Bergstrom, M. Fluet, M. Rainey, J. Reppy,

S. Rosen and A. Shaw. Principles and Practice of Parallel Programming (PPoPP 2013), February 2013.

Nested Data-Parallelism on the GPU. L. Bergstrom and J.Reppy. Proceedings of the 2012 International Conference on Functional Programming (ICFP 2012), September 2012.

Measuring NUMA Effects with the STREAM Benchmark. L. Bergstrom. University of Chicago Computer Science Department Technical Report TR-2012-04, May 2012.

Lazy Tree Splitting. L. Bergstrom, M. Fluet, M. Rainey, J. Reppy, and A. Shaw. Journal of Functional Programming, 2012. This paper is a longer version of the ICFP '10 paper.

Garbage Collection for Multicore NUMA Machines. S.Auhagen, L. Bergstrom, M. Fluet, J. Reppy. In the Proceedings of Memory Systems Performance and Correctness (MSPC), June 2011.

Lazy Tree Splitting. L. Bergstrom, M. Fluet, M. Rainey, J. Reppy, and A. Shaw. In the Proceedings of the International Conference on Functional Programming (ICFP), September 2010.

Arity Raising in Manticore. L. Begstrom and J.Reppy. In the Proceedings of the 21st International Symposium on the Implementation and Application of Functional Languages (IFL), September 2009.