College of Computer,
and Information Science
Northeastern University
202 WVH
Boston, MA 02115
USA

Phone: +1 (646) 301-1825
Fax:
heather@ccs.neu.edu
http://heather.miller.am

HEATHER MILLER

Research Interests

Concurrent, distributed, data-centric, and data-intensive (big data) programming, from the perspective of programming languages. I work on both theoretical ideas & implementations for the Scala programming language which seek to make it easier to build distributed systems.

Education

EPFL, *Lausanne, Switzerland* — *2009 – 2015*
Ph.D. in Computer Science
Advisor: Martin Odersky — *2011 – 2015*

University of Miami, *Coral Gables, FL* — *2006 – 2009*
BSEE in Electrical Engineering, Audio Engineering, *with honors, May 2009*

Cooper Union for the Advancement of Science and Art, *New York, NY* — *2004 – 2006*

Employment

Northeastern University, *Boston, MA, USA* — *9/2016 –*
Assistant Clinical Professor

Scala Center, EPFL, *Lausanne, Switzerland* — *10/2015 –*
Executive Director
Founded a new not-for-profit center dedicated to education, open source development, and research surrounding the Scala programming language.

Databricks, *Berkeley, CA, USA* — *8/2014 – 11/2014*
Research Intern
Supervisor: Matei Zaharia
Integrated Scala Pickling, our framework for fast, boilerplate-free, extensible serialization focused on distributed programming (OOPSLA'13) into Spark. Developed new function-passing programming model and framework, can be thought of as a generalization of Spark/MapReduce programming model.

Teaching Experience

Instructor, *Programming Models for Distributed Computation* — *2016*
Northeastern University PhD-level course on programming models for distributed systems.

Instructor, Co-Designer, *Reactive Programming & Parallelism* — *2015 & 2016*
EPFL Undergraduate course on parallel, distributed, and asynchronous programming (90 – 150 students)

Instructor, Co-Designer, *Parallel Programming & Data Analysis* *2015*
Upcoming Coursera MOOC on parallel, distributed, and asynchronous programming.

Lead, *Functional Programming Principles in Scala* *2012 – 2014*
Popular Coursera MOOC on functional programming in Scala, with >200,000 participants to date & largest completion rate for a course its size (~19%)

- Lead teaching staff organizing a team of graduate students, managing content production, designed course exercises with cloud-hosted grading, production of lecture videos, etc
- Created extensive course analysis with interactive visualizations; led to a publication at ICSE'14

(Lead) Teaching Assistant, *Programming Principles* *2011-2014*
Required EPFL undergraduate course on functional & logic programming (~160 students)

Instructor, *Scala as a Research Tool* *2013*
ECOOP Tutorial

Publications

Function Passing: A Model for Typed, Distributed Functional Programming *SPLASH 2016*
Heather Miller, Philipp Haller, Normen Müller, Joceyln Boullier
ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming & Software

Distributed Programming via Safe Closure Passing *PLACES 2015*
Philipp Haller, Heather Miller
Programming Language Approaches to Communication and Concurrency Centric Systems

Spores: A Type-Based Foundation for Closures in the Age of Concurrency and Distribution *ECOOP 2014*
Heather Miller, Philipp Haller, Martin Odersky
European Conference on Object Oriented Programming

Functional Programming For All! Scaling a MOOC for Students And Professionals Alike *ICSE 2014*
Heather Miller, Philipp Haller, Lukas Rytz, Martin Odersky
ACM SIGSOFT International Conference on Software Engineering

Instant Pickles: Generating Object-Oriented Pickler Combinators for Fast and Extensible Serialization *OOPSLA 2013*
Heather Miller, Philipp Haller, Eugene Burmako, Martin Odersky
ACM SIGPLAN Conference on Object Oriented Programming, Systems, Languages and Applications

RAY: Integrating Rx and Async for Direct-Style Reactive Streams *REM 2013*
Philipp Haller, Heather Miller
ACM SPLASH Workshop on Reactivity, Events and Modularity

FlowPools: A Lock-Free Deterministic Concurrent Dataflow Abstraction *LCPC 2012*
Aleksandar Prokopec, Heather Miller, Tobias Schlatter,
Philipp Haller, Martin Odersky
International Workshop on Languages and Compilers for Parallel Computing
Invited to Revised Selected Papers on the 25th International Workshop on Languages and Compilers for Parallel Computing, Lecture Notes in Computer Science, Vol. 7760, 2013

Tools and Frameworks for Big Learning in Scala: Leveraging the Language for High Productivity and Performance *BigLearn 2011*
Heather Miller, Philipp Haller, Martin Odersky
NIPS Workshop on Parallel and Large-Scale Machine Learning

Parallelizing Machine Learning – Functionally: A Framework and Abstractions for Parallel Graph Processing *Scala 2011*
Philipp Haller, Heather Miller
Scala Workshop

Submitted/In Preparation

A Programming Model and Foundation for Lineage-Based Distributed Computation
Heather Miller, Philipp Haller, Normen Müller

Specialising Parsers for Queries
Manohar Jonnalagedda, Jorge Vicente Cantero, Heather Miller, Martin Odersky

Self-Assembly: Lightweight Language Extension and Datatype Generic Programming, All-in-One!
Heather Miller, Philipp Haller, Bruno C. d. S. Oliveira

Improving Human-Compiler Interaction Through Customizable Type Feedback
Hubert Plociniczak, Heather Miller, Martin Odersky

Selected Tech Reports

The Function Passing Model: Types, Proofs, and Semantics
Philipp Haller, Normen Müller, Heather Miller
September 2016

Spores, Formally
Heather Miller, Philipp Haller
December 2013

FlowPools: A Lock-Free Deterministic Concurrent Dataflow Abstraction – Proofs
Aleksandar Prokopec, Heather Miller, Philipp Haller
June 2012

Open Source

Scala Programming Language, *member of the Scala team* *2011 –*

- **Scala Spores** (Scala Improvement Proposal SIP-21), ***project lead***
  novel type-based abstraction for using closures safely
  in concurrent and distributed environments

- **Scala Pickling**, ***project lead***
  novel framework for fast, boilerplate-free, extensible serialization.
  Adopted by sbt, the most widely-used build tool for Scala. Popular
  open-source project on GitHub with >480 stars & dozens of contributors
- **Scala Futures & Promises** (Scala Improvement Proposal SIP-14), ***team member***
  unified non-blocking concurrency substrate for
  Scala, Akka, Play, and others
- **Scala Documentation**, ***creator, writer, lead maintainer***
  a central website for community-driven documentation for
  the Scala programming language and core libraries
- **Scaladoc**, ***co-maintainer***
  documentation tool for Scala's official API documentation

Honors

US National Science Foundation Graduate Research Fellowship	*2011 – 2014*
EPFL Outstanding Teaching Award	*2012*
EPFL Computer Science Fellowship	*2009 – 2010*
Most Outstanding Audio Engineering Student, University of Miami	*2009*
Most Outstanding Eta Kappa Nu Student, University of Miami	*2009*
Information Technology Scholarship, University of Miami	*2006 – 2009*
John Farina Family Scholarship, University of Miami	*2006 – 2009*
Eta Kappa Nu	*2008*
Tau Beta Pi	*2008*
SMART US Department of Defense Scholarship Alternate	*2007*
Cooper Union Full Tuition Scholarship	*2004 – 2006*

Selected Talks

Function Passing Style: Typed, Distributed Functional Programming — *Strange Loop 2014*
St. Louis, MO, USA. September 19, 2014

Spores: A Type-Based Foundation for Closures in the Age of Concurrency and Distribution — *ECOOP 2014*
Uppsala, Sweden. August 1, 2014

Functional Programming For All! Scaling a MOOC for Students and Professionals Alike — *ICSE 2014*
Hyderabad, India. June 4, 2014

Academese to English: Scala's Type System, Dependent Types and What It Means To You — *NEScala 2014*
New York, NY, USA. March 1, 2014

Instant Pickles: Generating Object-Oriented Pickler Combinators for Fast and Extensible Serialization — *OOPSLA 2013*
Indianapolis, IN, USA. October 30, 2013

PL Abstractions for Distributed Programming: Pickle Your Spores! — *Indiana University* ***(invited)***
Bloomington, IN, USA. October 25, 2013

Spores: Distributable Functions in Scala — *Strange Loop 2013*
St. Louis, MO, USA. September 19, 2013

Open Issues in Dataflow Programming — *LaME 2013* ***(invited)***
Montpellier, France. July 1, 2013

Scala as a Research Tool — *ECOOP 2013 Tutorial*
Montpellier, France. July 1, 2013

On Pickles & Spores: Improving Scala's Support for Distributed Programming — *ScalaDays 2013*
New York, NY, USA. June 12, 2013

Futures & Promises in Scala 2.10 — *PhillyETE 2013* ***(invited)***
Philadelphia, PA, USA. April 2, 2013

I am also a frequent speaker in industry, at industrial conferences, developer "meet-ups", and everything in between. Some such events include:

f(by) (11/2014, Minsk, Belarus), SF Scala (11/2014, SF, USA), Scalapeño (9/2014, Tel Aviv, Israel), SoundCloud TechTalks (7/2014, Berlin, Germany), **Scala Days** (6/2014, Berlin, Germany), NEScala (3/2014, NYC, USA), amongst others.

External Activities

Scalawags Monthly Podcast, co-host — *2014 –*

External Service

SLE 2016, PC member — *11/2016*
Scala 2016, PC member — *11/2016*
REBLS 2016, PC member — *11/2016*
Curry On 2016, organizer (co-chair) — *7/2016*
ECOOP 2016, ERC member, organizing committee member (sponsorship) — *7/2016*
PMLDC 2016, co-chair — *7/2016*
TFP 2016, PC member — *6/2016*
SAC 2016, PC member — *4/2016*
Curry On 2015, organizer (co-chair) — *7/2015*
ECOOP 2015, organizing committee member (sponsorship) — *7/2015*
PLE 2015, program committee member — *7/2015*
DSLDI 2015, program committee member — *7/2015*
Scala Symposium 2015, organizer (co-chair) — *6/2015*
POPL 2015, artifact evaluation committee member — *1/2015*
Scala Workshop 2014, organizer (co-chair) — *7/2014*
Scala Workshop 2013, organizer (co-chair) — *7/2013*

External Reviewer for: ECOOP 2013, Scala 2013
Editor of proceedings for: Scala 2015, Scala 2014, Scala 2013

Students Supervised[1]

Joceyln Boullier, *Evaluating the Efficiacy of the Function Passing Model* *2/2016 – 8/2016*
M.Sc. thesis

Jorge Vicente Cantero, *Implementing the Function Passing Model* *2/2016 – 6/2016*
B.Sc. thesis

Louis Bliss, *Incremental Picklers for Scala Pickling* *9/2013 – 1/2014*
M.Sc. level, co-supervision with Philipp Haller

Thaddée Yann Tyl, *Learning Scala Style* *2/2013 – 6/2013*
M.Sc. thesis

Tobias Schlatter, *FlowSeqs: Barrier-Free ParSeqs* *9/2012 – 1/2013*
M.Sc. level, co-supervision w/ Philipp Haller & Aleksandar Prokopec

Tobias Schlatter, *Multi-Lane FlowPools* *2/2012 – 6/2012*
M.Sc. level, co-supervision w/ Philipp Haller & Aleksandar Prokopec

Pierre Grydbeck, *Parallel Machine Learning: An Expectation Maximization Algorithm for Gaussian Mixture Models* *2/2012 – 6/2012*
M.Sc. level, co-supervision with Philipp Haller

Bruno Studer, *Parallel Machine Learning: Collaborative Filtering via Alternating Least Squares* *2/2012 – 6/2012*
B.Sc. level, co-supervision with Philipp Haller

Stanislav Peshterliev, *Parallel Natural Language Processing Algorithms in Scala* *9/2011 – 1/2012*
M.Sc. level, co-supervision with Philipp Haller

Olivier Blanvillain & Louis Bliss, *Parallelization of a Collaborative Filtering Algorithm with Menthor* *9/2011 – 1/2012*
B.Sc. level, co-supervision with Philipp Haller

Florian Gysin, *Improving Parallel Graph Processing Through the Introduction of Parallel Collections* *9/2011 – 1/2012*
M.Sc. level, co-supervision with Philipp Haller

Georges Discry, *Extending the Menthor Framework for Parallel Graph Processing to Distributed Computing* *2/2011 – 6/2011*
M.Sc. level, co-supervision with Philipp Haller

References

Martin Odersky
Faculty of Computer, Communication, and Information Science
École Polytechnique Fédérale de Lausanne
☎ +41 21 693 68 63
✉ martin.odersky@epfl.ch

[1] At EPFL, research groups offer substantial projects for B.Sc./M.Sc. students to complete for credit. EPFL PhD students design and supervise these projects, as well as M.Sc. thesis projects.

Philipp Haller
School of Computer Science and Communication
KTH Royal Institute of Technology
☎ +41 76 205 39 32
✉ phaller@kth.se

Matei Zaharia
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
☎ +1-510-610-0001
✉ matei@mit.edu

Marius Eriksen
Twitter
✉ marius@twitter.com