

Philipp Haller

Chemin du Bochet 46
CH-1025 St-Sulpice
Switzerland

+41 76 205 3932
philipp.haller@a3.epfl.ch
<http://lampwww.epfl.ch/~phaller/>

Interests

Programming languages, type systems, concurrency, parallelism

My research has focused on the design, implementation, and foundations of concurrent programming models that support the construction of efficient, scalable, and safe real-world systems.

Work

Typesafe, Switzerland

2012 – present

Experience

Consultant and software engineer

Stanford University, USA

2011 – 2012

Department of Electrical Engineering
Post-doctoral fellow

EPFL, Switzerland

School of Computer and Communication Sciences

Post-doctoral researcher

2011 – 2012

Research assistant

2006 – 2010

Education

EPFL, Switzerland

2006 – 2010

Doctor of Science in Computer Science

Advisor: Martin Odersky

Thesis title: “Isolated Actors for Race-free Concurrent Programming”

Karlsruhe Institute of Technology, Germany

2001 – 2006

Dipl.-Inform. in Computer Science, awarded highest distinction

Awards &

Nominations

- Nomination for EPFL Doctorate Award 2010
School of Computer and Communication Sciences, EPFL
- Prize of Excellence (exceptional teaching contribution) 2008
School of Computer and Communication Sciences, EPFL
- Best Student Paper Award 2007
9th International Conference Coordination Models and Languages
- e-fellows Scholarship 2004 – 2010
Deutsche Telekom AG and McKinsey & Company
- Full Scholarship 2001 – 2006
German National Academic Foundation
- Karls-Preis (exceptional social contribution) 2000
Karls-Gymnasium Stuttgart, Germany

Research

Typesafe, Switzerland

2012 – present

Asynchronous and concurrent programming:

- *Scala Async (project co-lead)*, an industrial-strength, optimized implementation of asynchronous continuations
- *Scala Futures (specification lead)*, a non-blocking implementation of futures with pluggable execution environments

Data-centric and reactive programming:

- *Scala Pickling*, a framework for generating object-oriented pickler combinators for fast and extensible serialization
- *RAY*, a programming model that integrates the Reactive Extensions model and the Async model of C#/Scala

Stanford University, USA

2011 – 2012

Embedded domain-specific languages (DSLs):

- *Scala-Virtualized*, an experimental extension of the mainline Scala compiler specialized for embedding high-level DSLs
- *Delite Debugging and Profiling Extensions*, enabling rich debugging and profiling capabilities for parallel DSLs embedded in Scala based on the Delite DSL framework

EPFL, Switzerland

2006 – 2012

Concurrent programming:

- *Scala Actors*, an actor-based programming model that unifies thread-based and event-based programming
- *Scala Joins*, an implementation of join-calculus-style synchronization

Type systems:

- *Capabilities for Uniqueness and Borrowing*, a uniqueness type system suitable for full-featured languages like Scala that allows enforcing actor isolation using static capabilities

Teaching

Experience

UPMARC Multicore Computing Summer School

2011

Invited lecturer

Designed and taught a series of lectures for graduate students on multicore programming in Scala. A majority of the lectures were based on libraries and frameworks that I developed as part of my graduate work (Scala Actors, uniqueness types).

International Summer School on Trends in Concurrency

2008

Lecturer

Helped design and teach a series of lectures for graduate students on concurrent programming in Scala using actors and joins. All lectures were based on my graduate work.

EPFL, Switzerland

2006 – 2009

Teaching Assistant, Prof. Martin Odersky

Duties as head teaching assistant included shared administrative responsibilities with faculty instructor, fielding of all student inquiries, and oversight of student teaching assistants.

- Compiler Construction, Fall 2006, Fall 2007
- Advanced Programming, Spring 2007, Fall 2007 (head TA)
- Foundations of Software (M.Sc. level), Fall 2008 (head TA)
- Foundations of Software (M.Sc. level), Fall 2009 (sole TA)
 - Designed new student project using the Coq proof assistant

Industry
Experience**Typesafe, Switzerland**

2012 – present

- Open-source contributions to Scala and Akka (see below)
- Consulting for leading technology companies, start-ups, and investment banks in the USA, UK, Spain, and Argentina. Topics: high-performance, fault-tolerant systems based on actors and non-blocking futures, efficient and scalable immutable data structures
- Directing a new commercial offering for personalized tutorials for the Coursera MOOCs “Functional Programming Principles in Scala” and “Principles of Reactive Programming”
- Introductory and advanced trainings on Scala and Akka, Typesafe’s event-driven middleware for highly-scalable concurrent and distributed fault-tolerant systems

Open Source

Committer, Scala Project

- Project co-lead of the Scala Async extension
- Specification lead of the futures package of Scala 2.10
- Author of the original actors library (part of Scala’s standard library since 09/2006)
- Author of partest (used for all check-in and nightly tests since 02/2008)

Akka Project, Typesafe

- I contributed the design and implementation of the Stash extension for actors that allows modeling Erlang-style processes using Akka actors

Publications

Journal Papers**HOSC 2013**

- Tiark Rompf, Adriaan Moors, Nada Amin, Philipp Haller, and Martin Odersky. “Scala-Virtualized: Linguistic Reuse for Deep Embeddings.” *Higher-Order and Symbolic Computation*, September 2013.

TCS 2009

- Philipp Haller and Martin Odersky. “Scala Actors: Unifying Thread-based and Event-based Programming.” *Theoretical Computer Science, Volume 410, Numbers 2-3*, February 2009. (Google Scholar: **184 citations**) Significantly extended version of the COORDINATION’07 paper below.

Refereed Conference Papers

- ECOOP 2014**
- Heather Miller, Philipp Haller, and Martin Odersky. “Spores: A Type-Based Foundation for Closures in the Age of Concurrency and Distribution.” *Proceedings of the 28th European Conference on Object-Oriented Programming*, 2014.
- ICSE SEET 2014**
- Heather Miller, Philipp Haller, Lukas Rytz, and Martin Odersky. “Functional Programming For All! Scaling a MOOC for Students and Professionals Alike.” *Proceedings of the 36th International Conference on Software Engineering, Software Engineering Education and Training (SEET) Track*, 2014.
- OOPSLA 2013**
- Heather Miller, Philipp Haller, Eugene Burmako, and Martin Odersky. “Instant Pickles: Generating Object-Oriented Pickler Combinators for Fast and Extensible Serialization.” *Proceedings of the 27th ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications*, 2013.
- ECOOP 2012**
- Lukas Rytz, Martin Odersky, and Philipp Haller. “Lightweight Polymorphic Effects.” *Proceedings of the 26th European Conference on Object-Oriented Programming*, 2012.
- ECOOP 2010**
- Philipp Haller and Martin Odersky. “Capabilities for Uniqueness and Borrowing.” *Proceedings of the 24th European Conference on Object-Oriented Programming*, 2010. (Google Scholar: **42 citations**)
- COORDINATION 2008**
- Philipp Haller and Tom Van Cutsem. “Implementing Joins Using Extensible Pattern Matching.” *Proceedings of the 10th International Conference on Coordination Models and Languages*, 2008. (Google Scholar: **30 citations**)
- COORDINATION 2007**
- Philipp Haller and Martin Odersky. “Actors that Unify Threads and Events.” *Proceedings of the 9th International Conference on Coordination Models and Languages*, 2007. (**Best student paper award**)
The above paper has also been featured on Lambda the Ultimate, and in articles on IBM developerWorks, and DZone (more than **70,000 hits!**)
- JMLC 2006**
- Philipp Haller and Martin Odersky. “Event-based Programming Without Inversion of Control.” *Proceedings of the 7th Joint Modular Languages Conference*, 2006. (Google Scholar: **80 citations**)
The above paper has also been featured on Lambda the Ultimate (more than **20,000 hits!**), and in an article on IBM developerWorks.

Refereed Workshops (Excluding Short Versions of Papers Above)

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

- LCPC 2012**
- Aleksandar Prokopec, Heather Miller, Tobias Schlatter, Philipp Haller, and Martin Odersky. “FlowPools: A Lock-free Deterministic Concurrent Dataflow Abstraction.” *25th International Workshop on Languages and Compilers for Parallel Computing*, 2012.
Invited to *Revised Selected Papers of the 25th International Workshop on Languages and Compilers for Parallel Computing, Lecture Notes in Computer Science, Volume 7760*, 2013.
- Big Learning 2011**
- Heather Miller, Philipp Haller, and Martin Odersky. “Tools and Frameworks for Big Learning in Scala: Leveraging the Language for High Productivity and Performance.” *NIPS 2011 Workshop on Parallel and Large-scale Machine Learning*, 2011.
- SCALA 2011**
- Philipp Haller and Heather Miller. “Parallelizing Machine Learning – Functionally: A Framework and Abstractions for Parallel Graph Processing.” *2nd Scala Workshop*, 2011.
- CAP 2010**
- Philipp Haller. “Static Debugging of Programs Using High-level Concurrency Libraries.” *ACM SPLASH 2010 Workshop on Concurrency for the Application Programmer*, 2010. Position Paper.
- SCALA 2010**
- Philipp Haller. “Lightweight Language Support for Type-based, Concurrent Event Processing.” *1st Scala Workshop*, 2010.

Invited Papers

- AGERE! 2012**
- Philipp Haller. “On the Integration of the Actor Model in Mainstream Technologies: The Scala Perspective.” *Proceedings of the 2nd International Workshop on Programming based on Actors, Agents, and Decentralized Control*, 2012.

Books

- Philipp Haller and Frank Sommers. “Actors in Scala.” *Artima Inc.*, January 2012

Manuscripts under Submission

- Heather Miller, Philipp Haller, and Martin Odersky. “Spores: A Type-Based Foundation for Closures in the Age of Concurrency and Distribution.”

Other Work (Unrefereed)

- Michel Schinz and Philipp Haller. “A Scala Tutorial for Java Programmers.” *Scala Website (scala-lang.org)*. This popular tutorial has been translated to Spanish, Japanese, and German.

Research Students Supervised

- M.Sc. level, 09/2007 to 01/2008, Thomas Hofer, “Comet-style Web Applications in Scala”

- M.Sc. level, 09/2007 to 01/2008, Thibaud Hottelier, “A Flexible Regular Expression Library for Scala”
- M.Sc. level, 09/2007 to 01/2008, Federico Marmorì, “Remote Actors for Distributed Programming”
- B.Sc. level, 09/2007 to 01/2008, Antoine Yersin, “Scala AIO: New Facilities on top of Java NIO Programming”
- M.Sc. level, 02/2009 to 06/2009, Rémi Bonnet, “Alias Tracking through Capabilities”
- M.Sc. level, 02/2011 to 06/2011, Georges Discry, “Extending the Mentor Framework for Parallel Graph Processing to Distributed Computing,” co-supervision with Heather Miller (EPFL)
- M.Sc. thesis, 09/2011 to 03/2012, Christophe Pache, “Distribution of Workflows using an Actor Model Framework”
- M.Sc. level, 09/2011 to 01/2012, Florian Gysin, “Improving Parallel Graph Processing through the Introduction of Parallel Collections,” co-supervision with Heather Miller (EPFL)
- M.Sc. level, 09/2011 to 01/2012, Stanislav Peshterliev, “Parallel Natural Language Processing Algorithms in Scala,” co-supervision with Heather Miller (EPFL)
- M.Sc. level, 09/2011 to 01/2012, Eric Zbinden, “Natural Language Programming Analysis in Scala,” co-supervision with Philippe Suter (IBM Research)
- B.Sc. level, 02/2012 to 06/2012, Bruno Studer, “Parallel Machine Learning: Collaborative Filtering via Alternating Least Squares,” co-supervision with Heather Miller (EPFL)
- M.Sc. level, 02/2012 to 06/2012, Pierre Grydbeck, “Parallel Machine Learning: An Expectation Maximization Algorithm for Gaussian Mixture Models,” co-supervision with Heather Miller (EPFL)
- M.Sc. level, 02/2012 to 06/2012, Tobias Schlatter, “Multi-Lane FlowPools,” co-supervision with Heather Miller and Aleksandar Prokopec (EPFL)
- M.Sc. level, 09/2012 to 01/2013, Tobias Schlatter, “FlowSeqs: Barrier-Free ParSeqs,” co-supervision with Heather Miller and Aleksandar Prokopec (EPFL)
- M.Sc. level, 09/2013 to 01/2014, Louis Bliss, “Incremental Picklers for Scala Pickling,” co-supervision with Heather Miller (EPFL)
- M.Sc. thesis, since 02/2014 (ongoing), Alon Dolev (external student of Erik Meijer), “Advanced Control Operators for Reactive Programming”

External Visibility and Professional Activities

Invited talks

- Invited talk, *ECOOP Summer School*, July 2014, Uppsala, Sweden
- Invited talk, *The Next 700 Asynchronous Programming Models*, *ACM SPLASH-I Series*, October 2013, Indianapolis, IN, USA

- Invited talk, Correctly and Efficiently Combining Concurrency Abstractions, *Skills Matter*, December 2012, London, UK
- Invited talk, Futures and Promises – A New Take on Concurrency in Scala 2.10, *Scala eXchange*, November 2012, London, UK
- Keynote talk, On the Integration of the Actor Model in Mainstream Technologies: The Scala Perspective, *ACM SIGPLAN AGERE! Workshop*, October 2012, Tucson, AZ, USA
- Invited talk, The Many Flavors of Parallel Programming in Scala, *Scalathon*, July 2011, U. of Pennsylvania, Philadelphia, PA, USA
- Invited talk, Capabilities for External Uniqueness, *Saarland University*, May 2009, Saarbrücken, Germany
- Invited talk, Concurrent Programming in Scala, *Universität Freiburg*, November 2008, Freiburg, Germany
- Invited talk, An Overview of Scala, *IT University of Copenhagen*, June 2008, Copenhagen, Denmark

Conference &
Workshop
Committees

- AGERE! 2014: 4th International Workshop on Programming based on Actors, Agents, and Decentralized Control (affiliated with SPLASH 2014), **co-organizer**
- REBLS 2014: Workshop on Reactive and Event-based Languages & Systems (affiliated with SPLASH 2014)
- SCALA 2014: Scala Workshop (affiliated with ECOOP 2014), **co-chair**
- PLPV 2014: ACM SIGPLAN Workshop on Programming Languages meets Program Verification (affiliated with POPL 2014)
- AGERE! 2013: 3rd International Workshop on Programming based on Actors, Agents, and Decentralized Control (affiliated with SPLASH 2013)
- REM 2013: Workshop on Reactivity, Events and Modularity (affiliated with SPLASH 2013)
- ECOOP 2013: European Conference on Object-Oriented Programming
- SCALA 2013: Scala Workshop (affiliated with ECOOP 2013), **co-chair**
- ACME 2013: Workshop on ACadeMics Tooling with Eclipse (affiliated with ECOOP 2013)
- AGERE! 2012: 2nd International Workshop on Programming based on Actors, Agents, and Decentralized Control (affiliated with SPLASH 2012)
- LaME 2012: International Workshop on Languages for the Multi-core Era (affiliated with ECOOP 2012)

External
Reviewing

- ESOP 2011, 2014, Science of Computer Programming 2012, 2013, OOPSLA 2007, 2011, 2012, FMOODS-FORTE 2011, CC 2011, CAV 2010, COORDINATION 2008, TRANSACT 2007

- Presentations
- Simplifying Asynchronous Code With Scala Async, *Strange Loop 2013*, September 2013, St. Louis, MO, USA
 - Scala Async: A New Way to Simplify Asynchronous Code (Make the Compiler Do It!), *Scala Days 2013*, June 2013, New York City, NY, USA
 - Futures and Promises in Scala 2.10, *Boston Area Scala Enthusiasts*, August 2012, MIT, Boston, MA, USA
 - Parallelizing Machine Learning – Functionally, *Scala Workshop 2011*, June 2011, Stanford, CA, USA
 - Capabilities for Uniqueness and Borrowing, *ECOOP 2010*, June 2010, Maribor, Slovenia
 - Actor-Based Concurrency in Scala, *JavaOne 2009*, June 2009, San Francisco, CA, USA
 - Implementing Joins using Extensible Pattern Matching, *DAMP 2008 (affiliated with POPL 2008)*, January 2008, San Francisco, CA, USA
 - Actors that Unify Threads and Events, *COORDINATION 2007*, June 2007, Paphos, Cyprus
 - Scala Actors – Scalable Multithreading on the JVM, *IBM Development Lab Böblingen*, May 2007, Böblingen, Germany

Professional Memberships

Professional Member, ACM
Member, ACM SIGPLAN
Member, German Informatics Society (GI)