Jose Manuel Faleiro

Department of Computer Science
Yale University
51 Prospect Street
New Haven, CT 06511

Email: faleiro.jose.manuel@gmail.com
Phone: +1 203-676-9255
Web: http://www.jmfaleiro.com/

Research Interests

Parallel Programming Models, Database Systems, Operating Systems, Distributed Systems

Education

Yale University — Aug 2012 - present
PhD Computer Science (Masters expected in Dec 2014)
Advised by Daniel Abadi and Bryan Ford

Birla Institute of Technology and Science, Pilani, INDIA — Aug 2007-Jun 2011
B.E.(Honors) Computer Science
CGPA: **9.68/10**

Awards and Honors

Alan J. Perlis Fellowship 2012-2013 and 2013-2014
Awarded to top computer science PhD students at Yale University

Goa Scholars Scheme 2012-2013
Awarded to 10 students of Goan origin pursuing higher studies outside India

Microsoft Research Tech Transfer Award
Awarded for deploying DebugAdvisor [link] to Microsoft's Lync team [link]

Birla Institute of Technology and Science Merit Scholarship, Spring 2010
Awarded for outstanding academic performance

Publications

Lazy Evaluation of Transactions in Database Systems
Jose M. Faleiro, Alexander Thomson, Daniel J. Abadi
SIGMOD 2014

Generalized Lattice Agreement
Jose M. Faleiro, Sriram Rajamani, Kaushik Rajan, Ganesan Ramalingam, Kapil Vaswani
PODC 2012

CScale – A Programming Model for Scalable and Reliable Distributed Applications
Jose Faleiro, Sriram Rajamani, Kaushik Rajan, Ganesan Ramalingam, Kapil Vaswani
Monterey Workshop 2012

Professional Experience

Research Intern — Jun 2014 - Aug 2014
Microsoft Research Redmond

Research Intern — Jun 2013 - Aug 2013
Microsoft Research Silicon Valley

Research Developer — Jul 2011 - Jul 2012
Microsoft Research India

Research Intern — Jan 2011 - Jun 2011
Microsoft Research India

Research Experience

Geodistribution in the Orleans Programming Model Jun 2014 - Aug 2014
With Philip Bernstein, Sergey Bykov and Gabriel Kliot, at Microsoft Research Redmond
Orleans [link] is an actor-based distributed programming model that simplifies building large-scale, stateful distributed systems. Designed and implemented an extension to the Orleans runtime to allow actors (and applications) to span more than one datacenter.

Locality Preserving Distributed Systems [Draft] Jan 2014 - present
With Bryan Ford and Michael Nowlan, at Yale University
Distributed systems achieve scalability by balancing load across many machines, but wide-area distribution can introduce worst-case response latencies proportional to the network's delay diameter. Designed and implemented a general framework to build *locality preserving* distributed systems, by systematically structuring large distributed systems so that the latency of interactions between any pair of nodes is proportional to the network delay between them.

Lazy Transactions [Paper] [Slides] Jan 2013 - Mar 2014
With Daniel Abadi and Alexander Thomson, at Yale University
Inspired by lazy evaluation in programming languages, investigated the performance tradeoffs of deferred transaction execution in a database system. Designed and implemented a prototype system to evaluate the feasibility of lazy transaction processing. Our technique improves data-cache locality, is able to elegantly deal with transient load spikes, and improves concurrency in high-contention workloads.

Multicore Synchronization Performance [Slides] Jun 2013 - Aug 2013
With Paul Barham and Rebecca Isaacs, at Microsoft Research Silicon Valley
Investigated heuristics to identify poor parallel program performance due to synchronization overhead. Microbenchmarked several .NET concurrent data-structures to understand their behavior under varying workloads. Devised and evaluated a lightweight instrumentation technique to correlate poor performance in parallel programs with contention induced back-offs in lock implementations.

Stronger Semantics for Eventual Consistency [Paper] [Paper] Jun 2011 - Jun 2012
With Sriram Rajamani, Kaushik Rajan, Ganesan Ramalingam and Kapil Vaswani, at Microsoft Research India
Contributed to a distributed programming model built on commutative replicated data-types (CRDTs), a class of eventually consistent distributed data-structures. Contributed to the design of a protocol that builds serializable state machines from eventually consistent data-structures (such as CRDTs).