G-CSC Kolloquium

Programming with People: Integrating Human-Based and Digital Computation

Emery Berger, Computer Science, University of Massachusetts Amherst

Abstract
Humans can perform many tasks with ease that remain difficult or impossible for computers. Crowdsourcing platforms like Amazon's Mechanical Turk make it possible to harness human-based computational power on an unprecedented scale. However, their utility as a general-purpose computational platform remains limited. The lack of complete automation makes it difficult to orchestrate complex or interrelated tasks. Scheduling human workers to reduce latency costs real money, and jobs must be monitored and rescheduled when workers fail to complete their tasks. Furthermore, it is often difficult to predict the length of time and payment that should be budgeted for any given task.
Crucially, the results of human-based computations are not necessarily reliable, both because human skills and accuracy vary widely, and because workers have a financial incentive to minimize their effort.

This talk presents AutoMan, the first fully automatic crowdprogramming system. AutoMan integrates human-based computations into a standard programming language as ordinary function calls, which can be intermixed freely with traditional functions. This abstraction allows AutoMan programmers to focus on their programming logic. An AutoMan program specifies a confidence level for the overall computation and a budget. The AutoMan runtime system then transparently manages all details necessary for scheduling, pricing, and quality control. AutoMan automatically schedules human tasks for each computation until it achieves the desired confidence level; monitors, reprices, and restarts human tasks as necessary; and maximizes parallelism across human workers while staying under budget.

AutoMan is available for download at www.automan-lang.org.

Biography:

Emery Berger is an Associate Professor in the Department of Computer Science at the University of Massachusetts Amherst, the flagship campus of the UMass system. He graduated with a Ph.D. in Computer Science from the University of Texas at Austin in 2002. Professor Berger has been a Visiting Scientist at Microsoft Research and at the Universitat Politecnica de Catalunya (UPC) / Barcelona Supercomputing Center (BSC).

Professor Berger's research spans programming languages, runtime systems, and operating systems, with a particular focus on systems that transparently improve reliability, security, and performance. He is the creator of a number of influential software systems including Hoard, a fast and scalable memory manager that accelerates multithreaded applications (used by companies including British Telecom, Cisco, Crédit Suisse, Reuters, Royal Bank of Canada, SAP, and Tata, and on which the Mac OS X memory manager is based); DieHard, an error-avoiding memory manager that directly influenced the design of the Windows 7 Fault-Tolerant Heap; and DieHarder, a secure memory manager that was an inspiration for hardening changes made to the Windows 8 heap.

His honors include a Microsoft Research Fellowship, an NSF CAREER Award, a Lilly Teaching Fellowship, a Most Influential Paper Award at OOPSLA 2012, and a Google Research Award; he was named an ACM Senior Member in 2010. Professor Berger served as the General Chair of the Memory Systems Performance and Correctness workshop (MSPC 2008), Program Chair of the 2010 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE 2010), Program Chair of the 2012 Workshop on Determinism and Correctness in Parallel Programming (WoDET 2012), and is currently co-Program Chair of the Fifth USENIX Workshop on Hot Topics in Parallelism (HotPar 2013). He is also currently an Associate Editor of the ACM Transactions on Programming Languages and Systems.