

Technical Perspective

Computing with the Crowd

By Siddharth Suri

COMPUTER SCIENCE IS primarily focused on computation using microprocessors or CPUs. However, the recent rise in the popularity of crowdsourcing platforms, like Amazon's Mechanical Turk, provides another computational device—the crowd. Crowdsourcing is the act of outsourcing a job to an undefined group of people, known as the crowd, through an open call.³ Crowdsourcing platforms are online labor markets where employers can post jobs and workers can do jobs for pay, but they can also be viewed as distributed computational systems where the workers are the CPUs and will perform computations for pay. In other words, crowdsourcing platforms provide a way to execute computation with humans. In a traditional computational system when a programmer wants to compute something, they interact with a CPU through an API defined by an operating system. But in a crowdsourcing environment, when a programmer wants to compute something, they interact with a human through an API defined by a crowdsourcing platform.


Why might one want to do computation with humans? There are a variety of problems that are easy for humans but difficult for machines. Humans have pattern-matching skills and linguistic-recognition skills that machines have been unable to match as of yet. For example, FoldIt¹ is a system where people search for the natural configuration of proteins and their results often outperform solutions computed using only machines. Conversely, there are problems that are easy for machines to solve but difficult for humans. Machines excel at computation on massive datasets since they can do the same operations repeatedly without getting tired or hungry. This brings up the natural question: What kinds of problems can be solved with both human and machine computation that neither could do alone?

Systems like AutoMan, described in the following paper by Barowy et al., provide the first steps toward answering this question. AutoMan is a domain-specific programming language that provides an abstraction layer on top of the crowd. It allows the programmer to interleave the expression of computation using both humans and machines in the same program. In an AutoMan program, one function could be executed by a CPU and the next could be executed by humans.

This new type of computation brings new types of complexity, which AutoMan is designed to manage. Most of this complexity stems from the fact that unlike CPUs, humans have agency. They make decisions; they have needs, wants, and biases. Humans can choose what tasks to do, when to quit, what is and isn't worth their time, and when to communicate with another human and what about. CPUs, on the other hand, always execute whatever instructions they are given. Much of the design and implementation of AutoMan addresses this key difference between humans and machines. For example, AutoMan has extensive functionality for quality control on the output of the workers. It also has functionality to discover the price that will be enough to incentivize workers to do the given task and to reduce collusion among workers. Computation with CPUs does not require any of this functionality. AutoMan also addresses the natural difference in speed between human and machine computation by allowing eager evaluation of the machine commands and only blocking on the humans when necessary.

Being able to express human computation and interleave human and machine computation opens up interesting new research directions in human computation and organizational dynamics. In the nascent field

of human computation, since we can now express human computation in a programming language, we can next develop a model of human computation analogous to the PRAM.² This would, in turn, allow us to develop a theory of complexity for human computation to help us understand what problems are easy and difficult for humans to solve. Developing these theories might help us scale up AutoMan, which is currently designed to solve microtasks, in terms of complexity to solve bigger tasks and workflows.

Taking a broader and more interdisciplinary perspective, one can view a company as a computational device that combines the human computation of its employees with the machine computation of the company's computers. A better theoretical and empirical understanding of human computation could allow the field of computer science to inform how best to architect and organize companies for greater accuracy and efficiency. Whether or not AutoMan proves revolutionary as a programming language, it is important as an idea because it provides a “computational lens”⁴ on the science of crowdsourcing, human computation, and the study of group problem solving. 

References

1. Cooper, S. et al. Predicting protein structures with a multiplayer online game. *Nature* 446 (Aug. 2010), 756–760.
2. Fortune, S. and Wylie, J. Parallelism in random access machines. In *Proceedings of the 10th Annual Symposium on Theory of Computing* (1978). ACM, 114–118.
3. Howe, J. The rise of crowdsourcing. *Wired* (June 1, 2006).
4. Karp R.M. Understanding science through the computational lens. *J. Computer Science and Technology* 26, 4 (July 2011), 569–577.

Siddharth Suri (suri@microsoft.com) is Senior Researcher at—and one of the founding members of—Microsoft Research in New York City.

Copyright held by author.