Mor Harchol-Balter

Computer Science Department, Carnegie Mellon University

I have always been interested in finding better designs for computer systems that improve performance without the purchase of additional resources. When I look back at the problems that I have solved, and I look ahead to the problems I hope to solve, I realize that the problem formulations keep getting simpler and simpler, and my footing less secure. Every wisdom that I once believed, I have now come to question: If a scheduling policy helps one set of jobs, does it necessarily hurt some other jobs, or are these "conservation laws" being misinterpreted? Do greedy strategies make sense in server farms, or is what's good for the individual actually disastrous for the system as a whole? A single fast machine is more expensive than n *slow machines, each of* $1/n^{th}$ *the speed, but does that mean it is necessarily better? Should distributed systems really aim to balance load, or is this a convenient myth? Under threshold-based cycle stealing, where a "donor" machine helps a "beneficiary" machine based on some threshold on the number of jobs, who should control this threshold: the donor or the beneficiary? What can we do to combat workload variability, and when do we even want to? Will using an open workload generator yield the same results as using a closed one, given the same load, or do we need to be much more careful? Inherent in these questions is the impact of real user behaviors and real-world workloads with heavy tails and high variability. Also intertwined in my work are the tensions between theoretical analysis and the realities of implementation, each motivating the other. In my search to discover new research techniques which allow me to answer these and other questions, I find that I am converging towards the fundamental core that defines all these problems and that makes the counterintuitive more believable.*

This document consists of eight brief examples of my research. Each example deals with a very basic, fundamental question about *computer system design*, typically involving resource allocation. Many of the results *challenge common wisdom and intuition* about system design. Most results involve a mixture of **analysis** (queueing theory, stochastic processes, Markov chains), **implementation** (kernel-level scheduling in web servers, databases, and distributed systems implementations), and **workload characterization** (measurements of UNIX job lifetimes, supercomputing workloads, and studies of web workload generation models).

On the theoretical front, my research has frequently necessitated the invention of *new analytical techniques*, rather than the application of existing techniques. Existing queueing theoretic techniques are typically designed with manufacturing and human service applications in mind (queueing theory is most often taught in operations research departments or business schools), and hence are not always well-suited or sufficient for computer science problems. Briefly, some examples of new analytical queueing techniques which I have invented or co-invented include: (i) the single-queue-approximation technique for analyzing processor-sharing (time-sharing) server farms, which are prevalent in computer science, yet non-existent in operations research; (ii) the dimensionality reduction technique for analyzing Markov chains that are unbounded (infinite) in multiple dimensions, arising in computer science problems like threshold-based cycle stealing and multiserver systems with priorities; (iii) new scheduling analysis techniques, which allow the analysis of all SMART scheduling policies, a class of policies which is much more indicative of computer scheduling today than the idealized policies in textbooks; (iv) a host of new techniques for analyzing systems under fluctuating load conditions; (v) techniques for analyzing the *fairness* of scheduling policies, a metric which has proven to be very important for computer system designers; (vi) techniques for dealing with high-variability, heavy-tailed workloads, which are prevalent in computer systems, and rarely mentioned in most queueing texts.

I am heavily involved in the ACM Sigmetrics research community, which shares my love for combining theoretical and systems work. I have been serving as Treasurer of the Sigmetrics Board since 2005, and am Program Committee Co-Chair of both Sigmetrics 2007 and QEST 2007. I am also the recipient of multiple Sigmetrics best paper awards, which have influenced research directions in the Sigmetrics community and elsewhere. A devout believer in education, I make my Performance Modeling course notes available online. These are currently being used by over 2000 researchers world-wide.