Teaching Statement

Anuj Kalia

The opportunity to teach and mentor students is one of my main reasons for pursuing an academic career. With my background in networked systems research, I am excited to teach undergraduate courses on computer systems, distributed systems, and computer networks. I am also interested in teaching advanced graduate courses on distributed systems, and datacenter networks.

Classroom teaching

As a graduate student at CMU, I was a teaching assistant for two couses. In the advanced operating systems and distributed systems class, I presented two review lectures that covered all research papers discussed in the class. In the parallel computing course, I mentored ten teams during their final class project.

I believe that the best method to teach computer systems is hands-on system building. Computer systems as a field is rife with complex concepts that are deceptively simple at first look (distributed consensus is the foremost example). It has been my experience that a good way to understand such concepts is by implementing them. Courses that I teach will therefore have a strong emphasis on system implementation. I am also interested in teaching the art of building and modifying systems with large codebases, which is often quoted by industry practitioners as the most desired technical skill for students. Including short books such as *A Philosophy of Software Design* [2] as supplementary reading can be helpful in doing so.

The ending of Moore's law means that it is now crucial to teach future programmers how to make the best use of the available silicon. This includes teaching how to build efficient single- and multi-threaded systems, and providing experience with heterogeneous hardware. In CMU's masters-level parallel computing class that I TA-ed, all programming assignments were graded on performance. Students found this experience interesting and fulfilling, and I wish to include such assignments in more introductory courses. Today, academic institutions have free access to large computing clusters with state-of-the-art hardware [3] that can be used for these assignments.

Research mentoring

I believe that effective mentoring should adapt to the student's experience. Students who are starting research need much more guidance and structure than senior graduate students, who perform best when left to explore their own research interests.

During my PhD, I have mentored two masters students during their semester-long capstone research project. My task included formulating a research problem that could be solved in one semester; this helped me learn how to motivate a research problem, and set project direction. My first student, Chao Xin, created an experiment framework for comparing the performance of Remote Direct Memory Access ("RDMA") and simple datagram packet I/O. His experiments confirmed my expectation that the two had similar performance in our cluster setting; this observation formed the basis of our OSDI 2016 paper [1].

My second student, Xin Zhang, studied the relative performance of CPUs and FPGAs for regular expression matching in main-memory database systems. When Xin first started, she was a novice at performance benchmarking, having never even measured the execution time of a C function. At the end of the project, she had successfully integrated three fast pattern matching engines into a massive database codebase, and shown that their performance on a CPU was similar to a prior FPGA-accelerated design. We have submitted a short paper about this work with Xin as the first author to FAST 2019 [4].

References

[1] A. Kalia, M. Kaminsky, and D. G. Andersen. FaSST: Fast, scalable and simple distributed transactions with two-sided RDMA datagram RPCs. In *Proc. 12th USENIX OSDI*, Savannah, GA, Nov. 2016.

[2] J. Ousterhout. *A Philosophy of Software Design*. Yaknam Press, 2018.

[3] R. Ricci, E. Eide, and The CloudLab Team. Introducing CloudLab: Scientific infrastructure for advancing cloud architectures and applications. *USENIX ;login:*, 2014.

[4] X. Zhang, A. Kalia, M. Kaminsky, and D. G. Andersen. A Comparison of CPUs and FPGAs for Database Pattern Matching. Under submission to USENIX FAST, 2019.