Teaching Statement

Peter Bailis

I greatly enjoy teaching and working with students on research, and the ability to continue interacting and researching with students is a key factor in my decision to pursue an academic position.

My primary teaching experience comes from three semesters as a teaching assistant (TA) of Computer Science at the introductory, upper-division undergraduate, and graduate levels. My first exposure to teaching came as a sophomore at Harvard, when I served as a TA for Harvard's introductory Computer Science course, running a weekly discussion, assisting students during office hours, and grading assignments and exams. Helping a group of students with largely no prior experience in our field achieve basic proficiency and the ability to bootstrap new languages like PHP was early evidence of the power of technical CS education. This experience, coupled with my first exposure to research shortly after, cemented my decision to pursue doctoral education with the aim of becoming faculty at a research university. More recently, I have served as a TA for, respectively, Berkeley's undergraduate and graduate database courses. In the former, I ran two weekly discussion sections, prepared and graded exams, and designed a PostgreSQL-based hands-on lab. In the latter, I co-designed the syllabus, gave regular lectures (e.g., optimistic replication, streaming databases, hardware databases), mentored student research projects, and assisted in the revision of the "Red Book" (Readings in Database Systems, 4th edition) with Michael Stonebraker at MIT.

These experiences have contributed to a teaching philosophy that broadly centers around two core tenets. First, I strongly believe in exploration via interaction. Computer systems are, in most cases, real artifacts, and hands-on exploration of and production of actual code engenders both student empowerment and appropriate reverence for the complex systems that make technology today work at scale. Slowly peeling back the layers of abstraction that encapsulate the innate complexity of, say, an operating system, database management system, or five-stage MIPS pipeline, can—when done correctly—be an inspirational educational process, as can building one (or portions of one) for oneself. In the classroom, I encourage a high degree of participation; working through a problem and generating one's own hypotheses to begin teaches "CS thinking"—technologies may change quickly, but the core set of skills that allow students to reason about them does not. Second, I believe in a situated approach to learning that recognizes and speaks to students' orientations towards educational material. For example, not all students will become database systems professors, but many likely have encountered or will encounter data-intensive computation tasks in their career and/or research. I believe it is essential to couch core educational objectives in a familiar, often applied context (e.g., by use of examples, real-world datasets or projects), and highlight the *relevance* of the material. Appropriately tailoring course material, assignments, and projects towards these scenarios illustrates the beauty and power of CS concepts in a manner with which the student can empathize. In addition, I genuinely believe CS is a fun, exciting, and relevant field, and in my teaching, I seek to convey this sentiment.

As a faculty member, I would be qualified and excited to teach courses in data management, distributed systems, and cloud computing at both the graduate and undergraduate level. I would also be happy to teach courses in data science, operating systems, systems programming, and introductory computer science. I would enjoy giving seminars on large scale "Big Data" systems, distributed computing in theory and in practice, and emerging topics in data management (including new hardware architectures, storage media, and interaction models).

In addition to my formal teaching appointments, for the past year and a half, I have run the Berkeley Database Seminar (db.cs.berkeley.edu/seminar), a weekly meeting consisting of research presentations and paper discussion. Given a five-year gap in official graduate database course offerings, in mid-2013, I decided to reboot the previously defunct seminar. Since, the seminar has enjoyed regular attendance by students and postdocs largely but not exclusively advised by Berkeley database faculty. This experience has taught me the importance of cultivating a positive group culture that is both supportive and helps develop research taste among younger students. I plan to continue moderating such a regular meeting in the future. Since Fall 2014, I have also run the AMPLab Seminar (cloud.berkeley.edu), which consists of external talks I help curate. I also plan to continue organizing such a seminar in the future.

Finally, I am a strong proponent of hands-on research mentoring at both the graduate and undergraduate level.

My own positive experience with undergraduate research was key motivation behind my decision to enter graduate school. In my research mentoring as a graduate student, I have attempted to replicate my own mentors' excitement about research and patience in imparting wisdom. My primary and longest experience as a research mentor was with Aaron Davidson, an undergraduate, for approximately eighteen months. Aaron excelled as an undergraduate TA beside me in my second year of graduate school, and he subsequently joined one of my secondary research projects, on extending some of our work on probabilistic quorums (PBS). The next semester, based on Aaron's strong performance and enthusiasm, I decided to bring Aaron onto my main research project on available transaction semantics. While this was a potentially risky decision, I believe that the best way to learn about research is to *do* high-quality, novel research and participate meaningfully as a collaborator. Aaron helped build and evaluate large portions of the experimental infrastructure and directly contributed to our corresponding VLDB paper, where he earned his spot as second author. Aaron put his (unilateral) graduate admissions on hold to become the first engineer (and now tech lead) at one of my research advisors' startups. In addition, I have mentored another undergraduate, Anirudh Todi (now a distributed systems engineer at Twitter), and first-year graduate student, Aviad Rubenstein, also on extensions to our PBS work.

My mentoring philosophy is similar to my basic teaching philosophy: give students meaningful projects that engage their curiosity and work with them as valued contributors. I believe in starting with well-scoped, manageable tasks and gaining momentum and confidence. As a faculty member, I am excited to give others similar opportunities, to both empower them with the ability to ask and answer their own research questions and to learn from them along the way.