Teaching Statement

Jialin Li

As a Ph.D. student at UW, I was a teaching assistant for an operating systems class and a distributed systems class. I also tutored and mentored more than a dozen undergraduate and graduate students. One reason I am applying for academic jobs is that I enjoyed my experience teaching and mentoring students. I am excited by the opportunity to further expand my role in CS education, both as a course instructor and a research advisor.

1 Teaching Experience and Philosophy

1.1 Teaching Experience

At UW, I was a teaching assistant for a masters-level operating systems class and an undergraduate-level distributed systems class. The core of the distributed systems class was a set of programming projects that gradually build up a highly available, scalable, and fault-tolerant distributed storage system. The projects involved implementing complex distributed protocols such as primary/backup, Paxos, two-phase commit, and reconfiguration. While the instructors taught high-level concepts behind these protocols, I was responsible for teaching project design principles and implementation details in the recitation sessions. I found using example traces, illustrated by diagrams, especially effective for the students. Indeed, I have received student comments like “it was very interesting going through specific execution sequences that might occur on the homework” and “It grounded the theory from lecture”. I also presented some of my own research work to introduce students to distributed systems research. Students found those lectures particularly interesting and gave feedback such as “it was also nice to see their research” and “The lecture on NOPaxos was really cool”, and more than a dozen students in the class later expressed strong interests in working on research projects with us.

My teaching was very well received, with an overall rating of 4.2/5. Several TA related categories were ranked among the top five CS classes, including “Student confidence in TA’s knowledge”, “TA’s use of examples and illustrations”, “Explanations by the TA”, etc. Students also gave positive comments for the class, including “The TAs for this class were super knowledgeable, and helpful, and insightful”, “It is obvious they understand the material and they are very good at presenting it”, and “They have been the best TAs out of any class I have taken at UW”.

While at UW, I also volunteered to tutor CSE undergraduate students multiple times. I met with small groups of students on a weekly basis, reviewing class materials, answering questions regarding homework and projects, and preparing them for exams. Since the group size is very small (1-3 students), I strived to give tailored tutoring for each individual student. For students who struggled in the class, I reviewed core concepts of the class through detailed examples. Students often informed me they had a much better understanding of the class material after the sessions. And for the stronger students, I challenged them by asking deeper or even open-ended research questions. Before major exams, I offered extended review sessions to these students (and they often brought their friends), and many of them reported significant improvement in their class grades.
1.2 Teaching Philosophy

Substantial course projects. I believe that the most effective way of learning computer systems is through building and implementing real systems. In the operating systems class I taught, students were asked to implement a fairly complete operating system kernel for the x86 architecture; for the distributed systems class, students built a sophisticated distributed storage system that is fault tolerant and scalable. For both classes, students reported that they gained substantially better understanding of OS/distributed systems concepts through the programming assignments. At the same time, these projects present to them real challenges and trade-offs in computer systems design. Understanding these issues will prove valuable in their future careers.

I am currently writing a new teaching operating system, together with another graduate student and a faculty member. Our goal is to offer simple, robust, and extensible interfaces, as well as to be portable and take advantage of the newest hardware. Another focus of the project is to build better tool support for the OS. We observed that students often struggle with testing and debugging concurrency issues in the kernel, given the limited test cases we provide. We plan to develop automated testing tools such as model checkers, as well as tracing and visualization tools for better debuggability.

Encouraging student participation. Encouraging all students to participate in a class is a challenging task. It is quite common to see a few students dominating the discussion, which further discourages other students from participating. To address this issue, I often include a number of discussion questions in my lecture slides, and break students into smaller groups. To avoid implicit bias, I try to pick a diverse variety of students to present their solutions. In situations where different answers are presented, I encourage students to explain their thought process and compare trade-offs between different solutions. This was effective in getting more students to participate, and I plan to continue incorporating group discussions in my future courses.

Frequent student feedback. Course evaluation is a good way to get student feedback. However, these feedback are only received at the end of the quarter and therefore too late for the instructor to make any adjustments. As a solution, I always ask students for quick feedback and comments in the beginning of every class. This has proven to be effective: I incorporated some common feedback such as “more examples and execution sequences” and “upload class and discussion materials more promptly” to significantly improve the distributed systems class. I plan to introduce more feedback channels in my future classes, including surveys and anonymous online feedback forms.

2 Courses to Teach

I am excited to teach a wide range of classes in the systems area, including introduction to systems, distributed systems, operating systems, and networking, at both the undergraduate and graduate levels. I am also interested in developing new courses that are related to my research, such as systems design for the datacenter, programmable network hardware, and distributed systems and network co-design.

3 Student Mentoring and Advising

While at UW, I had the privilege to mentor a number of excellent undergraduate and graduate students. Mentoring and working closely with students has been the most exciting and
satisfying experience of graduate school.

I enjoy working closely with students. I am heavily involved in most of the student projects – including working on part of the system implementation – which is crucial especially for younger students. Being familiar with the code base also helps me to better answer students’ technical questions. In addition to regular weekly meetings, I try to make myself always available and to have frequent ad hoc discussions with students, so that I can give instant feedback and help them make progress. I also help students identify the right research questions to solve and always remind them to look at the bigger picture.

Graduate school can be a tough process and is filled with pitfalls. Besides research meetings, I often have informal conversations with my students regarding other aspects of the graduate school. Whenever possible, I share my own experience and offer advice on classes, relationship with other advisors, and work life balance. I believe having an active and meaningful advising relationship with my students is crucial in helping them succeed not just in graduate school, but also in their future career.