Yuan Tang — *Teaching Statement*

Teaching and research complement each other. The best way to understand and spread research is teaching, while the best way to prepare teaching material is by active research.

I have a strong passion for helping students who are dedicated, but not necessarily performing well in class. Though I am excited to work with high achieving students, it is also challenging and engaging to interact with students who find the material challenging. These are the students with whom I give spontaneous lectures that distill class material to its core concepts. The more I work with these students, the more I find that identifying the core of an idea and connecting the ideas together in a big picture is a key skill that young students need to learn to be successful in class and beyond. I have been teaching following three classes every year since 2012:

- **Introduction to Computer Systems (ICS) I / II (median score** $= 5.0 \in [0, 5]$**).** This is a two-semester introductory undergraduate class on computer systems. It serves as the foundation for later system classes such as operating systems, compilers, computer architecture, computer networks, concurrent programming, among others. This class includes topics ranging from assembly code, optimizing compilers, computer arithmetic, memory organization and management, networking technology and protocols, and concurrent computation. This class is recognized as a university-wide outstanding class of Fudan University.
- **Performance Engineering of Software Systems (median score** $= 4.97 \in [0, 5]$**).** The free performance from Moore's Law is coming to an end because of the power wall, heat wall and memory wall. A programmer has to be a performance engineer since performance is the currency in the world of computing, which can always be traded for new functionalities, such as scalability, security, reliability, to name a few. This hands-on project-based introduction to building scalable and portable high-performance software systems covers performance engineering techniques such as performance analyses, algorithmic techniques for high performance, instruction-level optimizations, cache and memory hierarchy optimization, and parallel programming.

Graduate level course on "Parallel Programming, Algorithms and Data Structures for real-world Applications": Given the opportunity, I would like to design a graduate-level course on parallel programming models, techniques, algorithms and data structures for real-world applications. This course not only teaches students how to program on modern computing system, including shared-memory or distributed-memory systems, but also how to fully utilize all available resources, including computing cores, hierarchical cache, and communication network, by provably and practically efficient algorithms and data structures. This course also covers programming techniques for hacker's delight.

Teaching Philosophy Students are the center of teaching, while professor just illuminates the path and give them a hand when in need.