

1)

A piece of software that would help my students learn concepts more effectively would be the **Click Modular Router** (M.I.T.). Click is a piece of software that acts as a software router, meaning that data packets arriving to a computer running Click are classified, manipulated, and properly routed via it. Click is modular, and consists of many discrete building blocks that are arranged in a graph structure. The students can modify this graph to support new protocols. Additionally they can use Click's instrumentation hooks to better understand the benefits and drawbacks of different styles of protocols.

2)

I think that some of the commercial benchmarking standards (like SPECWEB, or SPECJAPPSERVER) would be interesting for students to use. They would be able to run the benchmarks, optimize their code, and then compare the results to each other in the form of a mini competition. Unfortunately, these commercial benchmarks are extremely hard to setup and use (I had to do it last summer!). It would take a lot of work on the part of the course staff to hide that complexity from the students.

3)

Software that would facilitate collaborative activities among my students would be the **Java J2EE 1.5 platform**. Java middleware enables networked services to be deployed on replicated, scalable components. Students would build an example web service, and they would each be responsible for one of the components. This would allow them ownership of a discrete part of the service, but they would have to work together to define interfaces and appropriate semantics of what should happen to make the service possible.