MapReduce parallelism across the curriculum: an interim report

Bruce Char, William Mongan, and Jeffrey Popyack
Department of Computer Science
Drexel University
Philadelphia, PA USA 19104
Email: charbw@drexel.edu

The Computer Science (CS) curriculum is striving to include parallel and distributed computation (PDC), even though it is difficult to install it through additional or replacement whole courses. The idea of introducing it through a chain of modules across multiple courses was adopted since it facilitates incremental modification which can be more feasible in an already crowded curriculum. In this work, two modules are described. Both are scheduled for one week in multi-topic courses and contain both lecture content and student activities such as supervised lab exercises.

One module is for a first-term, first-year course required of all students enrolled in the College, which includes Computer Science, Computer Security and Technology, Information Science, Software Engineering, and Data Science majors. It designed to introduce an easily-accessible slice of PDC and activities designed to instruct on the form of MapReduce parallelism as realized by WebMapReduce. Its learning objectives include: being able to describe how parallel computation works on the machine level, and being able to come to further conclusions based on knowledge of that description. Another objective is to understand how parallel computation can achieve speedup over strictly sequential computation by breaking up a problem into pieces which can be done concurrently on different computation units.

The second module isfor a second-year course on System Programming required of all CS majors. It builds upon prior use of MapReduce but asks students to consider performance (speedup) as well as correctness operating on multiple processors. The module adds to the pre-existing course content on multithreaded “small concurrency” on multicore, to an understanding of a massive scalability on a distributed system. It discusses performance issues in MapReduce, discussing coordination and network latency bottlenecks, locality “in the large,” and the implementation of MapReduce algorithms on Hadoop. Students complete a non-trivial parallelizable algorithm on a sequential environment, then parallelize and study the speedup effects and attempt to determine performance factors at work.

The interim evaluation indicates that the modules for both courses are well-received by students and are regarded by them as enhancing their understanding of PDC. The in-class and homework activities appear to take an appropriate amount of time for most students. Sometimes (but not always) significant shifts in beliefs before/after the module that the individual understood parallel and distributed computation were observed. However, usually there was little alteration of belief pre-/post-module about the extent to which PDC would be important in the individual’s academic or professional career.

While almost all students successfully completed the module activities, which suggests that they understood the processes involved in WebMapReduce, it was difficult to observe transferability of module learning in all cases indicates that this approach needs further refinement in order to be successful when realized as a chain of lessons in multiple courses. For example, no significant change was observed pre-/post- to the question ”Every Hadoop installation uses multiple computers”, even though this was covered in the module. No significant difference was found in students’ ability to recognize whether a problem was a good candidate for speed up for MapReduce unless it was very similar to those done in the module activities.

Future work includes establishing the extent of student knowledge acquisition that come from largely successful completion of module activities that we currently see. It also will involve focusing on what is needed to ensure student ability to successfully transfer PDC knowledge to other problems or contexts within the limited time available in a one-week module. A third module will be developed for a senior-level course on Artificial Intelligence, where students will be asked to transfer their PDC learning to a significant domain-specific problem.