Improving the Efficiency of CHA through Parallelization

Miguel Velez and Jason Sawin
Computer & Information Sciences, University of St. Thomas

Interprocedural analysis is a software engineering technique that uses calling relationships among procedures to analyze an entire program. Researchers and software engineers rely on such analyses to help them build efficient and safe programs. The Class Hierarchy Analysis (CHA) algorithm is one of several popular sequential algorithms that analyzes the calling relationships between the methods in a program. In this project, we parallelized CHA, taking advantage of modern multi-core architectures, in order to decrease the execution time of the algorithm to output a list of reachable methods from a program's entry point. CHA usually starts with the entry method of a program; the main method in a Java program, and it discovers the methods being called within that method. For each method discovered, it queues all of the methods that have the same signature in all of the subclasses of the method class being called. The algorithm ends when all of the methods have been analyzed. By parallelizing CHA, we were able to decrease the execution time of the algorithm by dividing the work done limited to the number of cores available in the computer used to run the analysis. The results of our empirical study showed an improvement of ~8% to ~34% compared to the sequential algorithm, depending on the program being analyzed and the number of cores used for the analysis. As expected, the more cores used, the faster it took to run the analysis. However, there was a threshold of ~16 cores used that did not provide further improvement. This is expected because the increase in the amount of thread management and context switching overcomes the benefit of parallelizing the algorithm.