JavaLife

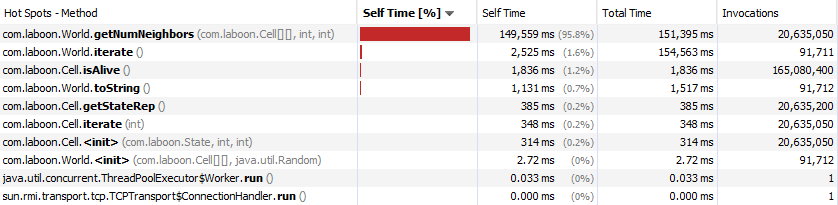
Austin Choi

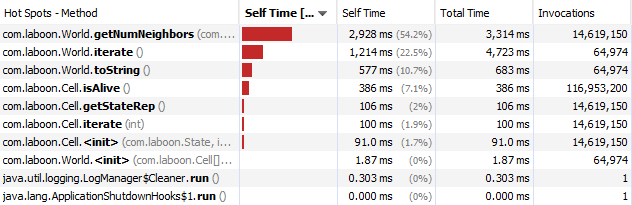
CS 1632 – DELIVERABLE 5: Performance Testing Conway’s Game of Life

<https://github.com/M94/JavaLife>

I used the Java VisualVM to profile the JavaLife program. I ran it for 100,000 iterations under the CPU profiler tool. Under the CPU profile results, the getNumNeighbors() method in the World class was clearly eating much more CPU time than any other of the methods defined as hot spots; it accounted for 95% of total self-time. The profiler identified iterate() as the second most resource intensive method, and it only accounted for 1.6% of self-time. Thus, clearly something needed to be done about getNumNeighbors().

When I examined the method, I noticed that a few if-statements were stuck in a for-loop that iterated 10,000 times. This for-loop was clearly the culprit behind the method’s long runtime. I also realized that the results of the if-statements would be same whether the for-loop ran 1 time or 10,000. Since removing the for-loop would not modify any behavior, I refactored it and took it out. The method’s runtime improved significantly on the program’s second run in the profiler. Although the getNumNeighbors() method still took up about 50% of total self-time, its actual millisecond count was much more sane now. And the program itself finished in noticeably much less time.

Java VisualVM: Before Refactor

Java VisualVM: After Refactor