JavaLife

Matthew Jeffreys

CS 1632 - DELIVERABLE 5: Performance Testing Conway's Game of Life

For deliverable 5, I first decided on a set of homogeneous command line arguments for the program JavaLife. Differences in performance due to altered inputs would be unacceptable for an accurate reading, so at each iteration I utilized size:10, seed:10, Saturation:10, and an iteration limit of 100000.

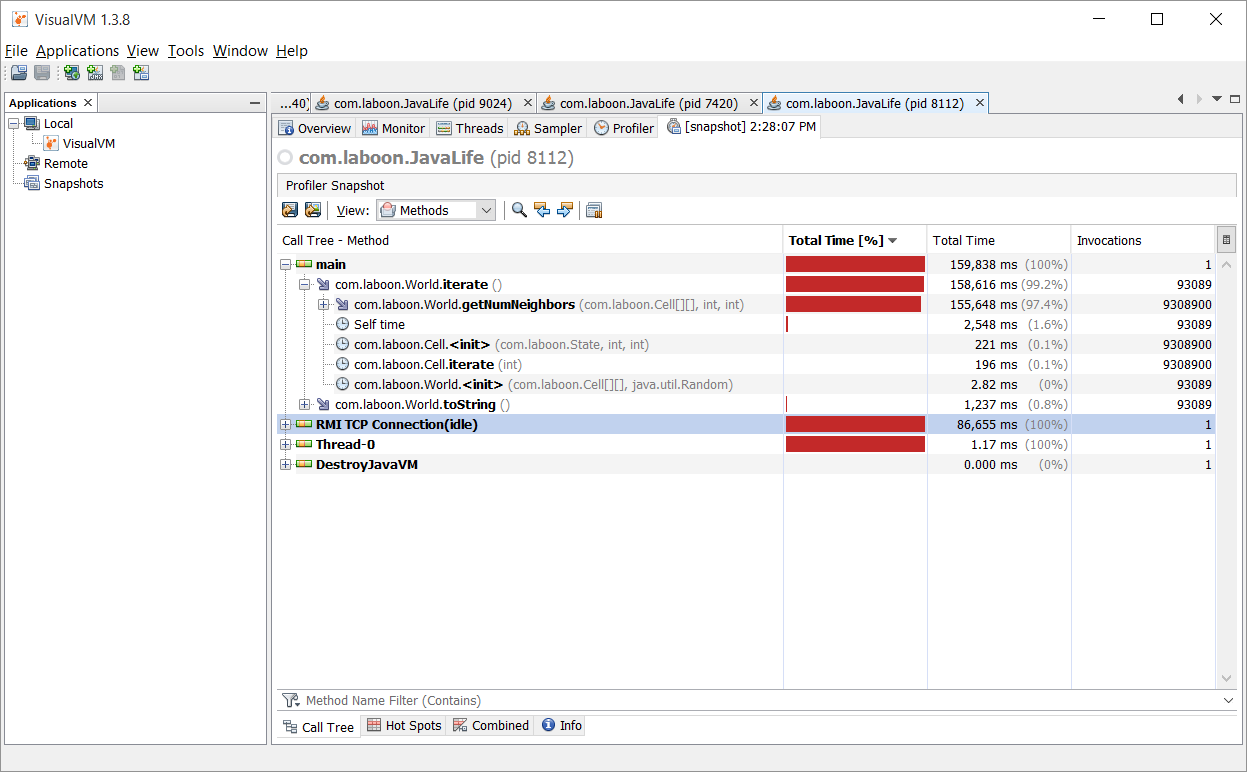
For the initial test I launched JavaLife with the standard arguments and began profiling the cpu usage per method, with JVisualVM. The results showed a obvious spike in cpu usage while the method World.getNumNeighbors() was executing. I therefore, inspected the class at the code level and determined that there was an unnecessary for-loop enclosing a section of code that resulted in an astounding 10000 unnecessary loops. I simply commented out the for-loop declaration as well as the trailing bracket. I recompiled JavaLife and repeated the test using the standard inputs, resulting in a much better World.getNumberNeighbors() execution-time ratio.

In a real life situation, this type of oversight might simply be the result of the poor re-factoring of a method or a rushed development cycle. Even a veteran software developer can make mistakes that result in unfortunate performance drops, so resource profiling is an important part of software testing.

Repository:

https://gitub.com/mdj20/JavaLife/

Initial profile:



Refactored profile: