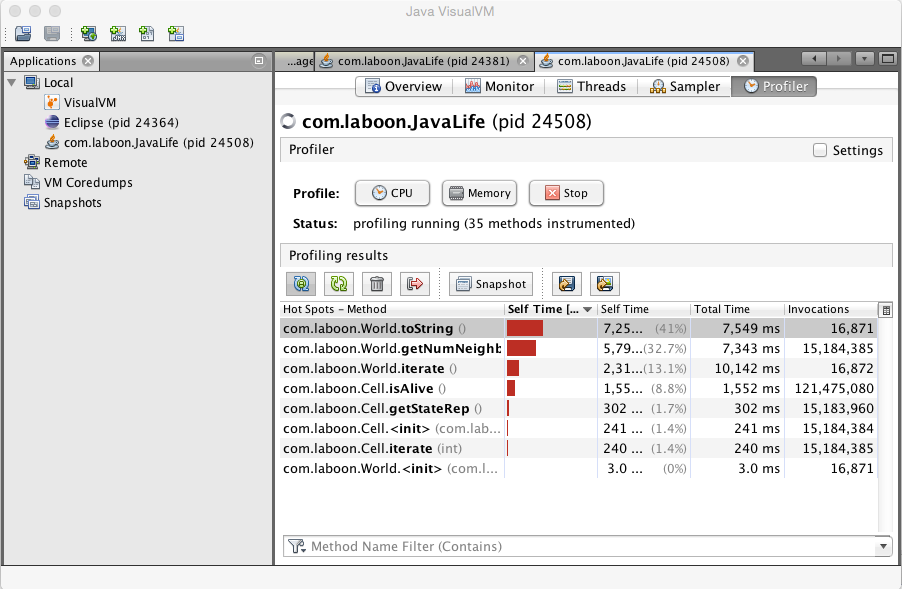
CS 1699 - DELIVERABLE 4: Performance Testing Conway's Game of Life

JavaLife - Barry Arons

In order to find the method that needed to be refactored I ran the game of life and profiled it using VisualVM. On the first run of VissualVM I thought that the getNumNeighbors function was the one that needed to be refactored. After running more tests with different numbers it became clear that the toString function was taking a lot more CPU time then it should. These results lead me to look at the method and see what it was doing. The first thing I noticed was that the function used a series of string concatenations instead of using a string builder. Before I changed anything I added unit tests to the code so that I would know if I broke anything. I decided to test a small grid by using all 4s for the values of the world. I then decided to move into edge cases. The edge cases I picked are a completely alive grid and a completely empty grid. The last edge case I picked is a zero by zero grid. I finished up the test suite by adding a few more normal test to make sure they all worked. Since all my tests passed I started to refactor the method making sure that my tests pass at all the steps. The way that I refactored the method was to change every string concatenation to part of a string builder. Due to the nature of the refactoring I needed to refactor the entire method before I could run my tests. After refactoring the function I ran the unit tests and they all passed. The passing tests lead me to believe that the method was refactored correctly. After the tests passed I ran VisualVM again and saw that the CPU time used by the toString method had been greatly reduced. I was amazed by how much faster the program ran after the method was refactored. I have always heard that concatenating things is much slower then using a string builder but this allowed me to see it first hand. After the project was completed I went back and used VisualVM to look at some of my old code and see how I could speed it up.

Before



After

