Sam Kopansky

Project: JavaLife

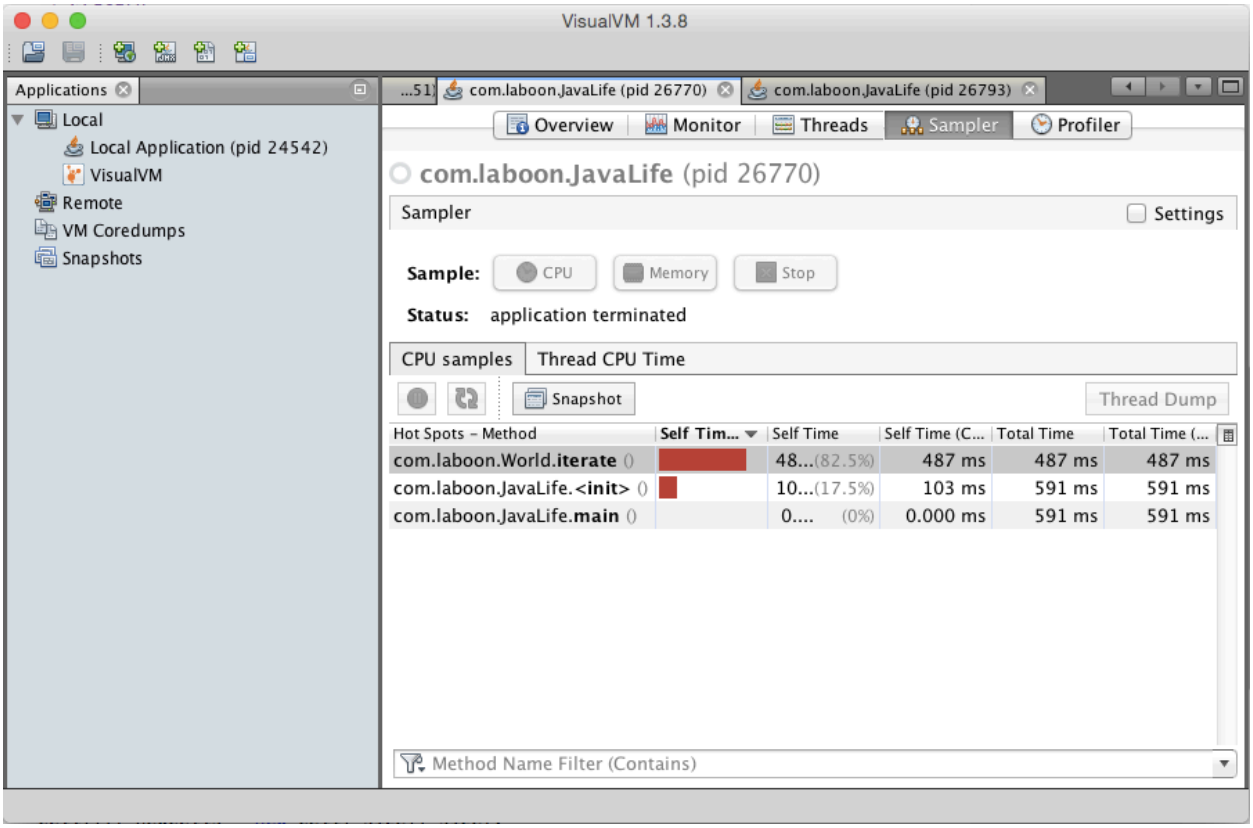
CS 1632 - DELIVERABLE 5: Performance Testing Conway's Game of Life

This deliverable provided me a lot of relief. The sole reason behind this sudden weight lifting off my shoulders is that I could finally stop using eclipse. I've had trouble with it every time I have tried to use it so I'm very glad that VisualVM had its own environment. VisualVm was actually really simple to set up. I didn't have to do anything other than install it. After that I started the assignment.

I first ran the original code and looked at the "Sampler" page while the program was still going and saw that a method in the "World" class with the name "iterate()" was taking up the most CPU time so I opened up the code and took a look at it. Everything in that method made sense to me and I didn't immediately see any problems with it so I started looking at the methods that it called. The first one was called "getNumNeighbors()" so I went up to there and sure enough, there was a for loop in that method that aimlessly iterated ten thousand times for every iteration of the previous "iterate()" method. I simply commented out the for loop and received a massive performance boost. Take a look at the screenshots on the next page to see the VisualVM output, performance boost and the changed method.

Before refactoring:

VisualVM



Method

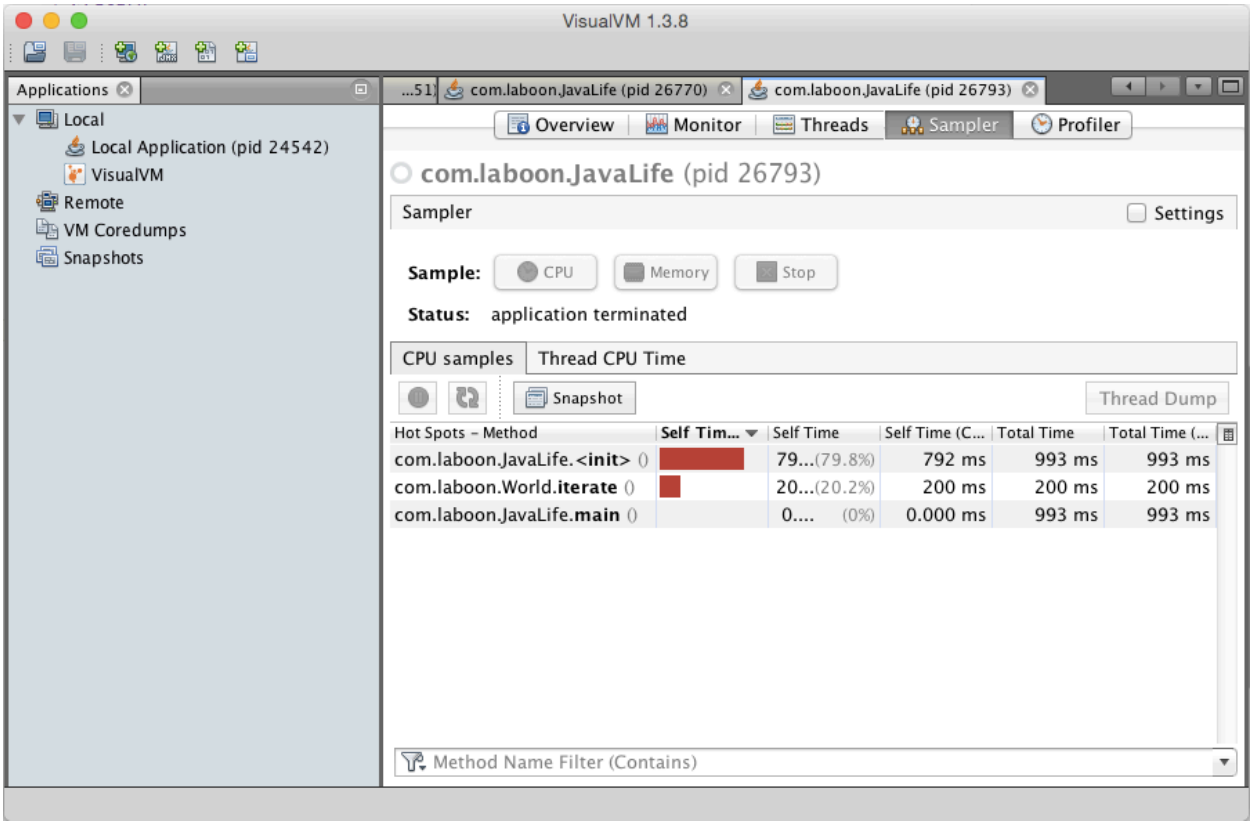
```
private int getNumNeighbors(Cell[][] world, int x, int y) {
        . . .
        for (int j = 0; j < 10000; j++) {
           if (leftX == -1) { leftX = size - 1; }
           if (rightX == -1) { rightX = size - 1; }
           if (upY == -1) { upY = size - 1; }
           if (downY == -1) { downY = size - 1; }
        }
        . . .
}
```

Time

```
real    2m44.448s
user    2m40.100s
sys     0m1.763s
```

After Refactoring:

VisualVM



Method

```
private int getNumNeighbors(Cell[][] world, int x, int y) {
        . . .
        //for (int j = 0; j < 10000; j++) {
           if (leftX == -1) { leftX = size - 1; }
           if (rightX == -1) { rightX = size - 1; }
           if (upY == -1) { upY = size - 1; }
           if (downY == -1) { downY = size - 1; }
        //}
        . . .
}
```

Time

```
real    0m5.918s
user    0m2.760s
sys     0m0.784s
```