CS 1632 – DELIVERABLE 4:

Performance Testing Using VisualVM

James Hahn & Ahmed Abdelsalam

Summary

* For the summary, describe how you profiled the application and determined the methods to refactor, and why you changed what you did. The summary should not be more than a few paragraphs - definitely less than a page.

For this project, VisualVM was ran immediately on different sized opinions, such as 100 or 200 so that hotspots would be much easier to see; if 5 monkeys were tested, it wouldn’t give as much of a well-rounded overview of the hotspots and CPU usage.

The methods that were chosen for refactoring were getFirstMonkey() – this was the largest CPU usage initially, then stringifyOutput(), generateId(), monkeyWithBanana(), and finally incrementRounds().

For getFirstMonkey(), it was starting from the end of the monkey list and traversing towards the front, checking the monkey number along the way to see if it was 1; and if it was, other counter variables had to be equal to 0, which required hundreds of thousands of extra operations that were unnecessary. A solution to this was to return the Monkey at index 1 (which always 1 as its monkeyNum because it’s static, starts at 0, and increments by 1 for every Monkey) in constant time.

As for stringifyOutput(), there was a loop that concatenated 50000 @ symbols to an initial return String, then it essentially wiped all of them and began creating the correct output String. The fix to this was to remove that useless for loop.

Code: <https://github.com/kingsman142/MonkeySim>

Screenshots

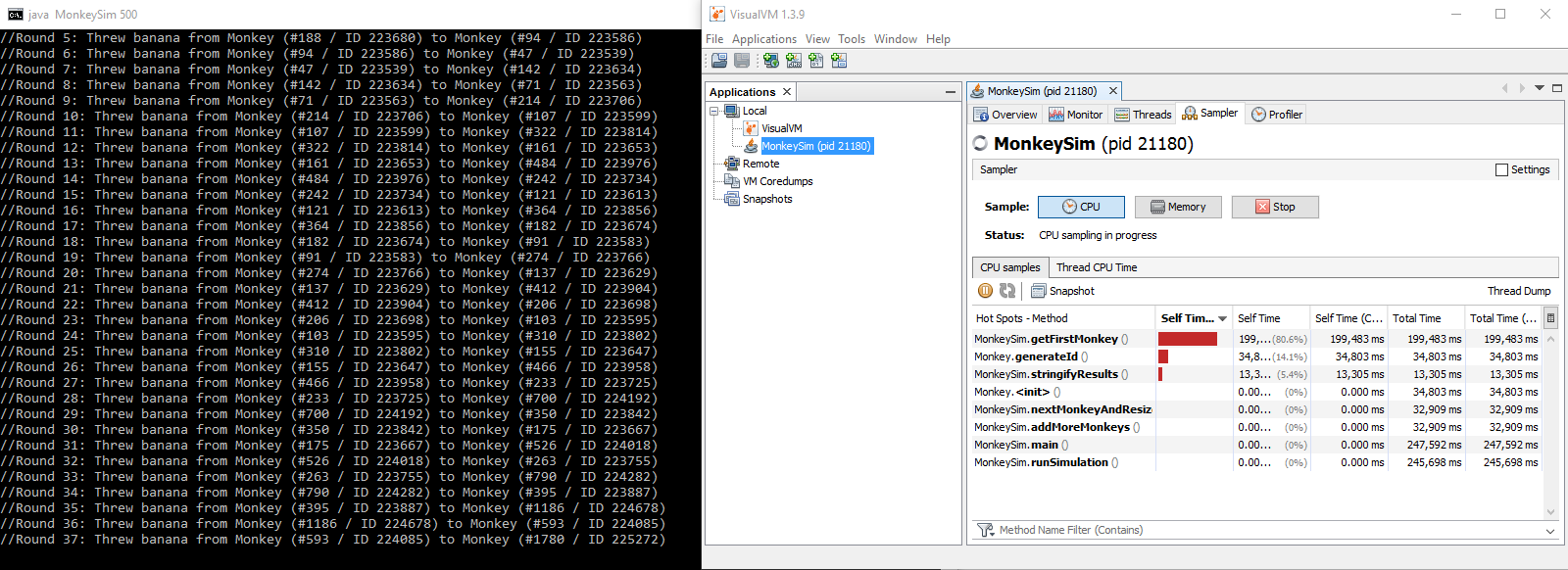
* After this, include screenshots of VisualVM (or another profiler, if you use that) both before and after the refactor. These screenshots should include the relevant sections that let you see what method to refactor.

**getFirstMonkey() :**

**BEFORE (gave up after 200 seconds for 37 rounds because I wasn’t going**

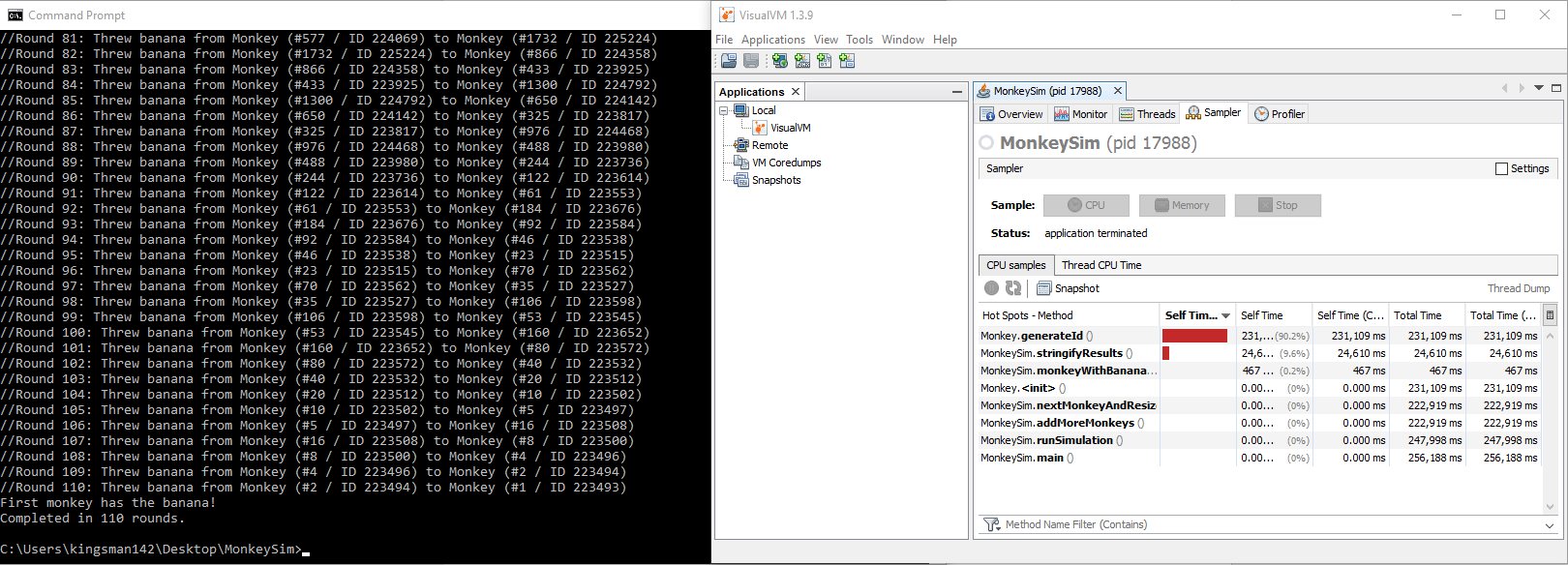
**to waste my entire day waiting for this thing):**

average time: 11.671s

**** longest time: 11.851s

**AFTER (simulated with 500 monkeys, hotspot is no longer existent):**

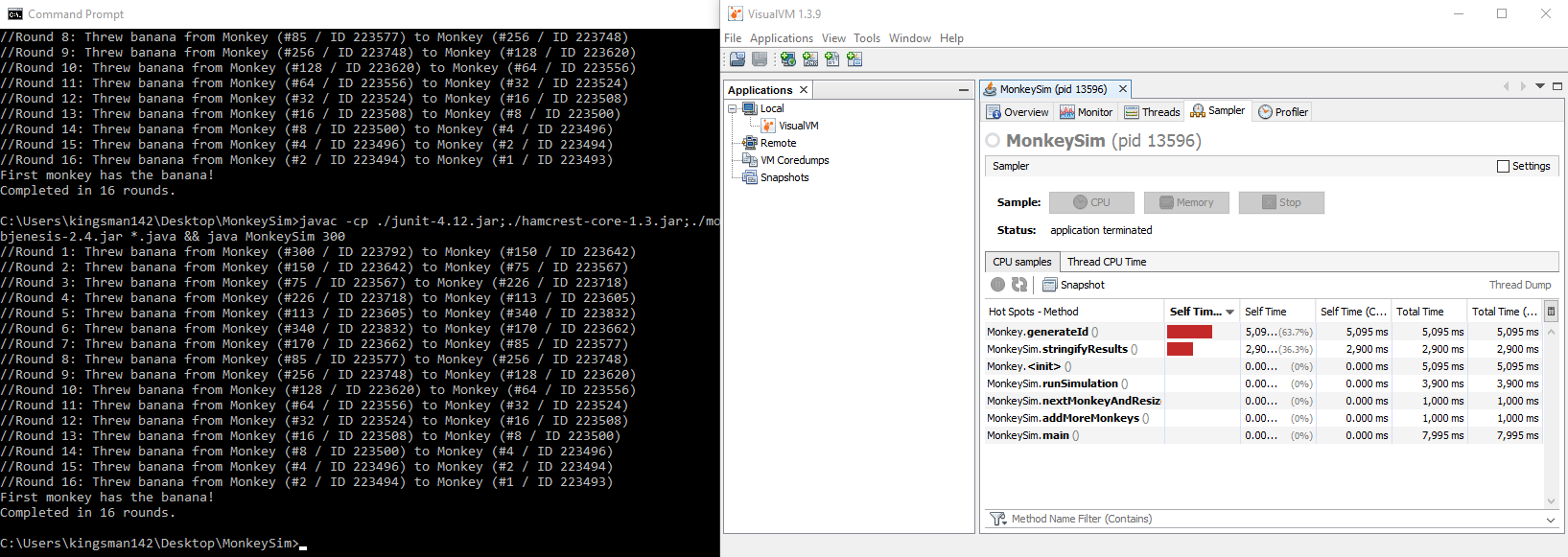
average time: 10.324s

**** longest time: 10.465s

**stringifyResults():**

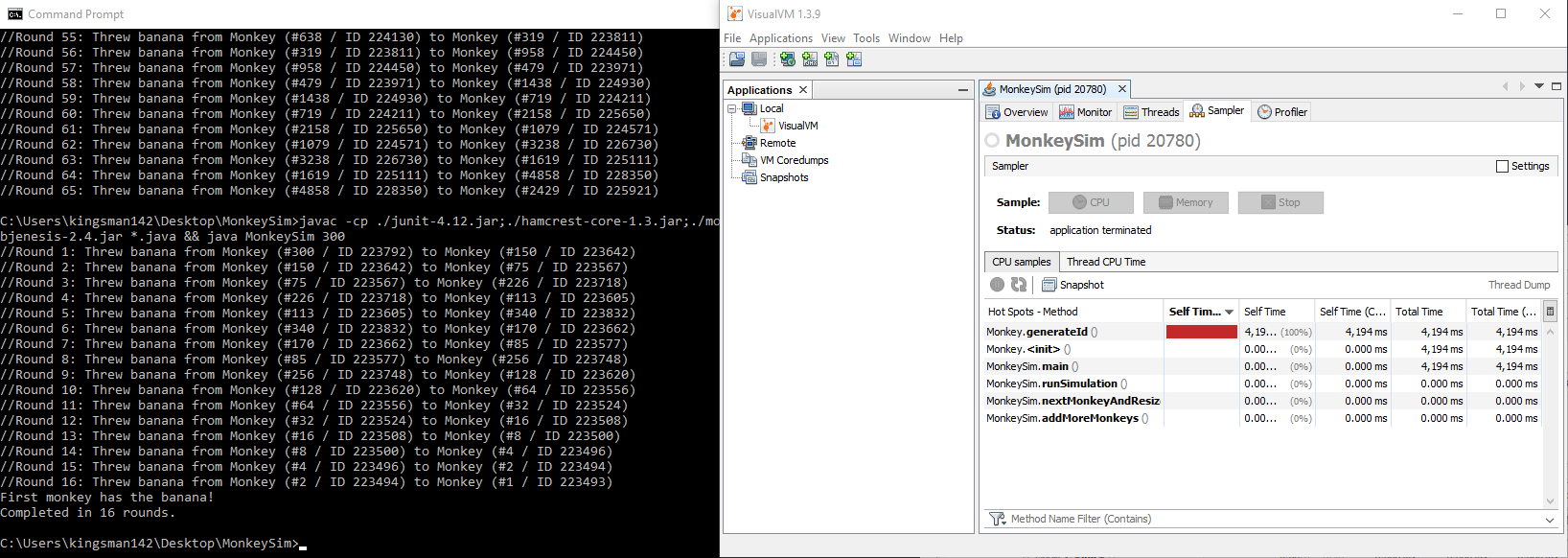
**BEFORE (300 monkeys):**

average time: 10.581s

longest time:**** 10.767s

**AFTER (300 monkeys, hotspot no longer existent):**

average time: 4.30s

 longest time: 4.502s