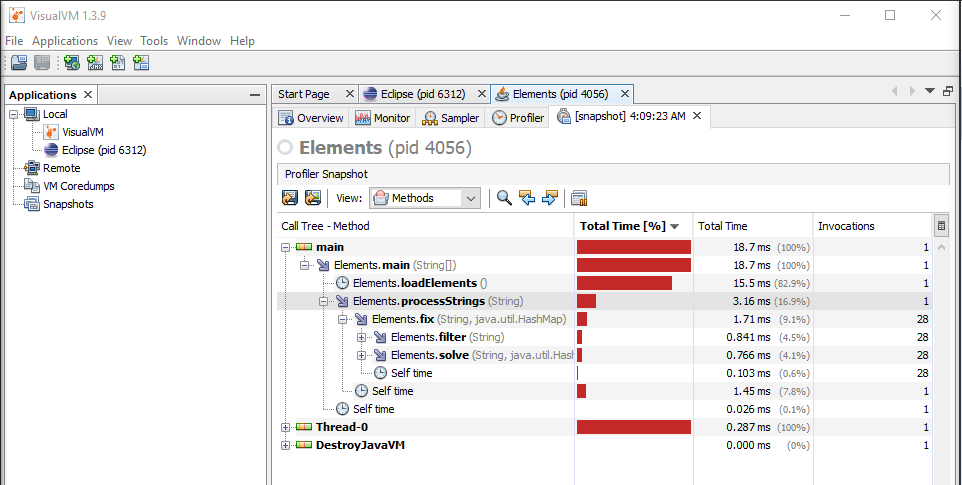
CS 1632 – DELIVERABLE 4: Performance Testing Using VisualVM

By Nigel Bernard & Josh Koshy

GitHub: https://github.com/nib58/Deliverable4

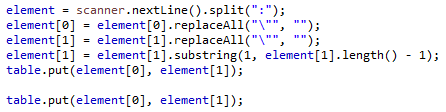
Summary:

Our program, Elements.java, is the main program used for this deliverable. It has two important methods that comprise the workload of our program: loadElements() and processStrings(String).

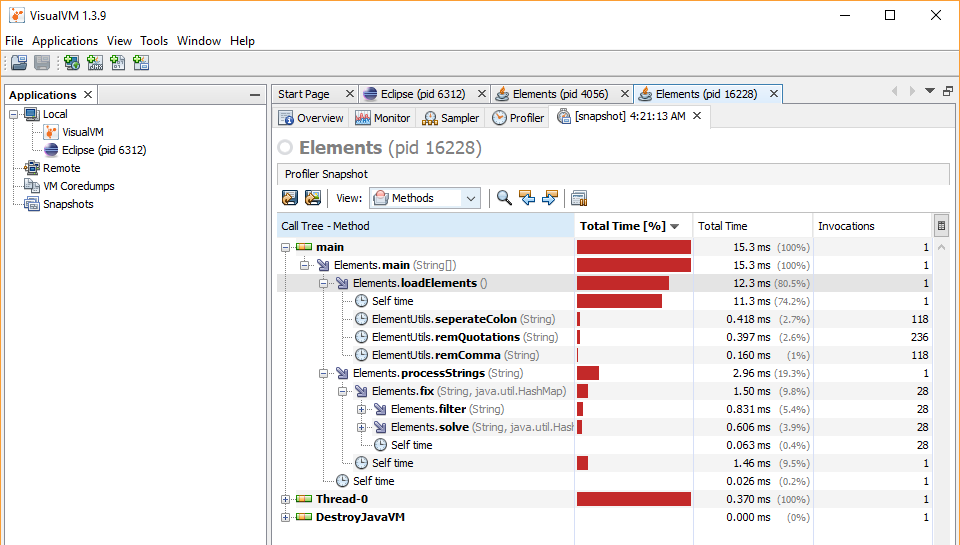
The first time we profiled Elements.java, we were guessing that our processStrings(String) method would be the main issue that we would need to prioritize. In terms of complexity, processStrings(String) has a lot more involved in its work. However, after profiling, we realized that this was not the case.

processStrings(String) was comprising of 16.9% of the workload while loadElements() was taking a whopping 15.5ms to finish, taking 82.9% of the workload to finish!

We changed our priority to fixing loadElements() and refactoring it so it wasn’t so inefficient.



Our while-loop was using this chunk of code, which utilized a lot of string methods in order to filter out unwanted characters. We realized this was the main culprit and we worked on solving it. Our final code now uses custom methods tailored to providing efficient time spent.



This was the final result, where you can see our loadElements() method taking an average of 3 seconds less than it would previously.

We executed the Elements class with the time command to evaluate the “real” execution time of the program. We recorded times of 116ms, 114ms and 120ms. The median execution time was 116ms and the mean execution time was 117ms. (Note: these times were taken on a different computer than the one used for the screenshots above.)