CS1632 - DELIVERABLE 4 : Performance Testing

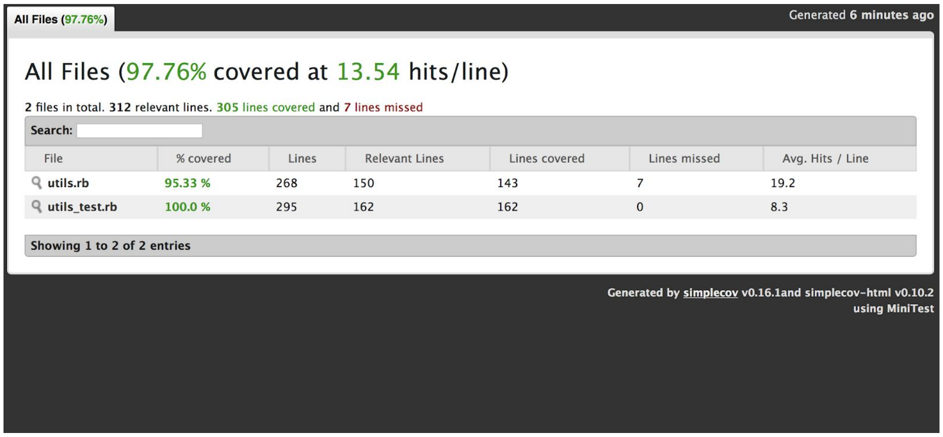
https://github.com/Trappie/D4

Di Wu: Trappie

Sophia Lin: sphjlin

- What was the most challenging about this deliverable?
    - When I was trying to enhance the performance with multiple threading, I found that CRuby doesn't support multithreading over multi-cores, it has a global lock to prevent this from happening. So the solution I chose was using multi-processing.

- What kind of edge cases and failure modes did you consider?
    - Edge cases:
        - test line 0 for **test_verify_line_transactions_valid** to see if prev_line_number, prev_hash and prev_time is valid
        - While testing **greater?(time1, time2)** method, I wrote "<", "==", ">", first_invalid, second_invalid, both_invalid. In both_invalid tests, I tested a two integer case and a two non-float string case. Both cases failed because they do not have a decimal point, but the two strings also test if there is a .to_i step.
        - In the **verify_transactions** method, test if the smallest non-negative "000000>000000(0)" is valid.
    - Failure modes:
    - While failure cases, I found out the original program hard to test with stub, so I took a bumb but accurate way: Test the methods stubbed first, after making sure these methods are correct, test the methods that needs stubs.
        - When testing invalid cases in valid_hash? method, valid_blocks? method andverify method, it is a tricky thing to also test **show_error_message** method, because the three valid method includes the error message in the end, and I need to test if there is a "\nBLOCKCHAIN INVALID\n" in the end.
    - Test Covered percentage (simplecov)

- 

  Generated **6 minutes ago**

  All Files (97.76% covered at 13.54 hits/line)

  2 files in total. **312** relevant lines. **305** lines covered and **7** lines missed

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
utils.rb	**95.33 %**	268	150	143	7	19.2
utils_test.rb	**100.0 %**	295	162	162	0	8.3

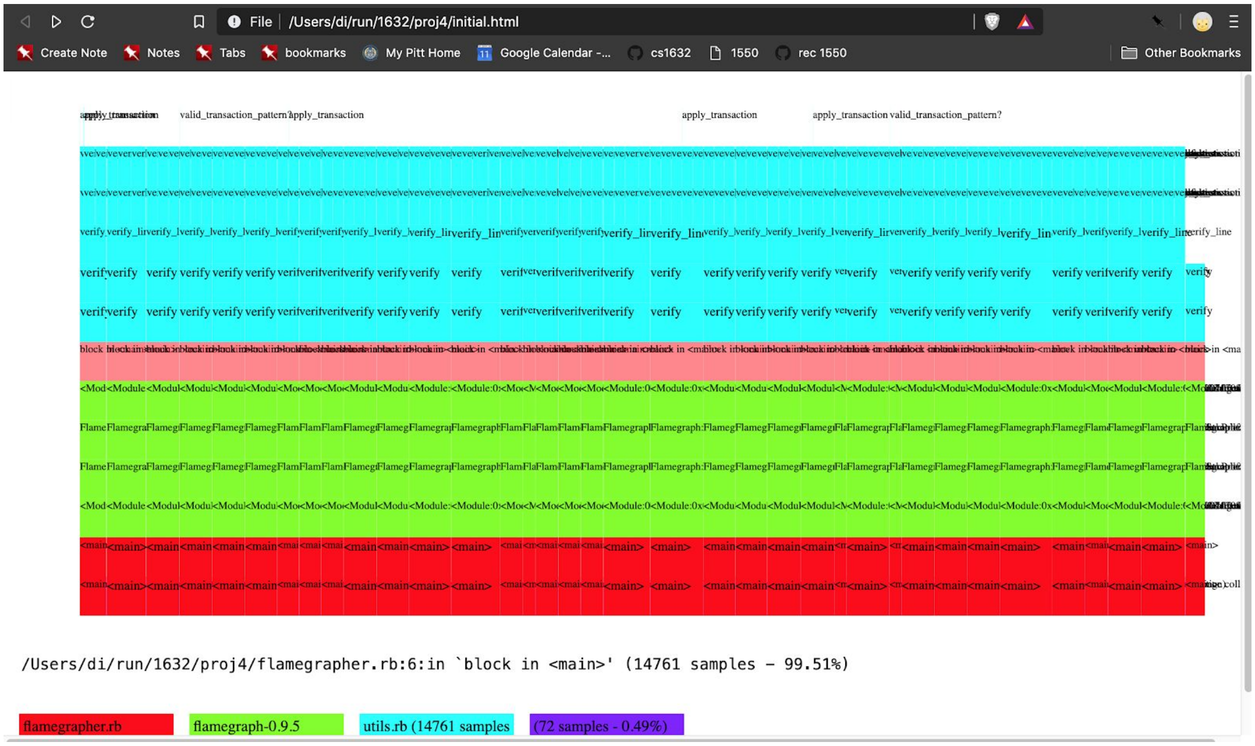
  Showing 1 to 2 of 2 entries

  Generated by simplecov v0.16.1and simplecov-html v0.10.2 using MiniTest

- Using the flame graph, what methods were taking up the most CPU time?
  - In INITIAL version the method **verify_transactions** was taking up 91.57% CPU time.
  - I found out that the problem was due to the data structure I used for accounts. I stored information about accounts in an 1_000_000 length array instead of a Hash, because I believed that it was faster. To enhance it, I used a Set data structure to hold the address of negative accounts, and check whether the set is empty after processing each block. This is much faster than loop through all the possible accounts in the 1_000_000 length account array for negative ones every block. This enhancement reduced the real time of the program against long.txt from 10 minutes to about 1 minute.
  - After this enhancement, the new hot spot became hash_str, which verify the hash value of a block. I noticed that the characters used in the block are relatively limited, so I pre-calculated all of them (“0123456789abcdef():>.SYTEM”) and put them into an array (actually, I used a Hash at first, and later I found out that array is about 100ms faster than Hash when dealing with long.txt), so that the **hash_str** was now just add all hashed values together and modulo it with 2 ** 16. I also tried to use bitwise operation on this modulo processing, but it’s not faster. This enhancement reduced the real time of the program against long.txt from 1 minute to about 550 ms.
  - After this, I made some small enhancements to reduced the real runtime of the program from 550ms to 370ms, then I thought I couldn’t make it faster without making use of multi cores.

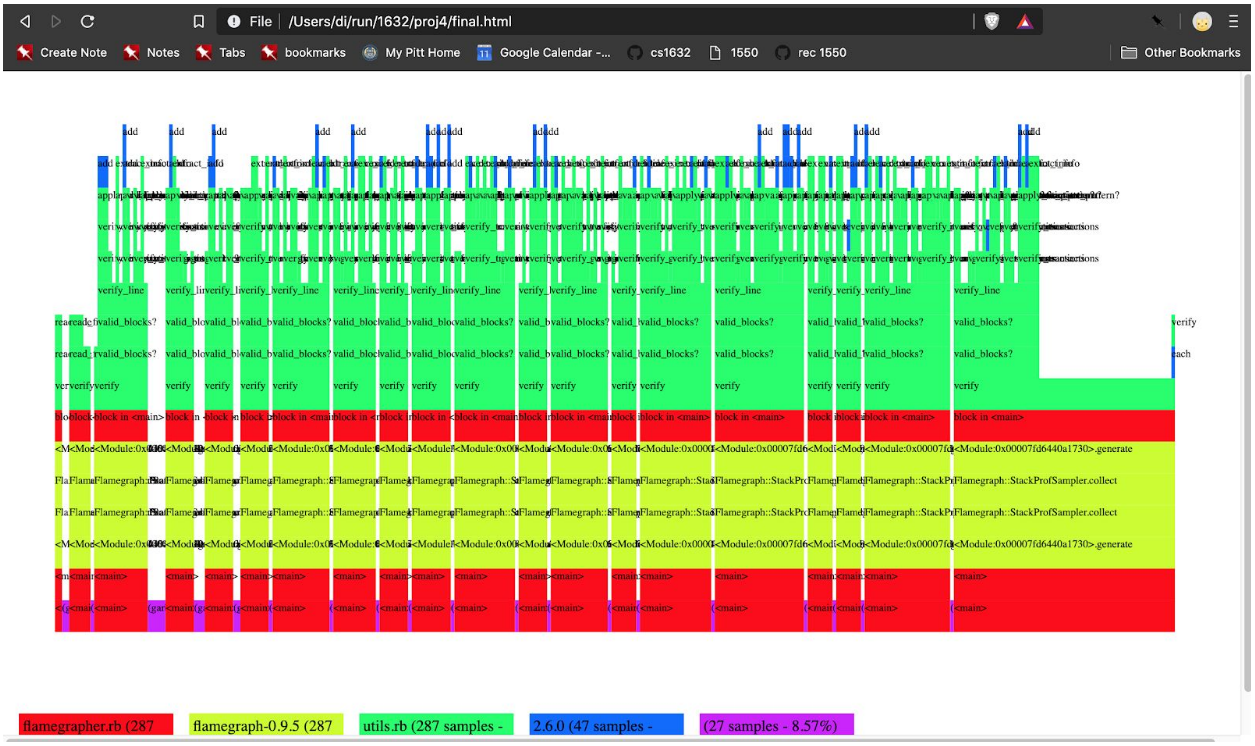
- I put the verifying process into another process, which finally reduced the real runtime from 370ms to 270ms, and now the cpu utility is about 150%.
- According to my design, it can be a little faster if using more processes on a quad-core, but I don’t think it worth the effort.

Flame graph of INITIAL version



```
/Users/di/run/1632/proj4/flamegrapher.rb:6:in `block in <main>' (14761 samples - 99.51%)
```

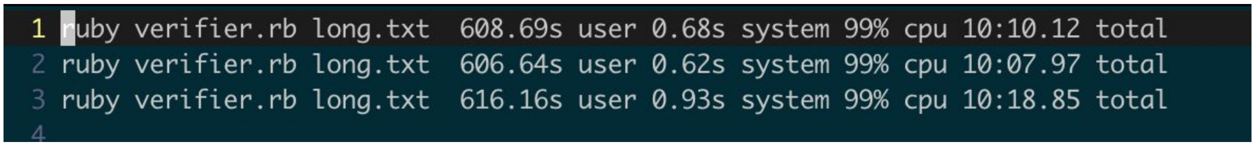
Flame graph of FINAL version



Runtime for INITIAL version

Mean: 10:12.313

Median: 10:10.12

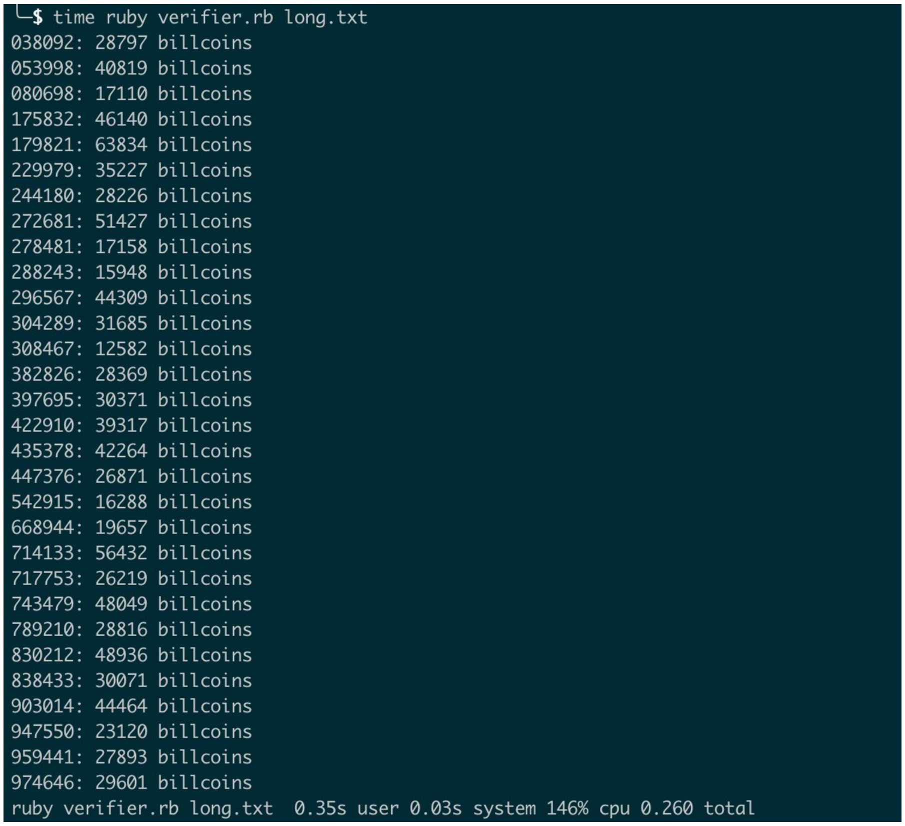


```
ruby verifier.rb long.txt  608.69s user 0.68s system 99% cpu 10:10.12 total
ruby verifier.rb long.txt  606.64s user 0.62s system 99% cpu 10:07.97 total
ruby verifier.rb long.txt  616.16s user 0.93s system 99% cpu 10:18.85 total
```

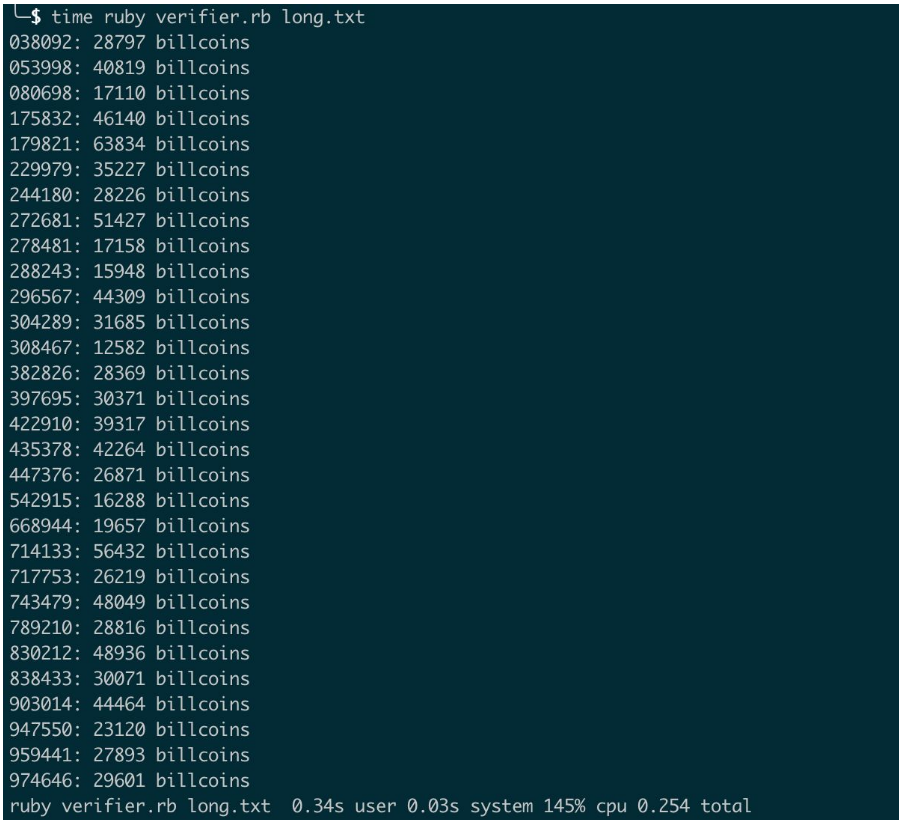
Runtime for Final version

Mean: 257.33 ms

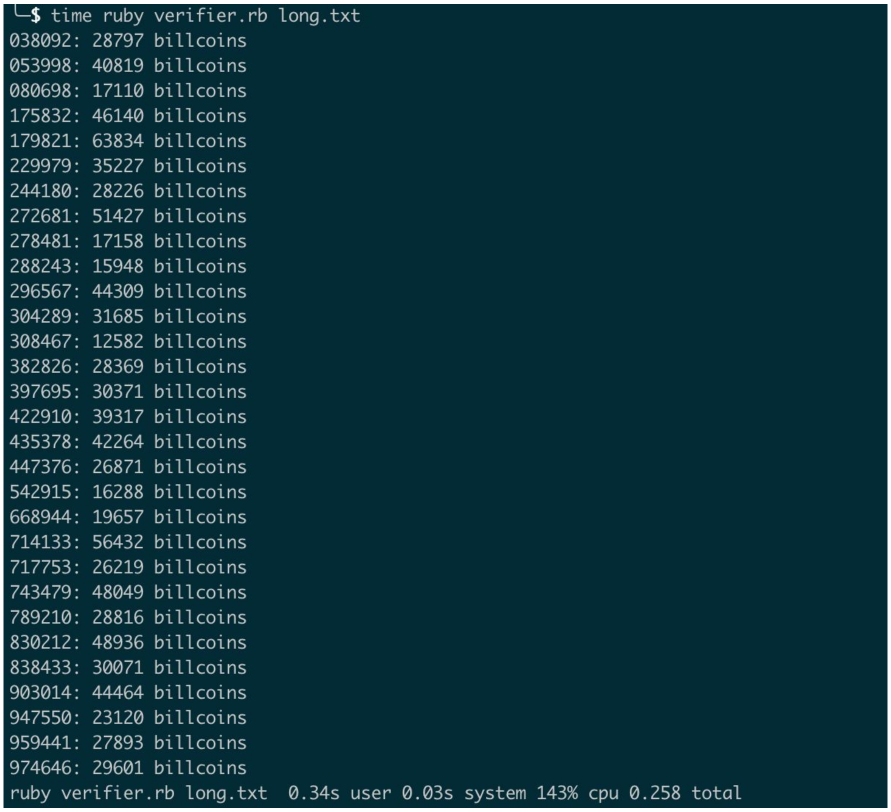
Median: 258 ms



```
 └─$ time ruby verifier.rb long.txt
038092: 28797 billcoins
053998: 40819 billcoins
080698: 17110 billcoins
175832: 46140 billcoins
179821: 63834 billcoins
229979: 35227 billcoins
244180: 28226 billcoins
272681: 51427 billcoins
278481: 17158 billcoins
288243: 15948 billcoins
296567: 44309 billcoins
304289: 31685 billcoins
308467: 12582 billcoins
382826: 28369 billcoins
397695: 30371 billcoins
422910: 39317 billcoins
435378: 42264 billcoins
447376: 26871 billcoins
542915: 16288 billcoins
668944: 19657 billcoins
714133: 56432 billcoins
717753: 26219 billcoins
743479: 48049 billcoins
789210: 28816 billcoins
830212: 48936 billcoins
838433: 30071 billcoins
903014: 44464 billcoins
947550: 23120 billcoins
959441: 27893 billcoins
974646: 29601 billcoins
ruby verifier.rb long.txt  0.35s user 0.03s system 146% cpu 0.260 total
```



```
└$ time ruby verifier.rb long.txt
038092: 28797 billcoins
053998: 40819 billcoins
080698: 17110 billcoins
175832: 46140 billcoins
179821: 63834 billcoins
229979: 35227 billcoins
244180: 28226 billcoins
272681: 51427 billcoins
278481: 17158 billcoins
288243: 15948 billcoins
296567: 44309 billcoins
304289: 31685 billcoins
308467: 12582 billcoins
382826: 28369 billcoins
397695: 30371 billcoins
422910: 39317 billcoins
435378: 42264 billcoins
447376: 26871 billcoins
542915: 16288 billcoins
668944: 19657 billcoins
714133: 56432 billcoins
717753: 26219 billcoins
743479: 48049 billcoins
789210: 28816 billcoins
830212: 48936 billcoins
838433: 30071 billcoins
903014: 44464 billcoins
947550: 23120 billcoins
959441: 27893 billcoins
974646: 29601 billcoins
ruby verifier.rb long.txt  0.34s user 0.03s system 145% cpu 0.254 total
```



```
└$ time ruby verifier.rb long.txt
038092: 28797 billcoins
053998: 40819 billcoins
080698: 17110 billcoins
175832: 46140 billcoins
179821: 63834 billcoins
229979: 35227 billcoins
244180: 28226 billcoins
272681: 51427 billcoins
278481: 17158 billcoins
288243: 15948 billcoins
296567: 44309 billcoins
304289: 31685 billcoins
308467: 12582 billcoins
382826: 28369 billcoins
397695: 30371 billcoins
422910: 39317 billcoins
435378: 42264 billcoins
447376: 26871 billcoins
542915: 16288 billcoins
668944: 19657 billcoins
714133: 56432 billcoins
717753: 26219 billcoins
743479: 48049 billcoins
789210: 28816 billcoins
830212: 48936 billcoins
838433: 30071 billcoins
903014: 44464 billcoins
947550: 23120 billcoins
959441: 27893 billcoins
974646: 29601 billcoins
ruby verifier.rb long.txt  0.34s user 0.03s system 143% cpu 0.258 total
```