Haoran Liang
hl74
COMP554

Analyze Cache System on Different Computers

Team members: Qichao Sun, Haorang Liang, Hao Ding

In this project, we extend the last lab homework to an analysis of different computers and their cache system. Our report contains the following parts:

- Configurations
- Analysis of Shuffle on
- Analysis of Shuffle off
- Problems and Thoughts

Configurations:

- Computer1:
  MacBook Air 2015(macOS)
  Intel core i5 @ 1.4 GHz
  L1i cache size: 32KB
  L1d cache size: 32KB
  L2 cache size: 256KB
  L3 cache size: 3MB

- Computer2:
  ASUS ROG GR8 II(Win 10)
  Intel core i5-7400 @ 3.0 GHz
  L1 cache size is: 256KB
  L2 cache size is: 1MB
  L3 cache size is: 6MB

- Computer3:
  MacBook Pro 2018 (macOS Majove):
  Intel Core i5 CPU @ 2.60GHz
  L1 cache size: 32768 Bytes = 32KB
  L2 cache size: 262144 Bytes = 256KB
  L3 cache size: 6291456 Bytes = 6MB.

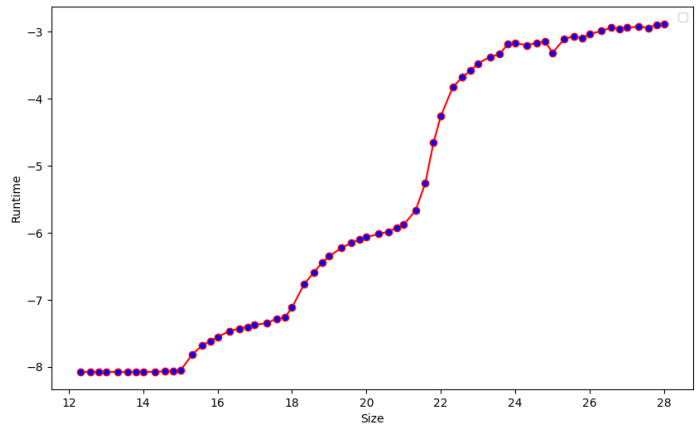
- Computer4:
  Lenovo Laptop (Win 10):
  Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz
  L1 cache size: 32768 Bytes = 256KB
  L2 cache size: 262144 Bytes = 1MB
  L3 cache size: 6291456 Bytes = 6MB.

- Computer5:
  Lenovo Y720 (Win 10):
  Intel core i5 8300HQ @ 2.5 Ghz

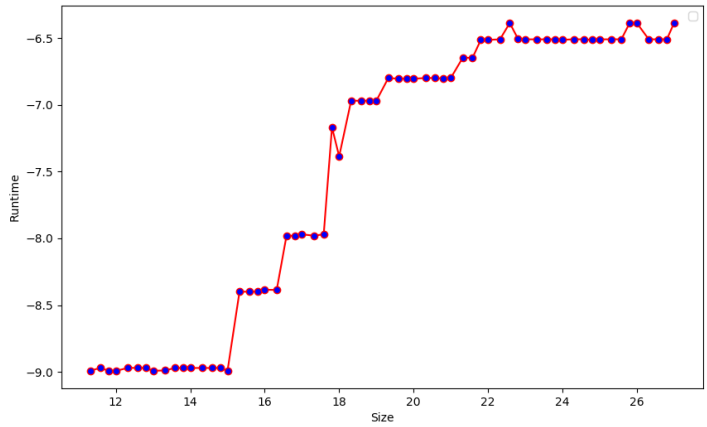
L1 cache size: 256KB
L2 cache size: 1MB
L3 cache size: 6MB

- **Analysis of Shuffle on**

- Experiment 1:
  Computer1:

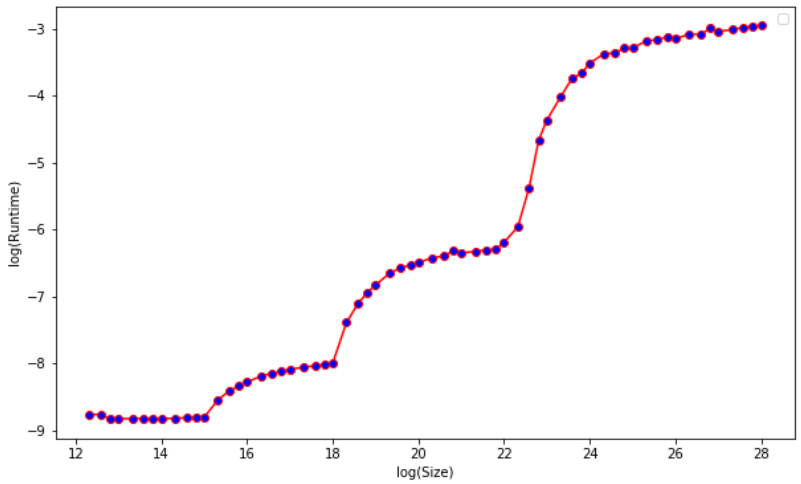


Computer2:

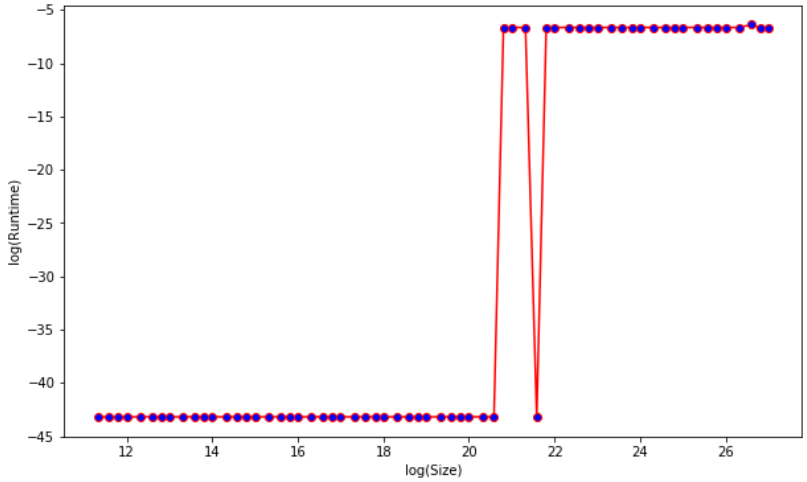


To compare with two diagram and two computer configuration. Computer2 has higherspeed at initial with 2^{-9}, and for computer1, it has 2^{-8} . Computer2 has higher slope that the speed changes more rapid. Computer1 has more flat slope. The reason of the speed difference and cache size difference is due to the difference of CPU and memory. As we can see the speed range of computer1 is from 2^{-9} to 2^{-3} and the speed range of computer2 is from 2^{-9} to $2^{-6.5}$.
The processor of both computer is Intel i5. The clock speed of computer1 is 1.4 GHz and the clock speed of computer2 is 3.0 GHz. Therefore, computer2 has more computation power than computer1. When two computers run the analysis code, the computer2 will perform higher running speed than the computer1 does. Consider the clock speed is the main factor that affect the computer running speed. To see why computer2 has a rapid change on speed and higher slope than computer1, the reason might be computer2 has larger L1 cache size and L2 cache size than the computer1.

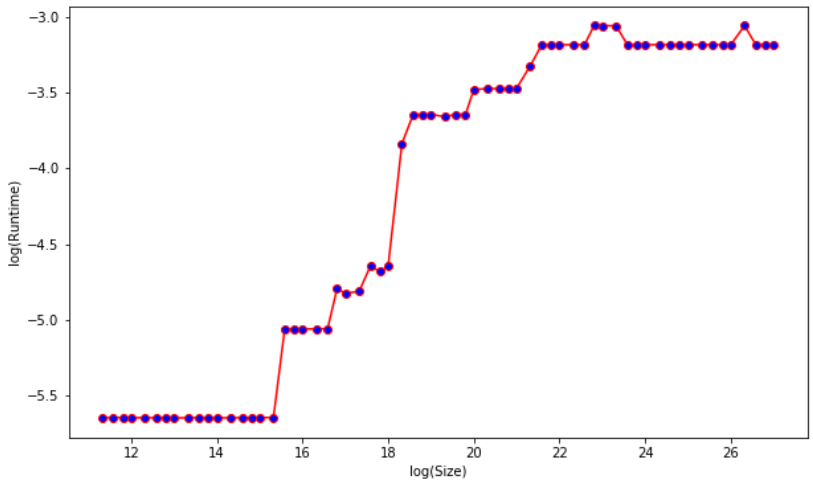
- Experiment 2:
  Computer3:
  - iterations = 1,000,000; /* perform one million accesses per test */



  Computer4:
  - iterations = 1,000,000; /* perform one million accesses per test */



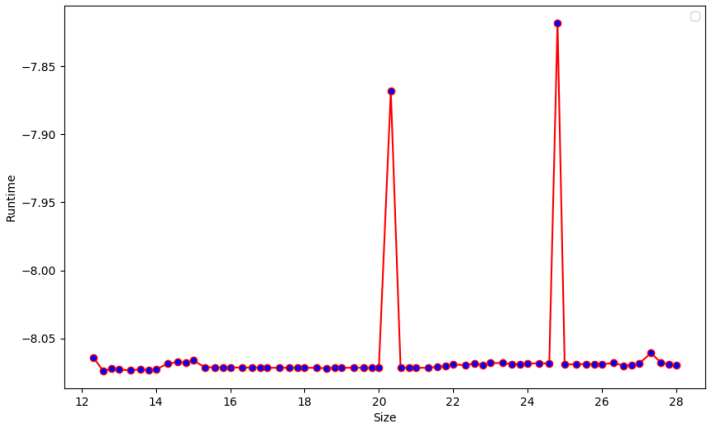
  - iterations = 10,000,000; /* perform 10 million accesses per test */



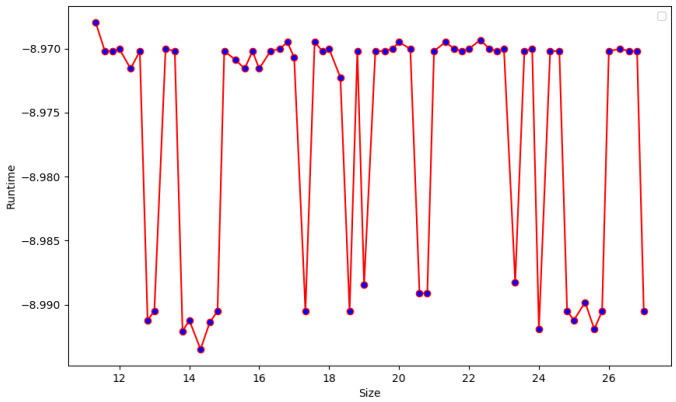
From the plots we can clearly see the cache-miss point where the runt tine increases sharply, but the run time is different from each other. If the number of iterations is equivalent (1,000,000 times per test), it is true that the run time of computer4 is much faster than computer3.
Here, what I want to draw your attention to is, the plot of computer4 is irregular while the number of iteration is 1,000,000, we can't identify the different cache and their memory hierarchy according to the plot, there is only sharp increasing and decreasing but no clear hierarchy. When I change the iteration number to 10,000,000, the run-time-to-data-size plots becomes regular, as the second Lenovo plot shows, we can easily observe cache size with different level.
As far as I'm concerned, the reason is that the number of accesses can influence the performance of computer in some cases. Maybe the number of accesses is too small for the computer4 to meet cache missing, there is some optimization mechanism to prefetch the data from memory under such "slow" accessing speed even though there is random shuffle.
Here we can also compare the memory bandwidth at L1 cache point between two types of computers, from the result of lab, the bandwidth of computer3 is 14.001899777761 Mb/s, but for computer4, in the case that the number of iterations is 1,000,000, the run time at L1 cache is almost 0, so the bandwidth is a huge number; but in the case that the number of iterations is 10,000,000, the bandwidth at L1 cache point is 262144 / 1024/ 1024 / 0.03999901 = 6.250154691328 Mb/s, which is smaller than that of computer3. So, the performance of cache will influence the value of bandwidth.

- **Analysis of Shuffle off**

- Experiment 1:
  Computer1:

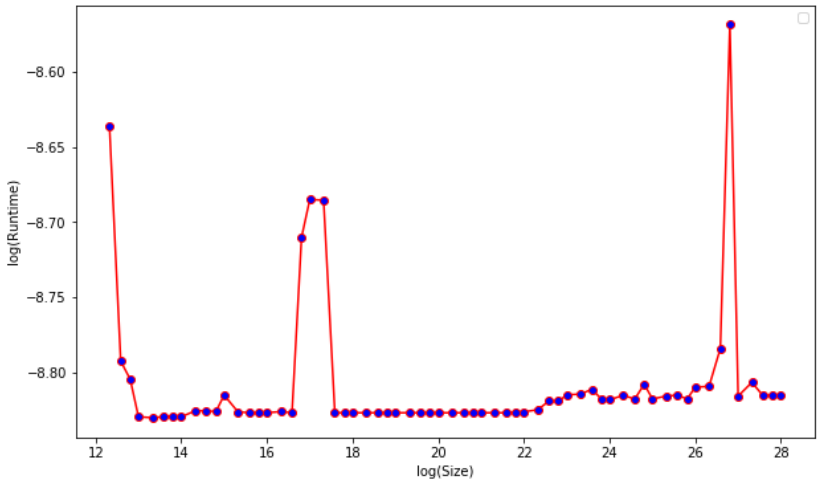


Computer2:



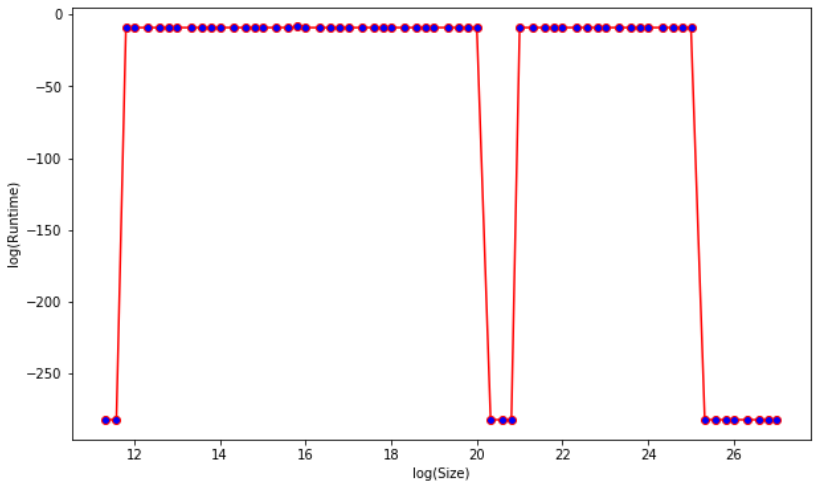
The result of computer1 has flat running time while size increase. There are two peak point on the diagram. As of diagram of computer2 result, to squeeze the range of running time, the result is nearly a flat line with no obvious peak point.

- Experiment 2:
  Computer3:
  - iterations = 1,000,000; /* perform one million accesses per test */

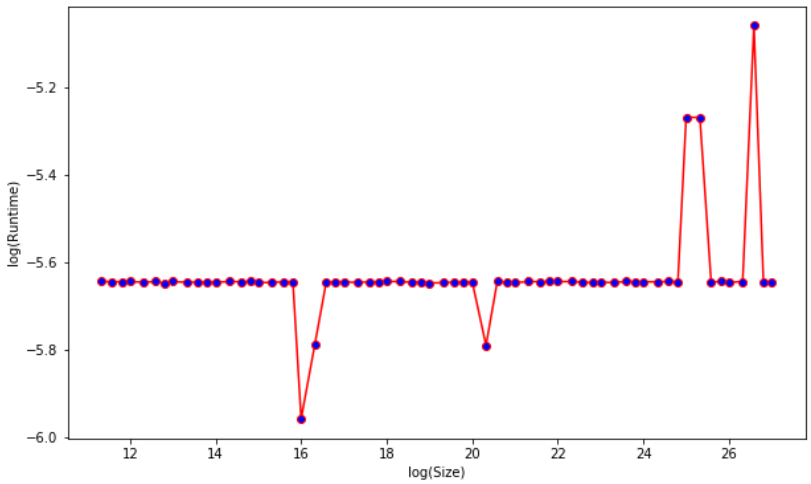


Computer4:

- iterations = 1,000,000; /* perform one million accesses per test */



- iterations = 10,000,000; /* perform 10 million accesses per test */

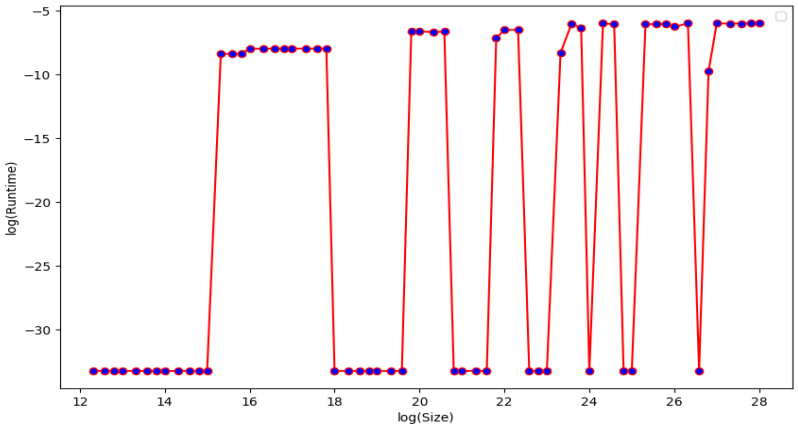


From the pictures above, we can't identify the cache missing point since there is stride prefetching to prefetch the data from memory, but there is still something we can discuss and explore.

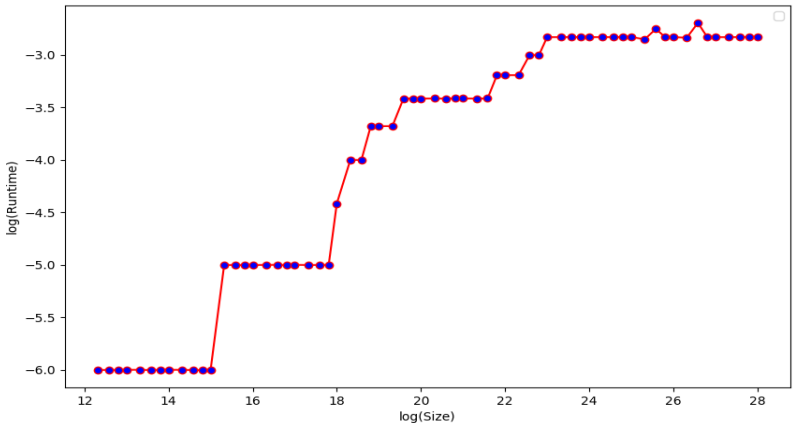
It is obvious that the run time of computer4 is much shorter than the run time of computer3 in the case that the number of iterations is 1,000,000. After our studying, the most likely reason is the run time of cache is closely related to the performance of CPU. Because the L1 cache has already fit into CPU, the L1D cache and L1I cache are set next to the pipeline of CPU, than the L2 cache is set next to the L1D cache, in such case the stride prefetching by cache is equal to stride prefetching by CPU. Therefore, the performance of cache is the performance of CPU, there is no surprising the computer4 did much better than computer3.

➢ Problems and Thoughts

1. Irregular result on shuffle on mode:
   Computer5:

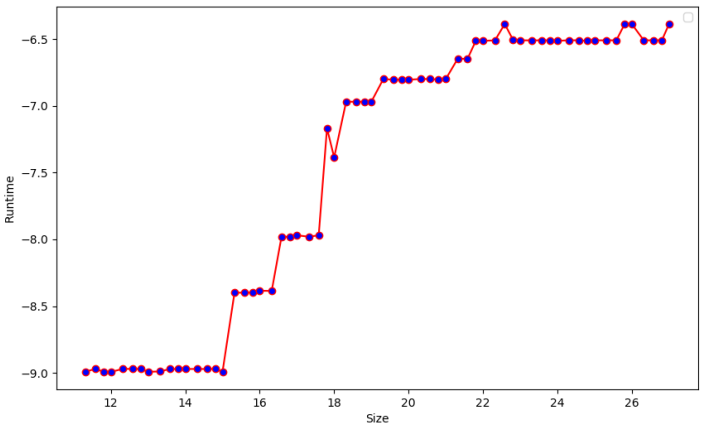


As we can see from the diagram, we got so many zero on runtime, which is not valid.
I change iterations from 1 millions to 10 millions, the new result:

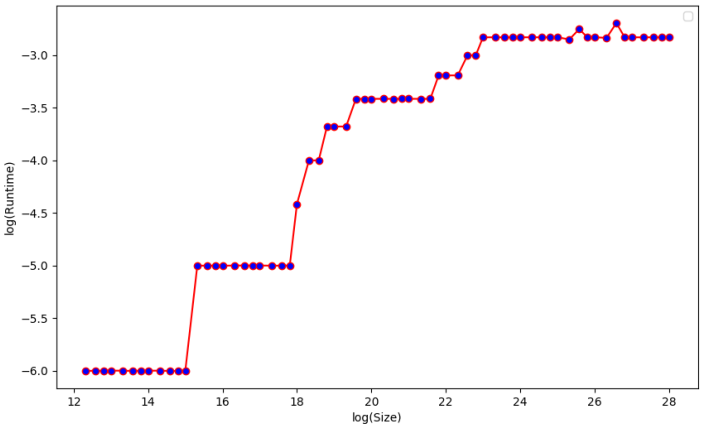


I guess with the high cpu performance, iteration cost much less time so sometimes we got invalid result. As we increase the iteration value, the runtime increases but the shape of diagram is correct.

2.inaccurate cache size:



According to this figure, we consider L1 size of this cpu is 2^15, L2 is , L3 is, which is significantly smaller than the number given from the Intel official site. After we close some application like chromes, the new figure is:



which seems more plausible but still lower than expect. We think windows system usually have more background process, which take up a lot of cache memory than Mac's.