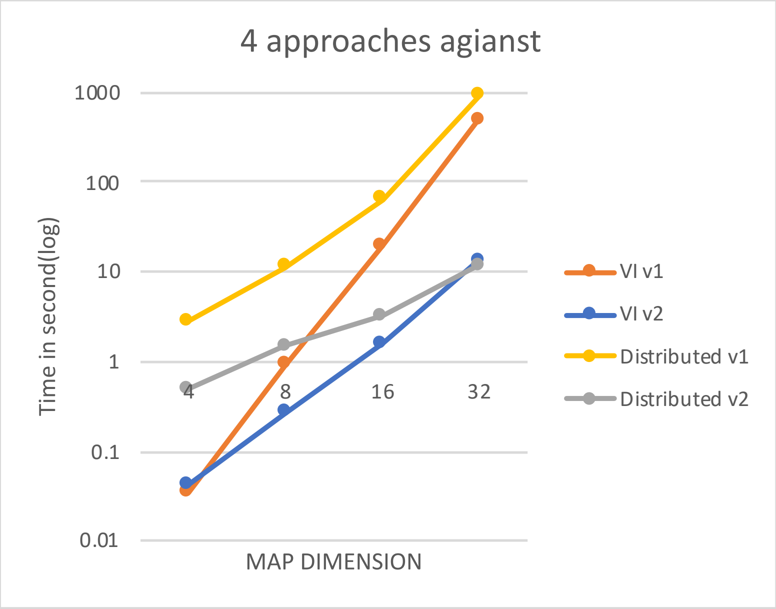
CS533 assignment 2

Jui-Hung Lu. 933293709 Chi Wen. 933276677

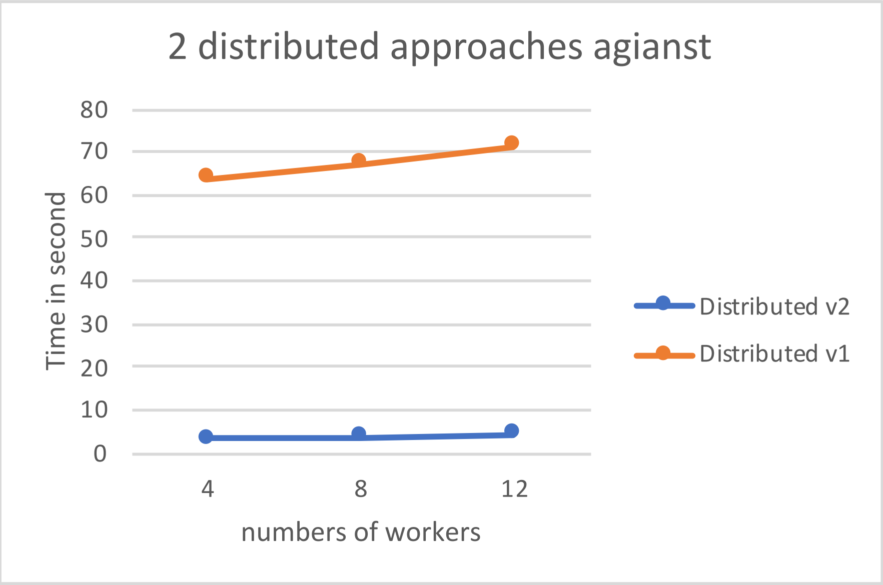
1.A plot that shows the running time of the above 4 approaches against the map sizes f 8, 16 and 32.



The y-axis is taken log$_{10}$

Map size	4*4	8*8	16*16	32*32
Sync Value Iteration v2:	0.04213	0.26719	1.55514	13.04104
Sync Value Iteration v1:	0.03537	0.91192	19.44035	481.38527
Sync distributed v2:	0.49584	1.48332	3.17386	11.53952
Sync distributed v1:	2.80208	11.22538	63.94160	895.57175

2.A plot that shows the running time of both distributed approaches agianst the number of the workers with 2, 4 and 8 workers.



3.Briefly explain why the second distributed method is faster than the first one?

The main reason is that it reduces the time of assigning job at each time, which include using ray.get, ray.wait and the remote function.
Take 8*8 dimension map with four workers for example.
In the first distributed method, after a worker finish processing, it will then be assigned another state. That is, 8*8=64 times for job assigning in first distributed method. However, in second distributed method, each worker will be handling each batch. Therefore, only 4 times job assigning will need to be done in this case.

4.Compere the best distributed method with the best non-distributed approach. Which one is better? Briefly explain why.

According to the plot in question one above, the best one is distributed method v2.
Although the non-distributed approach is faster than distributed approach in the beginning. However, in the 32*32 dimension map, the distributed approach v2 become faster than non-distributed approach v2, and even the distributed approach v1 is showing that it may be faster than non-distributed approach v1 in the bigger dimension map.
After several testing, we discover that the distributed approach can reduce the computing time, but it need a little time initializing.
In conclusion, when the map is smaller, the non-distributed approach has better running time because the Ray initializing time is longer than its reduction in computing time. However, in a bigger map like 32*32, the distributed approach can reduce lots of computing time and become the fastest one.