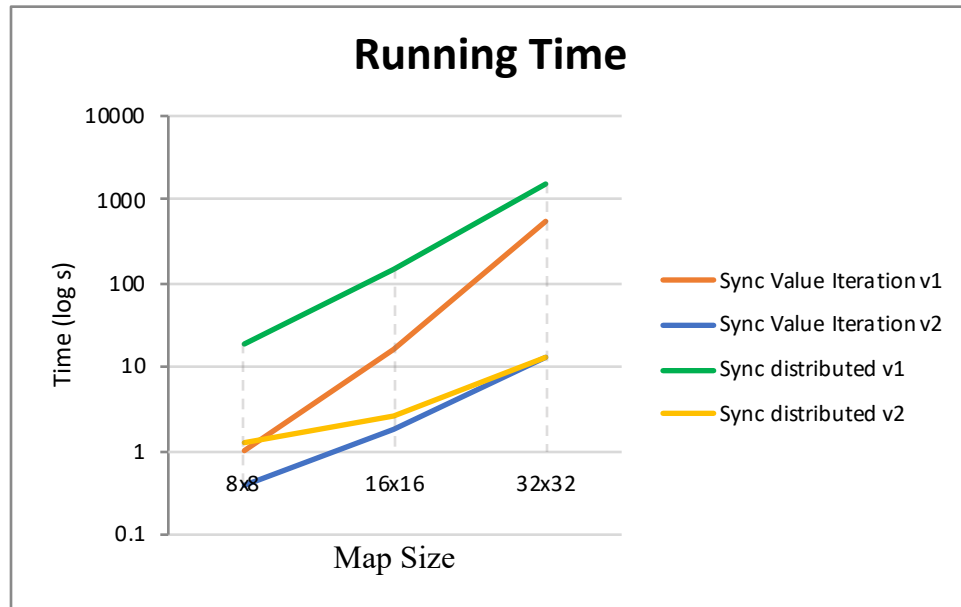
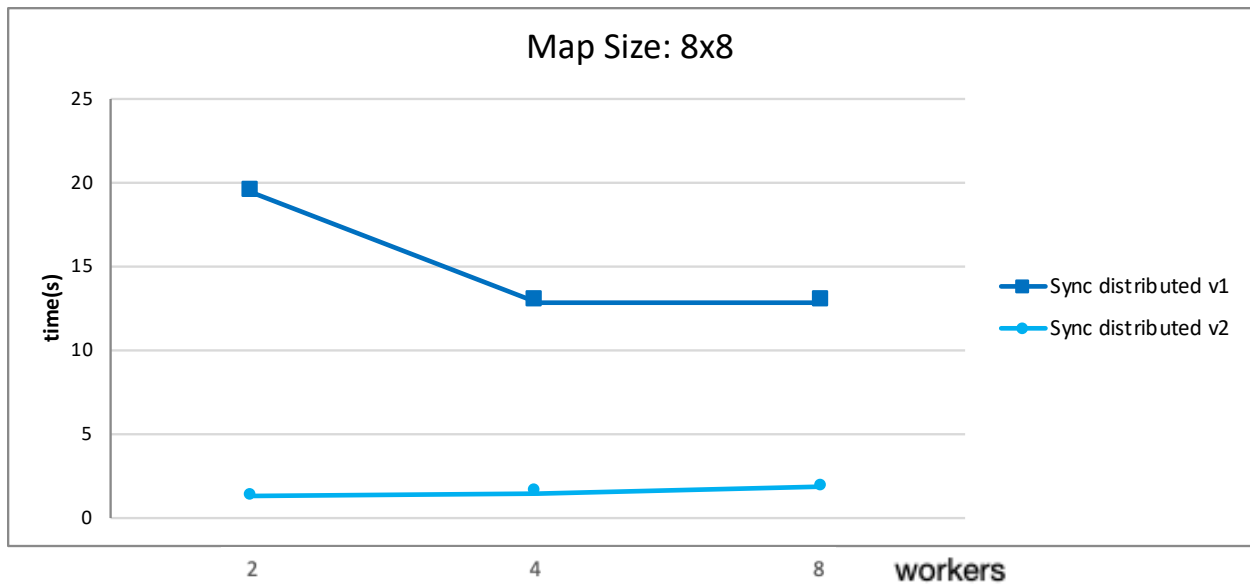


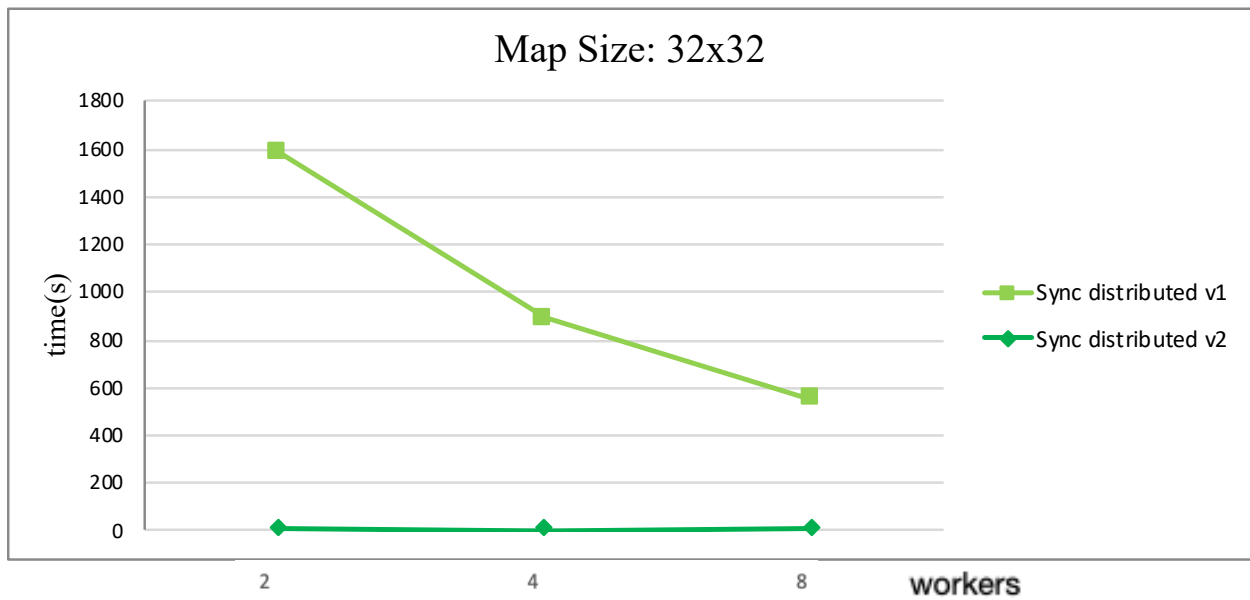
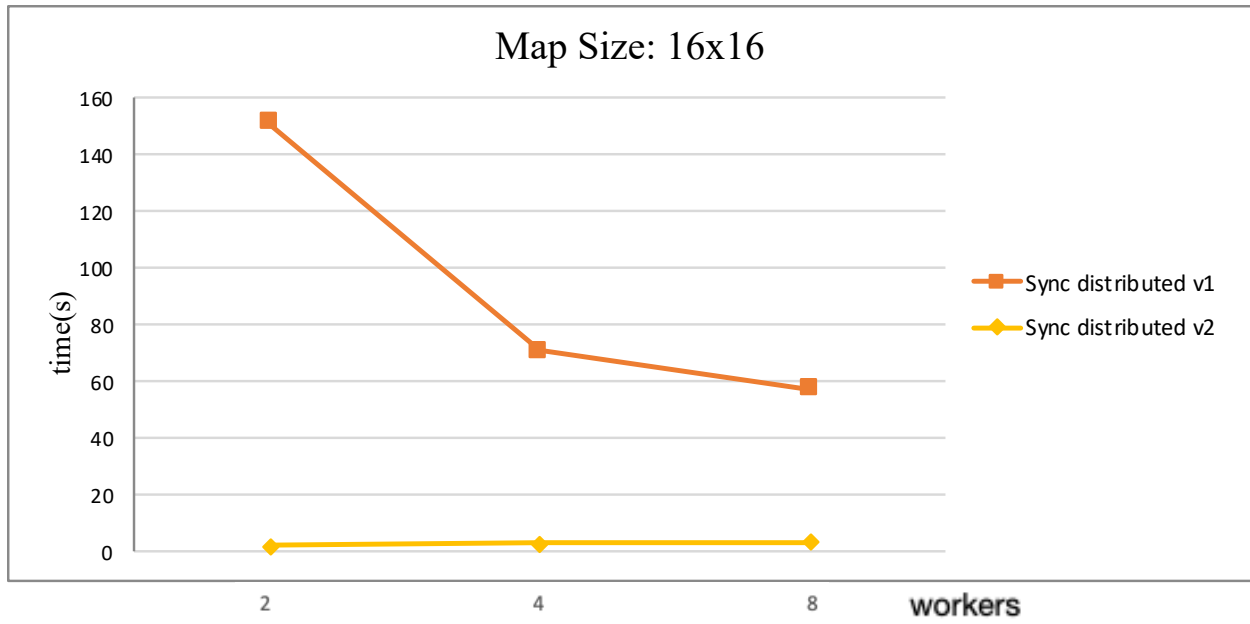
Running time of the 4 approaches against the map sizes of 8, 16 and 32.



Running time for both distributed approaches against the number of the workers with 2, 4 and 8 workers.



CS533 HW2



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

For the first method, a worker processes on a state for value iteration at a time and gets assigned another state when it is finished. However, this may waste time on `ray.wait`, `ray.get`, and remote function. In the comparison, the second one first divides the states into four chunks. Each worker receives a chunk of sub-states to work on. They can finish updating a series of states without getting interrupted for waiting other workers and receiving the task. In the 8x8 map example, the first method needs to assign 64 times for each update while the second method only needs to assign 4 times.

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

The best distributed method is “Distributed Synchronous Value Iteration V2” while the best non-distributed approach is “Synchronous Value Iteration Using `GetSuccessors()`”. When the map size is smaller than 32x32, non-distributed one is better. However, distributed one is better when the map is 32x32 and larger. The reason may be that it takes time to initialize Ray related function. The workers can do the computation in parallel to reduce computational time, but the reduction time may be less than the increased initialization time. While the size of map becomes larger, meaning the amount of computation grows larger, parallel workers show its strength to cut down on the running time.