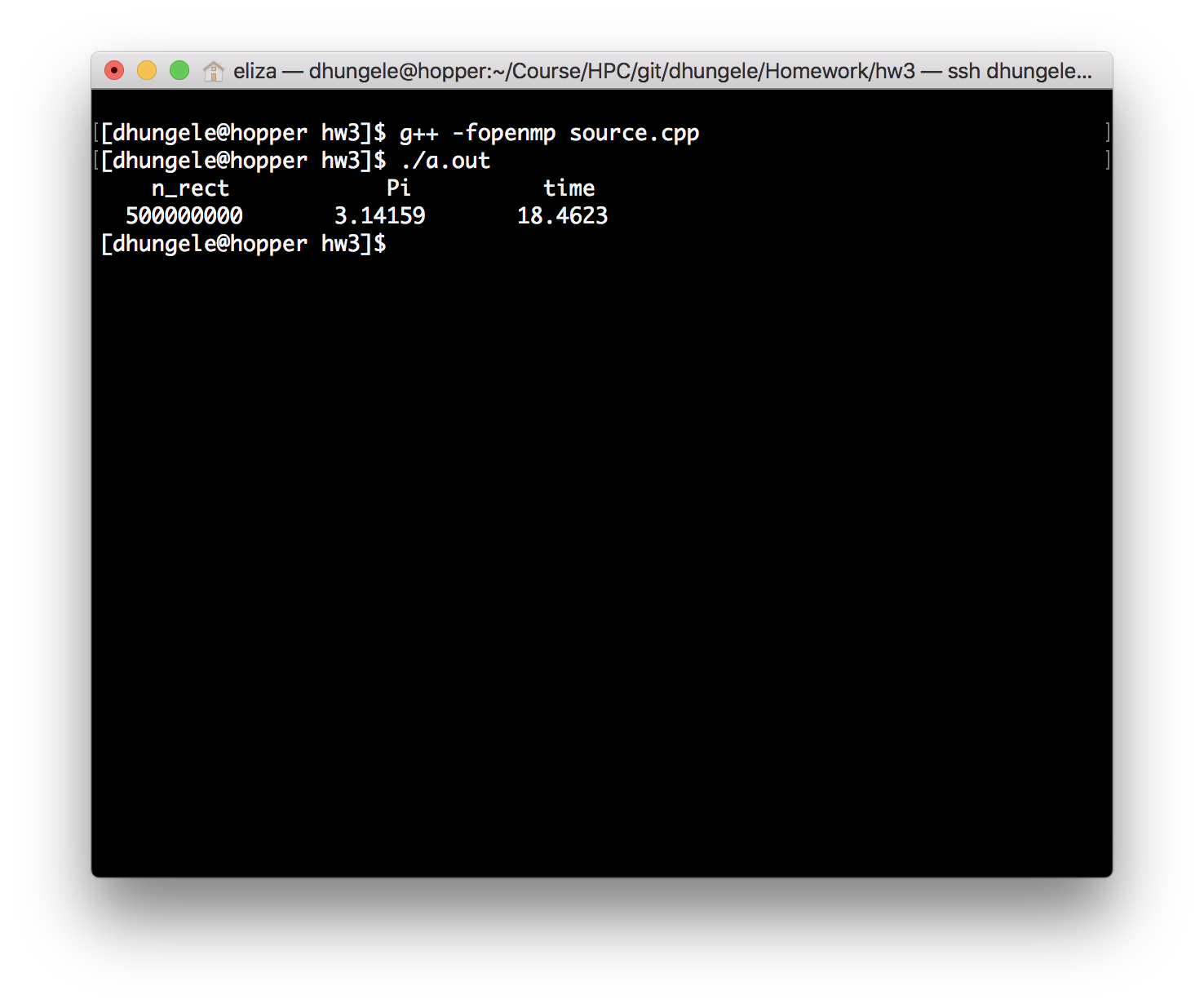
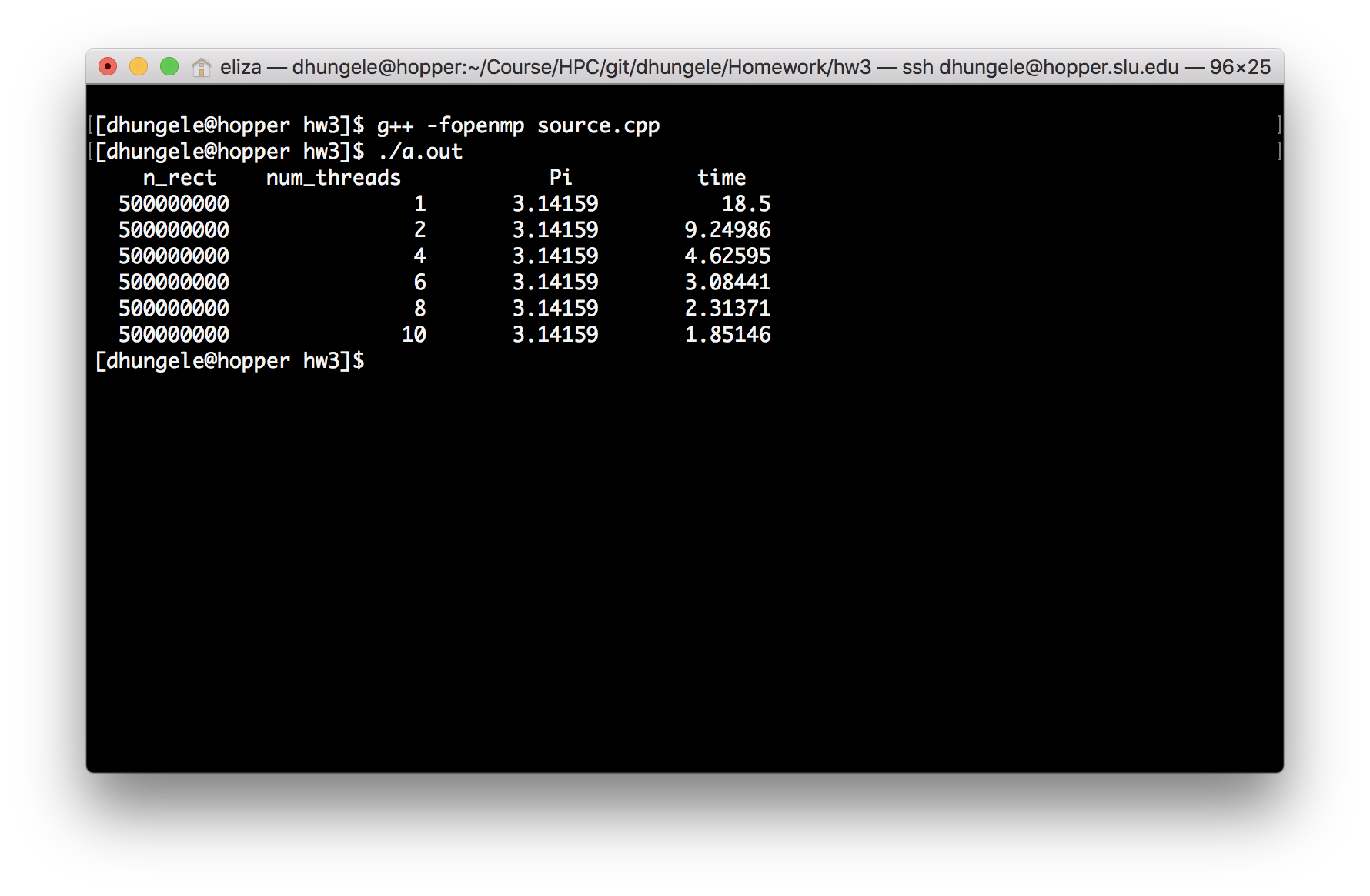
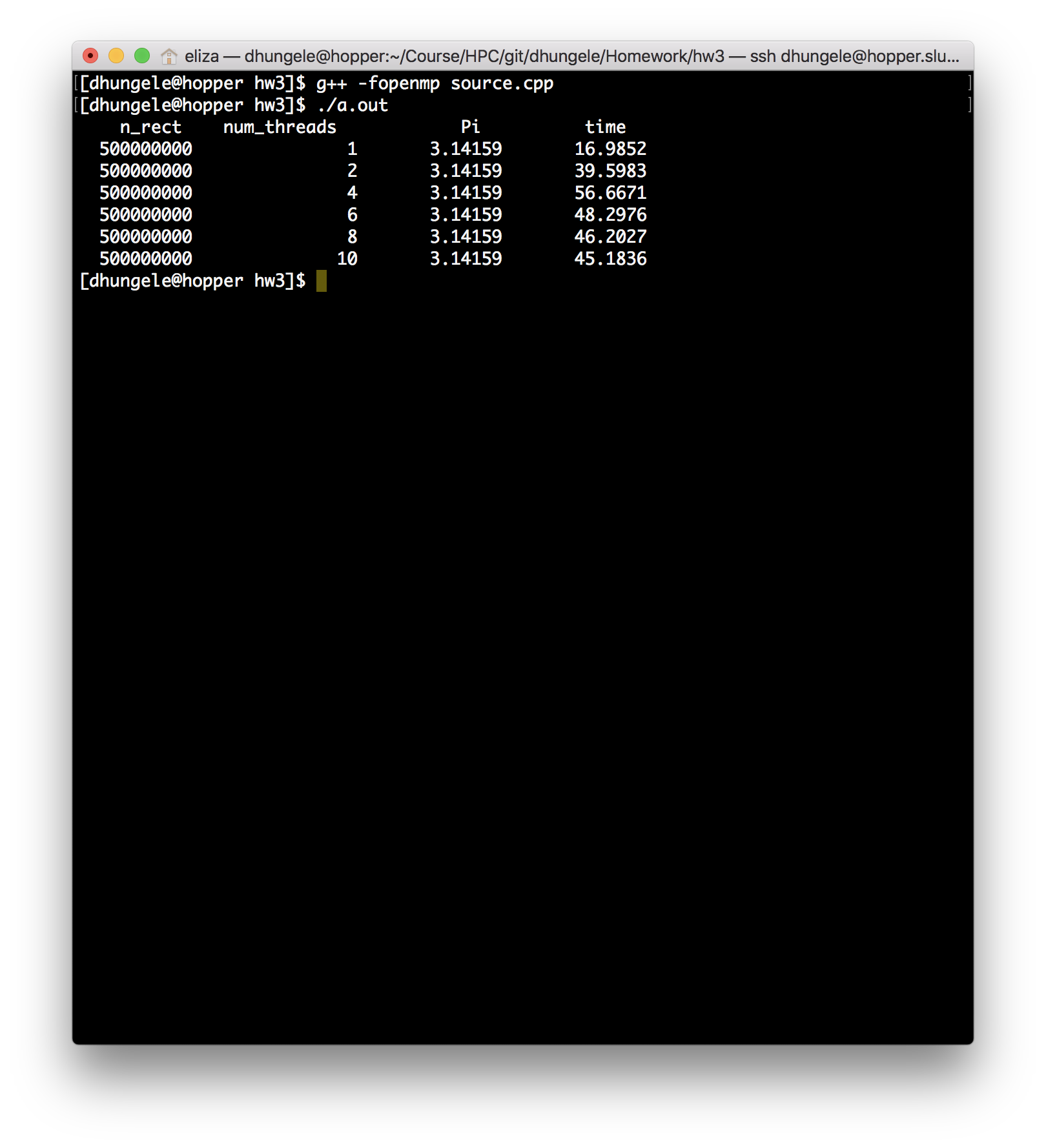
1.

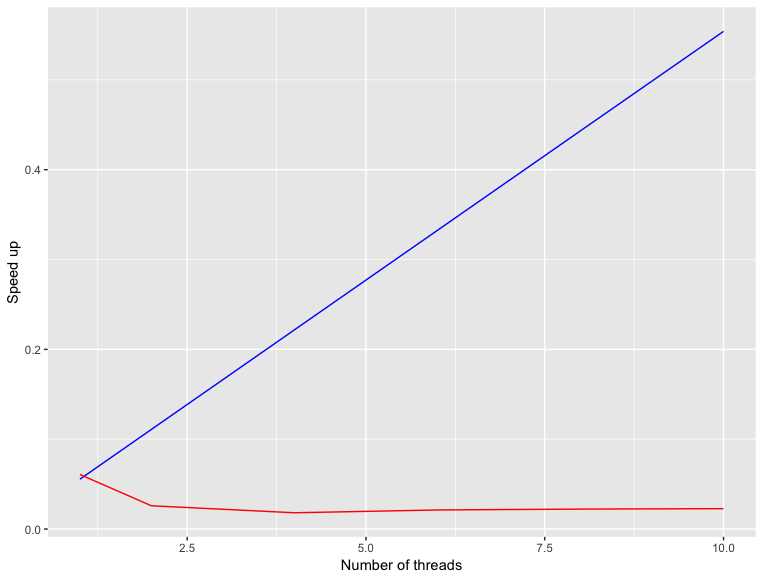


2.



3.



4. 

Plot in the left denote the speedup obtained for different number of processor; Blue: using atomic, Red : using reduction

ANALYSIS:

Atomic does not work better for this because the ATOMIC directive specifies that a specific memory location must be updated atomically, rather than letting multiple threads attempt to write to it because of which the sum variable is updated atomically and other threads can not update it.

On the other hand the reduction clause creates a private copy for each list variable and initialize it for each thread. At the end of the reduction, the reduction variable is applied to all private copies of the shared variable, and the final result is written to the global shared variable because of which there is no wait time until which the variable sum is updated. So, reduction is faster.