Trapezoidal Rule / Computation of PI

Pi can be written as area under the curve, we know

$$\pi=4*\int_{0}^{1}\frac{1}{(1+x^2)}dx$$

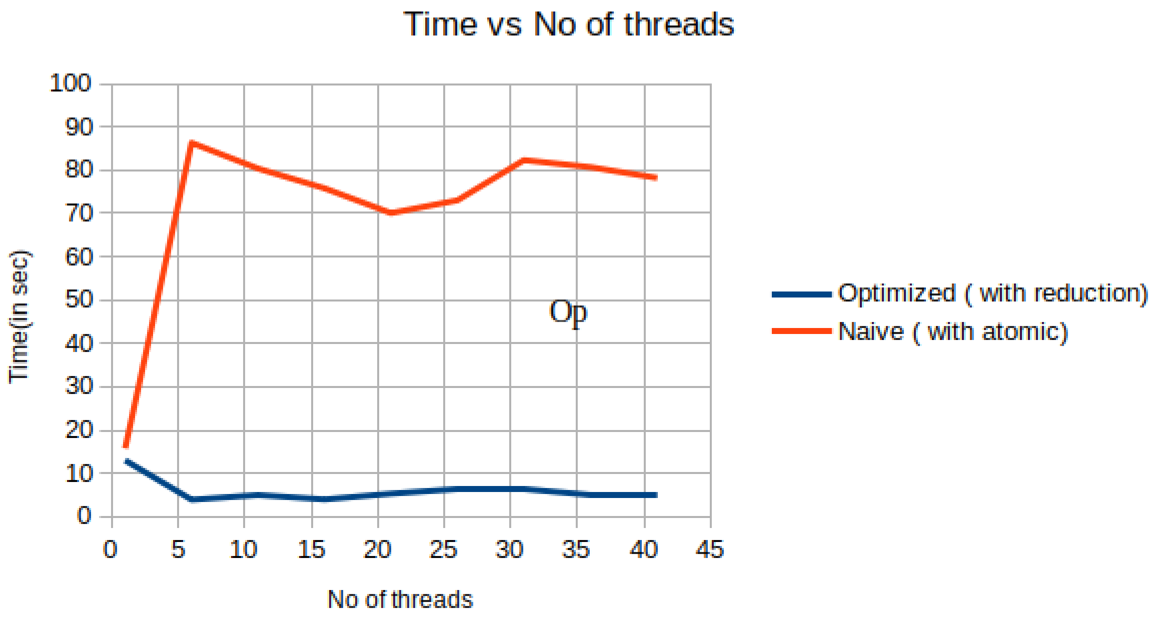
Using trapezoidal rule :

$$\pi=4*\sum_{i=0}^{n-1}\frac{1}{1+(i/n)^2}\frac{1}{n}$$

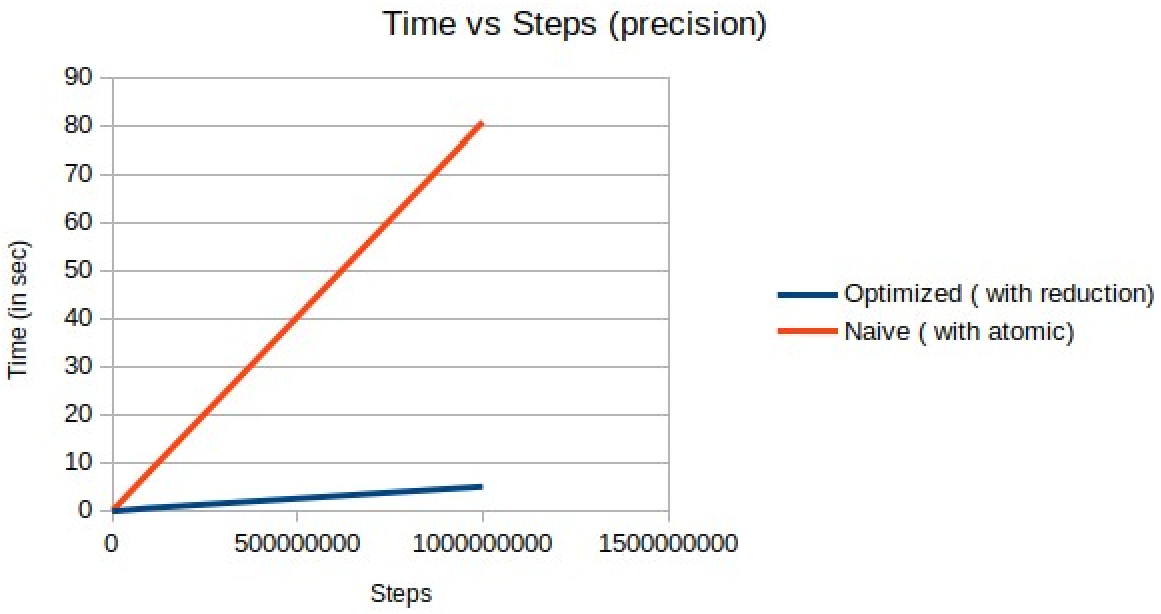
I used two methods to calculate the given sum
1. By adding each value of summation atomically. For atomic addition “#pragma omp atomic” is available which uses a built-in hardware instruction
2. Using the reduction by putting “reduction(+:sum)” in the #omp omp parallel”

The reduction method is fast because:
1. Reduction secretly creates a temporary private variable for each thread’s running sum. Each thread adding into its own running sum doesn’t interfere with any other thread adding into its own running sum, and so threads don’t need to slow down to get out of the way of each other.
2. Reduction automatically creates a binary tree structure, to add the N running sums in log2N time instead N time.

Variation in time v/s no of threads: It can be seen that time and number of threads are not strongly correlated, there might be other overheads coming into picture while increasing the number of threads.



Variation in time v/s precision: As the number of steps increases or precision decreases the time decreases as computation complexity increases.


Time vs Steps (precision)
Time (in sec)
Steps
Optimized (with reduction)
Naive (with atomic)