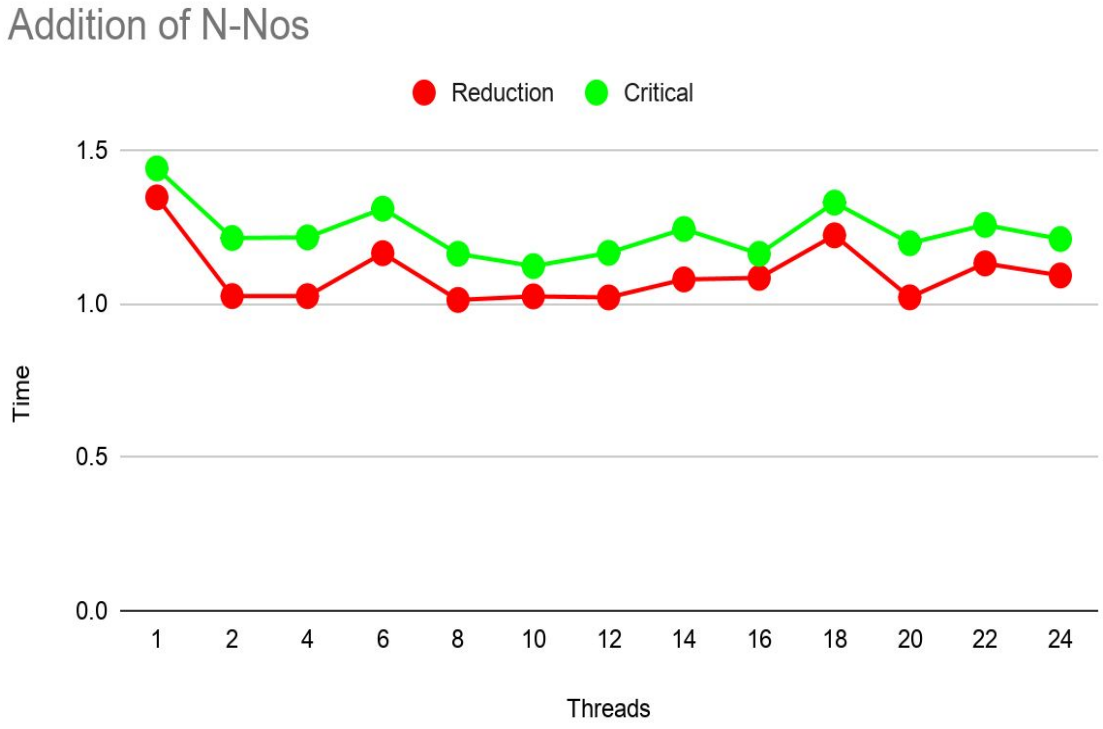
High Performance Computing

Lab-3 Report by Hrishikesh Vedantam

- OpenMP
  - Addition of N-numbers (double precision numbers)
    Identify the challenge
    - The main challenge here is that the data with which computations need to happen are dependent on one another which, if not handled properly, may lead to data races.
    - Implement using Reduction: Each thread gets its own copy of the sum variable, and at the end of the parallel for region, all the local copies of the total sum are summed up to get the grand total.
    - Critical section: The code in a critical section can only be executed by a single thread at any given point in time. We can use a critical section to prevent different threads from accessing the same memory location in an unsafe manner, and thereby avoid data races.
    - It takes the least amount of time for Reduction is with **8** threads.
    - The parallel fraction of Reduction for this problem is **0.7519271511**
    - It takes the least amount of time for the Critical Section with **10** threads.
    - The parallel fraction of Critical Section for this problem is **0.7791261462**
    - This can be recognized by the values and the graph as shown below:

Threads	Time	Time
1	1.34525	1.4396
2	1.02416	1.21306
4	1.02395	1.21513
6	1.16369	1.30854
8	1.01153	1.16173
10	1.02318	1.12163
12	1.01975	1.16475
14	1.07837	1.24249
16	1.08338	1.16158
18	1.22188	1.32837
20	1.02003	1.19609
22	1.13046	1.2557
24	1.09122	1.20974
	N-nos Reduction	N-nos Critical



- Vector Dot Product
  - The mathematical representation of Vector Product is:

$$\vec{a} \cdot \vec{b} = \sum_{i=0}^{n-1} a_i b_i$$

  - Implement using Reduction: Each thread gets its own copy of the final product variable, and at the end of the parallel for region, all the local copies of the total sum are summed up to get the grand total.
  - Critical section: The code in a critical section can only be executed by a single thread at any given point in time. We can use a critical section to prevent different threads from accessing the same memory location in an unsafe manner, and thereby avoid data races.
  - Here we can see the effect of the Critical section performing better than the reduction method due to the working mechanism.
  - It takes the least amount of time for Reduction is with **6** threads.
  - The parallel fraction of Reduction for this problem is **0.6659426104**
  - It takes the least amount of time for the Critical Section with **6** threads.
  - The parallel fraction of Critical Section for this problem is **0.6526687325**
  - This can be recognized by the values and the graph as shown below:

Threads	Time	Time
1	0.00153303	0.00157809
2	0.00188708	0.00137901
4	0.00128889	0.00159407
6	0.00102091	0.00102997
8	0.00160909	0.00163603
10	0.00248599	0.00168896
12	0.00159502	0.00114584
14	0.00152898	0.0011251
16	0.00182605	0.00142598
18	0.00118089	0.00147009
20	0.00164294	0.00105119
22	0.00135207	0.00135207
24	0.00166798	0.00142312
	VDP Reduction	VDP Critical