05/03/2025

| **version** | **result** | **tiempo** |
| --- | --- | --- |
| pi-v0 debug | 3.1416740338 | ? |
| pi-v0 omp | 3.1415926536 | 0.012847900 |
| pi-v2 debug | 3.1416740338 | ? |
| pi-v3 debug | 1.5864715641 |  |
| pi-v4 debug | 3.1416740338 |  |
| pi-v4 omp (critical) | 3.1415926536 | 5.009748936 seconds |
| pi-v5 omp (atomic) | 3.1415926536 | 0.703952074 seconds |
| pi-v6.debug (var local) | 3.1416740338 |  |
| pi-v6 omp | 3.1415926536 | 0.012123108 seconds |
| pi-7 debug (reduction) | 3.1416740338 |  |
| pi-7 omp | 3.1415926536 | 0.039309978 seconds |
| pi-8 debug | 3.1416740338  t0: pi=0.8086453736  t1: pi=0.7934991050  t3: pi=0.7617033646  t2: pi=0.7778261905 |  |
| pi-8 omp | 3.1415926536  t3: pi=0.7853980884  t1: pi=0.7853981884  t0: pi=0.7853982384  t2: pi=0.7853981384 | 0.012012959 seconds |
|  |  |  |

12. Finally, can you guess what access to which variable is causing the data race in pi-v3.c and, therefore, the incorrect result of pi?

sum ???

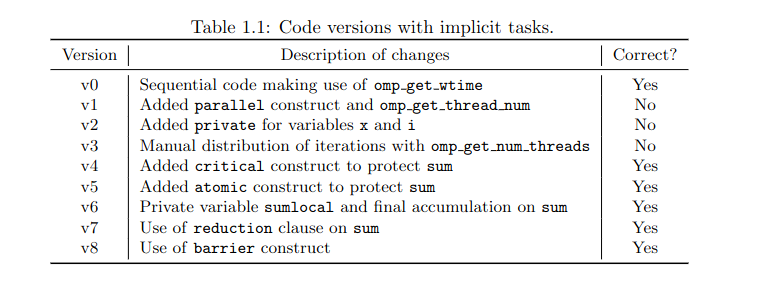
thread 1 → sum = 0.8

thread 2 → sum = 0.4

* atomic is faster than critical
* #pragma omp parallel private(i, x) reduction(+:sum)

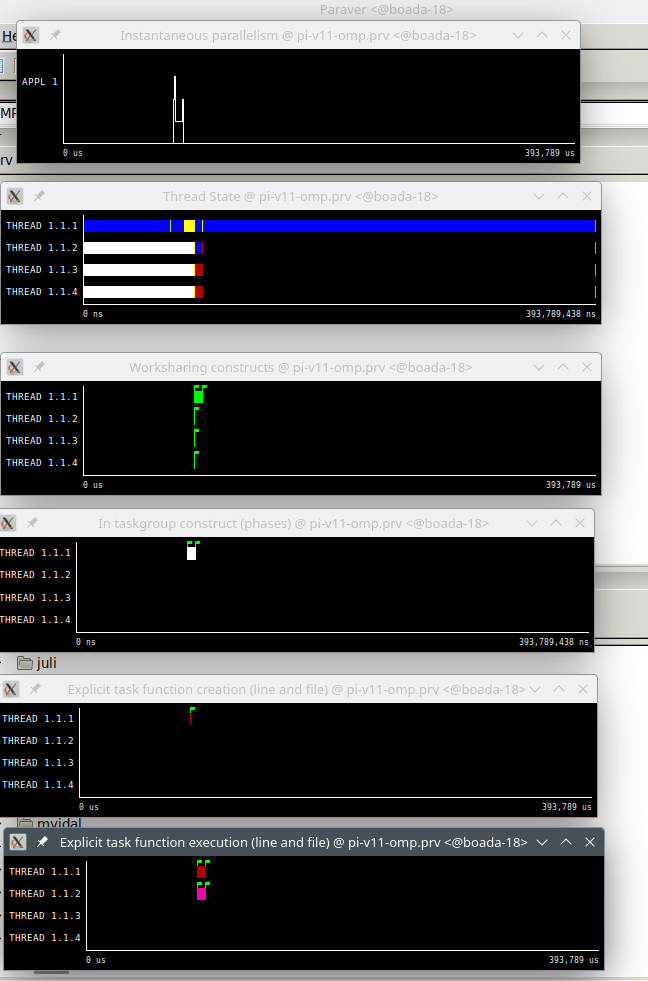
Reduction hace la copia local de la variable en cuestion

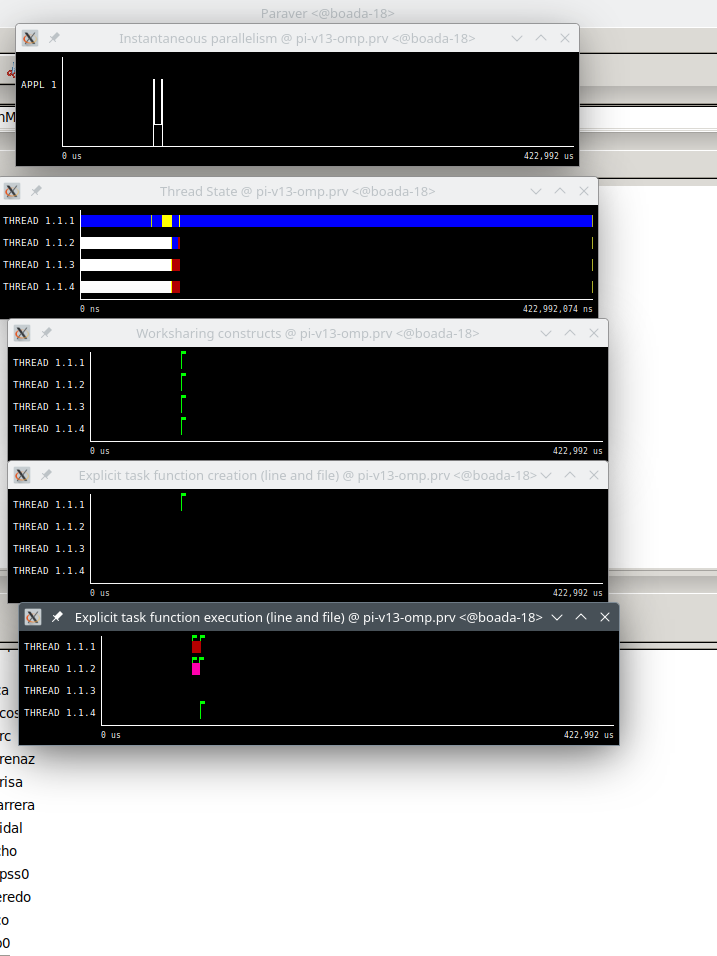
RESUMEN



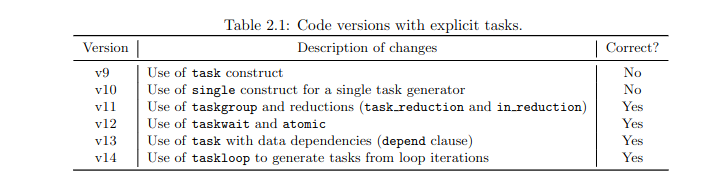
/\* Q1: What does omp\_get\_num\_threads return when invoked outside and \*/

/\* inside a parallel region?









1.single.c

