Connectarse al nostre server:

*ssh -X* [*paco1101@boada.ac.upc.edu*](mailto:paco1101@boada.ac.upc.edu)

contrassenya:

paucesco

cada cop q començem executar dins el directori labs:

*source environment.bash*

per transferir fitxers de boada al nostre directori:  
*scp paco1101@boada.ac.upc.edu:lab1/pi/pi seq.c .*

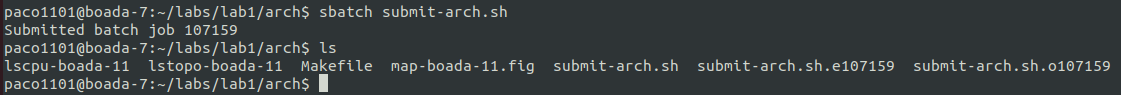
“you will be copying the source file pi seq.c located in directory lab1/pi of your home directory in boada to the current directory, represented with the ".", in the local machine, with the same name.”

per copiar el lab que toqui dins de boada:  
*cp /scratch/nas/1/par0/sessions/lab1.tar.gz .*

accedim al directori *lab1/arch/* i executem:

*sbatch submit-arch.sh*

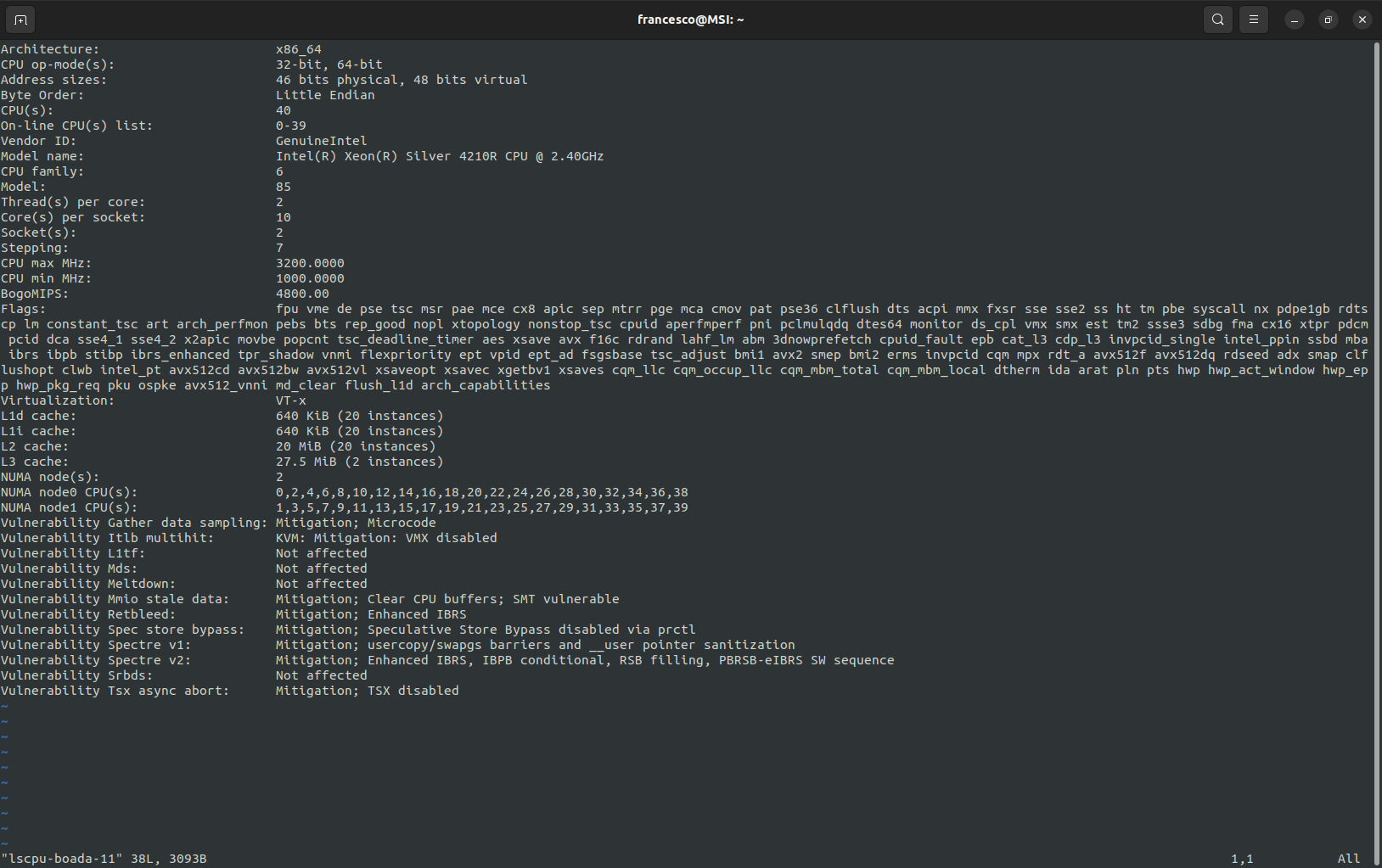
obtindrem 5 fitxers:



amb:

*vi lscpu-boada-11*

obtenim:



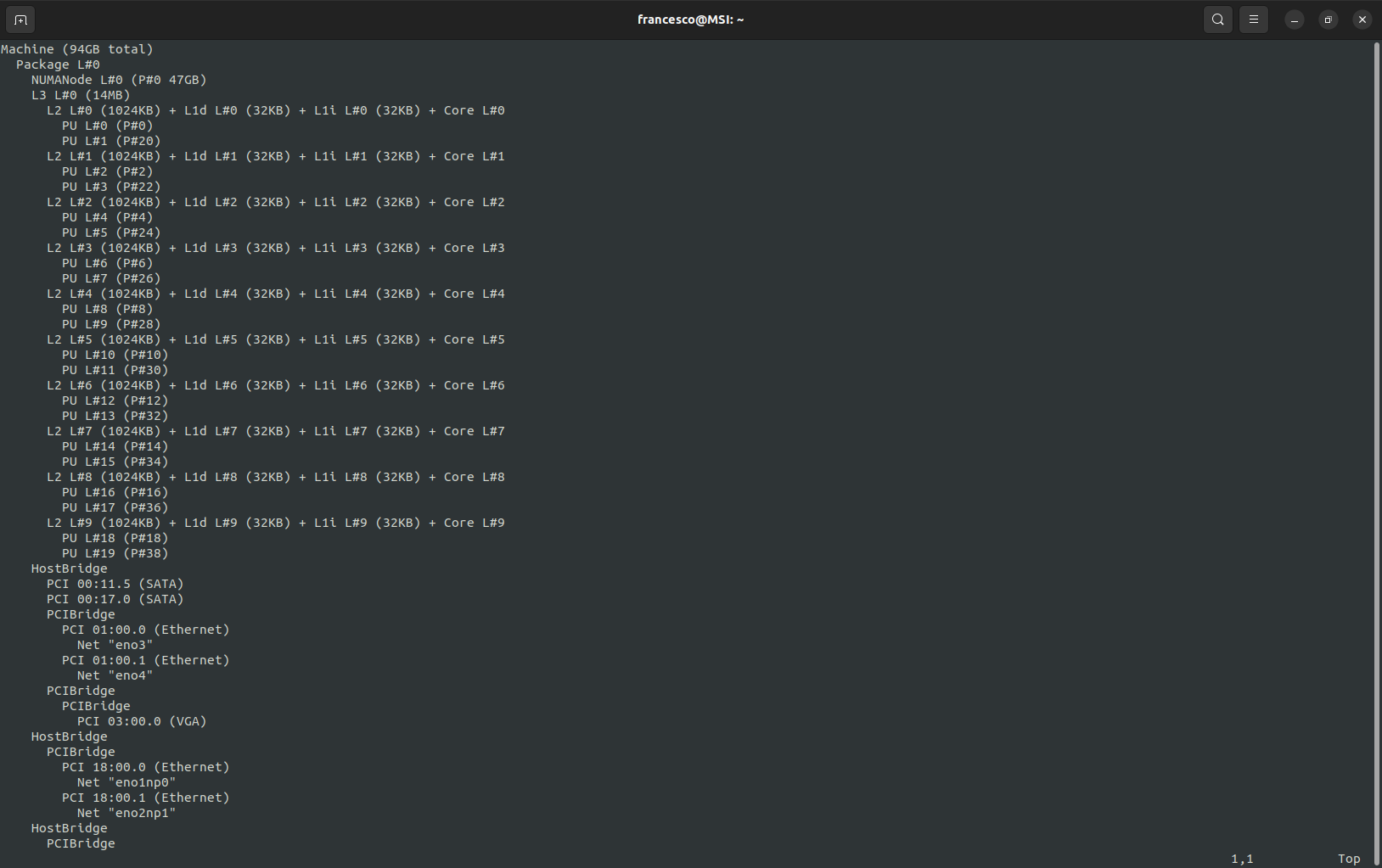
per tancar un arxiu obert (visualitzat amb la comanda vi):

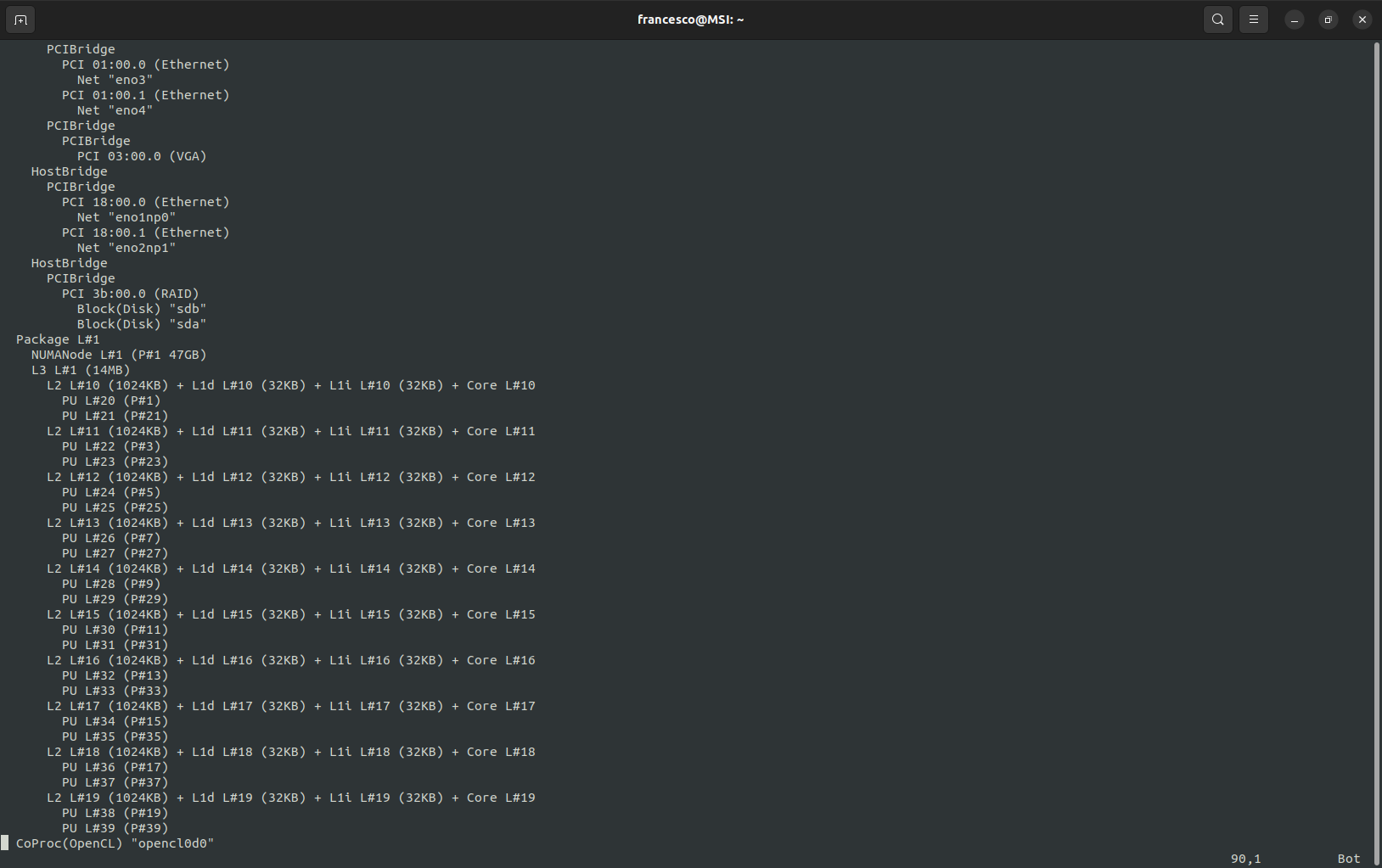
primer prémer *ESC* i després *:q*

amb:

*vi lstopo-boada-11*

obtenim:

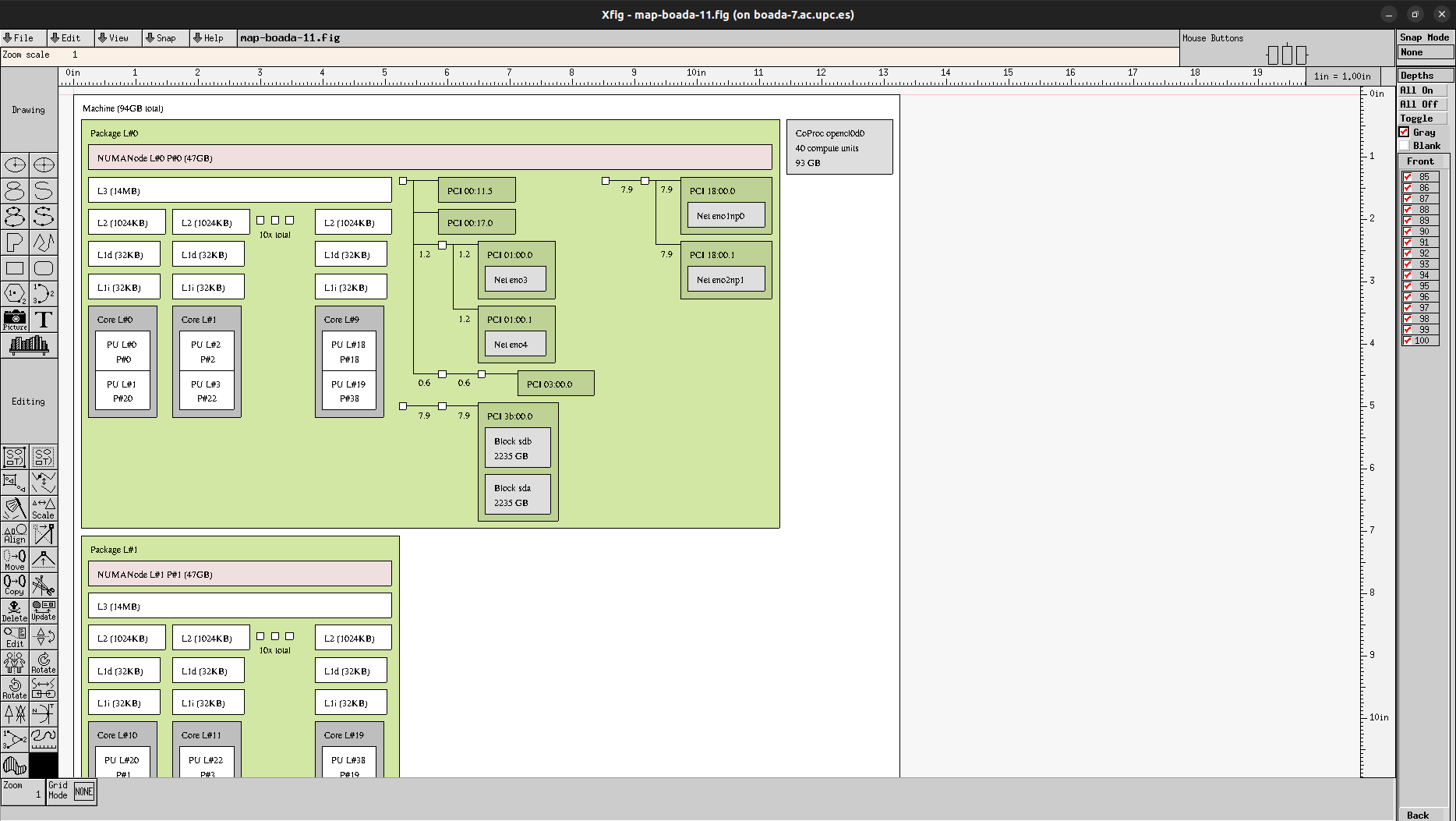


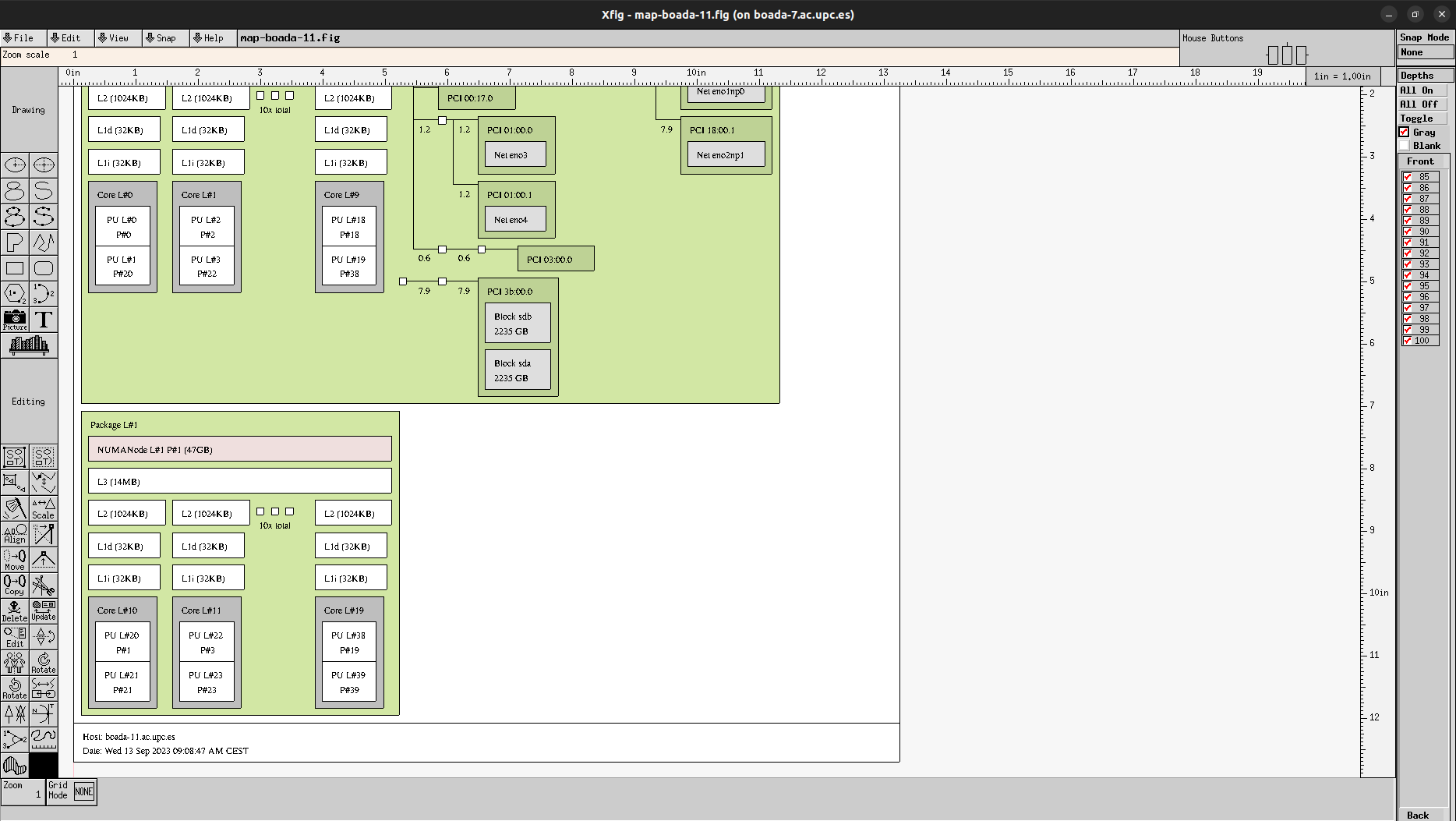


amb:

*xfig map-boada-11.fig*

s’obre:





Hi ha 20 cores

Cada core físic pot executar 2 threads en concurrència

Concurrència: que va saltant en el temps entre un i l’altre thread

Si li diem q executi fins a 20 instruccions les fara en paral·lel

I fins a 40 ho farà en concurrencia

Mem. Principal: 47G + 47G

Cada socket té 2 cores

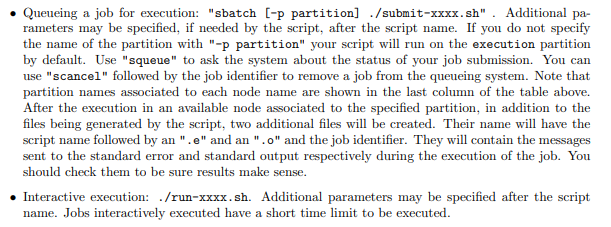
|  | Any of the nodes among boada-11 to boada-14 |
| --- | --- |
| Number of sockets per node | 2 |
| Number of cores per socket | 10 |
| Number of threads per core | 2 |
| Maximum core frequency | 3200 MHz |
| L1-I cache size (per-core) | 32KB |
| L1-D cache size (per-core) | 32KB |
| L2 cache size (per-core) | 1024KB |
| Last-level cache size (per-socket) | 14MB |
| Main memory size (per socket) | 47GB |
| Main memory size (per node) | 94GB |

per veure si l’script necessita paràmetres comanda:  
*usage*

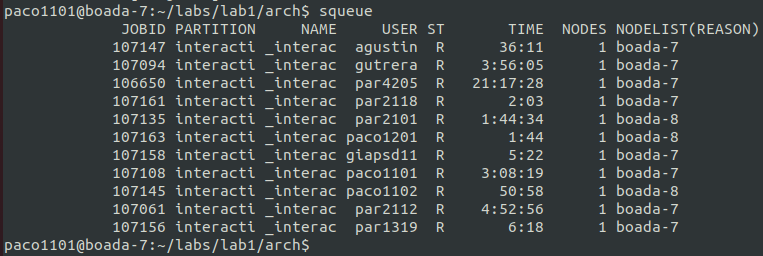
els submit s’envien a la cua amb:

*"sbatch [-p partition] ./submit-xxxx.sh"*

2 maneres d’executar:

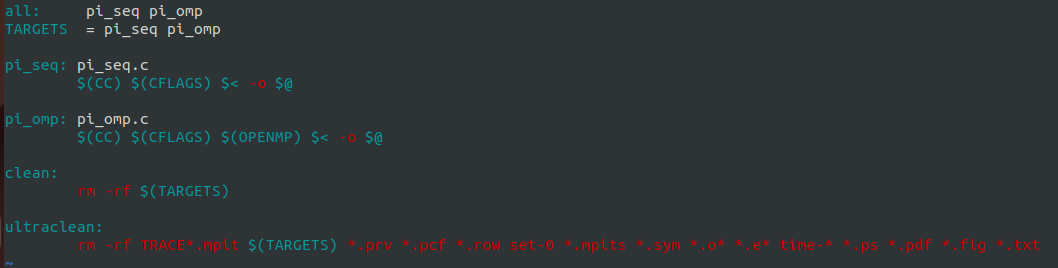


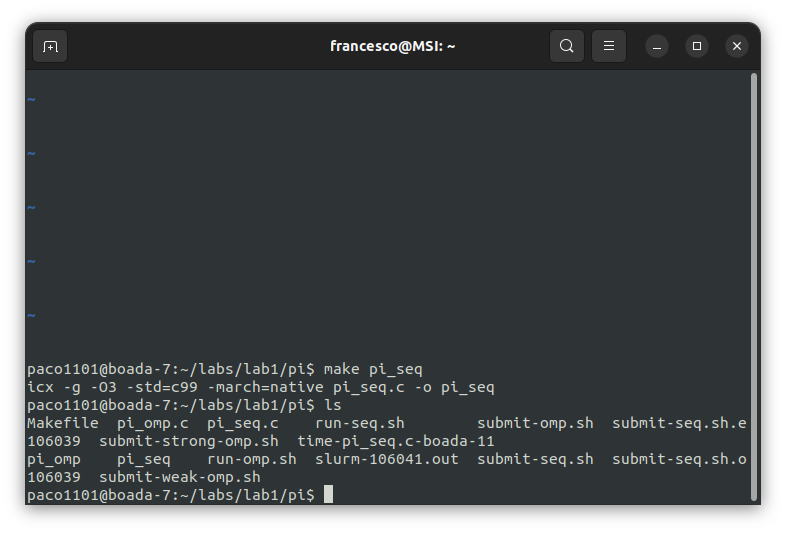
amb la comanda *squeue* obtenim:



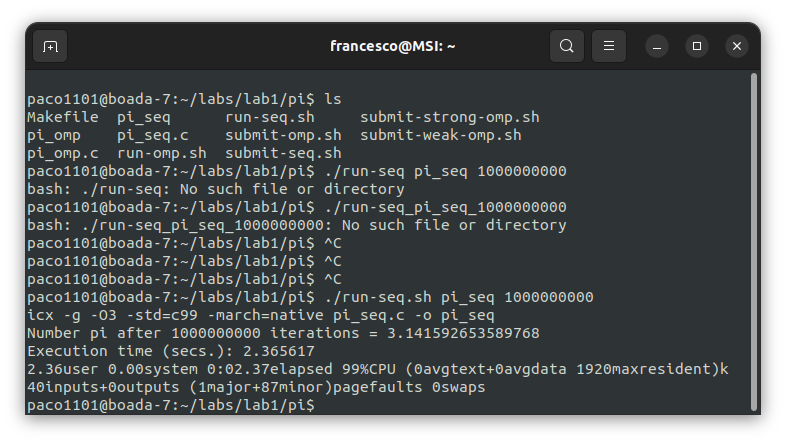
pi\_omp.c -> pragma reparteix els rectangles del pi de manera q cada thread executi un rectangle diferent.



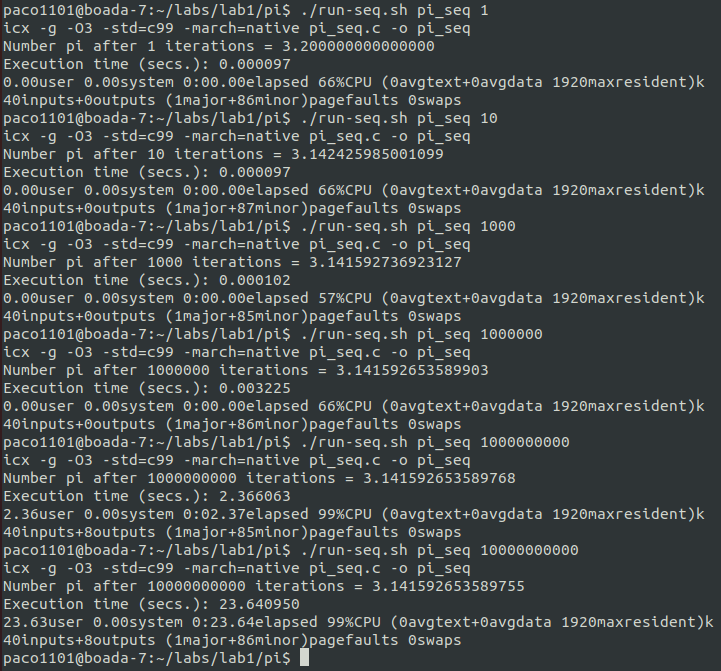


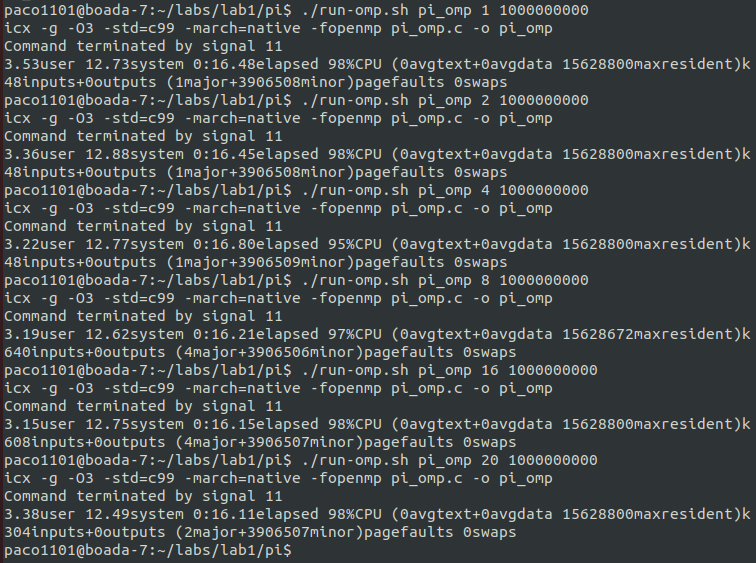


a part de pi\_seq es pot fer un make de pi\_omp (Targets del makefile)



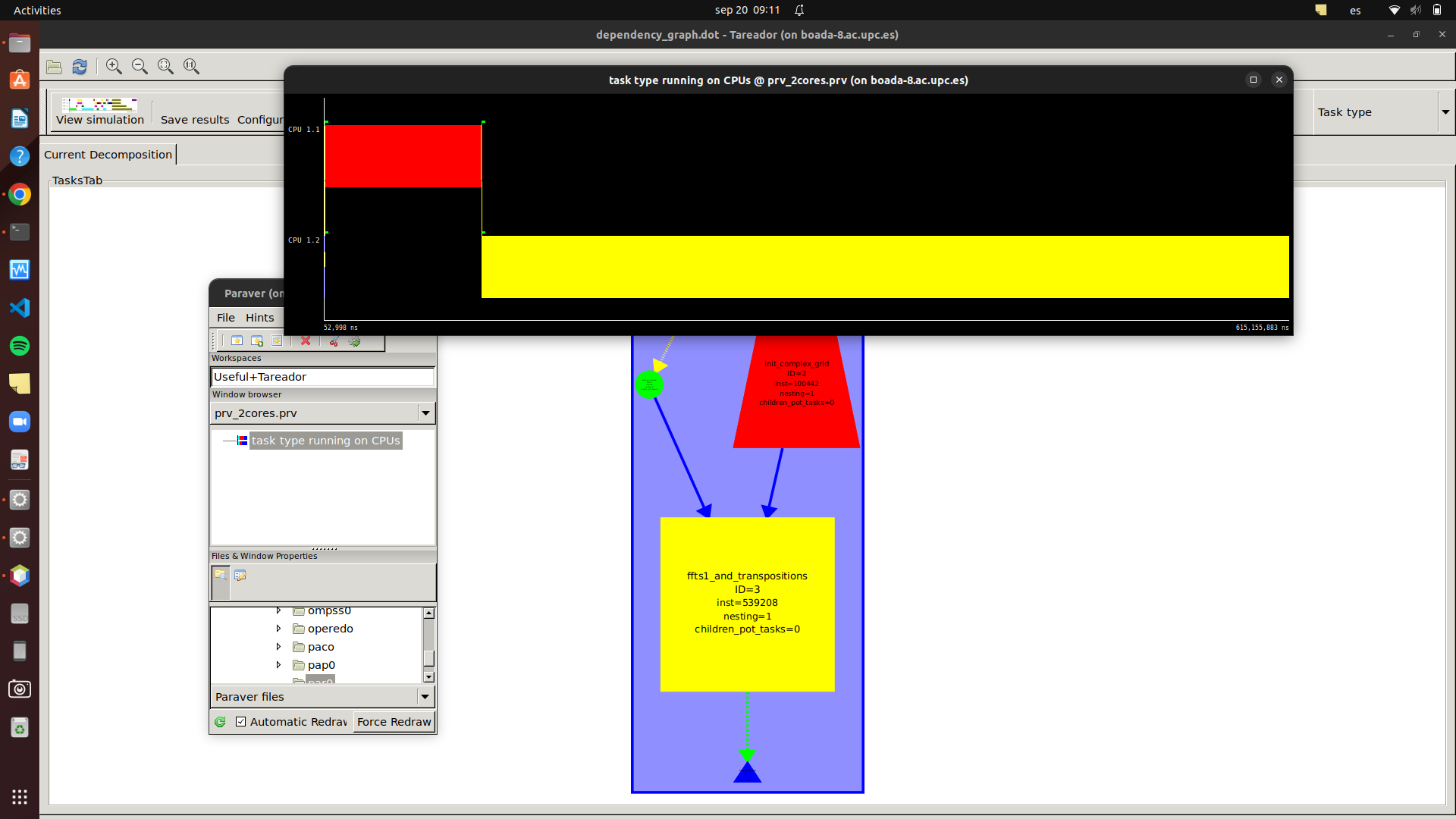
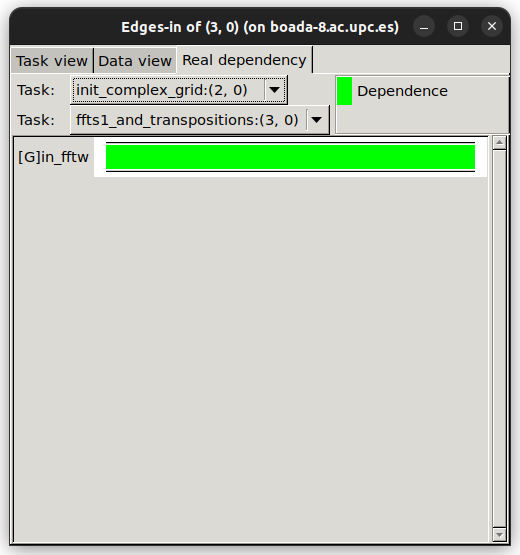
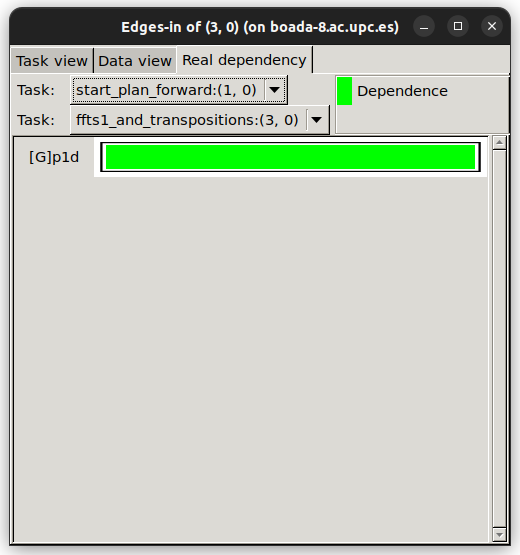
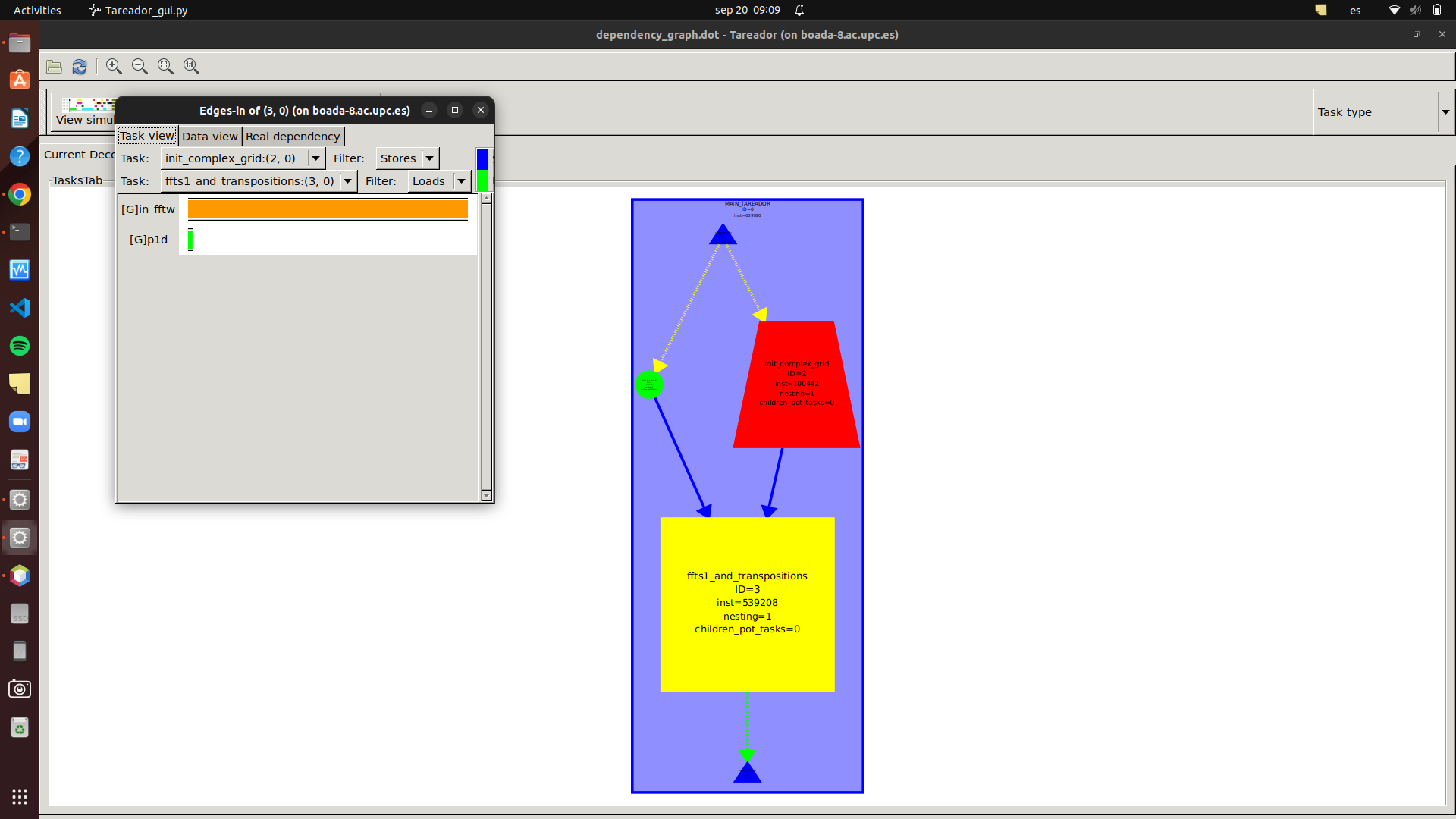
amb





**Interactive: Timing information Queued: Timing information**

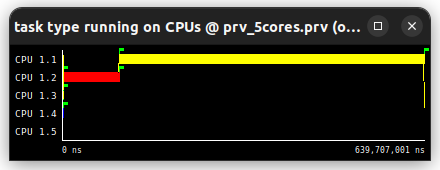
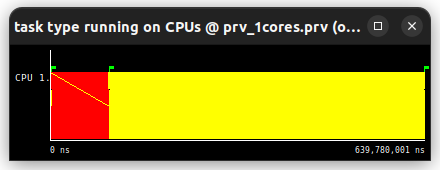
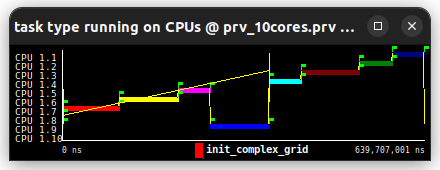
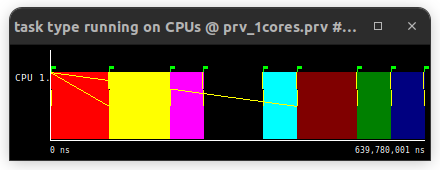
| **#threads** | **user** | **syst** | **elapsed** | **%CPU** | **user** | **syst** | **elapsed** | **%CPU** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** |  |  |  |  |  |  |  |  |
| **4** |  |  |  |  |  |  |  |  |
| **8** |  |  |  |  |  |  |  |  |
| **16** |  |  |  |  |  |  |  |  |
| **20** |  |  |  |  |  |  |  |  |

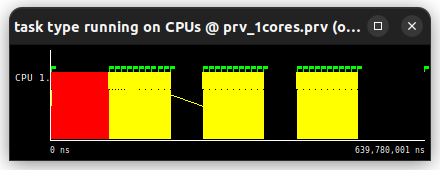
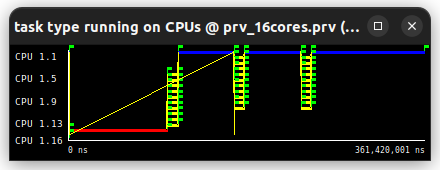


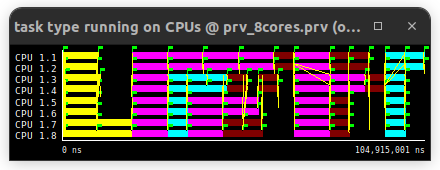
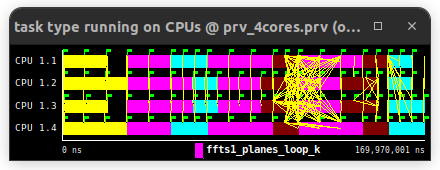
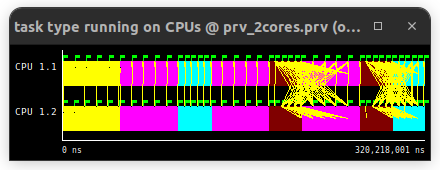
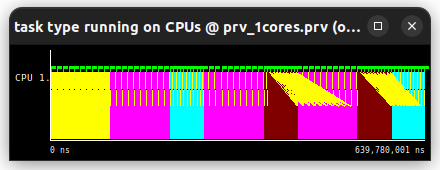
Although not useful for this code, you could disable the analysis of certain variables in the program using the following functions in the Tareador API: *tareador\_disable\_object(&name\_var)*

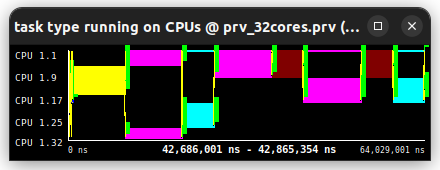
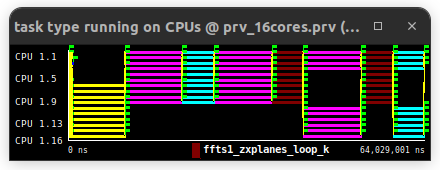
*// ... code region with memory accesses to variable name\_var tareador\_enable\_object(&name\_var)*

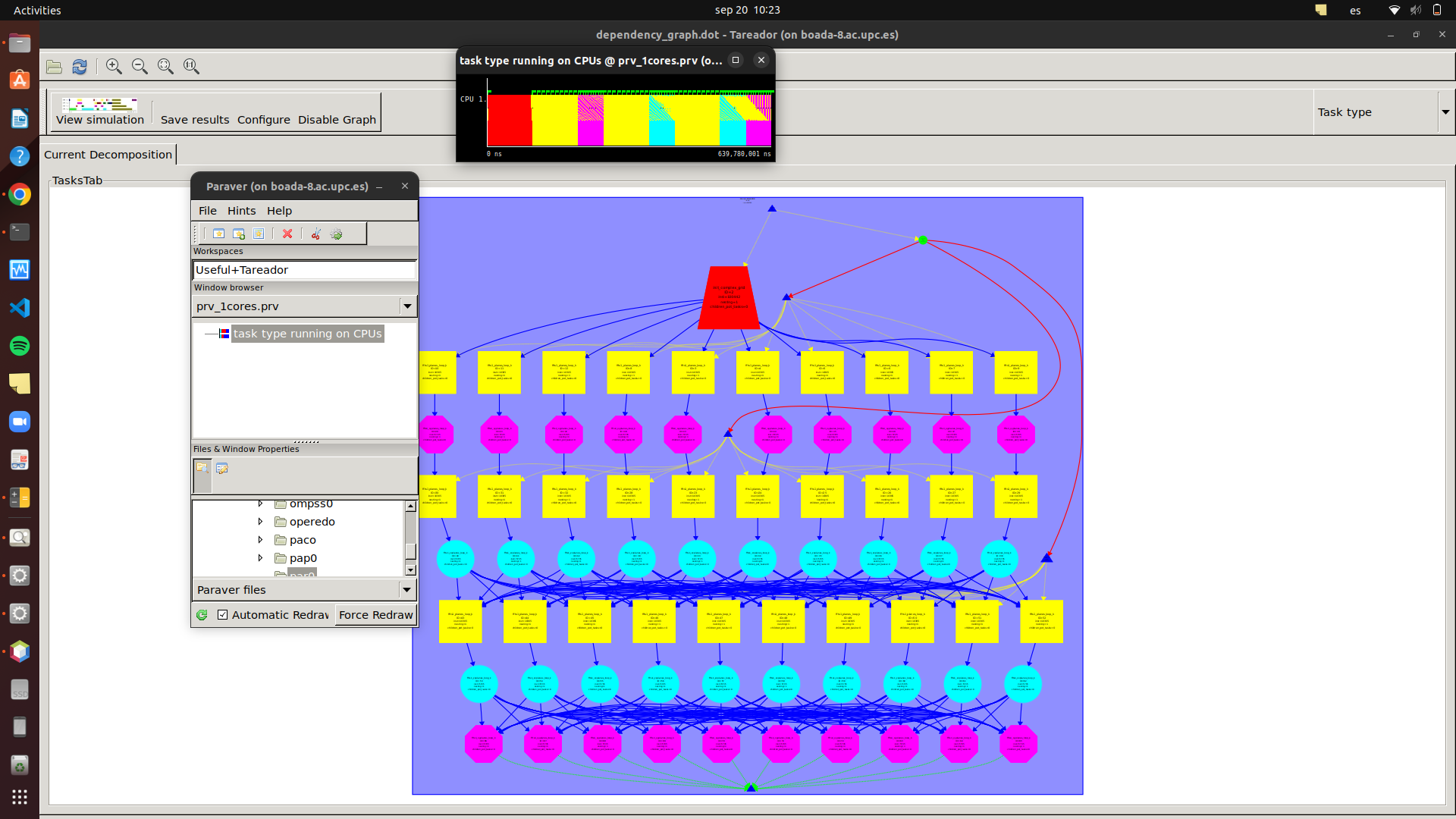
| Version | T1 | T∞ | Parallelism |
| --- | --- | --- | --- |
| seq | 639.780.001 ns | 639.707.001 ns | 1.000114115 |
| V1 | 639.780.001 ns | 639.707.001 ns | 1.000114115 |
| V2 | 639.780.001 ns | 361.420.001 ns | 1,770184271 |
| V3 | 639.780.001 ns | 154.354.001 ns | 4,144887705 |
| V4 | 639.780.001 ns | 64.029.001 ns | 9,992034719 |
| V5 |  |  |  |

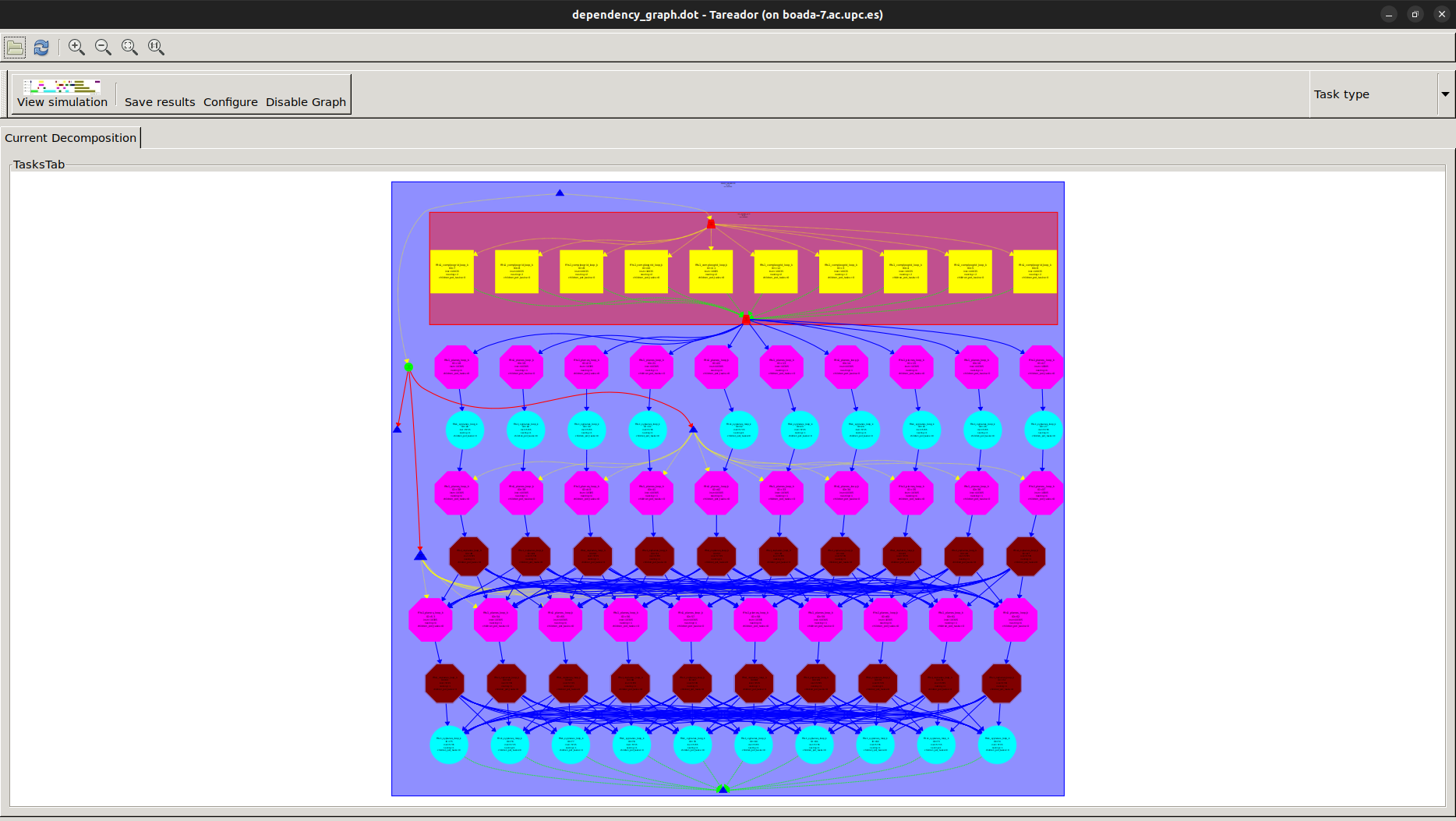
****









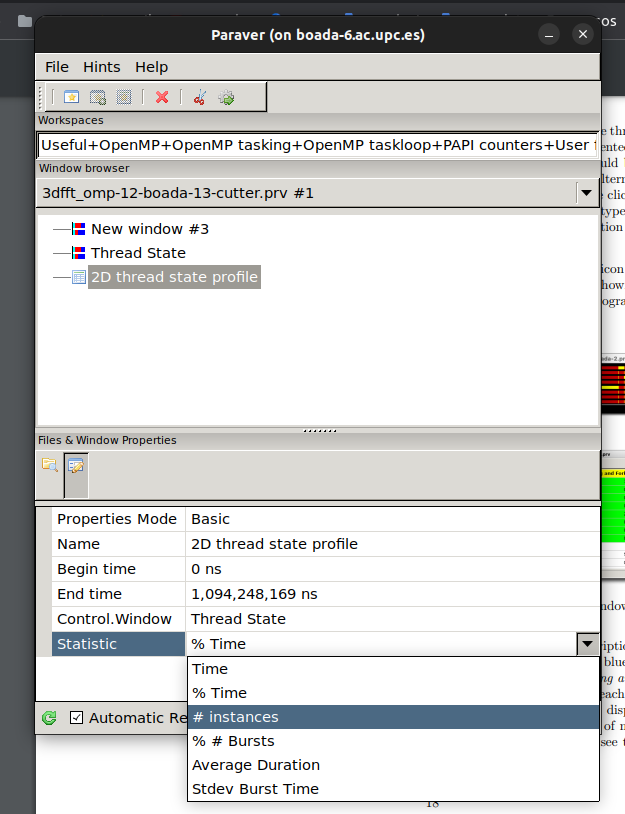
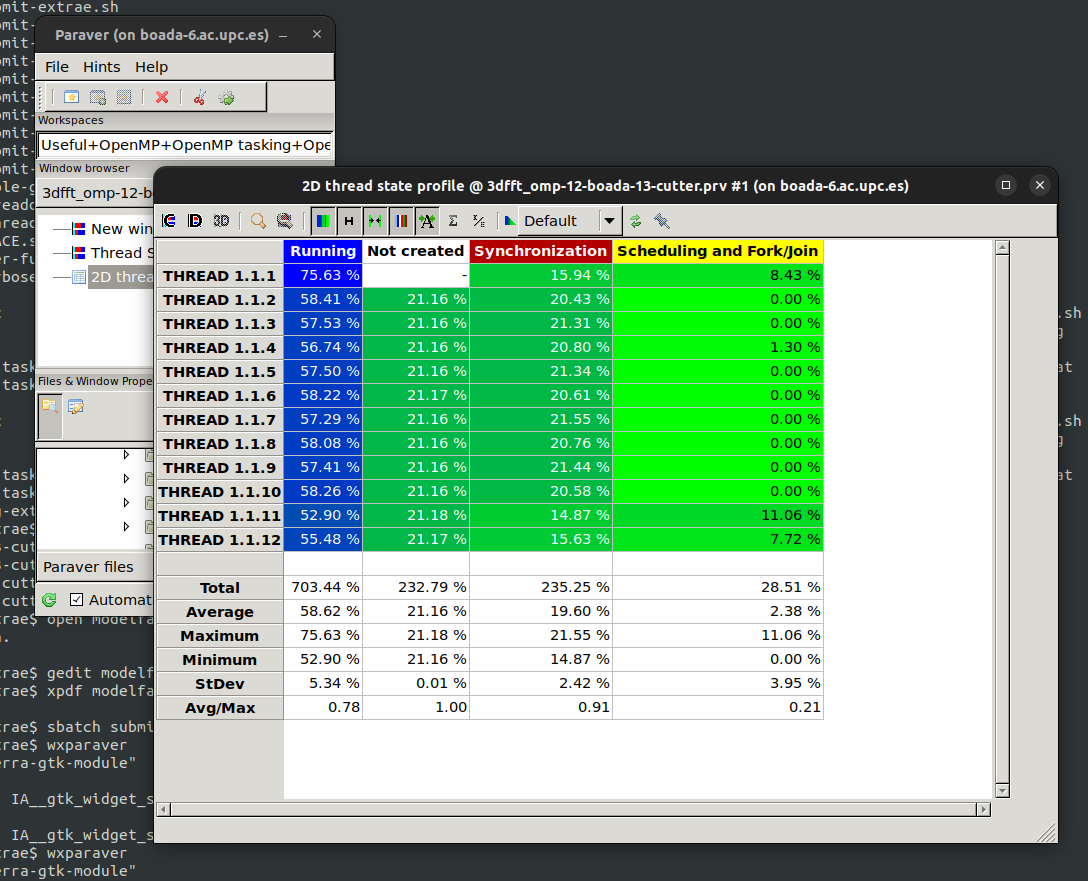
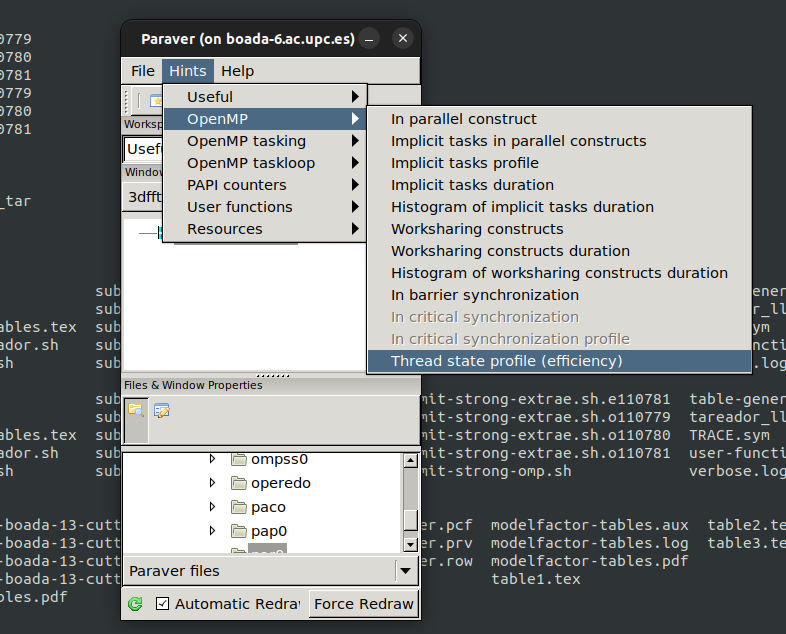
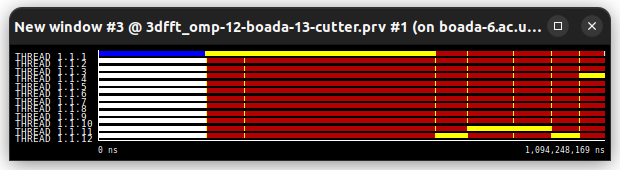
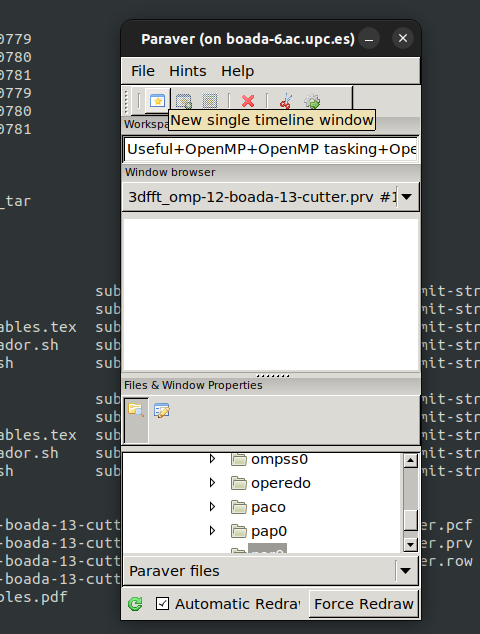
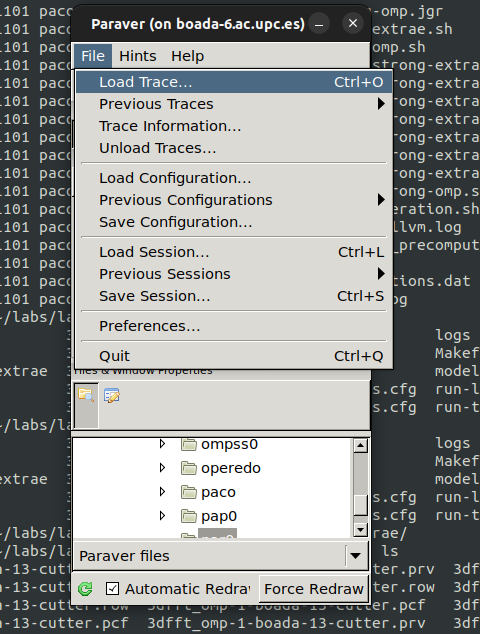


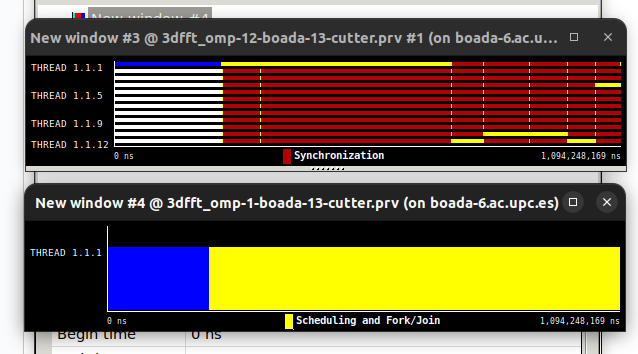
Hem compilat el document 3dfft\_omp.c amb el make

Hem fet un submit amb el Sbatch: *sbatch submit-strong-extrae.sh 3dfft\_omp*

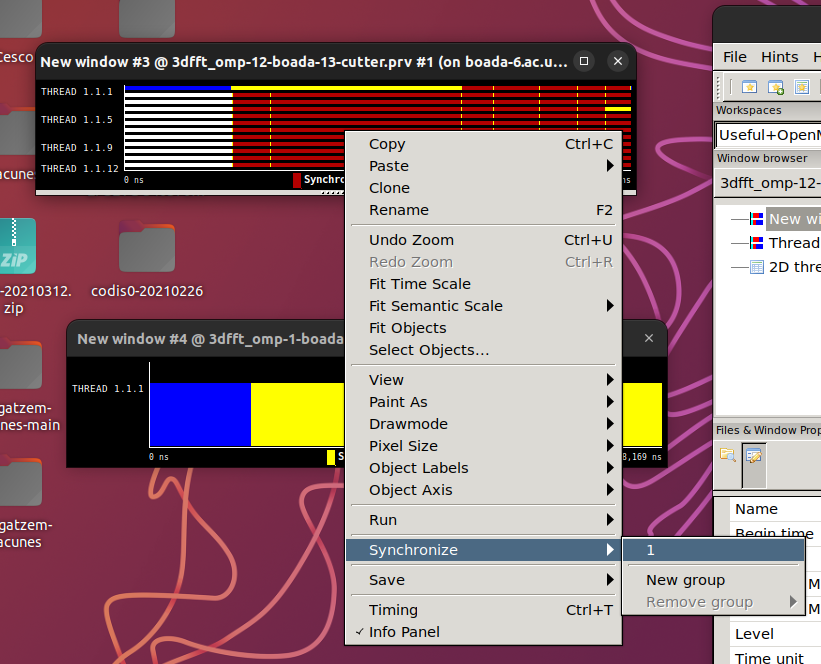
Això ens ha generat un directory: **3dfft\_omp-strong-extrae**

Hem obert el pdf dins el directory amb *xpdf modelfactor-tables.pdf*

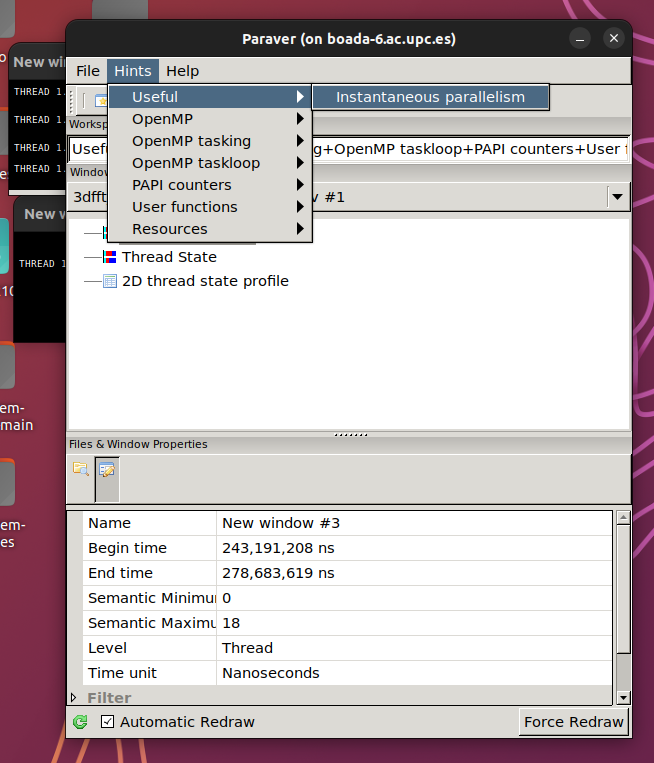


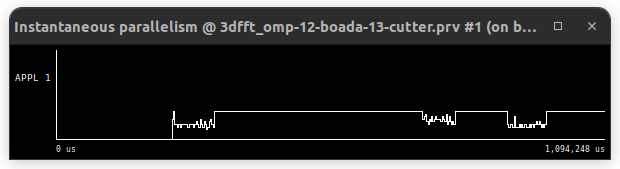


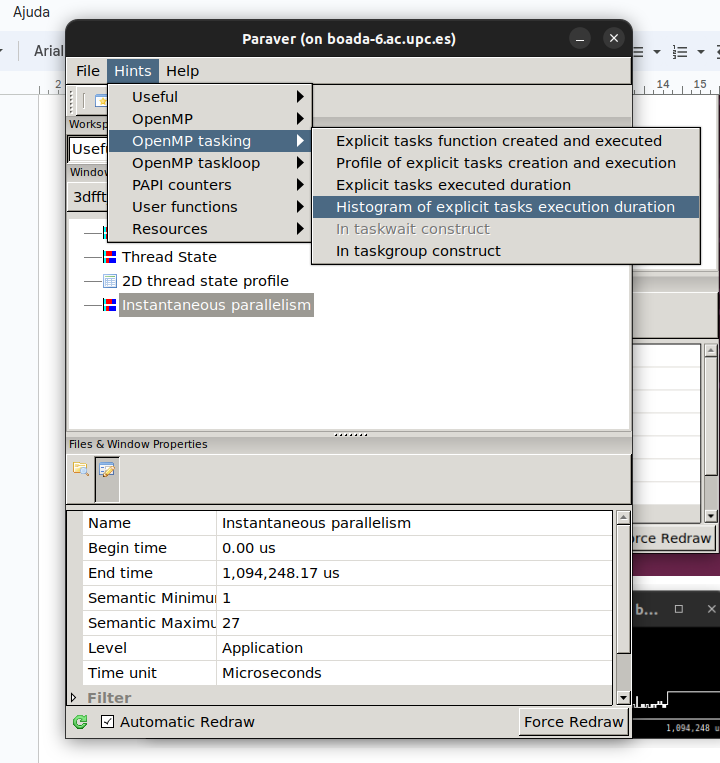
Si tenim 2 traces i volem veure la comparació en el temps hem de fer copy (botó dret) en una trace i paste default a l’altre.

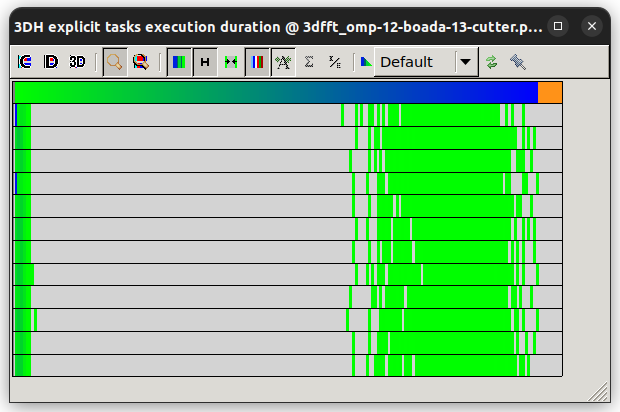


Hem de fer synchronize 1 a tots dos traces i si fem zoom a una farem a l’altre.

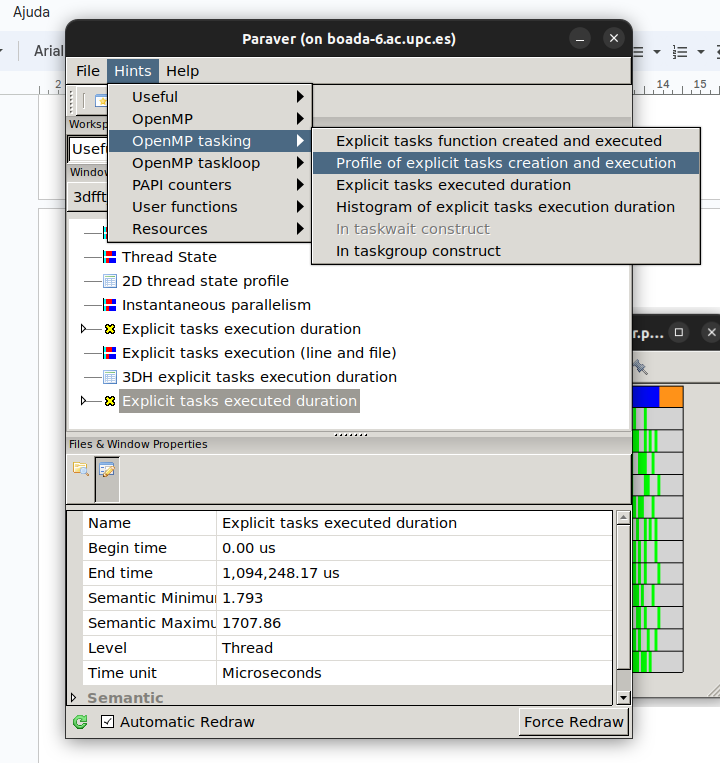


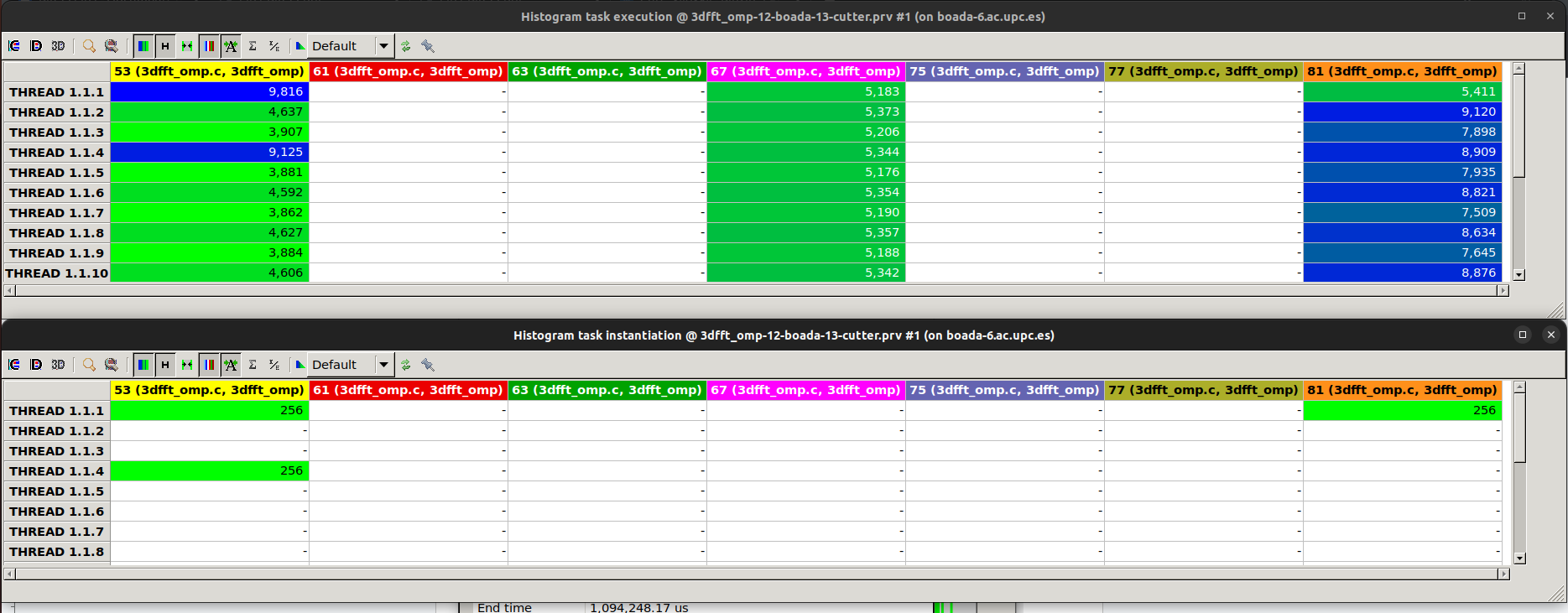






Lo de color verd són les tasques.



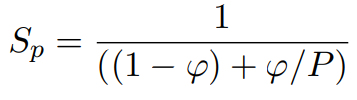


A la linia 53 passa 256 tasques i crea totes las tasques que es veuen a dalt. (La primera info. la veiem a la taula d’abaix, i la segona a la d’adalt).

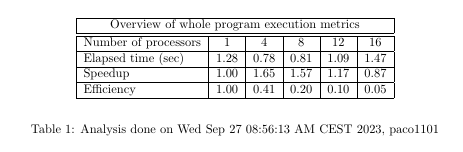
| **Version** | **φ (%)** | **ideal S12** | **T1 (ns)** | **T12 (ns)** | **real S12** |
| --- | --- | --- | --- | --- | --- |
| **initial version in 3dfft omp.c** | 82.99 | 10,008 | 12602315055 | 1050192921 | 1,05 |
| **new version with reduced parallelisation overheads** | 83,18 | 9,9816 | 2211838714 | 184319892,8 | 3,08 |
| **final version with improved φ** | 99,96 | 11,9952 | 2880363240 | 240030270 | 5,54 |

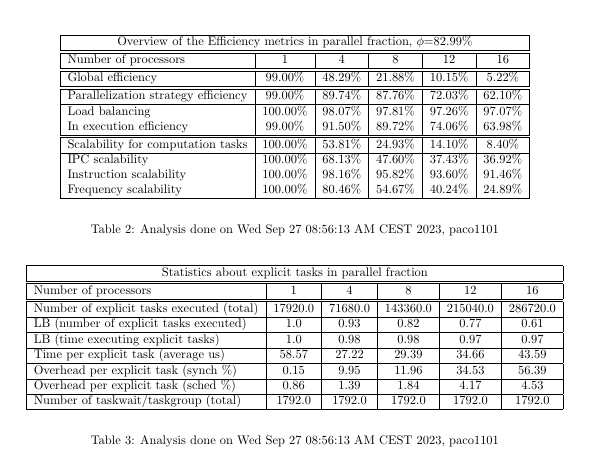
El ideal vol dir que tota la part paralitzada la dividim en 12 parts.

φ = Tpar/T1



Sp = T1 ÷ Tp





Com comparar traces:

fer copy - paste i synchronize