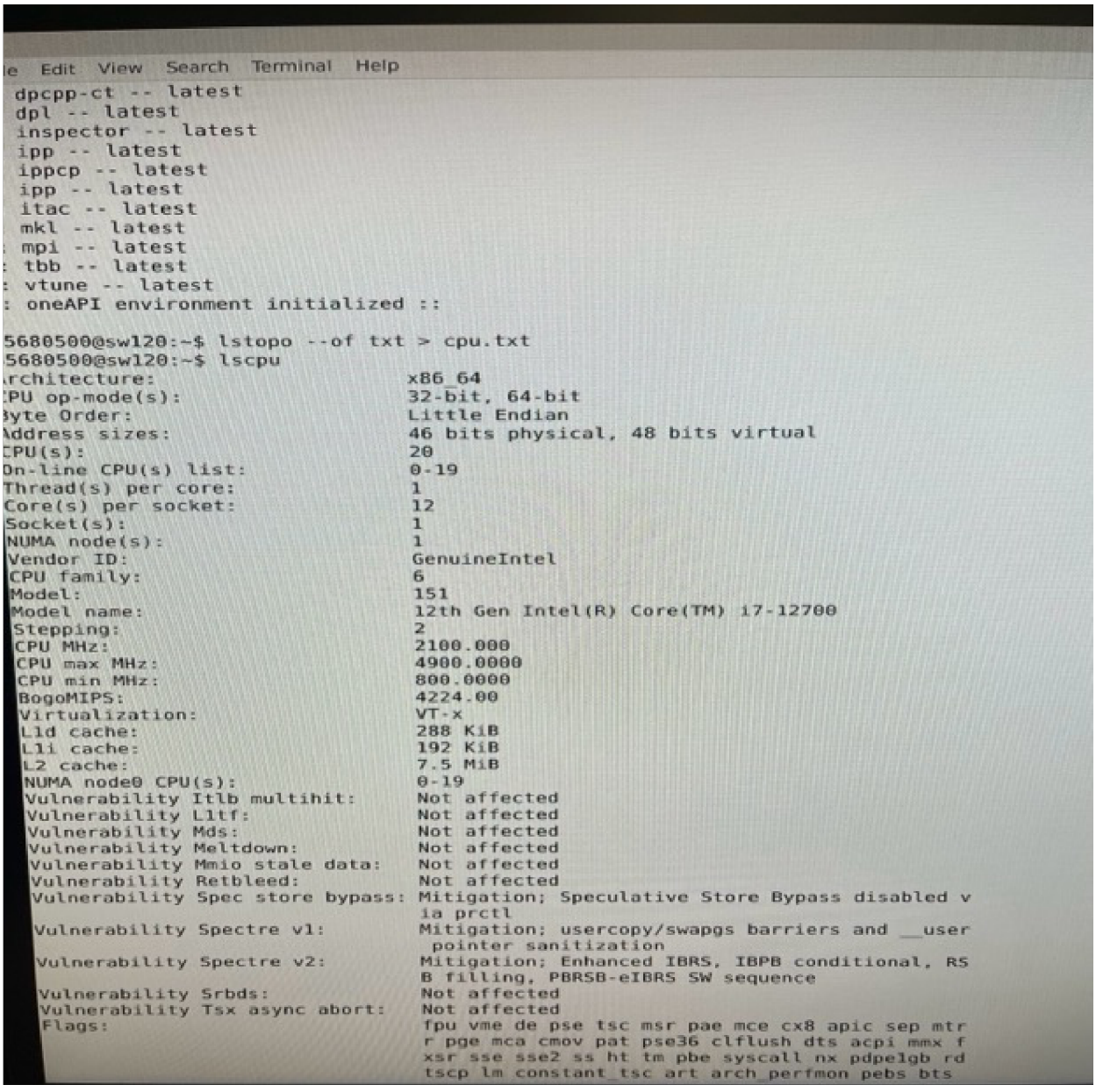
Quali sono le caratteristiche della nostra CPU?

CORE:

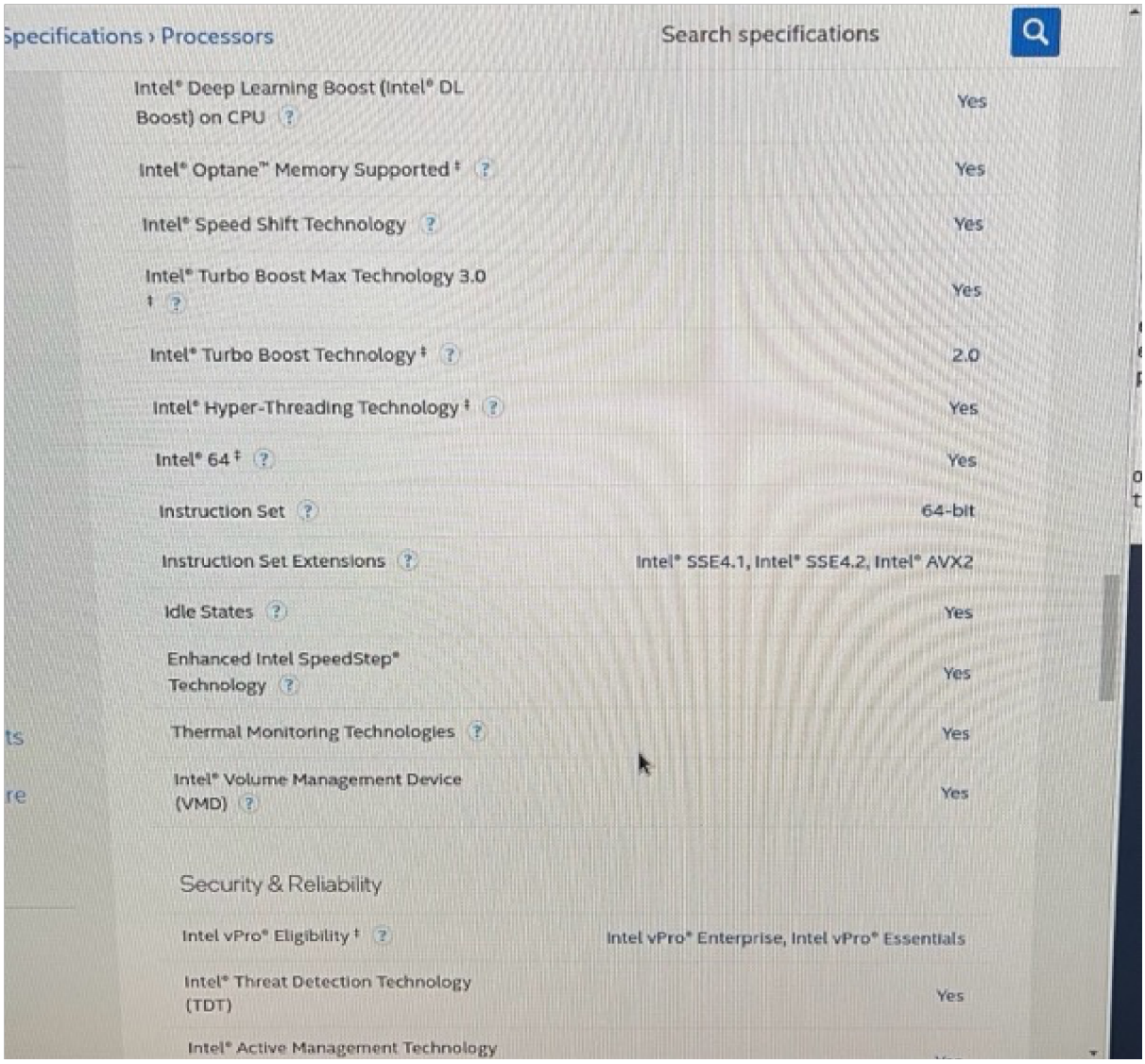
Thread(s) per core: 1
Core(s) per socket: 12

CACHE:

L1d cache: 288 KiB
L1i cache: 192 KiB
L2 cache 7.5 MiB



Nome architettura: intel SSE4.1, intel SSE4.2, intel AVX2



I flag di ottimizzazione:

Con :

```
gcc Driver.c Multiply.c  -o Mult -lm
```

Execution time = 80.866 secondi
GFlops: 0.999206

Con l'ottimizzazione ottengo:

```
gcc -O3 Driver.c Multiply.c  -o Mult -lm
```

Execution time = 18.676 secondi
GFlops: 4.326590

Con:

```
icc -diag-disable=10441 Driver.c Multiply.c  -o Mult
```

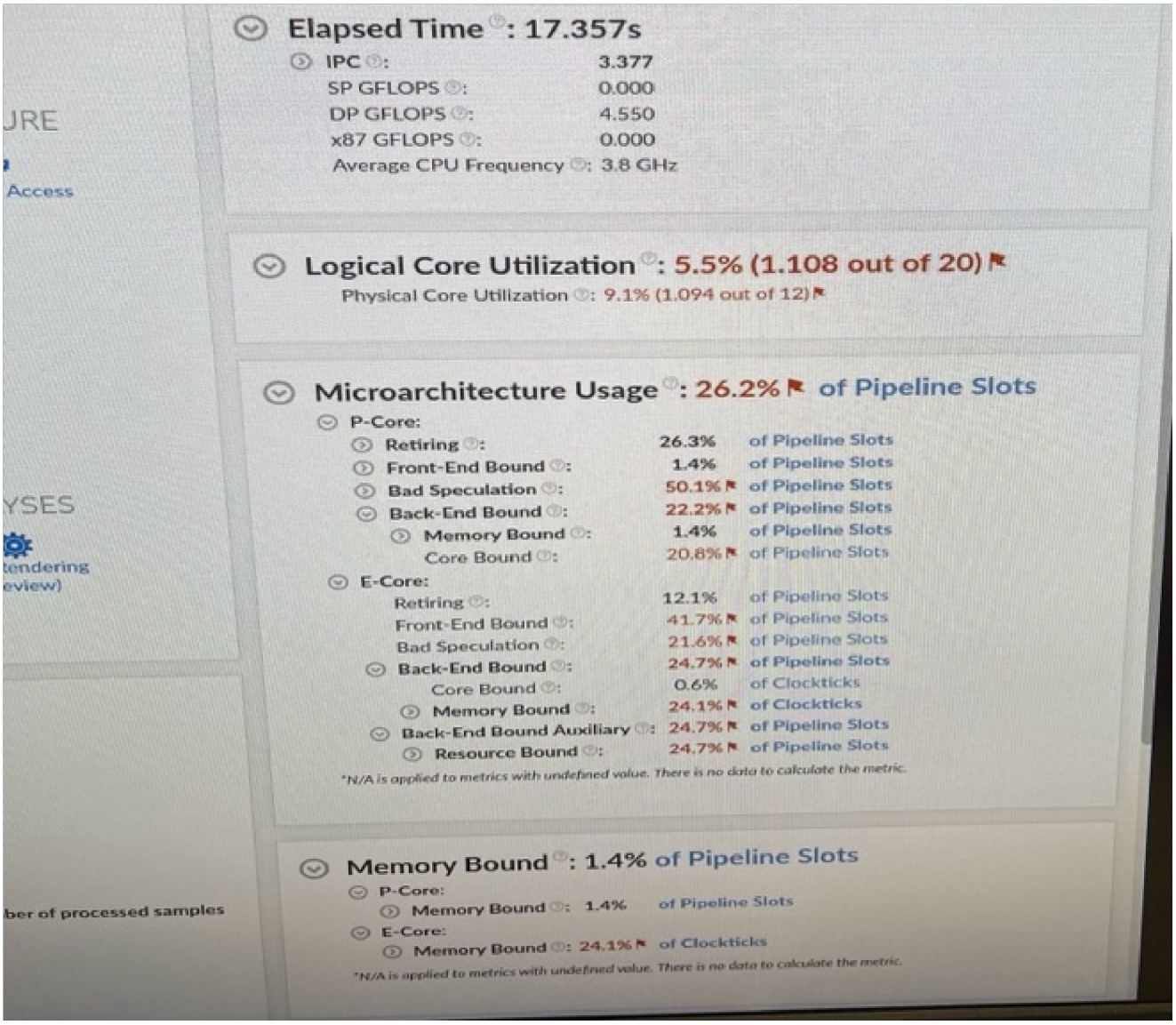
Execution time = 16.856 secondi
GFlops: 4.793709

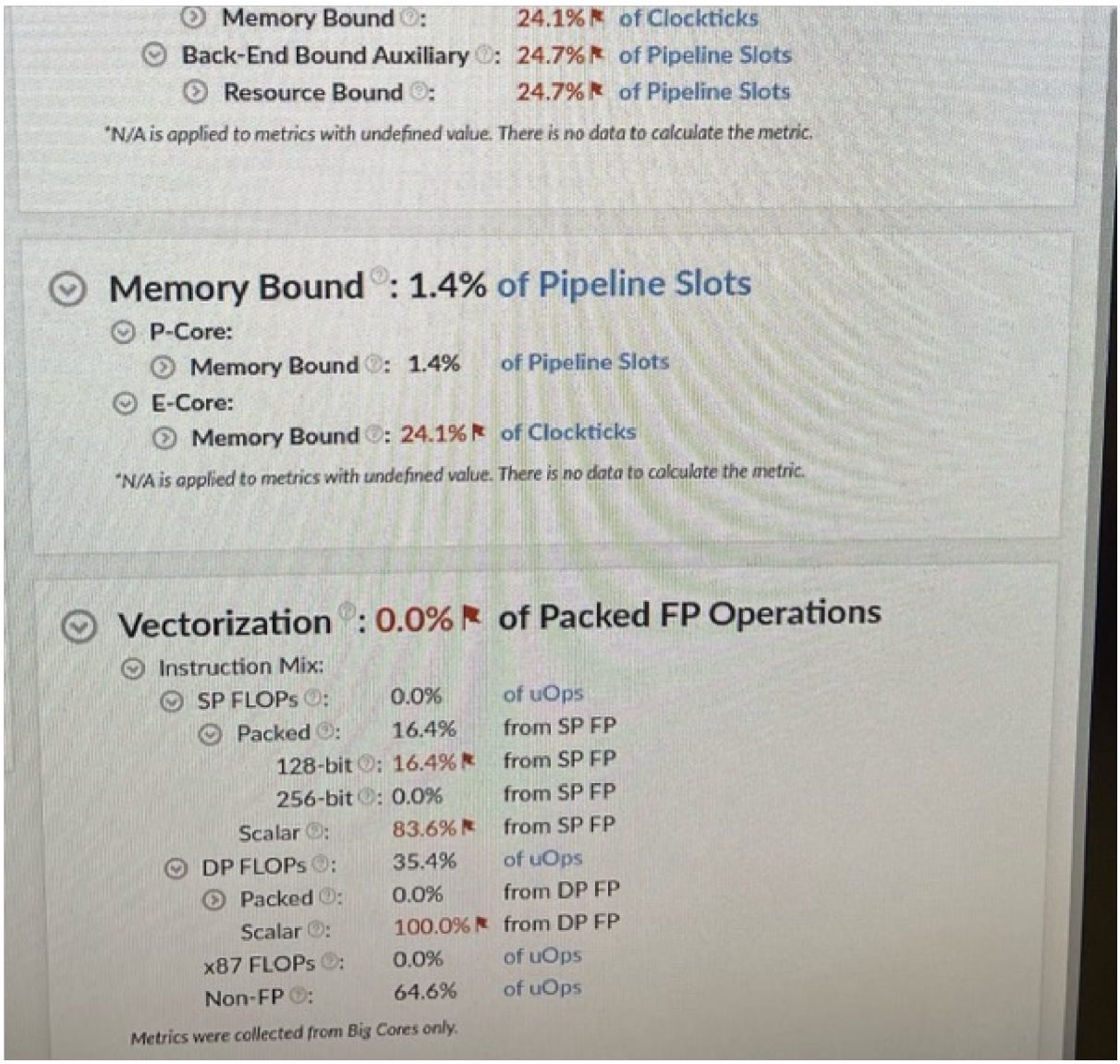
Il compilatore fa diverse ottimizzazioni legate all'architettura, La vettorizzazione in questo caso non riesce.

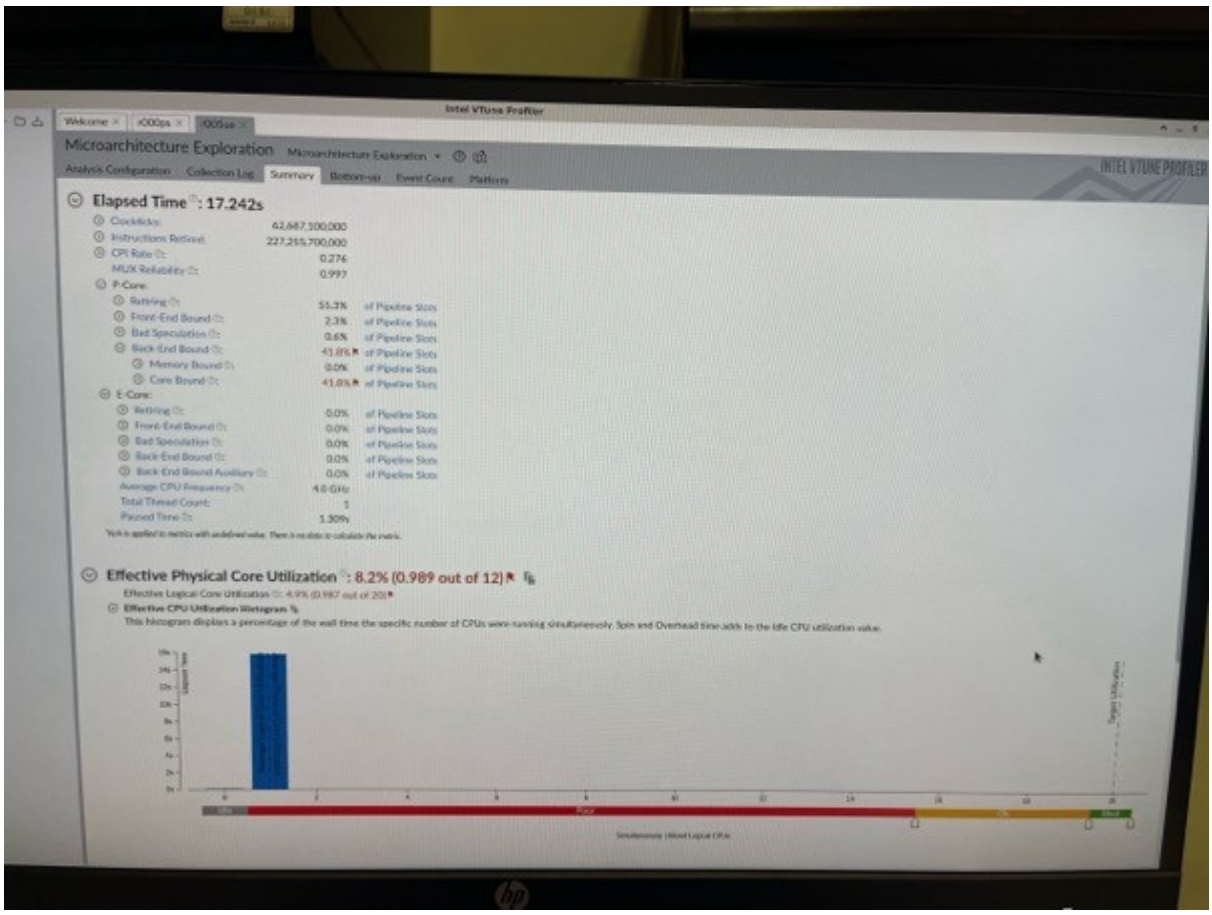
Analisi con i tool intel:

Il programma usa solo un thread sui 20 disponibili e per questo motivo non è ePiciente per quanto riguarda il parallelismo.

Il compilatore non è riuscito a fare la vettorizzazione.
Durante l'esecuzione possiamo osservare non vengono utilizzati gli E-core ma solamente i Pcore.
Per la speculazione utilizziamo soltanto il risultato del 55% delle istruzioni eseguite in quanto le altre 45 in quanto le altre non dovevano essere eseguire secondo la logica del programma.


Elapsed Time: 17.357s
IPC: 3.377
SP GFLOPS: 0.000
DP GFLOPS: 4.550
x87 GFLOPS: 0.000
Average CPU Frequency: 3.8 GHz
Logical Core Utilization: 5.5% (1.108 out of 20)
Physical Core Utilization: 9.1% (1.094 out of 12)
Microarchitecture Usage: 26.2% of Pipeline Slots
P-Core:
Retiring: 26.3% of Pipeline Slots
Front-End Bound: 1.4% of Pipeline Slots
Bad Speculation: 50.1% of Pipeline Slots
Back-End Bound: 22.2% of Pipeline Slots
Memory Bound: 1.4% of Pipeline Slots
Core Bound: 20.8% of Pipeline Slots
E-Core:
Retiring: 12.1% of Pipeline Slots
Front-End Bound: 41.7% of Pipeline Slots
Bad Speculation: 21.6% of Pipeline Slots
Back-End Bound: 24.7% of Pipeline Slots
Core Bound: 0.6% of Clockticks
Memory Bound: 24.1% of Clockticks
Back-End Bound Auxiliary: 24.7% of Pipeline Slots
Resource Bound: 24.7% of Pipeline Slots
*N/A is applied to metrics with undefined value. There is no data to calculate the metric.
Memory Bound: 1.4% of Pipeline Slots
P-Core:
Memory Bound: 1.4% of Pipeline Slots
E-Core:
Memory Bound: 24.1% of Clockticks
*N/A is applied to metrics with undefined value. There is no data to calculate the metric.
ber of processed samples

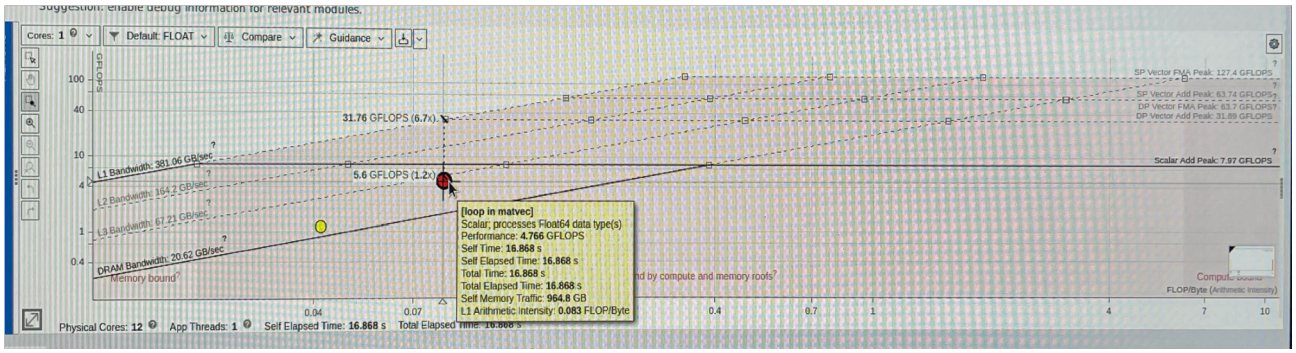

Memory Bound : 24.1% of Clockticks
Back-End Bound Auxiliary : 24.7% of Pipeline Slots
Resource Bound : 24.7% of Pipeline Slots
*N/A is applied to metrics with undefined value. There is no data to calculate the metric.
Memory Bound : 1.4% of Pipeline Slots
P-Core:
Memory Bound : 1.4% of Pipeline Slots
E-Core:
Memory Bound : 24.1% of Clockticks
*N/A is applied to metrics with undefined value. There is no data to calculate the metric.
Vectorization : 0.0% of Packed FP Operations
Instruction Mix:
SP FLOPs : 0.0% of uOps
Packed : 16.4% from SP FP
128-bit : 16.4% from SP FP
256-bit : 0.0% from SP FP
Scalar : 83.6% from SP FP
DP FLOPs : 35.4% of uOps
Packed : 0.0% from DP FP
Scalar : 100.0% from DP FP
x87 FLOPs : 0.0% of uOps
Non-FP : 64.6% of uOps
Metrics were collected from Big Cores only.



La memoryBound è pari allo 0% in quanto non ha carichi di istruzioni.

analisi del modello roofline

Abbiamo un programma non vettoriale ma scalare che sarebbero una somma di numeri in una matrice 201 x 201 .
Stiamo usando la memoria L3 con una performance di 4.766 Glflops in 16 secondi . Non stiamo usando tutte le CPU e stiamo utilizzando un solo core.



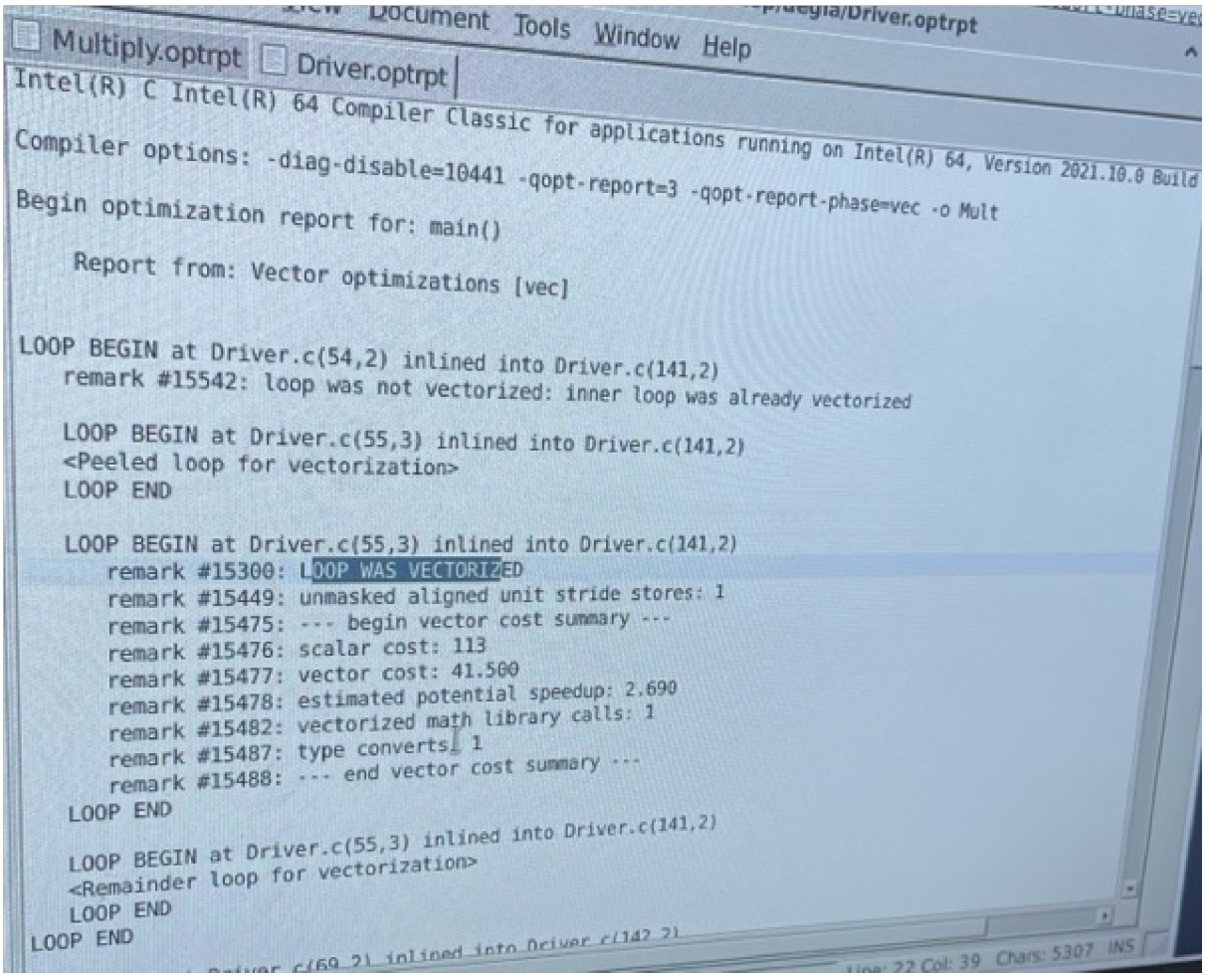
Vettorizziamo

Possiamo osservare che con la vettorizzazione il loop del file driver.c era già stato vettorizzato mentre quello delle moltiplicazioni(Multiply.c) ce una dipedenza RAW che non permette la vettorizzazione.

Con:

```
icc -diag-disable=10441 -qopt-report=3 -qopt-report-phase=vec  Driver.c Multiply.c
```

```
il tempo di esecuzione è 16.830 secondi
gigaFlops = 4.801001
```



Guardando il Driver.optrpt osserviamo che tutti i loop sono stati vettorizzati

Con:

```
icc -diag-disable=10441 -qopt-report=3 -qopt-report-phase=vec -DNOFUNCCALL
Driver.c Multiply.c  -o Mult
```

```
il tempo di esecuzione è 5.766 secondi
gigaFlops = 14.014247
```

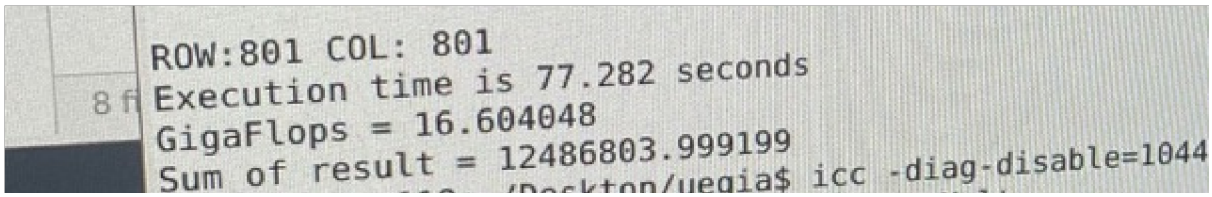
Con:

```
icc -diag-disable=10441 -qopt-report=3 -qopt-report-phase=vec -DNOFUNCCALL
march=alderlake Driver.c Multiply.c  -o Mult
```

```
il tempo di esecuzione è 4.056 secondi
gigaFlops = 19.922084
```

Parallelizzazione

Il tempo di esecuzione è di 77.782
GigaFlops = 16.604048



Dopo la modifica:

```
#ifdef NOFUNCCALL
        int i, j;
          #pragma omp parallel for private(j)
          for (i = 0; i < size1; i++) {
```

Il tempo di esecuzione è di 24.630
GigaFlops = 52.098944