

Exercici 1

Al realitzar execucions de les dues diferents versions de la multiplicació de matrius (tamany 1000) podeu veure variabilitat i que la segona versió obté pitjors resultats.

#execució	mm (s)	mm2 (s)
1	11,23	15,78
2	11,25	12,77
3	11,24	15,76
4	11,27	12,87

Exercici 1.1

L'execució de flops i flops2 mostra diferències com es mostra a continuació:

	flops	flops2
real_time	11,2617	12,7162
proc_time	11,2351	12,6854
MFLOPS	178,2516	157,9695

La primera versió proporciona més MFLOPS ja que la segona ha de dedicar més temps al moviment de dades a través de la jerarquia de memòria. Com que les matrius s'emmagatzemen de forma seqüencial de forma unidimensional, és d'esperar que l'accés a posicions de memòria de forma ordenada proporcionaran millor rendiment.

Exercici 1.2

Alguns comptadors significatius son aquells relacionats amb la utilització de la jerarquia de memòria com ara la fallades de cache. Tres exemples son:

PAPI_LST_INS → per a mesurar el número total d'instruccions de lectura/escriptura

PAPI_L1_DCM → per a mesurar les fallades de cache de nivell 1

PAPI_L2_DCM → per a mesurar les fallades de cache de nivell 2

Exercici 1.3

A continuació es mostren possibles resultats on es veu que la variabilitat principal es troba en les fallades de cache:

counters

counters2

PAPI_TOT_INS	33.074.987.583	33.074.989.019
PAPI_FP_OPS	9219	9219
PAPI_L1_DCM	1.873.629.237	3.043.753.161
PAPI_L2_DCM	73.446.810	232.259.159
PAPI_LST_INS	15.043.028.868	15.043.031.965

Exercici 1.4

La implementació amb OpenMP la podeu realitzar utilitzant les tècniques que veu fer servir a la PAC2, per exemple el bucle d'inicialització es possible que feu servir

```
#pragma omp parallel for
for(i=0...)
```

i podeu fer servir paral·lisme al bucle extern de la part de còmput o fins i tot fer servir directives com ara collapse:

```
#pragma omp parallel for collapse(3)
for(i=0...)
```

Amb la versió OpenMP, uns possibles resultats son els següents, on s'observen els mateixos patrons:

	counters (omp)	counters2 (omp)
exec time	4,48	6,88
PAPI_TOT_INS	11.022.923.445	11.023.486.554
PAPI_FP_OPS	9219	9219
PAPI_L1_DCM	477.951.858	868.433.699
PAPI_L2_DCM	19.341.759	307.680.785
PAPI_LST_INS	5.763.670.624	5.763.788.544

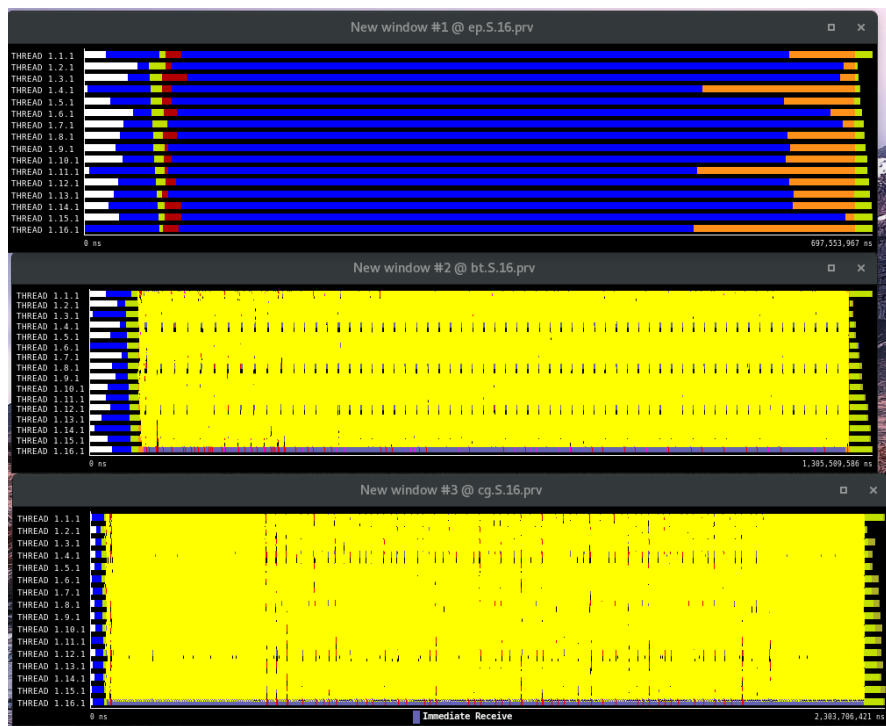
Exercici 2

Al vídeo publicat es va fer una demostració de d'utilització de les eines i de les principals diferències entre les dues implementacions del solver amb MPI (una amb crides síncrones i una altra amb crides asíncrones) fent servir TAU/jumpshot.

Exercici 3

Hi ha diferències importants entre els tres benchmakrs seleccionats, especialment en el patró de pas de missatges. Alguns exemples es mostren a continuació.

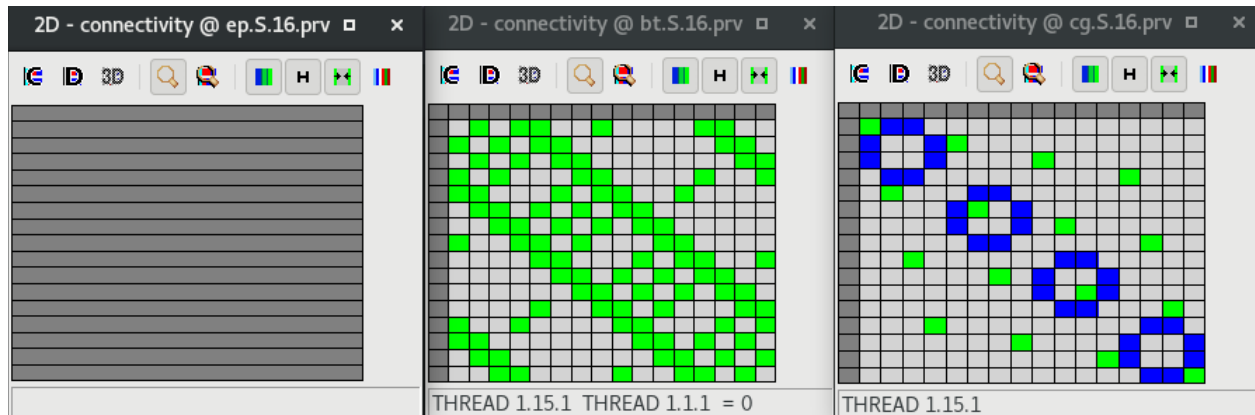
Vista general:



Crides MPI:



Patrons de pas de missatges, és interessant veure el pas de missatges entre nodes (més freqüent al BT) i dins d'un mateix node (al CG).



Crides col·lectives:

Time all in collective calls @ ep.S.16...			Time all in collective calls @ bt.S.16.prv					Time all in collective calls @ cg.S.16...		
	MPI_Barrier	MPI_Allreduce		MPI_Bcast	MPI_Barrier	MPI_Reduce	MPI_Allreduce		MPI_Barrier	MPI_Reduce
APPL 1	0.59	1.00	APPL 1	0.94	0.80	1	0.65	APPL 1	0.77	0.03
Total	0.59	1.00	Total	0.94	0.80	1	0.65	Total	0.77	0.03
Average	0.59	1.00	Average	0.94	0.80	1	0.65	Average	0.77	0.03
Maximum	0.59	1.00	Maximum	0.94	0.80	1	0.65	Maximum	0.77	0.03
Minimum	0.59	1.00	Minimum	0.94	0.80	1	0.65	Minimum	0.77	0.03
StDev	0	0	StDev	0	0	0	0	StDev	0	0
Avg/Max	1	1	Avg/Max	1	1	1	1	Avg/Max	1	1