

# Paral·lelisme

## Lab0

Segon entregable

Gerard Bayego

Martín Dans

Grup: par3101

Curs: 2014 - 15

Quadrimestre Tardor

### Pregunta 1:

L'orde de magnitud és de microsegons.

El temps no és constant, a mesura que anem augmentant els threads el temps de fork/join augmenta però el temps per thread disminueix.

Nthr	Time	Time per thread
2	2.0951	1.0475
3	1.9277	0.6426
4	2.5313	0.6328
5	2.5640	0.5128
6	2.6359	0.4393
7	2.9219	0.4174
8	3.4159	0.4270
9	3.4050	0.3783
10	3.5888	0.3589
11	3.5134	0.3194
12	3.5915	0.2993
13	4.1774	0.3213
14	4.1261	0.2947
15	4.3284	0.2886
16	4.6505	0.2907
17	4.4514	0.2618
18	4.7961	0.2665

### Pregunta 2:

L'ordre de magnitud és de milisegons. Augmenta exponencialment, ja que en afegir un processador més el temps de sincronització entre ells augmenta. Això passa perquè les instruccions que fan els processadors per saber si poden entrar a la zona crítica impedeix que el processador que intenta alliberar la zona crítica accedeixi immediatament i per tant, trigui molt més a poder fer-ho. Alhora implica fer molts més accessos a una adreça de memòria, que òbviament ha d'estar compartida entre tots els processadors. Per últim, hi ha dins del critical instruccions innecessàries, ja que la multiplicació es pot fer a fora i només volem fer la lectura i la següent escriptura després de la suma amb critical.

### Pregunta 3:

L'orde de magnitud és de milisegons. Aquest es comporta igual que l'anterior però triga menys. Això és perquè només s'estalvia temps d'operació però igualment es produeix aquest overhead de comunicació entre processadors.

### Pregunta 4:

Augmenta uns 0,5 milisegons, ja que el vector on emmagatzemem la informació de cada processador està a la mateixa línia de cache i això fa que cada cop que algú escriu, es posi invàlid la línia a les altres caches i s'hagi de tornar a fer el share. Si fem que les dades del vector estiguin separades entre elles de manera que cada una estigui al principi d'una línia de cache, ja no tindrem el problema. Així doncs farem una matriu d'amplada, amplada de la cache i alçada del nombre de processadors. Considerem que la dada que volem és la que es troba a la posició  $M[0][\text{num\_thread}]$ .

### Pregunta 5:

Versió	1 processador	8 processador	speed-up
pi_seq.c	0,8	-	1,00
pi_omp.c (sumlocal)	0,8	0,11	7,27
pi_omp_critical.c	1,83	12,22	0,14
pi_omp_atomic.c	1,45	8,42	0,17
pi_omp_sumvector.c	0,8	0,6	1,33
pi_omp_padding.c	0,8	0,11	7,27