Paral·lelisme

Lab0

Primer entregable

Gerard Bayego
Martín Dans

Grup: par3101
Curs: 2014-15
Quadrimestre Tardor

Pregunta 1:

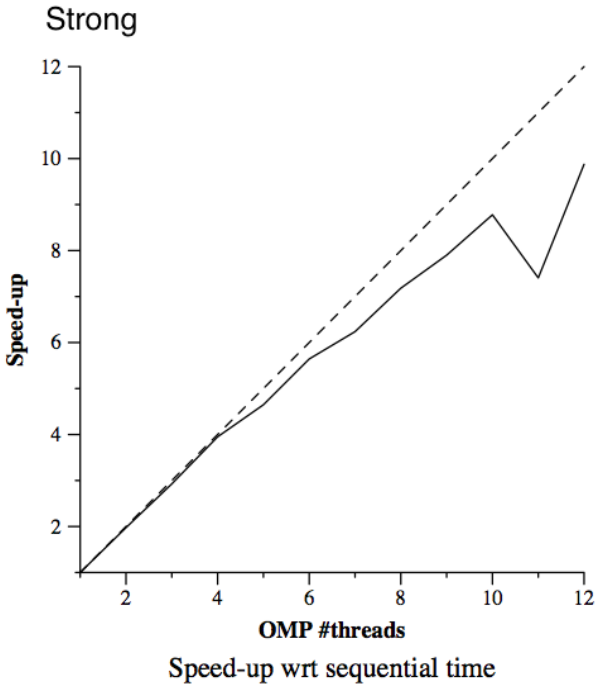
Nombre de sockets: 2
Cores per socket: 6
Threads per core: 2
Main memory: 12GB/NUMAnode
Cache: 12MB(shared) L3 + 256KB L2 + 32KB L1

Pregunta 2:

L'estructura timeval està declarada en la llibreria sys/time.h.
Conté:

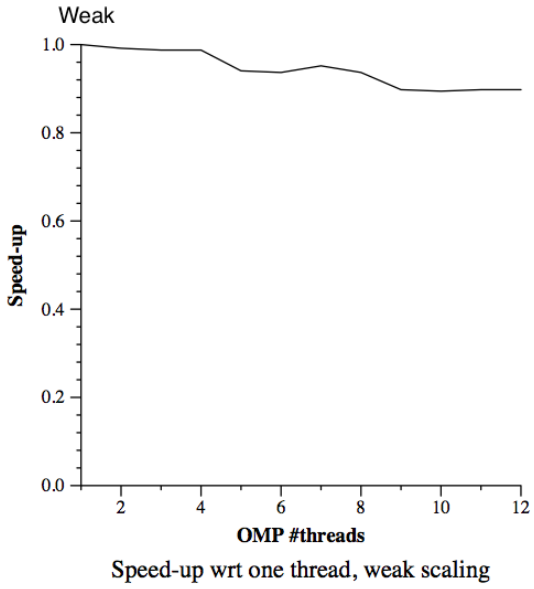
- long int tv_sec : Representa el nombre segons del temps.
- long int tv_usec: Representa el nombre de microsegons del temps.

Pregunta 3:



Strong scalability:

S'observa com en augmentar el nombre de processadors disminueix quasi proporcionalment el temps d'execució , de fet, al principi es pràcticament proporcional i a mesura que anem augmentant el nombre de processadors va disminuint el speed up que obtenim. Proporcionalmente seria la línia discontinua de la gràfica.



Weak scalability:

S'observa com en augmentar la mida del problema, si augmentem el nombre de processadors de forma proporcional veiem com l'exeució del programa quasi triga el mateix temps, ja que la línia de la gràfica s'apropa molt a una línia horitzontal que té com a speed up sempre 1.

Pregunta 4:
La regió paral·lela del pi_seq és 96.72%.

Pregunta 5:

	Running	Synchronization	Scheduling and Fork/Join
THREAD 1.1.1	95.35 %	4.65 %	0.00 %
THREAD 1.1.2	99.84 %	0.16 %	-
THREAD 1.1.3	99.92 %	0.08 %	-
THREAD 1.1.4	97.36 %	2.64 %	-
THREAD 1.1.5	96.22 %	3.78 %	-
THREAD 1.1.6	99.94 %	0.06 %	-
THREAD 1.1.7	99.86 %	0.14 %	-
THREAD 1.1.8	96.40 %	3.60 %	-
Total	784.91 %	15.09 %	0.00 %
Average	98.11 %	1.89 %	0.00 %
Maximum	99.94 %	4.65 %	0.00 %
Minimum	95.35 %	0.06 %	0.00 %
StDev	1.85 %	1.85 %	0 %
Avg/Max	0.98	0.41	1

Pregunta 6:

Paral·lel:

Core	MIPS	Instuccions
1	3349.62	362506787
2	2633.71	361085195
3	2631.99	361085195
4	2687.85	361085195
5	2713.91	361085195
6	2630.75	361085195
7	2633.44	361085195
8	2709.70	361085195

Sequencial:

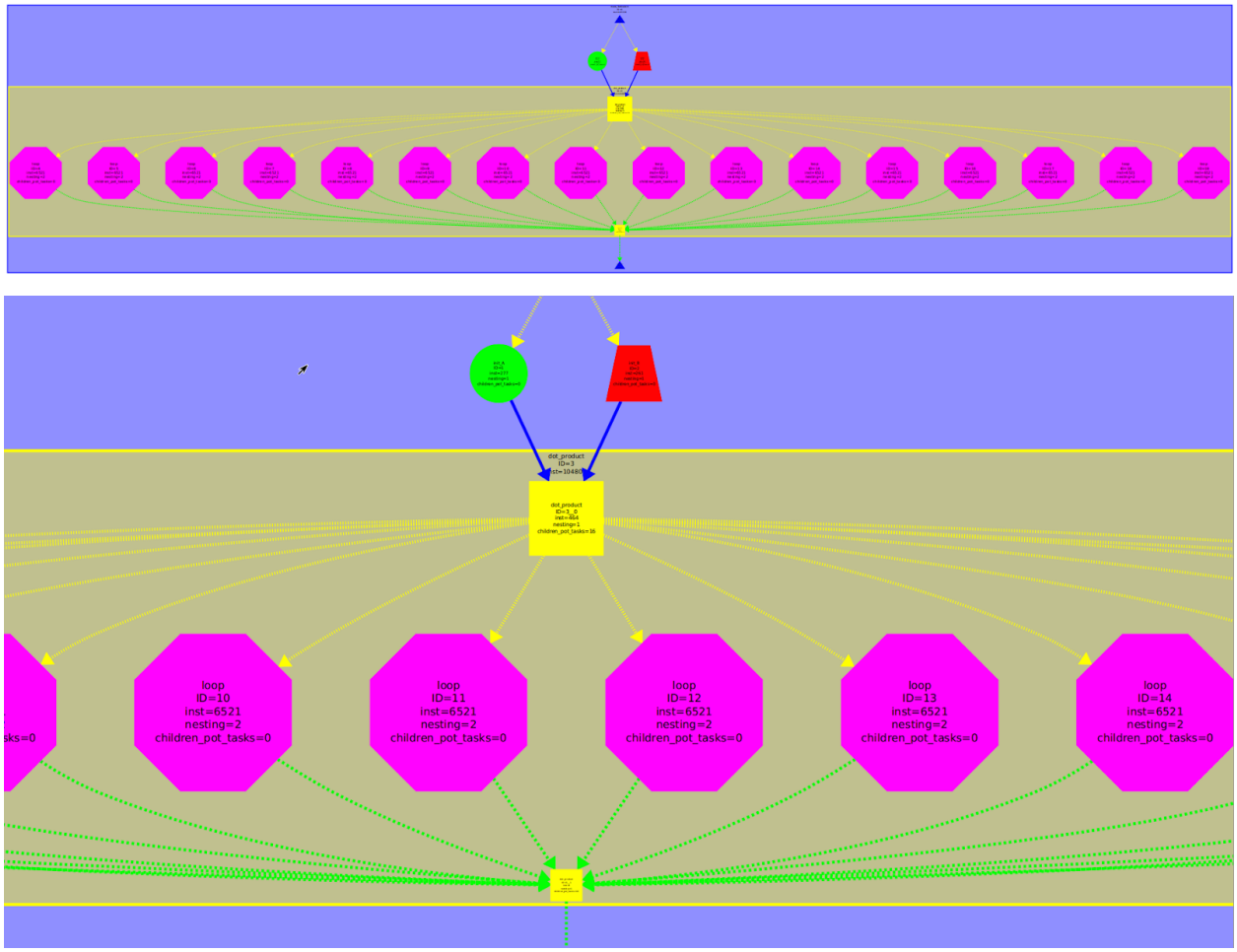
Core	MIPS	Instuccions
1	3608.55	2900048828

Si, són coherents, tots els processadors en la regió paral·lela executen el mateix nombre d'instruccions i van més o menys a la mateixa velocitat. Això té sentit ja que es reparteix la feina equitativament, si es pot, per tal de reduir el temps d'execució al mínim, alhora és lògic que els MIPS siguin semblants ja que estem treballant dins d'una mateixa màquina. Les instruccions del seqüencial són una mica menys que la suma de totes les instruccions del paral·lel. Això és degut a que el paral·lel ha d'executar a part de les instruccions del programa les instruccions per fer el paral·lelisme.

Pregunta 7:

Codi font adjunt (dot_product.c)

Pregunta 8:



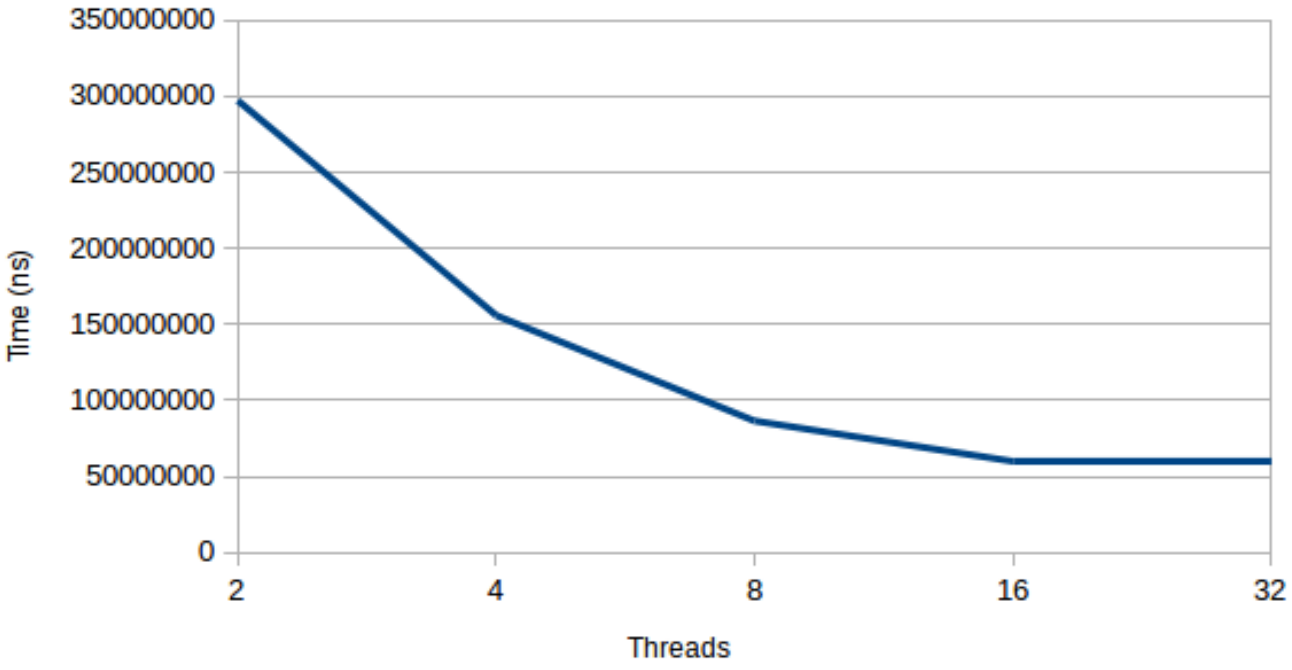
Pregunta 9:

Version	T_1	T_∞	Paralelisme
Seq	593762	593762	1
V1	593762	593747	1.000025
V2	593762	315188	1.883834
V3	593762	108336	5.480745
V4	593762	59412	9.993974

Pregunta 10:

Nota: En les dues gràffiques l'eix x (on hi han representats els Threads), estàn amb escala exponencial.

Gràfica de temps d'execució (ns) respecte el nombre de threads.



Gràfica de speedup de la versió v4 respecte l'inicial (seq).