

Práctica 2. Cálculo de pi

1. Error medio, por defecto y por exceso

Intervalo	Error medio (%)	Error por defecto (%)	Error por exceso (%)
100	0,00026525823845	0,31777936970689	0,31884040266069
1000	0,00000265258227	0,03182568345355	0,03183629378323
10000	0,00000002652607	0,00318304581062	0,00318315191306
100000	0,00000000026637	0,00031830935650	0,00031831041587
1000000	0,00000000000092	0,00003183098861	0,00003183098862
10000000	0,00000000000198	0,00000318310503	0,00000318309269
100000000	0,000000000002016	0,00000031832364	0,00000031829614
1000000000	0,00000000000565	0,00000003182671	0,00000003183527
10000000000	0,00000000000684	0,00000002257682	0,00000002257141
100000000000	0,00000000000068	0,00000002618651	0,00000002617775
1000000000000	0,00000000000116	0,00000002419171	0,00000002417870
10000000000000	0,00000000000336	0,00000011513809	0,00000011514809

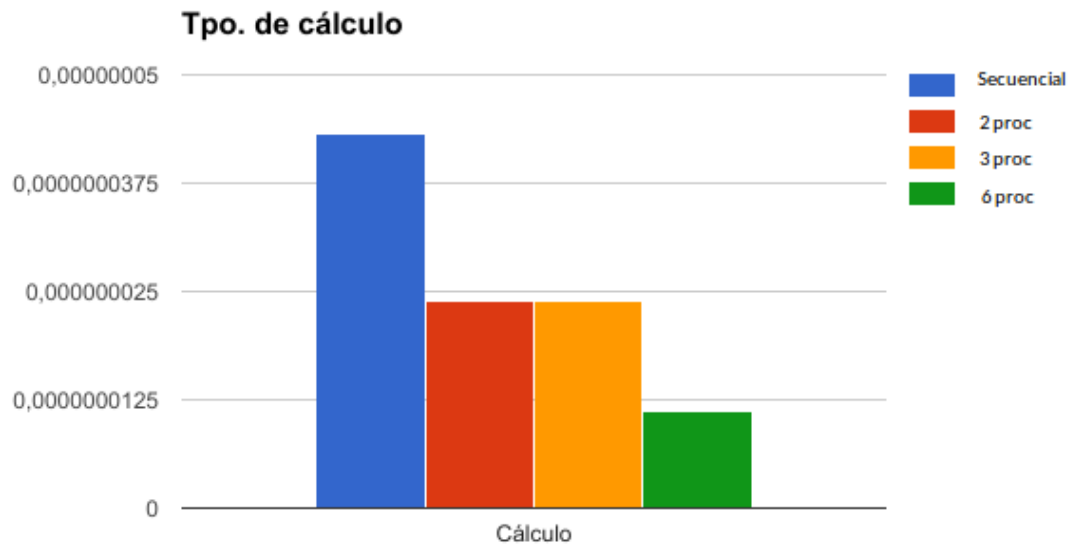
En esta tabla observamos que, como era de esperar, el porcentaje de error va disminuyendo conforme aumentamos el número de intervalos, si bien es cierto que hay algunos valores más altos que los previos en la tabla, pero estos son tan pequeños que no resultan significativos a simple vista.

2. Tiempo de ejecución y ganancia para un intervalo de 10000

Tiempos (s)	Secuencial	S(sec/sec)	2 procesadores	S(sec/2p)	3 procesadores	S(sec/3p)	6 procesadores	S(sec/6p)
Creación	0,0000136728	1	0,000016443	0,8315270936	0,0000229039	0,5969638359	0,000027828	0,4913324709
Cálculo	0,0000000432	1	0,0000000238	1,81512605	0,0000000238	1,81512605	0,0000000112	3,857142857
Recogida	0,0000000019	1	0,0000000141	0,134751773	0,0000000632	0,03006329114	0,0000001619	0,01173563928
Destrucción	0,000003	1	0,000004	0,75	0,000004	0,75	0,000004	0,75

En esta tabla vemos que el tiempo de cálculo va disminuyendo al añadir más procesos, al igual que lo hacen los tiempos de recogida y de destrucción, lo cual es normal puesto que tenemos que aunar más datos y finalizar más procesos.

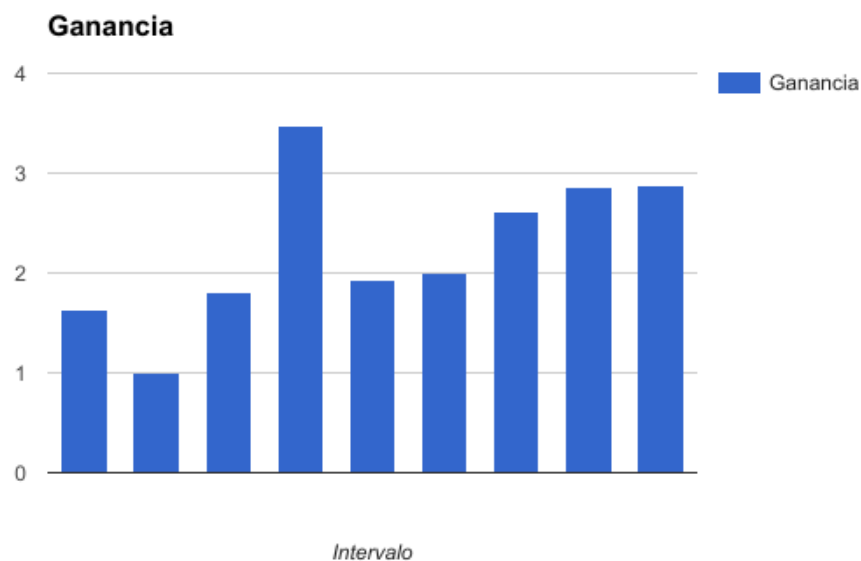
3. Representación gráfica del tiempo de cálculo



Aquí se observa lo que se decía en la tabla anterior, conforme más procesos, necesitamos menos tiempo de cálculo.

4. Representación gráfica de los datos con nivel 3 de paralelismo

Intervalo	Int. 100	Int. 1000	Int. 10000	Int. 100000	Int. 1000000	Int. 10000000	Int. 100000000	Int. 1000000000	Int. 10000000000
Error (%)	0,0002652582	0,0000026526	0,0000000265	0,0000000003	0	0	0	0	0
Tpo. creación	0,0000238521	0,0000264549	0,0000229039	0,0000240641	0,000024533	0,0000245781	0,0000294881	0,0000302041	0,0000244589
Tpo. cálculo	0,0000000019	0,0000000005	0,0000000238	0,0000002251	0,000002229	0,0000214021	0,0001661069	0,001489655	0,0020906949
Tpo. recogida	0,0000000432	0,0000000281	0,0000000632	0,0000001249	0,000000118	0,0000000298	0,0000000401	0,0000001202	0,000000735
Tpo. destrucción	0,000004	0,000004	0,000004	0,000004	0,000004	0,000004	0,000004	0,000004	0,000004
Ganancia	1,631578947	1	1,81512605	3,47401155	1,9269179	1,998822545	2,620004949	2,864304151	2,875731366



Vemos cómo aumenta la ganancia al aumentar el número de intervalos, puesto que el tiempo de cálculo será menor. Sin embargo vemos que esta ganancia se mantiene o deja de crecer cuando llegamos al intervalo 1000000000.

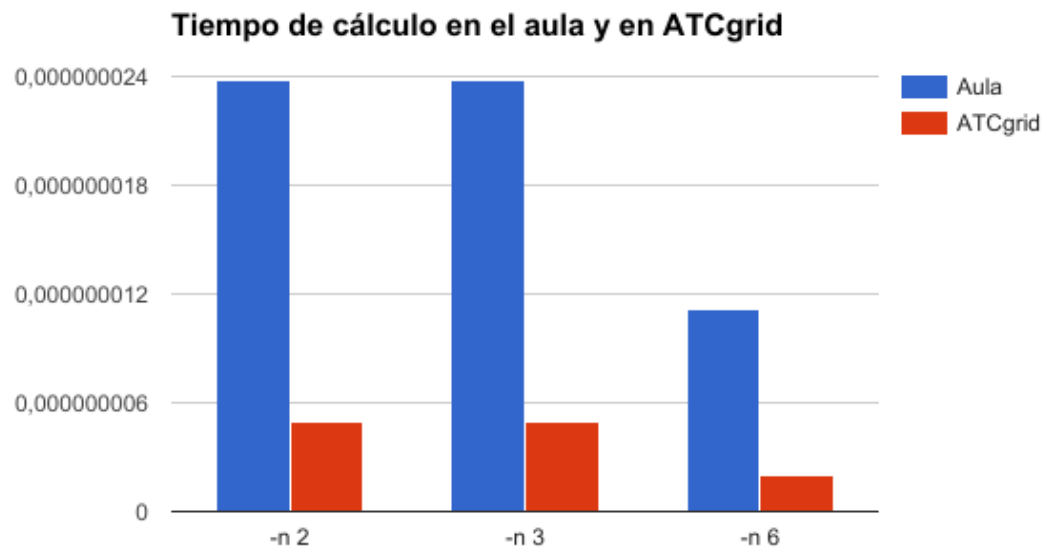


5. Ejecución en ATCgrid

	-n 2	-n 3	-n 6
Error (%)	0,0000026526	0,0000026526	0,0000026526
Tpo. creación	0,000129513	0,000135855	0,000148455
Tpo. cálculo	0,000000005	0,000000005	0,000000002
Tpo. recogida	0,000000031	0,000000114	0,000000054
Tpo. destrucción	0,000002	0,000003	0,000003

Estos son los datos que nos proporciona ATCgrid para los niveles 2, 3 y 6 de paralelismo.

Vemos ahora una gráfica comparando los tiempos de cálculo de los computadores de las aulas y ATCgrid.



Como era de esperar ATCgrid realiza los cálculos mucho más rápido que los computadores del aula.