

ACAP: PRÁCTICA 1

Paralelización del código

Para paralelizar el código lo que he hecho es:

- Función piLeibniz: he añadido a la función dos parámetros extras, uno el “rango” del proceso y otro el número total de procesos que se están utilizando para ejecutar paralelamente. En esta función, como es un sumatorio lo que hago es que, cada proceso, hace la suma de los términos del sumatorio cuyo índice es $\text{rango} + \text{numProceso} * i$, donde i es el número de iteración empezando en 0. De esta forma aprovechamos que la suma es una operación asociativa, y así cada proceso hace el mismo número de sumas, pudiendo variar el número de sumas en uno como mucho (porque puede ser que el número total de sumas no sea múltiplo del número de procesos). He tenido que ajustar dos parámetros, uno el numerador, ya que será uno o menos uno dependiendo de si el término de la suma es par o impar (si es par 1.0 y si es impar -1.0), y el denominador, pues hay que ajustarlo en función del término de la suma ($\text{denom} = 1.0 + \text{rango} * 2.0$). Además a la hora de actualizar el denominador hay que actualizarlo teniendo en cuenta que cada proceso hace unos términos concretos de la suma ($\text{denom} += \text{numProcesos} * 2.0$) y de igual forma hay que tener en cuenta que para actualizar el numerador hay que tener en cuenta el término de la suma.
- Función piRectangles: de nuevo le he añadido los dos parámetros, los mismo que en la función anterior. Y esta vez no ha habido que ajustar los parámetros, pues no dependen del término de la suma. Simplemente hay que modificar el bucle para que de nuevo cada proceso haga las sumas $\text{rango} + \text{numProceso} * i$, donde i es el número de la iteración.

Finalmente lo que se hace es utilizar la función MPI_Reduce de forma que todos los procesos le pasan su parte de la suma al proceso MASTER, para que esta haga la suma de los valores. Esta función se invoca después de cada función de cálculo, luego se invoca dos veces y se tiene en cuenta en el cálculo del tiempo. Finalmente se considera el tiempo del proceso MASTER, pues es el último en salir de la función MPI_Reduce, ya que es el que recoge el valor de la suma.

Ejecución Secuencial PC:

Tamaño	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Tiempo 5	Media	Desviación T
214748362	1,96	1,992	1,98	1,95	1,97	1,9704	0,0164560019
429496727	3,709	3,753	3,724	3,724	3,732	3,7284	0,0160717143
644245092	5,907	5,555	5,557	5,699	5,548	5,6532	0,1553035737
858993457	7,449	7,407	7,432	7,386	7,404	7,4156	0,0248455227
1073741822	9,405	9,225	9,24	9,241	9,244	9,271	0,0752695157
1288490187	11,14	11,028	11,118	11,08	11,108	11,0948	0,0431184415
1503238552	12,956	12,941	12,931	12,947	12,945	12,944	0,0091104336
1717986917	14,844	14,787	14,809	14,972	14,779	14,8382	0,0789221135
1932735282	16,709	16,648	16,625	16,624	16,645	16,6502	0,0346799654
2147483647	18,667	18,6	18,539	18,493	18,47	18,5538	0,0804468769

Ejecución Paralela con 2 cores PC:

Tamaño	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Tiempo 5	Media	Desviación T
214748362	0,929	0,937	0,922	0,935	0,913	0,9272	0,009859006
429496727	1,853	1,833	1,876	1,912	1,833	1,8614	0,0333811324
644245092	2,761	2,762	2,802	2,846	2,759	2,786	0,0380328805
858993457	3,655	3,666	3,676	3,763	3,668	3,6856	0,0439124128
1073741822	4,677	4,58	4,672	4,665	4,569	4,6326	0,0533507263
1288490187	5,601	5,503	5,517	5,652	5,508	5,5562	0,0668483358
1503238552	6,517	6,435	6,422	6,422	6,379	6,435	0,0504925737
1717986917	7,314	7,336	7,331	7,345	7,36	7,3372	0,0170205758
1932735282	8,238	8,247	8,204	8,215	8,235	8,2278	0,0177115781
2147483647	9,164	9,136	9,115	9,139	9,126	9,136	0,0182619824

Ejecución Paralela 3 cores PC:

Tamaño	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Tiempo 5	Media	Desviación T
214748362	0,649	0,626	0,626	0,65	0,627	0,6356	0,0127003937
429496727	1,296	1,228	1,244	1,342	1,233	1,2686	0,0491406146
644245092	2,016	1,834	1,869	1,888	1,876	1,8966	0,0697050931
858993457	2,574	2,453	2,478	2,513	2,466	2,4968	0,0485870353
1073741822	3,079	3,102	3,068	3,07	3,13	3,0898	0,0262144998
1288490187	3,704	3,701	3,707	3,83	3,739	3,7362	0,0546232551
1503238552	4,315	4,279	4,308	4,455	4,298	4,331	0,070629314
1717986917	4,915	5,007	4,925	5,074	4,944	4,973	0,0668318786
1932735282	5,525	5,502	5,66	5,856	5,555	5,6196	0,1453179273
2147483647	6,143	6,16	6,266	6,135	6,136	6,168	0,0556911124

Ejecución Paralela 4 cores PC:

Tamaño	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Tiempo 5	Media	Desviación T
214748362	0,56	0,537	0,487	0,515	0,47	0,5138	0,0364376179
429496727	1,149	0,942	0,969	0,955	0,98	0,999	0,0850676202
644245092	1,471	1,405	1,48	1,464	1,402	1,4444	0,0377796241
858993457	1,915	1,882	1,941	1,887	1,88	1,901	0,0264291506
1073741822	2,36	2,446	2,409	2,4	2,394	2,4018	0,0308901279
1288490187	2,856	2,829	2,819	2,843	2,809	2,8312	0,0187136314
1503238552	3,284	3,289	3,269	3,352	3,305	3,2998	0,0318857335
1717986917	3,744	3,749	3,755	3,773	3,783	3,7608	0,0165589855
1932735282	4,24	4,264	4,202	4,254	4,211	4,2342	0,026873779
2147483647	4,742	4,678	4,691	4,692	4,7	4,7006	0,0244499489

Ejecución Paralela 8 cores PC:

Tamaño	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Tiempo 5	Media	Desviación T
214748362	0,3	0,279	0,29	0,281	0,29	0,288	0,0083964278
429496727	0,589	0,569	0,558	0,57	0,566	0,5704	0,0114149025
644245092	0,841	0,849	0,85	0,84	0,842	0,8444	0,0047222876
858993457	1,217	1,112	1,115	1,125	1,154	1,1446	0,0437412848
1073741822	1,431	1,429	1,419	1,435	1,413	1,4254	0,0090994505
1288490187	1,69	1,739	1,741	1,709	1,69	1,7138	0,0251535286
1503238552	1,984	2	2,012	1,982	1,952	1,986	0,022627417
1717986917	2,243	2,295	2,259	2,277	2,279	2,2706	0,02001999
1932735282	2,538	2,519	2,746	2,568	2,517	2,5776	0,0963395038
2147483647	2,822	2,814	2,836	2,846	2,79	2,8216	0,0215592208

Ejecución Secuencial ATCGRID:

Tamaño	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Tiempo 5	Media	Desviación T
214748362	3,97	3,985	4,294	4,256	4,26	4,153	0,1609751534
429496727	7,94	8,112	8,065	8,262	8,229	8,1216	0,1299242087
644245092	12,187	12,219	12,215	12,228	12,227	12,2152	0,0166793285
858993457	16,191	16,314	16,191	16,151	16,185	16,2064	0,0624083328
1073741822	20,139	20,114	20,087	20,118	20,115	20,1146	0,0185013513
1288490187	24,119	24,295	23,874	24,075	24,119	24,0964	0,1503189941
1503238552	28,081	28,073	28,083	28,083	28,081	28,0802	0,0041472883
1717986917	32,039	32,044	32,045	32,039	32,012	32,0358	0,0135904378
1932735282	35,98	35,979	36,008	36,003	36,005	35,995	0,0142653426
2147483647	39,974	39,957	39,933	39,945	39,973	39,9564	0,0177707625

Ejecución Paralela ATCGRID 2 cores:

Tamaño	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Tiempo 5	Media	Desviación T
214748362	2,279	2,289	2,275	2,278	2,276	2,2794	0,0055946403
429496727	4,269	4,27	4,38	4,329	4,257	4,301	0,0523115666
644245092	6,287	6,25	6,248	6,253	6,831	6,3738	0,2560814324
858993457	8,228	8,241	8,235	8,257	8,243	8,2408	0,0107796104
1073741822	10,317	10,243	10,247	10,232	10,236	10,255	0,0351496799
1288490187	12,235	12,204	12,241	12,226	12,236	12,2284	0,0146731046
1503238552	14,213	14,192	14,202	14,448	14,182	14,2474	0,1127288783
1717986917	16,169	16,244	16,198	16,191	16,17	16,1944	0,0305172083
1932735282	18,554	18,183	18,165	18,158	18,172	18,2464	0,1722013357
2147483647	20,153	20,156	20,255	20,135	20,157	20,1712	0,0476885731

Ejecución Paralela ATCGRID 3 cores:

Tamaño	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Tiempo 5	Media	Desviación T
214748362	1,571	1,586	1,581	1,595	1,594	1,5854	0,0099146356
429496727	3,006	2,979	2,964	3,011	2,974	2,9868	0,0206082508
644245092	4,356	4,336	4,342	4,344	4,327	4,341	0,0106770783
858993457	5,699	5,628	5,717	5,699	5,706	5,6898	0,0353227972
1073741822	7,074	7,047	7,005	7,045	6,996	7,0334	0,0323001548
1288490187	8,324	8,331	8,388	8,402	8,358	8,3606	0,0342315644
1503238552	9,719	9,702	9,727	9,764	9,769	9,7362	0,0291496141
1717986917	11,084	11,016	11,083	11,163	11,125	11,0942	0,054833384
1932735282	12,404	12,373	12,499	12,444	12,397	12,4234	0,0493791454
2147483647	13,865	13,708	13,862	13,773	13,838	13,8092	0,0676069523

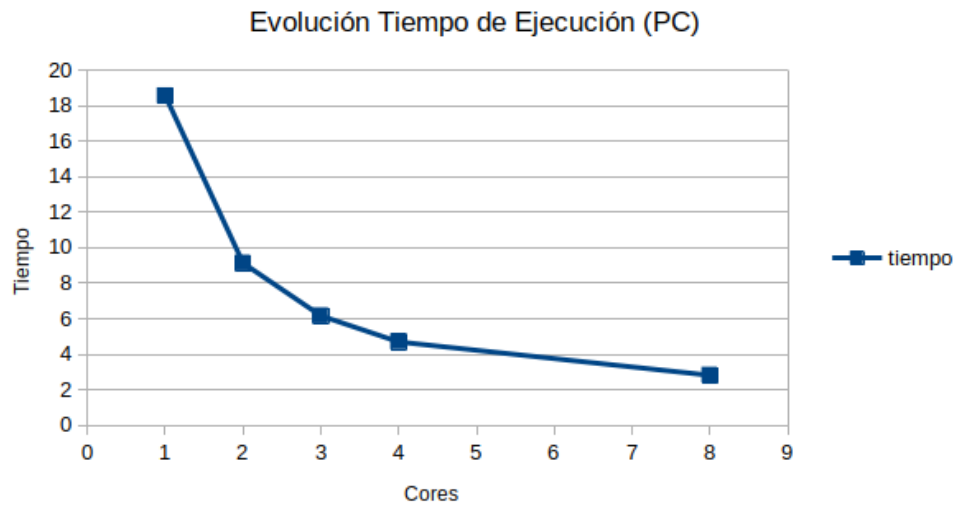
Ejecución Paralela ATCGRID 4 cores:

Tamaño	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Tiempo 5	Media	Desviación T
214748362	1,183	1,167	1,179	1,171	1,175	1,175	0,0063245553
429496727	2,326	2,325	2,301	2,304	2,293	2,3098	0,0148895937
644245092	3,319	3,298	3,308	3,325	3,331	3,3162	0,0132551877
858993457	4,345	4,318	4,308	4,318	4,335	4,3248	0,0148895937
1073741822	5,303	5,315	5,34	5,321	5,342	5,3242	0,0166643332
1288490187	6,345	6,302	6,307	6,296	6,329	6,3158	0,0205353354
1503238552	7,314	7,305	7,323	7,33	7,287	7,3118	0,016754104
1717986917	8,334	8,364	8,296	8,31	8,32	8,3248	0,025946098
1932735282	9,367	9,301	9,394	9,351	9,322	9,347	0,0366264931
2147483647	10,359	10,356	10,304	10,375	10,304	10,3396	0,03329114

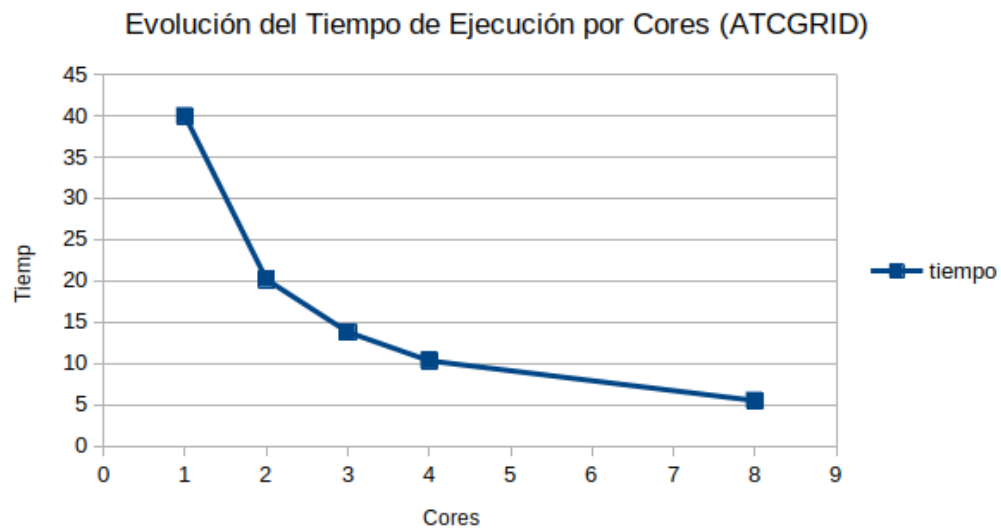
Ejecución Paralela ATCGRID 8 cores:

Tamaño	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Tiempo 5	Media	Desviación T
214748362	0,594	0,582	0,607	0,603	0,599	0,597	0,0096695398
429496727	1,168	1,171	1,198	1,161	1,168	1,1732	0,0143422453
644245092	1,768	1,764	1,771	1,749	1,746	1,7596	0,0113710158
858993457	2,325	2,31	2,302	2,298	2,306	2,3082	0,0104019229
1073741822	2,846	2,882	2,815	2,836	2,848	2,8454	0,0242857983
1288490187	3,38	3,377	3,351	3,39	3,368	3,3732	0,0146867287
1503238552	3,903	3,952	3,963	3,932	3,921	3,9342	0,0239729014
1717986917	4,471	4,43	4,459	4,388	4,376	4,4248	0,0420321306
1932735282	4,971	4,983	4,99	4,971	4,897	4,9624	0,0374539718
2147483647	5,447	5,534	5,559	5,527	5,539	5,5212	0,0431532154

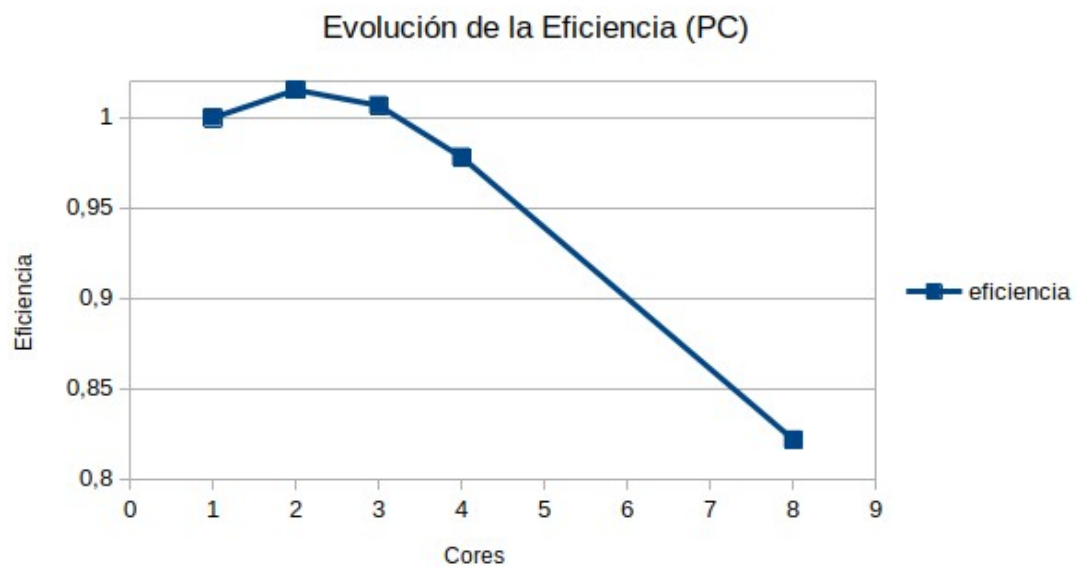
Comparativa por cores PC:



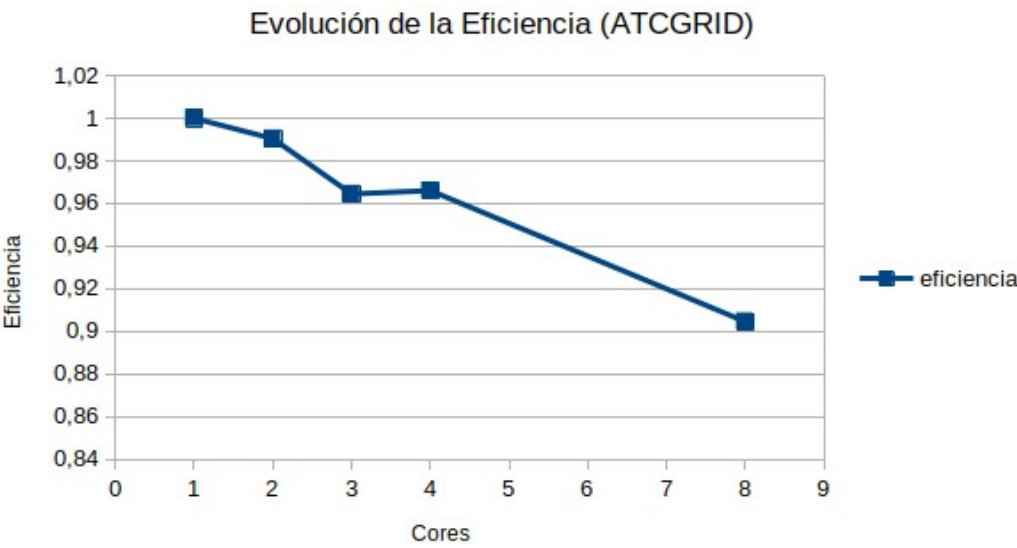
Comparativa por cores ATCGRID:



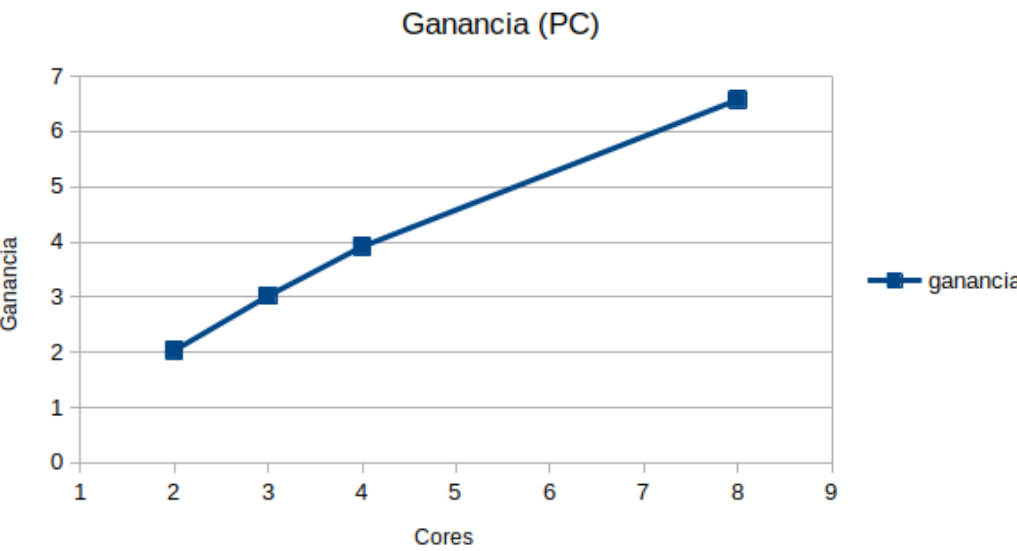
Eficiencia PC:



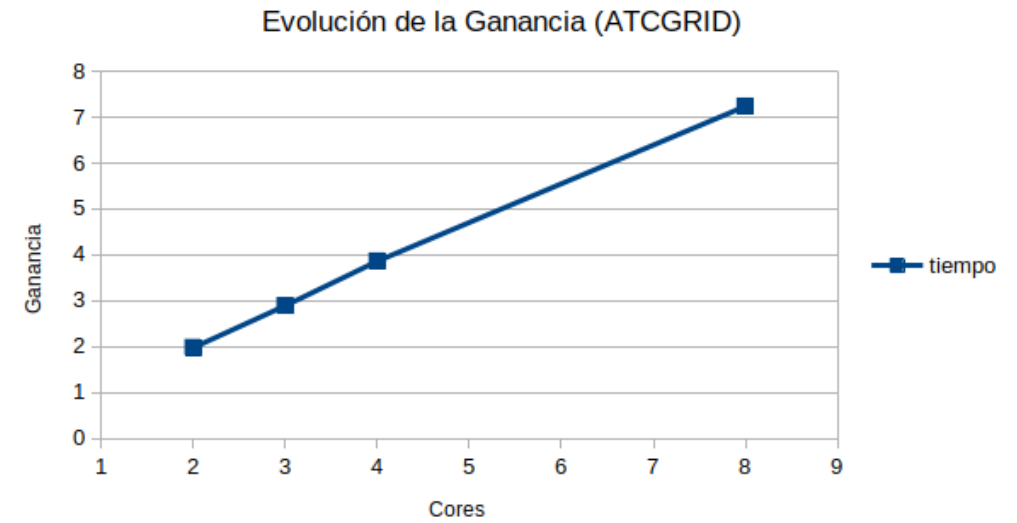
Eficiencia ATCGRID:



Ganancia PC:



Ganancia ATCGRID:



CONCLUSIONES EJECUCIONES SECUENCIALES Y PARALELAS:

- De la ejecución secuencial podemos sacar en claro que el tiempo obtenido por mi PC es bastante mejor que el obtenido en ATCGRID, pues es algo menos de la mitad del obtenido por ATCGRID. Lo que nos indica que un core de mi PC tiene un rendimiento mejor que un core de ATCGRID.
- Respecto a las ejecuciones paralelas vemos que en ambos casos que según aumenta el tamaño linealmente, el tiempo aumenta también de forma lineal.
- Por otro lado si nos fijamos únicamente en los tiempos obtenidos por cores para el tamaño máximo (2147483647), vemos en ambos casos que la reducción del tiempo no es lineal, sino que va reduciendo su pendiente conforme aumentan el número de cores, ya que la sobrecarga se hace mayor conforme aumentamos los cores. Se aprecia además, que en ATCGRID la reducción de la pendiente es menor que en mi ordenador, aunque en ninguno de los casos es lineal.

CONCLUSIONES GANANCIA Y EFICIENCIA:

- Si nos fijamos en la ganancia para ambos casos vemos que la ganancia es aparentemente lineal. Sin embargo no lo es, pues si nos fijamos en ambas gráficas, para 2, 3 y 4 cores se alcanza prácticamente la ganancia máxima (de hecho en mi ordenador la ganancia es un poco superior a la máxima, debido a que al realizar la toma de datos de la ejecución paralela estaba conectado cargando el ordenador), mientras que cuando utilizamos 8 cores la ganancia no se acerca tanto a la máxima, aunque se obtienen buenos resultados.
- También hay que destacar que la ganancia obtenida por ATCGRID es un poco peor que la obtenida por mi ordenador, exceptuando el caso de los 8 cores donde obtiene una ganancia mejor, lo que se puede deber a que mi ordenador no dispone de 8 cores físicos, sino que habrá tenido que utilizar cores lógicos, mientras que ATCGRID sí dispone de 8 cores físicos para hacer los cálculos.
- Si nos fijamos en las gráficas de eficiencia, mi ordenador obtiene una eficiencia por encima de 1 para los casos de 2 y 3 cores (por que cuando tomé el tiempo secuencia no estaba conectado el cargado, mientras que para la ejecución paralela sí), y posteriormente cae en picado hasta la eficiencia obtenida por los 8 cores. Por su parte ATCGRID obtiene una eficiencia que es más estable, aunque se va reduciendo poco a poco hasta obtener una eficiencia de 0,9 para 8 núcleo, que es bastante superior al 0,82 obtenido por mi ordenador.
- En conclusión, mi ordenador obtiene mejores tiempos de ejecución y mejores datos de eficiencia y ganancia cuando se utiliza cores físicos, sin embargo, cuando utiliza cores lógicos, al tener menos cores físicos que ATCGRID obtenemos que a partir de 4 cores mi ordenador escala peor que ATCGRID.
- Finalmente, recalcar la mejora obtenida al paralelizar el código, y los buenos resultados obtenidos