

Práctica 2 PPR

José Manuel Pérez Lendínez

February 24, 2021

Contents

1	Plataforma de ejecución	3
2	Tablas de datos	3
3	Gráfica	4
4	Explicación	4

1 Plataforma de ejecución

Como plataforma de ejecución he utilizado el nodo compute4 de hgpu@ugr.es

2 Tablas de datos

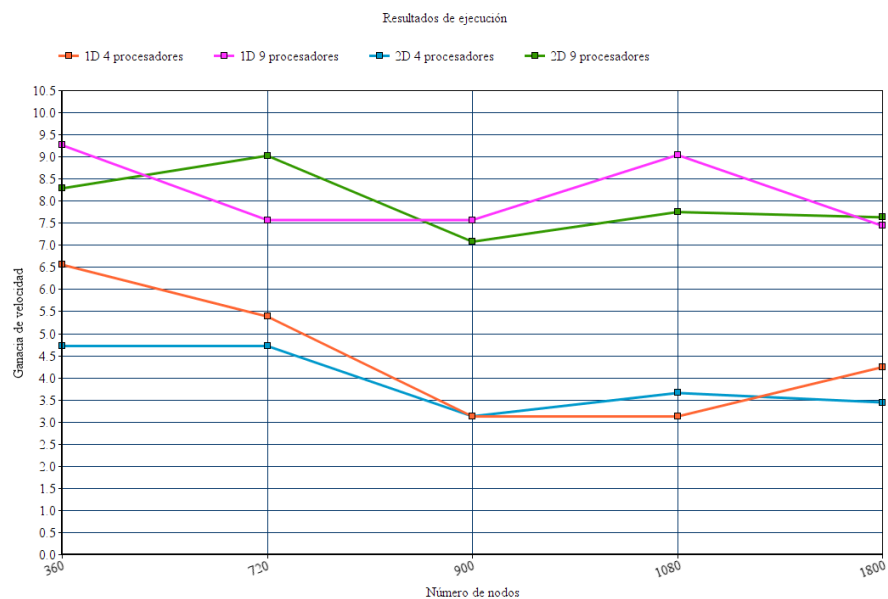
1. Datos del algoritmo 1d:

N	T(P=1)(sec)	Tp(P=4)	S(P=4)	Tp(P=9)	S(P=9)
360	0.186834	0.028482	6.55972193	0.020168	9.26388338
720	1.09248	0.202727	5.388922048	0.101846	10.726783575
900	1.56142	0.499318	3.127105372	0.206231	7.571218682
1080	2.98102	0.953895	3.125102868	0.329645	9.043122147
1800	12.9065	3.04101	4.244149148	1.73491	7.439290799

2. Datos del algoritmo 2d:

N	T(P=1)(sec)	Tp(P=4)	S(P=4)	Tp(P=9)	S(P=9)
360	0.186834	0.0395689	4.721738537	0.0225415	8.288445756
720	1.09248	0.202478	5.395549146	0,121023	9.027044446
900	1.56142	0.498633	3.131401251	0.220507	7.081045046
1080	2.98102	0.814724	3.65893235	0.384503	7.752917402
1800	12.9065	3.74596	3.445445226	1.69085	7.633143094

3 Gráfica



4 Explicación

Como se ve en la gráfica la ejecución con 9 procesos mejora en tiempo a la ejecución en 4 procesos. Esto se cumple tanto para el algoritmo 1d como el 2d. Con esto podemos entender que al utilizar mas procesos, dividimos la matriz de datos en partes mas pequeñas y cada proceso necesitara menos tiempo de ejecución para calcular su parte. En este caso aunque también necesitemos mas comunicaciones entre procesos al tener mas procesos, la cantidad de operaciones que paralelizamos es mas significativa y hace que las ejecuciones con 9 procesos sean de media el doble de rápidos que las que utilizan 4.

A la hora de comparar un algoritmo con otro vemos que están muy igualados tanto con 4 como con 9 procesos. Esto se puede deber a que el algoritmo 1d necesita menos comunicaciones aunque con un tamaño mayor, mientras con 2d necesitamos mas comunicaciones de menor tamaño.

Por ejemplo para $P=9$ procesos tendremos los siguientes datos para cada K en el calculo de floyd:

1. **1D:**

$$\text{Número de envíos} = P-1 = 8$$

$$\text{Tamaño de envío} = Ne(\text{Número de vértices.})$$

2. **2D:**

$$\text{Número de envíos} = ((\sqrt{P}-1)^2 \cdot 2) + (\sqrt{P} \cdot 2 - 2) = ((\sqrt{9}-1)^2 \cdot 2) + (\sqrt{9} \cdot 2 - 2) = 8 + 4 = 12$$

$$\text{Tamaño de envío} = \frac{Ne}{\sqrt{P}} = \frac{Ne}{\sqrt{9}} = \frac{Ne}{3}$$

Con esto podemos obtener lo siguiente:

1. **1D:** Tendrá menos comunicaciones pero en el total de estas enviara mas datos que 2d, siendo peor en este sentido. En cambio al tener menos comunicaciones los procesos estarán menos tiempo esperando a que el proceso que tiene que enviarle los datos de la fila k llegue al broadcast y los envíe. También gastara menos tiempo en crear las comunicaciones.
2. **2D:** En este caso en total enviá menos datos que el algoritmo 1d. Pero gastara mas tiempo en crear las comunicaciones, al necesitar mas de ellas. Ademas un proceso puede llegar a necesitar la fila k de un proceso y la columna k de otro distinto, gastando mas pudiendo gastar mas tiempo en esperar que el algoritmo 1d.

Debido a lo explicado en los puntos anteriores los algoritmos tendrán un tiempo parecido de ejecución.