Readme

July 9, 2024

Contents

1 PRAM

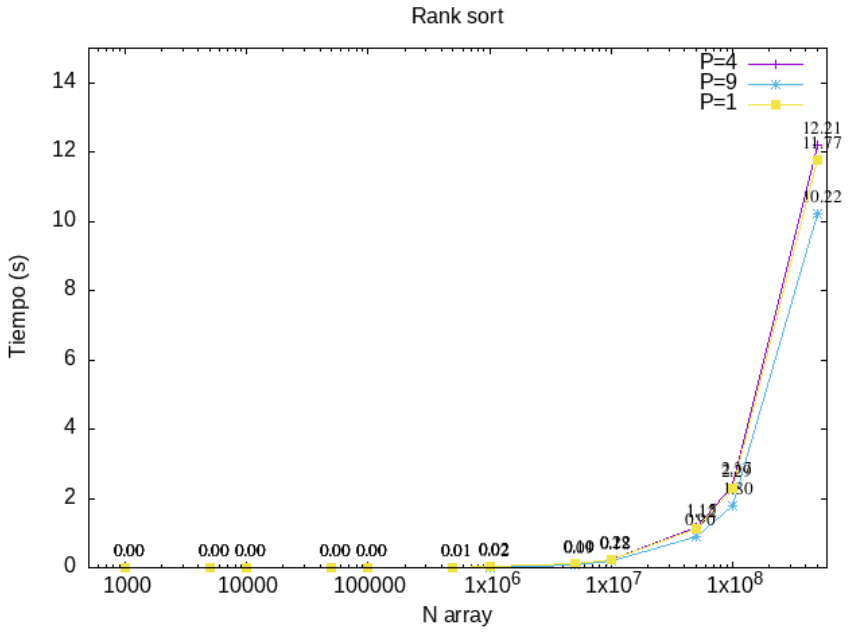
```
funcion OrdenamientoPorRanking (D, P):
    // Input
    for each processor P_i in parallel:
        D_i = D[local data of P_i]

    // Gossip
    for each processor P_i in parallel:
        for each processor P_j:
            Send D_i to P_j
        Receive D_j from all processors

    // Broadcast
    for each processor P_i in parallel:
        D_total = concatenate(D_1, D_2, ..., D_P)

    // Sort
    for each processor P_i in parallel:
        Sort(D_total)

    // Local ranking
    for each processor P_i in parallel:
```

```
    Local_Rank(D_total)

// Reduce
for each processor P_i in parallel:
    Ajuste_global (D_total)

return D_total
```

2 AVG

N=	P=4	P=9	P=1 (std::sort)
1000	1.0964253e-4	1.833024e-4	2.0885867e-5
5000	2.0050533e-4	2.6279727e-4	1.048308e-4
10000	1.0267657e-3	3.6683713e-4	1.95725e-4
50000	1.040205e-3	1.0267657e-3	9.8106287e-4
100000	2.229278e-3	1.809614e-3	1.9839953e-3
500000	0.010410888	8.814666e-3	9.92553e-3
1000000	0.021732107	0.0182426	0.02043482
5000000	0.1077866	0.09061254	0.10477986
10000000	0.21800473	0.17713373	0.21763607
50000000	1.1539427	0.89548933	1.1215313
100000000	2.3665887	1.7995367	2.2878553
500000000	12.206007	10.223427	11.7709

Gráfico con escala logarítmica en el axis x.


Rank sort
P=4
P=9
P=1
Tiempo (s)
N array
0
2
4
6
8
10
12
14
1000
10000
100000
1x10^6
1x10^7
1x10^8
12.21
10.22
0.00
0.01
0.02