

はじめに

○用いるプログラムと実行環境

- ・ 魔方陣の全解を求めるプログラム
- ・ 筑波大学のスーパーコンピュータT2K-Tsukuba上で実行
- ・ プログラムに与えるパラメータによる実行時間の変化を調査

○魔方陣

- ・ 縦X、横X、計X²マス
- ・ 1からX²までの数が1つつ入れられている
- ・ 縦のX列、横のX列、斜めのX列それぞれの数字の合計がすべて等しく、Lとなる

$$L = \frac{1}{X} \sum_{i=1}^{X^2} i = \frac{1}{2}X(X^2 + 1)$$

- ・ Xが6以上の場合については正確な総数は求められていない

X	総数[個]
1	1
2	0
3	1
4	880
5	275305224

2	9	4
7	5	3
6	1	8

X=3

9	6	3	16
4	15	10	5
14	1	8	11
7	12	13	2

X=4

1	7	13	19	25
18	24	5	6	12
10	11	17	23	4
22	3	9	15	16
14	20	21	2	8

X=5

アルゴリズムの開発

○アルゴリズムの概要

- ・ 総当りをベースにし、枝刈り法を改良
- ・ 列中の(X-1)個のマスが埋まったとき、残りの1マスはLから(X-1)個マスの数の合計を引くことにより計算可能
- ・ 数字を入れる順番を工夫することで、X=5のとき総当りで数字を入れるマスの個数Aを25から14に削減

①	⑧	⑨	✓	⑤
⑩	②	⑪	⑥	✓
⑫	⑬	③	⑭	✓
✓	⑦	✓	④	✓
✓	✓	✓	✓	✓

丸数字は総当りで数字を入れる順番、✓は自動的に求められるマスを表す

斜めの列上のマスを優先的に埋めることで、自動的に求められるマスの個数を増加

並列プログラムの開発

○並列方式

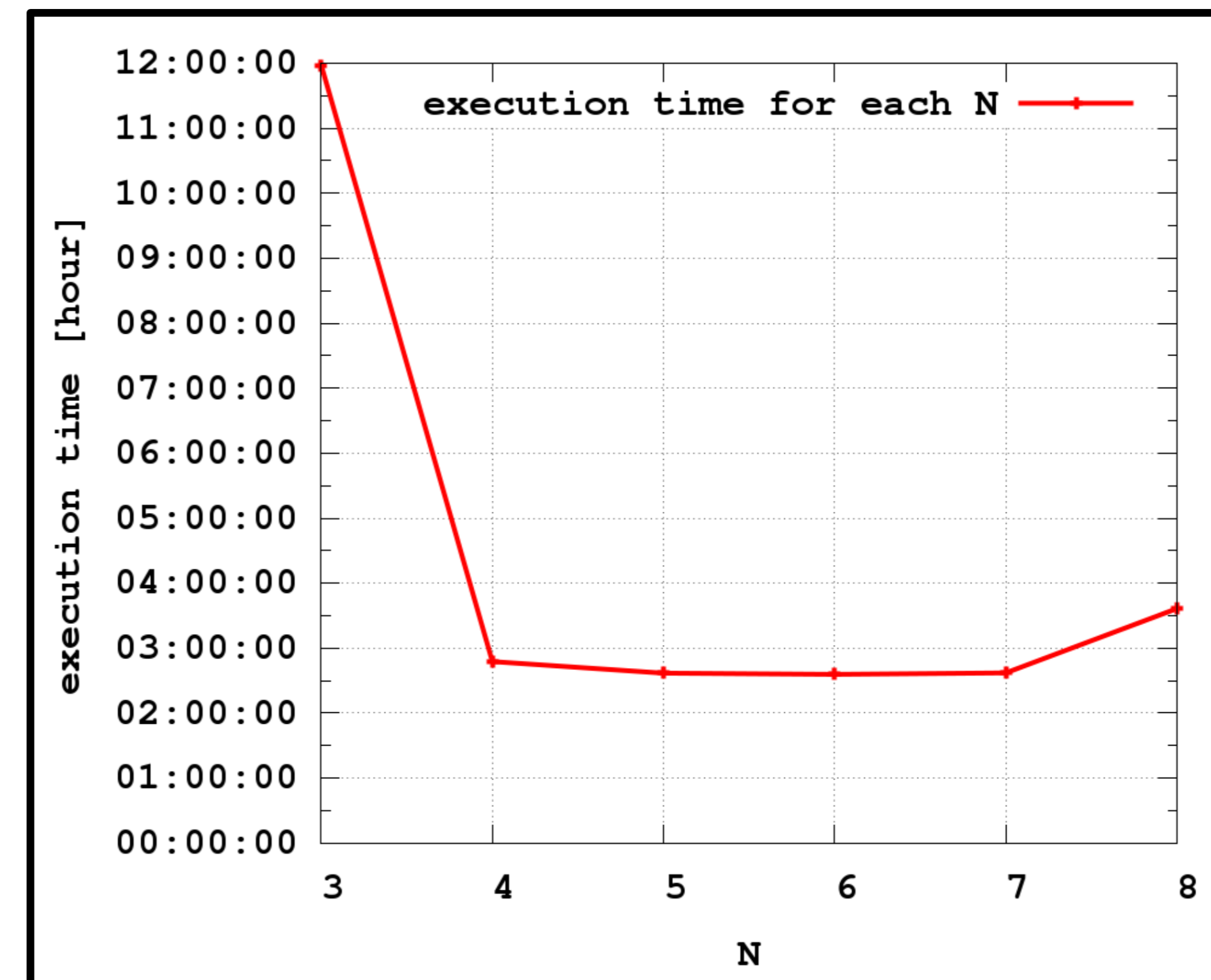
- ・ マスタ・ワーカー型並列
- ⇒ マスタがN番目 (0 ≤ N ≤ A) のマスまで総当りし、それをワーカーに配布
ワーカーはN+1番目のマスから総当り
- ・ Nの値が大きいほど粒度が小さく、Nの値が小さいほど粒度が大きい

並列プログラムの実行

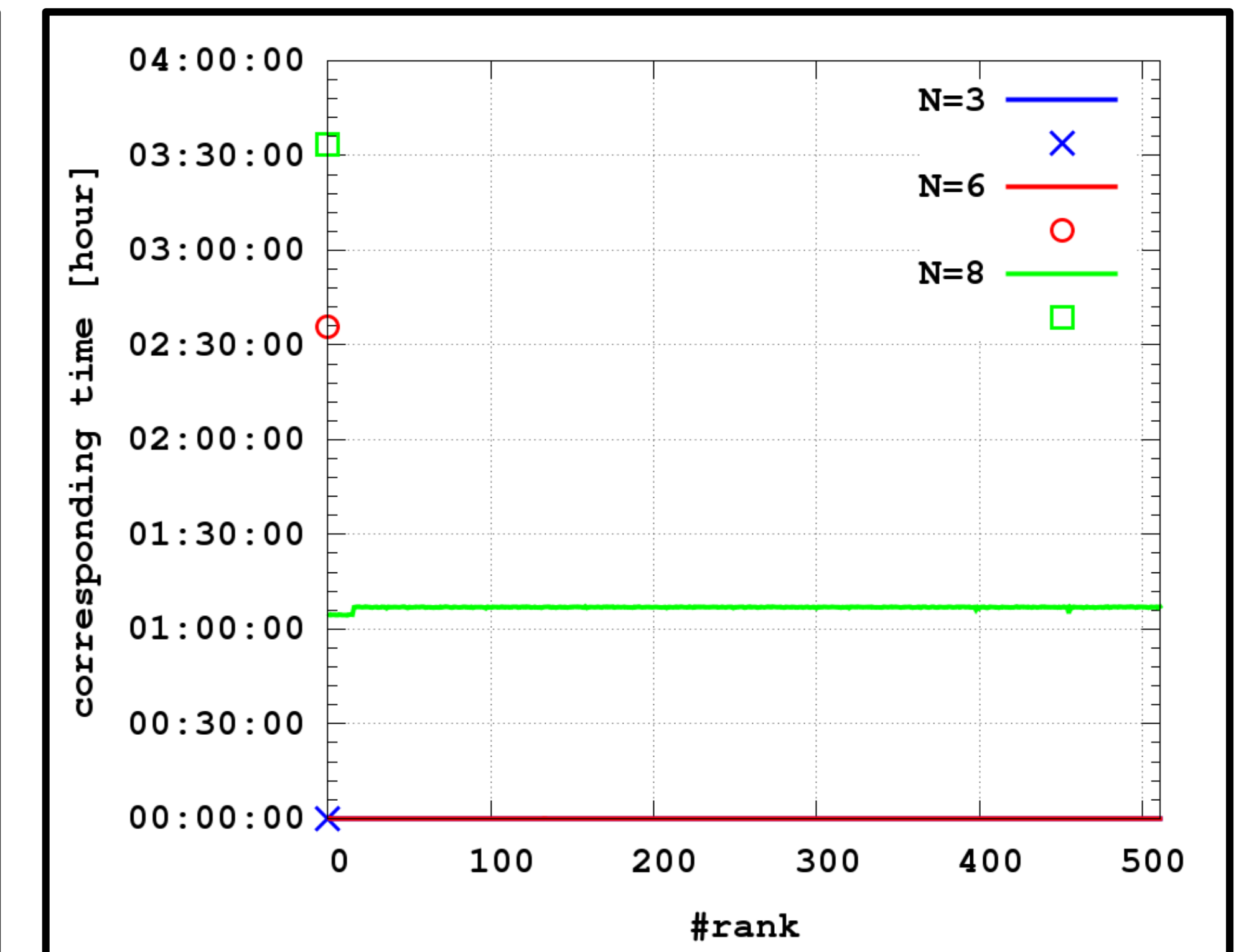
○実行の詳細

- ・ T2K-Tsukubaの512CPUコア (32ノード) 上で実行
- ・ 条件として X=5、N=0から14とする

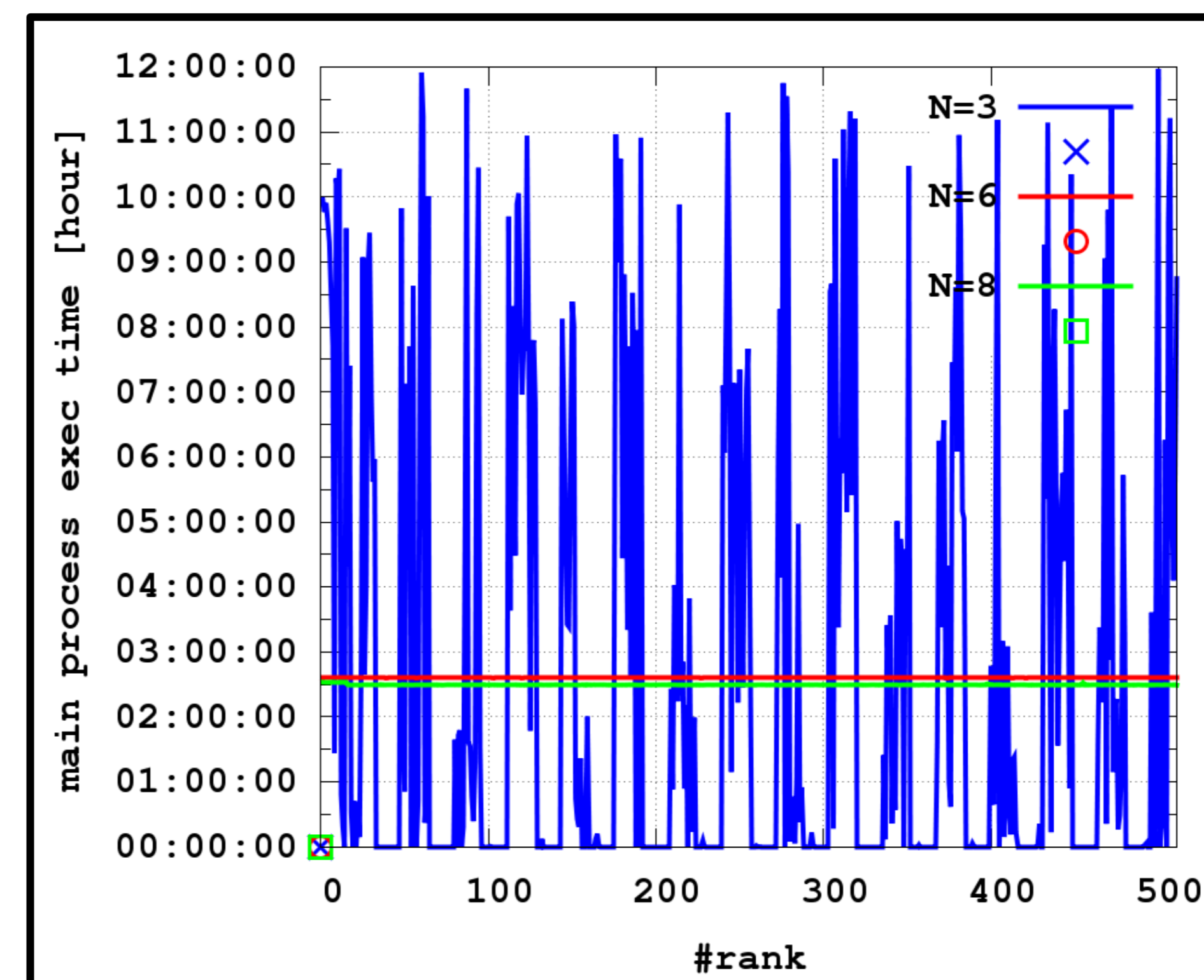
○実行結果



各Nにおける実行時間



各Nにおける各ワーカーの総通信時間



各Nにおける各ワーカーの主要処理実行時間

N	標準偏差[時間]
3	3:46:51.442588
6	0:00:00.417556
8	0:00:25.616082

各Nにおける各ワーカーの主要処理実行時間の標準偏差

○考察 (実行時間が顕著なN=3, 6, 8のみに注目)

- ・ 各Nにおける各ワーカーの総通信時間はN=3, 6で短く、N=8で長くなった。これはN=8では粒度が小さすぎるため、通信回数が多くなったことが原因であると考えられる
- ・ 各Nにおける各ワーカーの主要処理実行時間の標準偏差はN=6, 8で小さく、N=3で大きくなった。これはN=3では粒度が大きすぎるため、各ワーカーの処理量が偏ってしまったことが原因であると考えられる
- ・ バランスのとれたN=6が最も短い時間で実行終了したのだと考えられる

本研究における数値計算は筑波大学計算科学研究センター学際共同利用プログラムによる

研究チーム: 杉崎 行優 (並木中等教育学校)、朴 泰祐 (筑波大学)