

n 次魔方陣全解問題における並列化と性能解析

はじめに

魔方陣とは、縦 n 、横 n 、計 n^2 マスに 1 から n^2 までの数が入れられており、縦の n 列、横の n 列、斜めの n 列それぞれの数字の合計が全て等しく、 L (式 1) となるものを指す (図 1)。また、各 n における魔方陣の総数を表 1 に示す。 n が 6 以上の場合については正確な個数は求められていない。

$$L = \frac{1}{n} \sum_{i=0}^{n^2} i = \frac{1}{2}n(n^2 + 1)$$

式 1: 列の合計 L

n	総数
1	1
2	0
3	1
4	880
5	275,305,224

表 1: n 魔方陣の総数

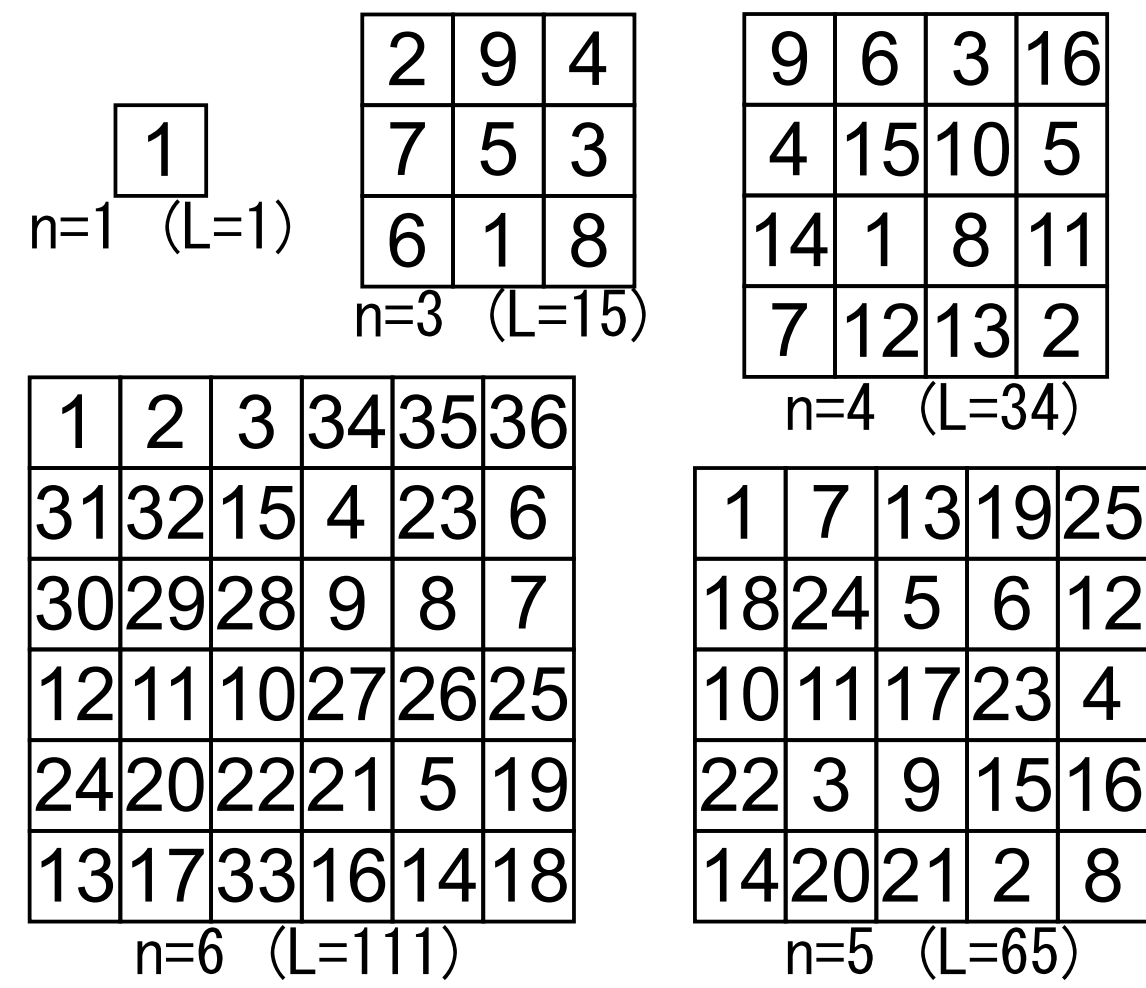


図 1: n 魔方陣の一例

今回は魔方陣の全解を求めるプログラムを筑波大学のスーパーコンピュータ T2K-Tsukuba で実行し、プログラムに与えるパラメータによる総実行時間の変化を調べた。

アルゴリズムの開発

総当たりのアルゴリズムを用い、その上で枝刈り法を工夫した。具体的には、全マスの数字の総当たりをする際、列中の $X-1$ 個のマスが埋まれば残りの 1 マスは L からそれらの数の合計を引くことにより求められることを利用した。

さらに今回は、数字を埋める順番も工夫した。まず初めに斜めのマスを埋め、次にその他のマスを埋めていくというものである。斜めの列状のマスは縦と横だけでなく斜めの列にも属しているため、それらのマスを優先的に埋めていくことで引き算により求められるマスの個数が増える。

各 n における数字を埋めていく順番を図 2 に示す。数字は順番、点 (・) は引き算により求められるマスを表している。また、各 n における数字を埋めるマスの個数 A を表 2 に示す。

最終的に判定される候補の個数は高々 $X^2 P_A$ 個である。

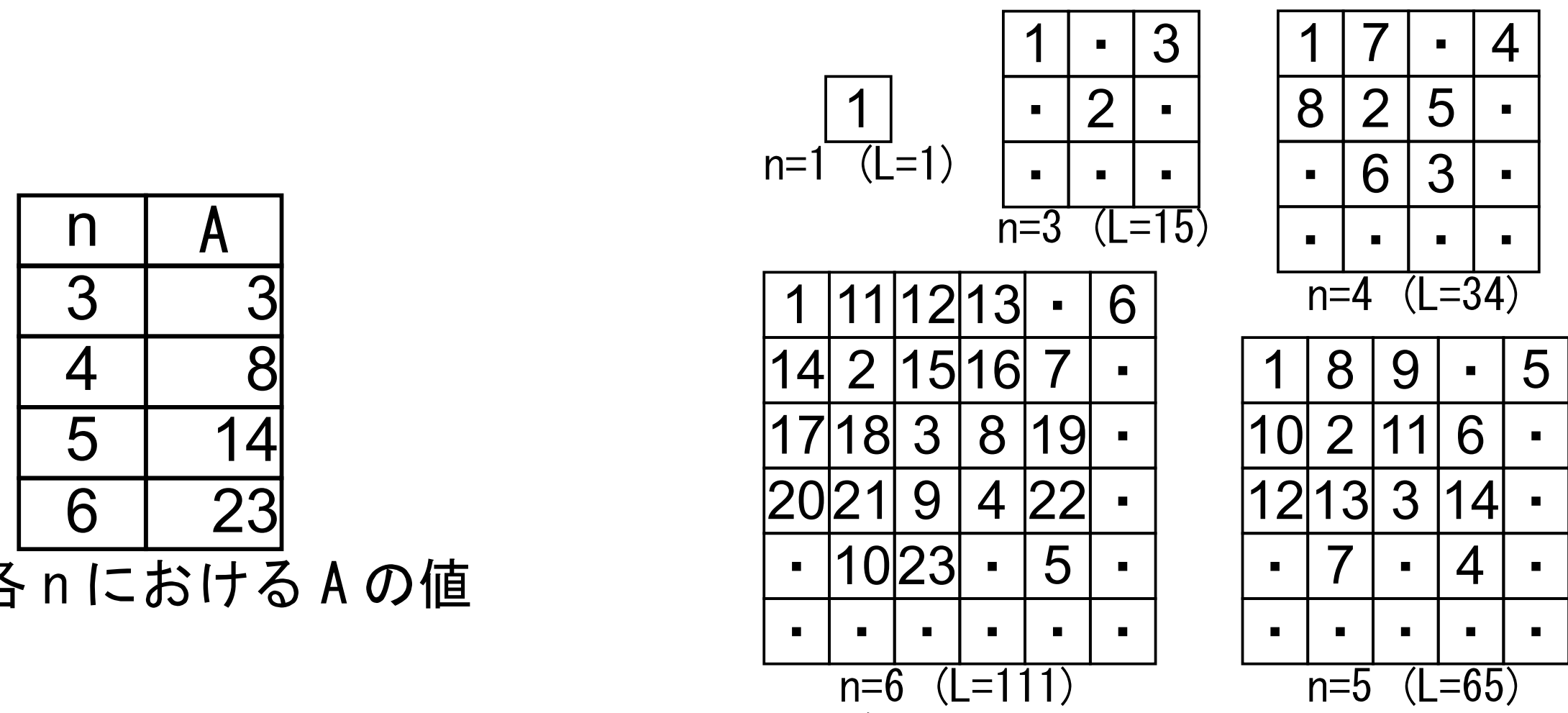


図 2: 各 n における数字を埋めていく順番

n	A
3	3
4	8
5	14
6	23

表 2: 各 n における A の値

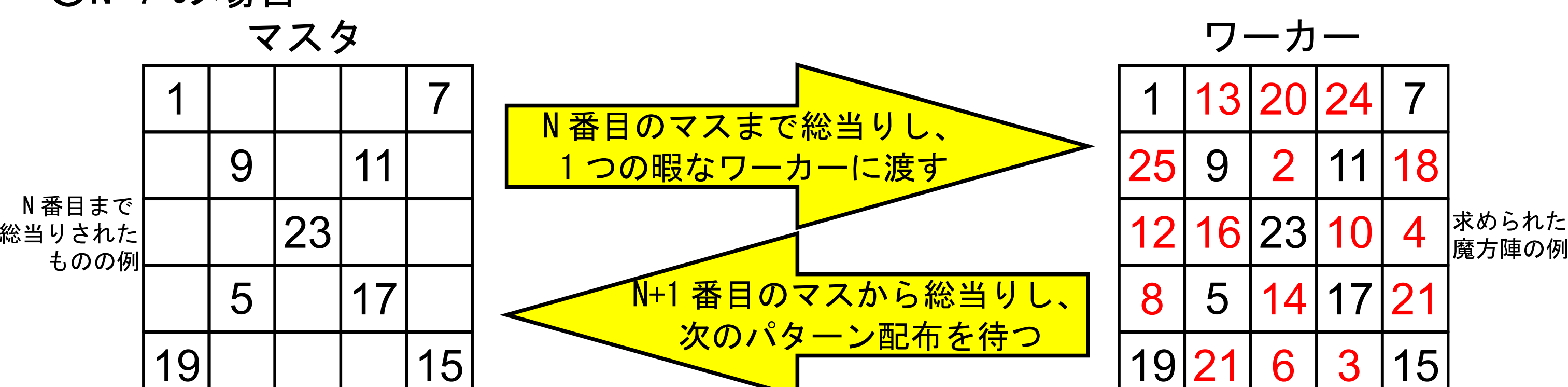
並列プログラムの開発

並列方式にはマスタ・ワーカー型並列を用いた。具体的な手順を以下に示す。

1. マスタが N 番目 ($0 < N \leq A$) のマスまで、数字の総当たりをする。
2. そのうちの 1 つのパターンを 1 つの暇なワーカーに配る。
3. ワーカーがそれを受け取る。
4. そのワーカーが $N+1$ 番目からの数字を総当たりする。マスタは手順 2 から繰り返す。
5. ワーカーの 1 つのパターンに対する処理が終了したらマスタからのパターン配布を待ち、手順 3 から繰り返す。
6. マスタの N 番目までの総当たり処理が終了したらマスタが全ワーカーに終了を伝え、結果を出力し、実行を終了する。

ワーカーがマスタから受け取るパターンによってワーカーの処理時間は異なる。また、マスタがワーカーに配るパターン数は、 N の値が小さいほど少なく、 N の値が大きいほど多い。つまり、 N の値が小さいほど問題を荒く分けているといえ、 N の値が大きいほど問題を細かく分けているといえる。

$N=7$ の場合



並列プログラムの実行

プログラムは T2K-Tsukuba の 32 ノード (512CPU コア) 上で実行し、1 プロセスをマスタ、511 プロセスをワーカーとした。また、最大連続実行可能時間は 24 時間となっている。

プログラムは、 $n=5$ とし、 N を 0 から 13 まで変化させ並列実行した。各 N における実行時間のグラフを図 3、各 N における各ワーカーの総主要処理実行時間のグラフを図 4、その標準偏差を表 3、各 N における各ワーカーの総通信時間のグラフを図 5 に示す。

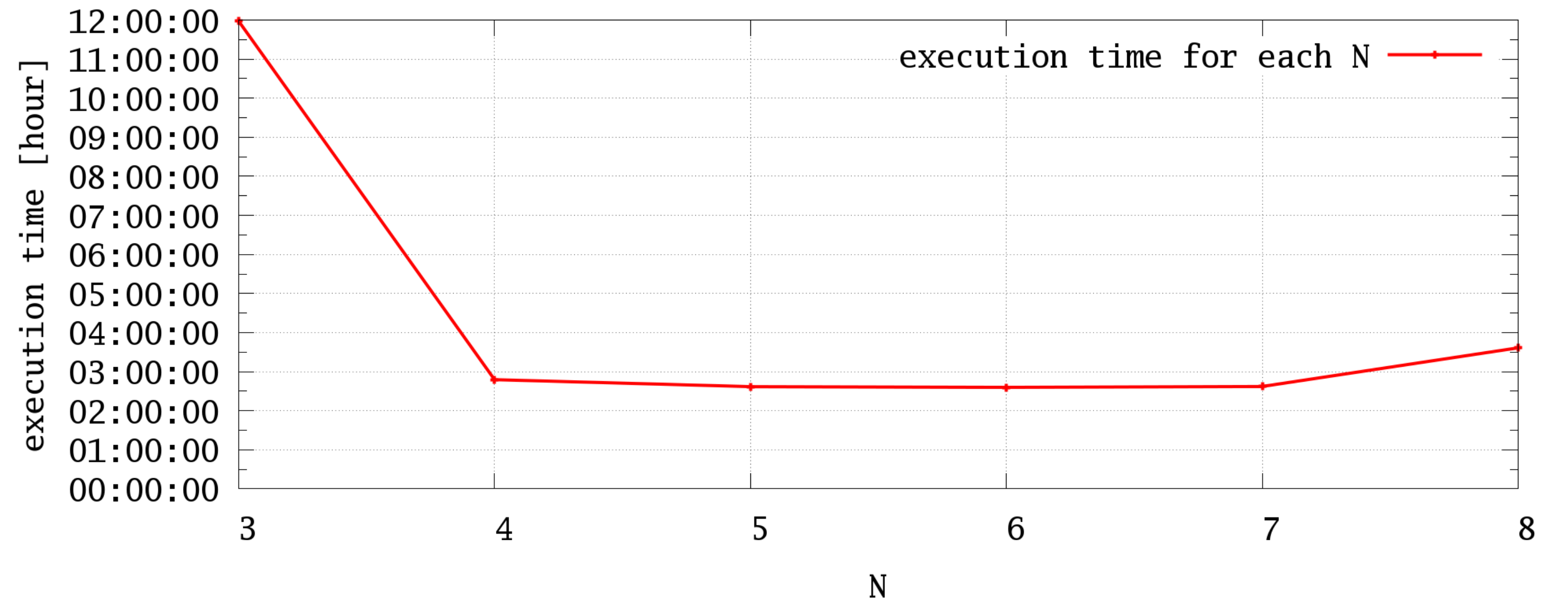


図 3: 各 N における実行時間

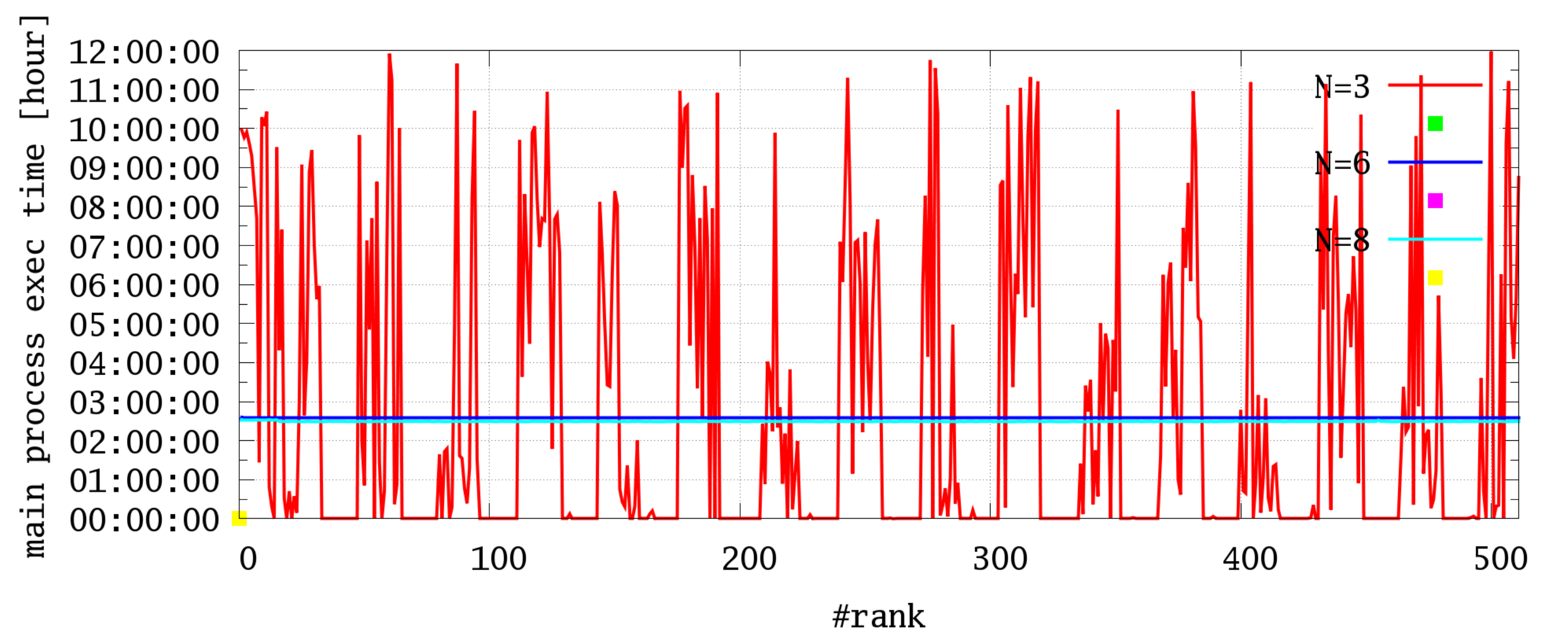


図 4: 各 N における各ワーカーの総主要処理実行時間

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

表 3: 各 N における各ワーカーの総主要処理実行時間の標準偏差

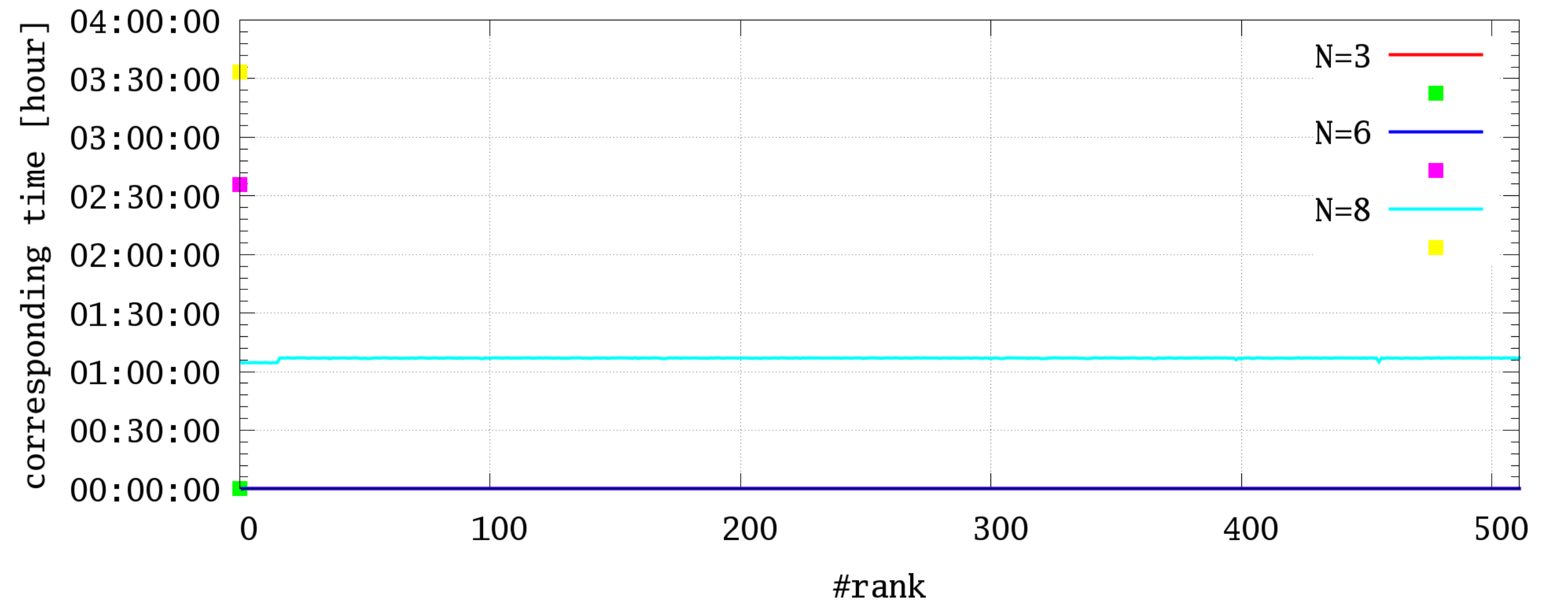


図 5: 各 N における各ワーカーの総通信時間

考察

- ・ 24 時間以内に実行が終了したのは N が 3 から 8 のときであった。
- ・ 実行時間は $N=3$ で最も長く、 $N=6$ で最も短く、 $N=8$ で再び長くなっている。
- ・ 各 N における各ワーカーの総主要処理実行時間の標準偏差は、 $N=6, 8$ では小さくなっているが、 $N=3$ では大きくなっていることが分かる。これは、 $N=3$ ではマスタからワーカーに配られる問題の粒度が他の N と比べて荒く、各ワーカーの処理量に偏ってしまったからである。これにより $N=3$ では実行時間が長くなってしまったと考えられる。
- ・ 各 N における各ワーカーの総通信時間の平均は、 $N=3, 6$ では短くなっているが、 $N=8$ では長くなっていることが分かる。これは、 $N=8$ ではマスタからワーカーに配られる問題の粒度が他の N と比べて細かく、マスタと各ワーカーの通信回数が多くなってしまったからである。これにより $N=8$ では実行時間が長くなってしまったと考えられる。
- ・ $N=6$ では粒度が十分に細かく、通信時間が主要処理実行時間と比べ誤差なので、最も実行時間が短くなっ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。

杉崎 行優*1, 吉田 真也*1, 朴 泰佑*2
*1 茨城県立並木中等教育学校 *2 筑波大学