n次魔方陣全解問題における並列化と性能解析

並木中等教育学校 4年次B組　杉崎 行優

動機

- 一般的な整数問題である「n次魔方陣」の全解探索を超並列処理によって行う試み

n次魔方陣

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

- 図: n次魔方陣（n=5）の一例

- 縦n、横n、計n^2マス
- 1～n^2がuniqueに入る
- 縦・横・斜めの列それぞれの合計がすべて等しい

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

（n=5の場合L=65）

- n≧6については正確な個数が未解明

目的

- nの増大に伴い、階乗的に増加する探索空間を数学的に絞り込む

- 動的負荷分散処理をMPIによりプログラムし効率的処理を行う

- 当初目的は、未だに解が求まっていないn=6の求解であったが、計算リソース上困難
  ⇒目標をn=5（解が求まっている）に対して正しく動作するプログラムと、その動的負荷分散機能の検証及び評価に修正

研究方法（計算環境）

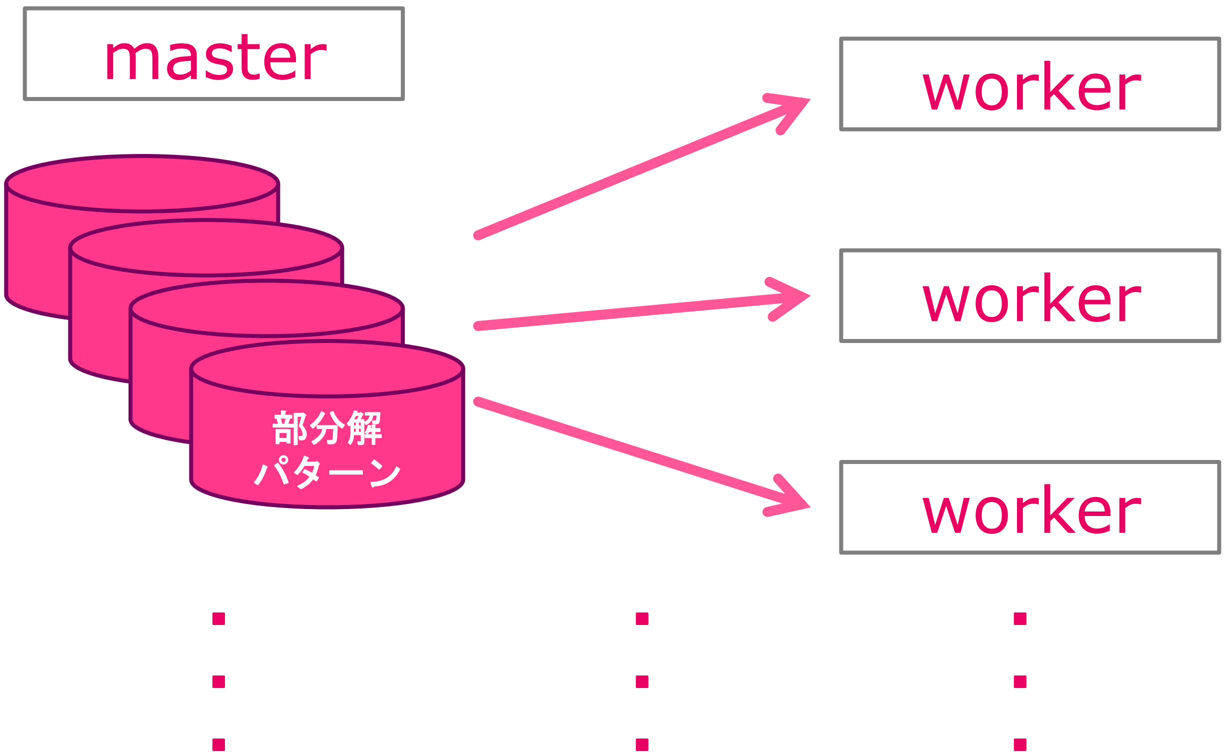
- 筑波大学計算科学研究センター学際共同利用プログラムにより、筑波大学のスーパーコンピュータであるT2K-Tsukubaを利用
- 32ノード上で512スレッドを実行
- 最大連続実行時間は24時間

研究方法（アルゴリズムの改良）

- 鏡像性、途中解から一意に求まる解、等の性質から、できるだけ計算量を絞る
- 部分的に解けているパターンからその先の全解探索を行うよう、探索空間を木構造化
- 部分解からの探索の計算量は事前に求められないため、動的タスクスケジューリングが必要
- 部分解パターンの表現データが小さいのに対し、探索空間は大きいため、master-worker型並列処理が有効

master-worker型並列

- masterが暇なworkerに問題を投げ、全ての候補に対する処理が終わるまで繰り返す



研究方法（並列化時のポイント）

- 部分解を「25マス中N個まで埋まった状態」とし、このパターンをworkerに分配

- Nが大きいとその先の探索空間が小さく、Nが小さいとその逆

- worker数が大きいとmasterの分配処理が大きくなり、小さいとmasterはidleになりやすい

⇒これらを踏まえ、worker数とNの関係の最適値を経験的に調査

進捗どうですか

- 進捗ありました！

今後の計画

- 逐次アルゴリズムを改良する
- 最終的にn=5をn=6の部分問題と考え、どのようなパラメータで探索すべきかを求める

最後に

- 戯言…