

日程計画における作業履歴を活用した ファジィ・ランダム多目的最適化の並列分散

平成 30 年 12 月 5 日

富山県立大学 電子・情報工学科 情報基盤工学講座 4 年

杉山桃香

はじめに

発表の流れ

- 1 現状と目標
- 2 環境設定
- 3 **RaspberryPi と Mpich を用いた並列計算デモ**
- 4 まとめ

1. 現状と目標

現状

- 1 ファジィ・ランダム変数を含む問題の定式化
- 2 RaspberryPi8 台の初期設定とパスワードなし ssh 接続

目標

- 1 RaspberryPi の作業ディレクトリを共有する
- 2 RaspberryPi8 台に Mpich をインストール
- 3 RaspberryPi8 台で簡単な並列計算の実行

└ 2. 環境設定

└ 2.1 NFS の設定

2. 環境設定

2.1 NFS の設定

NFS

あるノードが所有する 1 つのディレクトリを、他の複数のノードで共有して使うための仕組み

NFS を使う理由

MPI を使う際には実行ファイルが全ノードで同じ位置に置かれている必要があるから

完了

master の作業ディレクトリを各 slave に共有した

└ 2. 環境設定

└ 2.2Mpich の導入と使用方法

2. 環境設定

2.2Mpich の導入と使用方法

host ファイルの準備

外部ネットワークでプロセス間通信を行うために各ノードの IP アドレスかホスト名を書いたテキストファイルをマスターに置く

Mpich の導入

- 1 インストール
→ `sudo apt-get install mpich`
- 2 コンパイル
→ `sudo mpicc hellow.c`
- 3 実行 (host ファイルに書いた 8 台で)
→ `sudo mpirun -hostfile host -np 8 ./hellow`

3. 実験準備

3.1hello world を並列実行

Raspberry Pi8 台で「hellow world」の表示を並列実行し
動作が正常か確認した

```
pi@master:/home/mpil/test $ mpirun --hostfile host -np 8 ./hellow
Hello world from process 0 of 8
Hello world from process 1 of 8
Hello world from process 2 of 8
Hello world from process 6 of 8
Hello world from process 4 of 8
Hello world from process 3 of 8
Hello world from process 5 of 8
Hello world from process 7 of 8
```

3.RaspberryPi と Mpich を用いた並列計算デモ

3.2 円数率の計算

円周率の計算を **RaspberryPi** 1 台, 2 台, 4 台, 8 台で実行したときの処理時間比較した

台数	処理時間[sec]
1台	0.0125
2台	0.0142
4台	0.0316
8台	0.0380

3.RaspberryPi と Mpich を用いた並列計算デモ

3.3 島 GA を用いた巡回セールスマン問題

TSP 問題を RaspberryPi 1 台, 2 台, 4 台, 8 台で実行したときの最短距離と処理時間比較した

台数	最短距離[km]	処理時間
1台	222.29	2分46秒
2台	215.93	3分03秒
4台	206.43	2分52秒
8台	218.32	2分48秒

5. まとめ

完了したこと

- 1 **Mpich** のインストール
- 2 **Mpich** を用いた並列処理の動作確認 (**hello world**)
- 3 **Mpich** を用いた計算の並列分散処理と処理時間の計測 (円周率の計算)
- 4 **Mpich** で並列 **GA** を実行する (**TSP** 問題)

今後の課題

- 1 ファジィ・ランダム変数の処理を加え
目的関数の部分を変更して並列 **GA** を実行