補助内容

水上 和秀

富山県立大学　電子・情報工学科

February 4, 2022

分散処理の環境構築

Dask（分散処理をするためのライブラリ）について調べて、Daskの環境構築をした。

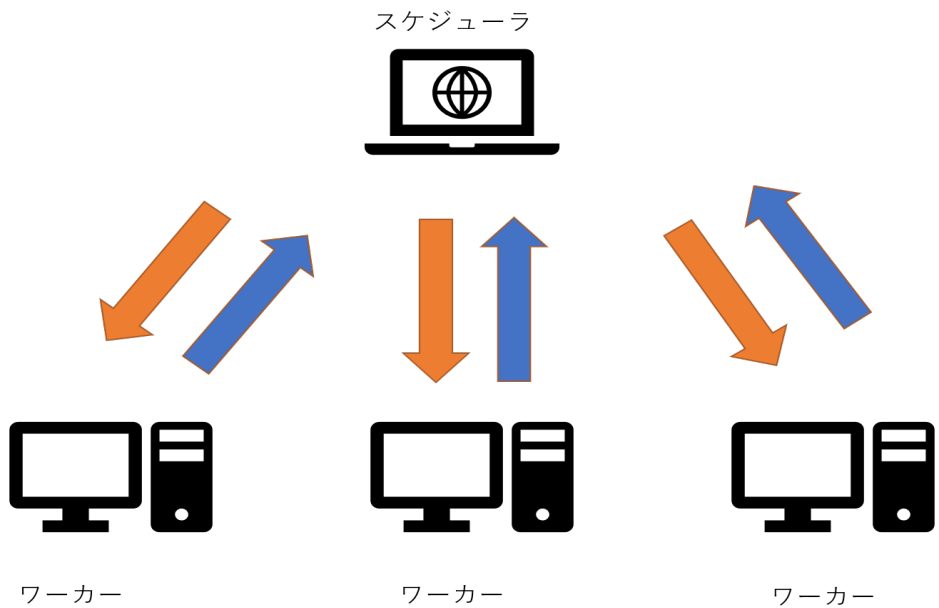
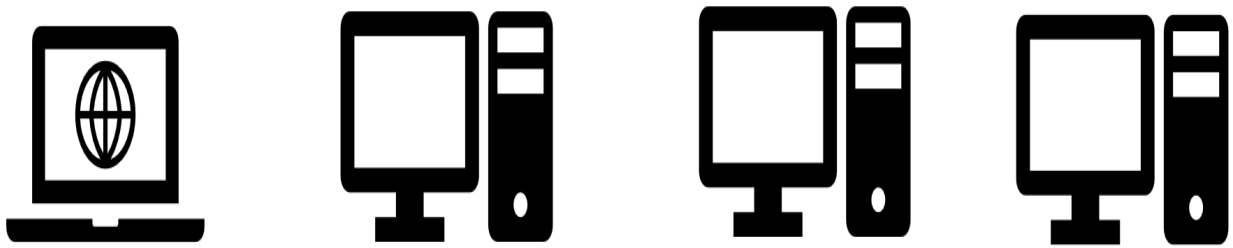


図 1: イメージ図

実装

分散処理の実装

また、Dask を使って一つのプログラムを複数の pc で処理できるようにした。研究室の pc を使い、1 つの pc、2 つの pc、4 つの pc の 3 つのパターンでプログラムを動かし、処理にかかった時間を調べた。



pc1	pc2	pc3	pc4
スレッド数:4	スレッド数:4	スレッド数:4	スレッド数:4
メモリー: 3.96Gib	メモリー: 3.98Gib	メモリー: 3.96Gib	メモリー: 3.95Gib

Table 1: 使用した pc

検証

選択コマンド プロンプト - py NSGA2_recipe.py

23	34500	6.00000E+01	2.18463E+02	1	0.00000E+00	f
24	36000	6.00000E+01	2.04975E+02	1	0.00000E+00	f
25	37500	6.00000E+01	1.92984E+02	1	0.00000E+00	f
26	39000	5.50000E+01	1.81795E+02	1	1.36000E+03	ideal
27	40500	5.50000E+01	1.71037E+02	1	0.00000E+00	f
28	42000	5.50000E+01	1.60337E+02	1	0.00000E+00	f
29	43500	3.50000E+01	1.50763E+02	1	2.16000E+02	ideal
30	45000	3.50000E+01	1.41709E+02	1	0.00000E+00	f
31	46500	3.50000E+01	1.32726E+02	1	0.00000E+00	f
32	48000	1.50000E+01	1.24176E+02	1	3.32000E+02	ideal
33	49500	1.50000E+01	1.16328E+02	1	0.00000E+00	f
34	51000	1.50000E+01	1.08480E+02	1	0.00000E+00	f
35	52500	6.800000000	1.00787E+02	1	2.16100E+03	ideal
36	54000	6.800000000	9.31798E+01	1	0.00000E+00	f
37	55500	0.00000E+00	8.62940E+01	1	1.00900E+03	ideal
38	57000	0.00000E+00	8.00793E+01	1	0.00000E+00	f
39	58500	0.00000E+00	7.51263E+01	1	0.00000E+00	f
40	60000	0.00000E+00	7.03377E+01	1	0.00000E+00	f
41	61500	0.00000E+00	6.55510E+01	2	1.000000000	ideal
42	63000	0.00000E+00	6.08802E+01	2	0.00000E+00	f
43	64500	0.00000E+00	5.63966E+01	3	0.440366972	ideal
44	66000	0.00000E+00	5.24744E+01	3	0.00000E+00	f
45	67500	0.00000E+00	4.92184E+01	4	0.240796020	ideal
46	69000	0.00000E+00	4.56575E+01	4	0.093750000	ideal
47	70500	0.00000E+00	4.28497E+01	4	0.00000E+00	f
48	72000	0.00000E+00	3.98387E+01	4	0.00000E+00	f
49	73500	0.00000E+00	3.70291E+01	6	0.057692308	ideal
50	75000	0.00000E+00	3.44179E+01	6	0.00000E+00	f
51	76500	0.00000E+00	3.20674E+01	6	0.00000E+00	f
52	78000	0.00000E+00	2.99003E+01	4	0.333333333	ideal
53	79500	0.00000E+00	2.75809E+01	4	0.00000E+00	f
54	81000	0.00000E+00	2.60179E+01	3	0.381235154	ideal
55	82500	0.00000E+00	2.39474E+01	3	0.00000E+00	f
56	84000	0.00000E+00	2.23099E+01	3	0.336485422	ideal
57	85500	0.00000E+00	2.03292E+01	3	0.00000E+00	f
58	87000	0.00000E+00	1.85921E+01	3	0.140515504	f
59	88500	0.00000E+00	1.72280E+01	3	0.00000E+00	f
60	90000	0.00000E+00	1.58595E+01	3	0.00000E+00	f

Dask: 2046.271320104599 sec
Function value: [1360. 29352.]

図 2: pc1 のみで処理をした時の実行結果

実装

選択コマンド プロンプト - py NSGA2_recipe.py

23	34500	6.00000E+01	2.18463E+02	1	0.00000E+00	f
24	36000	6.00000E+01	2.04975E+02	1	0.00000E+00	f
25	37500	6.00000E+01	1.92984E+02	1	0.00000E+00	f
26	39000	5.50000E+01	1.81795E+02	1	1.36000E+03	ideal
27	40500	5.50000E+01	1.71037E+02	1	0.00000E+00	f
28	42000	5.50000E+01	1.60337E+02	1	0.00000E+00	f
29	43500	3.50000E+01	1.50763E+02	1	2.16000E+02	ideal
30	45000	3.50000E+01	1.41709E+02	1	0.00000E+00	f
31	46500	3.50000E+01	1.32726E+02	1	0.00000E+00	f
32	48000	1.50000E+01	1.24176E+02	1	3.32000E+02	ideal
33	49500	1.50000E+01	1.16328E+02	1	0.00000E+00	f
34	51000	1.50000E+01	1.08480E+02	1	0.00000E+00	f
35	52500	6.800000000	1.00787E+02	1	2.16100E+03	ideal
36	54000	6.800000000	9.31798E+01	1	0.00000E+00	f
37	55500	0.00000E+00	8.62940E+01	1	1.00900E+03	ideal
38	57000	0.00000E+00	8.00793E+01	1	0.00000E+00	f
39	58500	0.00000E+00	7.51263E+01	1	0.00000E+00	f
40	60000	0.00000E+00	7.03377E+01	1	0.00000E+00	f
41	61500	0.00000E+00	6.55510E+01	2	1.000000000	ideal
42	63000	0.00000E+00	6.08802E+01	2	0.00000E+00	f
43	64500	0.00000E+00	5.63966E+01	3	0.440366972	ideal
44	66000	0.00000E+00	5.24744E+01	3	0.00000E+00	f
45	67500	0.00000E+00	4.92184E+01	4	0.240796020	ideal
46	69000	0.00000E+00	4.56575E+01	4	0.093750000	ideal
47	70500	0.00000E+00	4.28497E+01	4	0.00000E+00	f
48	72000	0.00000E+00	3.98387E+01	4	0.00000E+00	f
49	73500	0.00000E+00	3.70291E+01	6	0.057692308	ideal
50	75000	0.00000E+00	3.44179E+01	6	0.00000E+00	f
51	76500	0.00000E+00	3.20674E+01	6	0.00000E+00	f
52	78000	0.00000E+00	2.99003E+01	4	0.333333333	ideal
53	79500	0.00000E+00	2.75809E+01	4	0.00000E+00	f
54	81000	0.00000E+00	2.60179E+01	3	0.381235154	ideal
55	82500	0.00000E+00	2.39474E+01	3	0.00000E+00	f
56	84000	0.00000E+00	2.23099E+01	3	0.336485422	ideal
57	85500	0.00000E+00	2.03292E+01	3	0.00000E+00	f
58	87000	0.00000E+00	1.85921E+01	3	0.140515504	f
59	88500	0.00000E+00	1.72280E+01	3	0.00000E+00	f
60	90000	0.00000E+00	1.58595E+01	3	0.00000E+00	f

Dask: 1436.8695645332336 sec
Function value: [1360. 29352.]

図 3: pc1 と pc2 で処理をした時の実行結果

選択コマンド プロンプト

26	39000	5.50000E+01	1.81795E+02	1	1.36000E+03	ideal
27	40500	5.50000E+01	1.71037E+02	1	0.00000E+00	f
28	42000	5.50000E+01	1.60337E+02	1	0.00000E+00	f
29	43500	3.50000E+01	1.50763E+02	1	2.16000E+02	ideal
30	45000	3.50000E+01	1.41709E+02	1	0.00000E+00	f
31	46500	3.50000E+01	1.32726E+02	1	0.00000E+00	f
32	48000	1.50000E+01	1.24176E+02	1	3.32000E+02	ideal
33	49500	1.50000E+01	1.16328E+02	1	0.00000E+00	f
34	51000	1.50000E+01	1.08480E+02	1	0.00000E+00	f
35	52500	6.800000000	1.00787E+02	1	2.16100E+03	ideal
36	54000	6.800000000	9.31798E+01	1	0.00000E+00	f
37	55500	0.00000E+00	8.62940E+01	1	1.00900E+03	ideal
38	57000	0.00000E+00	8.00793E+01	1	0.00000E+00	f
39	58500	0.00000E+00	7.51263E+01	1	0.00000E+00	f
40	60000	0.00000E+00	7.03377E+01	1	0.00000E+00	f
41	61500	0.00000E+00	6.55510E+01	2	1.000000000	ideal
42	63000	0.00000E+00	6.08802E+01	2	0.00000E+00	f
43	64500	0.00000E+00	5.63966E+01	3	0.440366972	ideal
44	66000	0.00000E+00	5.24744E+01	3	0.00000E+00	f
45	67500	0.00000E+00	4.92184E+01	4	0.240796020	ideal
46	69000	0.00000E+00	4.56575E+01	4	0.093750000	ideal
47	70500	0.00000E+00	4.28497E+01	4	0.00000E+00	f
48	72000	0.00000E+00	3.98387E+01	4	0.00000E+00	f
49	73500	0.00000E+00	3.70291E+01	6	0.057692308	ideal
50	75000	0.00000E+00	3.44179E+01	6	0.00000E+00	f
51	76500	0.00000E+00	3.20674E+01	6	0.00000E+00	f
52	78000	0.00000E+00	2.99003E+01	4	0.333333333	ideal
53	79500	0.00000E+00	2.75809E+01	4	0.00000E+00	f
54	81000	0.00000E+00	2.60179E+01	3	0.381235154	ideal
55	82500	0.00000E+00	2.39474E+01	3	0.00000E+00	f
56	84000	0.00000E+00	2.23099E+01	3	0.336485422	ideal
57	85500	0.00000E+00	2.03292E+01	3	0.00000E+00	f
58	87000	0.00000E+00	1.85921E+01	3	0.140515504	f
59	88500	0.00000E+00	1.72280E+01	3	0.00000E+00	f
60	90000	0.00000E+00	1.58595E+01	3	0.00000E+00	f

Dask: 1089.1242501735687 sec
Function value: [1360. 29352.]

図 4: 4 つの pc で処理をした時の実行結果

実装結果

実装結果

・1 つの pc でプログラムを動かしたときのプログラムの処理時間は約 34 分,2 つの pc でプログラムを動かしたときの処理時間は約 24 分,4 つの pc でプログラムを動かしたときのプログラムの処理時間は約 18 分であった。
・また、コンピュータの数が多いほど実行時間が短いことが確認できた。
・このことより並列処理のプログラムは有用であることが確認できた。

取り組んだ内容その２

並列処理の実装

concurrent.futures（並列処理をするためのライブラリ）について調べて、並列処理のプログラムを実装した。
また、ノート pc を使い、スレッド数が 1 から 10 のパターンでプログラムを動かし、処理にかかった時間を調べた。

取り組んだ内容その２

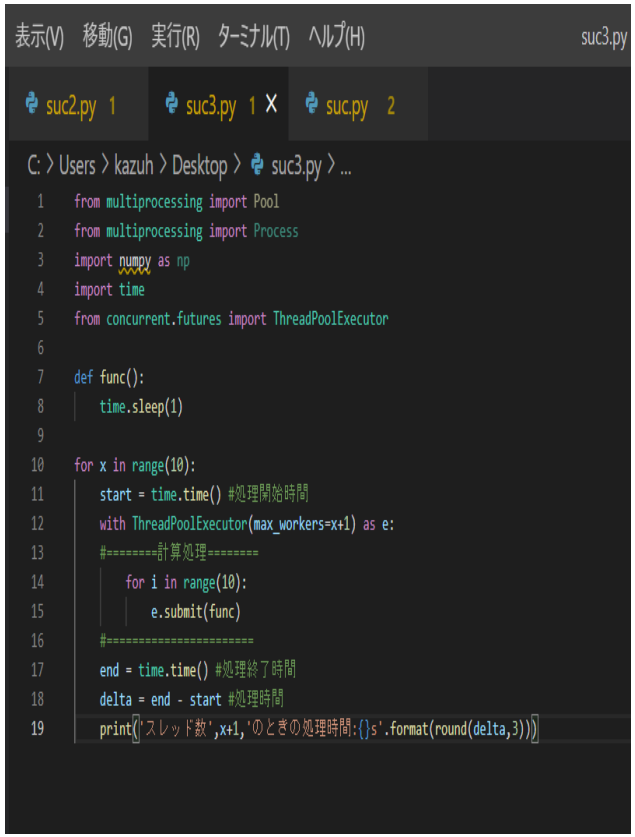


図 5: ソースコード

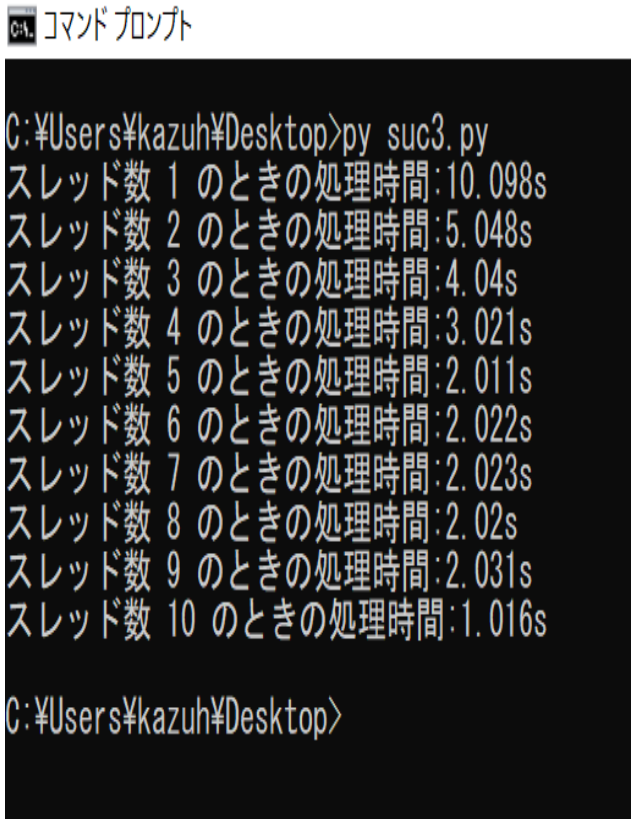


図 6: 実行結果

取り組んだ内容その２

ウェブスクレイピングの実装

python で HTML からデータを引き出せる BeautifulSoup というライブラリを使ってウェブサイトからデータを抽出した
また、抽出したデータを CSV に出力した。

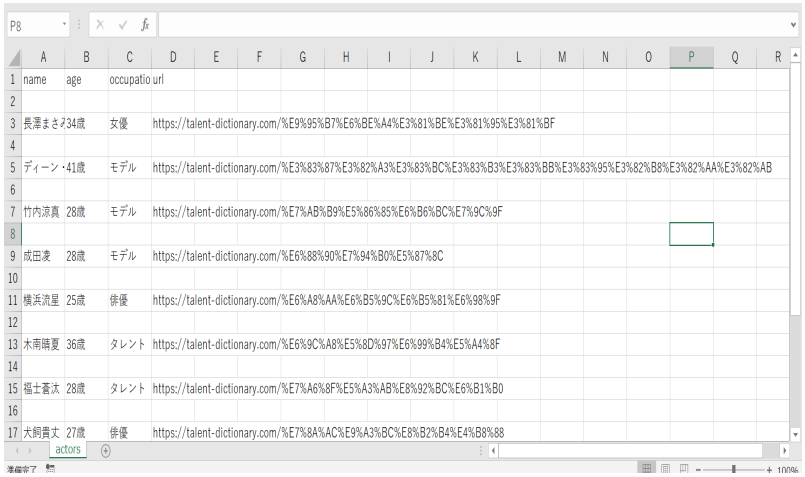
C: > Users > kazuh > Desktop > suc.py > ...

```python
import csv
import requests
from bs4 import BeautifulSoup

HEADER = ['name', 'age', 'occupation', 'url']

url = 'https://talent-dictionary.com/s/jobs/3/20'
r = requests.get(url)

soup = BeautifulSoup(r.text, 'html.parser')
actors = soup.find('ul', class_='list').find_all('li')

with open('actors.csv', 'w', encoding='Shift-Jis') as f:
    writer = csv.writer(f)
    writer.writerow(HEADER)
    for actor in actors:
        prof = actor.find('div', class_='right')

        name = prof.find('a', class_='title').text
        url = prof.find('a', class_='title').get('href')
        occupation = prof.find('a', class_='job').text
        age = prof.find('span', class_='age').text

        row = [name, age, occupation, url]
        writer.writerow(row)
```

図 7: スクレイピングのソースコード

取り組んだ内容その２

図 8: スクレイピングしたサイト



	A	B	C	D
1	name	age	occupatio	url
2				
3	長澤まさ	34歳	女優	https://talent-dictionary.com/%E9%95%B7%E6%BE%A4%E3%81%BE%E3%81%95%E3%81%BF
4				
5	ディーン・	41歳	モデル	https://talent-dictionary.com/%E3%83%87%E3%82%A3%E3%83%BC%E3%83%B3%E3%83%BB%E3%83%95%E3%82%B8%E3%82%AA%E3%82%AB
6				
7	竹内涼真	28歳	モデル	https://talent-dictionary.com/%E7%AB%B9%E5%86%85%E6%B6%BC%E7%9C%9F
8				
9	成田凌	28歳	モデル	https://talent-dictionary.com/%E6%88%90%E7%94%B0%E5%87%8C
10				
11	横浜流星	25歳	俳優	https://talent-dictionary.com/%E6%A8%AA%E6%B5%9C%E6%B5%81%E6%98%9F
12				
13	木南晴夏	36歳	タレント	https://talent-dictionary.com/%E6%9C%A8%E5%8D%97%E6%99%B4%E5%A4%8F
14				
15	福士蒼汰	28歳	タレント	https://talent-dictionary.com/%E7%A6%8F%E5%A3%AB%E8%92%BC%E6%B1%B0
16				
17	犬飼貴丈	27歳	俳優	https://talent-dictionary.com/%E7%8A%AC%E9%A3%BC%E8%B2%B4%E4%B8%88

actors

図 9: 実行結果

今後の課題

今後やっていくこと

- より高速にプログラムを動かせるようにする
- 先輩の手伝いを引き続き行う
- 先輩のプログラムをサーバー上で動かせるようにする。