

並列分散処理 最終レポート

チーム E 大城 慶知 眞榮城 隆守 伊波卓浩 宮良友也

July 21, 2018

最終報告書に載せるやつ (あとで消すやつ)

演習の背景、目的、方法、結果、考察を A410 ページ以内で適切にまとめる。個々のメンバーの役割分担を明記する。記載がない場合、あるいは、曖昧な場合には、減点の対象となる。例えば、あるタスクを複数名で担当した場合でも、個々のメンバーの役割をできる限り区別して説明する。最終報告書にはソースコードの github リポジトリも記載する。

1 テーマ

Python における非同期処理を用いた I/O の並列処理を行う。

2 Python 並列処理の基礎知識

2.1 スレッドの制約

Python では、GIL(Global Interpreter Lock) と呼ばれる制約がある。GIL とは、Python を実行した際に一つだけしかスレッドのリソースを起動できない制約である。そのため、Python の CPU バウンドの並列処理はプロセスを使って、I/O バウンドの並列処理はスレッドを行う事が多い。

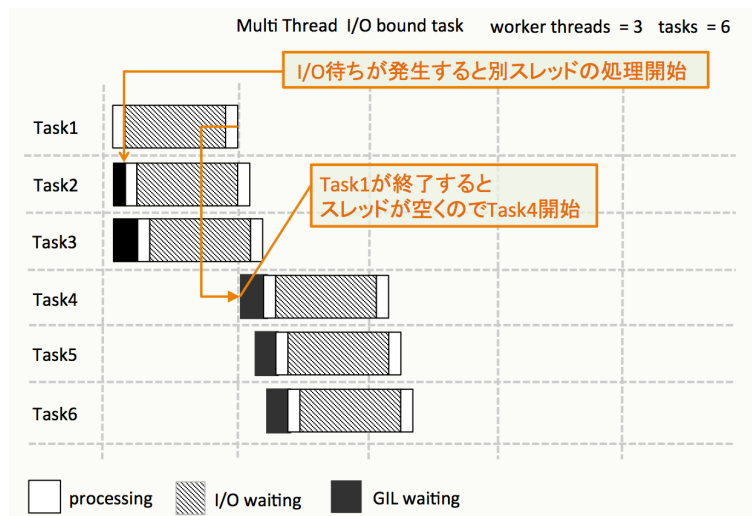


Figure 1: 林檎の図

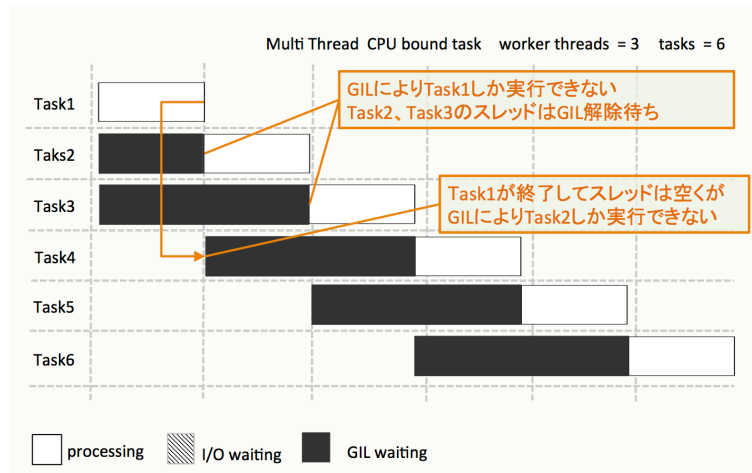


Figure 2: 林檎の図

2.2 プロセスを用いた

コード 1: シンプレクス法プログラム

3 実験方法

HTTP の GET を用いて実験を行った。GET を複数回実行する場合を考えると、逐次処理の場合ではレスポンスがあるまで次の GET を送信することができない。これを並列処理によりレスポンスを待つことなく次の GET を実行した。これにより効率よく GET を実行し、結果を受け取ることができると想定した。

4 実行結果

example1.py は逐次処理をしてくれるスクリプト。
example2.py はコルーチンを 20 個一気に呼び出す。
example3.py はコルーチンを 5 個ずつ 4 回呼び出している。
example4.py はスレッドを使用。

実行結果を以下に示す。

• example1.py

```
1 % python example1.py
2 1.7710762023925781
3 3.6481258869171143
4 5.234184980392456
5 6.819124937057495
6 8.661988735198975
7 10.357062816619873
8 12.127899885177612
9 13.939480066299438
10 15.66226577758789
11 17.136775016784668
12 18.71082091331482
13 22.602702856063843
14 24.2241370677948
15 25.92159104347229
```

```
16 | 27.67835807800293
17 | 29.555225133895874
18 | 31.347226858139038
19 | 33.13314199447632
20 | 34.98287010192871
21 | 36.86141490936279
```

• example2.py

```
1 | % python example2.py
2 | 6.204233169555664
3 | 6.204301834106445
4 | 6.204308032989502
5 | 6.204311847686768
6 | 6.204314947128296
7 | 6.204318046569824
8 | 6.204322099685669
9 | 6.204324960708618
10 | 6.2043280601501465
11 | 6.204332113265991
12 | 6.20433497428894
13 | 6.204339027404785
14 | 6.2043421268463135
15 | 6.204345941543579
16 | 6.204349040985107
17 | 6.204353094100952
18 | 6.204355955123901
19 | 6.204360008239746
20 | 6.204363107681274
21 | 6.204365968704224
```

• example3.py

```
1 | % python example3.py
2 | 1.983794927597046
3 | 1.9838616847991943
4 | 1.983867883682251
5 | 1.9838709831237793
6 | 1.983874797821045
7 | 3.5528876781463623
8 | 3.552910804748535
9 | 3.5529158115386963
10 | 3.5529186725616455
11 | 3.552921772003174
12 | 5.265967845916748
13 | 5.265992879867554
14 | 5.265997648239136
15 | 5.2660017013549805
16 | 5.266004800796509
17 | 6.929059743881226
18 | 6.929084777832031
19 | 6.929089784622192
20 | 6.929093837738037
21 | 6.929096937179565
```

• example4.py

```
1 | あいうえお
```

5 考察

6 感想・意見

GitHub の URL

<https://github.com/e165719/ParallelDistributedProcessing>