

手続き型プログラムのFaaS基盤での 実行可能性向上

株式会社富士通研究所

上野 優

ueno.masaru@fujitsu.com

開発における問題点

Function as a Service (FaaS) は経済的な新しいプラットフォームであるがリソース制約が厳しいため、既存の任意のプログラムを実行できるとは限らない

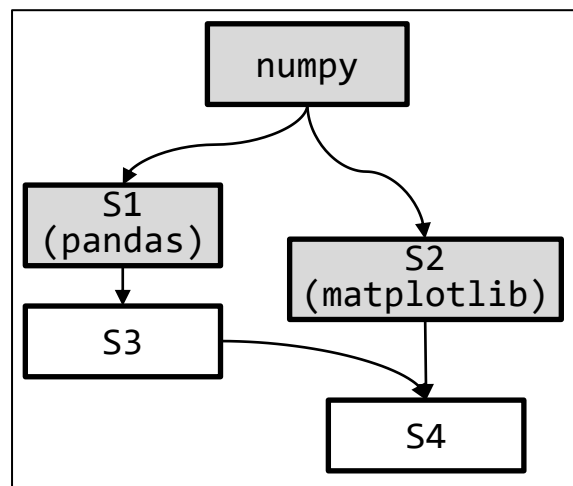
解決手法

- プログラム分割によりリソース制約を回避することで実行可能性を向上
- データ依存グラフ (Data Flow Graph) により、パッケージサイズ制約を満たすプログラム分割箇所の決定を支援

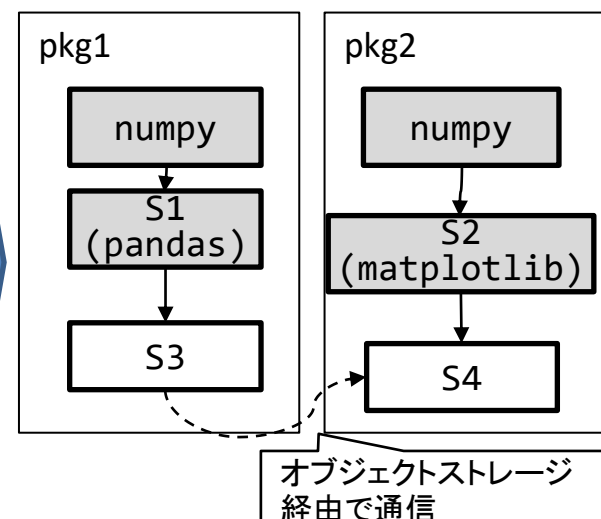
データ依存グラフ (Data Flow Graph)に基づくプログラム分割

```
S1: import pandas
S2: import matplotlib.pyplot
    as plt
S3: df = pandas.DataFrame(...)
S4: plt.plot(df)
```

既存プログラム



データ依存グラフ (Data Flow Graph)



分割後のライブラリサイズが
制約を満たすように分割

評価

手法の機能性・性能の評価

依存ライブラリサイズが大きくそのままでは動かない既存プログラムを3分割して約16000行・19MBのCSVを入力に実行し通信オーバーヘッドは無視できるほど小さいことを確認

手法の汎用性の評価

- 自然言語処理の専門家が公開されているPythonノートブック形式の16プログラムを対象にFaaSでの実行可能性を調査 (<https://github.com/nlptown/nlp-notebooks>)
- 結果
 - 分割不能: 9プログラム(56%)
 - 分割可能: 7プログラム(44%)、うち3プログラム(19%)は本手法適用前はパッケージサイズ制約を満たさずそのまま実行できなかったもの

議論

考察

- 本手法は文を最小単位として分割するが、反復的なパラメータ調整を行うプログラム(例えば機械学習のモデル学習)では単一のfor文の実行時間が長く実行時間上限を超えてしまう
- パブリッククラウドのインターコネクトが高速であるため通信オーバーヘッドが小さい

今後の課題

- 文より細かい粒度での分割
- 分割位置の最適化
- 転送サイズや実行時間を考慮