大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group

Cefore を用いた関数キャッシングの性能評価

大阪公立大学 大学院情報学研究科 基幹情報学専攻
小川 瞬也・藤本 まなと・阿多 信吾

1. 背景・研究目的

大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group

◘IoT機器の普及

- M2M(machine to machine)のプログラム製作を通じたIoT機器の組み合わせ連携によるサービス創出、さまざまな応用の広がり

◘エッジコンピューティング

- クラウドで一括処理するのではなくエッジで分散処理
- 遅延の削減や通信量の軽減により、品質向上

課題点

開発者がプログラムのうちどの部分をエッジでどの部分をクラウドで動かすかを適切に判断することが難しい

1. 背景・研究目的

大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group

➡ プログラムを修正することなく自動で処理をエッジまたはクラウドに適切に配置する分散処理アーキテクチャ

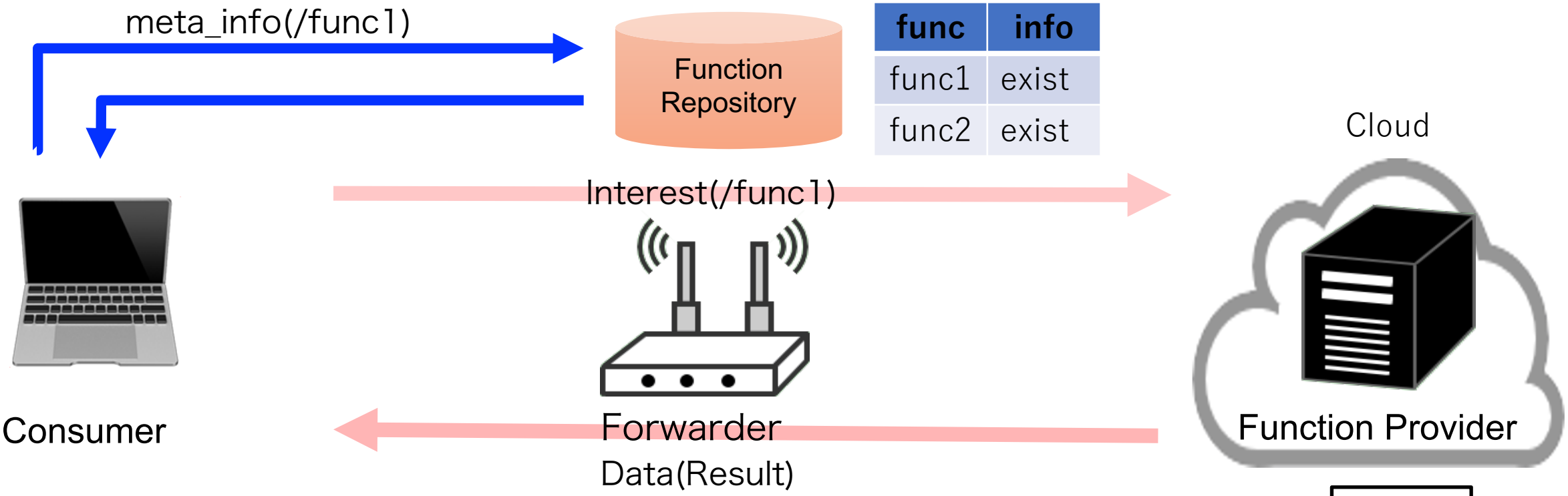
同一プログラムにより自動処理をエッジ又はクラウドに適切に配置する分散処理アーキテクチャの実現

➡ 目的関数に基づき処理をエッジで実行

本研究では遅延時間を短縮させることを目的

2.アーキテクチャの概要

大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group




```
-------------
func1(-----)
-------------
```

program

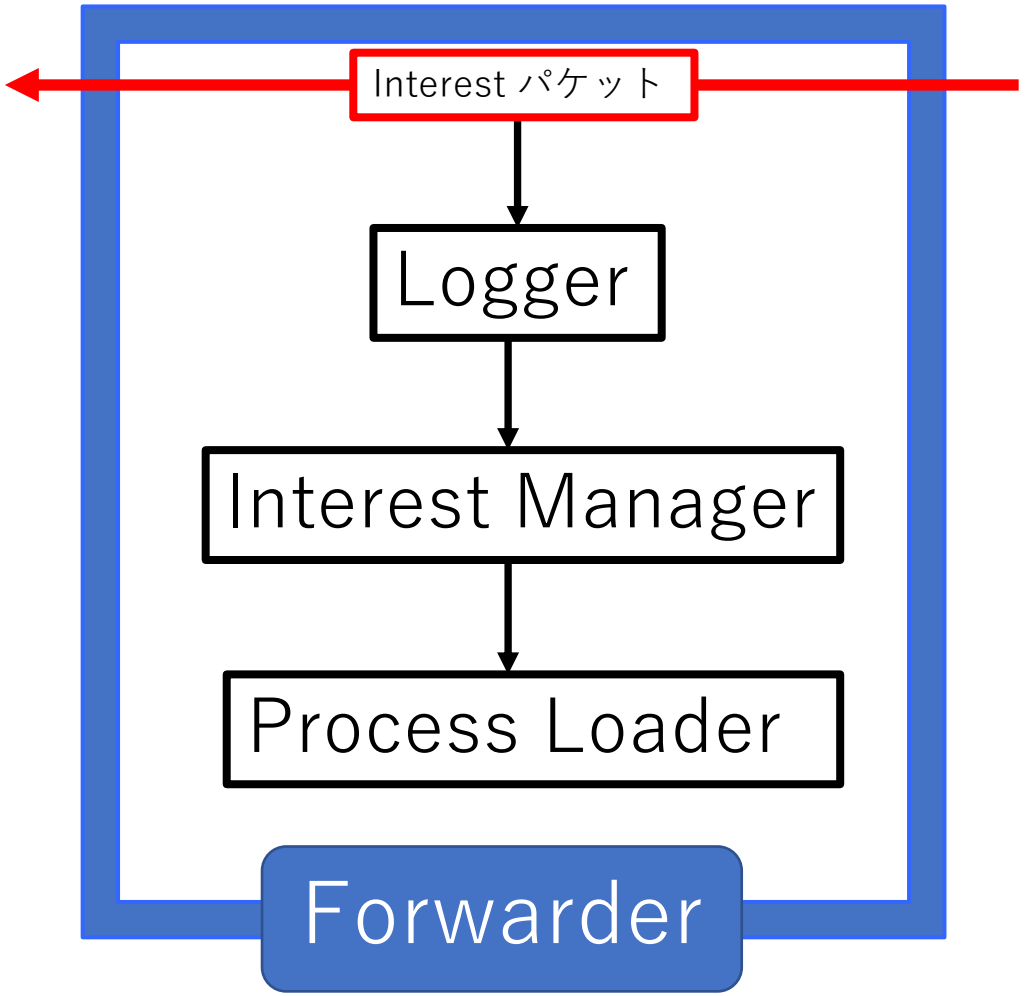
Function Repository ：ファンクションのメタ情報の登録・管理
Consumer ：プログラムを実行
Forwarder ：パケット転送処理及びファンクションのキャッシュ実行
Function Provider ：ファンクションの実行元

```
--------
--------
--------
```

fの実体

3. ファンクションキャッシング概要

大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group

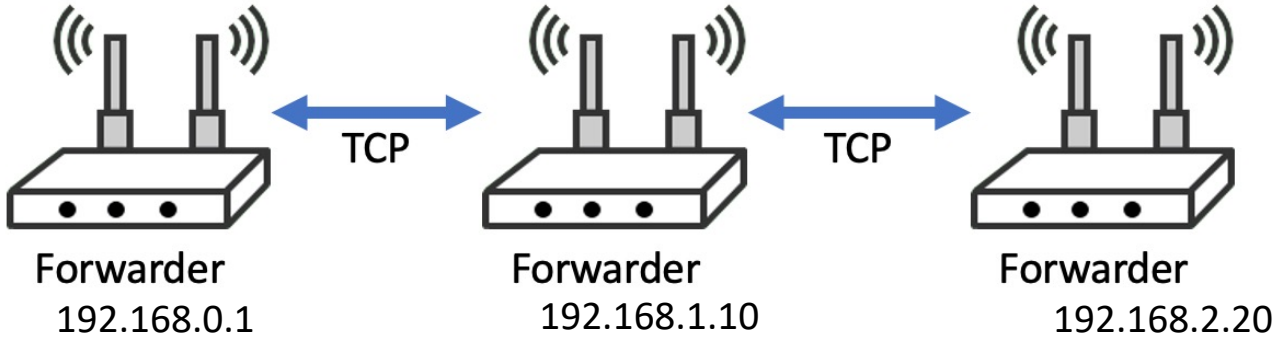


- Logger
  Interest パケットの解析およびファンクション名をInterest Managerへ通知
- Interest Manager
  Interest の関数名ごとにカウント値を管理
- Process Loader
  ファンクションキャッシュを行うと判断されたファンクションを実行

4. Logger における Interest 観測

大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group

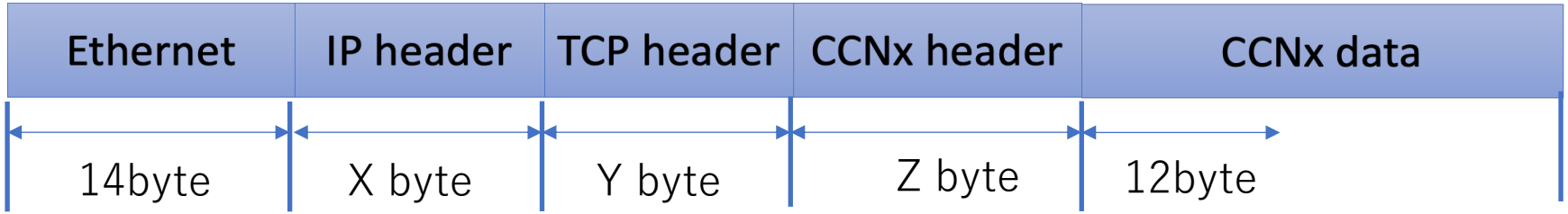
- TCP/IP ネットワーク上で ICN を実装する場合、ICN ノード間でオーバレイネットワークを構成
- TCP/IP パケットのペイロードに CCNx プロトコルのヘッダおよびペイロードが格納されている
- pcap ライブラリによるパケットキャプチャの結果はデータリンク層から取得
- プロトコルスタックに基づき順にパケットヘッダを解析（Ethernet→IP→TCP→CCNx）
- CCNx data 中に含まれるインタレスト名を取得



5.ヘッダ解析の実装

大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group

- Ethernet のヘッダ長は固定（14バイト）
- IP (IPv4) のヘッダ長は可変（1オクテット目の下位4ビットで4バイト単位の数値が格納 = **X**）
- TCP のヘッダ長は可変（13オクテット目の上位4ビットで4バイト単位の数値が格納 = **Y**）
- CCNx のヘッダ長は可変（8オクテット目にバイト単位の数値が格納 = **Z**）
- Interest 名は CCNx ペイロード部に格納：11.12オクテット目に名前の長さ = **N**、13オクテット目から名前本体

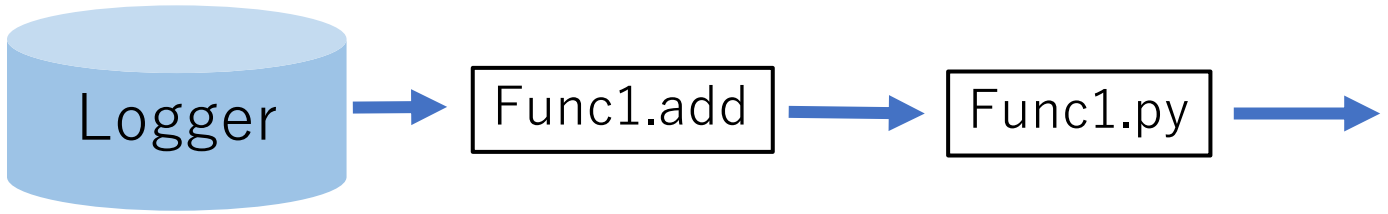


- pcap で取得したデータの 14 + X + Y + Z + 13 バイト目から N バイトを取得することで名前を取得可能

6. ファンクション管理

大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group

- Logger で出力された関数呼び出しのInterest パケットを新規のものはテーブルを追加し、管理テーブルにてカウント
- Logger より抽出されたファンクション名から格納されているファイル名のみを管理テーブルにて保存
- メタ情報のやり取り以外のファンクションが実行された際のもののみをカウントする。
- カウント値があらかじめ設定された閾値を超えた際、閾値を超えた関数名をProcess Loaderへ通知



module name	Count
Func1.py	4
Func2.py	2

Interest Manager での管理テーブル

7. ファンクションのロードから実行まで

大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group

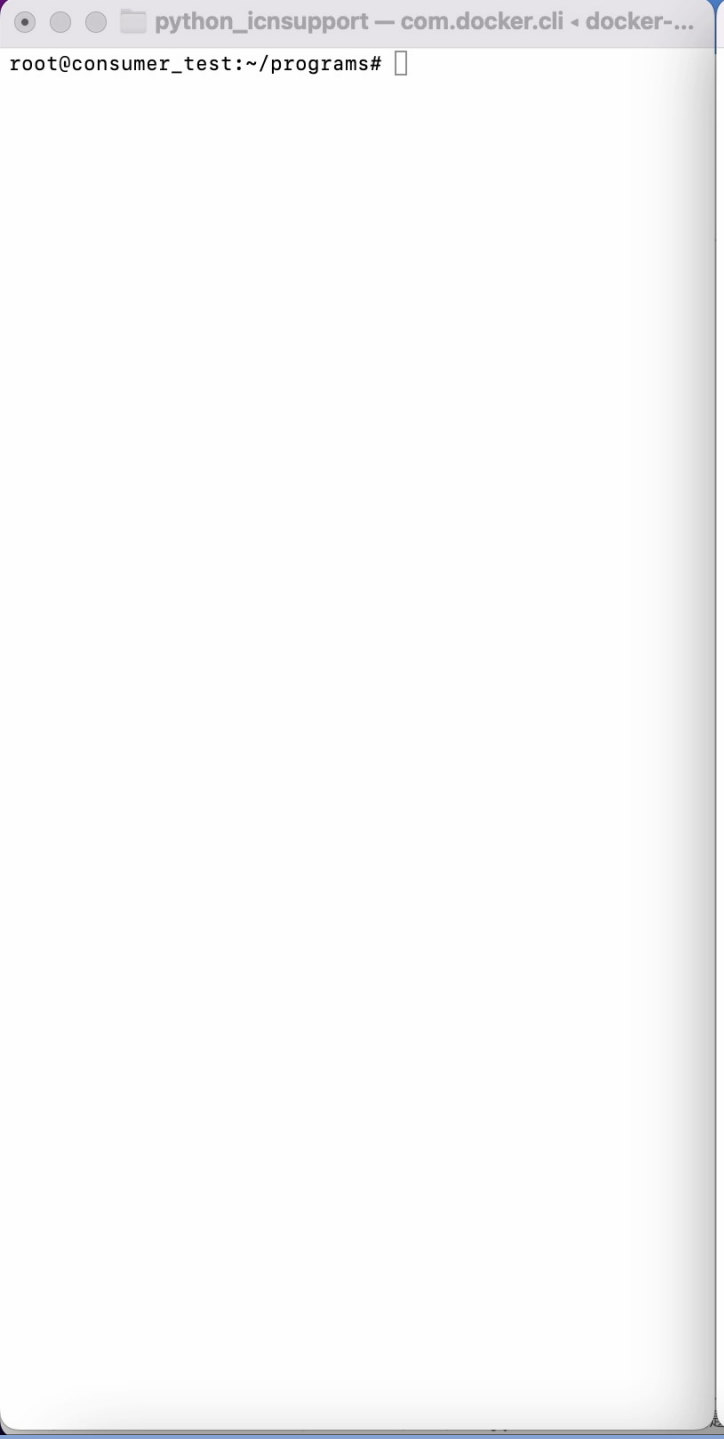
- Interest Manager からの通知を受け次第、インタレストパケットを作成し Function Provider へと送信
- Function Provider は受け取り次第、Ceforeのキャッシュコマンドを用いて該当するファンクションファイルをキャッシュ
- キャッシュコマンドが実行され次第 Forwarder へ通知
- Forwarder は通知を受け取り次第、Cefore のコマンドを用いて Function Provider へキャッシュされているファンクションをロード
- Forwarder はロードを確認次第、ファンクションの処理待ち状態となり、 ロードしたファンクションのinterestパケットを受け取ると処理を行う。

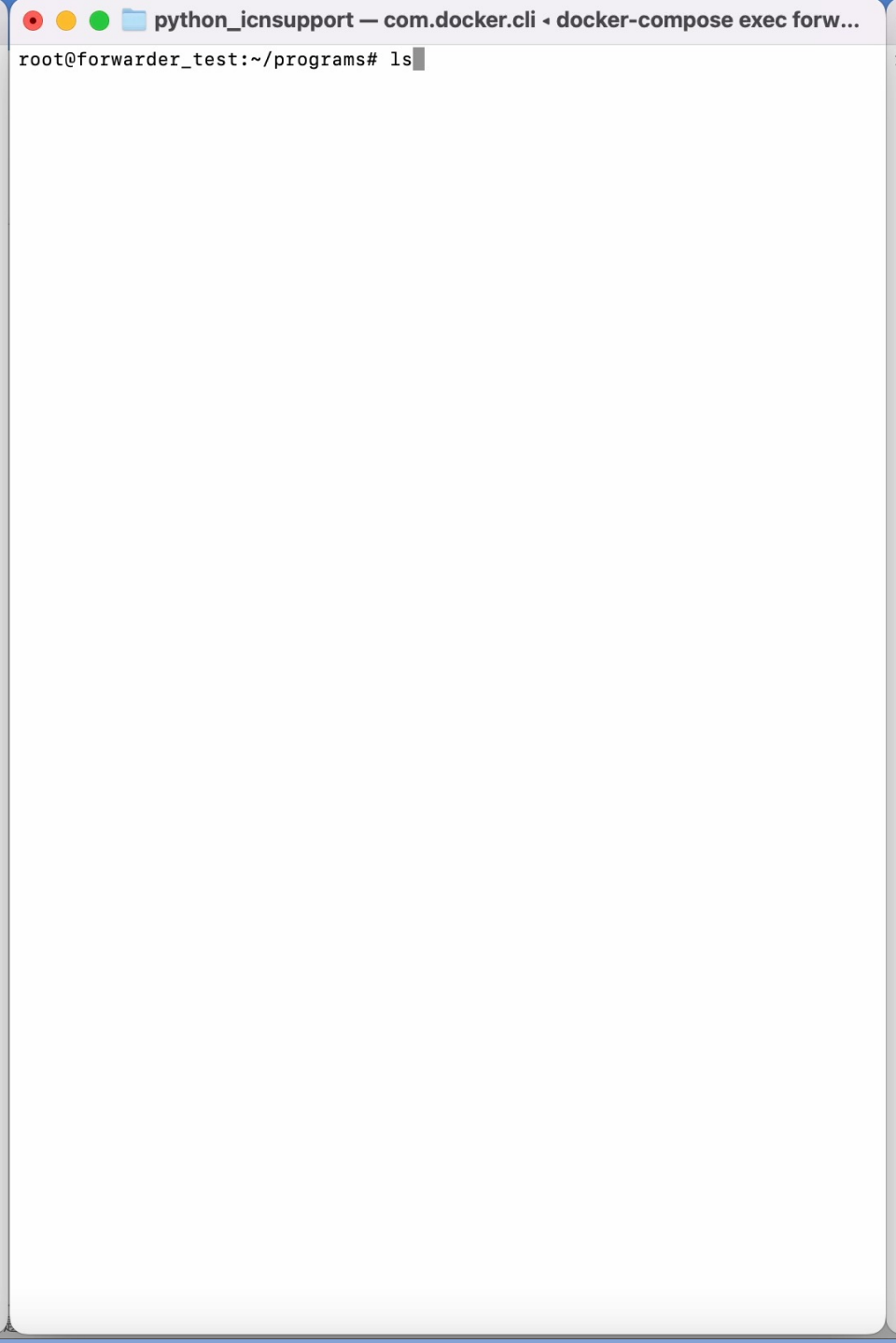
8. Consumerで実行するプログラム

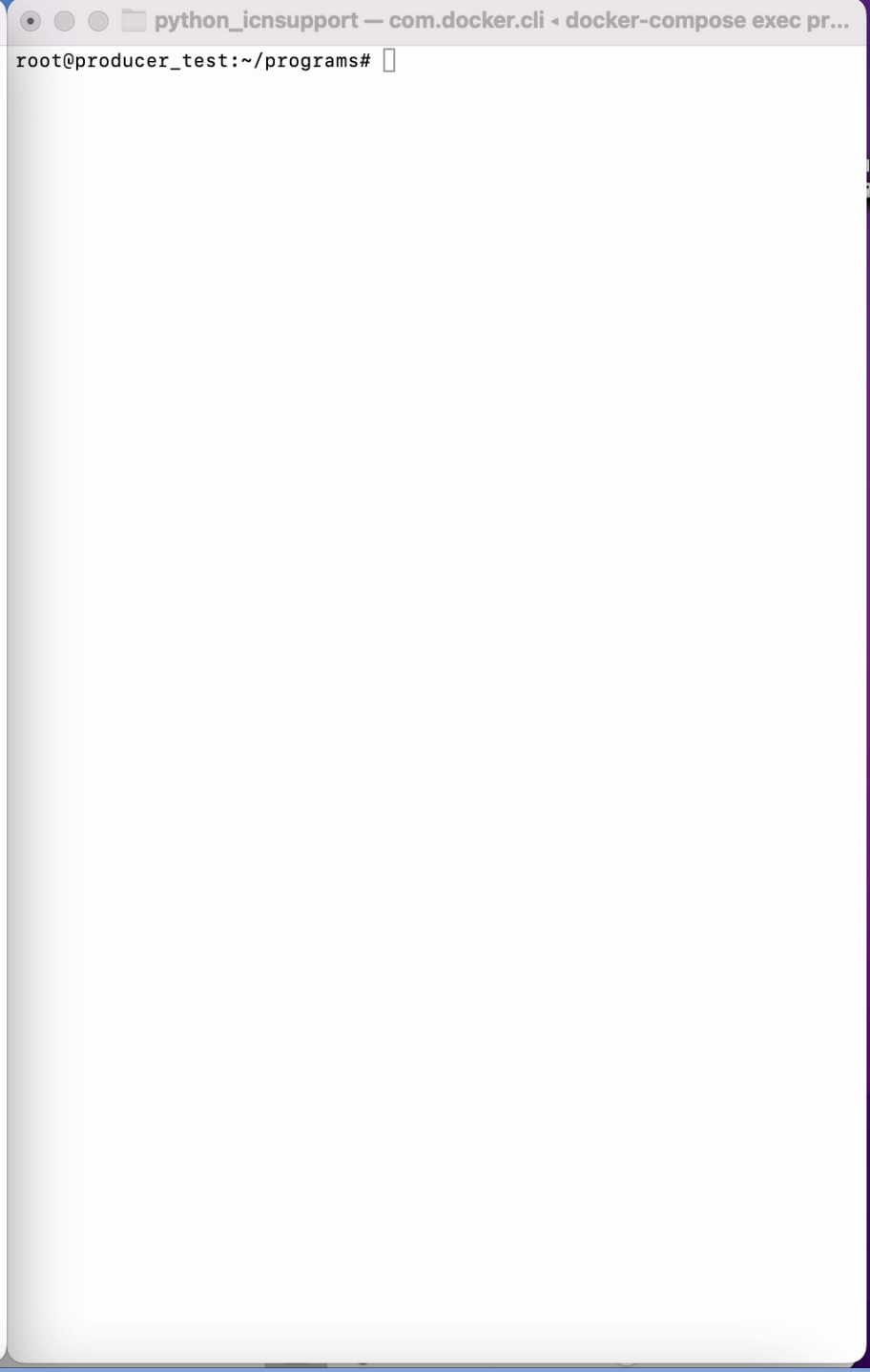
大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group

```python
import random
import numpy
import icnsupport
import statistics

import add
import multi

for a in range(30):
    parm1 = random.randrange(100000)+1
    parm2 = random.randrange(100000)+1
    parm3 = random.randrange(5)+1
    if parm3 == 1:
        ret = multi.array(parm1, parm2)
        print("%d * %d = %s" % (parm1, parm2, ret))
    else:
        ret = add.array(parm1, parm2)
        print("%d + %d = %s" % (parm1, parm2, ret))

```

実行されるプログラムには処理位置は指定されておらず、ローカルにはないモジュールを使用

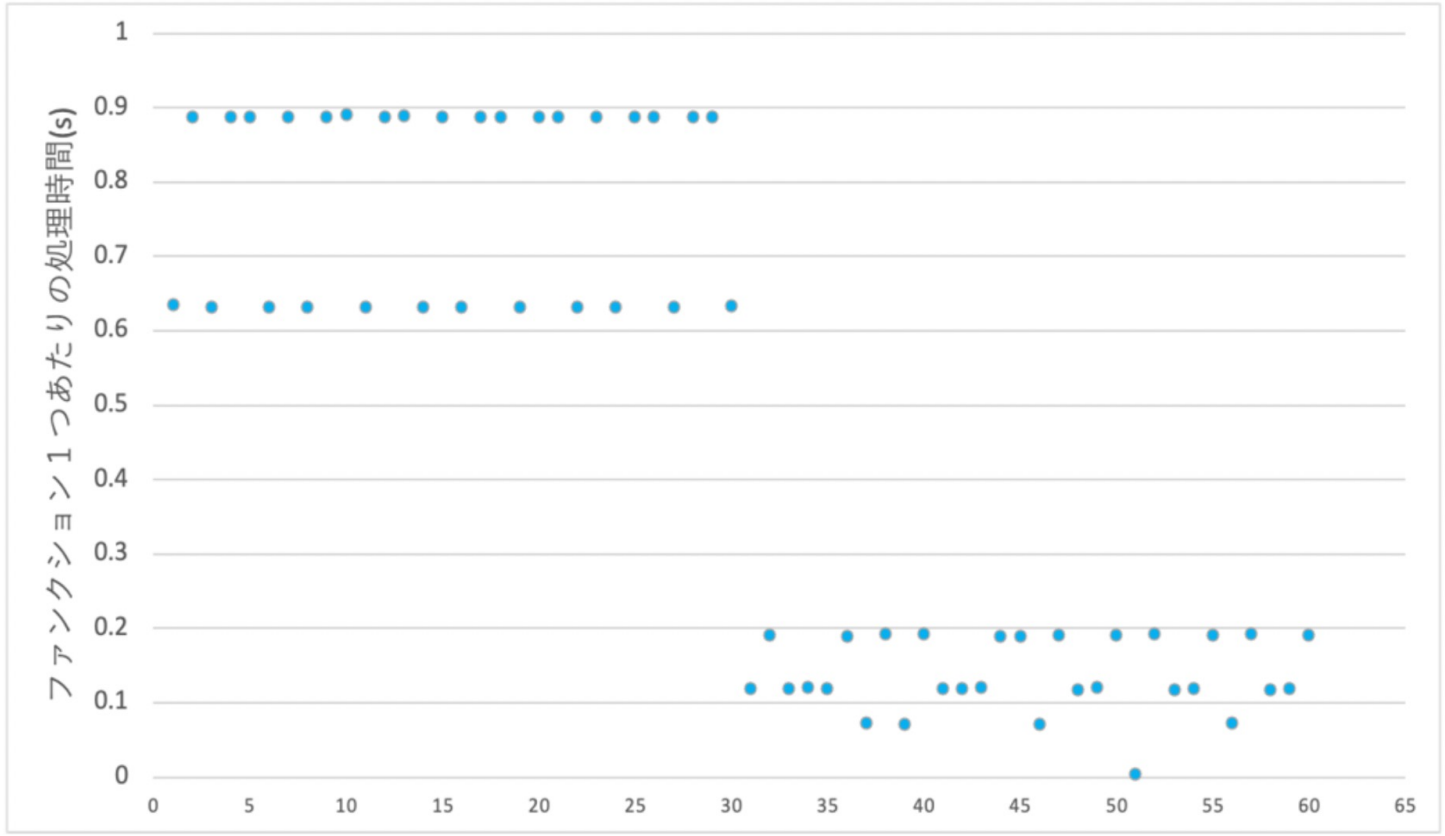

python_icnsupport — com.docker.cli ◂ docker-...
root@consumer_test:~/programs#


python_icnsupport — com.docker.cli ◂ docker-compose exec forw...
root@forwarder_test:~/programs# ls


python_icnsupport — com.docker.cli ◂ docker-compose exec pr...
root@producer_test:~/programs#

9. 評価と考察

大阪公立大学 大学院情報学研究科
スマートプラットフォーム研究室
Graduate School of Informatics, Smart Platform Research Group



閾値を30とした時のファンクションの処理時間

- Producerでの実行に比べ、Forwarderでの実行時間の減少が見られた
- 一部処理時間にまとまりが見られたのは、引数の大きさが要因と見られる

10. 考察と課題

- 呼び出すファンクションの種類の増加

  処理できるファンクションの種類を増加させ、より効率的なエッジコンピューティングを目標とする

- ファンクション呼び出し機構の効率化

  ファンクションを呼び出すまでの呼び出すプロセスの数を減少させることでファンクションを呼び出すまでの時間の削減を目指す