

研究概要

背景

- エミュレータを活用して大規模IoTシステムテストを実施
- 多数のコンピュータ群で構成されたネットワークテストベッドStarBED上で実施

課題

- 大規模IoTエミュレーション環境の構築作業が煩雑



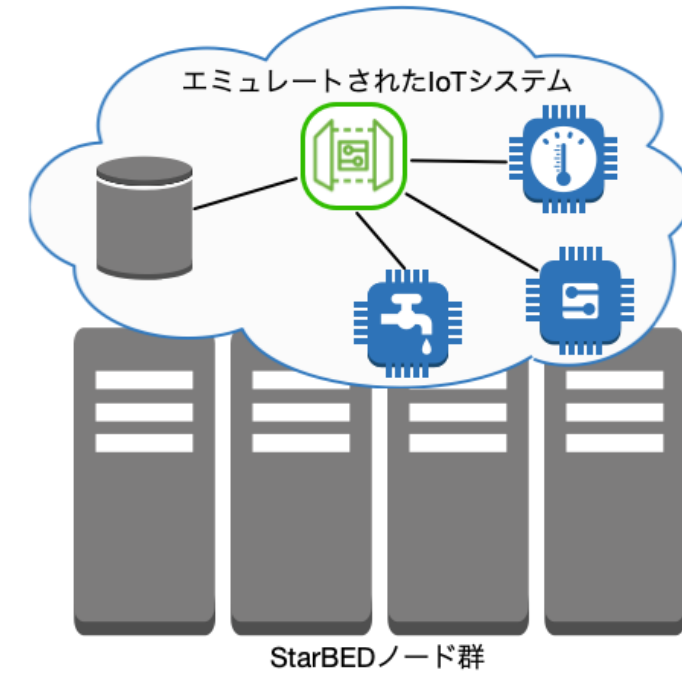
大規模実験環境を簡単に構築するための機能を開発

実験環境構築機能

- 各エミュレータの実行にDockerコンテナを利用
 - Docker Swarm を用いて環境構築機能を設計・実装
 - 設定ファイル：YAML形式
 - コントロールスクリプト：python
- ⇒ 実行するStarBEDノード(ホスト)数を柔軟に変更可能

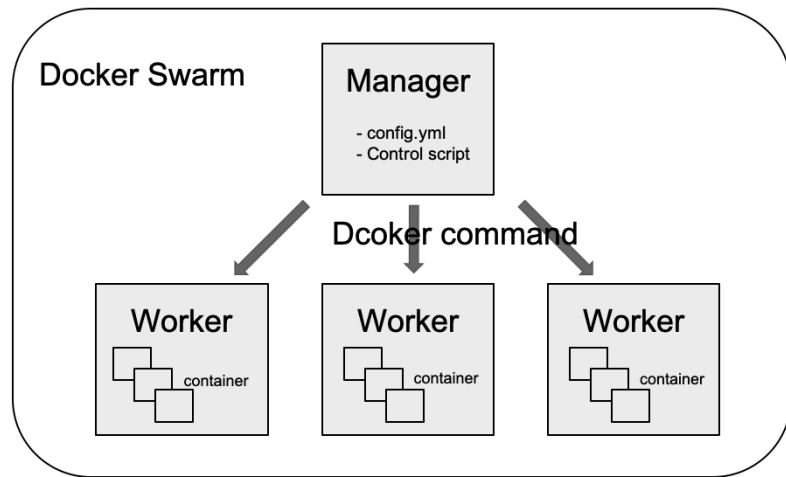
評価

- 環境構築に要する時間を計測
- ⇒ 複数ノードでの分散構築によって構築時間が短縮



ネットワークテストベッドStarBED

全体構成



Manager:管理ノード Worker:実行ノード

- 1つのManagerと複数のWorkerで構成
- 1つのWorkerが1つの物理ノードに対応
- 1つのエミュレータが1つのコンテナに対応
- Managerには
 - 設定ファイル (config.yml)
 - コントロールスクリプト(Python)
 が配置
- エミュレータ(Dockerコンテナ)はWorker上で実行
- コントロールスクリプトが環境構築を実施

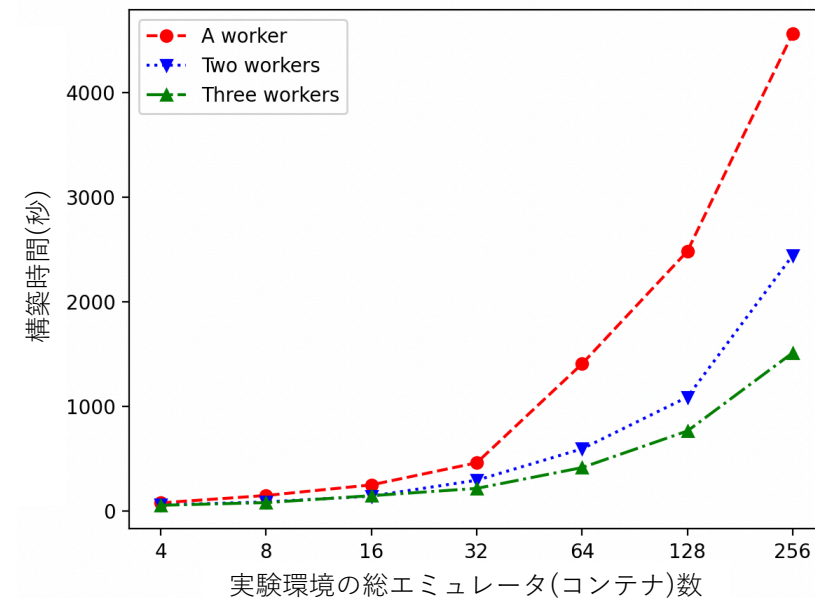
設定ファイル

```

p118:
  subnet: "192.168.0.0/16"
  http_nodes:
    server: "192.168.0.2"
  gateway_nodes:
    GW-1:
      ip: "192.168.0.100"
      volume: "p118sock1"
      sensor_nodes:
        GroupA: 3
        GroupB: 2
    GW-2:
      ip: "192.168.1.100"
      volume: "p118sock2"
      sensor_nodes:
        GroupC: 3
        GroupD: 1
p120:
  subnet: "192.169.0.0/16"
  http_nodes:
    server: "192.169.0.2"
  gateway_nodes:
    GW-3:
      ip: "192.169.0.100"
      volume: "p120sock"
      sensor_nodes:
        GroupE: 1
        GroupF: 2
  
```

- 構築する実験環境の情報をYAML形式で記載
 - ネットワーク設定
 - 各Workerでのエミュレータ台数

評価：Worker数に応じた実験環境構築時間の比較



- Worker数1~3で実験環境構築時間を比較
- Worker数が増えるほど環境構築時間が短縮
- 構築するエミュレータ数が増えるほど、削減できる時間は増加

今後の展望

- 構築済みの実験環境を管理する機能の追加
- 実験実行時のログ収集機能の追加