



国立研究開発法人
情報通信研究機構

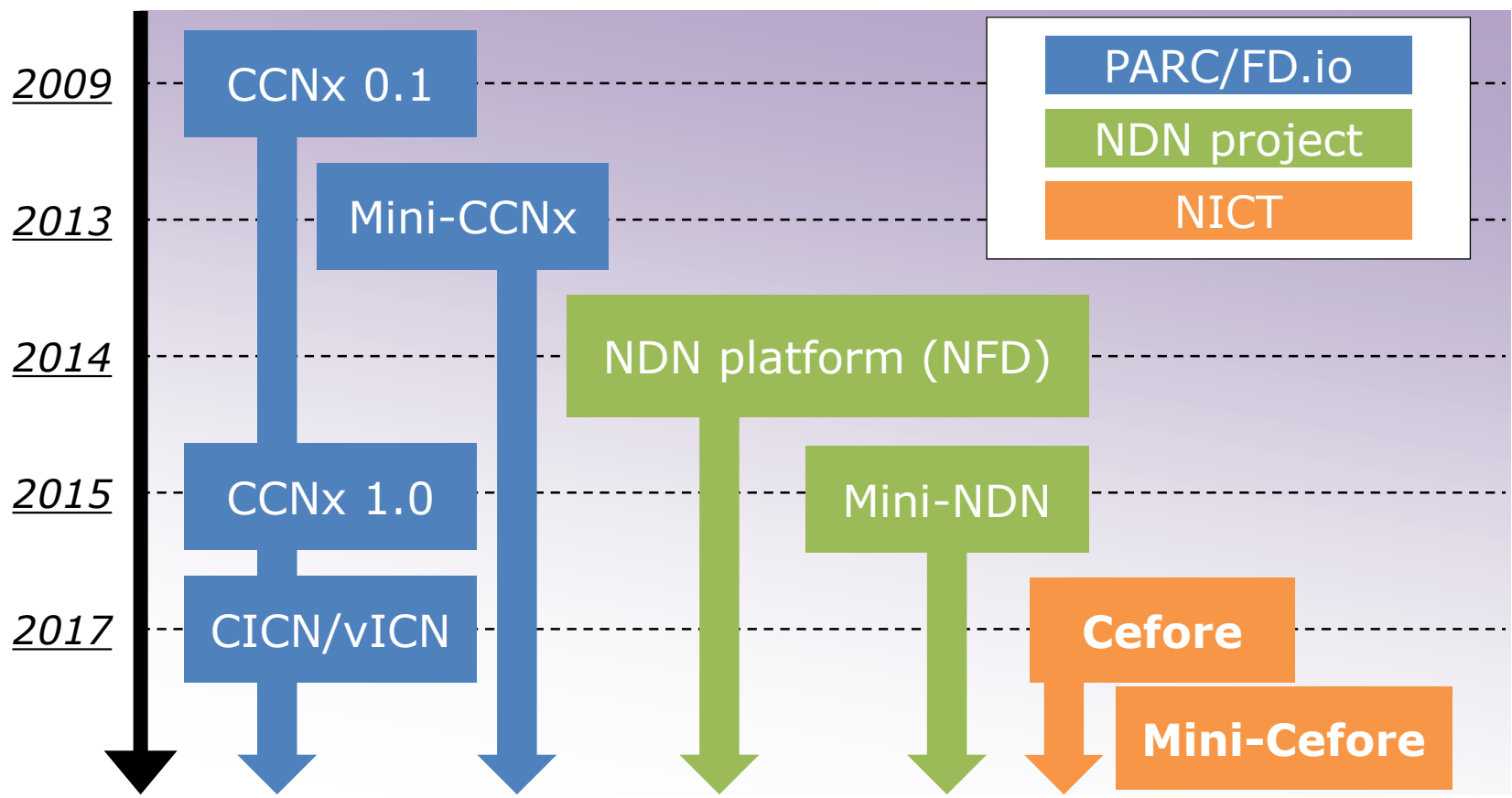
Mini-Cefore: Container-Based Large-Scale Cefore Emulator

大岡 睦, 朝枝 仁

National Institute of Information and
Communications Technology (NICT)

- 背景
- 実験プラットフォームの比較
 - テストベッド・シミュレーター・エミュレーター
- エミュレーターの実装方式の比較
 - VM (Virtual Machine) 方式・コンテナ方式
- Mini-Ceforeの概要と使用方法
- ユースケース：ネットワーク内キャッシュの評価
- 拡張性評価

ICN/CCN 技術実装の活発化



現実の環境における実証実験の機運の高まり

■ Cefore 用のネットワークエミュレーター

■ 現実的なCCN実験環境を提供

- シミュレーションモデルではなく、コードが実際に動く

■ 軽量かつ拡張性の高いコンテナ方式

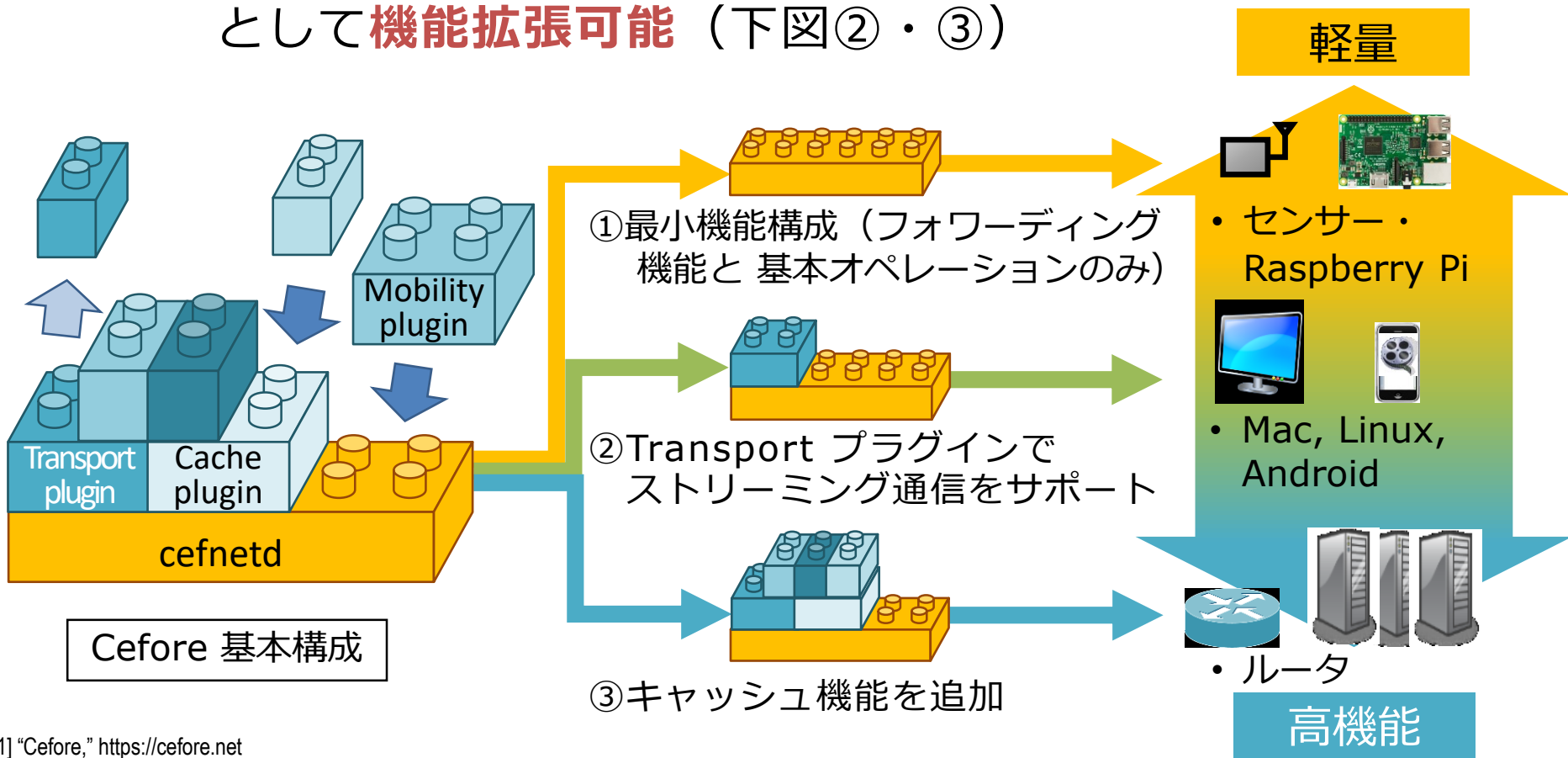
- 高速にネットワークをエミュレート
 - 100ノードを数秒、1,000ノードを約1分で起動可能

■ 簡単な設定ファイルで大規模ネットワーク実験が可能

- 支援ツールによって必要なファイルを自動生成し、エミュレーションを自動実行

軽量かつ汎用的な CCN ソフトウェア実装

- リソースの乏しいセンサーノードでは**軽量構成**（下図①）
- 最小機能以外はプラグインまたは外部機能として**機能拡張可能**（下図②・③）



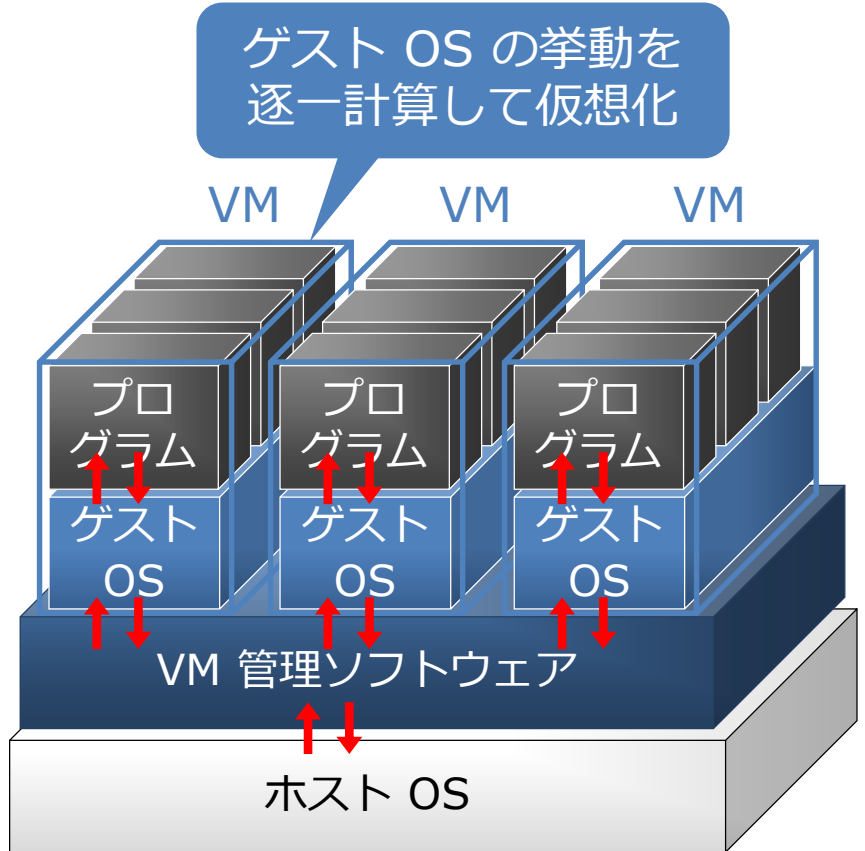
[1] "Cefore," <https://cefore.net>

- 現実性：実際のコードが実時間で動作する
- 拡張性：大規模なネットワークを構築できる
- 柔軟性：ノードやリンクの性能を容易に設定できる

	テストベッド	シミュレーター	エミュレーター
現実性	高	低	高
拡張性	低	高	高
柔軟性	低	中	高
実装例	NDN testbed, CUTEi	ndnSim, ccnSim	Mini-CCNx, Mini-NDN, vICN, Mini-Cefore

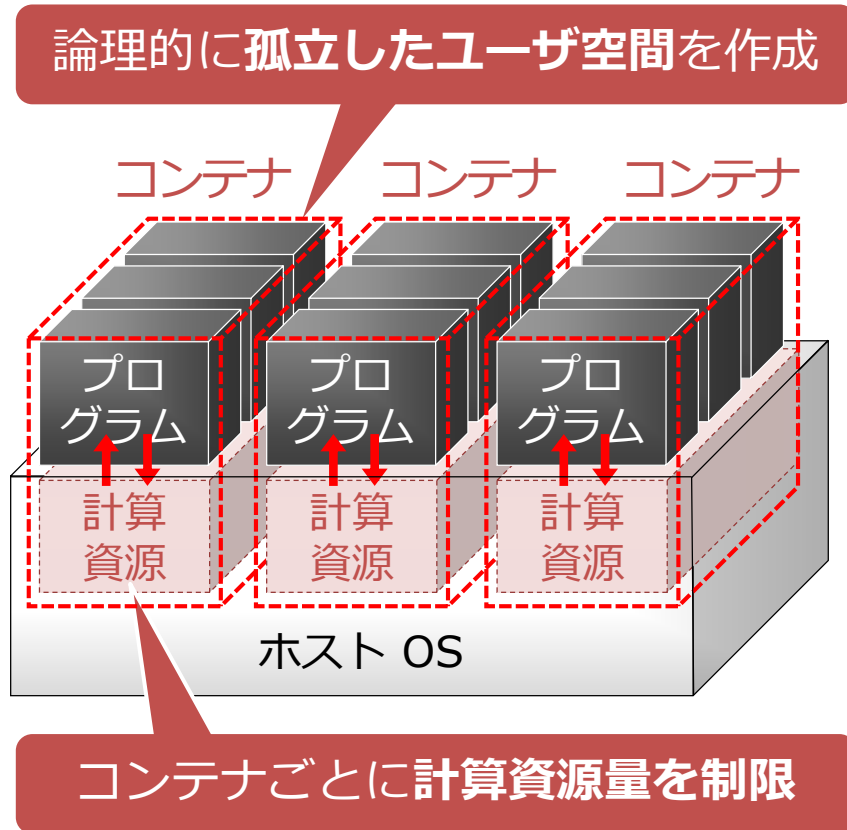
■ Virtual Machine (VM) 方式

- VM 起動やその上のプログラム実行が**低速**
- 多数の VM 起動・管理が**煩雑**



■ コンテナ方式

- コンテナ起動やその上のプログラム実行が**高速**
- コンテナ大量起動が**容易**



Mininet を Cefore 用に改造・機能追加

Mininet



Mininet-HiFi



Mini-CCNx

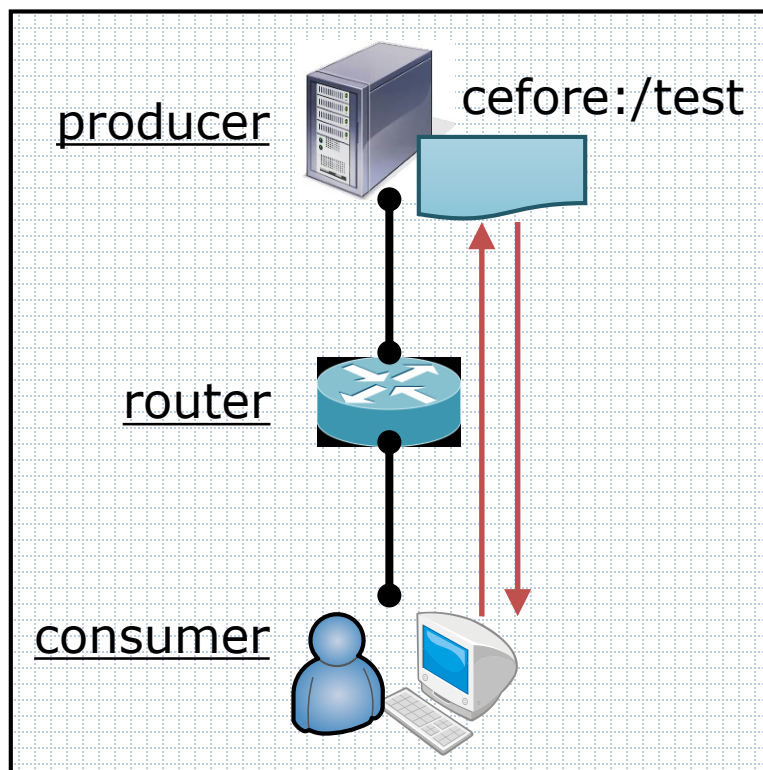


Mini-Cefore

- **コンテナ方式のネットワークエミュレーター**
- Software Defined Networking (SDN) 用に開発
- コンテナごとの **CPU・メモリ資源量を制限**
- コンテナ間の干渉を無くし、**帯域等の信頼性を確保**
- CCNx が各ノードで起動するように改造
- **CCN ネットワーク作成を支援する機能を追加**
- **Cefore** が各ノードで起動するように改造
- **大規模シナリオ実行を支援する機能を追加**

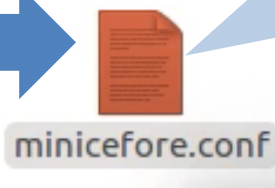
開発の変遷

1. 設定ファイル minicefore.conf を記述
2. "sudo minicefore" を実行
3. コマンドラインインターフェース (CLI) で操作



説明用トポロジー

1. 設定ファイルを記述



minicefore.conf

```

minicefore.conf (/tmp/Mini-CeforeTest) - gedit
開く(O)  x  minicefore.conf
1 [preferences]
2
3 [nodetypes]
4 default: ./cefore
5
6 [hosts]
7 producer:
8 consumer: _ cefore:/,router
9
10 [routers]
11 router: cefore:/,producer
12
13 [links]
14 producer:router
15 consumer:router
16
    
```

2. sudo minicefore

3. CLI で操作

```

Mini-CeforeTest
$ sudo minicefore
Parse of minicefore.conf done.
*** Creating network
*** Adding hosts:
consumer producer router
*** Adding links:
(consumer, router) (producer, router)
*** Configuring hosts
consumer producer router
*** Launch hosts
consumer producer router
Setup time: 0

*** Done

*** Starting CLI:
minicefore>
    
```

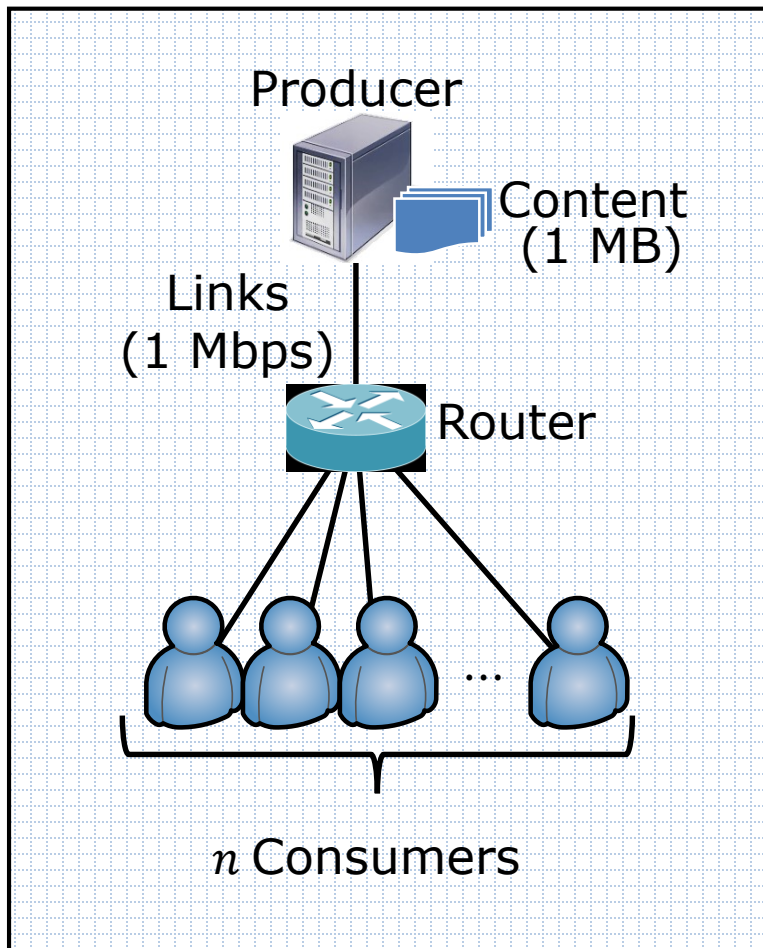


cefore

1. キャッシュ有効時・無効時のダウンロード時間を評価
 - 同時ダウンロード数の変化に伴うダウンロード時間の変動を**忠実に再現**

2. 大規模ネットワークでキャッシュ置換方式の性能比較
 - 大規模エミュレーション支援ツールを用いて必要なファイルを準備し、**容易に実験可能**

キャッシュ有効時・無効時のダウンロード時間を評価



評価用トポロジー

```
[preferences]

[nodetypes]
default: ./cefore
nocache: ./nocache_cefore

[hosts]
prod: _ cpu=0.8
router: _ cpu=0.8 cefore: ./prod
cons01: _ type=nocache cpu=0.2 cefore:/,router
cons02: _ type=nocache cpu=0.2 cefore:/,router
cons03: _ type=nocache cpu=0.2 cefore:/,router
cons04: _ type=nocache cpu=0.2 cefore:/,router
cons05: _ type=nocache cpu=0.2 cefore:/,router

[routers]

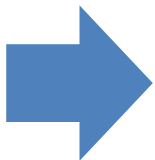
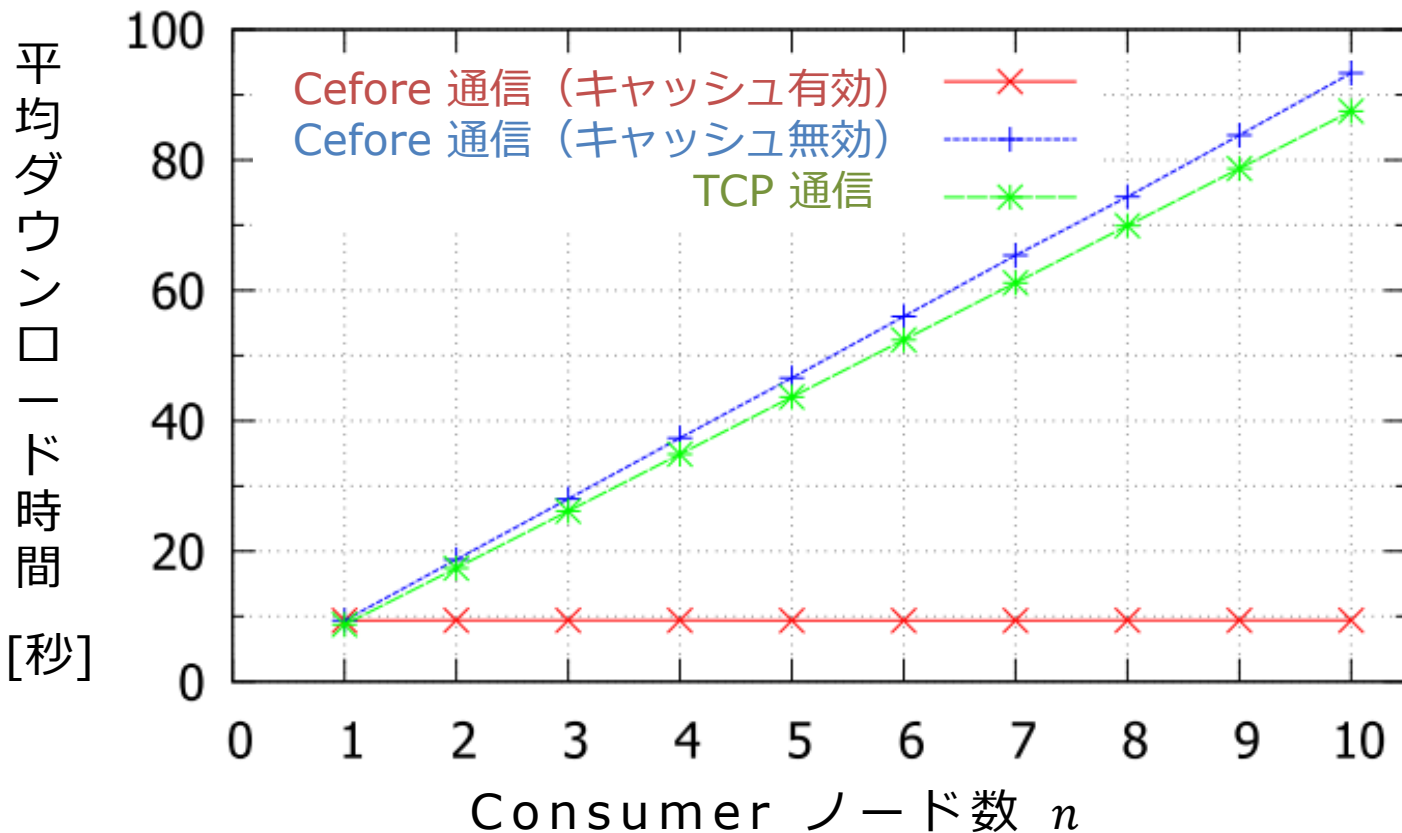
[links]
prod:router bw=1
cons01:router bw=1
cons02:router bw=1
cons03:router bw=1
cons04:router bw=1
cons05:router bw=1
```

“cpu=0.2”
信頼性向上のために
CPU 使用率を制限

“bw=1”
リンクの帯域を
1 Mbps に制限

設定ファイル minicefore.conf
(n = 5 の場合)

キャッシュ有効時・無効時のダウンロード時間を評価

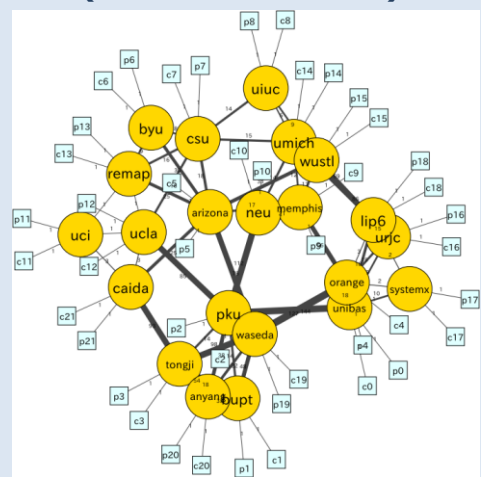


キャッシュ有効時は一定時間、
キャッシュ無効時は n に比例した
時間がかかる傾向を**忠実に再現**

大規模ネットワークでキャッシュ置換方式の性能比較

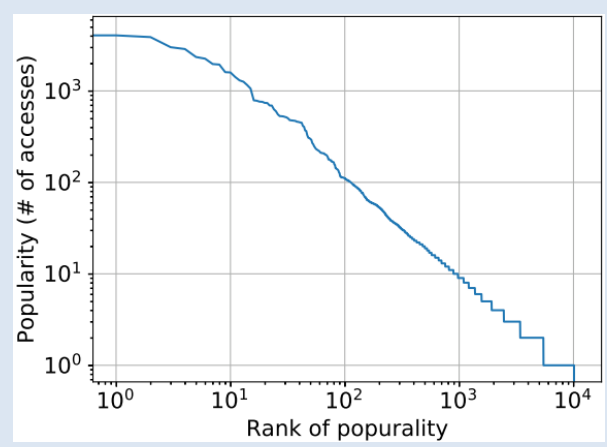
トポロジー設定

CAIDA トポロジー
(全 66 ノード)



コンテンツ・要求分布設定

- Zipf 分布 + RankShift
- ポアソン到着過程



キャッシュ方式設定

- Cache Everything Everywhere
- FIFO, Random, LRU, LFU, CLOCK, CUSH[2]
- キャッシュサイズ : 100KB, 1MB, 10MB

↓
18 シナリオ

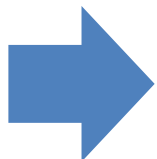
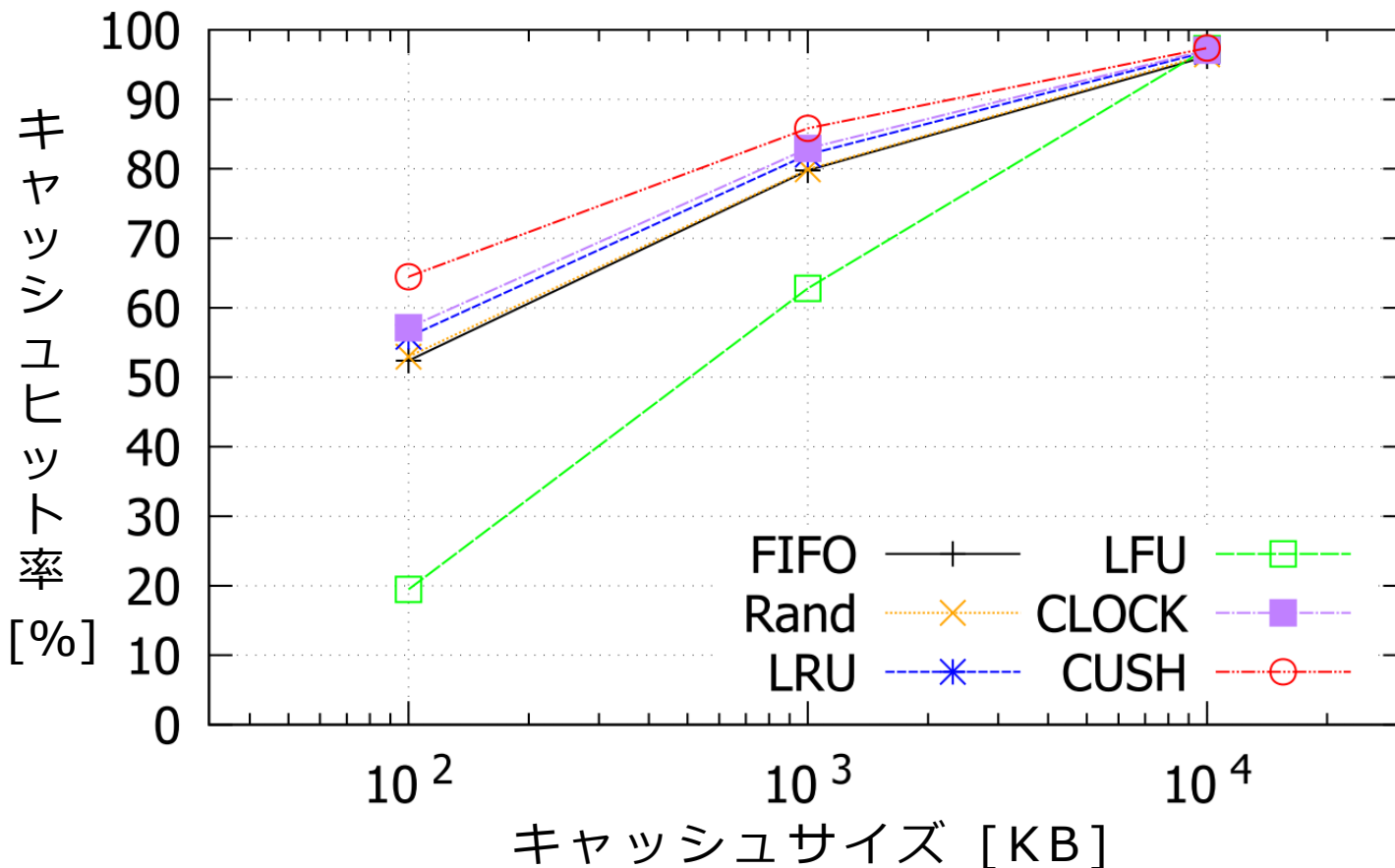
↓ **大規模エミュレーション支援ツールで自動生成・実行**

Mini-Cefore 用設定ファイル・自動実行用スクリプト

各ノードの Cefore 用設定ファイル・アプリケーション

× シナリオ数

大規模ネットワークでキャッシュ置換方式の性能比較



大規模ネットワークでの評価を容易に実行

Mini-Cefore の拡張性評価

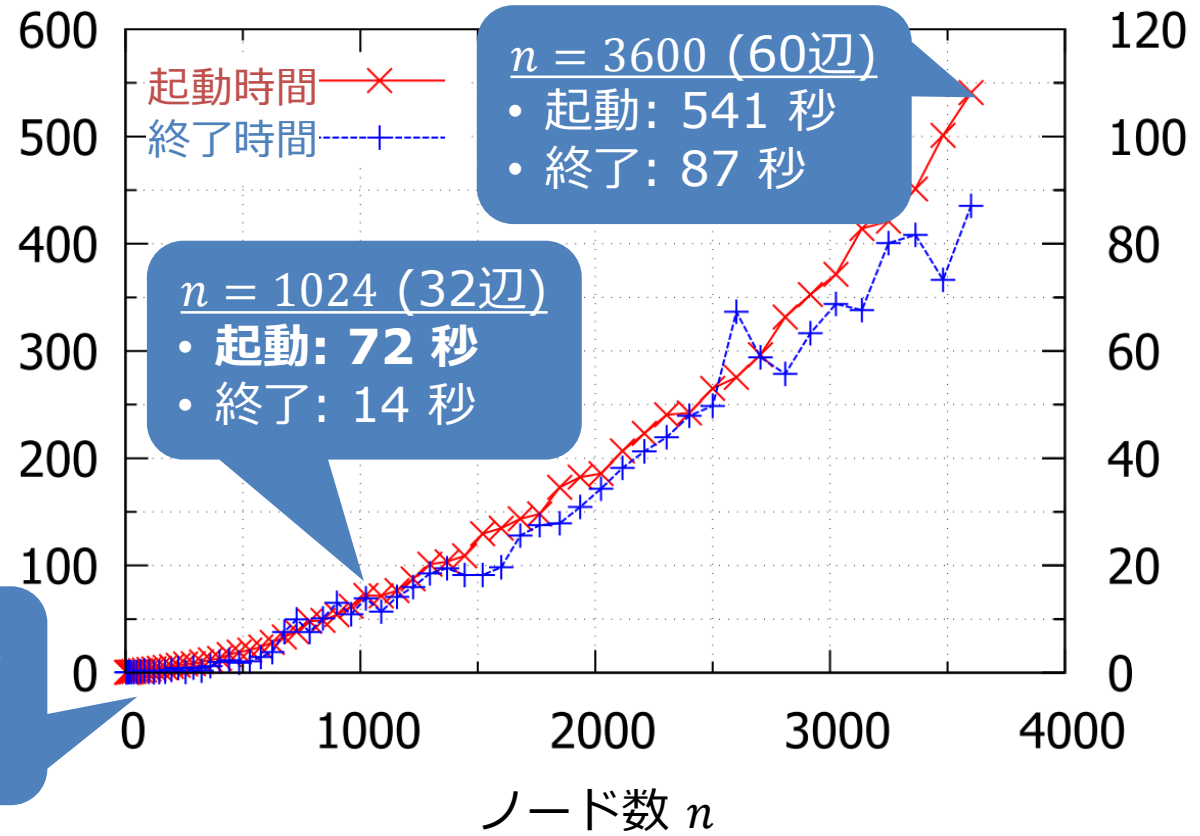
Mini-Cefore が格子状トポロジー（2辺～60辺）の起動・終了に要した時間を評価

実験 PC 性能

- Ubuntu 16.04 LTS
- Intel Xeon E6-2690 (2.90 GHz)
- 64 GB RAM
- 1 TB hard disk

起動時間 [秒]

終了時間 [秒]



100ノードなら3秒、
1,000ノードでも約1分で起動可能

- Cefore 用のネットワークエミュレーター **Mini-Cefore**を開発
 - 現実性・拡張性・柔軟性が高い
- **軽量**かつ拡張性の高いコンテナ方式
 - 100ノードを数秒、1,000ノードを約1分で起動可能
- 簡単な設定ファイルで**大規模ネットワーク実験**が可能
 - Cefore のプラグイン機能と組み合わせることで機能拡張も容易
- <https://cefore.net/> にて近日公開予定

