

## 時空間データ利活用に向けた F-CPS 基盤の実環境における性能評価

田中 明弥<sup>†</sup> 田中 直樹<sup>†</sup> 赤司 廉樹<sup>†</sup> 川口 光輝<sup>†</sup> 野林 大起<sup>††</sup>  
塚本 和也<sup>†††</sup> 池永 全志<sup>††</sup> 関川 柊<sup>††††</sup>

<sup>†</sup>九州工業大学大学院 工学府 〒804-8550 福岡県北九州市戸畑区仙水町 1-1

<sup>††</sup>九州工業大学大学院 工学研究院 〒804-8550 福岡県北九州市戸畑区仙水町 1-1

<sup>†††</sup>九州工業大学大学院 情報工学研究院 〒820-8502 福岡県飯塚市川津 680-4

<sup>††††</sup>株式会社 KDDI 総合研究所

E-mail: <sup>†</sup>{tanaka.meiya767,tanaka.naoki845,akashi.renju810,kawaguchi.koki851}@mail.kyutech.jp,  
<sup>††</sup>{nova,ike}@ecs.kyutech.ac.jp, <sup>†††</sup>sukamoto@csn.kyutech.ac.jp, <sup>††††</sup>sh-sekigawa@kddi.com

**あらまし** 地域特化型のサイバーフィジカルシステム (CPS) として、発生場所や時間に依存する時空間データ (STD) とそれを処理するための機能を、特定の地域内に滞留させるフローティングサイバーフィジカルシステム (F-CPS) を提案している。F-CPS では「時空間データ滞留システム」により、STD を生成された場所から直接配信し、その場に留めておく。また、STD を用途に応じて柔軟に利活用するために必要な機能 (アプリケーション) は、ハードウェアや OS に依存せず様々なデバイス上で実行できる必要があるため、データサイズが軽量かつ Web 基盤上で実行可能な WebAssembly コンテナを利用する。本稿では、大学施設内の環境情報を STD として滞留させ、Wasm を活用した機能により分析・表示する F-CPS 基盤の統合試験を実施し、F-CPS によるデータの地産地消の実現可能性を評価する。

**キーワード** 時空間データ, F-CPS, 情報滞留, WebAssembly, データの地産地消

## Experimental Evaluation of F-CPS Platform for Spatio-Temporal Data Utilization

Meiya TANAKA<sup>†</sup>, Naoki TANAKA<sup>†</sup>, Renju AKASHI<sup>†</sup>, Koki KAWAGUCHI<sup>†</sup>, Daiki NOBAYASHI<sup>††</sup>,  
Kazuya TSUKAMOTO<sup>†††</sup>, Takeshi IKENAGA<sup>††</sup>, and Shu SEKIGAWA<sup>††††</sup>

<sup>†</sup> Graduate School of Engineering, Kyushu Institute of Technology

1-1 Sensui-cho, Tobata-ku, Kitakyushushi, Fukuoka, 804-8550 Japan

<sup>††</sup> Faculty of Engineering, Kyushu Institute of Technology

1-1 Sensui-cho, Tobata-ku, Kitakyushushi, Fukuoka, 804-8550 Japan

<sup>†††</sup> Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology

680-4, Kawadu, Iizukashi, Fukuoka, 820-8502 Japan

<sup>††††</sup> KDDI Research, Inc.

E-mail: <sup>†</sup>{tanaka.meiya767,tanaka.naoki845,akashi.renju810,kawaguchi.koki851}@mail.kyutech.jp,  
<sup>††</sup>{nova,ike}@ecs.kyutech.ac.jp, <sup>†††</sup>sukamoto@csn.kyutech.ac.jp, <sup>††††</sup>sh-sekigawa@kddi.com

**Abstract** A Floating Cyber-Physical System (F-CPS) has been proposed to promote flexible data utilization by retaining Spatio-Temporal Data (STD), which depends on the location and time of data generation, and the functions (applications) within the specific area. In order to retain of STD, we have proposed the "STD Retention System" that distributes them directly from the place where they are generated and retains them there. Furthermore, in F-CPS, the use of WebAssembly container enables the realization of application execution with small-size functions regardless of hardware and operating systems. In this paper, we conducted integrated experiments from the retention of STD to the execution of function using actual environmental information within college facilities to evaluate the feasibility of local production and consumption of data by F-CPS.

**Key words** Spatio-Temporal Data, F-CPS, Data Retention, WebAssembly, Local Production and Consumption of Data

## 1. はじめに

Internet of Things (IoT) やクラウド技術、AI 技術の発展と普及に伴い、現実世界の様々な課題を解決するネットワークアーキテクチャとしてサイバーフィジカルシステム (Cyber Physical System: CPS) の実現が期待されている。これはスマートフォンや車両、センサ等の IoT デバイスによって収集された現実空間 (フィジカル空間) の情報をサイバー空間に集積・分析し、その分析結果や結果に基づくサービスを現実世界にフィードバックすることで課題解決を目指すシステムである。CPS 実現に向けて、効率的にデータを収集・配信・活用するための、Beyond 5G/6G などの次世代無線通信技術や新たなネットワーク基盤が必要となる。

一方で、現実空間において生成されるデータの中には、交通情報や災害情報、気象情報、時限的な広告情報といったデータの発生場所や時間に依存する「時空間データ (Spatio-Temporal Data: STD)」が存在する。我々の研究グループでは、地域特化型の CPS として、特定の地域の時空間データとそれを処理・活用する「機能 (アプリケーション)」を滞留させるフローティングサイバーフィジカルシステム (Floating Cyber Physical System: F-CPS) を提案している [1]-[3]。F-CPS により、STD を地産地消することで、データ流通を活性化させ、効果的なデータ利活用の実現を目指す。

図 1 に F-CPS の概要を示す。F-CPS におけるデータ流通手段として、特定の地域内のユーザが所有する通信・計算能力を有したデバイスエッジを用いて、データを拡散・維持する「時空間データ滞留システム」が提案されている [4]-[8]。時空間データ滞留システムは、地域内のデバイスエッジが周期的なブロードキャスト配信を繰り返すことによって、遠隔地に設置されたサーバを経由せず、STD をユーザに直接提供する。このシステムにより、STD が滞留するエリアに存在するユーザは、データを検索することなく受動的に情報を取得可能となる。

また、F-CPS では、特定の場所、時間における柔軟なデータの利活用を目的に、ユーザが受信した STD を有効活用する手段として、STD を処理する「機能」の滞留も提案している。この機能は、不特定多数かつ多種多様なデバイスエッジ上において共通のアプリケーションとして実行する必要があるため、ハードウェアやオペレーティングシステム (Operating System: OS) に依存しないシステムである必要がある。現状における一般的な実現手段として、仮想化技術の一つであるコンテナの利用が考えられる。しかし、無線通信を用いて不特定多数の地域内のデバイスにアプリケーションを配信する F-CPS では、データサイズが大容量となるコンテナを利用した場合、機能及びデータ流通に支障が出る可能性がある。また、コンテナ技術では、ハードウェアや OS 毎に、それぞれに応じた設定ファイルやライブラリを含むコンテナを作成する必要がある。そこで、我々の研究グループでは、F-CPS の実現に向けて様々なデバイスエッジで実行可能で、軽量かつ汎用性の高いアプリケーション実行基盤である WebAssembly (Wasm) の活用を提案している [9][10]。機能及びその実行基盤に Wasm を活用することで、多くのデバ

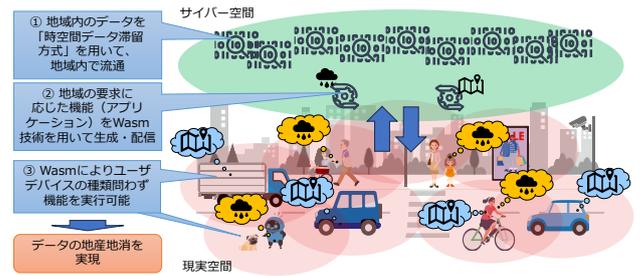


図 1 F-CPS の概要

イスエッジが共通して有する Web 基盤上で、機能を利用可能となり、アーキテクチャやカーネルに依存せず実行可能なシステムを実現する。

本稿では、F-CPS 基盤によるデータの地産地消の実現可能性を評価するため、時空間データ滞留システムと Wasm コンテナによるアプリケーション実行基盤の統合試験を実施する。九州工業大学施設内の各種環境情報を STD とし、時空間データ滞留システムによる STD の滞留から、一定のエリア内をカバーする Multi-access Edge Computing (MEC) サーバ経由での機能取得、デバイスエッジでの機能実行までの一連の動作を検証すると共に、統合試験を通じて、時空間データ滞留システムや、Wasm アプリケーションとその実行について検証・評価を行う。

以下、2 節では関連研究、3 節では時空間データ滞留システム、4 節では F-CPS における機能について述べる。5 節では統合試験における検証結果と評価を示し、最後に、6 節でまとめを述べる。

## 2. 関連研究

本節では関連研究として、移動ノードにおける通信制御手法に関する研究や、ハードウェアに制約のある IoT デバイスにおけるコンテナとしての、Wasm の適用可能性に関する研究について述べる。

### 2.1 移動ノードにおける情報伝達制御

時間的・空間的不連続性の下でエンドツーエンド通信を実現するための中継転送技術として、DTN (Delay Tolerant Networking) 技術がある。DTN は、情報の通信ノード、中継ノード、受信ノードが連携して情報伝達の時間・空間・符号化に関する制御を行い、網資源共用や通信性能の最適化を目指す技術である [11]。DTN における通信手法の 1 つにエピソードルーティングがあり、この手法では専用の通信インフラを持たずに、近距離無線通信で移動ノードから移動ノードへ繰り返しメッセージを転送することで、ネットワーク内にメッセージを拡散する。[12] によると、エピソードルーティングにおける制御「(1) メッセージに利用可能領域 (アンカーゾーン) と生存期限 (Time-to-Live: TTL) を埋め込み、(2) アンカーゾーン内の移動ノードにのみメッセージを転送し、(3) 情報が期限切れになったら削除し、(4) 情報を転送した移動ノードがアンカーゾーンを離れたら、必要に応じてメッセージを削除する」において、移動ノードのバッファの

オーバーフローによりメッセージが消滅してメッセージ間の不公平性が生じる可能性がある。そこで、メッセージ占有率(アンカーゾーンでメッセージを運ぶ移動ノードの割合)を制御する PFCS (Proportional control for Floating Content Sharing) と呼ばれる制御手法を提案している。PFCS は移動ノードのメッセージ転送確率を周囲の状況から動的に調整することにより、無線通信帯域の浪費や使用する移動ノードのストレージ容量を最小限に抑え、メッセージ間の公平性の低下の解決を図っている。

## 2.2 IoT デバイスにおけるコンテナとしての WebAssembly

コンテナ技術の現状として、Docker による実行・管理が普及している。Docker は高機能であるが、大規模で複雑なシステムであり、データセンタ内のサーバで実行する場合と異なり、ハードウェアに制約のある IoT デバイスでは、システムメモリの消費や起動時間の面での欠点が顕著になる。そこで、[13] では、IoT デバイスのための小型で軽量なコンテナ実行基盤として Wasm の利用を提案している。Wasm とは仮想マシン上で実行することを目的とした仮想命令セットアーキテクチャ (Virtual Instruction Set Architecture: Virtual ISA) であり、広義ではバイナリコードを生成し、実行する環境全体を指す。ISA とは、特定のマシンで実行することを目的として設計されるバイナリフォーマットのことであるが、Wasm においては仮想マシン上での実行を目的としており、様々なプラットフォーム上での展開が可能である。Wasm が注目される理由として、Chrome や Firefox, Edge などの主要な Web ブラウザに対応しており、プラグインを使うことなく利用可能である点と、Wasm 実行ファイルがバイナリコードであるためデータサイズが小さく、高速に動作するという点がある。そのため、Wasm をクライアント側に導入することで、サーバの負荷やネットワークトラフィックを大幅に削減することが可能となる。

[13] では、Wasm と Docker コンテナにおいて、IoT デバイスでの定量的な比較実験を通じた様々なパフォーマンス指標の分析を行っている。その結果として Wasm コンテナはオーバーヘッドが大きい、システムメモリの消費と起動時間の短縮により Docker コンテナを上回る性能であると述べている。よって、Wasm コンテナは、単純なプログラムの散発的な実行が必要な場合に最適な選択であり、ハードウェアに制約のある IoT デバイスに適している可能性があると結論づけている。そのため、不特定多数のデバイスエッジ上での機能実行を想定している F-CPS の実現に向け、Wasm コンテナの利用は有効な手段であると考えられる。

## 3. 時空間データ滞留システム

本節では、F-CPS における地域内でのデータ流通基盤である時空間データ滞留システムの目的や概要、本稿における統合試験で対象とする STD とその送信制御手法について記述する。

### 3.1 時空間データ滞留システムの目的

時空間データ滞留システムでは、STD を発生場所からユーザへ直接提供することで、STD の効果的な活用を目指す。一定の期間に特定のエリア内のデバイスエッジによる分散管理の下、STD が配信される地域を「滞留エリア」と定義する。この滞留

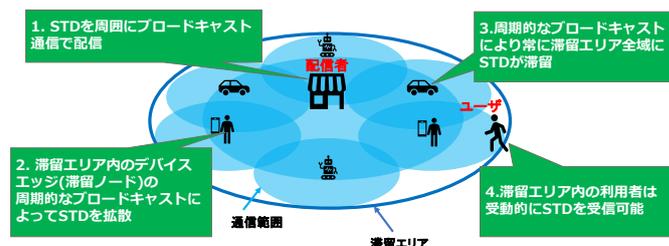


図2 時空間データ滞留システムの概要

エリア内の各デバイスエッジが STD を周期的に送信し、滞留エリア全体に拡散・維持することで、エリア内の利用者に STD を届けることが可能となる。このシステムにより、リアルタイムに STD を特定の地域のみには配信できるため、既設のインターネットなどのネットワーク基盤への負担を削減することができ、データの地産地消の促進が期待される。

### 3.2 時空間データ滞留システムの概要

時空間データ滞留システムは、STD の配信者、滞留エリア内で STD の拡散・維持を担うデバイスエッジ (滞留ノード)、データの受信者であるユーザから構成される。図2に時空間データ滞留システムの概要を示す。発信者は、配信する STD に制御に必要な情報 (制御情報) を付加し、ブロードキャスト通信により周囲の滞留ノードに向けて送信する。滞留ノードは、配信者から STD を受信すると、制御情報を確認し配信する必要がある場合にブロードキャスト通信を用いて送信する。この中継動作がエリア内に存在する滞留ノードによって連鎖することで、エリア全体に STD を拡散・維持する。これにより、ユーザは滞留エリア内に滞在している間、STD を受動的に受信することが可能になる。時空間データ滞留システムは、滞留エリア内に存在する滞留ノードが STD の拡散・維持を担うことによって、データの地産地消を実現する概念として提案したものである。

### 3.3 本統合試験で対象とする時空間データと送信制御手法

本統合試験では、環境センサと Wi-Fi センサを用いて取得した STD を対象とする。環境センサは、気温、湿度、不快指数、照度、気圧、騒音指数を取得し、Wi-Fi センサは半径 10m 以内に存在していたデバイス数を取得する。Wi-Fi センサから取得するデバイス数には、事前実験の結果により取得した送信者から 10m の距離の受信電波強度 (RSSI) の平均値 -71dBm を閾値として、受信電波強度が閾値よりも大きいデバイス数を用いた。本システムは、STD を生成し送信するセンサ A, B、センサから受け取った STD を中継する滞留ノード、STD を受信するユーザ端末から構成される。図3に時空間データ滞留システムによる送信制御のフローチャートを示す。センサ A, B は環境センサと Wi-Fi センサを利用して STD を取得し、STD の生存期限を 35 秒、送信期限を 30 秒に設定して送信期限まで 6 秒毎に 1 回、計 5 回 STD を送信する。送信期限後は、再び各センサより STD を取得し周期的な送信を行う。滞留ノードでは、受信した STD の生存期限を確認し、現在時刻が生存期限を過ぎていない場合は 1 秒待機後に 1 回中継動作を行い、過ぎていない場合は STD を破棄する。ユーザ端末は受信した STD の生存期限を確認

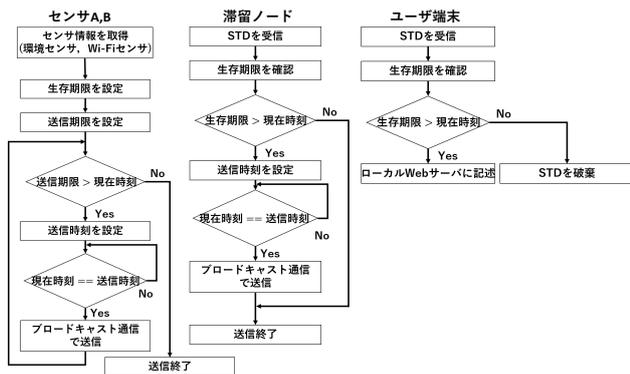


図3 時空間データ滞留システムによる送信制御のフローチャート

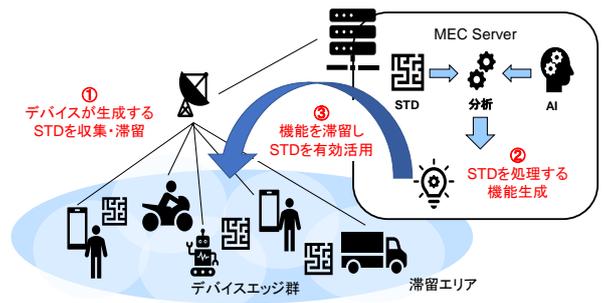


図4 F-CPSにおける機能滞留の概要

認し、現在時刻が生存期限を過ぎていない場合はSTDをローカルWebサーバに記述し、過ぎていた場合はSTDを破棄する。

#### 4. F-CPSにおける機能及び機能実行基盤

本節では、F-CPSにおいて提案している機能、及びその実行基盤の概要と、その配信・実行について記述する。

##### 4.1 F-CPSにおける機能の概要

F-CPSでは、STDと機能を対象のエリアに滞留させることで、データの地産地消を実現する。この機能は、ユーザが自身の用途に応じて、STDを柔軟に分析・加工する手段として提供され、特定の場所・時間に応じた柔軟なデータ利活用の促進を目的としている。図4にF-CPSにおける機能滞留の概要を示す。F-CPSでは、基地局に有線接続され、ユーザの近い位置に存在するMECサーバを利用し、STDの収集や機能作成を行う。機能は、収集されたSTDや、各ユーザの要求に応じて複数生成され、時空間データ滞留システムとMECサーバが連携することで、対象の地域内に機能を滞留させる。先行研究では、MECサーバをオーケストレータとして、地域内で利用可能な機能を一元的に管理し、機能を適切な場所・時間に提供する機能滞留システムを提案している[3][9]。また、機能を適切に各ノードへと配信する手法として、機能自体ではなく、MECサーバのIPアドレスと機能の名称・場所を含む「機能を受信するための情報」を地域内にSTDとして滞留させる手法を提案している[10]。

次に、F-CPSでは特定の地域内にあるデバイスエッジ上で様々な機能を実行するため、Wasmの活用を提案している。機能としてWasmを採用することで、Web基盤上でアプリケーションを実行可能となり、汎用性のある実行基盤を実現している。さらに、バイナリコードで記述されたWasmプログラムは、従来のコンテナと比較してデータサイズが極めて軽量であり、高速かつ安定したデータ流通を可能とする。MECサーバが管理する機能をWasmコンテナとして、ユーザの要求に応じて適切な場所・時間に配布することで、STDの利活用を可能とし、F-CPSによるデータの地産地消を実現する。

##### 4.2 本統合試験における機能配信及び実行

本稿における統合試験では、大学内施設における各種環境情報を対象のSTDとしている。そこで、機能として、受信した

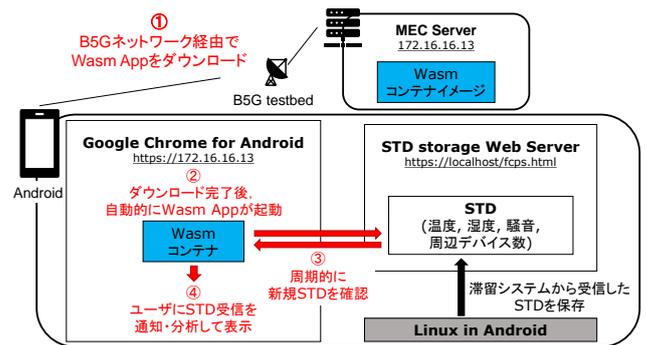


図5 Wasmアプリケーションの動作フロー

STDを分析・表示し、STD受信時にユーザへ通知するWasmアプリケーションを作成した。本システムでは、受信したSTDをユーザ端末内のローカルWebサーバに保存し、それを参照することで、WasmアプリケーションでのSTD利活用を実現している。これは、Wasmがバイナリコードである仕様上、ファイアウォール等による保護が困難であり、セキュリティの観点からファイルシステムへのアクセスが制限されているためである。そこで、ローカルWebサーバからHTTPS通信でSTDを取得する機能を開発することで、ファイルシステムを利用しないセキュアなアプリケーションを実現した。

図5に作成したWasmアプリケーションの動作フローを示す。本統合試験においては、機能の滞留手法による配信は行わず、5G/6Gネットワークに接続されたMECサーバから直接的に配信する。MECサーバには予めWasmコンテナイメージを配置しており、ユーザ端末においてMECサーバへアクセスすることでイメージとコンテナ実行基盤をダウンロードする(図5-①)。本システムでは、ユーザ端末上のGoogle Chromeを利用して、ダウンロードが完了すると、自動的にWasmコンテナが起動する(図5-②)。Wasmコンテナでは、前段落で言及したSTDを保存してあるローカルWebサーバを周期的に参照し、新規のSTDの有無を確認する(図5-③)。新規のSTDが存在した場合、STDを取得し、不快指数や騒音指数、周辺デバイス数を分析することで、各センサ端末周辺の過ごしやすさを設定した指標に基づき表示する(図5-④)。その際、Google Chromeの通知機能を利用して、ユーザへ新規STDの受信を伝達する。

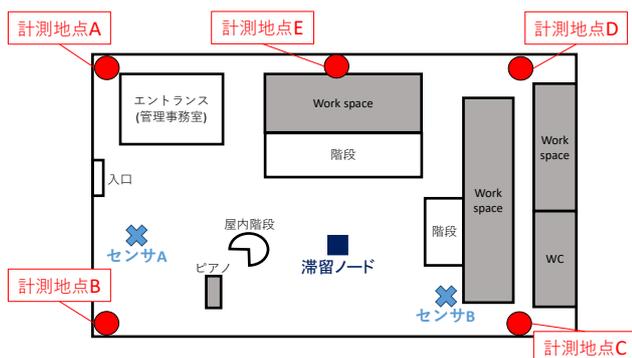


図6 実験環境

## 5. 統合試験による性能評価

本稿における統合試験では、F-CPS 基盤によるデータの地産地消の実現可能性の評価を目的としている。そこで、九州工業大学施設内の各種環境情報を STD とし、時空間データ滞留システムによる STD の滞留から、機能の取得及び実行までを検証する。

### 5.1 実験環境

本稿では、九州工業大学内施設において統合試験を実施した。以下で実験環境や使用機器の詳細について述べる。

図6に実験環境を示す。本稿における統合試験は、九州工業大学戸畑キャンパス内のコワーキングスペースである GYMLABO にて実施した。本施設には、国立研究開発法人情報通信研究機構 (National Institute of Information and Communications Technology: NICT) 高信頼・高可塑 B5G/IoT テストベッドのモバイルアプリケーション実行環境が設置されており、ライセンス契約された端末を通じて、B5G テストベッドネットワークを利用することができる。また、NICT B5G テストベッド上には、MEC サーバを設置しており、作成した Wasm コンテナイメージと実行環境を配置している。

本統合試験では、センサ端末及び滞留ノードとして、合計3台の Raspberry Pi 3 Model B+ を使用した。また、センサ端末には、環境情報センサとして Omron 2JCIE-BU01、Wi-Fi センサとして BUFFALO WI-U3-866DS を使用した。ユーザ端末には、NICT B5G テストベッドに接続可能である Android 端末として FCNT SD01 と、ノート PC として Dell latitude E5440 を使用した。

本統合試験では、時空間データ滞留システムにより配信する STD を Android 端末で受信するために、センサ端末及び滞留ノード、Android 端末が同一ネットワーク内に存在する必要がある。そこで、Wi-Fi Direct によるピア・ツー・ピア (P2P) 通信ネットワークを使用した実験環境を構築した。機能である Wasm アプリケーションについては、NICT B5G テストベッドを経由した無線通信により、ユーザ端末へ配信している。

### 5.2 評価指標及び検証方法

本稿では、構築した F-CPS 基盤のシステム検証として、一連の動作検証の後、時空間データ滞留システムと Wasm アプリ

ケーションによる機能実行の各項目を評価する。

時空間データ滞留システムの評価として、STD 滞留のカバー範囲と STD 取得時間を検証する。カバー範囲を評価する指標として、到達率を用い、式(1)のように定義する。図6に示す各計測地点に Android 端末を設置し、各センサ端末の送信開始から次の送信開始までを1サイクルとして、100サイクルずつ計測を行い、評価する。

$$\text{到達率} = \frac{\text{ユーザ端末が受信した回数}}{\text{センサ端末及び滞留ノードが送信した回数}} \quad (1)$$

また、STD 取得時間として、センサ端末及び滞留ノードの STD 送信開始から、Android 端末において受信した STD をローカル Web サーバに記述するまでの時間を計測した。STD 取得時間については、Android 端末がセンサ A, B のどちらか一方の STD 取得までに要した時間と、両方の STD 取得までに要した時間を評価する。

最後に、Wasm アプリケーションによる機能実行の評価として、Wasm アプリケーションの起動時間、実行時のバッテリー使用割合を検証する。起動時間については、Android 端末の Google Chrome 上で検証し、MEC サーバへのアクセスを開始してから、画面に STD を表示完了するまでの時間を測定した。また、キャッシュが存在する場合としない場合を測定し、キャッシュが起動時間に与える影響を評価する。本検証におけるキャッシュとは、Google Chrome における閲覧サイト履歴、サイトデータ、キャッシュされた画像・ファイルを指す。さらに、F-CPS 基盤のシステムがユーザ端末へ与える負荷を評価するため、バッテリー使用割合を計測する。バッテリー使用割合については、Android 端末上で計測し、滞留プログラム及び機能実行時に消費された合計バッテリー使用量を1として、割合を算出した。このバッテリー使用量は、Android 端末がバックグラウンドで収集するログを参照することで計測した。

### 5.3 システム検証及び性能評価

本節では、節5.2で述べた各項目について検証及び評価を実施する。

#### 5.3.1 F-CPS 基盤における一連の動作検証

F-CPS 基盤の動作検証として、図6に示す地点にセンサ端末及び滞留ノードを配置し、ユーザ端末として Android 端末及びノート PC を使用して検証を行った。一連の動作検証の結果、センサ端末における STD の収集から、時空間データ滞留システムによる STD の滞留、Wasm アプリケーションによる機能の実行まで、正常に動作することを確認した。図7, 8は、作成した Wasm アプリケーションを Android 端末及びノート PC 端末の、Google Chrome 上で実行した機能実行画面である。図7, 8では、同一の Wasm アプリケーションを実行しており、端末に依存せず Google Chrome 上で実行可能なことが明らかになった。作成した Wasm アプリケーションにより、滞留システムから受信した新規 STD を確認し、不快指数や、密集度、騒音値を指標に基づき分析することで、ユーザに各センサ端末周辺の過ごしやすさを表示することが可能となった。よって、F-CPS における機能提供に Wasm を活用することで、Web 基盤を有する端末

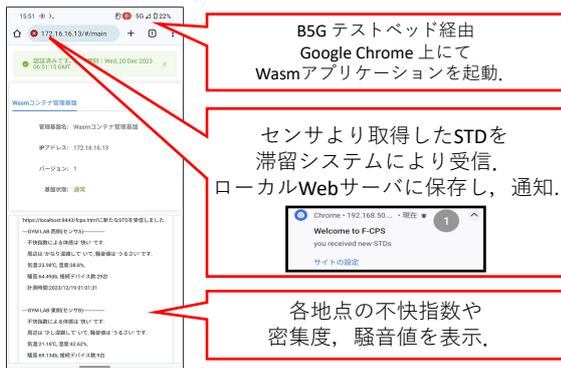


図 7 Android 端末における機能実行画面



図 8 ノート PC 端末における機能実行画面

における、同一アプリケーションでの機能実行を実現した。

### 5.3.2 時空間データ滞留システムにおける STD 滞留の評価

時空間データ滞留システムにおける STD の滞留について評価する。まず、STD 到達率による滞留エリアのカバー率を評価する。図 9 に、各センサ端末から直接 Android 端末へ送信された STD の到達率と、滞留ノードによる中継を介した STD の到達率を示す。図 9 より、計測地点 A, B, C ではセンサ A, B からの到達率と滞留ノードからの到達率は共に 99.0% を超えていた。一方、計測地点 D ではセンサ A からの到達率が 96.6%、計測地点 E ではセンサ B からの到達率が 96.0% と、他の地点に比べて低下しているが、滞留ノードからの受信によりそれぞれ 99.0%、98.6% まで改善された。今回は、2、3% しか改善していないが、滞留エリアがより広域の場合には中継により更に到達率が改善されると予想する。以上の結果から、滞留ノードを介した中継によりセンサからの到達率が低い地点を含め、施設内全域に STD を配信可能であることが明らかになった。

次に、STD 取得時間について評価する。表 1 に各計測地点における STD 取得時間を示す。表 1 より各地点において 10ms 以内に一方の STD を受信し、300ms 以内に両方の STD を受信できることが分かった。また、センサ端末からの到達率が低い計測地点 D, E は、センサ A, B の両方の STD を受信するまでにそれぞれ 290.24ms、232.60ms と他の地点と比べ時間を要することが明らかになった。

### 5.3.3 Wasm アプリケーションによる機能実行の評価

F-CPS における機能として作成した Wasm アプリケーション及びその実行について評価する。本統合試験で作成した Wasm

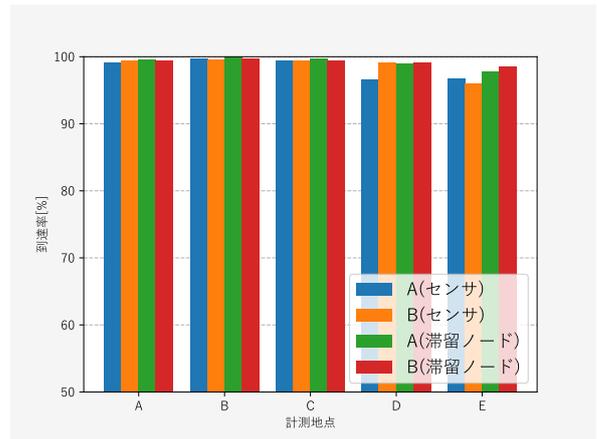


図 9 各計測地点における STD 到達率

表 1 各計測地点における STD 取得時間

計測地点	一方の STD を取得するまでの時間 [ms]	両方の STD を取得するまでの時間 [ms]
A	8.920	182.103
B	9.052	172.283
C	8.033	221.017
D	9.052	290.244
E	7.990	232.599

表 2 Wasm アプリケーション起動時間

キャッシュがない場合の起動時間 [s]	キャッシュがある場合の起動時間 [s]
20.678	5.274

アプリケーションのコンテナイメージサイズは、2.42 MB となった。軽量なデータサイズの機能を実現した要因として、Wasm コンテナを利用したことで、バイナリコードで記述されたプログラムであることや、従来のコンテナに必要とされた設定ファイルやライブラリが不要であることが考えられる。

次に、Wasm アプリケーション起動時間について評価する。表 2 に Android 端末における Wasm アプリケーション起動時間を示す。表 2 より、キャッシュがない場合は約 20.7s の起動時間に要した。一方で、キャッシュがある場合は、約 5.3s で起動可能であることが明らかになった。これは、MEC サーバへのアクセスの際に認証が必要なためである。ユーザ端末が MEC サーバに接続する際、Wasm コンテナ情報等を取得するための認証鍵が送付され、認証処理が実行される。そのため、キャッシュがない場合と比較して、キャッシュがある場合には起動時間が約 15.4s 削減された。

最後に、Wasm アプリケーションのバッテリー使用割合について評価する。図 10 に Android 端末において測定した、滞留プログラム及び機能実行時のバッテリー使用割合を示す。滞留プログラムを実行するために利用した Android 上で Linux を起動するツールである Termux と、滞留ネットワークに接続するために利用された DNS のバッテリー使用割合は全体の 2.2% となった。また、機能を実行している Google Chrome 及び Google Service に利用されたバッテリー使用率は全体の 27.2% となった。この際、

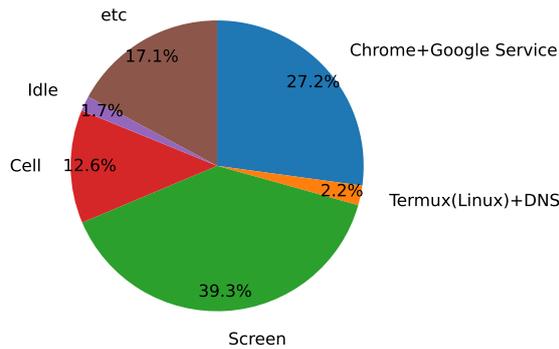


図 10 機能実行時のバッテリー使用割合

Google Chrome に関しては、約 682.8s 間起動しており、バッテリー消費率は 1.58% であった。よって、1s あたりの消費率は約 0.002314% となり、ユーザ端末への負担は少なく、実用可能なアプリケーションであると言える。

以上の結果より、本統合試験において、センサ端末から取得した STD を大学施設内全域に滞留させ、ユーザ端末上で実行した Wasm アプリケーションにより、取得した STD を分析・表示することで、F-CPS 基盤におけるデータの地産地消を実現した。

## 6. まとめ

本稿では、F-CPS によるデータの地産地消の実現可能性を示すため、九州工業大学施設内の各種環境情報を STD として、時空間データ滞留システムによる STD の滞留から、機能の取得及び実行までの統合試験を実施した。センサ端末及び滞留ノードより配信した STD を、MEC サーバより取得した Wasm アプリケーションによって分析することで、施設内のユーザへ目的に適した場所を提示可能となった。システム検証の結果、統合したシステムが正常に動作し、センサ端末において STD 取得後から約 300ms 以内にユーザ端末で受信可能であることを示した。また機能については、イメージサイズが 2.42MB と軽量で、ハードウェアや OS に依存せず、機能を実行している Google Chrome と Google Service の合計バッテリー使用率も 1s あたり 0.0023% と、実用可能なアプリケーションであることが明らかになった。これらの結果から、特定の地域内で発生する STD をその場で活用するデータの地産地消を実現し、F-CPS 基盤の有効性を示した。今後は、機能配信のための適切な送信制御手法について検討し、より実用的なアプリケーションによる検証を行う。

**謝辞** 本研究の一部は、国立研究開発法人情報通信研究機構 (NICT) の委託研究 JPJ012368C05501 による成果を含む。ここに記して謝意を表す。

## 文献

- [1] 野林大起, 「地域データの地産地消を実現するフローティングサイバーフィジカルシステム」電子情報通信学会 第 35 回 情報伝送と信号処理ワークショップ (CSWS), 2023 年 11 月。
- [2] Hotaka Kaneyasu, Daiki Nobayashi, Kazuya Tsukamoto, Takeshi Ikenaga, Myung Lee, "Spatio-temporal Data Retention System for Floating Cyber Physical System," The 10th International Symposium on

Applied Engineering and Sciences (SAES2022), Dec. 2022.

- [3] S. Sekigawa, et al. "Web Application-Based WebAssembly Container Platform for Extreme Edge Computing," 2023 IEEE Global Communications Conference, pp. 3616-3621, 2023.
- [4] Daiki Nobayashi, Ichiro Goto, Hiroki Teshiba, Kazuya Tsukamoto, Takeshi Ikenaga, Mario Gerla, "Adaptive Data Transmission Control for Spatio-temporal Data Retention over Crowds of Vehicles," IEEE Transactions on Mobile Computing, Nov. 2022, Vol.21, pp. 3822-3835, Nov. 2022.
- [5] Ichiro Goto, Daiki Nobayashi, Kazuya Tsukamoto, Takeshi Ikenaga, Myung J. Lee, "Transmission Control Method for Data Retention Taking into Account the Low Vehicle Density Environments," IEICE Transactions on Information Systems, Vol.E104-D, NO.4, pp.508-512, Apr. 2021.
- [6] Shunpei Yamasaki, Daiki Nobayashi, Kazuya Tsukamoto, Takeshi Ikenaga, Myung J. Lee, "Efficient Data Diffusion and Elimination Control Method for Spatio-temporal Data Retention System," IEICE Transactions on Communications, Vol.E104-B, No.7, pp.805-816, Jul. 2021.
- [7] Ichiro Goto, Daiki Nobayashi, Kazuya Tsukamoto, Takeshi Ikenaga, Myung J. Lee, "Beacon-Less Autonomous Transmission Control Method for Spatio-Temporal Data Retention," INCoS 2020, AISC 1263, Springer, pp.503-513, Sep. 2020.
- [8] 後藤一郎, 野林大起, 塚本和也, 池永全志, "車両を用いた時空間データ滞留システムにおける送信制御手法の検討 ルクセンブルグモデル (LuST) を用いた評価," 信学技報, vol.120, no.413, NS2020-132, pp.55-60, 2021 年 3 月。
- [9] 関川 柊, 佐々木 力, 田上 敦士, "F-CPS に向けた高可搬型 WebAssembly コンテナの検討," 電子情報通信学会 2023 年 総合大会, B-6-87, 2023 年 3 月。
- [10] 田中明弥, 野林大起, 塚本和也, 池永全志, 関川 柊, "F-CPS における機能滞留を実現するための WebAssembly を利用したアプリケーション実行基盤の可搬性の性能評価," 信学技報, vol.123, no.107, NS2023-47, pp.77-82, 2023 年 7 月。
- [11] 鶴 正人, 内田 真人, 滝根 哲哉, 永田 晃, 松田 崇弘, 已波 弘佳, 山村 新也, "DTN 技術の現状と展望," 電子情報通信学会通信サイエティマガジン, no.16, pp.57-68, 2011 年 3 月。
- [12] Ryo Hagihara, Yasuhiro Yamasaki, Hiroyuki Ohsaki, "On delivery control for floating contents sharing with epidemic broadcasting," 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 353-356, Jan 2017.
- [13] Jonah Napieralla, "Considering WebAssembly Containers for Edge Computing on Hardware-Constrained IoT Devices," Master of science Computer Science, June 2020.