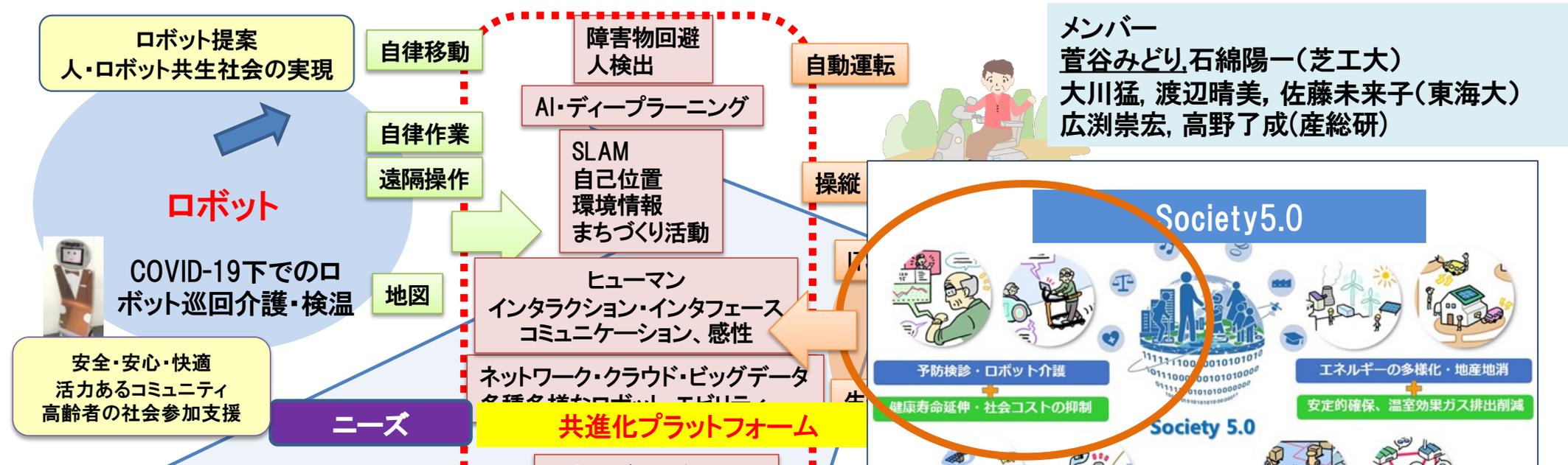


ソフトウェアG:ロボットを日常的に利用するSociety5.0 社会に向けて

- MEC用マルチノード統合システムの開発におけるシステムソフトウェア基盤の実現のためのG
 - ロボット・自動車共進化に向けて(芝浦工業大学ロボット・コンソーシアム)での活動からの要求抽出^[1]



- 多様なハード・ソフト・ミドルウェアの**不統一**
- 人・物・移動に関わる高度な物理シミュレーションが必要:**計算機パワー不足/応答性の課題**
 - ロボット・自動車などの通信量の増大・**データ量の増大**
 - セキュアなデータ保存**

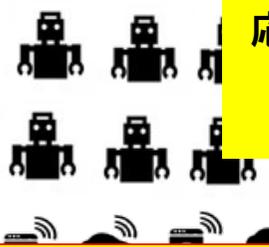
MEC(Multi-Access Edge Computing) 活用による課題解決

2021年度に向けた課題と方針

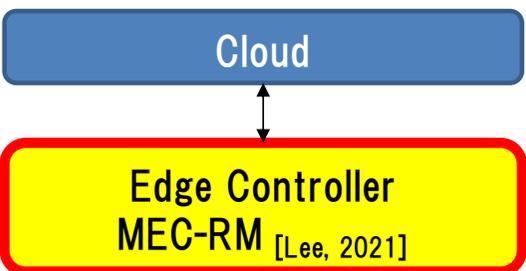


- 2020年度(昨年度)からの大きな改善点
 - FiCからM-KUBOSとなり計算レイテンシー低減, 性能改善
- 昨年のサイトビジットのQ&A
 - これから広まってゆくには, ポータビリティが重要(住元Advisor)
 - PYNQの開発環境を利用して, これを使ってプログラム開発を行い, マルチボードを作り, これを使い **ヘテロジニアシステム** を作れば **国際的に展開が可能である(天野)**
 - Domain Specific ArchitectureをM-KUBOSのPS-Partの豊富なインターフェースで繋いで前提で **ヘテロでコントロールしていくべきだ** と思っている。これが当初の目的であるし、**FPGAだけで全部やるのは、やりたい気もあるが無理だ** と思っている(天野)
 - 全体のアーキテクチャを考慮しつつそのうえでソフトウェアをどう作りこんでその中でハードウェアをどう活用するか、といった視点で進めていただくと見通しもよくなると思う(Advisor)
- ソフト基盤G
 - これらの意見をもとに実用的な **ヘテロジニアシステム** (DSA/ROS Cluster) システムの構築
 - 1)ヘテロジニアシステム設計・実装 2)通信効率化 3) MEC における高信頼・応答性 4)オフローディング, 自動生成
 - ロボットとAI, セキュリティ(暗号化)を例に上記を実現する仕組みを検討

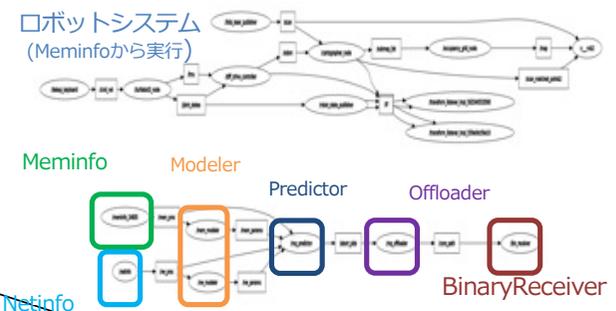
1) MEC-Cloud (ヘテロジニアスプラットフォームミドルウェア)



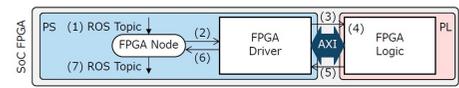
応答性・信頼性
セキュリティ
高速化要求



ROS+Kuberunetes 動的オフローディングシステム
[長濱, 2020] APRIS2020, Best Short Paper Award



ROS2 通信高速化 FOrEST



ROS2-FPGAノードの自動生成 [大川, 2021]

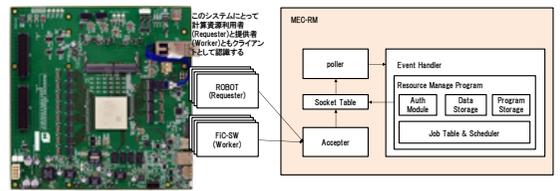
通信高速化(TBON バイナリ通信) [Lee, 2021]

ROS Cluster (FPGA Server) 高性能・省電力

ROS Cluster (KVS+DRAM/NVMM) 高信頼・応答性

ROS Cluster (ManyCore, GPGPU Server) 汎用性

MEC-RM [Lee, 2021]
汎用性・高速応答を実現するMEC RM



NVMM/DRAM + KVS + ROS SLAM による
応答性と信頼性の向上 [小沢, 2020, 東, 2021, IEICE Transaction, 2021]
2021 情報処理学会全国大会, 奨励賞, 大会優秀賞

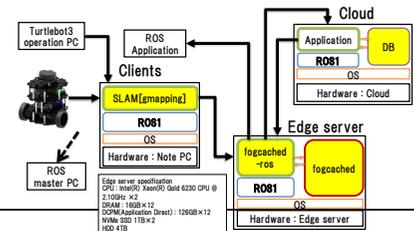
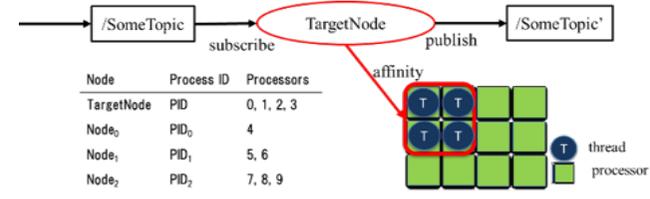


図7: 探索システム概要

アフィニティを用いたMany-core への動的
ROS Node マッピング [福井, 2021]

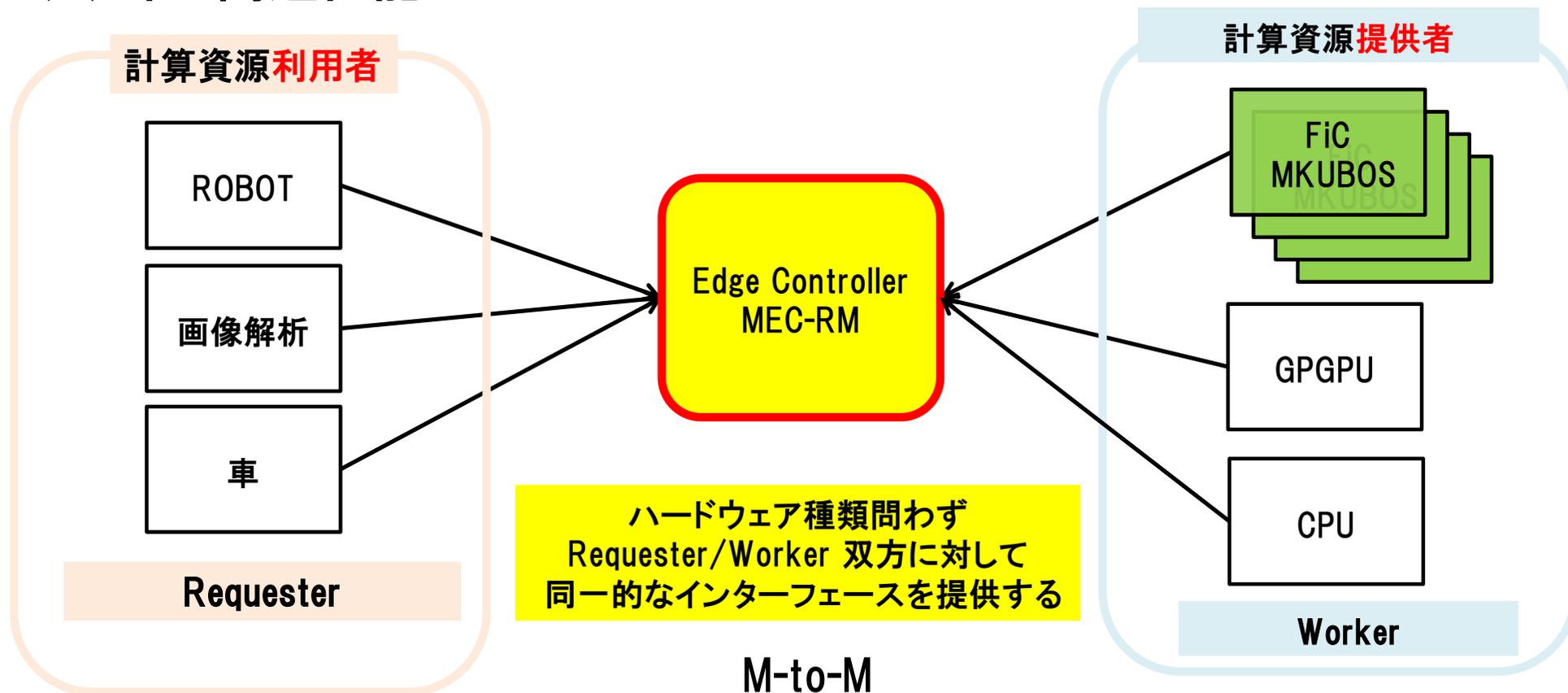


MEC-RM [Kou, 2021] AES並列処理
2021/10/17

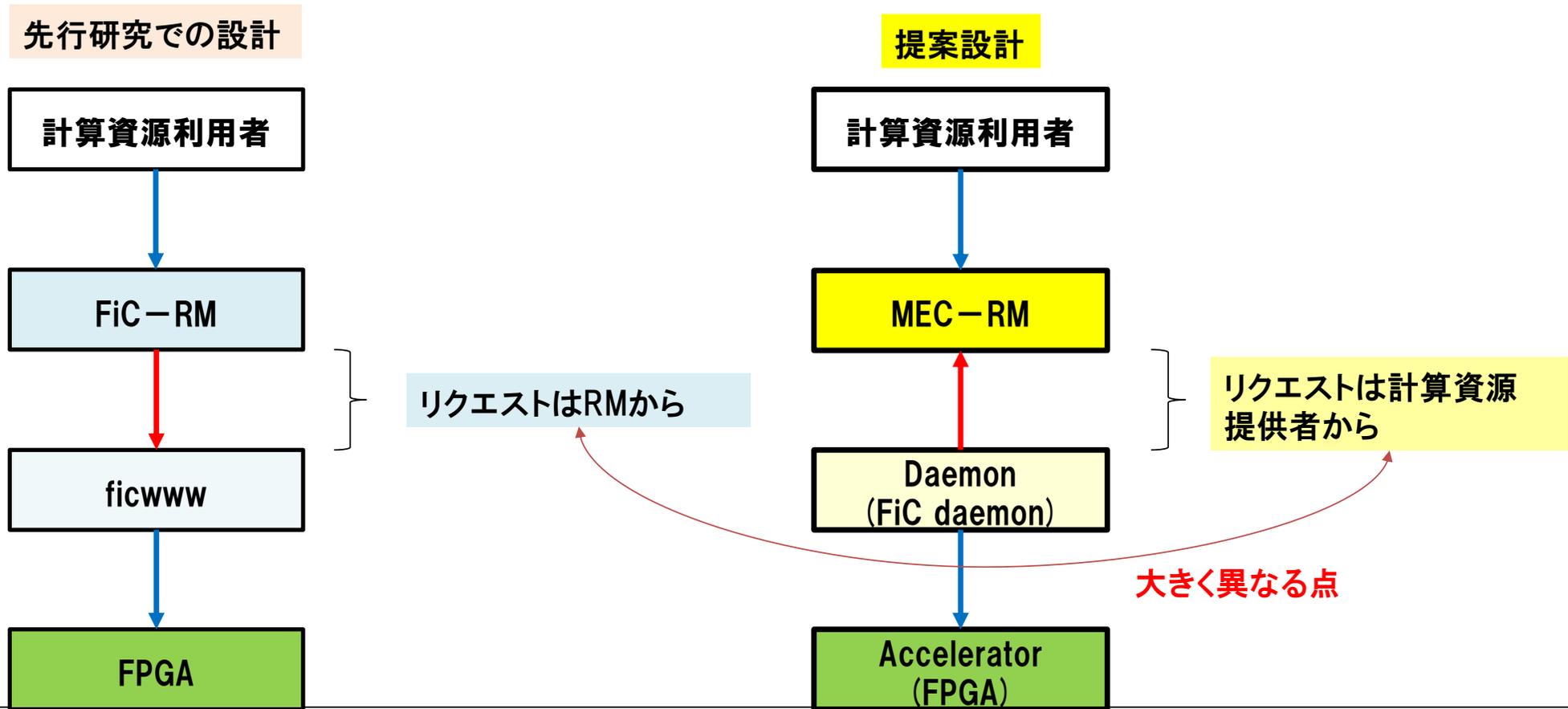
Edge Controller MEC-RM (Resource Manager)



- 汎用性・高速性能



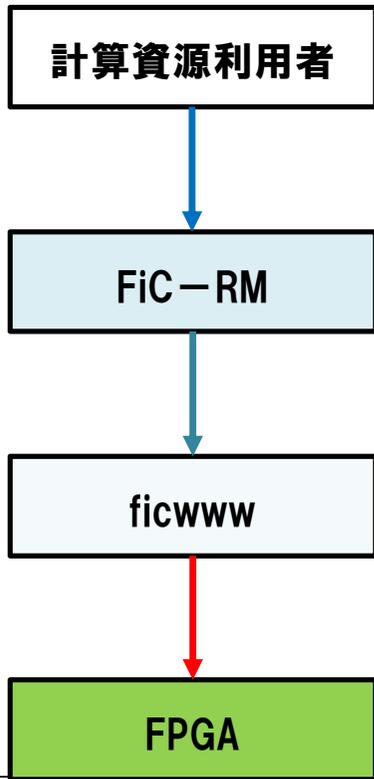
既存のシステムとの比較 (1)



既存システムとの比較 (2)



先行研究での設計



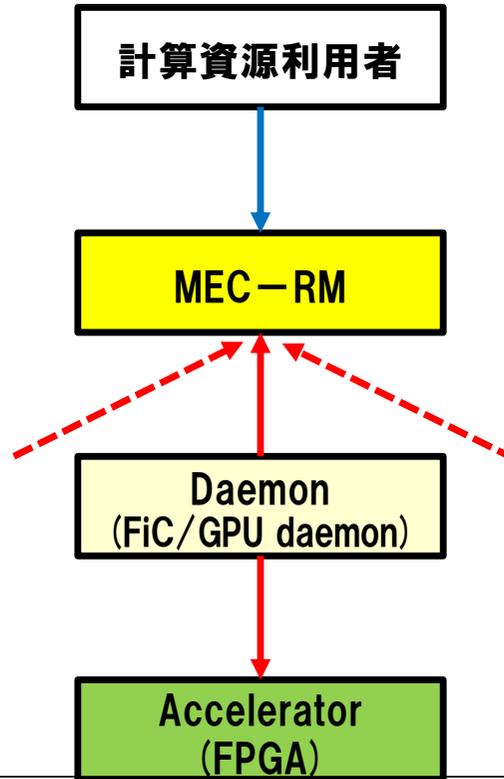
FiC-RM : 計算資源利用者にとってのサーバ

システム複雑さが高い

計算資源提供者にとってのクライアント

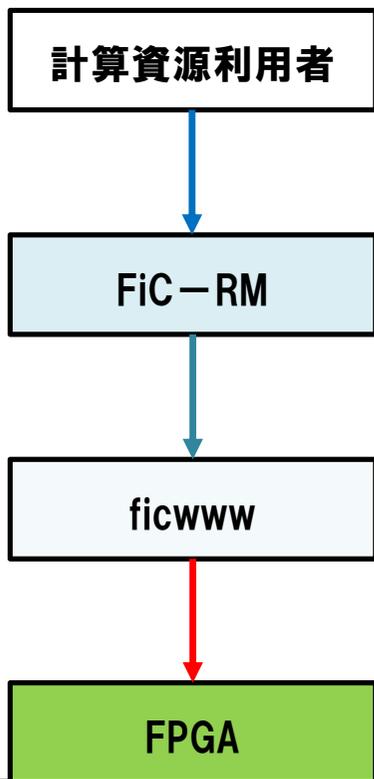
計算資源のリクエストはFiCに限られている

提案設計



既存システムとの比較 (2)

先行研究での設計



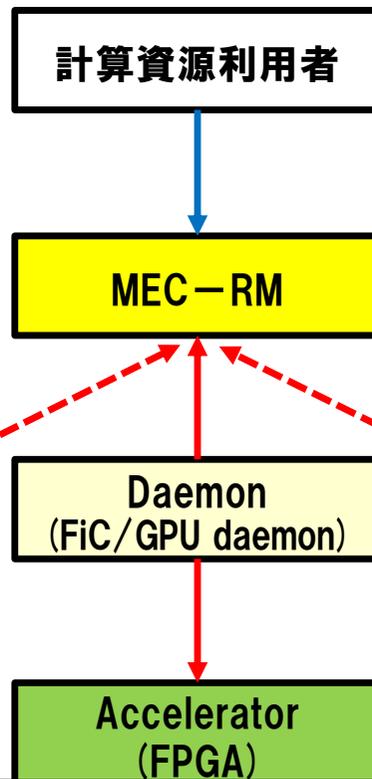
FiC-RM : 計算資源利用者にとってのサーバ

システム複雑さが高い

計算資源提供者にとってのクライアント

計算資源のリクエストはFiCに限られている

提案設計



MEC-RM : サーバ役に専念より実用的な仕組みができる



MEC-RM : 計算資源利用者にとってのサーバ

システム複雑さが低い

計算資源提供者にとってのサーバ

動的な更新がしやすい

計算資源のリクエストはFiCに限られていない

様々な Acceleratorへ拡張可能

Demo



```
cmdline [ ] → cmdline [ ] → cmdline [ ] → cmdline [ ]

=====
[MEC JOB CONSOLE]
#150
> [3 FINISHED] PROG: conv WORKER: 2
> [2 FINISHED] PROG: vec_add WORKER: 3
> [1 FINISHED] PROG: relu2 WORKER: 4
> [0 FINISHED] PROG: conv2d WORKER: 3
#149
> [3 FINISHED] PROG: conv WORKER: 4
> [2 FINISHED] PROG: vec_add WORKER: 2
> [1 FINISHED] PROG: relu2 WORKER: 1
> [0 FINISHED] PROG: conv2d WORKER: 3
#148
> [3 FINISHED] PROG: conv WORKER: 1
> [2 FINISHED] PROG: vec_add WORKER: 2
> [1 FINISHED] PROG: relu2 WORKER: 4
> [0 FINISHED] PROG: conv2d WORKER: 3
-----
[ ]

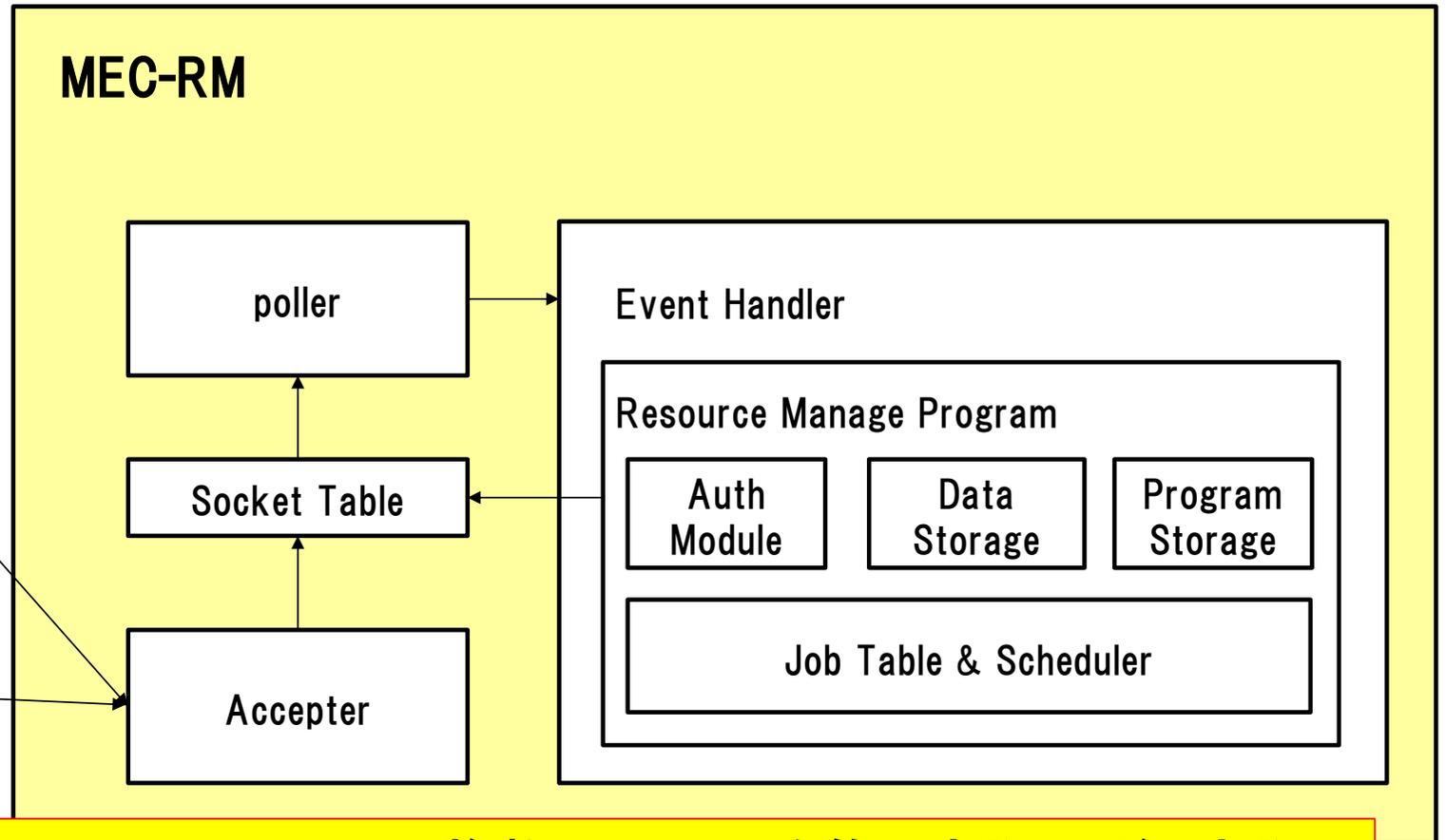
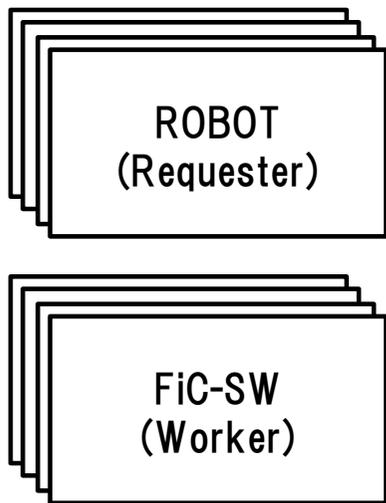
yeon@Subaru:~/Project/SIT/oh-my-tpga/cmdline
→ cmdline git:(prototype) x ./jobseq1.sh [ ]

yeon@Subaru:~/Project/SIT/oh-my-tpga/cmdline
→ cmdline git:(prototype) x ./jobseq2.sh [ ]
```

システム構成(Socket層)



本システムにおいては、
計算資源利用者
(Requester) と、提供者
(Worker)とも、クライア
ントとして認識する



独立したサーバーとして MEC-RM は複数のリソースを管理することができる

OSSでの公開に向け



<https://github.com/enjoydolylab/oh-my-fpga>

<> Code Issues Pull requests Actions Projects Security Insights Settings

prototype 2 branches 0 tags

Go to file Add file Code

yeonzi Allow to fetch server status 8d91eaf 3 days ago

サーバ側コード

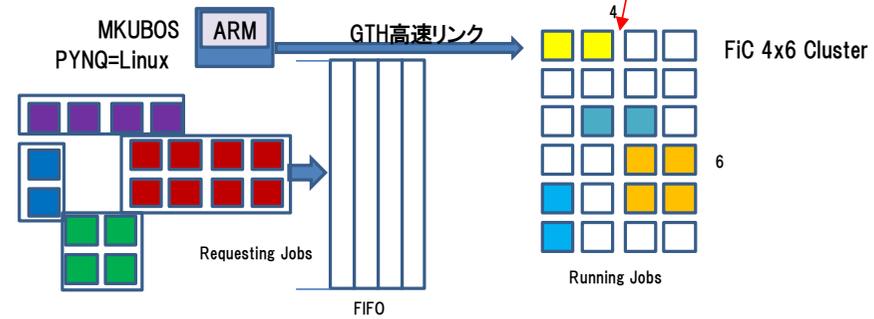
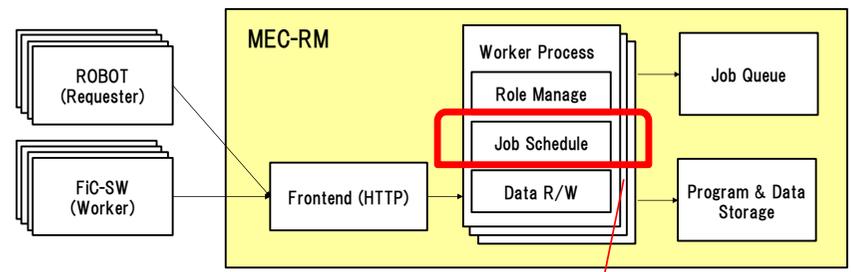
- cgibin Allow to fetch server status
- cmdline query jobs by script
- lib Add 'lib/excal/' from commit 'cb7a248926388454'
- .gitignore Initial commit
- LICENSE Initial commit 15 months ago
- README.md Initial commit 15 months ago
- arch.md Architecture Design 15 months ago

クライアントプログラム

サーバ側ライブラリ

Language	files	blank	comment	code
C	42	2472	1039	8479
Markdown	18	659	0	1316
C/C++ Header	25	246	208	734
TeX	2	67	67	363
make	4	57	0	139
Ruby	3	20	0	50
YAML	3	8	0	38
Bourne Shell	1	4	0	8
SUM:	98	3533	1314	11127

ベースは
C言語で実装
(total, 11,127 lines)



FPGAのFiCのスケジューリングアルゴリズム,
天野研[Yamakura, 2021]などを適用
Yamakura et.al., A Multi-tenant Resource Management System for
Multi-FPGA Cloud Systems, IEICE Transaction, 2021(to be appeared)

基礎的な評価(投稿準備中), 非同期I/Oシステムの拡張整備

2) 通信の高速化: バイナリデータフォーマット

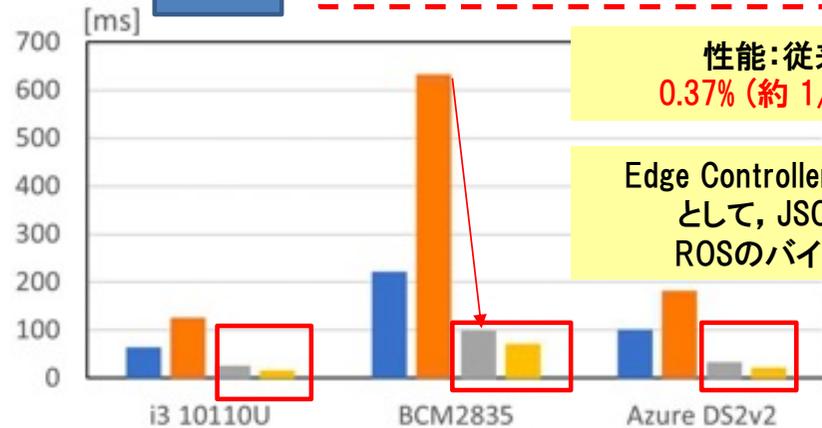
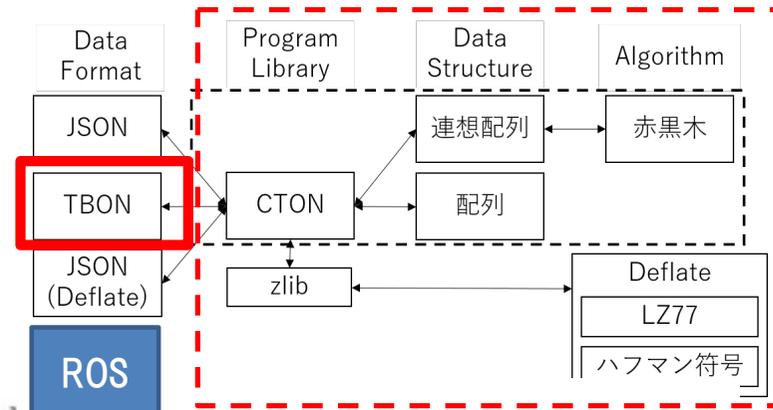
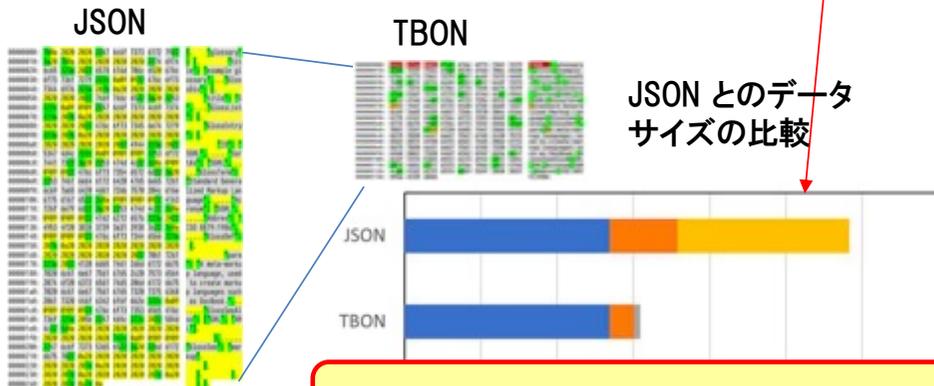


- バイナリデータフォーマット **TBON** [Lee, 2021] による通信の高速化(OSSにて公開)



産総研: https://www.digiarc.aist.go.jp/news/news_2021_06_10.html

提案方式 **TBON** はデータサイズ **JSON** の $\frac{1}{2}$



性能: 従来方式と比較
0.37% (約 1/3) の処理速度

Edge Controllerの標準フォーマットとして, JSONと互換で利用
ROSのバイナリ通信へ対応

MEC クラスタ間での通信の高速化

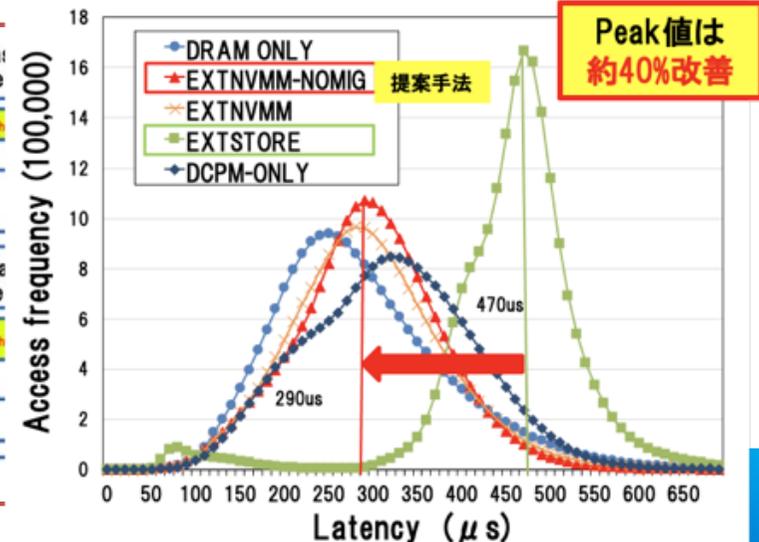
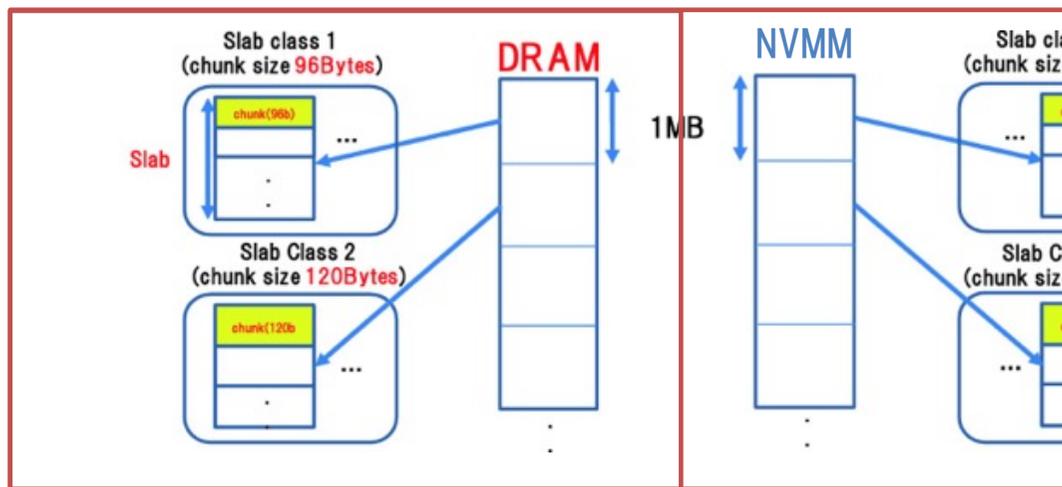
性能比較

3) MECにおける高信頼・高応答性:Fogcached



- 不揮発性メインメモリを利用し応答性と信頼性を両立させる Hybrid KVS Server

→ fogcachedは, memcachedを拡張し, RAM と NVMM(DCPM)からなるハイブリッド型メインメモリ上でIn-Memory Key-Value Store を展開, エッジサーバの応答性の向上を実現^[1,2]



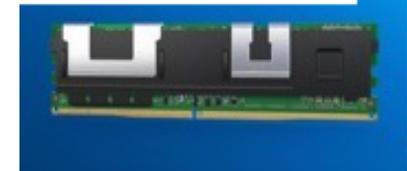
[1] Kouki Ozawa, Takahiro "Hybrid Memory-Based KVS Server for Edge Computing", Edge Computing, Springer, 2020, pp. 50-62

[2] Kouki Ozawa, Takahiro Hirofuchi, Ryousei Takano, Midori Sugaya "fogcached: DRAM/NVMM hybrid KVS Server for Edge Computing" IEICE TRANSACTIONS on Information and Systems

Memcached に実装されているLRU を NVMM へ拡張

Hybrid Memory-Based KVS Server in Computer Science 12407,

Edge 2020 Proceedings 採録, IEICE Transaction 採録



ロボットから収集したデータの逐次保存・利用

• Fogcached-ros [3]

- Hybrid KVSサーバをROSに対応
- Topicを介して, fogcached にデータを保存
- ROS(Robot Operating System)からアクセスするためのAPIを提供

過去のデータを保存できる

利便性の向上

信頼性の向上

(1) setAPI
 int set_fogcached(topic, class message, ID);

(2) getAPI
 int get_fogcached(topic, class& message, ID);

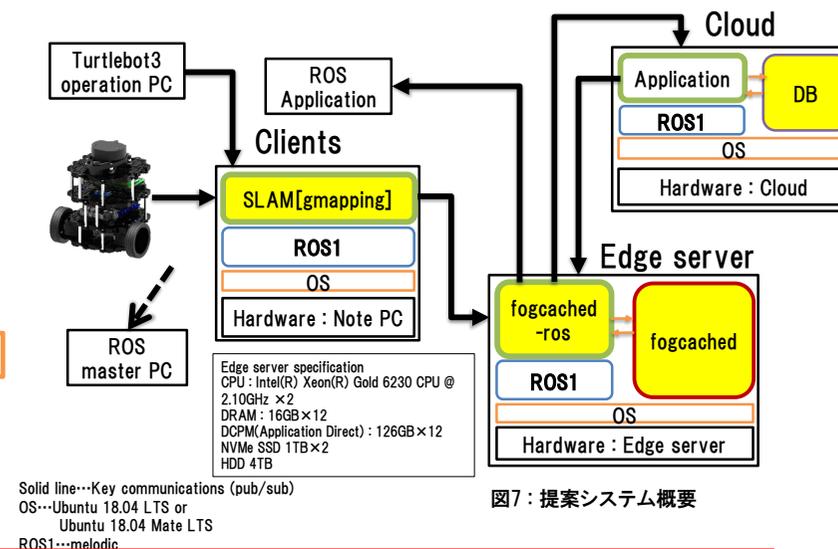
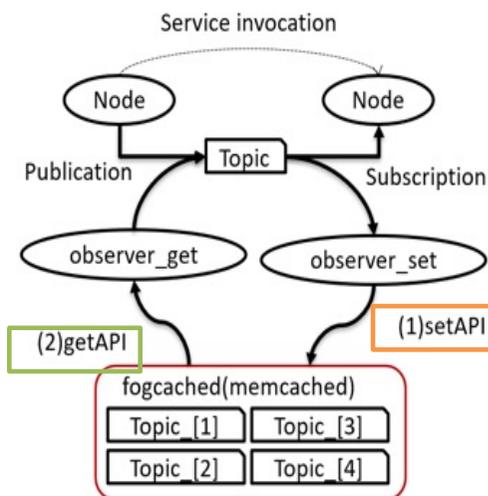
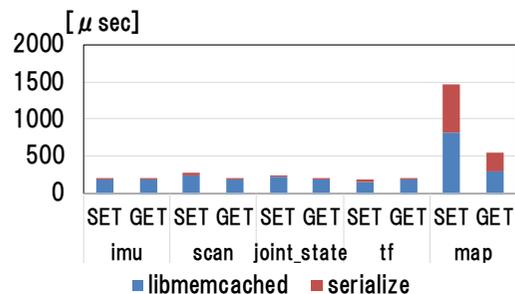


図7: 提案システム概要

Fogcached-ros : Hybrid KVS DRAM/NVMM + ROSの連携

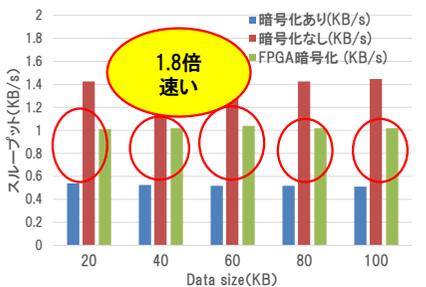
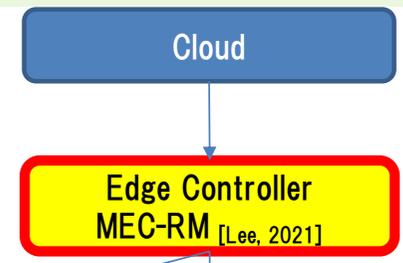
既存アプリ (SLAM) を変更せず利用可能, 自動的にEdge-Cloudへデータがストアされる

[1] Koki Higashi
 Based Extension f

実例： MEC利用例：暗号化+DB



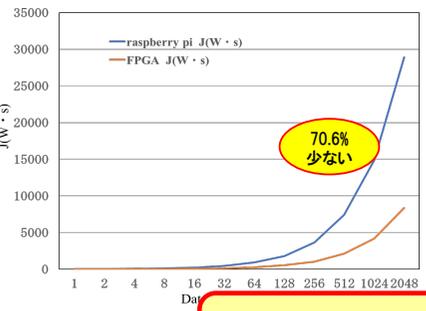
- ハードウェアの特性を活かしたサービスプラットフォーム
 - MECを中心とした暗号化+KVS 高応答性・高信頼サーバ
 - 暗号化データを高信頼・応答サーバにセキュアに保存する
 - データのセキュリティと高性能・高信頼, 低消費電力の実現
 - AES 暗号化, ECB (Electronic CodeBook mode; 電子符号)



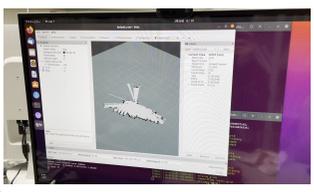
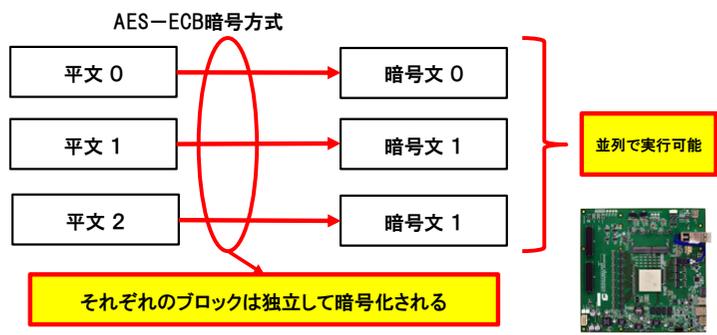
**暗号化(セキュリティ)
高速化・省電力処理**

**ROS Cluster
(FPGA Server)
高性能・省電力**

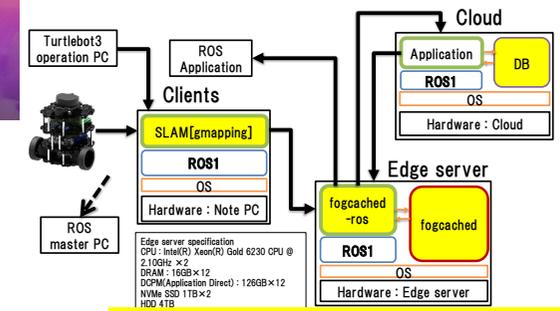
**ROS Cluster
(KVS+DRAM/NVMM)
高信頼・応答性**



暗号化サーバとして利用



DBサーバとして利用



**DRAM /NVMM による
高信頼性**

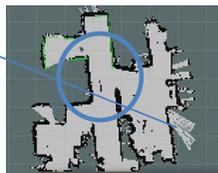
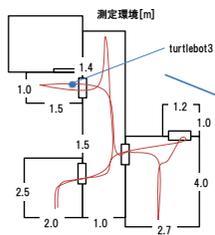
異なるハードウェアを連携させたサービスプラットフォームとして利用が可能

FPGAは暗号化モジュール

2021/10/17

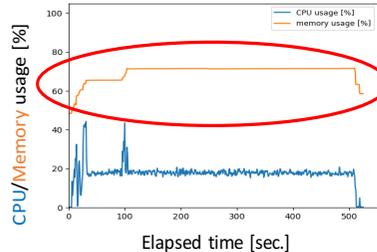
クラスターサーバへのROS コンポーネント動的オフローディングシステム

エッジへのオフローディング

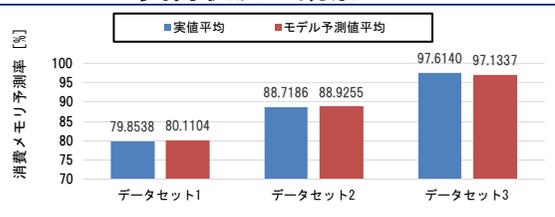


The robot on SLAM progress was heavily loaded !!

Mapping



負荷状況の観察



システムメモリによる負荷状況の多変量モデル

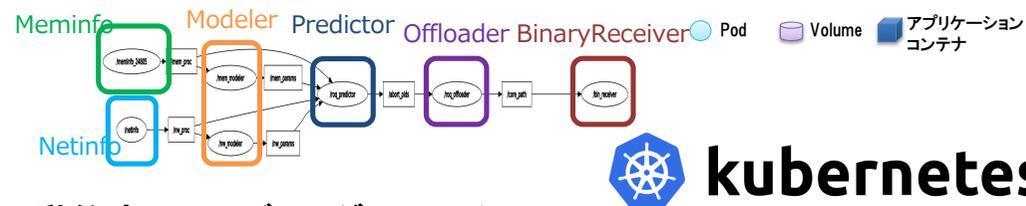
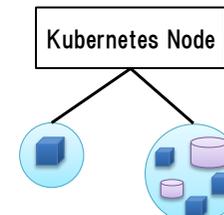
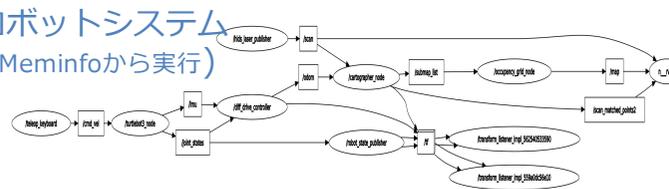
	上乗せ負荷	モデルの評価に用いる式
データセット1	0%	$= -71983 + 0.066 \cdot M_{heap} + 1.340 \cdot M_{stack} - 12.205 \cdot M_{buf} + 200.237 \cdot M_{cache}$
データセット2	25%	$= -40847 + 0.113 \cdot M_{heap} - 9.085 \cdot M_{stack} - 7.972 \cdot M_{buf} + 200.749 \cdot M_{cache}$
データセット3	50%	$= 92.899 + 0.120 \cdot M_{heap} + 4.624 \cdot M_{stack} - 0.080 \cdot M_{buf} + 0.008 \cdot M_{cache}$

APRIS 2020(Asia Pacific Conference on Robot IoT System Development and Platform (APRIS), Nov. 9-10 (2020) Conference
Best Short Paper Award を受賞
 IEEE CANDAR Workshop 採録
 論文誌 conditional accept

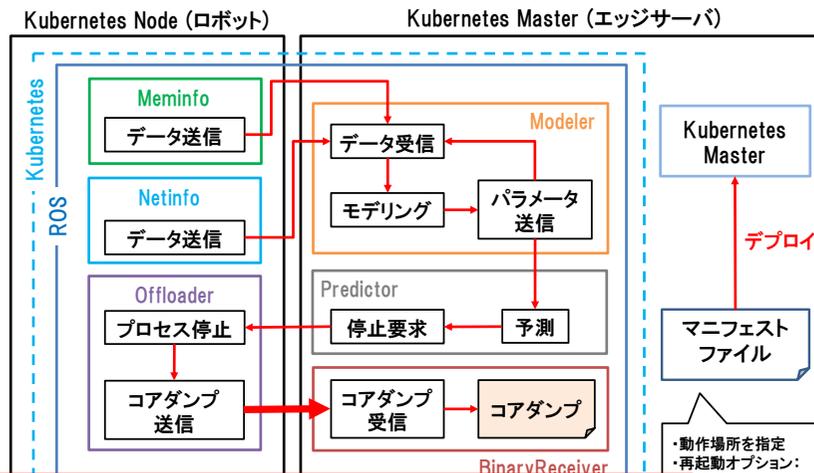
$$M = b_1 \times M$$

負荷の高いコンポーネントをMEC-RM 経由でクラスターサーバに動的にオフロード

ロボットシステム (Meminfoから実行)



動的オフローディングシステム



4) ROS2-FPGAノードの自動生成ツールFOrEST

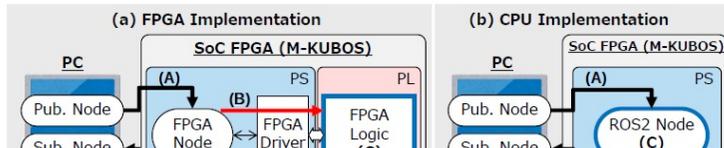
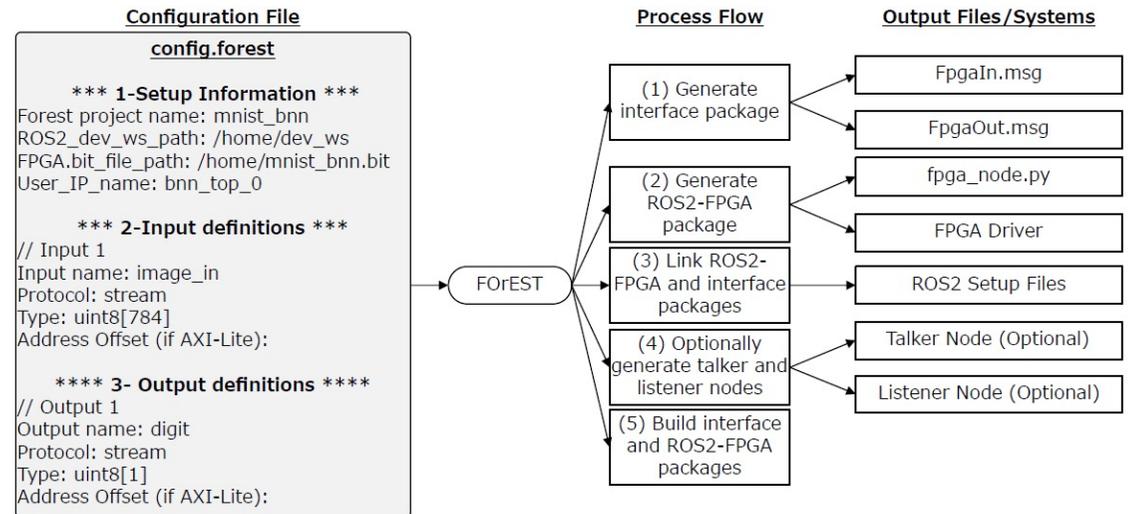
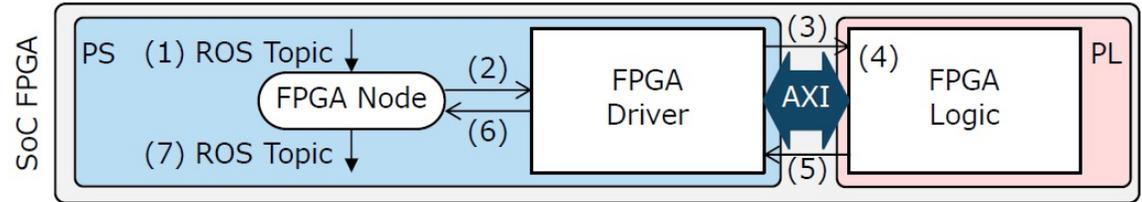


● FOrEST (FPGA-Oriented Easy Synthesizer Tool)

- HLS(高位合成)可能C/C++言語で書かれたFPGAモジュールから、ROS2ノードを自動生成するツール
- C/C++言語の関数の引数の型に応じたROS2メッセージ型定義(interface pkg)を自動で生成
- FPGAを含むROS2ノードを自動生成
- ランダムテストするROS2ノードを自動生成

● 現在

- 画像フィルタ[1]・強化学習NN[2]で機能実証



(A) Subscribe to Input In
(D) Read p a

- [1] - D. Pinheiro Leal, M. Sugaya, H. Amano, T. Ohkawa "Automated Integration of High-Level Synthesis FPGA Modules with ROS2 Systems", International Conference on Field Programmable Technology (FPT), 2020.
 [2] - D. Pinheiro Leal, M. Sugaya, H. Amano, T. Ohkawa "FPGA Acceleration of ROS2-Based Reinforcement Learning", International Conference on Field Programmable Technology (FPT) Lectures (CSA'20), 2020.

ROSコンポーネントを自動的にFPGA化 → 本システムの利便性の向上



今後の展望

FIC  MKUBOS 

