
NCESシンポジウム2026

リアルタイムLinux関連の研究活動

2026年10月2日

名古屋大学大学院情報学研究科
附属組込みシステム研究センター
李 奕驍

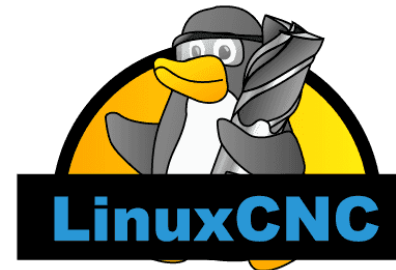
背景

・近年Linuxベースの組み込みシステムが増えている

特に機能が複雑化した制御システムによく使われている

- ・ロボティクス ROS/ROS2
- ・車載IVI AGL (Automotive Grade Linux)
- ・ものづくり LinuxCNC、Klipper

ROS
ROS2



リアルタイム性と信頼性の課題が多い

- ・カーネル設計当初は汎用処理を想定
制御処理に使う場合、リアルタイム性の改善が必要
- ・共有リソースが多くてアプリ間に干渉が生じやすい
組み込みシステムでは高信頼アプリの保護が求められる

背景

・リアルタイム性と信頼性向上に関する技術

カーネル実装の改善

- ・ Low-Latency Desktop
不要な排他制御等を最適化、短いクリティカルセクション
- ・ Fully Preemptible Kernel (PREEMPT_RT)
割込み等の処理中も完全にPreemptできる拡張



リソース分離の強化

- ・ 仮想化技術で信頼度が低いアプリの干渉を制限
軽量仮想化（コンテナ）、Hypervisor（QEMU/KVM）
- ・ 独立の専用カーネルでリアルタイム処理を行う
LinuxとRTOSを共存させるコカーネル技術Xenomai



ハードウェアアクセラレータの活用

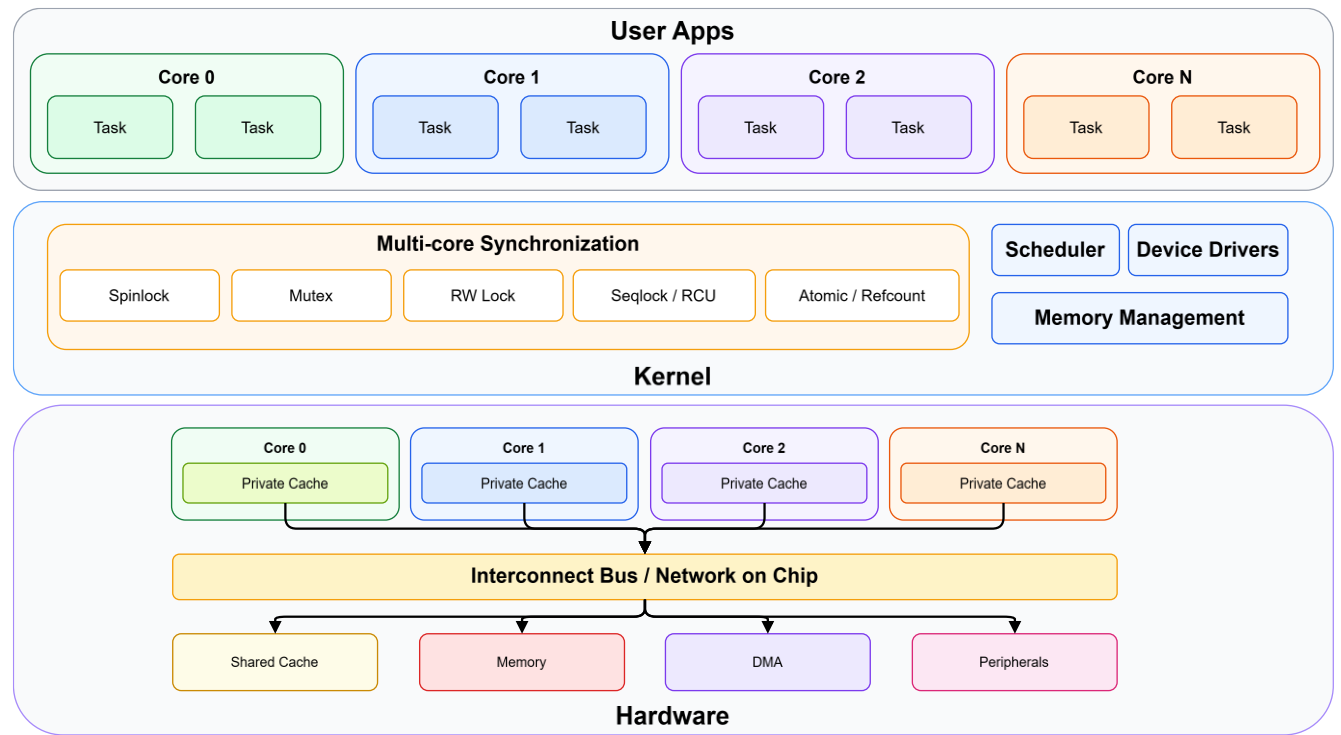
- ・ 遅延要件が厳しい処理 → DSP、FPGA等
- ・ 並列計算が必要な処理 → GPU、NPU等
CPUで処理するとシステム全体の性能が低下



・リソース競合を考慮した性能評価方法

複雑化したシステムで競合が生じやす

- ・ ユーザ空間：マルチコア上の複数アプリ間で通信・同期
- ・ カーネル空間：排他制御、デバイス共有
- ・ ハードウェア：キャッシュ、メモリバスの同時アクセスによる干渉等

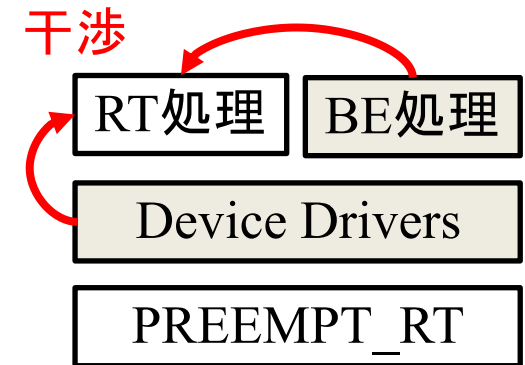


研究活動

・リソース競合を考慮した性能評価方法

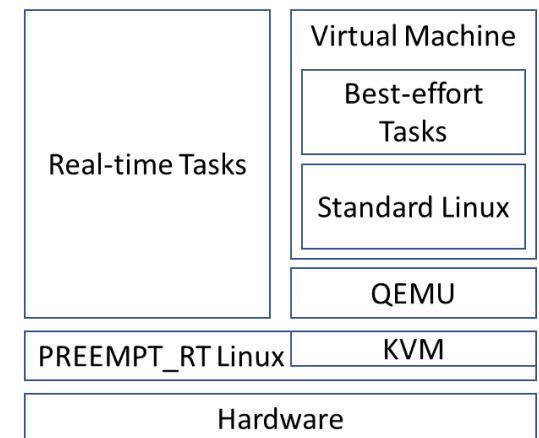
既存評価方法の問題点

- ・ 実験条件は理想すぎる
殆どはRT処理しかないシンプル構成
実世界は複雑なMCS (Mixed-criticality System)
- ・ カーネル以外の干渉要因は未考慮
Linuxのデバイスドライバは膨大
HW (メモリ、キャッシュ等) も性能に影響



構成により干渉要因・発生条件が異なる

- ・ MCS評価環境を構築
- ・ BEタスクで負荷をかけて干渉を評価
多種多様なストレスが望ましい
- ・ ハードウェアの影響も考慮できる
Performance Counter等の情報で分析
- ・ 仮想マシンで干渉の影響を改善
カーネルの分離による効果を測定



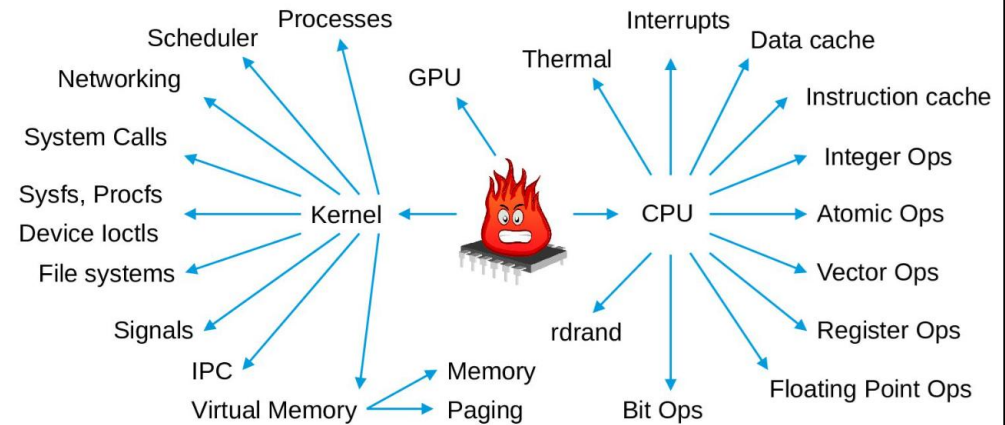
研究活動

・リソース競合を考慮した性能評価方法

干渉負荷にstress-ngを採用

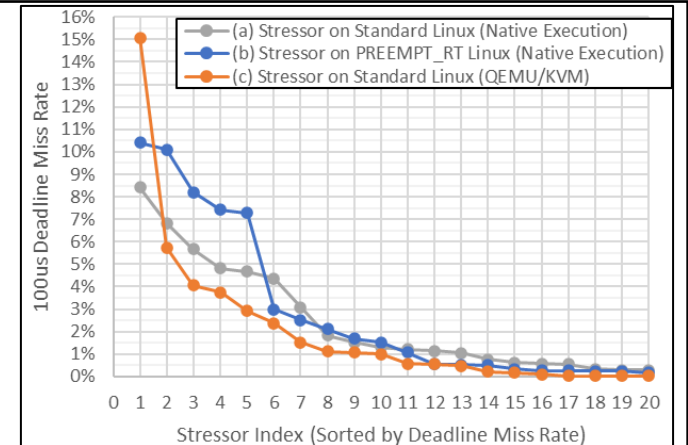
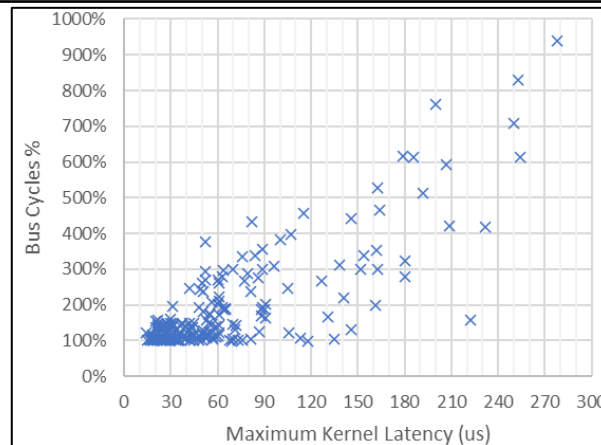
- ・ 大量なストレスを提供
200種類以上のテストプログラム
- ・ システム全体を評価できる
カーネルだけでなく、デバイス
やH/Wの要因も分析可能

※ [Kernel Recipes 2023](#) →



評価データの分析

- ・ 影響大きい負荷を
特定できた
- ・ ハードウェア及び
デバイスの要因に
よる干渉が多いと
判明



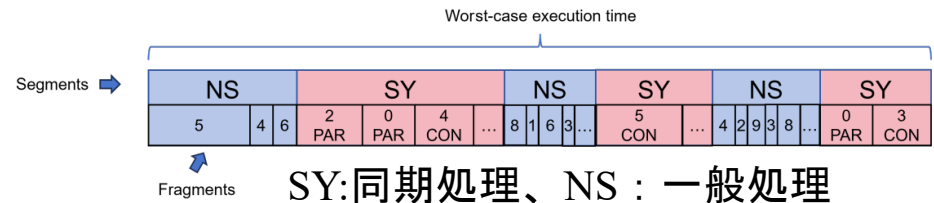
研究活動

・リソース競合を考慮した性能評価方法

タスクセット生成に基づくLinuxベンチマーク

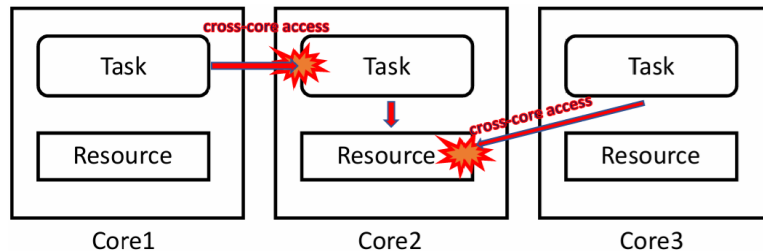
- stress-ngの課題
コア間通信・同期は決められたパターン
でしか行われない
固定タスクセットでタイミング
関連の性能を反映できない

パラメータに従って生成
(競合度合、CPU利用率等)



ランダム生成のタスクセットで評価

- マルチコア制御処理に近い負荷
- 異なるカーネルの性能特徴の比較が可能



(M =コア数、 ρ = SYの割合、 α = コア間競合の割合)

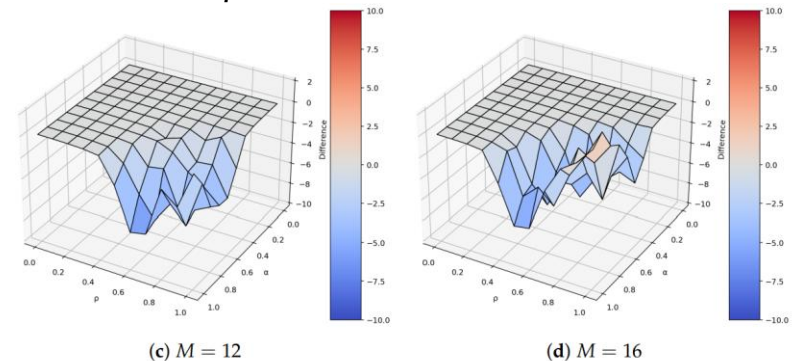
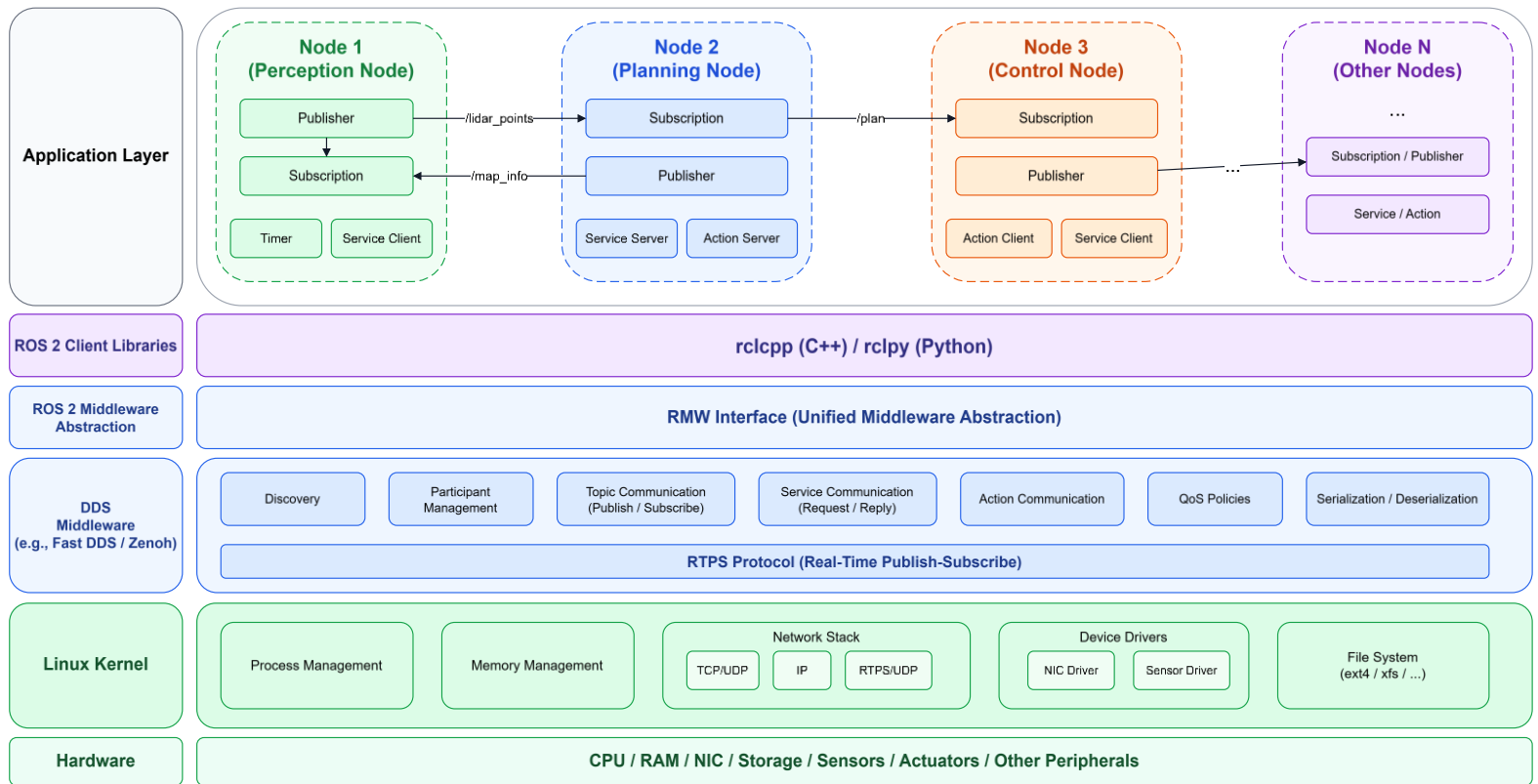


Figure 8. Comparison of differences in average delay ratio (Linux 6.12 PREEMPT_RT – Linux 6.12 PREEMPT).

• ROS2アプリの性能問題解析と改善

規模が大きくボトルネックの解析が難しい

- ・ ロボットには、時間制約が厳しい制御タスクが存在
- ・ ROS2のアーキテクチャは複雑で、ボトルネックが生じやすい

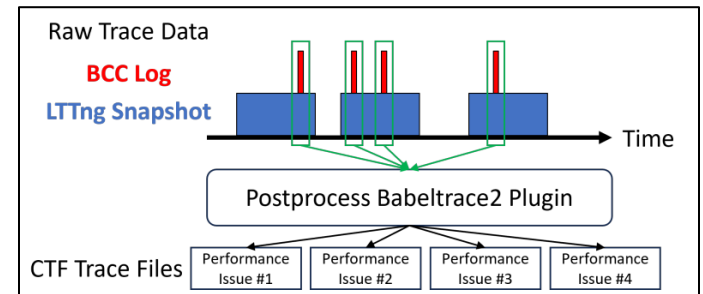
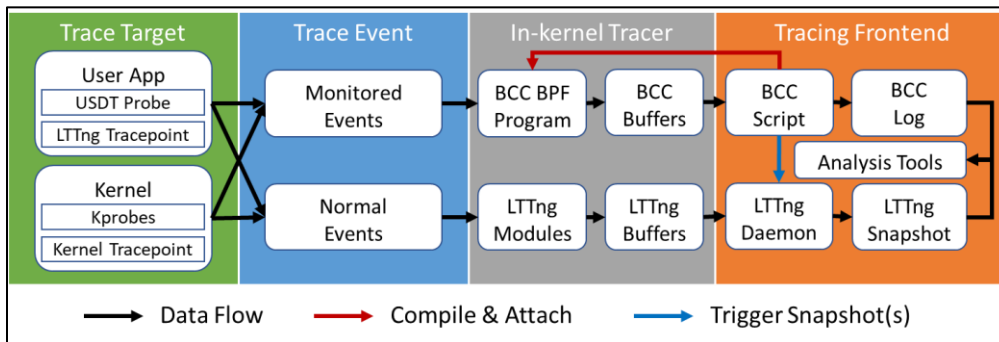


研究活動

• ROS2アプリの性能問題解析と改善

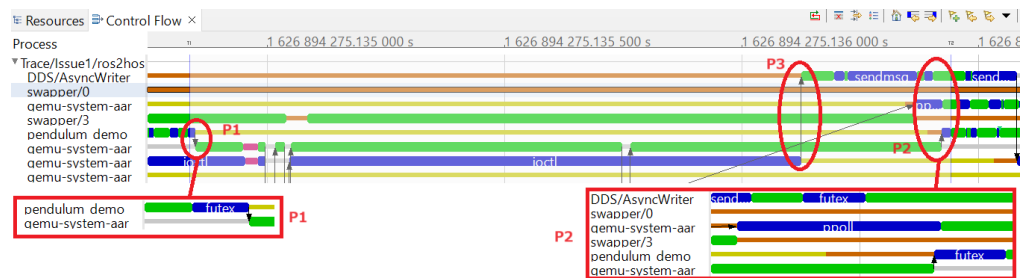
性能問題自動トレースツール

- ・ 低オーバーヘッドでロボット稼働中にも性能影響せず使用可能
- ・ カーネルを含めて全階層のトレースに対応
- ・ 発生頻度が低い性能問題の解析を支援



解析手法の効率化

- ・ 性能問題単位にトレース後処理
- ・ 可視化ツール対応
- ・ AIエージェントで自動解析



研究活動

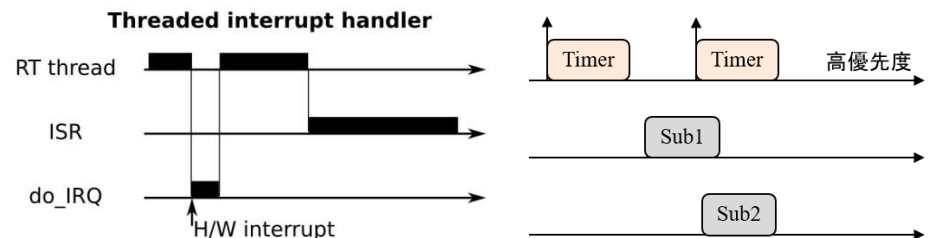
• ROS2アプリの性能問題解析と改善

リアルタイム性向上手法の検証

- 各種処理のThread化で応答性（遅延）改善

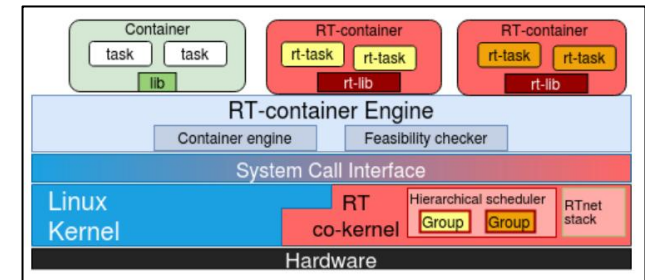
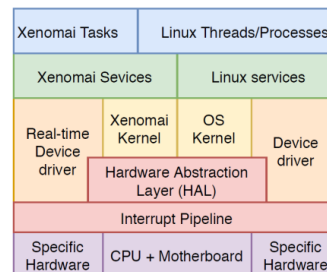
PREEMPT_RT：割り込み

ROS2 Executor：Subscription
Callbacks



- リアルタイムコンテナ

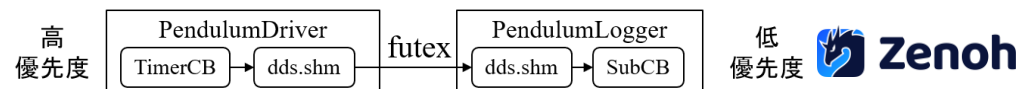
Xenomai4 RTOSで制御Nodeの
CallbackをLinuxカーネルよりも
高い優先度での処理を実現



※10.4230/LIPIcs.ECRTS.2022.15

- 優先度逆転の解消

例：通信ミドルウェアFastDDS
→Zenohに変更

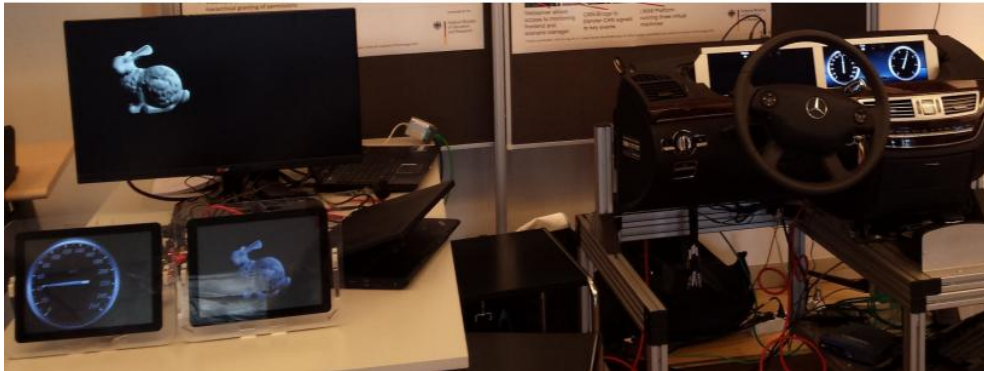


研究活動

• GPU描画処理のリアルタイム性向上

3D描画を行うアプリの数・種類が増加中

- ・ 設計時に負荷を予測できない非信頼アプリが存在
- ・ メーター等優先度が高い処理のFPS保証が必要



※先行研究「Real-time scheduling for 3D rendering on automotive embedded systems」

<http://dx.doi.org/10.18419/opus-10345>

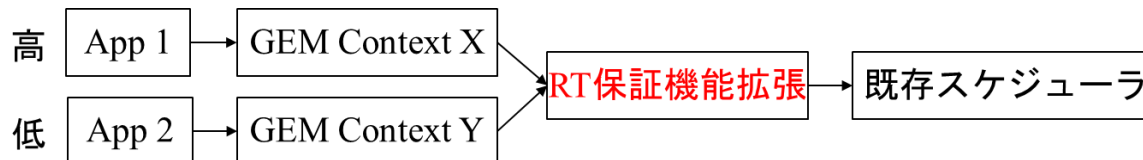
既存研究の問題点

- ・ 殆どは、描画ではなく、GPGPU処理を想定している
タスクの性能特徴と保証目標が異なる
- ・ 非信頼アプリへの適用を考慮していない
正確な設計情報に依存

• GPU描画処理のリアルタイム性向上

Linux Graphics Stackの拡張

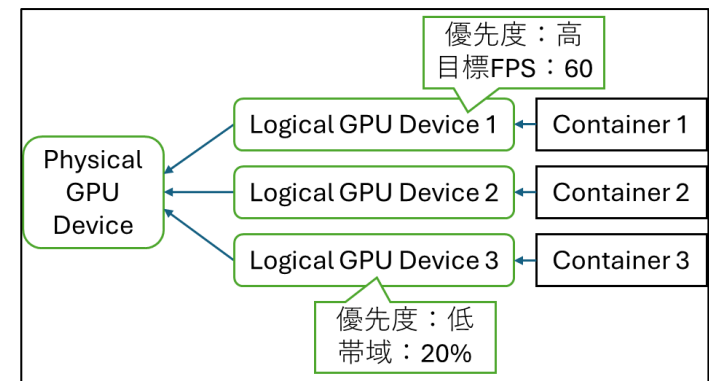
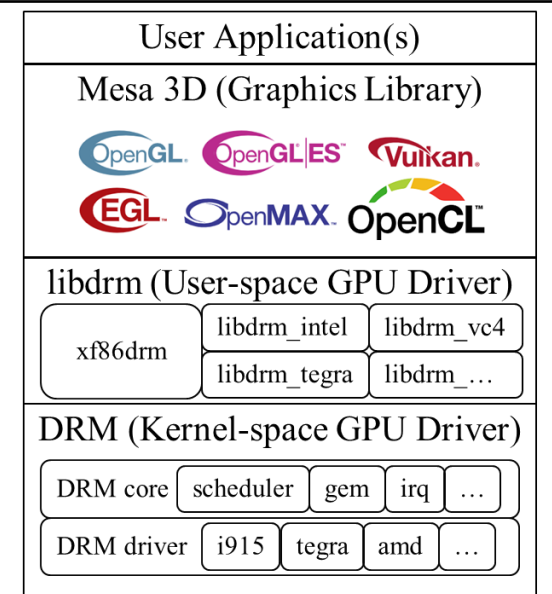
- ・ スケジューラの改善
優先度ベースのリアルタイム性保証機能



- ・ コンテナ対応でGPUリソース分離の粒度を改善

課題：CDI（Container Device Interface）は、GPUデバイスの粒度でしか指定できない

対応：一個のGPUを帯域等でLogical Deviceに分割できる機能を開発

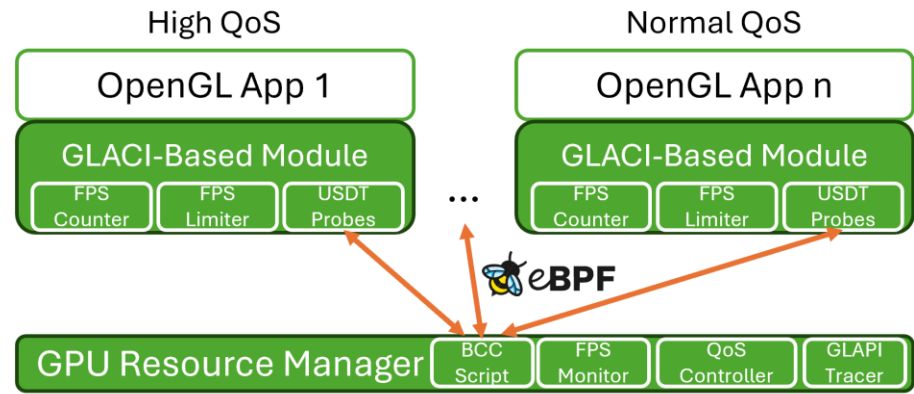
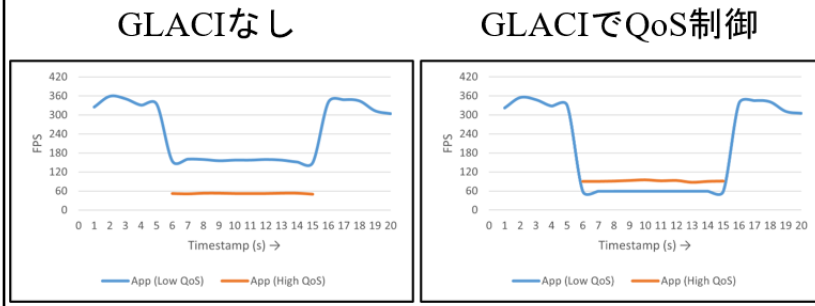
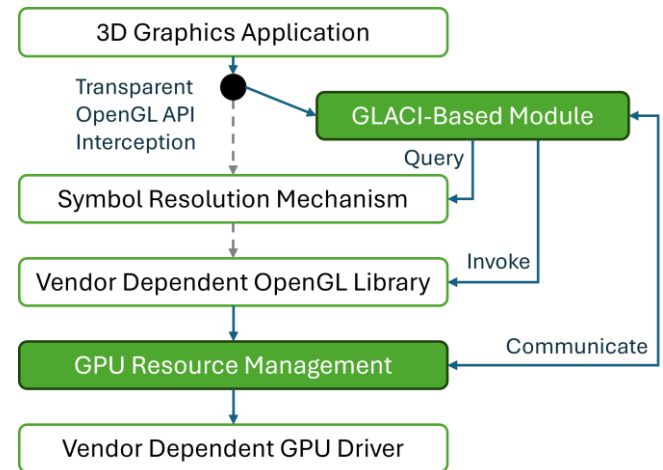


研究活動

• GPU描画処理のリアルタイム性向上

OpenGL描画アプリの対応

- ・ 汎用フックツールGLACIを開発
GPUベンダに依存しない
既存ライブラリのソース修正が不要
機能を実行中に動的変更できる
- ・ 低コストの機能拡張が実現可能に
モニタリングや動的QoS制御等



-
- ご清聴ありがとうございました