Case Study Proposal: Dynamic IoT Applications and Isomorphic IoT Systems Using WebAssembly

Uddhav P. Gautam, Ram Mude
upgautam@vt.edu, ramm@vt.edu

Selected Paper for Case Study

We propose to conduct a case study analysis of the paper *Dynamic IoT Applications and Isomorphic IoT Systems Using WebAssembly* by Kuribayashi et al. [1], published at the 2023 IEEE World Forum on Internet of Things (WF-IoT).

Case Study Overview

This case study will examine the paper's approach to solving IoT deployment challenges through WebAssembly technology. We will analyze the paper's technical contributions, evaluate its methodology and results, and assess its implications for IoT system development. Additionally, we will compare WebAssembly with similar runtime technologies to understand the tradeoffs and advantages of different approaches for dynamic IoT applications.

Rationale for Selection

We selected this paper for our case study because it addresses critical challenges in modern IoT development that are highly relevant to current industry needs:

- **Real-world relevance:** The paper tackles actual problems faced by IoT developers today, including rapid update requirements and multi-layer system complexity.
- **Innovative approach:** The use of WebAssembly for IoT applications represents a novel and promising solution to traditional IoT deployment challenges.
- **Practical implementation:** The authors provide concrete implementation details and performance evaluations, making it suitable for in-depth technical analysis.
- **Comprehensive scope:** The paper covers both dynamic applications and isomorphic systems, providing rich material for case study analysis.

This paper offers excellent opportunities to analyze technical design decisions, evaluate performance tradeoffs, and assess the practical viability of WebAssembly-based IoT solutions. A key learning objective is understanding why the authors chose WebAssembly over other platform-agnostic runtime technologies for their IoT solution.

Case Study Analysis Framework

Our case study will examine the following key aspects of the paper:

1. **Technical Analysis:** How does the WebAssembly-based approach solve traditional IoT deployment challenges?
2. **Architecture Evaluation:** What are the design strengths and limitations of the proposed dynamic and isomorphic IoT systems?
3. **Performance Assessment:** How effective are the authors' performance evaluations and what do the results reveal about practical viability?
4. **Implementation Review:** What insights can be gained from the authors' implementation choices and methodology?
5. **Impact Analysis:** What are the broader implications of this research for IoT system development and deployment?
6. **Technology Comparison:** How does WebAssembly compare to alternative runtime technologies (eBPF, containers, virtual machines, language-specific runtimes) for dynamic IoT applications?

Key Technologies and Concepts to Analyze

Our case study will focus on the following technical elements from the paper:

Core Technologies

We want to validate and see the internals of the system, which was built using WebAssembly (Wasm) runtime and execution environment. We will investigate how the Wasmtube bridging library [2] enables Elixir–Wasm communication and examine the Apache TVM framework [3] for machine learning model compilation. Additionally, we will explore how the Nerves platform [4] facilitates embedded Elixir development and analyze the Linux `inotify` API for dynamic update detection mechanisms.

Performance and Design Analysis

We want to validate and see the internals of the performance improvements, which demonstrated significant update time reductions from traditional firmware updates ($>$10s) to Wasm-based updates ($\sim$1.4s). We will investigate the performance overhead analysis of Wasm function calls ($\sim$200μs) and examine how machine learning model execution performs with ResNet-50 and MobileNetV2. Additionally, we will analyze the cross-platform deployment capabilities and code reuse benefits that enable isomorphic system architectures.

Architecture Evaluation

We want to validate and see the internals of the architecture design, which implements dynamic IoT application design patterns. We will investigate how the isomorphic system architecture operates across device, edge, and cloud layers, examining the security and isolation mechanisms provided by WebAssembly. Additionally, we will analyze the implementation challenges and tradeoffs that the authors encountered during system development.

Educational Value and Learning Outcomes

This case study will enhance our understanding of IoT system design and modern runtime technologies through hands-on analysis of a real research implementation. We will develop practical skills in evaluating technical tradeoffs, understanding performance implications, and making informed technology choices for IoT applications. The comparative analysis with alternative runtime technologies will provide insights into when and why to choose different approaches, preparing us for future IoT system design decisions. Additionally, this study will strengthen our ability to critically analyze research papers, extract practical insights, and apply theoretical concepts to real-world scenarios.

References

[1] K. Kuribayashi, Y. Miyake, K. Rikitake, K. Tanaka, and Y. Shinoda, “Dynamic iot applications and isomorphic iot systems using webassembly,” in *2023 IEEE 9th World Forum on Internet of Things (WF-IoT)*, pp. 1–8, 2023.

[2] K. Kuribayashi, “Wasmtube: A bridging library which allows you to communicate between elixir and wasm.” `https://github.com/kentaro/wasmtube`, 2023. Accessed 2023.

[3] Apache Software Foundation, “Apache tvm: An end to end machine learning compiler stack.” `https://tvm.apache.org/`, 2023. Accessed 2023.

[4] The Nerves Project, “Nerves: Build bulletproof, embedded software in elixir.” `https://nerves-project.org/`, 2023. Accessed 2023.