Agents of Autonomy: A Systematic Study of Robotics on Modern Hardware

Mohammad Bakhshalipour
bakhshalipour@cmu.edu
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

Phillip B. Gibbons
gibbons@cs.cmu.edu
Carnegie Mellon University
Pittsburgh, Pennsylvania, USA

ABSTRACT
As robots increasingly permeate modern society, it is crucial for the system and hardware research community to bridge its long-standing gap with robotics. This divide has persisted due to the lack of (i) a systematic performance evaluation of robotics on different computing platforms and (ii) a comprehensive, open-source, cross-platform benchmark suite.

To address these gaps, we present a systematic performance study of robotics on modern hardware and introduce RoWild, an open-source benchmark suite for robotics that is comprehensive and cross-platform. Our workloads encompass a broad range of robots, including driverless vehicles, pilotless drones, and stationary robotic arms, and we evaluate their performance on a spectrum of modern computing platforms, from low-end embedded CPUs to high-end server-grade GPUs.

Our findings reveal that current architectures experience significant inefficiencies when executing robotic workloads, highlighting the need for architectural advancements. We discuss approaches for meeting these requirements, offering insights for improving the performance of robotics.

The full version of the paper is available in [11], and the source code of the benchmark suite is available in [2].

CCS CONCEPTS
• Computing methodologies → Robotic planning; Modeling methodologies; • General and reference → Measurement; Evaluation; Performance; • Computer systems organization → Real-time system architecture; Architectures.

KEYWORDS
Robotics, Computer Architecture, Benchmarking

1 INTRODUCTION
The advancement of robotics technology is rapidly changing the world we live in. With predictions of 20 million robots by 2030 [1] and a market capitalization of US$210 billion by 2025 [23], it is clear that robotics will play an increasingly important role in society. To become widespread, robots need to meet the demands of real-world environments, which necessitates them being autonomous and capable of performing complex artificial intelligence (AI) tasks in real-time.

Computer hardware and architecture play a paramount role in realizing real-time robotics, evidenced by the deployment of robot-specific hardware accelerators in the architecture of the latest edge processors. Recent robotic platforms [14, 15, 17] include hardware accelerators for operations like tree-extension and ray-casting that have massive usage in robotics; Intel’s multi-robot system [14] has a full-fledged “Robot SoC.”

Surprisingly, the computer systems research community has under-explored robotics. This is borne out by the scant few publications in top computer systems conferences. There is a large gap between robotics and the computer systems community, depriving robotics of many improvements achievable by system-level techniques.

The gap is largely because of the lack of (i) a systematic performance study and (ii) a comprehensive, open-source, cross-platform benchmark suite. As a result of (i), the robotic tasks, their performance requirements, and their system-level implications are unclear to the community. And, due to (ii), the few research papers include only one [9, 19] or a couple of applications [20, 22] in their evaluations, leaving the impact on other applications unknown.

2 THE ROWILD BENCHMARK SUITE
This paper aims to bridge this gap by introducing RoWild, a comprehensive, open-source, cross-platform robotic benchmark suite. The challenge of benchmarking robotics lies in the vast array of applications, from self-driving cars to home-assistant robots, with more to come in the future. It is impractical to represent all of these applications in a single benchmark suite. RoWild overcomes this challenge by exploiting the fact that different robots, despite their different applications, perform a finite set of common ”robotic tasks” such as scene understanding and pathfinding. For instance, both self-driving cars and home-assistant robots require scene understanding. Nevertheless, the algorithms and constraints in conducting such tasks can vary widely across different applications.

RoWild comprises a wide range of robotic tasks, encompassing the software pipeline of practically all autonomous robots. With versatility in mind, RoWild implements each task with various algorithms and parameters. This flexible approach enables the configuration and pipelining of tasks to model the end-to-end computation of diverse robotic applications. By including a broad set of tasks and algorithms,
RoWild is capable of modeling numerous robotic applications, thus providing a comprehensive benchmark suite.

Our choice of implemented tasks and algorithms stems from an analysis of 29 industrial robots, encompassing a diverse range from arm manipulators and home-cleaning robots to self-driving vehicles. This analysis was undertaken to ensure a broad yet relevant selection. Additionally, we incorporate state-of-the-art research algorithms (e.g., deep learning-based pathfinding) into RoWild, positioning it as a suitable suite for future robotics. We develop RoWild with essential considerations: specifically, RoWild is high-performance, simulator-friendly, versatile, and modular. Table 1 compares key features of RoWild and the related work.

### Table 1: Comparison of related work features with RoWild. More ✔ is better. ✘ shows variable outcomes (X or △) based on specifics like kernel or simulator.

<table>
<thead>
<tr>
<th>Paper/Repository</th>
<th>Scope</th>
<th>End-to-End</th>
<th>High-Friendly</th>
<th>Simulator-Friendly</th>
<th>Multi-Level</th>
<th>Versatile</th>
<th>System-Level Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin et al. [18]</td>
<td>Self Driving Cars</td>
<td>✔</td>
<td>✔</td>
<td>Unknown (private)</td>
<td>✔</td>
<td>Unknown (private)</td>
<td>✔</td>
</tr>
<tr>
<td>Shao et al. [9, 13, 16]</td>
<td>Single Task</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
</tr>
<tr>
<td>RoWild</td>
<td>Bread</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

### 3 COMPREHENSIVE PERFORMANCE STUDY

This paper’s second contribution is to investigate the system-level implications of robotics. Using RoWild’s tasks, we model six different end-to-end robotic applications and evaluate them on a spectrum of platforms, ranging from low-end embedded CPUs to high-end server-grade GPUs. Table 2 summarizes the modeled applications, and Table 3 presents the evaluated platforms.

### Table 2: The modeled end-to-end applications.

<table>
<thead>
<tr>
<th>Name</th>
<th>Mission</th>
<th>Environment</th>
<th>Platform</th>
<th>Core</th>
<th>Freq. (GHz)</th>
<th>TDP (W)</th>
<th>Memory (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DellBot</td>
<td>Delivery</td>
<td>Our Campus</td>
<td>ARM Cortex A57 CPU</td>
<td>4</td>
<td>1.43</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>MoveBot</td>
<td>Manipulation</td>
<td>Synthetic</td>
<td>Nvidia Maxwell GPU</td>
<td>128</td>
<td>0.92</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>FlyBot</td>
<td>Photography</td>
<td>FR Campus [4]</td>
<td>Intel Xeon Gold CPU</td>
<td>20 (x2)</td>
<td>2.10</td>
<td>125</td>
<td>384</td>
</tr>
<tr>
<td>PatrolBot</td>
<td>Home</td>
<td>Cleaning</td>
<td>Nvidia Titan X GPU</td>
<td></td>
<td>3584</td>
<td>1.41</td>
<td>250</td>
</tr>
</tbody>
</table>

While previous studies [10, 12] have conducted some system-level analysis (e.g., CPU vs. GPU) specific to their applications, these analyses remained at a high level. In contrast, our study delves deeper to investigate low-level implications, including the efficacy of caching, prefetching, and vectorization.

### 4 EVALUATION HIGHLIGHTS

Our system-level investigations reveal significant inefficiencies in the architecture of today’s prevailing compute platforms when executing robotic workloads. Specifically, we find:

- **Vectorization Is Ineffective.** Despite the large silicon real-estate it occupies, CPU vectorization does little for robotic tasks: in the common case of an axis-unaligned orientation, the robot’s memory layout is not vectorization-friendly.
- **On-Edge Parallelism Hits The Memory Wall.** Massive parallelism on edge platforms (e.g., Nvidia’s Jetson Nano) is bottlenecked by memory: the memory wall is hit well before Amdahl’s law.
- **Simple Prefetchers Are Inadequate.** While simple data prefetchers can help with robotics workloads, complex AI algorithms used in robotics often use irregular data structures, producing memory patterns that defeat commercial prefetchers.
- **Caches Perform Significant Unnecessary Data Movements.** Hardware caches are unaware of the robots’ software semantics; in many cases, caches work in opposition to them. As a consequence, caches perform excessive data movements, utilizing the memory hierarchy inefficiently.

### REFERENCES

[17] Youchang Kim et al. 2016. A 0.65V 1.1mW Artificial-Intelligence Processor with PVT Compensation for Micro Robots. In ISSCC.
[19] WEIZHUANG LIU ET AL. 2021. ARCHYTAS: A FRAMEWORK FOR SYNTHESIZING AND DYNAMICALLY OPTIMIZING ACCELERATORS FOR ROBOTIC LOCALIZATION. In MICRO.