Thesis Proposal for Revolution on the road: optimizing resource management and improving deep learning models' efficiency in autonomous vehicle systems

Aiza Maksutova,
Faculty of Computer Science, Department of Applied Mathematics and Information Science
National Research University Higher School of Economics
Moscow, Russia
aamaksutova@edu.hse.ru

***Abstract*—The advent of autonomous vehicles (AVs) marks a significant milestone in robotics, marking a new era of transportation. Despite their growing popularity, the functionality of AV systems often needs to be more robust. Due to the absence of comprehensive documentation on the technology underlying these vehicles, it is not easy to understand and analyze it. This opacity, particularly concerning the hardware utilized, poses substantial challenges for engineers seeking to harness and optimize these systems fully. This thesis aims to demystify one of the most vital components in contemporary AV and robotics applications: the Jetson AGX Orin. Through a detailed benchmarking study, this work seeks to elucidate the capabilities and limitations of this device within the context of AV systems. Furthermore, it proposes a series of optimizations designed to maximize the efficiency and performance of autonomous vehicles by leveraging the full potential of the Jetson AGX Orin. By providing a clearer understanding of this critical hardware component, this thesis enhances the reliability and efficacy of autonomous vehicles.**

***Keywords*—Jetson AGX Orin, Autonomous Vehicles (AVs), bandwidth, SSD, benchmarking systems, robotics, CPU, GPU, LPDDR5 memory, eMCC flash memory, bottlenecks in AV pipelines**

I. Introduction

The rapid advancement of autonomous vehicle (AV) technology represents a significant leap in machine learning and robotics. Despite these strides, a critical gap remains in the comprehensive understanding and utilization of the hardware devices at the core of these systems. This underutilization hinders efficiency and prolongs models' training and inference times, presenting a substantial bottleneck in their development and deployment. This thesis contends with the nuanced interplay between AV engineering and the hardware devices predominantly employed in the field. Specifically, it seeks to bridge the knowledge gap that often leads to suboptimal exploitation of these devices. The focal point of this investigation is the Jetson AGX Orin, a cornerstone in AV hardware whose potential has yet to be fully realized due to the lack of detailed documentation and understanding of its operational dynamics.

Drawing on the foundational work presented in the "D3: A Dynamic Deadline-Driven Approach for Building Autonomous Vehicles" [1] paper, this research adopts a structured approach to dissecting the standard AV pipeline. It aims to identify and benchmark the performance of critical components such as SSD and eMCC flash memory, LPDDR5 memory, and CPU/GPU interactions. The methodology encompasses a blend of empirical benchmarking, leveraging tools and insights from the official NVIDIA Jetson AGX Orin documentation [2] and an array of scholarly resources within operating systems.

The novelty of this thesis lies in its targeted focus on the Jetson AGX Orin, addressing the conspicuous absence of in-depth analysis regarding its component performance under varying operational stresses. This endeavor is poised to unravel the intricacies of device performance, offering a granular understanding that transcends the current landscape of generalized hardware analyses.

Anticipated outcomes include a comprehensive performance analysis of important device components under diverse system loads and strategic optimization recommendations. These insights are expected to signal a new era of efficiency in AV pipelines, enhancing their real-world applicability and performance.

The paper unfolds structured, beginning with an exposition of prevalent AV pipeline architectures. This foundation sets the stage for a deep dive into identifying critical bottlenecks warranting rigorous benchmarking. Subsequent sections delve into the benchmarking methodology, component performance analyses, and the articulation of optimization strategies. The culmination of this thesis presents a cohesive framework for enhancing the efficiency and practicality of AV systems, marking a significant contribution to the field.

II. Literature Review

The evolution of autonomous vehicle (AV) technology necessitates a nuanced understanding of its underlying computational frameworks. A pivotal contribution to this domain is the "D3: A Dynamic Deadline-Driven Approach for Building Autonomous Vehicles" paper [1], which delineates the core

modules integral to AV operation: perception, localization, prediction, planning, and control. This paper meticulously maps these modules to their respective computational domains, with localization and planning primarily allocated to CPU processes and the more computationally demanding perception tasks reliant on GPU support. This delineation underscores the inherent tradeoffs between computational accuracy and temporal efficiency within AV pipelines, a balance critical to ensuring vehicular safety and operational fluidity.

The D3 methodology emerges as a novel paradigm in organizing AV computational pipelines, leveraging a directed operator graph alongside a deadline policy (Π_{DP}) to enforce an end-to-end deadline (D). This approach ensures vehicular safety by preempting excessive response times and unnecessary emergency maneuvers and optimizes the runtime-accuracy tradeoff by allocating individual deadlines (D_i) to different operators. The dynamic nature of Π_{DP} allows for real-time adjustments in response to missed deadlines, ensuring a minimal risk condition through proactive corrective measures or, in extreme cases, activating a safety backup mode.

While the D3 approach offers a structured framework for managing computational tasks within AV systems, it does not directly address the challenges of efficiently utilizing hardware resources. Reading this paper, I highlighted potential bottlenecks such as the transition between SSD and eMMC flash memory, the limitations imposed by LPDDR5 memory bandwidth, and the intricacies of CPU/GPU dynamics within the Jetson architecture. These areas represent critical investigation points for enhancing AV systems' performance and efficiency. The D3 methodology, while invaluable in understanding and organizing computational tasks, leaves open the question of how to maximize hardware utilization to improve overall system performance.

Building upon the foundational insights provided by the "D3: A Dynamic Deadline-Driven Approach for Building Autonomous Vehicles" [1] and recognizing the computational bottlenecks inherent in AV systems, it becomes imperative to explore the hardware capabilities that underpin these technologies. A critical component in this exploration is the Jetson AGX Orin, a device that has recently entered the market and is pivotal in developing advanced AV systems. Despite its potential, the nascent stage of the Jetson AGX Orin in the market landscape has resulted in a lack of comprehensive benchmarks that delineate its performance across varied operational conditions.

Without extensive external benchmarks, the official NVIDIA documentation [2] is a primary resource for understanding the Jetson AGX Orin's architecture and its components. The documentation offers an in-depth view of the device's internal connections, bandwidth capacities, and memory configurations. Notably, it specifies a singular PCIe (Peripheral Component Interconnect Express) link between the eMMC flash memory and SSD, with a bandwidth capacity of 4GB/s. Furthermore, it elaborates on the theoretical maximum bandwidth of LPDDR5 memory at 204.8 GB/s, coupled with a significant 2 MB L3 and 4 MB system cache facilitating data flow between the CPU cores and LPDDR5 memory.

This foundational knowledge, gleaned from the official documentation, informs the experimental approach of this thesis. It highlights critical areas of focus, such as the PCIe bandwidth and memory throughput, which are essential for understanding the device's operational limits. These insights establish a baseline for the anticipated benchmarks, guiding the experimental design to explore the Jetson AGX Orin's performance thresholds and identify optimization opportunities within AV pipelines.

III. Methodology

In the initial phase of my research, I focused on examining the data transfer bandwidth between Solid State Drives (SSDs) and embedded MultiMediaCard (eMMC) Flash Memory, which are critical components in the data management systems of autonomous vehicles. Given the importance of the swift model and data loading/unloading in maintaining the operational efficiency of these vehicles, I posited that the bandwidth utilization between SSDs and eMMC directly influences the speed of these processes. More specifically, I hypothesized that an increased bandwidth occupancy would correlate with faster data handling speeds, particularly in the context of loading and offloading models. To empirically test this hypothesis, I employed the 'fio' [3] tool on a Linux platform renowned for simulating real-world I/O workloads by generating specific read/write operations. This tool enabled me to mimic diverse data transfer scenarios between the SSD and eMMC Flash memory, replicating various model/data interchange workflows typically encountered in autonomous vehicle systems. By systematically varying the simulated data workloads, I aimed to observe and measure the impact of bandwidth utilization on the efficiency of data transfer processes, thereby testing the validity of my initial hypothesis.

After examining the bandwidth between SSDs and eMMC Flash Memory, my investigation progressed to the dynamics of model loading from SSDs into the device's internal memory. This phase specifically concentrated on utilizing Open Neural Network Exchange (ONNX) models [4], a format prevalently adopted in autonomous driving systems, to analyze the efficiency of data transfer processes. To assess the transfer times quantitatively, I developed a custom benchmarking tool that measures the duration required to load and offload various ONNX models, considering their differing sizes. The central hypothesis guiding this research phase posited that segmenting the models into more minor, manageable slices would enhance the transfer speed. This is predicated on the assumption that smaller data packets could utilize the available bandwidth more effectively, facilitating quicker data transfers via the Peripheral Component Interconnect Express (PCIe) interface. The experimental setup involved systematically altering the model sizes and observing the corresponding impact on transfer times, aiming to ascertain the optimal model segmentation strategy that maximizes data transfer efficiency through increased bandwidth utilization.

Upon concluding my analysis of the PCIe interface's role in SSD and eMMC flash memory interactions, my research focused on LPDDR5 memory, specifically its bandwidth and its implications for CPU performance. The primary objective was to ascertain the read and write speeds of the CPU with LPDDR5 memory. This is crucial for understanding the efficiency with which computational models, relying predominantly on CPU resources, can access and manipulate data.

To explore these dynamics, I employed the 'lmbench' [5] benchmarking tool, renowned for evaluating UNIX/POSIX systems by measuring key performance metrics such as latency and bandwidth. This tool facilitated a nuanced examination of the CPU's performance concerning LPDDR5 memory, distinguishing it from the potential influence of cache memories.

I hypothesized that the apparent bandwidth of LPDDR5 memory might be constrained by the bandwidths of the L3 and system caches, thereby necessitating a meticulous approach to isolate the LPDDR5 memory's performance metrics. To address this, the methodology involved incrementally loading the CPU to engage the L3 cache, followed by the system cache, ultimately focusing on the LPDDR5 memory. This hierarchical approach was designed to ensure that the measured bandwidth accurately reflected the LPDDR5 memory's capabilities, devoid of cache effects.

Initial tests were conducted on a single CPU core to establish a baseline, which was subsequently extrapolated to a 12-core configuration, representing the maximum core count available on the Jetson AGX Orin platform. This strategy aimed to comprehensively assess the LPDDR5 memory's bandwidth under varied computational loads, testing the hypothesis that the LPDDR5 memory bandwidth would demonstrate stability across different operational scales beyond cache limitations.

IV. Results

In this section, I present the findings of my experimental evaluations, focusing on benchmarking bandwidths and latencies across various system components. These results are pivotal in understanding the performance bottlenecks and efficiencies within the data management strategy.

1) PCIe Bandwidth Benchmarking: My initial experiments, as depicted in Figure 1, were aimed at benchmarking the bandwidth of the Peripheral Component Interconnect Express (PCIe) between the Solid-State Drive (SSD) and the embedded MultiMediaCard (eMMC) flash memory. The results from these tests indicate a notable latency peak during the sequential write operations from the eMMC flash memory to the SSD. This observation is critical as it unveils a potential bottleneck in the data transfer process from internal memory to remote storage, a fundamental operation in the data management framework.

2) Model Loading Latency: Subsequent tests, illustrated in Figure 2, focused on the latency involved in loading models from the SSD into the internal memory. This aspect is particularly salient in scenarios where the device's storage capacity is constrained, yet there is an immediate requirement

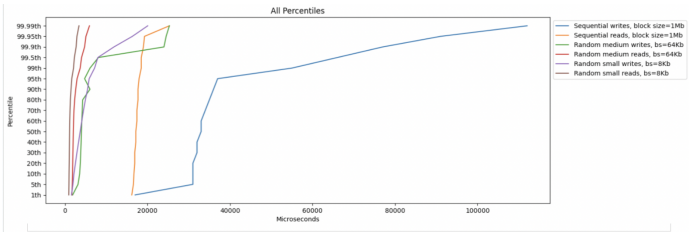

Fig. 1. Bandwidth analysis of PCIe between SSD and eMMC flash memory

to load essential models, such as those for anomaly detection. The rapid loading of such models is crucial for enabling the device to swiftly transition into a safety mode in response to potentially hazardous situations.

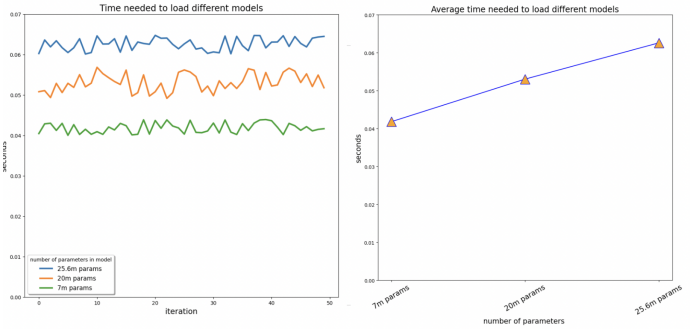

Fig. 2. Analysis of model loading timings

3) Data Offloading Impact: In Figure 3, I explore the impact of data offloading on the PCIe bus that facilitates communication between the SSD and the eMMC. This factor is vital, especially under concurrent operations or when there is an imperative need for quick data transmission to cloud services. Aligning with the earlier latency trends, my findings highlight that writing data to the SSD, particularly during model offloading, is significantly more time-consuming, averaging 0.1 seconds longer than model loading operations. This discrepancy further emphasizes the importance of devising efficient data management strategies to cater to high-demand scenarios.

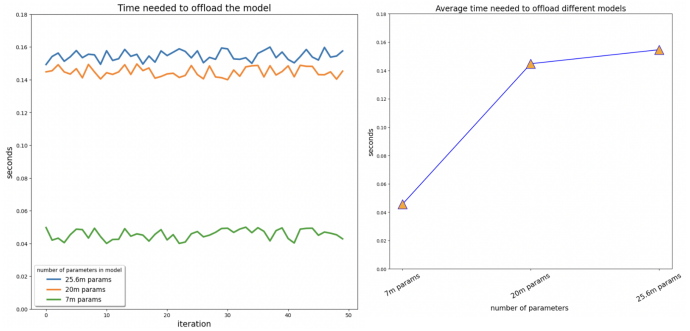

Fig. 3. Analysis of model offloading timings

4) LPDDR5 Memory Bandwidth Analysis: My comprehensive analysis of the LPDDR5 memory bandwidth is presented

in Figure 4. The findings reveal that for data loads below 2 MB, the operations predominantly utilize the L3 cache, achieving an impressive bandwidth of approximately 8000 MB/s on a single CPU core. As the size of the data transfers escalates, the system transitions to leveraging the system cache bandwidth, leading to a decline in performance due to the inherent constraints of the system cache. For larger data transfers, the bandwidth further diminishes to around 3000 MB/s, indicating direct engagement with the LPDDR5 memory.

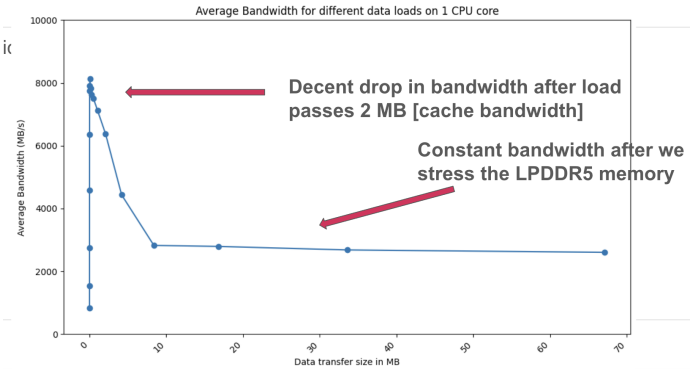

Fig. 4. Analysis of LPDDR5 memory bandwidth

5) Comparison with Official Documentation: Lastly, Figure 5 addresses the discrepancies between my experimental bandwidth measurements for the LPDDR5 memory, which averaged around 3 MB/s, and the specifications cited in the official Nvidia documentation. The documentation may refer to the maximal theoretical bandwidth achievable under optimal conditions, a metric that can diverge significantly from practical performance outcomes.

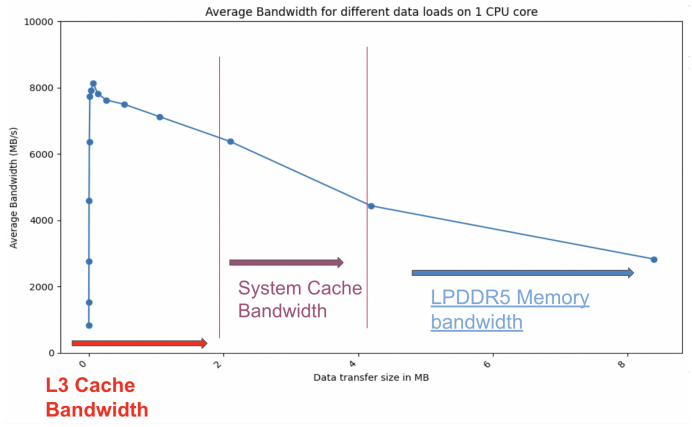

Fig. 5. Analysis of LPDDR5 memory bandwidth

V. Conclusion

The investigation into system performance has uncovered significant findings essential for advancing data transfer efficiency and optimizing system responsiveness. The identified latency during sequential write operations from eMMC to SSD is particularly critical, as it highlights a significant bottleneck that could hinder performance in autonomous vehicles applications. This finding is invaluable for developing strategies to enhance data transfer rates and ensure system reliability.

Furthermore, the latency in loading critical models underscores the necessity for rapid data access in real-time applications, where delays can have severe implications. This analysis emphasizes the importance of designing AV systems with swift retrieval capabilities to support urgent operational requirements.

The engagement of the PCIe bus during data offloading operations and the longer duration compared to loading operations underline the need for efficient data management, especially in scenarios with high demand and concurrent operations. This aspect is crucial for maintaining high system performance during the autonomous car's work and avoiding data transfer bottlenecks.

Additionally, the observed discrepancies between theoretical and practical bandwidths for LPDDR5 memory signal a gap in expected versus actual performance, stressing the importance of empirical evaluations in system design. These observations provide a foundation for enhancing AV systems architectures, influencing theoretical approaches, and optimizing practical implementations of data management systems.

In the next phase of my research, I aim to explore the integration of Jetson AGX Orin with cloud servers, a critical development for benchmarking rapid communication capabilities between vehicles on the road. This connection has the potential to significantly enhance vehicular performance by enabling more efficient data exchange and processing, laying the groundwork for advanced vehicular communication systems and more innovative, more responsive transportation networks.

References

[1] "D3: a dynamic deadline-driven approach for building autonomous vehicles." https://sukritkalra.github.io/data/papers/erdos.pdf, accessed: 2022-03-22.

[2] "Jetson agx orin developer kit user guide - hardware," https://developer.nvidia.com/embedded/learn/jetson-agx-orin-devkit-user-guide/developer_kit_layout.html, accessed: 2023-02-22.

[3] "fio(1) - linux man page," https://linux.die.net/man/1/fio, accessed: 2023-02-22.

[4] "Onnx models repository," https://github.com/onnx/models, accessed: 2023-02-22.

[5] "Lmbench github," https://github.com/intel/lmbench, accessed: 2023-02-22.