Concept paper (proposal) for the master´s project

Name: Jiajia Zhang

Matriculation number: 03751019

(Project-) Title: A Real-Time Hypervisor for the Nvidia Orin

Institute: Chair of Cyber-Physical Systems in Production Engineering

Name of supervisor: Dr. Alexander Züpke

I Content presentation created by:

München, 11.10.2024

Place, date

Zhang, Jiajia

Student

Examiner

1. Topic

In modern industrial fields, there is a growing demand for high-performance embedded computing platforms like the NVIDIA Jetson AGX Orin Series. Applications such as autonomous robotics in manufacturing, precision agriculture, medical imaging devices, and intelligent infrastructure rely heavily on the processing power offered by platforms like the NVIDIA Jetson AGX Series. These systems need to process huge amounts of data, often involving real-time decision-making. The Jetson AGX Orin module is composed of a 12-core ARM Cortex-A78AE CPU, an NVIDIA Ampere GPU, and dedicated accelerators for deep learning (DLAs) and vision tasks and delivering up to 275 TOPS of AI performance, High speed IO, 204 GB/s of memory bandwidth, and 32GB or 64GB of DRAM enable these modules to meet the growing demands of advanced AI applications [1].

Installing a hypervisor on the NVIDIA Orin board, such as Jailhouse, can enable the partitioning of hardware resources, allowing multiple operating systems or tasks to run simultaneously without interfering with each other. In real-time systems, hypervisors ensure that critical tasks have dedicated resources, improving performance and predictability. Moreover, features like per-core partitioning and cache isolation provided by hypervisors can be crucial in optimizing resource allocation and preventing cache contention in multi-core environments.

This thesis aims to investigate the porting of the Jailhouse hypervisor to the NVIDIA Jetson AGX Orin platform, focusing on evaluating the effectiveness of per-core partitioning and cache partitioning. By exploring these aspects, the research will contribute to better resource management and performance optimization for real-time systems running on multi-core embedded platforms.

2. State of science and technology

The NVIDIA Jetson Platform and The NVIDIA Jetson Orin Series

The NVIDIA Jetson platform is designed to meet the growing demand for high-performance AI and edge computing. It enables real-time, energy-efficient AI computing, making it ideal for embedded systems that need to process data locally without relying on cloud-based services. The Jetson AGX Orin Developer Kit used for this thesis includes a Jetson AGX Orin module with heatsink, a reference carrier board, and a power supply. The NVIDIA Orin system on chip (SoC) of the Jetson AGX Module is organized into three main processing complexes as shown in **Figure 1**: CPU, GPU, and hardware accelerators. For the 64GB module, it has 12 Cortex-A78AE arm cores [1].

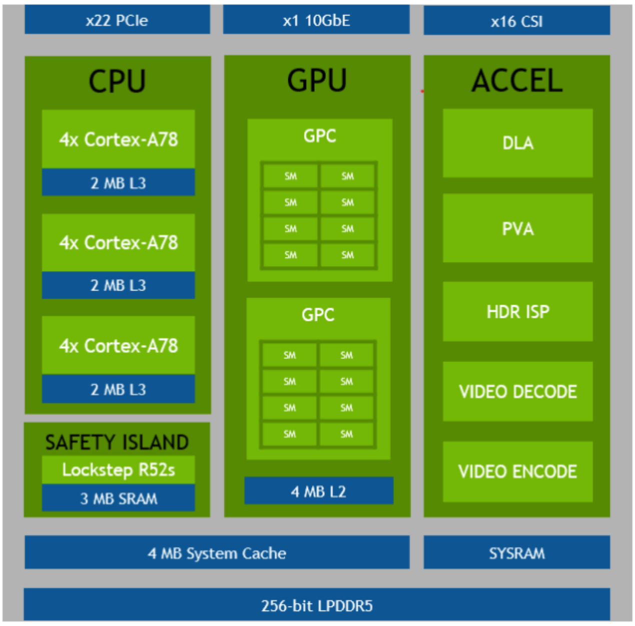


Figure 1: Orin System-on-Chip (SoC) Block Diagram [1]

Arm Architecture and Exception Level

The ARM architecture is the basis for the design of processors, called Processing Elements (PEs), which define how the CPU executes instructions, handles data and manages hardware resources. It is widely used in a variety of technologies, including system-on-chip (SoC) devices found in smartphones, embedded systems, servers and even supercomputers. The three architecture profiles: A, R and M, allow the ARM architecture to be adapted to different use cases. The A-profile (Application) is designed for high-performance systems running complex operating systems like Linux or Windows, commonly used in smartphones and computers. The R-profile (Real-Time) targets systems with real-time requirements, such as networking equipment and embedded control systems. The M-profile (Microcontroller) focuses on small, power-efficient devices, often found in IoT applications. The Cortex-A78AE used in this thesis is based on the ARMv8-A architecture, which was announced in 2011 and was the first 64-bit version of the ARM Architecture. ARM supports four exception levels in AArch64 to manage different privilege levels, as shown in **Figure 2**. From EL 0 to EL 3, the privilege increases, that means more hardware abilities are activated. It supports two distinct worlds: the Secure World and the Non-Secure World. The Secure World, using Trust Zone technology, handles trusted operations like managing sensitive data, while the Non-Secure World is responsible for running general applications [2].

- EL0: User space, where regular applications run with the least privilege.
- EL1: Kernel space, where the operating system runs, with more control over hardware.
- EL2: Hypervisor level, responsible for managing virtual machines.
- EL3: Secure monitor, managing transitions between secure and non-secure worlds.

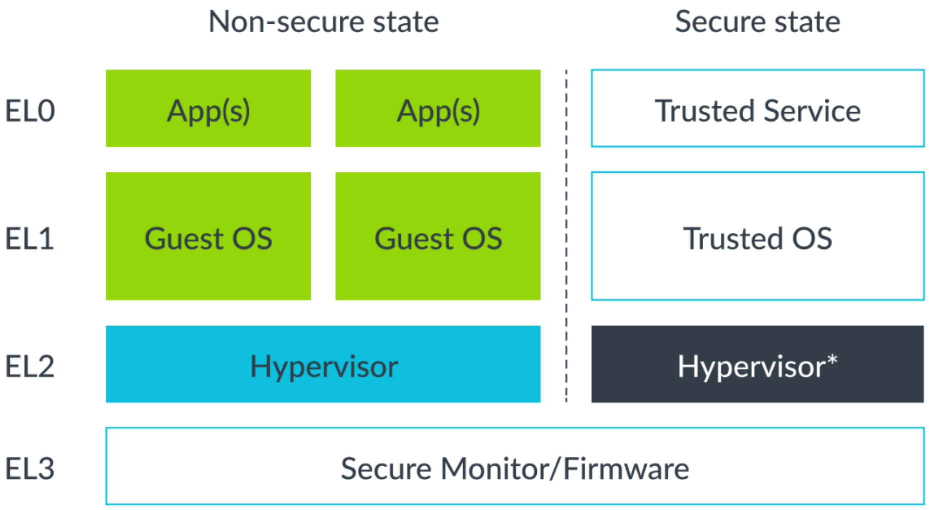


Figure 2: Exception levels of ARMv8-A architecture [2]

Jailhouse Hypervisor

A hypervisor, also known as a Virtual Machine Monitor (VMM), enables multiple operating systems running on one single physical server. In such setup, different services, like email servers, web servers, FTP servers, and e-commerce servers do not need to run on separate machines. However, on the virtualized system based on hypervisor, each operating system runs in its own isolated environment. That means if one virtual machine fails, it does not affect the others. This allows for better resource utilization, cost savings, and easier maintenance, while also providing security benefits by containing potential attacks or failures within individual VMs. Hypervisors can be divided into two categories: Type 1 and Type 2. The Type 1 hypervisor runs directly on the hardware, they offer better performance, efficiency, and security because they have direct access to the hardware resources. The Type 2 hypervisors rely on the host operating system for device support and resource management [3].

Jailhouse is an open-source, Type 1 hypervisor that is based on Linux (**Figure 3**). It divides hardware into isolated units known as "cells". Each cell can control the resources assigned to it, ensuring that the workloads running within them are independent. Among these cells, there is a Linux root cell, which is the first to start and includes a kernel module along with tools for creating, managing, stopping, and destroying other cells [4].

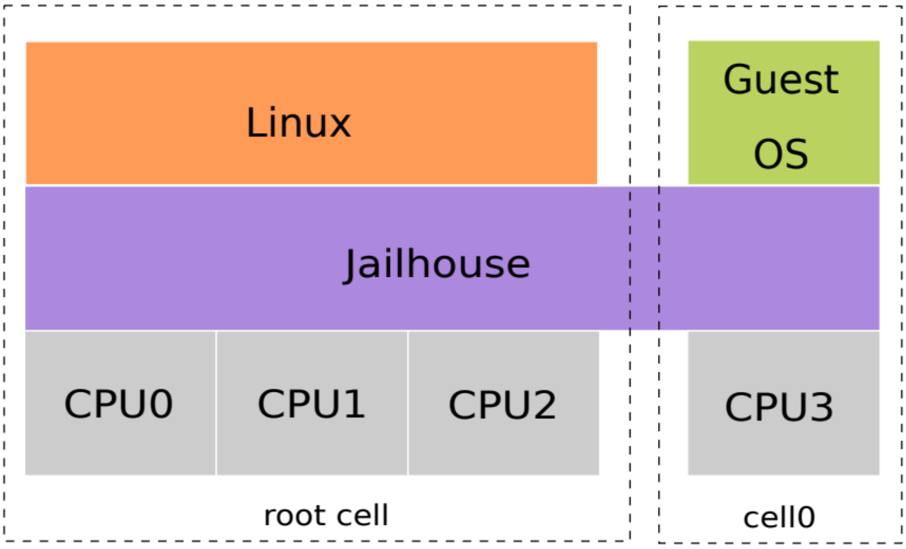


Figure 3: The Architecture of Jailhouse hypervisor [4]

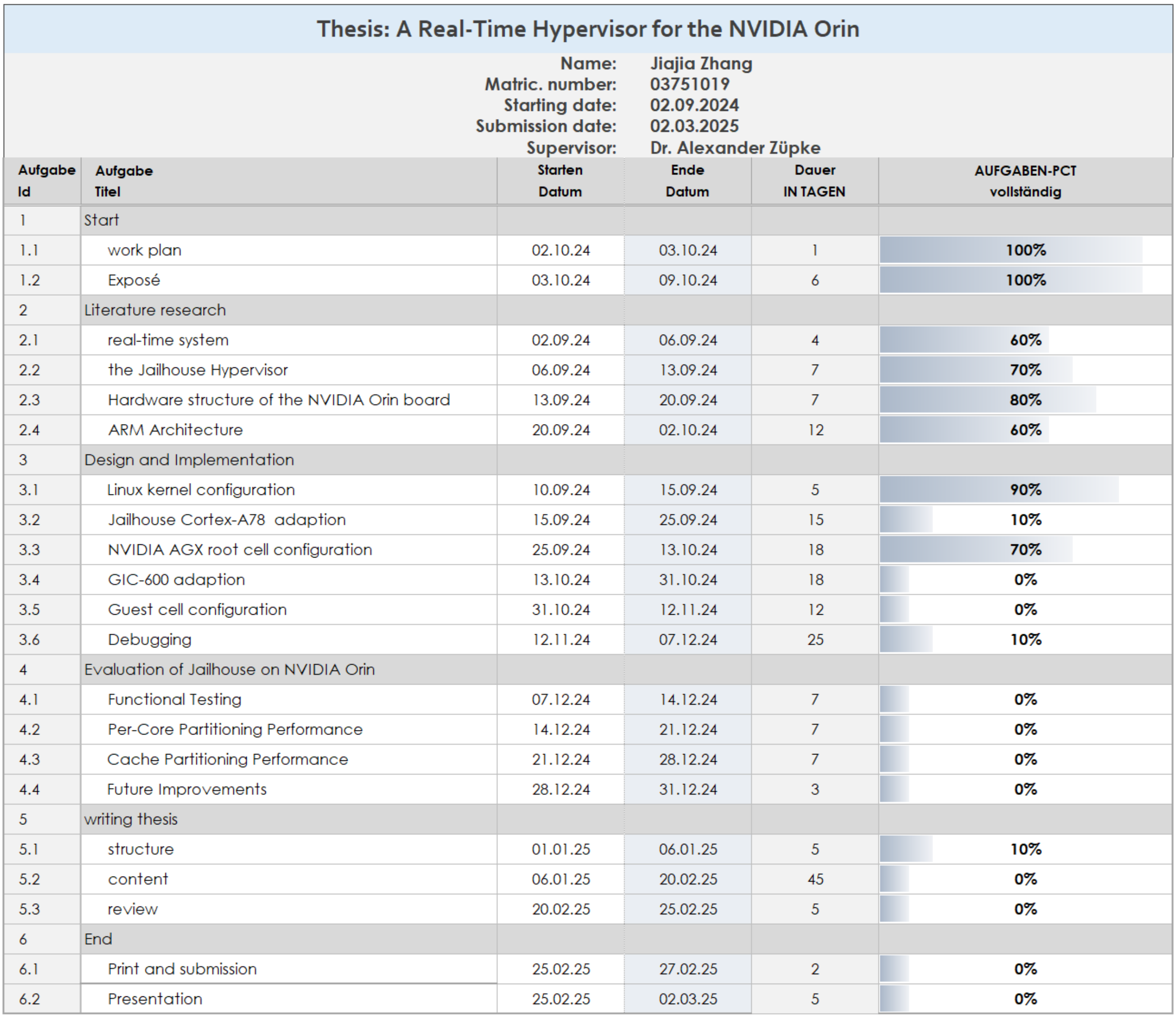
3. Objective of the project

Jailhouse is currently deployed on the Jetson TX1 and TX2 modules of NVIDIA's platform. This thesis will enhance AI and edge computing performance by extending the hypervisor functionality to the Jetson AGX Orin modules. The first objective is to successfully port the Jailhouse hypervisor to the Orin architecture, ensuring efficient operation on this high-performance platform. Subsequently, the effectiveness of per-core partitioning on Orin will be evaluated, focusing on its impact on resource isolation and system performance. Furthermore, the thesis will assess the cache partitioning capabilities of NVIDIA Orin, examining how these features can improve system performance and predictability by optimizing resource management in multi-core environments.

Technical Challenges

The porting to the Nvidia Orin poses several challenges for the Jailhouse hypervisor. Firstly, the Orin has a new CPU architecture (Cortex-A78) that is not yet supported by Jailhouse, and the number of cores on Orin exceeds the number used in previous Jailhouse ports. Also, the implementation of the interrupt controller (GIC-600) is newer than for previous boards, and a first analysis shows that the Orin seems to use more advanced features of the interrupt controller than other boards. Lastly, the firmware architecture is based on an UEFI bootloader rather than the U-Boot bootloader, and we don't know if the Nvidia kernel requires special support from the firmware that needs to be virtualized by the Jailhouse hypervisor.

4. Work plan, necessary resources



Thesis: A Real-Time Hypervisor for the NVIDIA Orin

Name: Jiajia Zhang
Matric. number: 03751019
Starting date: 02.09.2024
Submission date: 02.03.2025
Supervisor: Dr. Alexander Züpke

Aufgabe Id	Aufgabe Titel	Starten Datum	Ende Datum	Dauer IN TAGEN	AUFGABEN-PCT vollständig
1	Start				
1.1	work plan	02.10.24	03.10.24	1	100%
1.2	Exposé	03.10.24	09.10.24	6	100%
2	Literature research				
2.1	real-time system	02.09.24	06.09.24	4	60%
2.2	the Jailhouse Hypervisor	06.09.24	13.09.24	7	70%
2.3	Hardware structure of the NVIDIA Orin board	13.09.24	20.09.24	7	80%
2.4	ARM Architecture	20.09.24	02.10.24	12	60%
3	Design and Implementation				
3.1	Linux kernel configuration	10.09.24	15.09.24	5	90%
3.2	Jailhouse Cortex-A78 adaption	15.09.24	25.09.24	15	10%
3.3	NVIDIA AGX root cell configuration	25.09.24	13.10.24	18	70%
3.4	GIC-600 adaption	13.10.24	31.10.24	18	0%
3.5	Guest cell configuration	31.10.24	12.11.24	12	0%
3.6	Debugging	12.11.24	07.12.24	25	10%
4	Evaluation of Jailhouse on NVIDIA Orin				
4.1	Functional Testing	07.12.24	14.12.24	7	0%
4.2	Per-Core Partitioning Performance	14.12.24	21.12.24	7	0%
4.3	Cache Partitioning Performance	21.12.24	28.12.24	7	0%
4.4	Future Improvements	28.12.24	31.12.24	3	0%
5	writing thesis				
5.1	structure	01.01.25	06.01.25	5	10%
5.2	content	06.01.25	20.02.25	45	0%
5.3	review	20.02.25	25.02.25	5	0%
6	End				
6.1	Print and submission	25.02.25	27.02.25	2	0%
6.2	Presentation	25.02.25	02.03.25	5	0%

5. Literature

[1] Leela S. Karumbunathan, NVIDIA Jetson AGX Orin Series Technical Brief. A Giant Leap Forward for Robotics and Edge AI Applications, https://www.nvidia.com/content/dam/en-zz/Solutions/gtcf21/jetson-orin/nvidia-jetson-agx-orin-technical-brief.pdf, Jul, 2022.

[2] Arm Limited, Learn the architecture - AArch64 virtualization Guide, https://documentation-service.arm.com/static/6627a0f92f51dc4fe7262f81?token=, Sep, 2024.

[3] A. S. Tanenbaum and H. Bos, Moderne Betriebssysteme. Hallbergmoos/Germany. Pearson, 2016.

[4] S. Alonso, L. Muguira, J. I. Garate, C. Cuadrado, and U. Bidarte, “Interrupt Latency Accurate Measurement in Multiprocessing Embedded Systems by Means of a Dedicated Circuit,” Electronics, vol. 13, p. 1626, 2024.

6. Structure

1. Introduction
    - Motivation
    - Thesis Objective
    - Structure of the thesis
2. State of the Art in Research and Technology
    - Real-Time System
    - Overview of the NVIDIA AGX Orin board
        - Jetson-SDK
    - ARM Architecture
        - Exception level on ARM
        - the cortex-a78 cores
    - Hypervisor
        - Difference between Jailhouse and other hypervisors
        - Core isolation
        - Cache partitioning
3. Porting Jailhouse to the NVIDIA AGX Orin
    - Initial Analysis and System Requirements
        - Overview of the hardware and software requirements for porting Jailhouse to NVIDIA Orin.
        - the architecture differences between Orin and other platforms
    - specific patches and configurations
    - Challenges Faced During Porting
    - Remaining Gaps and Future Improvements
        - Areas where the current port is incomplete or suboptimal.
        - Discussion of potential future improvements and ongoing development needs.
4. Evaluation of Jailhouse on NVIDIA Orin
    - Functional Testing
        - Description of test cases to ensure Jailhouse is functioning correctly on Orin.
        - Does the port meet the functional requirements?
    - Per-Core Partitioning Performance
        - How well does Jailhouse partition CPU cores?
    - Cache Partitioning Performance
        - Effects of running real-time applications with cache isolation and shared cache.
5. Disscution
6. Summary and Outlook