
AUTOMOTIVE
GRADE LINUX
ALL MEMBER MEETING

AUTOMOTIVE GRADE LINUX
ALL MEMBER MEETING

MAY 13 – 14, 2026
TOKYO, JAPAN

Challenge of VirtIO for Automotive Use

- Latency Measurement and Solution -

Gen Shimada
Linaro

AGENDA

- Introduction to Linaro
- VirtIO Today
- Challenge of VirtIO

Linaro Introduction

AUTOMOTIVE GRADE LINUX
ALL MEMBER MEETING

linaro™ is the **software engine** of the **Arm Ecosystem**

Linaro empowers rapid product deployment within the dynamic arm Ecosystem.

Linaro has enabled trust, quality and collaboration since 2010

- Our cutting-edge solutions and services facilitate the swift **development, testing, and delivery of Arm-based innovations,** enabling businesses to stay ahead in today's competitive technology landscape.
- **Linaro** fosters an environment of collaboration, standardization and optimization among businesses and **open source ecosystems to accelerate the deployment of Arm-based products and technologies** along with representing a pivotal role in open source discovery and adoption.
- **Automotive, Testing, Linux Kernel, Security, Cloud & Edge Computing, IoT & Embedded, AI, CI/CD, Toolchain, Virtualization**

Linaro Introduction

We connect businesses & Open Source communities

To be competitive in the arm ecosystem you need influence.

Linaro has that influence.

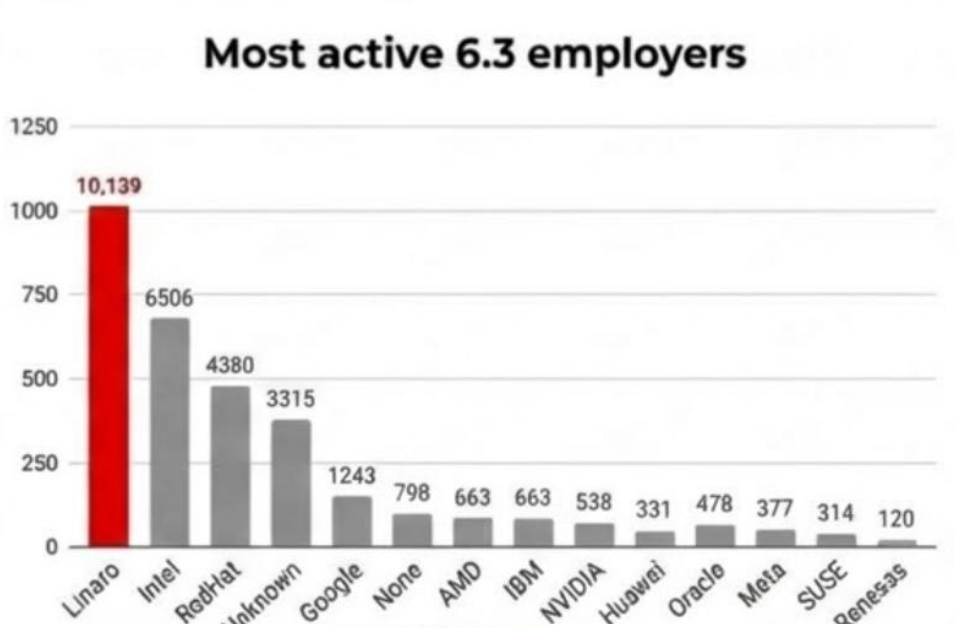
---

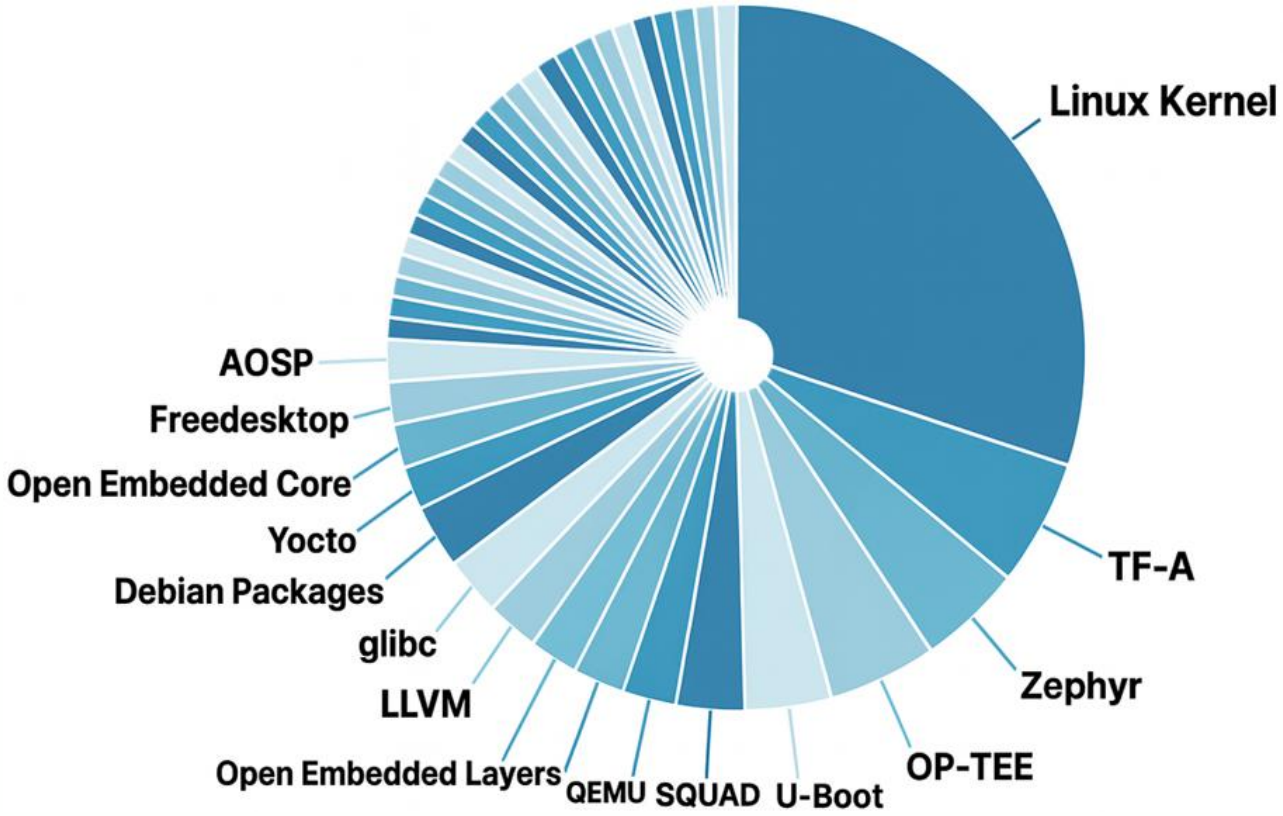
68 **Open Source Software Maintainers work at Linaro**

Many of our engineers are maintainers on open source projects in the arm software ecosystem. This makes us uniquely positioned to drive the conversations needed between the open source community and industry leaders to identify opportunities, highlight problems and propose solutions.

Google arm Qualcomm Innovation Center AMD XILINX
HISILICON Microsoft CIX Lenovo
Red Hat SAMSUNG FUJITSU RENESAS

arm Solutions at Lightspeed





From Upstream to Production

Linaro is your production-grade software and CRA lifecycle partner

Research	Secure Development	Test	Comply	Deploy	Maintain
Feasibility Study Architectural Design Gap Analysis Prototyping Cybersecurity assessment and CRA compliance analysis Training	Compliance documents including risk assessments SoC, BSP, HW Bring-Up Customer Applications Full Operating System Image	Code Hosting CI/CD setup Automated Device Testing Functional, Regression, Performance	Certification Artifacts including CRA Documents of conformity CVEs report IP compliance and SBOMs 3rd Parties Notice File	Software updates infrastructure Software signature Monolithic to granular updates (separation CVEs updates) Fault tolerant	Single Integration Partner (including closed source components) **LTS** Maintenance and Support 10 Years + (Linux & Android)

CRA: EU Cyber Resilience Act

Linaro supports Customers for CRA Certification

Legal & Technical Consultation

Support with understanding the regulation, defining scope, product categorisation, running a GAP analysis and developing a compliance program

Cybersecurity Risk Assessment

Based on the products relevant factors, we can support with putting together a cybersecurity risk assessment, defining risk mitigations and security requirements

Securing Your Product

Securing your product through our deep knowledge in the secure software development lifecycle

Keeping your product secure though continuous updates to open source components

Penetration Testing

Product security validation through our penetration testing services

Software Supply Chain Management

Software supply chain management services enabling continuous vulnerability monitoring and SBOM generation

VirtIO today

VirtIO Current Status

VirtIO today

Role of OASIS and the current status

Role of AGL

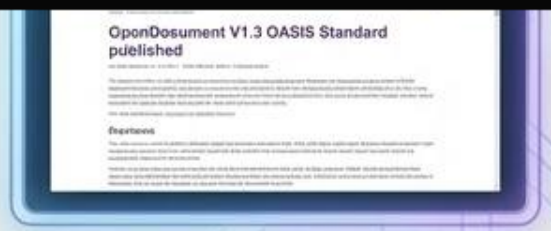
OASIS: The Role

OASIS (Organization for the Advancement of Structured Information Standards)

About: One of the most respected, non-profit standards bodies in the world, OASIS Open offers projects—including open source projects—a path to standardization and de jure approval for reference in international policy and procurement.

OASIS Open projects—a path to standardization and de jure approval for reference in international policy and procurement.

Each device is upstream to Linux



https://www.oasis-open.org/2021/06/16/opendocument-v1-3-oasis-standard-published/

OASIS Virtual I/O Device TC

Device Spec Definition:
Block, Net, GPIO, I2C, SCMI, Sound, CAN.

Enhancing VirtIO devices

Linux kernel Upstream:

Each device is upstream to Linux

OASIS: The Status

V1.0 (March, 2016) → V1.1 (April, 2019) → V1.2 (July, 2022)

- V1.0 (March, 2016): VirtIO-blk, VirtIO-net, VirtIO-console
- V1.1 (April, 2019): VirtIO-entropy, VirtIO-input, VirtIO-vsock, VirtIO-crypto, VirtIO-gpu(Basic 2D)
- V1.2 (July, 2022): VirtIO-sound, VirtIO-gpu(3D: Virgl)

Enhancement toward Automotive function

Merging to Linux kernel - on going

The Evolution of VirtIO Standard

Summary: List of VirtIO Devices

OASIS ver	release	VirtIO devices
v1.0	2016	VirtIO-net, VirtIO-console, VirtIO-blk, VirtIO-rng, VirtIO-baloon, VirtIO-scsi
v1.1	2019	VirtIO-gpu, VirtIO-input, VirtIO-vsock, VirtIO-crypto
v1.2	2022	VirtIO-gpio, VirtIO-sound, VirtIO-scmi, VirtIO-fs, VirtIO-rpmb, VirtIO-mem, VirtIO-i2c, VirtIO-pmem
v1.3	2026	VirtIO-iommu, VirtIO-bt, VirtIO-video, VirtIO-camera
v1.4	2026	VirtIO-media, VirtIO-can, VirtIO-rtc, VirtIO-wifi

Summary: OASIS and AGL

OASIS: International Standards Organization

Defines and Approves International Standards for the VirtIO Protocol

Leads the open standardization process by publishing technical specifications.

VirtIO Specification Evolving for Automotive Grade

Adds specifications for control devices important to vehicles, such as GPIO, I2C, and Sound.

Continuously Driving Standardization from v1.4 Onwards

Aims to standardize new devices such as VirtIO-media, wifi, and rtc.

AGL: Driver of Automotive Implementation and Feedback

Community Implementing and Integrating Standardized Specs into Automotive Systems

An automotive open-source platform promoted by the Linux Foundation.

Functions as the "AGL VirtIO Proving Ground"

A proof-of-concept field to implement and verify pre-standardized functions early and identify issues.

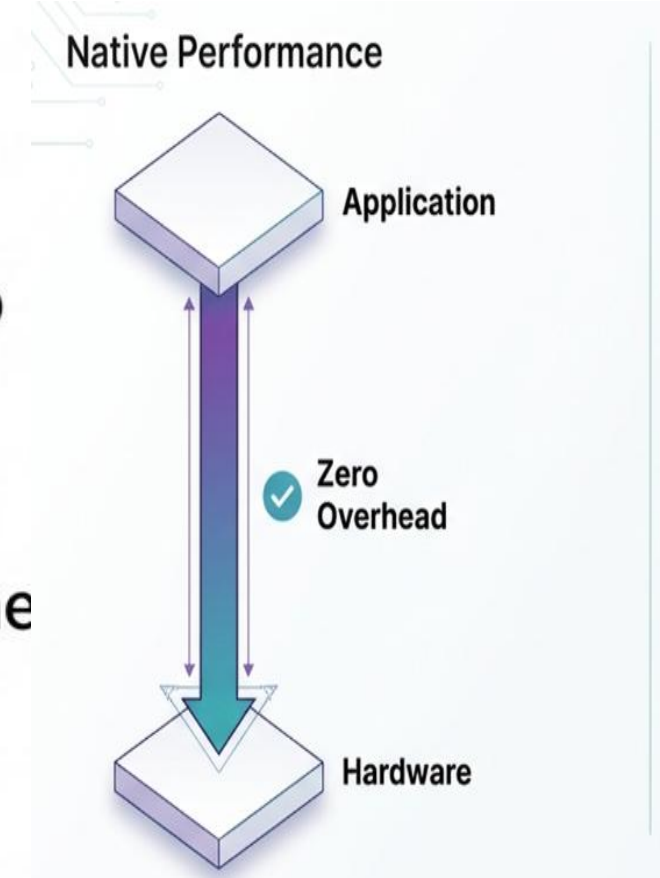
Feedback of Implementation Results to OASIS

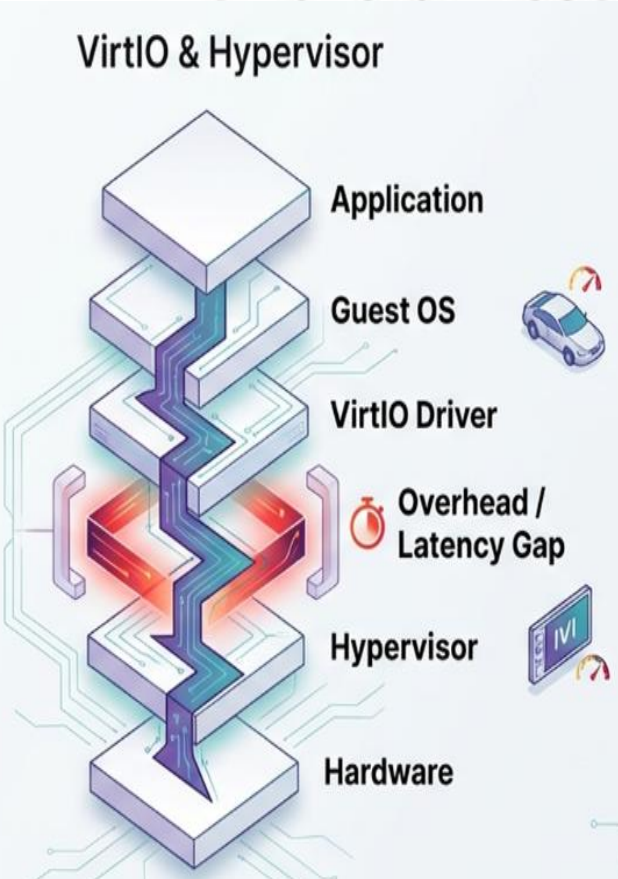
Feeds back verification results from real environments to promote more practical standardization.

Challenges

Challenge: Overhead

- VirtIO, in combination with virtualization technology, incurs overhead compared to native performance.
- This must be considered not only for ADAS, where real-time is required, but also for IVI.



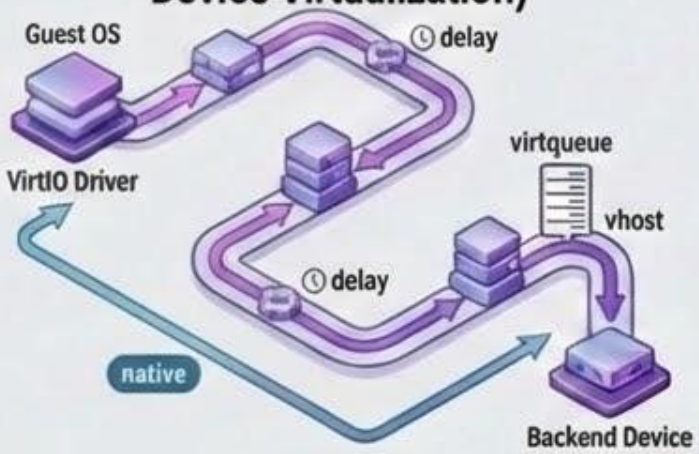


*Since VirtIO implementation relies on the combination of the virtualization platform (Hypervisor) and the VirtIO implementation itself, overhead and latency occur in virtualized environments using VirtIO.

Mechanism of VirtIO Latency

The main mechanisms causing latency in VirtIO/virtualized environments are summarized into the following three aspects.

1. I/O Path Delay (Prolonged Communication Path due to Device Virtualization)

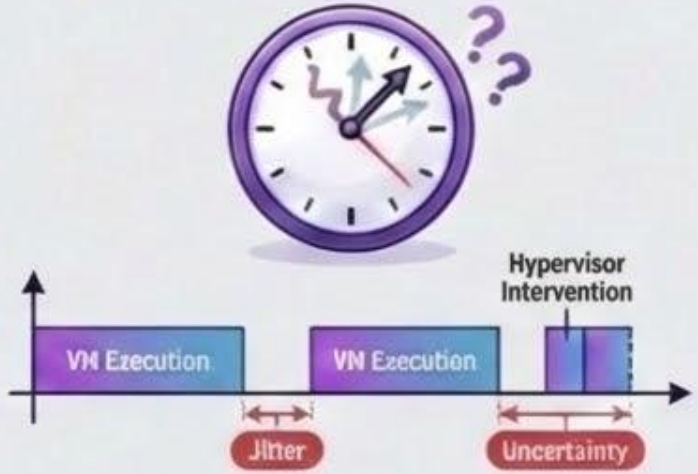


- **Difference from normal flow**: In a native environment, the Guest OS accesses the device directly, but in a virtualized environment, processing is passed from the Guest OS's VirtIO driver through the hypervisor to the host-side backend device (vhost).
- **Overhead generation**: The processing via this hypervisor and the operation of the queue (virtqueue) for passing data between the guest and host take time, and the longer communication path increases latency.

2. Resource Competition

- **Shared Resources:** Multiple VMs compete for hardware resources such as CPU and memory. When a high-load task is executed in one VM, the processing of other VMs (especially those requiring real-time performance) may be delayed (Noisy Neighbor problem).
- **Interference generation**: Non-critical tasks can cause interference that hinders the execution of safely-critical tasks.

3. Scheduling Uncertainty (Lack of Determinism and Jitter)



- **Hypervisor Intervention**: The hypervisor pauses and resumes VM execution without the Guest OS's knowledge, making it difficult to guarantee "when" important tasks will be executed.
- **Jitter generation**: This causes variation (jitter) in task execution timing, making it difficult to guarantee the real-time performance (determinism) required in automotive systems, etc., that "Important tasks are completed within the deadline".

Where the Latency occurs?

In a Virtio environment, latency tends to increase in the following three areas compared to a Native environment. These are always the subject of measurement as a baseline for evaluation.

Command Submission

Overview:

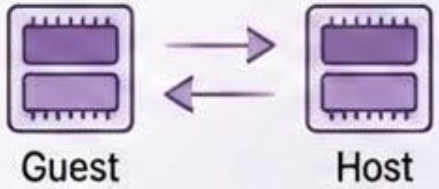
Bypass time from when the Guest OS issues a Vulkan command until it reaches the Host OS's physical GPU (context switch cost).

Example Evaluation Tool/Method:

Measurement of `vkQueueSubmit`, RenderDoc (Draw Call Duration)

Memory Mapping

Overview:



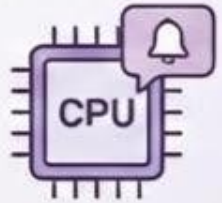
Synchronization and copying of video memory between Host and Guest (Shared Memory efficiency).

Example Evaluation Tool/Method:

Measurement of virtio-gpu resource sharing time

Interrupts

Overview:



Overhead when notifying the Guest OS that the GPU process is complete.

Example Evaluation Tool/Method:

cyclictest, LatencyMon

Furthermore, MangoHud (real-time display and recording of Frame Time) is effective for

Latency Measurement Items – Use Case Example

AUTOMOTIVE GRADE LINUX
ALL MEMBER MEETING

IVI

Overview: In the automotive field, 'safety' and 'certainty of display' are emphasized, making the following items essential as a basis for evaluation.

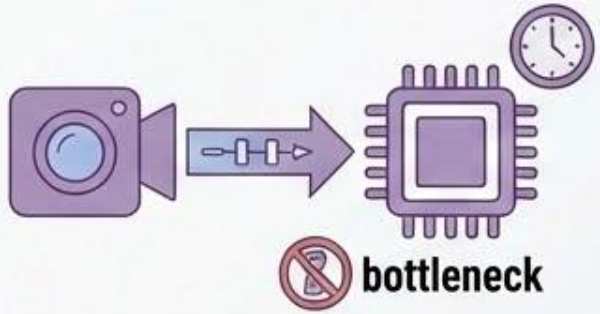
Measurement Item	Importance/Validity in Automotive	Specific Metric / Tool
Frame Time Variance (Jitter)	Prevents meter needle stuttering. Measures the frequency of outliers, such as 99th Percentile Latency, rather than the average value.	**MangoHud** 
Guarantee of Worst-Case Execution Time (WCET)	Emphasizes certainty, such as "maximum delay is always within 100ms".	**Jitter analysis, Maximum delay measurement during load concentration**
Zero-copy Efficiency (Zero-copy / DMABUF)	Measures the efficiency of physical memory sharing between host and guest to avoid CPU load and delay caused by memory copying.	**virtio-gpu resource sharing time**
Startup / Flip Latency	Total delay from power ON to display (Time-to-first-frame), such as for rearview camera footage.	**Motion-to-Photon** (External camera measurement)
Mixed Criticality	Confirms whether cluster drawing latency increases when the load on the IVI side is high.	**Load interference test using Stress-ng**

Latency Measurement Items – Use Case Example

ADAS

Overview: In ADAS, the GPU is used not only for "rendering" but also as "Compute (AI inference/image processing)", making these measurement items more "critical metrics related to life and death" than IVI.

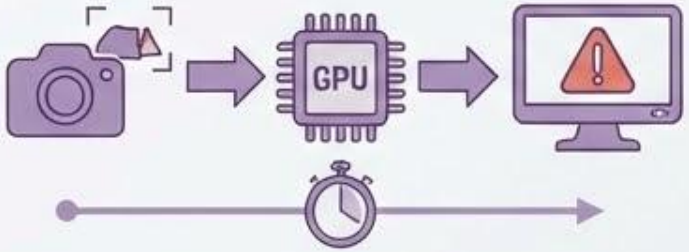
Sensor Data Pipeline Delay



Evaluate whether the overhead of data transfer due to virtualization leads to delays in pedestrian detection.

Measurement Method: vkmark Compute scene, Vulkan Compute inference time

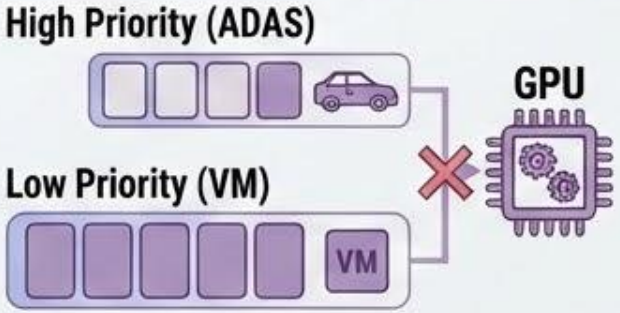
End-to-End Latency



Total time from when an obstacle is captured by the camera, detected by the GPU, until a warning appears on the screen.

Measurement Method: Video comparison using an external high-speed camera

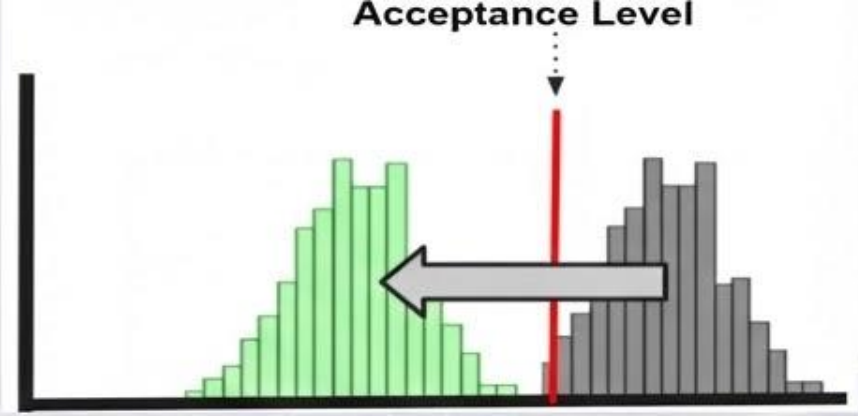
Priority Inversion



Confirm whether a low-priority VM occupies the GPU and hinders high-priority ADAS processing (GPU Scheduler).

Measurement Method: Confirm behavior of the host-side GPU scheduler

Solution



Measure latency of each item

Understand acceptable latency range for each item

Optimize to reduce latency of necessary items

- Linaro performs latency measurement.
- We also conduct optimization work to reduce latency and bring it within the acceptable range.

AUTOMOTIVE GRADE LINUX
ALL MEMBER MEETING

Gen.Shimada@Linaro.Org

Thank you

Challenge of VirtIO for Automotive Use