Hypervisor Project Readout

October 10, 2018 | GENIVI Technical Summit, Bangalore

Gunnar Andersson

Development Lead, GENIVI Alliance

Hypervisor Project - Project Goals

Support the industry in difficult transition from individual “boxes” toward system consolidation

Fix the knowledge gap

Simplify, clarify. Don’t create a new mountain of industry challenge (We have more important things to do!)

When and why, and when not to use virtualization

Requirements and Architectures

Standard APIs → Platform standardization!

API Standards

→ The API between virtual machine
(the kernel/system running in that machine) and the hypervisor

→ New APIs between virtual machines?

Led to the topic with the most interest and potential influence:

The definition of *The* Automotive Virtual Platform.

Project drivers

Why Virtualization?

Virtual Platform definition

Virtual Device standardization, a.k.a. Automotive Virtual Platform definition

Common I/O devices for hypervisor guests with standardized features and interface, such that device drivers (and thereby systems) are more portable.

- Device driver definition (for paravirtualization) don't need to be maintained uniquely
- Ability to move hypervisor guests between different hypervisor environments
- Test systems in isolation, and away from final environment
- Some potential for shared device driver implementation across hypervisors

Extending this: Standardizing a contract/standard between guest and hypervisor. Compare the OCI initiatives for containers. Container runtimes → can we have standardized "hypervisor runtime environment" that allows a standards compliant virtual (guest) machine to run.

Hypervisors can fulfil the specification (with local optimizations / advantages)
Similarly, this specification is what guests can be engineered to.
Compare: Linux Device Tree – ability to discover and configure devices.

Virtual Platform definition – approach

Iteratively consider:

- Automotive Requirements
- Existing definitions (e.g. VIRTIO as a starting point)
- Do Gap-analysis

  → new and complementary requirements

Concrete activity right now, step by step:

- Discuss and write down the automotive requirements
- Read VIRTIO chapter. Present to group + new ideas
- Decide if VIRTIO is appropriate and complete for requirements
- Write down what the industry needs to do to close the gap

Write findings. Let it evolve into *The Automotive Virtual Platform*
*… possibly using "Meta/D*elta specification" (see Wiki)

Virtual Platform definition – goals

PORTABLE

The virtual platform definition would allow the development of virtual machine guests that like appliances in the enterprise world, could be moved among different hypervisor systems without (minimal) modification.

SPECIFIED

A clear and detailed specification is required to achieve true portability, and to give real support to vendors. Hand-waving about some virtual architecture is not our goal. The specification shall enable efficient reuse and collaborative progress among companies in this industry.

REUSE

Specifications like these are best developed as open specifications and just like open-source code, it should stand on the shoulders of previous work.

Virtual Platform – device categories

Block Storage
Graphics/GPU
Network
Time-sensitive networking
Audio
Filesystem share (9pfs in virtio)
Bluetooth
CAN
USB
Console
Crypto, Hardware-security-modules
Special CPU cores
Special CPU modes (e.g. TrustZone)
Serial devices
Sensors
Input devices
Memory Balooning
Virtual IOMMU
vsock (VM to Hypervisor communication)

Requirements and Architecture

Look at any “typical” use-case

Compare “traditional approach” – study electrical-architecture & network

Apply virtualization – new architecture variations

What are the new constraints, challenges, …

Which virtualization requirements can we derive from that?
–> Feedback loop into API / Virtual Platform topic.

Topic brainstorm

Just a peak into thebacklog
(original list, some we *don't* really spend time on at this moment)

API for security, e.g. access control / MAC
  --> NOTE that a time block on security on ARM in Working Session Tomorrow
VM management tool
Instrumentation & tools – better understanding / profiling / performance issues...
Safety compliance: ISO26262
  --> NOTE "Functional safety challenge" in Graphics Working Session Tomorrow
Security compliance: Common Criteria, EAL
System design to optimize Boot Time,
Boot requirements, e.g. secure boot, integrity check,
Terms / Nomenclature

Interesting challenges

Scheduling jobs on complex SoCs, especially if real-time – how the heck...?

Special hardware – GPU (of course).
But also audio? DSP/Signal processing? “Tensor processing”, ….?

“Sensors”

(Cyber-) Security – do hypervisors really increase “security”? For which threats?
Yes/No? We need to know how exactly.

Study, classify, agree on the decision drivers, so that they can be applied quickly and correctly.

Working Session tomorrow

GPU Sharing & related graphics-in-virtualization
- Exchange between GSHA and HV project participants

Presentation by ARM on TrustZone developments and consequences for
This will be a foundation for the main Security topic discussion

The role of specialized hardware vs generalized “virtual” platforms

+ The other interesting topics (as time permits)

Thank you!

Visit GENIVI at http://www.genivi.org or http://projects.genivi.org

Contact us: help@genivi.org

Case Study: Xvisor

Welcome, Anup Patel