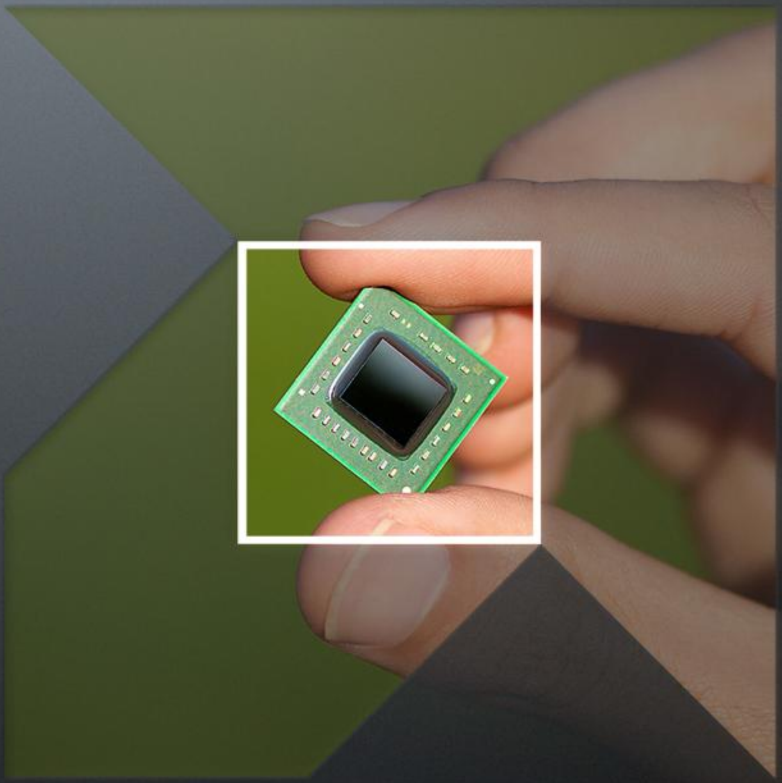


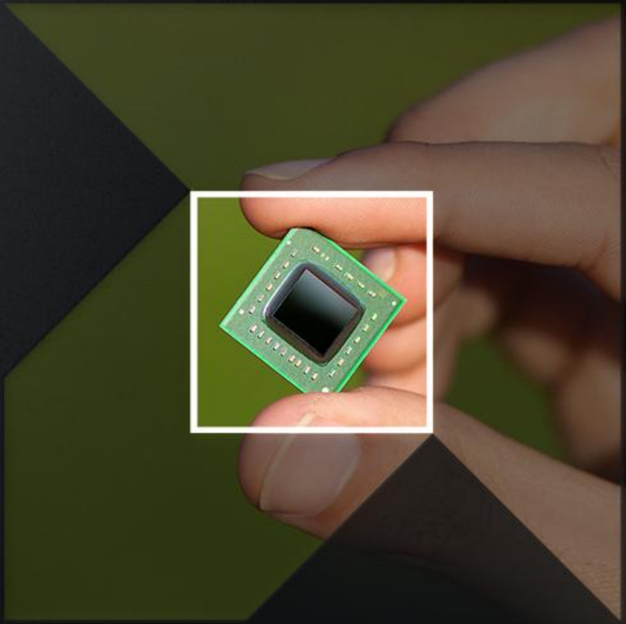
AMD IOMMU VERSION 2

How KVM will use it

Jörg Rödel

August 16th, 2011

AMD IOMMU VERSION 2
WHAT’S NEW?



NEW FEATURES - OVERVIEW

- Two-level page tables
  - Similar to nested paging on the CPU
  - Second-level page-table format equal to AMD64 long mode
  - Multiple second-level page tables per device

- Demand paging support
  - PPF according to the PCI ATS specification
  - Device can notify about failed ATS requests
  - IOMMU can send retry request to device

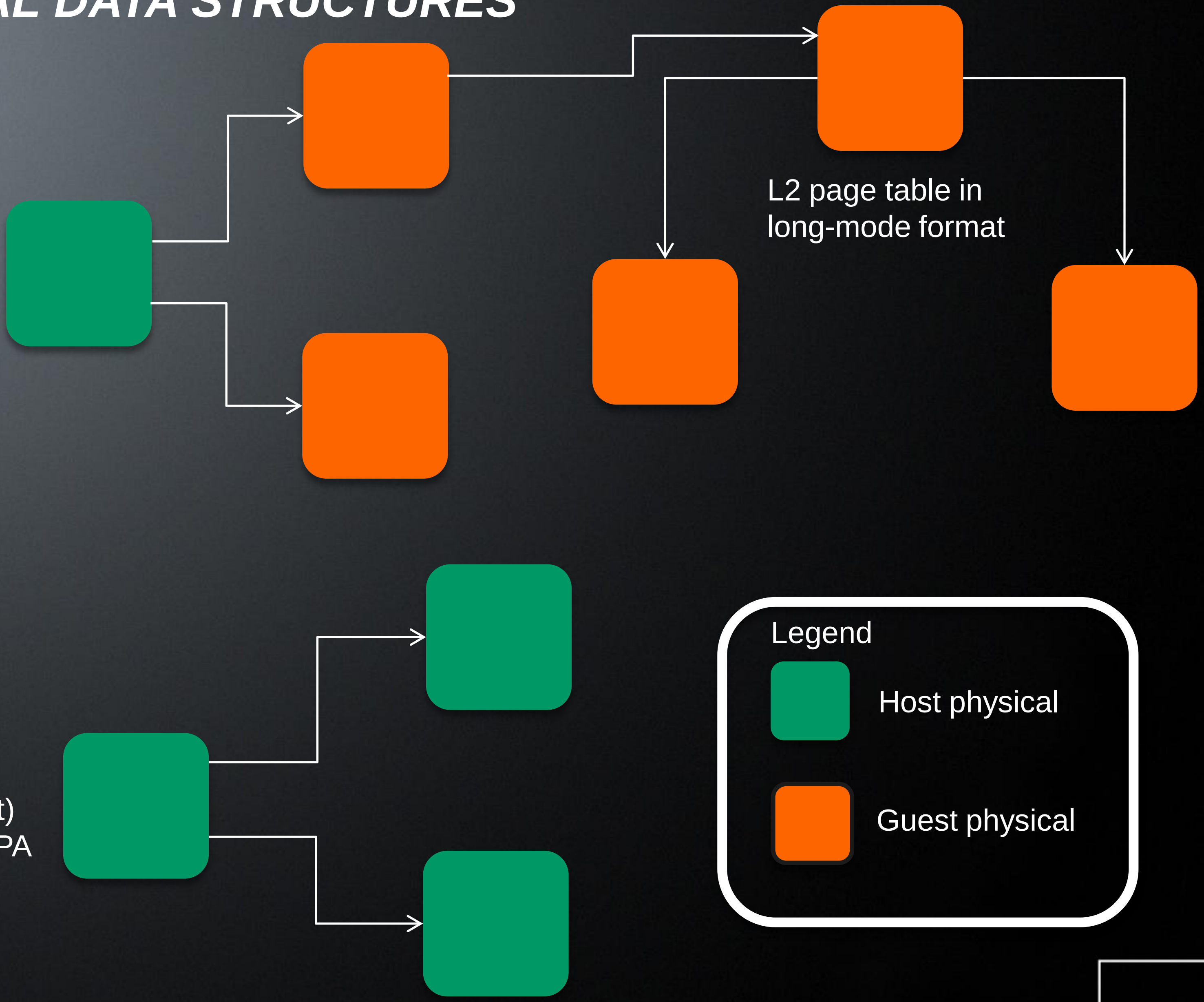
- Support for performance counters

TWO-LEVEL PAGE TABLES

- Second-level page table has AMD64 long-mode format
  - IOMMU atomically updates accessed / dirty bits
  - Allows sharing of page tables with processes
  - Zero-copy DMA

- Device can choose to support multiple contexts
  - Each context has its own second-level page table
  - Unique identifier: PASID
  - Up to 2^{20} PASIDs supported

ADDITIONAL DATA STRUCTURES



Guest CR3 table translates PASIDs into L2 CR3s

L2 page table in long-mode format

L1 page table (in IOMMUv1 format) translates GPA in HPA

Legend

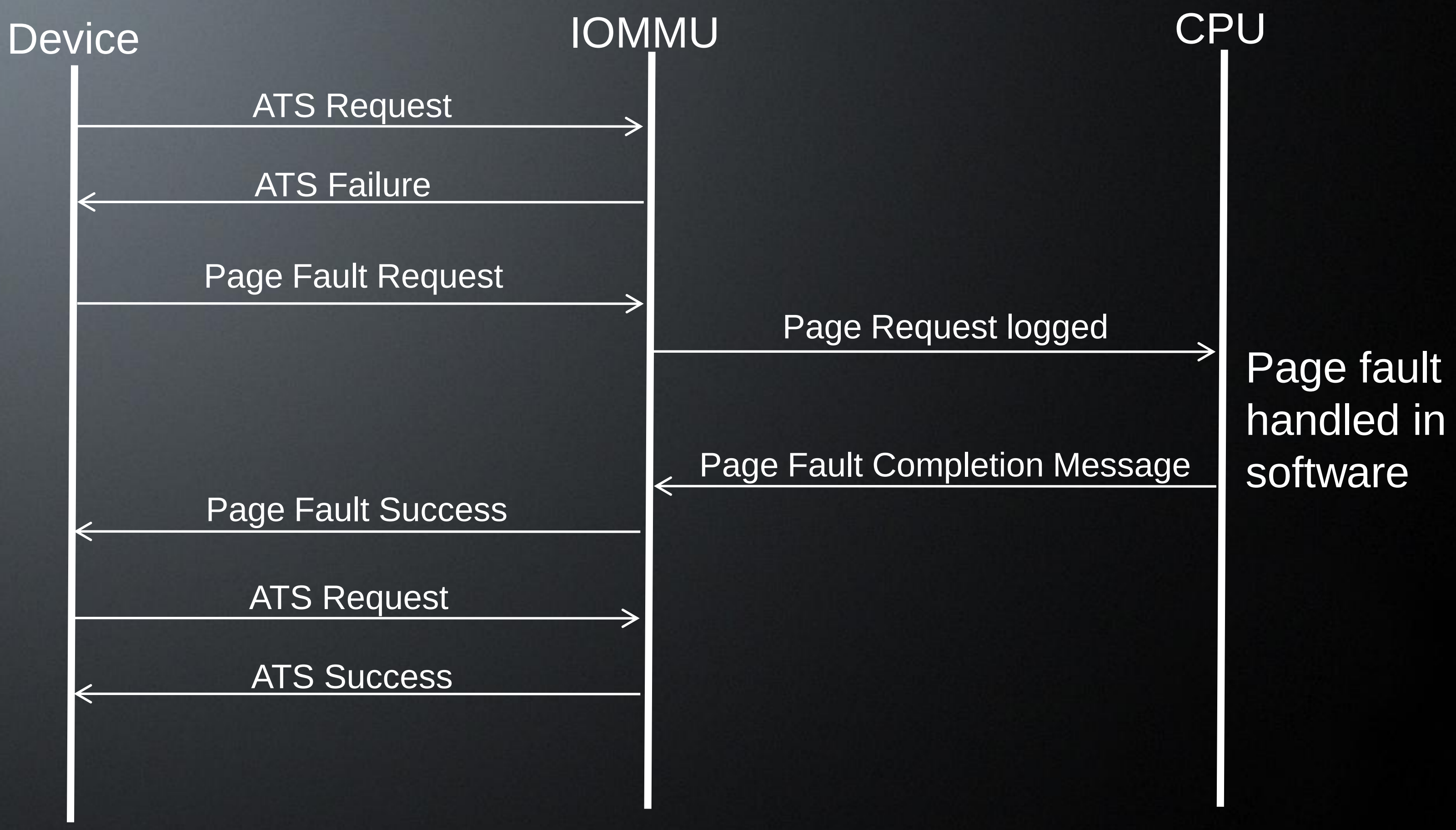
🟩 Host physical

🟧 Guest physical

DEMAND PAGING SUPPORT

- Devices can signal a page fault condition
  - Today, IOMMU page faults are not recoverable
  - Devices need to support the PPR capability
- Depends on ATS
  - Device first sends ATS request
  - On ATS failure, device can send a page fault request
  - Page fault request can be tagged with a PASID
  - When fault is completed, ATS request is sent again
- Page fault handling is done in the IOMMU driver

PERIPHERAL PAGE FAULTS



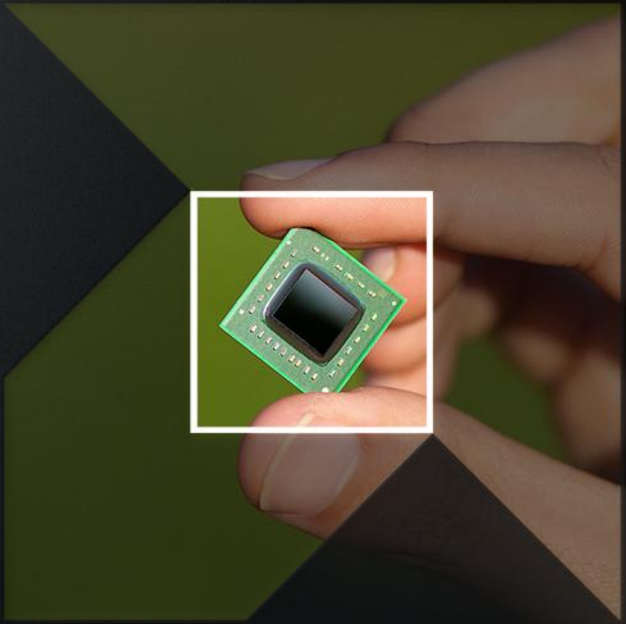
SUMMARY

- IO page faults are no longer fatal errors
- Additional support needed from devices
  - Support needed for ATS and PPR capabilities
  - PASID capability is optional

- Devices without these features are handled like today

- New data structures introduced
  - Most of them are guest physical
  - Easy to virtualize
  - Long-mode Ffrmat of L2 page tables allows sharing them with processes

AMD IOMMU VERSION 2 USE IN KVM



KVM SUPPORT – FIRST STEP

- Devices may only implement ATS and PPR

- Get rid of guest memory pinning when all assigned devices support PPR
  - DMA may be a bit slower on memory overcommit
  - But: removes a major disadvantage of direct device assignment

- Requires some changes in the KVM device assignement code

This is the easy part.

KVM SUPPORT – FURTHER STEPS

- Target devices supporting PASIDs
- Using the PASID feature requires an IOMMUv2
  - Some functionality of the device may only be available with PASID
  - This functionality gets lost when device is assigned to a guest

- For assigning those devices, an IOMMUv2 is needed in the guest
  - Supported reasonably well by hardware design

- Some data structures need shadowing
  - Command log, Event and PPR buffers
  - L1 page tables (probably not present most of the time)

KVM SUPPORT – INTERFACES

- Starting point is to get the current AMD IOMMU emulation patchset upstream
- VFIO needs to be extended to support IOMMU emulation for assigned devices
- The emulation of IOMMUv2 features is planned on this
  - VFIO interface needs to be extended for that
  - The exact design is not clear yet – discussion needed

Looks like a long way to go.

- **DISCLAIMER**
- The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.
- The products, features, specifications and information contained herein are under development and/or in their definition stage. This document is preliminary and tentative in nature and may contain technical inaccuracies, omissions and errors. It is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.
- **AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.**
- AMD ASSUMES NO LIABILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS PRESENTATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES

Trademark Attribution

AMD, the AMD Arrow logo and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. Other names used in this presentation are for identification purposes only and may be trademarks of their respective owners.

Questions?