INTEL® FPGA LINUX* DRIVER SOLUTION AND UPSTREAMING STATUS

Figo Zhang, Hao Wu

AGENDA

Background
- Intel® FPGAs and the Modern Datacenter
- Platform Options and the Acceleration Stack
- Open Programmable Acceleration Engine

Hardware Overview
- FPGA Interface Manager (FIM)

Intel FPGA Linux Driver solution
- Driver Architecture
- Partial Reconfiguration
- Virtualization
- Example: Simple DMA Operation
- Upstream Status

Background

DATA MOVEMENT AND PROCESSING EXPLOSION

>20B
Connected Devices
by 2020[‡]

5G Wireless

Hyper-connected World

Big Data Processing and Analytics

High-Performance Computing Demands

Markets

- Government
- Enterprise
- Cloud
- Communications

Infrastructure

- Network
- Storage
- Compute

Workloads

- Security
- Big Data Processing and Analytics
- Video processing and transcode
- Artificial Intelligence & Machine Learning
- Packet processing

[‡] Source: "Gartner Says 8.4 Billion Connected 'Things' Will Be in Use in 2017, Up 31 Percent From 2016", 2/7/2017, http://www.gartner.com/newsroom/id/3598917 (Table 1 - IoT Units Installed Base by Category, 2020 column – Grand Total, including consumer+business units)

ACCELERATION ASSIST TO PROPEL DATA INSIGHT & OPERATIONAL EFFICIENCY

Workload Optimization:
Ensure Intel® Xeon® CPU cores serve highest value processing

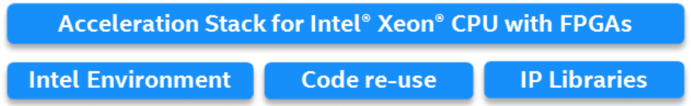
Workload 1
Workload 2
Workload N

Efficient Performance:
Improve performance/watt

Real-Time:
High bandwidth connectivity and low-latency parallel processing

Milliseconds

Developer Advantage:
Code re-use across Intel FPGA data center products

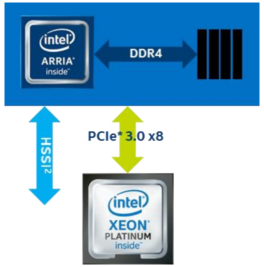


The Intel® Xeon® processor with FPGA acceleration can reduce TCO and solve new problems

INTEL® FPGA DATA CENTER FORM FACTORS OPTIONS

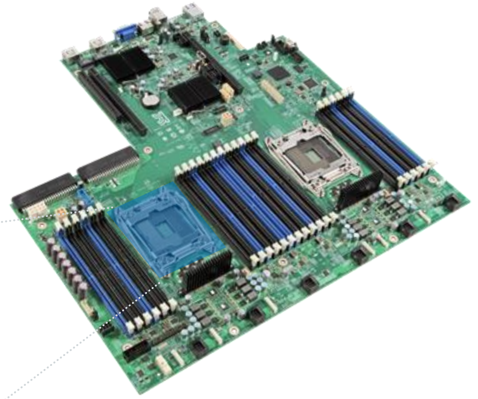
Enabled By The Acceleration Stack for Intel® Xeon® CPU with FPGAs

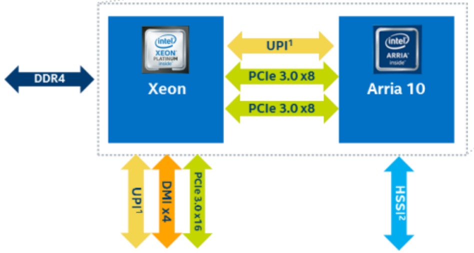
PCIe Acceleration Cards



- System flexibility with Intel Xeon CPU SKU options
- Dedicated local memory
- Can be slotted into 1U servers

Server Platform Option with In-Package FPGA



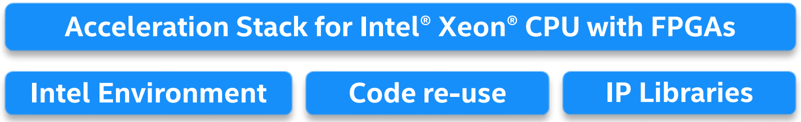


- Coherent interface benefits software developers
- Superior performance for bandwidth & latency sensitive applications

Choose the Intel FPGA form factor matched to your application needs

[1]UPI = Intel® Ultra Path Interconnect
[2]HSSI = High Speed Serial Interface

THE INTEL® APPLICATION DEVELOPER ADVANTAGE



Acceleration Stack for Intel® Xeon® CPU with FPGAs – *Enhanced Performance, Simplified*

- Saves developer time to focus on unique value-add of their solution
- Enables unprecedented code re-use across multiple Intel FPGA form-factor products
- World's first common developer interface for Intel FPGA data center products
- Optimized and simplified hardware and software APIs provided by Intel
- Enables easier development and deployment of Intel FPGAs for workload optimization

The stable and optimized foundation for building your Intel FPGA-accelerated solution

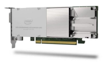
ACCELERATION STACK FOR INTEL® XEON® CPU WITH FPGAS

Enhanced Performance, Simplified

Dynamically Allocate Intel® FPGAs for Workload Optimization	**Rack-Level Solutions**	intel Rack Scale Design, openstack
Simplified Application Development	**User Applications**	Deep Learning, Networking, Genomics, etc.
Leverage Common Frameworks	**Industry Standard SW Frameworks**	Caffe, theano, gatk
Fast-Track Your Performance	**Acceleration Libraries**	LZ4, Snappy, etc. zlib, DPDK
Workload Optimization with Less Effort	**Intel Developer Tools** *(Intel Parallel Studio XE, Intel FPGA SDK for OpenCL™, Intel Quartus® Prime)*	Parallel Studio XE, OpenCL, Quartus Prime Design Suite
Common Developer Interface for Intel FPGA Data Center Products	**Acceleration Environment** *(Intel Acceleration Engine with OPAE Technology, FPGA Interface Manager (FIM))*	

Intel Hardware

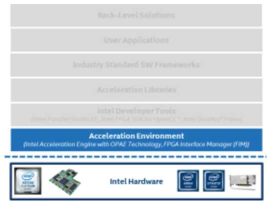
intel XEON PLATINUM inside — intel ARRIA inside — intel STRATIX inside



Intel® delivers a system-optimized solution stack for your data center workloads

OPEN PROGRAMMABLE ACCELERATION ENGINE (OPAE) TECHNOLOGY

Simplified FPGA Programming Layer for Application Developers


Rack-Level Solutions
User Applications
Industry Standard SW Frameworks
Acceleration Libraries
Intel Developer Tools
(Intel Parallel Studio XE, Intel FPGA SDK for OpenCL™, Intel Quartus* Prime)
Acceleration Environment
(Intel Acceleration Engine with OPAE Technology, FPGA Interface Manager (FIM))
Intel Hardware

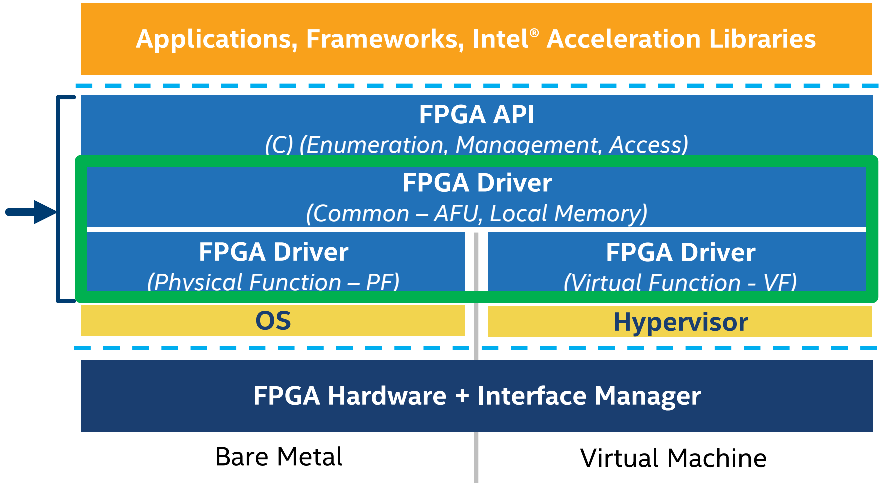
Consistent cross-platform API

Minimal software overhead and latency

Supports virtual machines and bare metal platforms

Open source code licensing and developer community
- Intel FPGA drivers being upstreaming to Linux kernel

SDK includes:
- Guides, utilities and sample code
- **AFU Simulation Environment (ASE)**:**
  - Develop and debug Accelerator Functions faster


Applications, Frameworks, Intel® Acceleration Libraries
FPGA API
(C) (Enumeration, Management, Access)
FPGA Driver
(Common – AFU, Local Memory)
FPGA Driver
(Physical Function – PF)
FPGA Driver
(Virtual Function - VF)
OS
Hypervisor
FPGA Hardware + Interface Manager
Bare Metal
Virtual Machine

Start developing for Intel FPGAs with OPAE today: http://01.org/OPAE

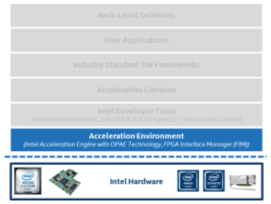
***ASE requires Acceleration Functions written in RTL and a properly installed RTL simulator: Synopsys* VCS-MX, Mentor Graphics* ModelSim-SE*/QuestaSim*

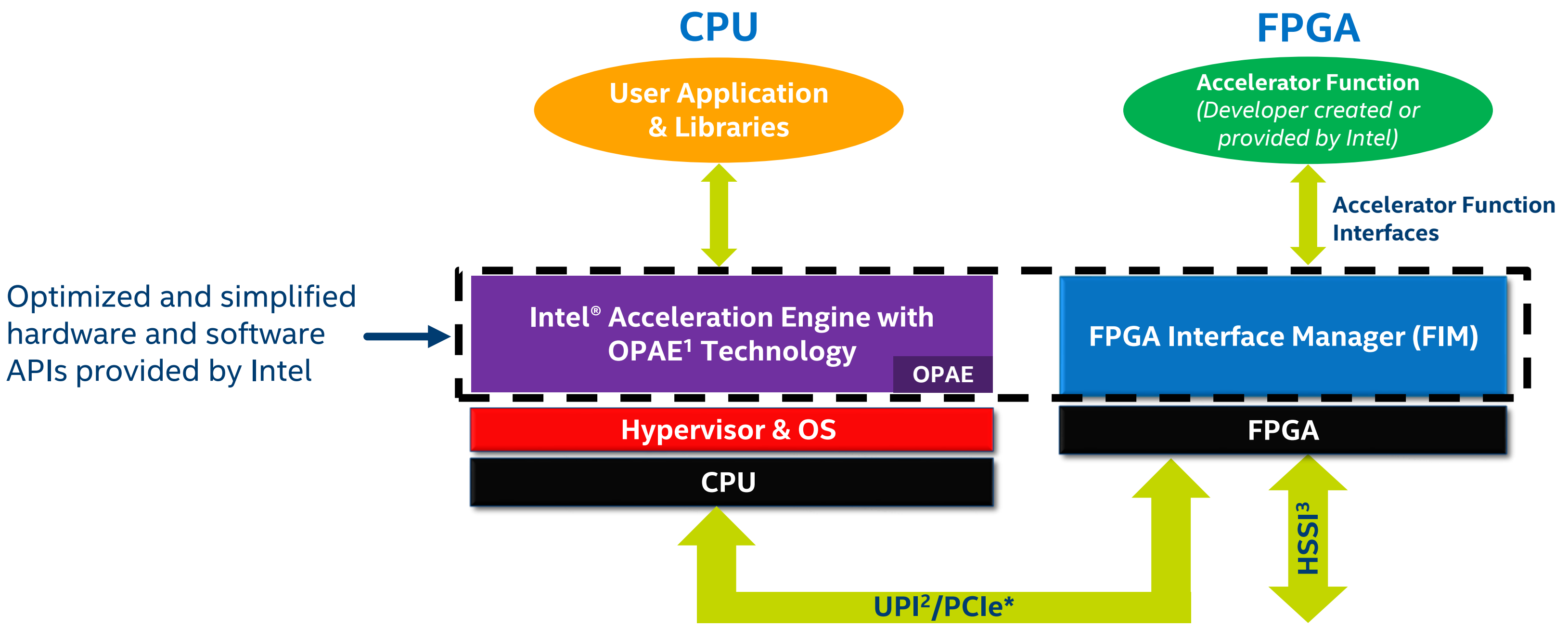
Supports: Red Hat Enterprise Linux* 7.3 w/ kernel 4.7, Intel® Xeon® Processors v4 or newer

Hardware Overview

ACCELERATION ENVIRONMENT

Common Developer Interface For Intel FPGA Data Center Products

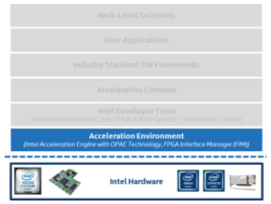

Rack-Level Solutions
User Applications
Industry Standard SW Frameworks
Acceleration Libraries
Intel Developer Tools
(Intel Parallel Studio XE, Intel FPGA SDK for OpenCL™, Intel Quartus® Prime)
Acceleration Environment
(Intel Acceleration Engine with OPAE Technology, FPGA Interface Manager (FIM))
Intel Hardware


CPU
FPGA
User Application & Libraries
Accelerator Function (Developer created or provided by Intel)
Accelerator Function Interfaces
Optimized and simplified hardware and software APIs provided by Intel
Intel® Acceleration Engine with OPAE[1] Technology
OPAE
FPGA Interface Manager (FIM)
Hypervisor & OS
FPGA
CPU
UPI[2]/PCIe*
HSSI[3]

[1]OPAE = Open Programmable Acceleration Engine
[2]UPI = Intel® Ultra Path Interconnect
[3]HSSI = High Speed Serial Interface

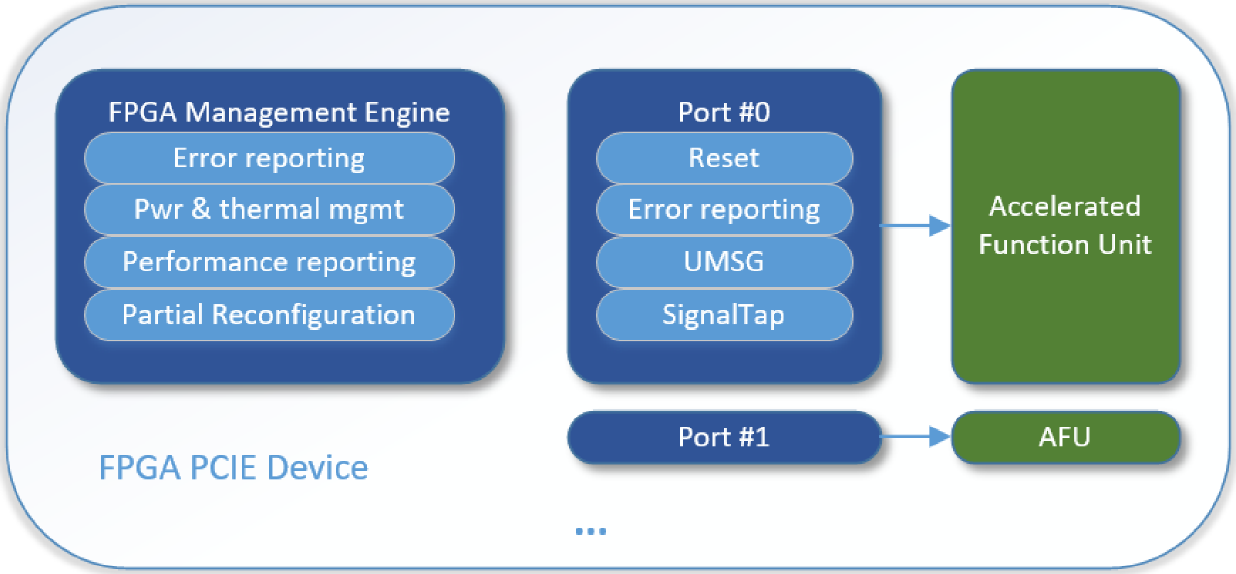
Supports: Red Hat Enterprise Linux* 7.3 w/ kernel 4.7, Intel® Xeon® Processors v4 or newer

FPGA INTERFACE MANAGER (FIM) OVERVIEW

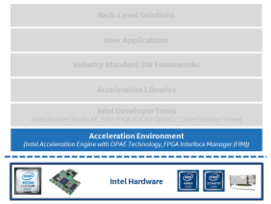

Rack-Level Solutions
User Applications
Industry Standard SW Frameworks
Acceleration Libraries
Intel Developer Tools
(Intel Parallel Studio XE, Intel FPGA SDK for OpenCL™, Intel Quartus® Prime)
Acceleration Environment
(Intel Acceleration Engine with OPAE Technology, FPGA Interface Manager (FIM))
Intel Hardware

Device memory organized in Device Feature List data structure

Supported features exposed through Device Feature List



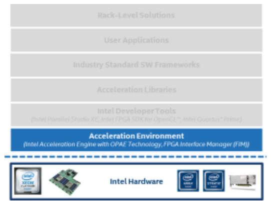
FPGA INTERFACE MANAGER (FIM) DETAILS



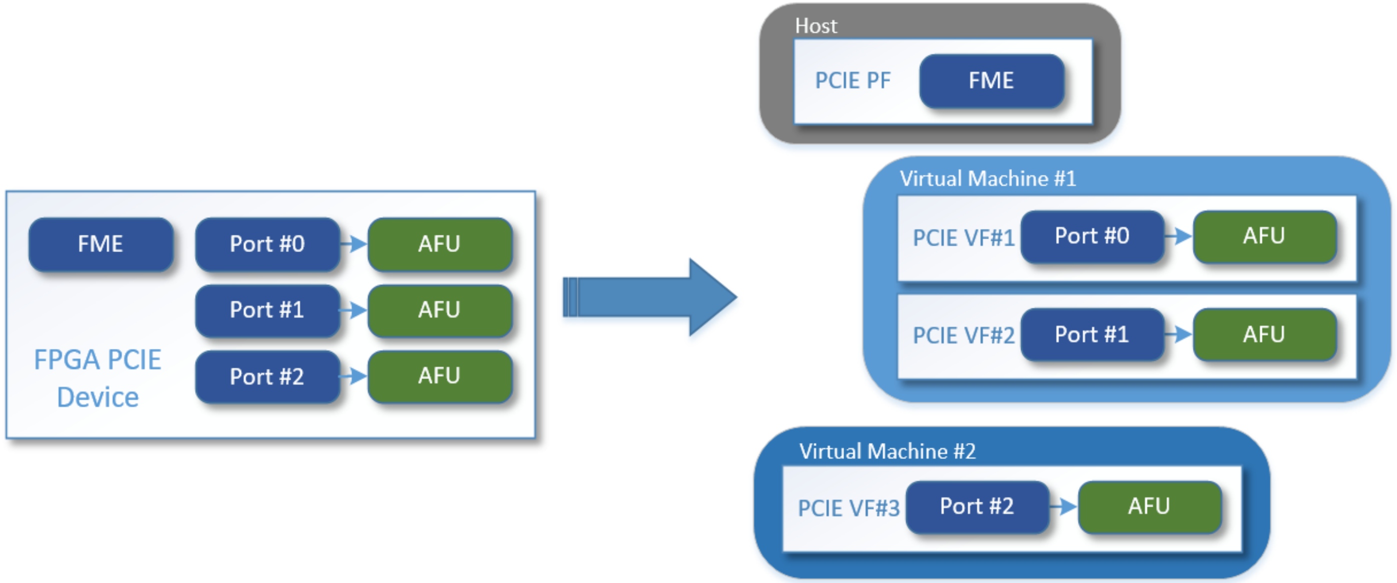
Rack-Level Solutions
User Applications
Industry Standard SW Frameworks
Acceleration Libraries
Intel Developer Tools
(Intel Parallel Studio XE, Intel FPGA SDK for OpenCL™, Intel Quartus® Prime)
Acceleration Environment
(Intel Acceleration Engine with OPAE Technology, FPGA Interface Manager (FIM))
Intel Hardware

- **FPGA Management Engine**
  - Provides: power and thermal management, error reporting, partial reconfiguration, performance reporting, and other infrastructure functions.
  - Each FPGA has one FME, accessible through the physical function.
- **Port**
  - Interface between the static FPGA fabric (FIM) and a partially reconfigurable region containing an Accelerated Function Unit (Accelerator Function).
  - Controls communication from SW and exposes features such as reset and debug.
- **Accelerated Function Unit**
  - Attached to a port and exposes a MMIO region for accelerator-specific control registers.

FPGA INTERFACE MANAGER (FIM) – VIRTUALIZATION SUPPORT

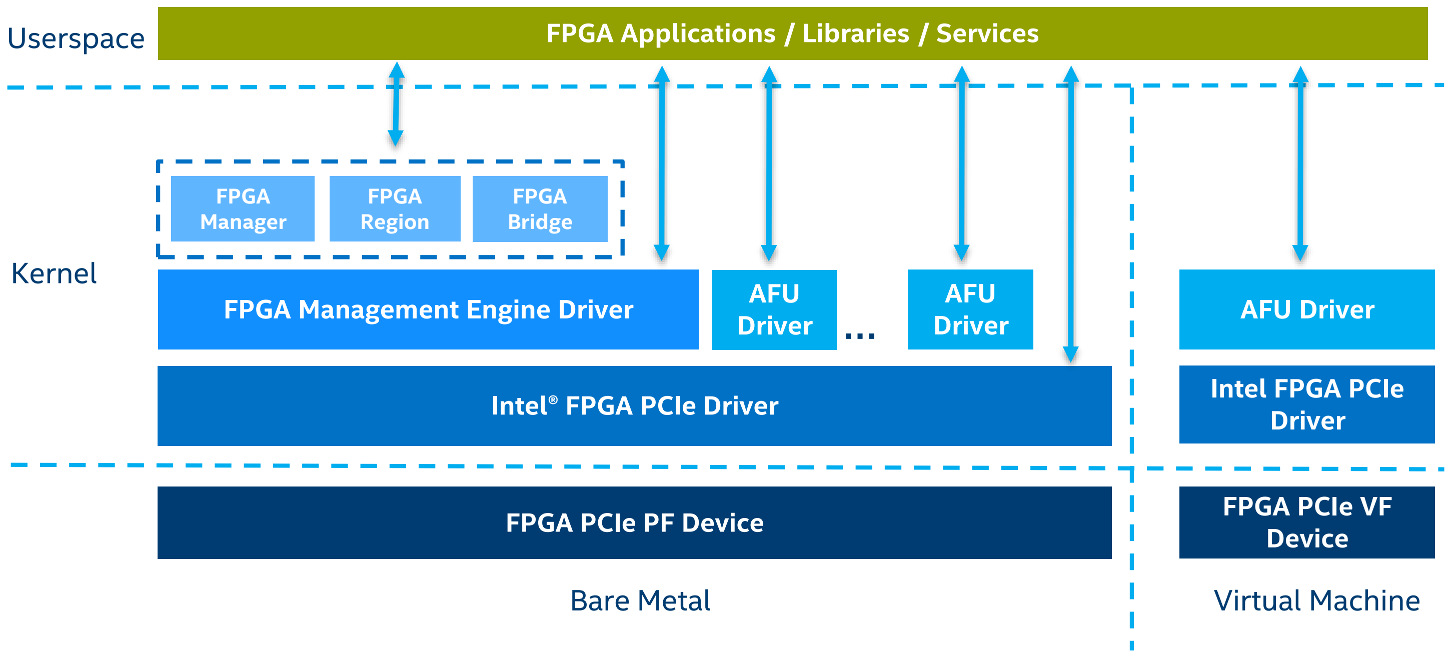

Rack-Level Solutions
User Applications
Industry Standard SW Frameworks
Acceleration Libraries
Intel Developer Tools
(Intel Parallel Studio XE, Intel FPGA SDK for OpenCL™, Intel Quartus® Prime)
Acceleration Environment
(Intel Acceleration Engine with OPAE Technology, FPGA Interface Manager (FIM))
Intel Hardware

Supports PCIe SR-IOV function to create virtual functions (VFs) which can be used to assign individual accelerators to virtual machines.



Intel® FPGA Linux Driver Solution

INTEL® FPGA DRIVER ARCHITECTURE



FPGA DRIVER COMPONENT – PCIE DEVICE DRIVER

- Enumeration
  - Discover Feature Devices by walking through the Device Feature List.
  - Create Platform Device with associated resources for Feature Devices.
    - Port/AFU and FME are Feature Devices.
  - Feature Device Framework:
    - Helper functions to manage feature devices and sub-features.
- SR-IOV Support
  - Sysfs interface to enable/disable VFs.
  - Both PF and VFs share the same driver.

FPGA DRIVER COMPONENT – ACCELERATED FUNCTION UNIT

- Platform Device Driver
- Expose Accelerated Function Unit MMIO Resource
  - MMIO region (mmap)
- Provide DMA buffer mapping service
- Implement other sub features
  - Error reporting
  - UMSG (Unordered Message)
  - Debug

ACCELERATED FUNCTION UNIT – DRIVER INTERFACES

- ioctl
  - Get driver API version (FPGA_GET_API_VERSION)
  - Check for extensions (FPGA_CHECK_EXTENSION)
  - Get port info (FPGA_PORT_GET_INFO)
  - Get MMIO region info (FPGA_PORT_GET_REGION_INFO)
  - Map DMA buffer (FPGA_PORT_DMA_MAP)
  - Unmap DMA buffer (FPGA_PORT_DMA_UNMAP)
  - Reset AFU (FPGA_PORT_RESET)
  - Enable UMsg (FPGA_PORT_UMSG_ENABLE)
  - Disable UMsg (FPGA_PORT_UMSG_DISABLE)
  - Set UMsg mode (FPGA_PORT_UMSG_SET_MODE)
  - Set UMsg base address (FPGA_PORT_UMSG_SET_BASE_ADDR)
- mmap
  - mmap() accelerator MMIO regions.
- sysfs
  - Path: /sys/class/fpga_region/regionX/intel-fpga-afu.n/
  - Read Accelerator GUID (afu_id)
  - Error Reporting (errors/)

FPGA DRIVER COMPONENT – FPGA MANAGEMENT ENGINE

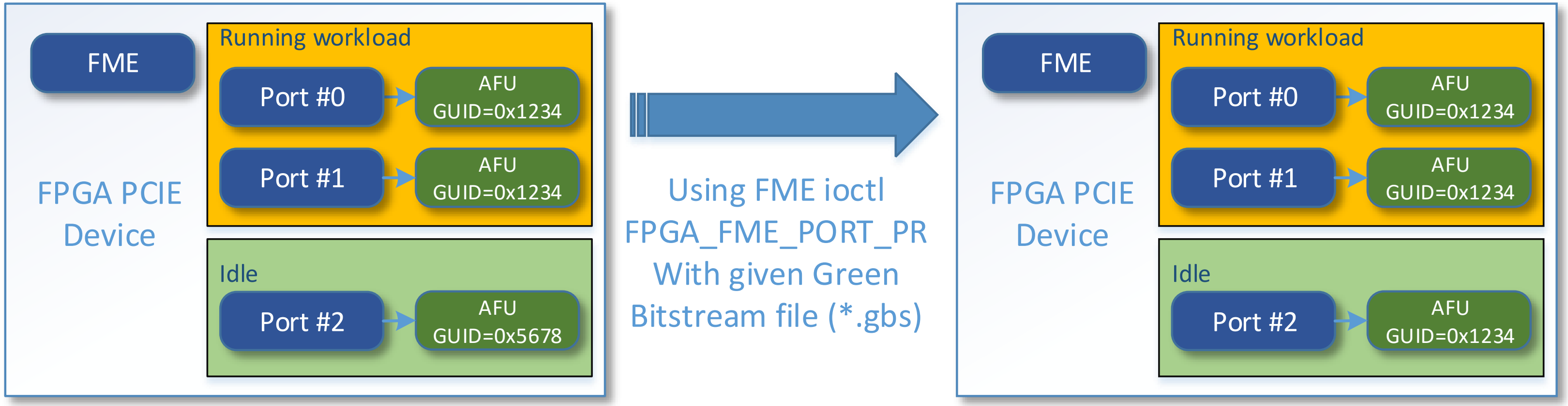
- Platform Device Driver
- Implements management features
  - FPGA capability and status.
  - Thermal & Power management.
  - Partial Reconfiguration function.
  - Global Error reporting.
  - Global Performance reporting.
  - Port Management.

FPGA MANAGEMENT ENGINE – DRIVER INTERFACES

- ioctl
  - Get driver API version (FPGA_GET_API_VERSION)
  - Check for extensions (FPGA_CHECK_EXTENSION)
  - Assign port to PF (FPGA_FME_PORT_ASSIGN)
  - Release port from PF (FPGA_FME_PORT_RELEASE)
  - Program Bitstream (FPGA_FME_PORT_PR)
- sysfs
  - Path: /sys/class/fpga_region/regionX/intel-fpga-fme.n/
  - Read bitstream ID/metadata (bitstream_id / bitstream_metadata)
  - Read number of ports (ports_num)
  - Read socket ID (socket_id)
  - Read performance counters (perf/)
  - Power management (power_mgmt/)
  - Thermal management (thermal_mgmt/)
  - Error reporting (errors/)

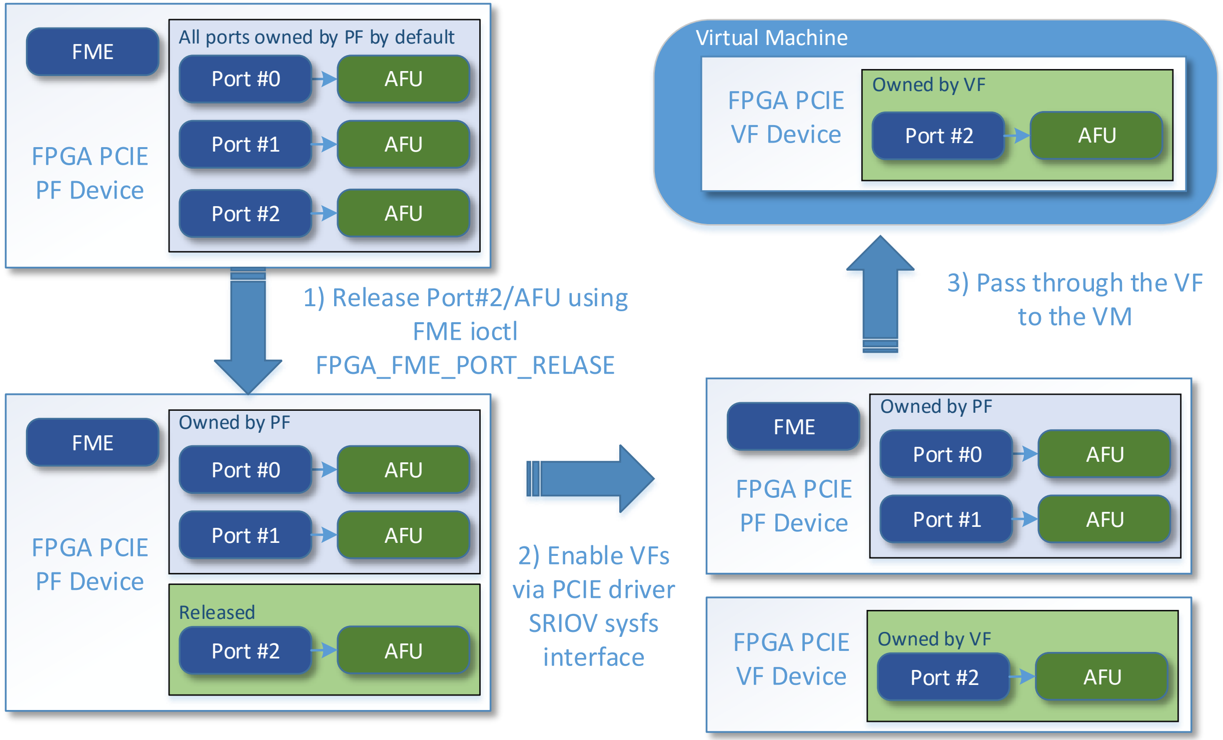
PARTIAL RECONFIGURATION (PR)

- Accelerators reconfigured through partial reconfiguration.
- Other AFUs could run workload at the same time.
- Interface compatibility needed before start PR.



VIRTUALIZATION

To enable access to an accelerator from within a virtual machine, AFU ports must be assigned to a VF



EXAMPLE: SIMPLE DMA OPERATION

- Open AFU device file.
- Use AFU Ioctl FPGA_PORT_GET_REGION_INFO to get AFU MMIO region information.
- Invoke AFU mmap to map AFU MMIO region for CSRs access.
- Allocate buffer and do DMA mapping via AFU ioctl FPGA_PORT_DMA_MAP.
- Program the DMA address to related CSRs in mapped area.
- Start DMA by programming related CSRs in mapped area.
- Poll on CSRs for completion.
- Unmapp DMA by AFU ioctl FPGA_PORT_DMA_UNMAP
- Close AFU device file as task is done.

UPSTREAMING STATUS

- First batch of patches has been submitted, version 2 is under review now.
  - Basic functions to enable AFU usage and Partial Reconfiguration.
    - Link: https://marc.info/?l=linux-fpga&m=149844232609819&w=2
  - Documentation/fpga/intel-fpga.txt:
    - https://marc.info/?l=linux-fpga&m=149844234509825&w=2

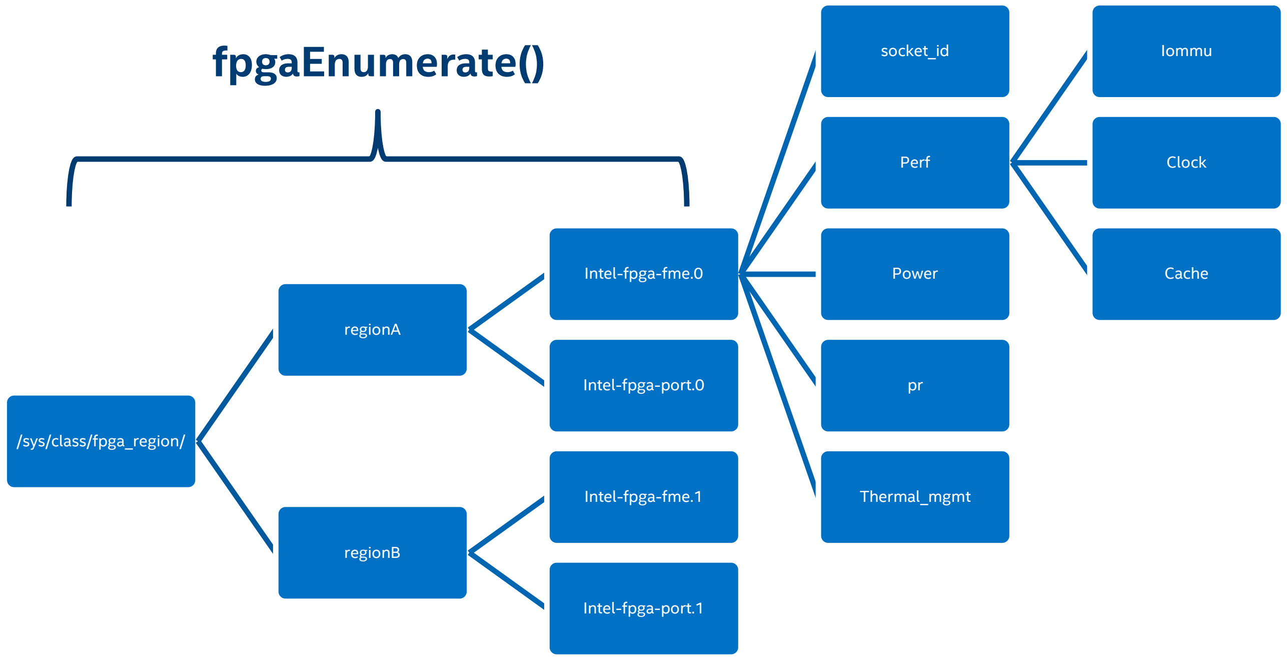
- More advanced features to be submitted next step
  - SR-IOV support and other sub features for FME and PORT/AFU.

START DEVELOPING FOR INTEL® FPGAS WITH OPAE TODAY

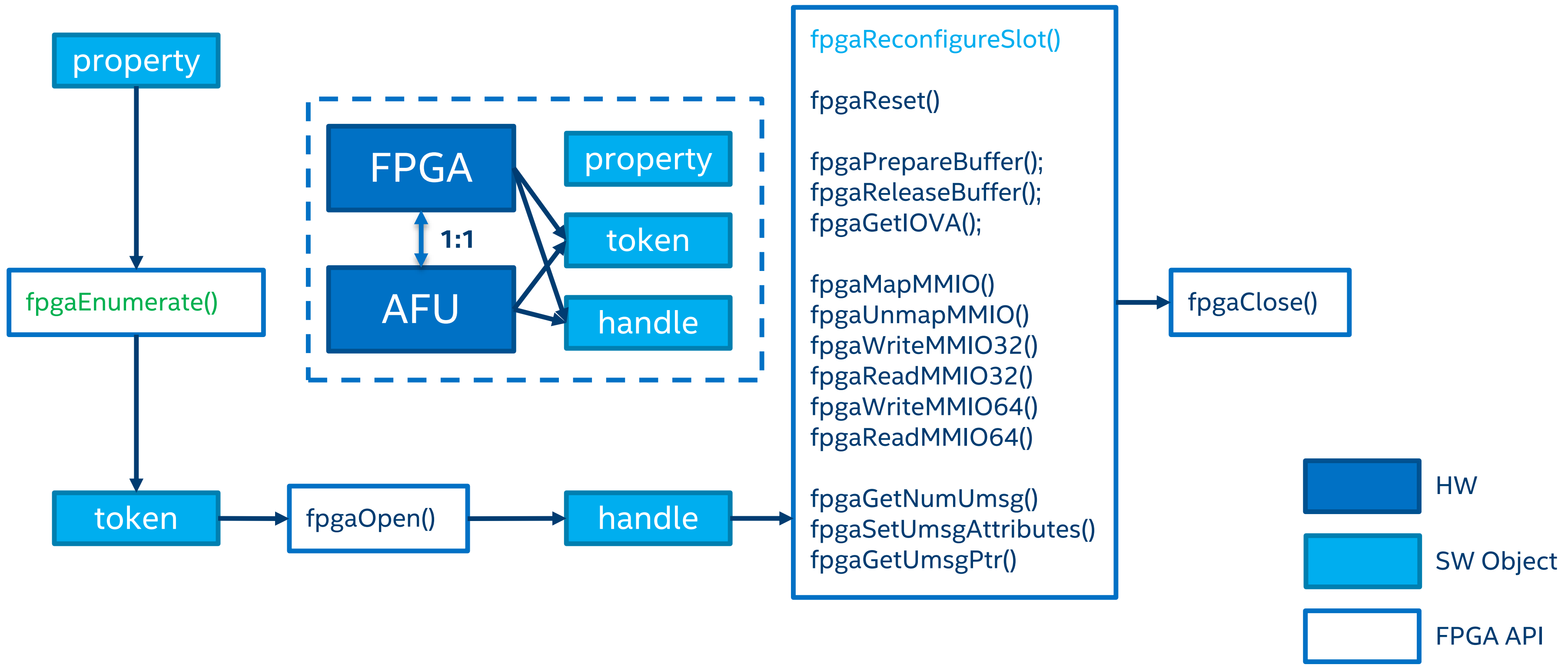
- Learn more about OPAE by visiting: http://01.org/OPAE
- Join the OPAE mailing list: https://lists.01.org/pipermail/opae/

BACKUP

FPGA API – SYSFS & ENUMERATION



FPGA API – ENUMERATE, MANAGE & ACCESS



fpgaReconfigureSlot()

fpgaReset()

fpgaPrepareBuffer();
fpgaReleaseBuffer();
fpgaGetIOVA();

fpgaMapMMIO()
fpgaUnmapMMIO()
fpgaWriteMMIO32()
fpgaReadMMIO32()
fpgaWriteMMIO64()
fpgaReadMMIO64()

fpgaGetNumUmsg()
fpgaSetUmsgAttributes()
fpgaGetUmsgPtr()

OPAE USERSPACE API ACCESS

```
fpga_result fpgaOpen(fpga_token token, fpga_handle *handle, int flags);
fpga_result fpgaClose(fpga_handle handle);
fpga_result fpgaReset(fpga_handle handle);
fpga_result fpgaPrepareBuffer(fpga_handle handle, uint64_t len, void **buf_addr, uint64_t *wsid, int
flags);
fpga_result fpgaReleaseBuffer(fpga_handle handle, uint64_t wsid);
fpga_result fpgaGetIOVA(fpga_handle handle, uint64_t wsid, uint64_t *iova);

fpga_result fpgaMapMMIO(fpga_handle handle, uint32_t mmio_num, uint64_t **mmio_ptr)
fpga_result fpgaUnmapMMIO(fpga_handle handle, uint32_t mmio_num)
fpga_result fpgaWriteMMIO32(fpga_handle handle, uint32_t mmio_num, uint64_t offset, uint32_t value)
fpga_result fpgaReadMMIO32(fpga_handle handle, uint32_t mmio_num, uint64_t offset, uint32_t *value)
fpga_result fpgaWriteMMIO64(fpga_handle handle, uint32_t mmio_num, uint64_t offset, uint64_t value)
fpga_result fpgaReadMMIO64(fpga_handle handle, uint32_t mmio_num, uint64_t offset, uint64_t *value)
fpga_result fpgaWriteMMIO(fpga_handle handle, uint32_t mmio_num, uint64_t offset, uint64_t value)
fpga_result fpgaReadMMIO(fpga_handle handle, uint32_t mmio_num, uint64_t offset, uint64_t *value)

fpga_result fpgaGetNumUmsg(fpga_handle handle, uint64_t *value)
fpga_result fpgaSetUmsgAttributes(fpga_handle handle, uint64_t value)
fpga_result fpgaGetUmsgPtr(fpga_handle handle, uint64_t **umsg_ptr)
```

Notices & Disclaimers