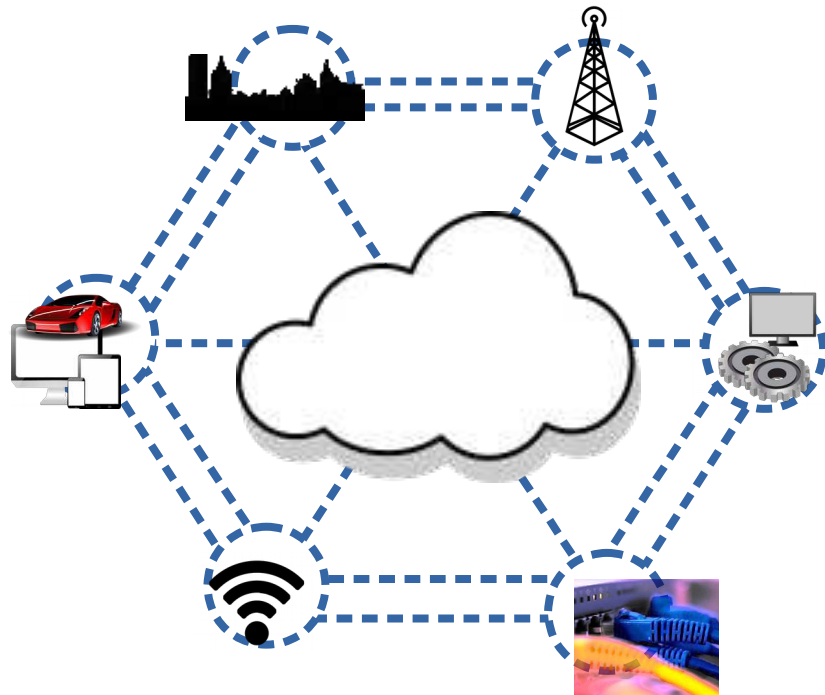




vFPGAmanager: a Virtualization Framework for Orchestrated FPGA Accelerator Sharing in 5G Cloud Environments



**IEEE BMSB 2018, June 2018,
Valencia**

Sébastien PINNETERRE

contact@virtualopensystems.com

www.virtualopensystems.com



Authorship and sponsorship

Sébastien PINNETERRE, FPGA designer at Virtual Open Systems.

Virtual Open Systems is a high-tech software company active in open source virtualization solutions and custom services for complex mixed-criticality automotive systems, NFV networking infrastructures, consumer electronics, mobile devices and in general for embedded heterogeneous multicore systems around new generation processor architectures.

This work is part of the H2020 “**Next Generation Platform As A Service**” project (<http://ngpaas.eu/>).

NGpaas



FPGA Virtualization challenges

Network Function Virtualization is a key enabler of the coming 5G in different areas like Multimedia broadcasting, Telecommunications and networking technology.

Applications need FPGA acceleration to provide right level of performance. The domains that could benefit from FPGA are many : Advanced Driver Assistance Systems (ADAS), Internet of Things (IoT), High Performance Computing (HPC), Artificial Intelligence (AI).

However in 5G NFV distributed cloud computing environment:

- FPGA (Development tools, drivers, etc) today doesn't fully support virtualization
- VIMs (e.g. OpenStack) and orchestrators don't support FPGA
- GPUs and CPUs do not provide the same performance/watt level ratio as an FPGA

Virtual Open Systems propose an FPGA virtualization solution which uses virtualization technologies, SR-IOV and dynamic partial reconfiguration



SotA: Virtualization hardware extensions features

Virtualization extensions provide hardware support for virtualization.

Examples are:

- Intel Virtualization Technology (Intel VT-x),
- AMD Secure Virtual Machine (SMV),
- ARM Virtualization extensions

For input and output operations processor companies use Input/Output Memory Management Unit (IOMMU) hardware component:

- Hardware block (AMD-Vi, Intel VT-d, ARM SMMU)
- Intercepts any memory access from hardware components
- Enables direct device access to userspace



SotA: Single Root I/O Virtualization feature (I)

Single Root I/O Virtualization (SR-IOV) is an extension of the PCI Express specification.

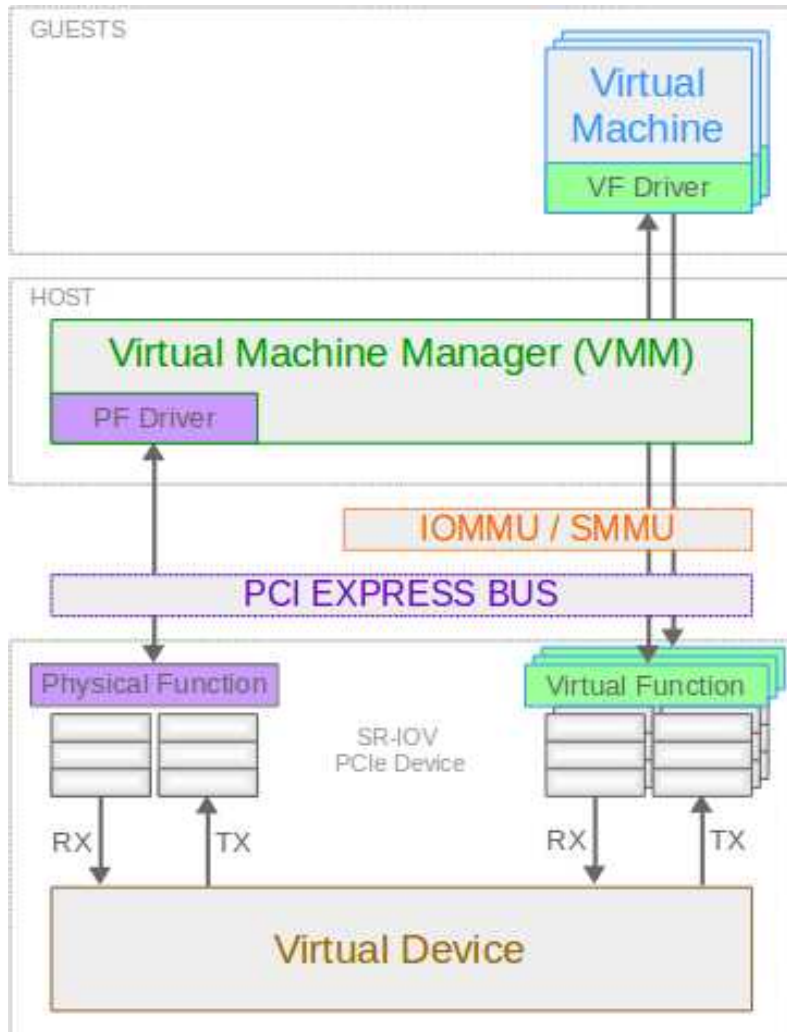
It allows a device to separate access to its resources among various PCIe hardware functions of the following types:

- PCIe Physical Functions (PFs)
- PCIe Virtual Functions (VFs)





SotA: Single Root I/O Virtualization feature (II)



Physical Function (PF):

- Manages the SR-IOV functionality
- Is fully featured PCIe function and used like a PCIe device
- Configure and manage its associated VFs

Virtual Function (VF):

- Lightweight PCIe function associated with a PF
- Shares one or more physical resources with the PF and with VF that are associated with that PF
- VF can only configure its own behavior



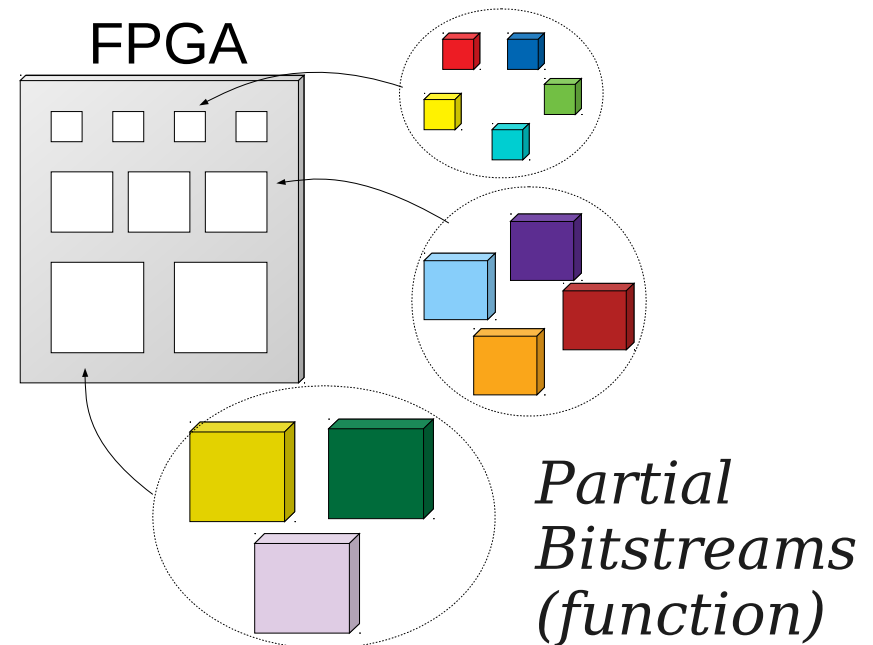
SotA: Dynamic Partial Reconfiguration feature

FPGA Dynamic Partial Reconfiguration:

- Ability to dynamically modify blocks of logic by downloading partial bit files while the remaining logic continues to operate without interruption
- Permits to change functionality on the fly

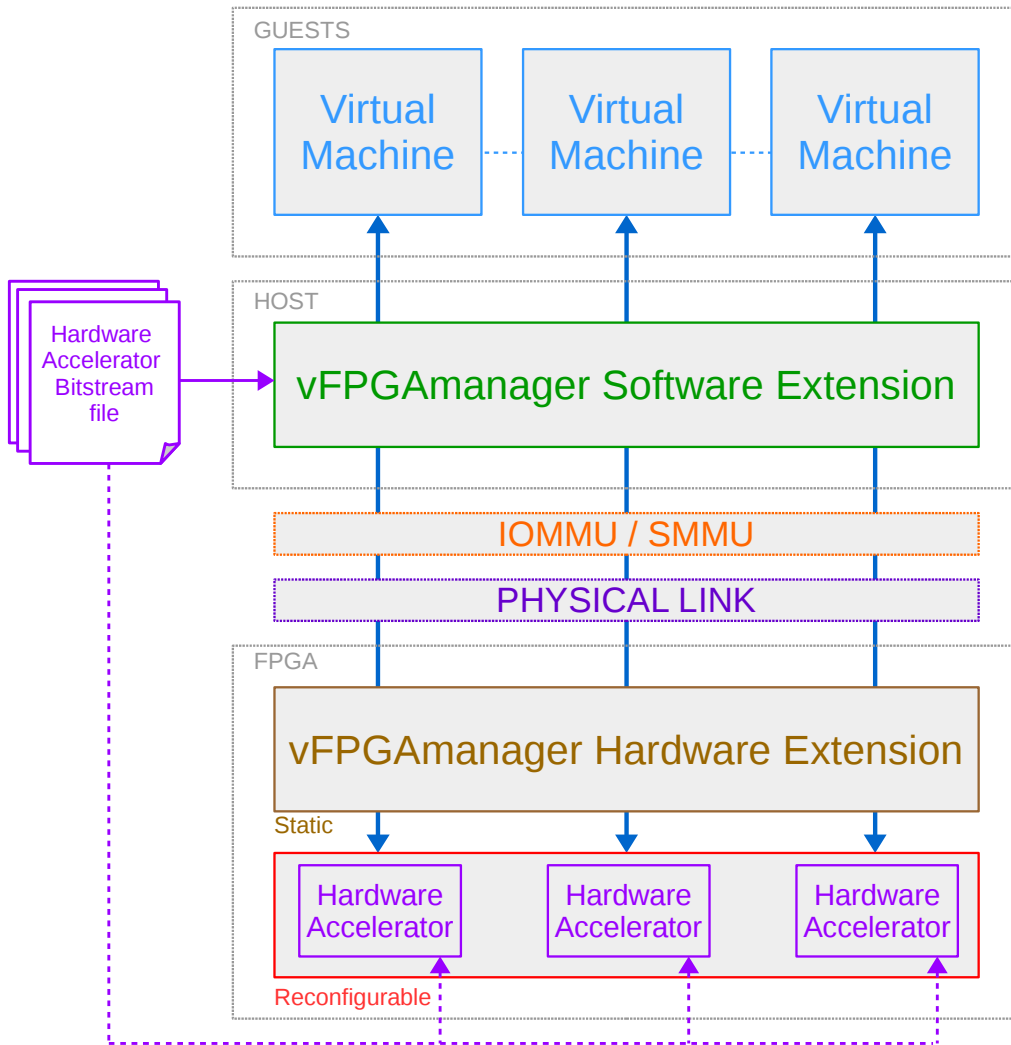
Example:

- Accelerators are configured in reconfigurable partitions
- Runtime reconfiguration
- For one accelerator, one bitstream is required per partition





What is the vFPGAmanager ?



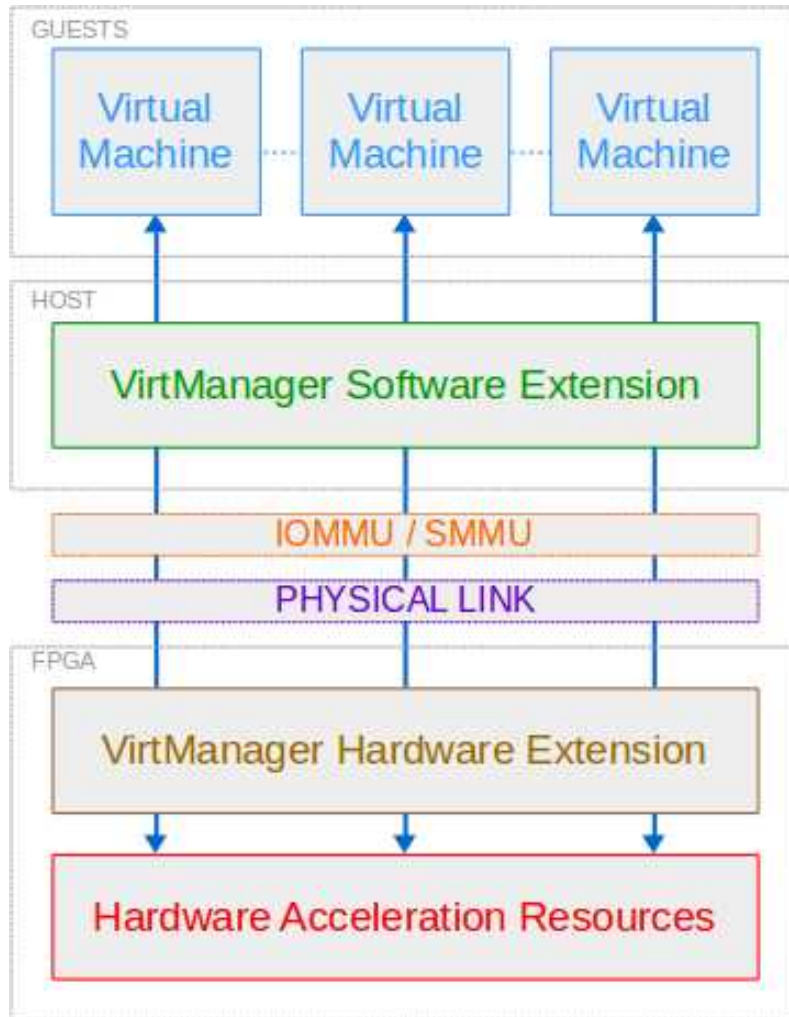
The vFPGAmanager component is an FPGA IP core which enables direct communication between the VMs and the HW accelerators.

- Allocates an accelerator to multiple guests (needed for microservices)
- Schedules accelerators deployment at run time based on QoS policies
- Supports existing VM and hardware accelerators

vFPGAmanager is a VOSYS patented technology which relies on key technologies like Single Root I/O Virtualization (SR-IOV) and FPGA Dynamic Partial Reconfiguration (DRP)



vFPGAmanager: Components overview

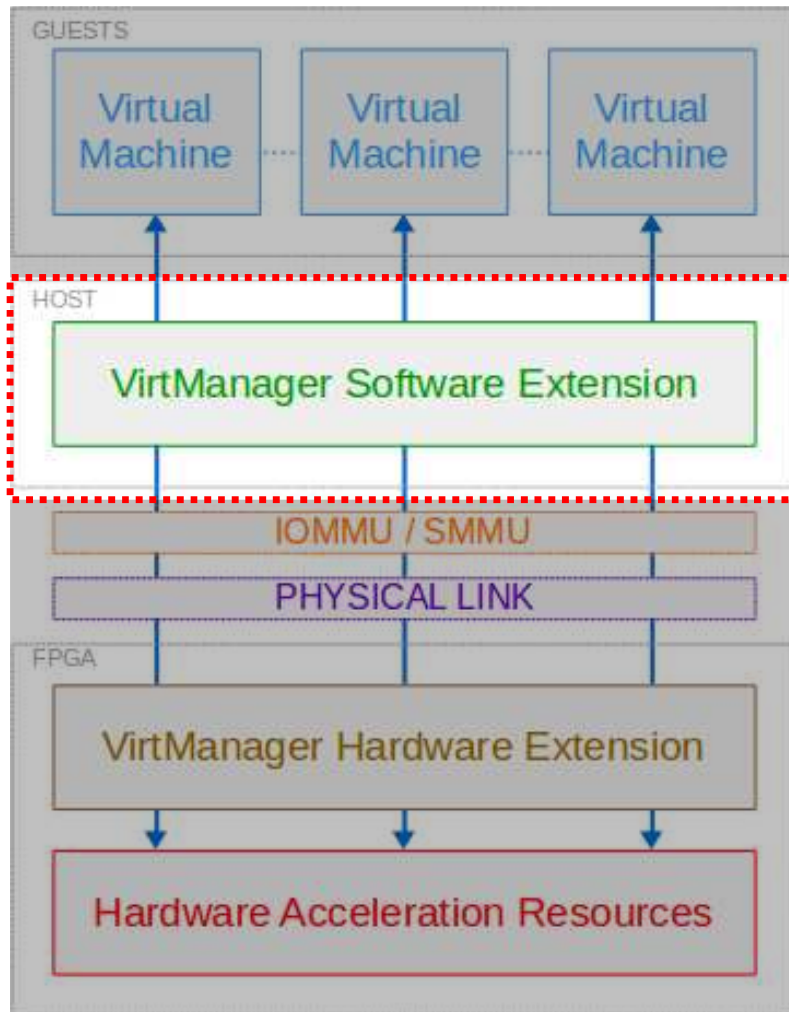


Several components are required to enable the virtualization of hardware resources. These components can be split to 3 categories:

- Host software
- Guest software
- FPGA hardware



vFPGAmanager: Host software components (I)



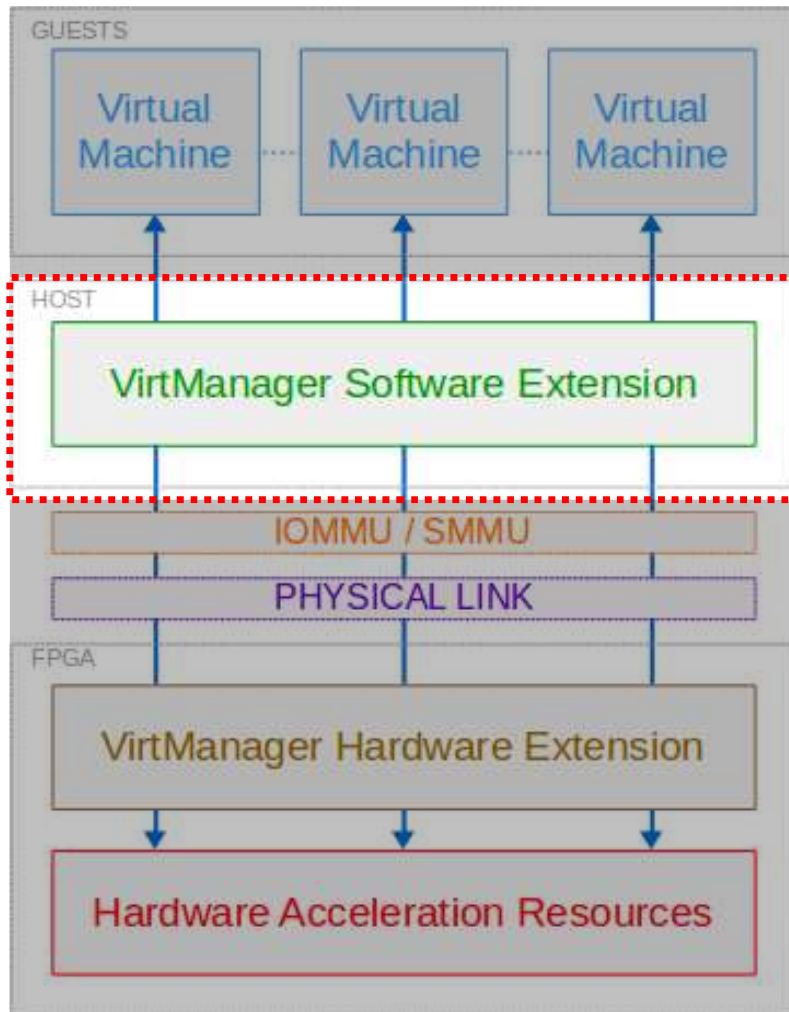
Host software component are:

- KVM and QEMU
 - It provides a front-end to the KVM capabilities of the Linux kernel, enables direct access to hardware PCI devices and live migrations of virtual machines between hosts
- Libvirt
 - It provides a collection of API, daemon and a console tool, manages virtualization platforms and interfaces with number of virtualization solutions (Qemu/KVM)

vFPGAmanager supports also docker and unikernel



vFPGAmanager: Host software components (II)

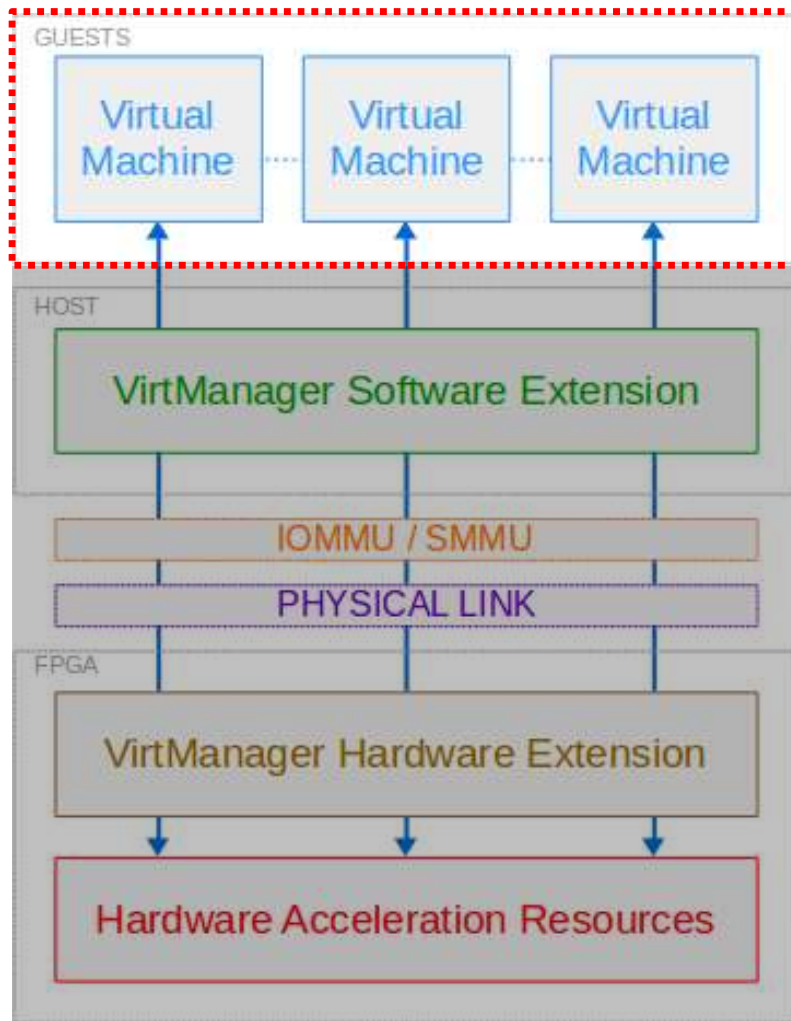


Host software component are:

- Drivers
 - PCI PF Driver
 - Manages SR-IOV capability
 - Controls the vFPGAmanager hardware components
 - VFIO Driver
 - Exposes direct device access to userspace to allow a VMs to be interfaced directly to an hardware accelerator
- Software daemons for accelerators sharing
- Cloud/Orchestration application



vFPGAmanager: Guests software components



Guest software component are:

➤ Drivers

■ PCI VF Driver

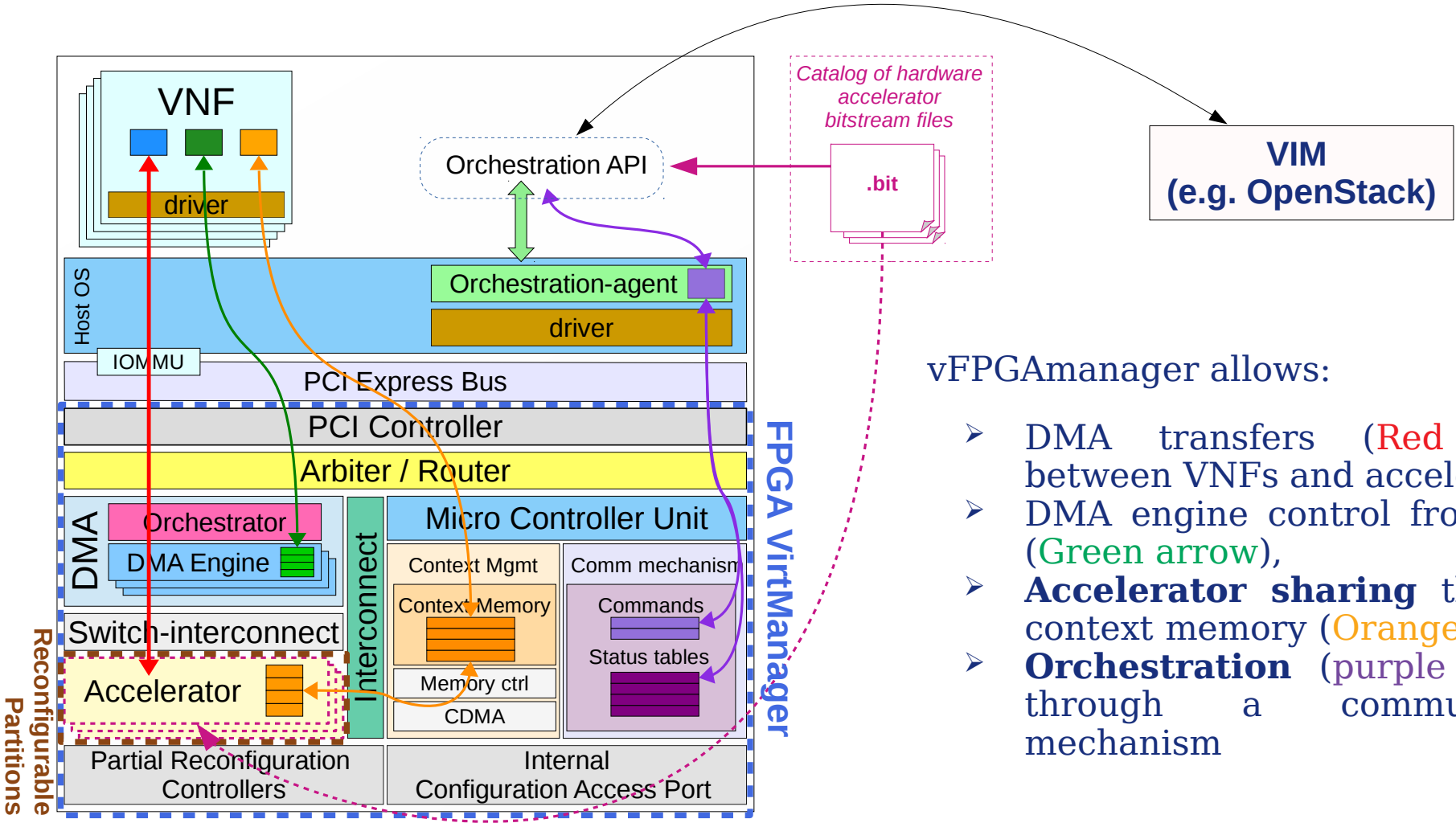
- Accelerators drivers addressing several types of hardware accelerators

■ Application

- The application running in a VM, a container or a unikernel using hardware FPGA resources through the VF driver



vFPGAmanager: Hardware Architecture

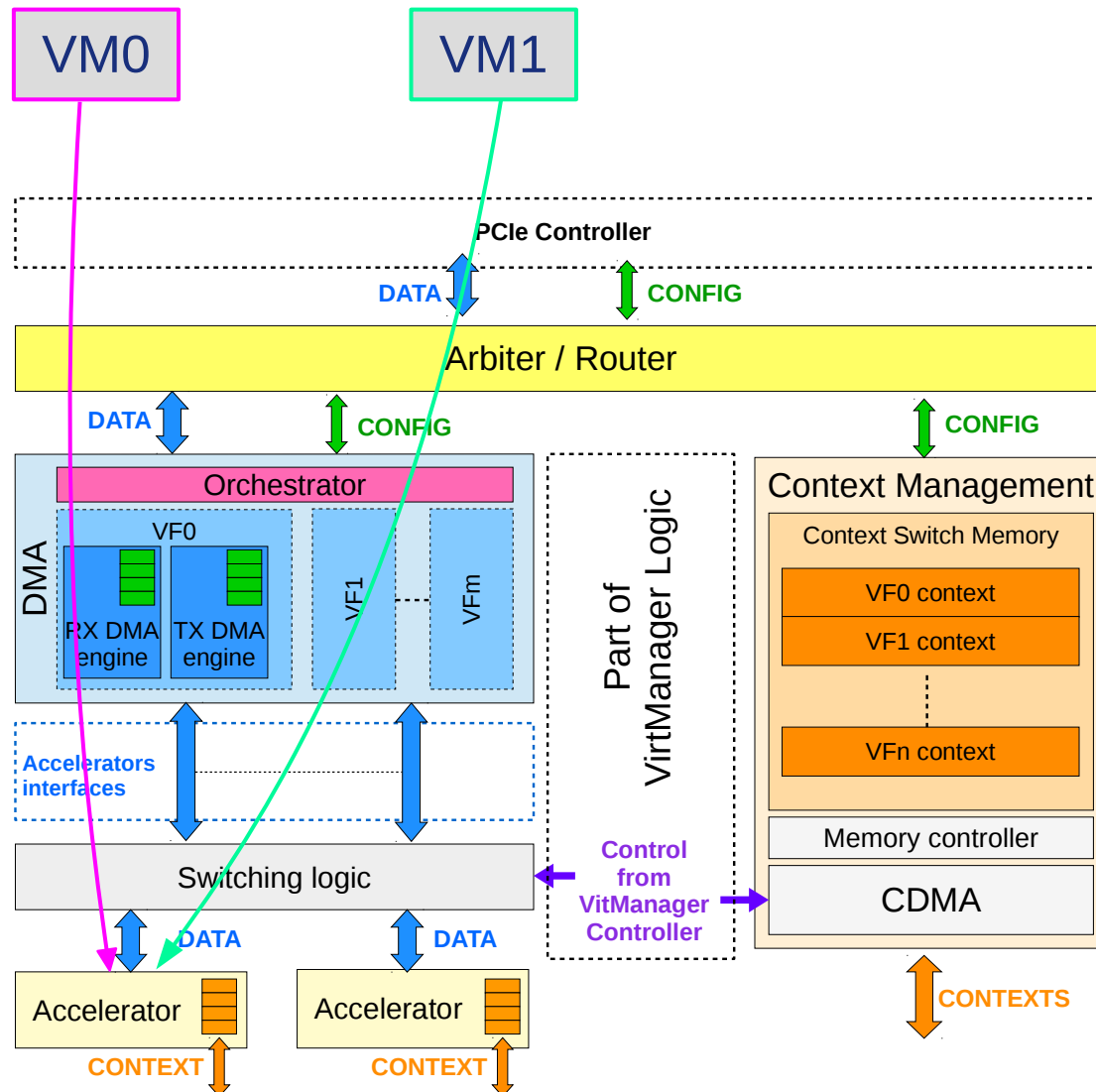


vFPGAmanager allows:

- DMA transfers (Red arrow) between VNFs and accelerators,
- DMA engine control from VNFs (Green arrow),
- **Accelerator sharing** through a context memory (Orange arrows)
- **Orchestration** (purple arrows) through a communication mechanism



vFPGAmanager: Hardware accelerator sharing

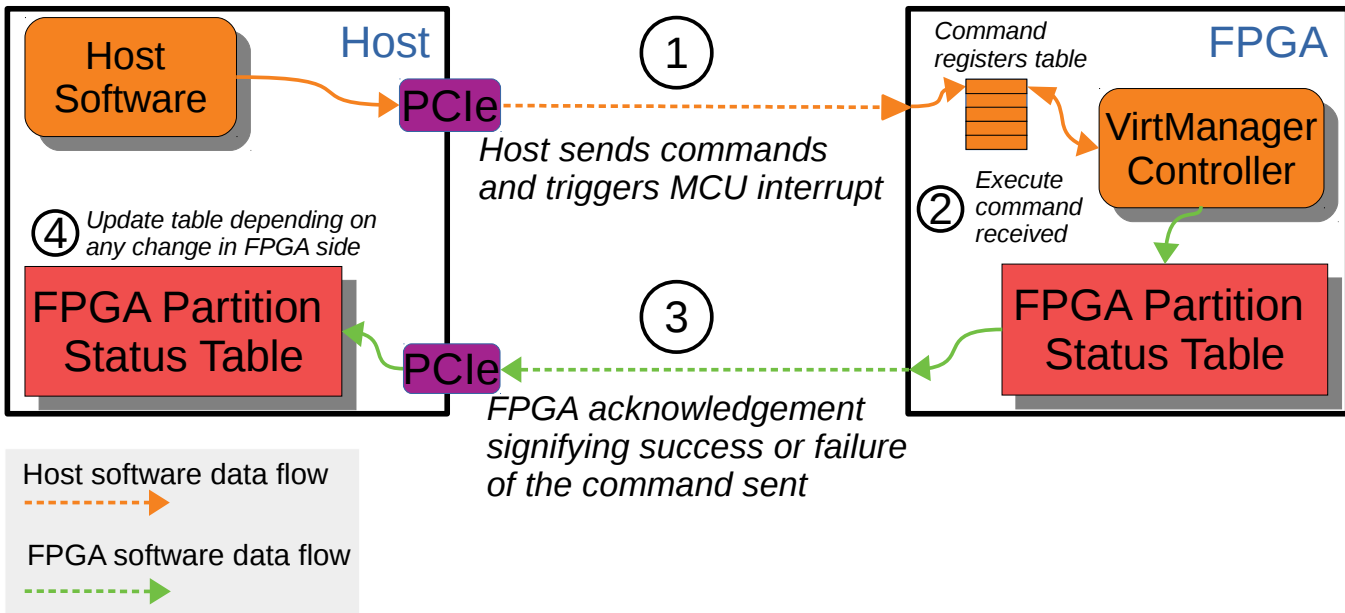


The vFPGAmanager hardware accelerators sharing feature uses software and hardware components which are:

- VFIO to expose DMA engines (RX & TX) and the context memory space exposed to VMs
- A SR-IOV compliant DMA to move data between VMs and hardware accelerators
- An AXI switch logic controlled by the vFPGAmanager MCU to map DMA and hardware accelerator
- **Context Management** block controlled by the vFPGAmanager MCU to perform a hardware accelerator **context switch**



vFPGAmanager: Orchestration Command Flow



A **communication mechanism**, based on set of commands (Attach VM ID to Accelerator ID, detach, etc), exposes to the orchestration (e.g. OpenStack), control and status registers, that allow resource monitoring and control of accelerator deployment.



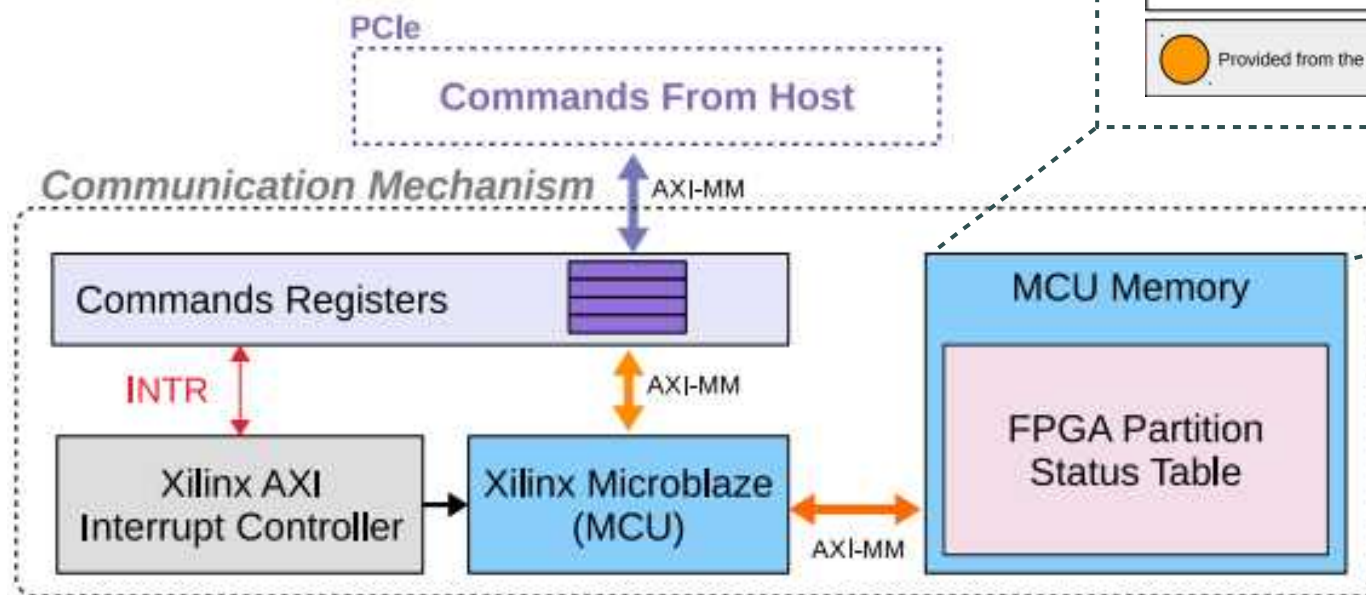
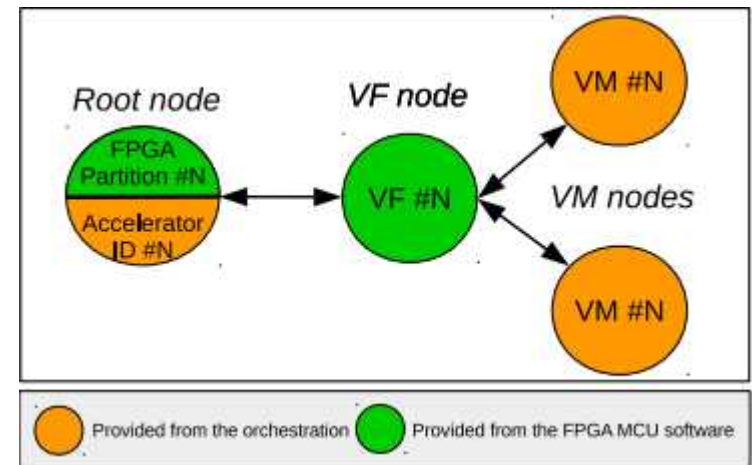
vFPGAmanager: Communication mechanism simplified architecture

Partition status table is based on AVL tree

Tracked informations are:

- VMs \Leftrightarrow VFs
- VFs \Leftrightarrow Partition
- Acc ID \Leftrightarrow Partition

AVL tree

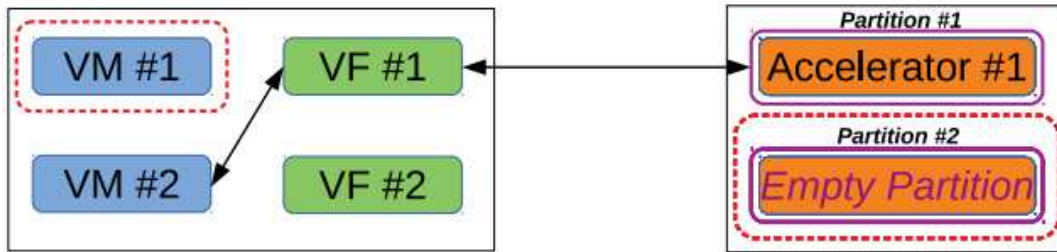




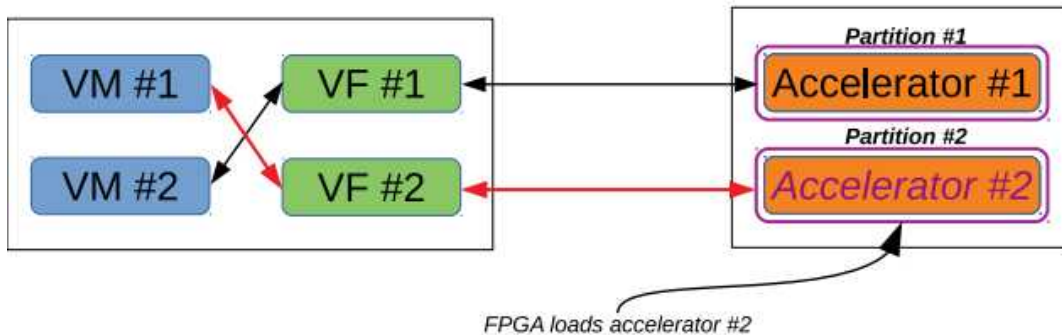
vFPGAmanager: Attachment command example and performances measurement

Example of an accelerator attachment

Request for Attachment on Accelerator #2



Attachment Established



VM Attach Time:

#VM concurrent request to attach	Time (us)
1	10
10	111
100	1468
1000	18079

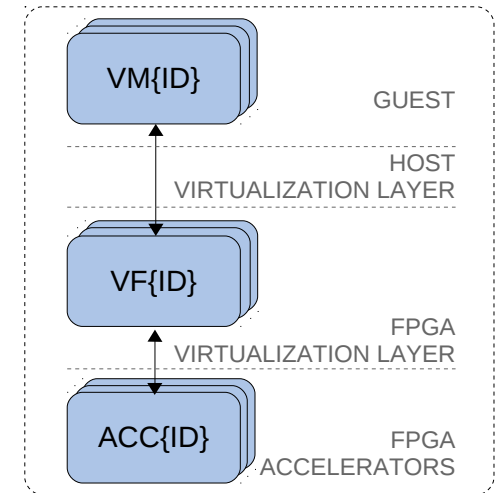
Time for a “VM attach command” to be processed by the FPGA MCU and return an acknowledgment



vFPGAmanager: VIM orchestration

vFPGAmanager orchestration commands:

Command	Argument(s)	Description
Attach	VM ID ; ACC ID	Attach a VM{ID} to an ACC{ID} Return a VF{ID}
Detach	VM ID ; ACC ID	Detach a VM{ID} from an ACC{ID} Release a VF{ID}
...



OpenStack Cyborg REST API v1.0:

Verb	URI	Description
GET	/accelerators	Return a list of accelerators
GET	/accelerators/{uuid}	Retrieve a certain accelerator info identified by `{uuid}`
POST	/accelerators	Create a new accelerator.
PUT	/accelerators/{uuid}	Update the spec for the accelerator identified by `{uuid}`
DELETE	/accelerators/{uuid}	Delete the accelerator identified by `{uuid}`





Conclusion

Outcomes:

FPGA acceleration is today a must for several applications. VOSYS FPGA virtualization framework provides:

- Hardware accelerator sharing
- Orchestration
- High performance
- Accelerator direct communication

Future work:

This feasibility study of FPGA accelerator orchestration for 5G Cloud Environments will be consolidated to be compliant with major open source project like OpenStack Nova and Cyborg. Additionally, next development steps will go in the direction of adding support for partial reconfiguration at the accelerators level.



Questions?
Thank you!



contact@virtualopensystems.com

Web: [virtualopensystems.com](http://www.virtualopensystems.com)

Products: <http://www.virtualopensystems.com/en/products/>

Demos: [virtualopensystems.com/en/solutions/demos/](http://www.virtualopensystems.com/en/solutions/demos/)

Guides: [virtualopensystems.com/en/solutions/guides/](http://www.virtualopensystems.com/en/solutions/guides/)

Research projects: [virtualopensystems.com/en/research/innovation-projects/](http://www.virtualopensystems.com/en/research/innovation-projects/)