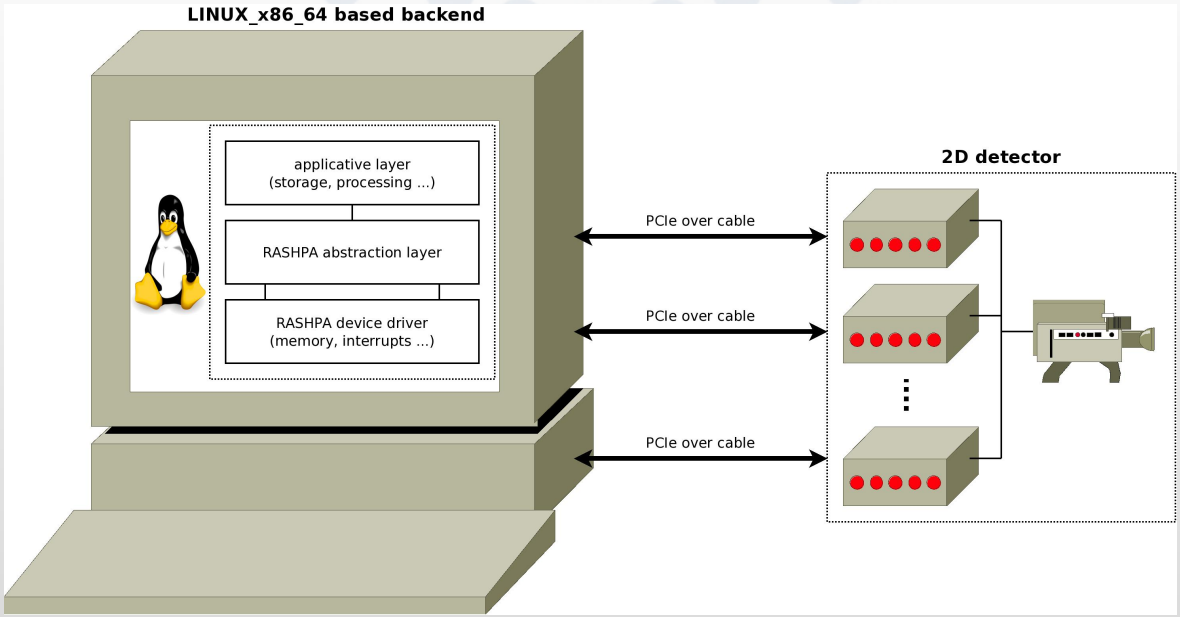
The use of hardware virtualization in RASHPA

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

Contents

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

Context - RASHPA reminder



RASHPA basic goal is to transfer **detector** memory contents into a **backend** for further processing, visualisation or storage. To do so, RASHPA relies on low level hardware and software components:

- data **transport layer**, currently PCI Express over cable
- a data **transmission engine** implemented on XILINX FPGA
- a LINUX **software stack**, ie. kernel device driver and applications

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

Context - RASHPA project issues

Still being prototyped

- how to simplify hardware software **codesign** ?

Aims at using multiple PCIe links. Yet, the first prototype is single link

- how to **scale** the prototype platform ?

Well positionned to interface with PCIe based accelerating technologies

- how to test **unavailable hardware** (too recent or expensive) ?

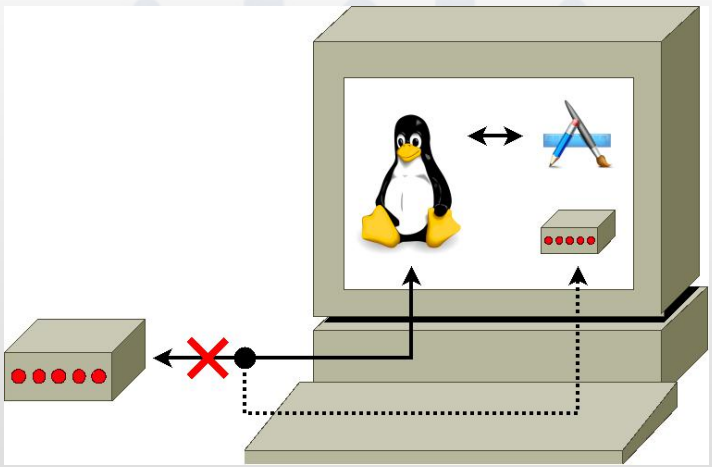
CRISP
The Cluster of Research Infrastructures for Synergies in Physics

Context - Our approach

Our approach: **device virtualization**

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

Device virtualization - main idea



Applications on a **host** machine access hardware via interfaces. By **instrumenting** these interfaces, one can **redirect** the accesses to a software implementing the device. The device is said to be **virtualized**.

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

Device virtualization - assumptions

Virtualization assumes device accesses are made using a known interface

- software library API (system calls)
- memory mapped registers
- CPU specific instructions (in, out)

Device virtualization - mechanisms

The interface is then instrumented to redirect access to the virtual device

- software hooking (code instrumentation, library replacing)
- **instruction emulation (QEMU)**
- hardware traps (page protection)
- architecture support (INTEL VT)
- paravirtualization (XEN)

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

Device virtualization - applications

Device virtualization example applications

- support: application running on unmaintained platform
- security: sandboxing, reverse engineering
- quality: debugging, fault injection
- testing: milkymist, zinq, android

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - introduction and goals

VPCIe: a framework to virtualize PCIe devices

- made at the ESRF in the context of CRISP/RASHPA
  - still, generic enough to fit other project needs
- the backend software must run **without any modification**
- virtual device implementation usable in the real device
  - made possible with **VHDL**

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - opensource building blocks

VPCIe mostly relies on opensource projects

QEMU

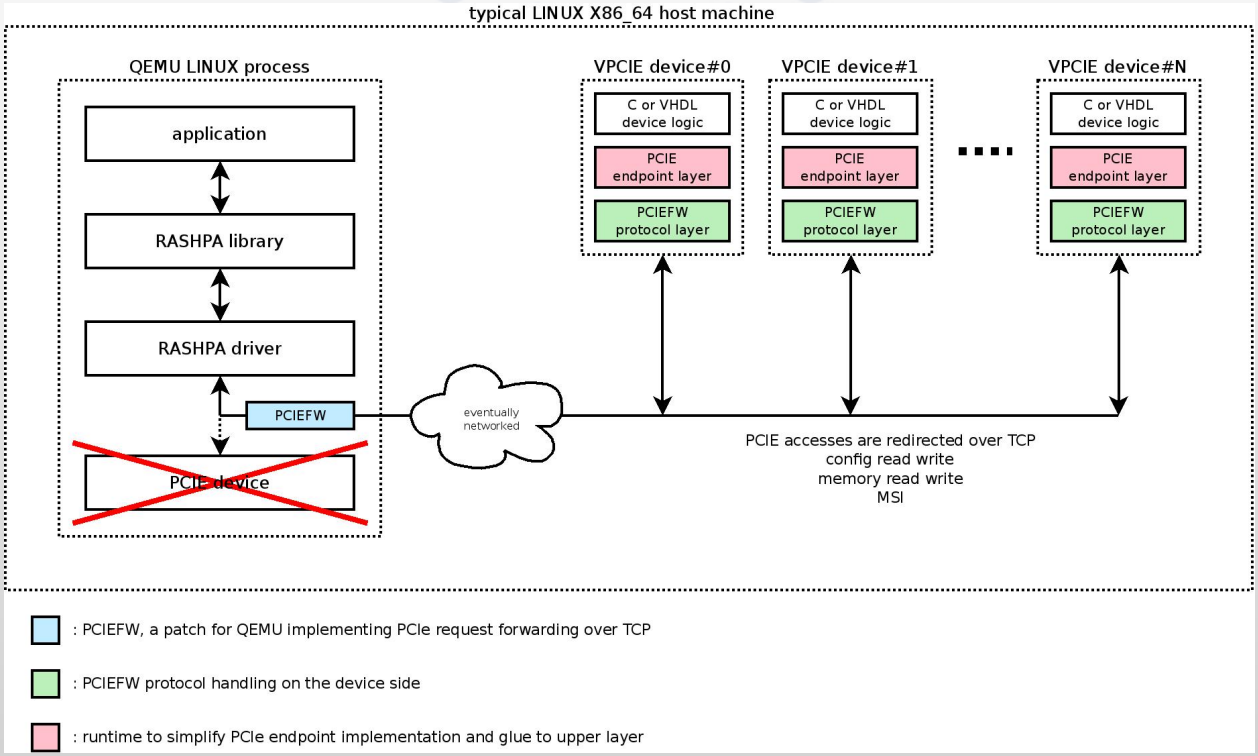
- http://wiki.qemu.org/Main_Page
- architecture emulator (X86_64, ARM ...)
- used to trap PCIe hardware accesses

GHDL

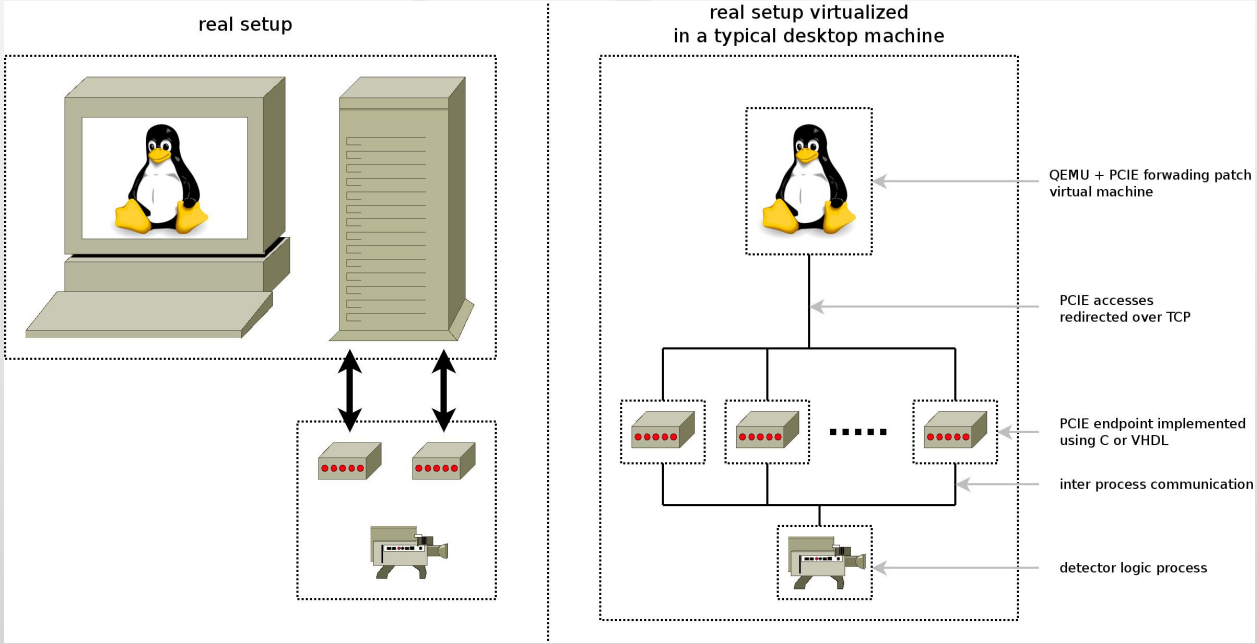
- http://ghdl.free.fr
- VHDL frontend for GCC
- used to implement device in VHDL

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - implementation



VPCIe - RASHPA possible setup



CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - RASHPA backend

RASHPA backend is a full featured LINUX system

- runs in a QEMU virtual machine
- PCIe accesses are trapped and sent over TCP to the devices
- PCIe forwarder is available as a QEMU patch
  - discussions in progress for a merge

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - RASHPA emulated devices

RASHPA emulated devices

- run as a LINUX processes, can be **duplicated** at will
- can be implemented in **C** or **VHDL**
- PCIe made **simple**, focus on device logic
  - but close to a true PCIe transaction layer (XILINX ...)

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - emulated device VHDL interface

```
entity endpoint is
 port
 (
  rst: in std_ulogic;
  clk: in std_ulogic;

  req_en: out std_ulogic;
  req_wr: out std_ulogic;
  req_bar: out std_ulogic_vector(pcie.BAR_WIDTH - 1 downto 0);
  req_addr: out std_ulogic_vector(pcie.ADDR_WIDTH - 1 downto 0);
  req_data: out std_ulogic_vector(pcie.DATA_WIDTH - 1 downto 0);

  rep_en: in std_ulogic;
  rep_data: in std_ulogic_vector(pcie.DATA_WIDTH - 1 downto 0);

  mwr_en: in std_ulogic;
  mwr_addr: in std_ulogic_vector(pcie.ADDR_WIDTH - 1 downto 0);
  mwr_data: in std_ulogic_vector(pcie.PAYLOAD_WIDTH - 1 downto 0);
  mwr_size: in std_ulogic_vector(pcie.SIZE_WIDTH - 1 downto 0);

  msi_en: in std_ulogic
 );
end entity;
```

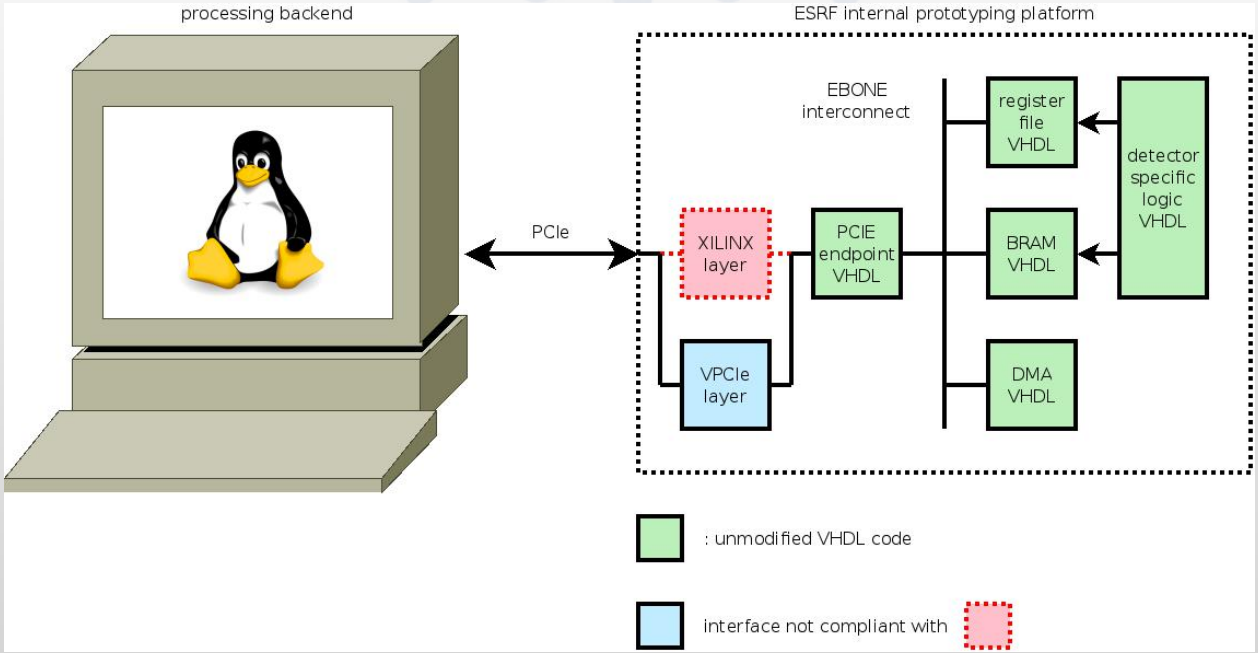
CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - emulated device C interface

```c
/* runtime initialization */
int pcie_init_net(pcie_dev_t*, ...);
int pcie_fini(pcie_dev_t*);
int pcie_loop(pcie_dev_t*);

/* misc config byte accessors */
int pcie_set_deviceid(pcie_dev_t*, ...);
int pcie_set_vendorid(pcie_dev_t*, ...);

/* PCIe BAR access handlers */
typedef void (*pcie_readfn_t)(uint64_t, void*, size_t, void*);
typedef void (*pcie_writefn_t)(uint64_t, const void*, size_t, void*);
int pcie_set_bar(pcie_dev_t*, ..., pcie_readfn_t, pcie_writefn_t, ...);

/* host memory read write operations */
int pcie_write_host_mem(pcie_dev_t*, uint64_t, size_t*);
int pcie_read_host_mem(pcie_dev_t*, uint64_t, size_t*);

/* send an MSI */
int pcie_send_msi(pcie_dev_t*);
```

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - tested to virtualize ESRF prototyping plateforms



CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - benefits

Hardware software codesign

- reduce development time
- no **modification** on the backend software (esp. driver)
- act as a stimuli generator for VHDL simulations

Platform scaling

- one PCIe endpoint per LINUX process

Investigate unavailable technologies

- NVM Express support for QEMU is available

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - more benefits

Virtual machine

- resources easily changed (RAM amount ...)
- test with different CPU architectures (x86, x86_64, ARM ...)
- small LINUX disk images reduce reboot times (few seconds)

Fault injection

- LINUX and software response to PCIe device disconnection

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - availability

VPCIe source is available online

- https://github.com/texane/vpcie
- documentation still poor, but clear examples
- feedbacks or contributions are welcome

CRISP
The Cluster of Research Infrastructures for Synergies in Physics

VPCIe - questions

Thanks for your attention.

Feel free to ask for a demo after the talk.

Any question?

CRISP
The Cluster of Research Infrastructures for Synergies in Physics