KVMFORNFV Design

Release brahmaputra.1.0 (2b76a9c)

OPNFV

August 22, 2016

CONTENTS

CHAPTER

ONE

INTRODUCTION

Purpose:

This document provides an overview of the areas that can be addressed to enhance the KVM Hypervisor for NFV. It is intended to capture and convey the significant changes which have been made on the KVM Hypervisor.

CHAPTER

TWO

PROJECT DESCRIPTION

The NFV hypervisors provide crucial functionality in the NFV Infrastructure(NFVI).The existing hypervisors, however, are not necessarily designed or targeted to meet the requirements for the NFVI.

This design focuses on the enhancement of following area for KVM Hypervisor

- **Minimal Interrupt latency variation for data plane VNFs:**
  - Minimal Timing Variation for Timing correctness of real-time VNFs
  - Minimal packet latency variation for data-plane VNFs
- Fast live migration

While these items require software development and/or specific hardware features there are also some adjustments that need to be made to system configuration information, like hardware, BIOS, OS, etc.

Minimal Interrupt latency variation for data plane VNFs

Processing performance and latency depend on a number of factors, including the CPUs (frequency, power management features, etc.), micro-architectural resources, the cache hierarchy and sizes, memory (and hierarchy, such as NUMA) and speed, inter-connects, I/O and I/O NUMA, devices, etc.

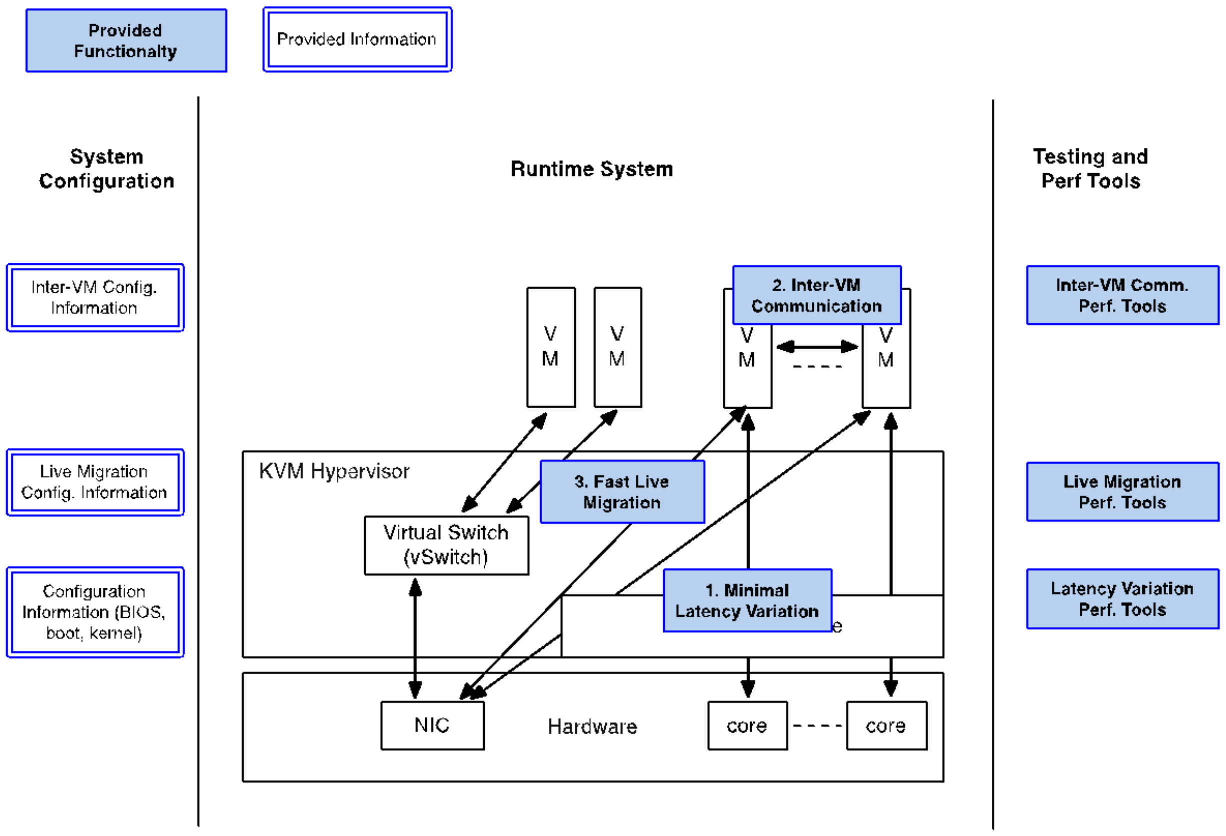
There are two separate types of latencies to minimize:

1. Minimal Timing Variation for Timing correctness of real-time VNFs – timing correctness for scheduling operations(such as Radio scheduling)
2. Minimal packet latency variation for data-plane VNFs – packet delay variation, which applies to packet processing.

For a VM, interrupt latency (time between arrival of H/W interrupt and invocation of the interrupt handler in the VM), for example, can be either of the above or both, depending on the type of the device. Interrupt latency with a (virtual) timer can cause timing correctness issues with real-time VNFs even if they only use polling for packet processing.

We assume that the VNFs are implemented properly to minimize interrupt latency variation within the VMs, but we have additional causes of latency variation on KVM:

- Asynchronous (e.g. external interrupts) and synchronous(e.g. instructions) VM exits and handling in KVM (and kernel routines called), which may have loops and spin locks
- Interrupt handling in the host Linux and KVM, scheduling and virtual interrupt delivery to VNFs
- Potential VM exit (e.g. EOI) in the interrupt service routines in VNFs
- Exit to the user-level (e.g. QEMU)


Provided Functionalty
Provided Information
System Configuration
Runtime System
Testing and Perf Tools
Inter-VM Config. Information
Live Migration Config. Information
Configuration Information (BIOS, boot, kernel)
VM
VM
VM
VM
2. Inter-VM Communication
KVM Hypervisor
3. Fast Live Migration
Virtual Switch (vSwitch)
1. Minimal Latency Variation
NIC
Hardware
core
core
Inter-VM Comm. Perf. Tools
Live Migration Perf. Tools
Latency Variation Perf. Tools

CHAPTER THREE

DESIGN CONSIDERATIONS

The latency variation and jitters can be minimized with the below steps (with some in parallel):

1. Statically and exclusively assign hardware resources (CPUs, memory, caches,) to the VNFs.
2. Pre-allocate huge pages (e.g. 1 GB/2MB pages) and guest-to-host mapping, e.g. EPT (Extended Page Table) page tables, to minimize or mitigate latency from misses in caches,
3. Use the host Linux configured for hard real-time and packet latency, Check the set of virtual devices used by the VMs to optimize or eliminate virtualization overhead if applicable
4. Use advanced hardware virtualization features that can reduce or eliminate VM exits, if present, and
5. Inspect the code paths in KVM and associated kernel services to eliminate code that can cause latencies (e.g. loops and spin locks).
6. Measure latencies intensively. We leverage the existing testing methods. OSADL, for example, defines industry tests for timing correctness.

CHAPTER

FOUR

GOALS AND GUIDELINES

The output of this project will provide :

1. A list of the performance goals, which will be obtained by the OPNFV members (as described above)
2. A set of comprehensive instructions for the system configurations (hardware features, BIOS setup, kernel parameters, VM configuration, options to QEMU/KVM, etc.)
3. The above features to the upstream of Linux, the real-time patch set, KVM, QEMU, libvirt, and
4. Performance and interrupt latency measurement tools

CHAPTER FIVE

TEST PLAN

The tests that need to be conducted to make sure that all components from OPNFV meet the requirement are mentioned below:

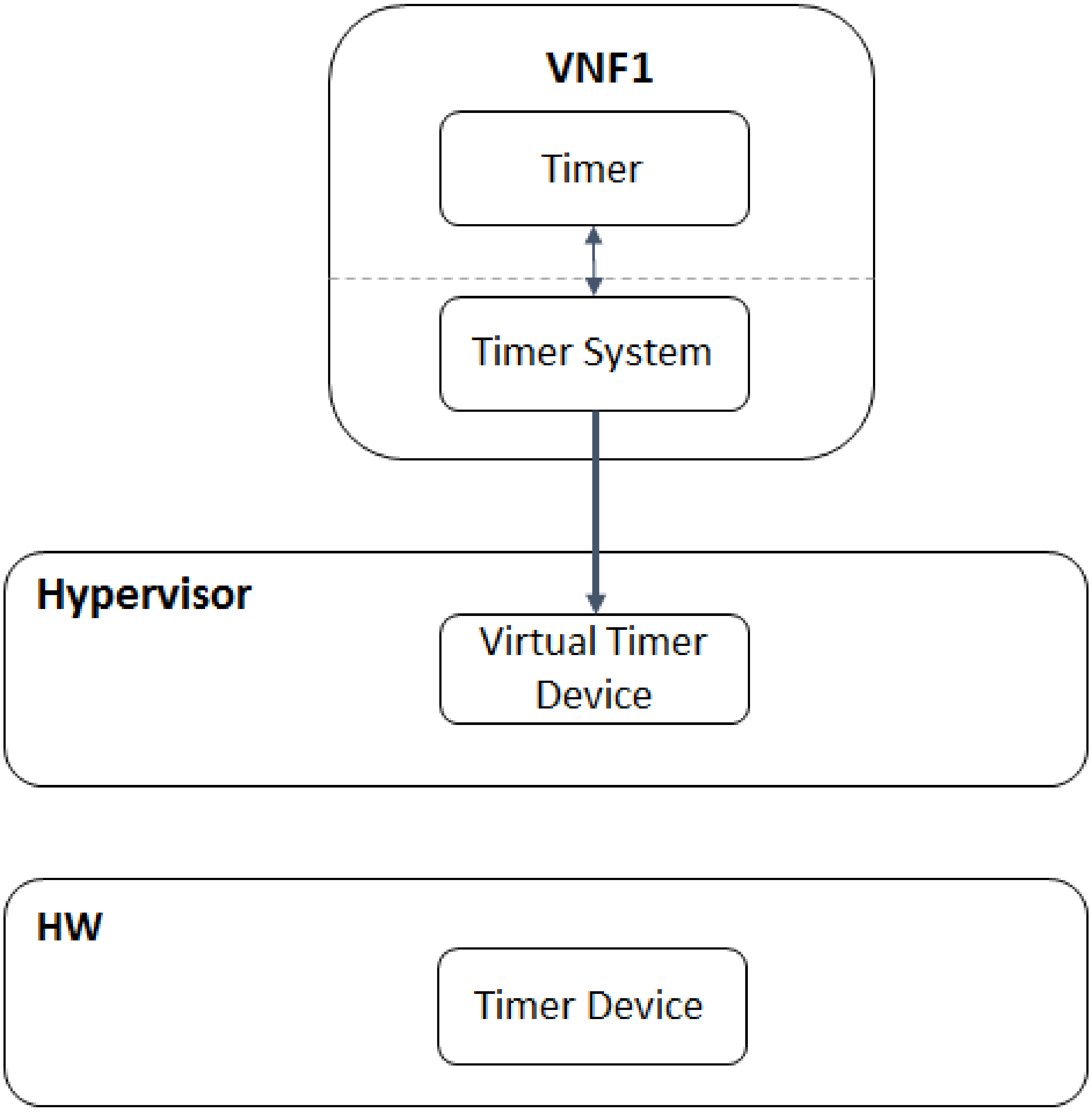
Timer test:This test utilize the cyclictest (https://rt.wiki.kernel.org/index.php/Cyclictest) to test the guest timer latency (the latency from the time that the guest timer should be triggered to the time the guest timer is really triggered).

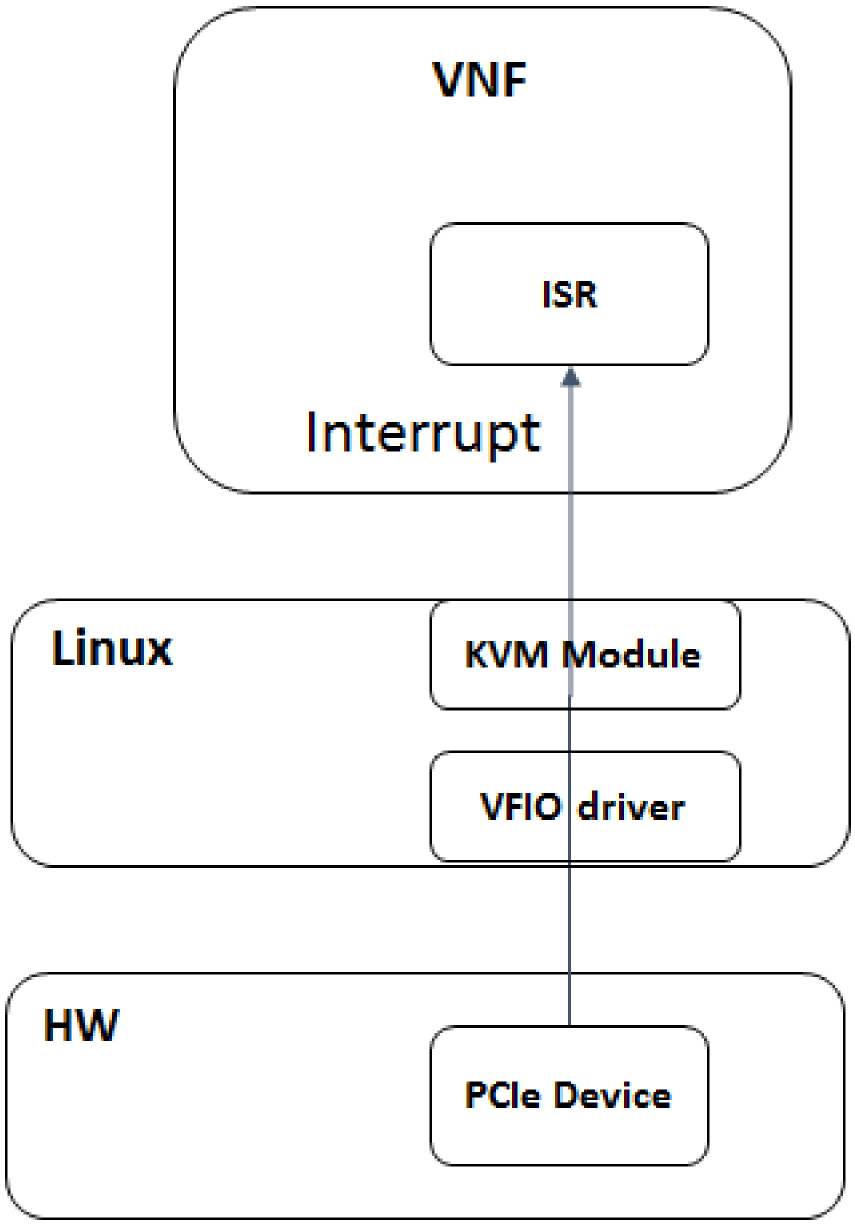
Device Interrupt Test:A device on the hardware platform trigger interrupt every one ms and the device interrupt will be delivered to the VNF. This test cover the latency from the interrupt happened on the hardware to the time the interrupt handled in the VNF.

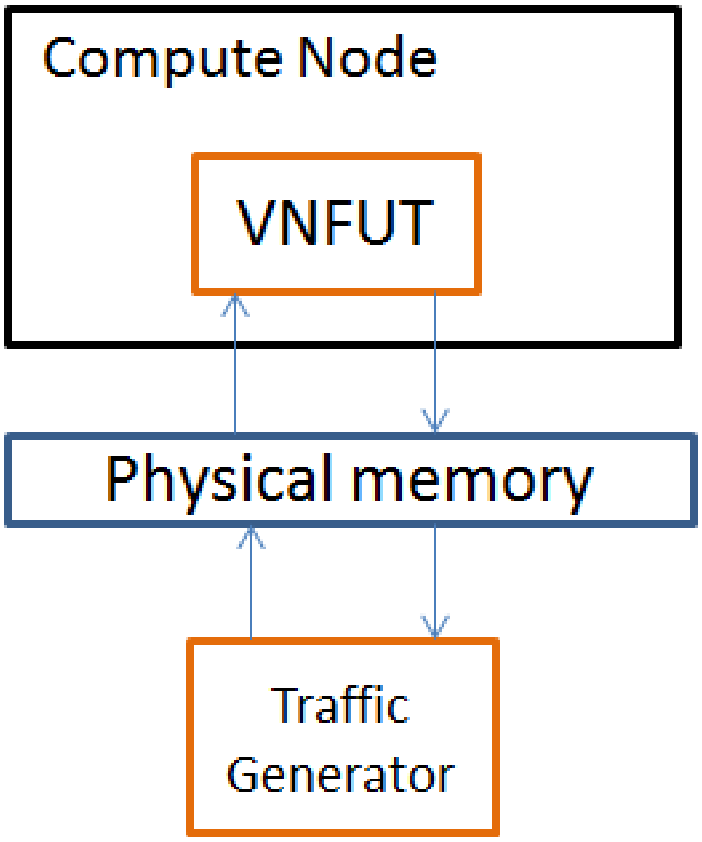
Packet forwarding (DPDK OVS):A packet is sent from TC (Traffic Generator) to a VNF. The VNF, after processing the packet, forwards the packet to another NIC and in the end the packet is received by the traffic generator. The test check the latency from the packet is sent out by the TC to the time the packet is received by the TC.

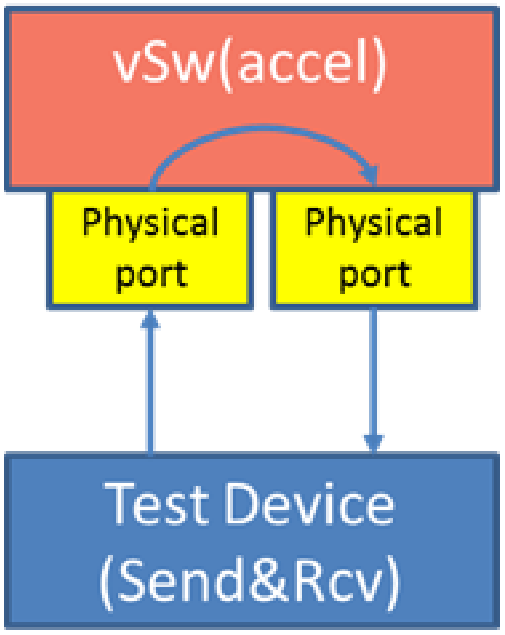
Packet Forwarding (SR-IOV):This test is similar to Packet Forwarding (DPDK OVS). However, instead of using virtio NIC devices on the guest, a PCI NIC or a PCI VF NIC is assigned to the guest for network acess.

Bare-metal Packet Forwarding:This is used to compare with the above packet forwarding scenario.


VNF1
Timer
Timer System
Hypervisor
Virtual Timer Device
HW
Timer Device


VNF
ISR
Interrupt
Linux
KVM Module
VFIO driver
HW
PCIe Device


Compute Node
VNFUT
Physical memory
Traffic Generator


vSw(accel)
Physical port
Physical port
Test Device (Send&Rcv)

CHAPTER SIX

REFERENCE

https://wiki.opnfv.org/display/kvm/