MontaVista

Solution Brief

User-Space Network I/O : Taking Data Path on Hyper-drive

Scalable network performance in a virtualized environment

Next generation networks are characterized by a much higher native flexibility and programmability for all non - radio network segments including SDN, NFV and IoT Networks. Multicore processing and virtualization are rapidly becoming ubiquitous in software development.

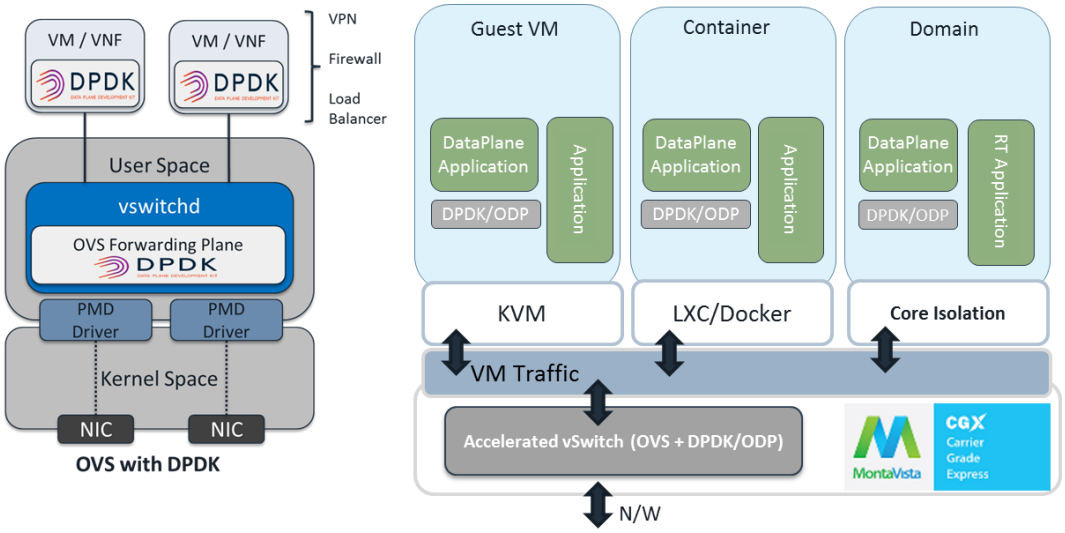


Fig 1: Virtualization & Fast Path Solution with MontaVista™ CGX

MontaVista continues to participate in the way networks are created and behave by providing necessary software, tools and support to help reduce Capex and improve Opex. This is being achieved by transitioning from Physical Network Function (PNF) (i.e. on a single purpose hardware & software platform) to a Virtual Network Function (VNF) that is deployable on a wide variety of general purpose hardware and software combination.

An additional requirement for the success of this flexible and scalable solution is to speed up network I/O performance.

Data Plane Development Kit (DPDK)

- Software based accelerated packet processing
- Contains libraries and drivers for select NICs

Open Virtual Switch (OVS)

- Enables network automation with programmability
- Used in virtualized server, SDN & NFV use-cases

BENEFITS (MontaVista Dataplane Profile)

- OUT OF BOX EXPERINECE WITH PRE-TESTED BSPs
- REAL TIME LINUX + Virtualization (KVM + Containers)
- Multi-Architecture & HW/FPGA Optimization
- BUILT IN FLEXIBILITY, RELIABILITY AND SECURITY
- SECURE LIVE KERNEL AND APPLICATION UPDATE

APPLICATIONS / USE CASES

- 5G CARRIER GRADE INFRASTRUCTURE
- DELIVER NFV & SDN WITH SERVICE CHAINING
- TELCO CLOUD & vROUTER

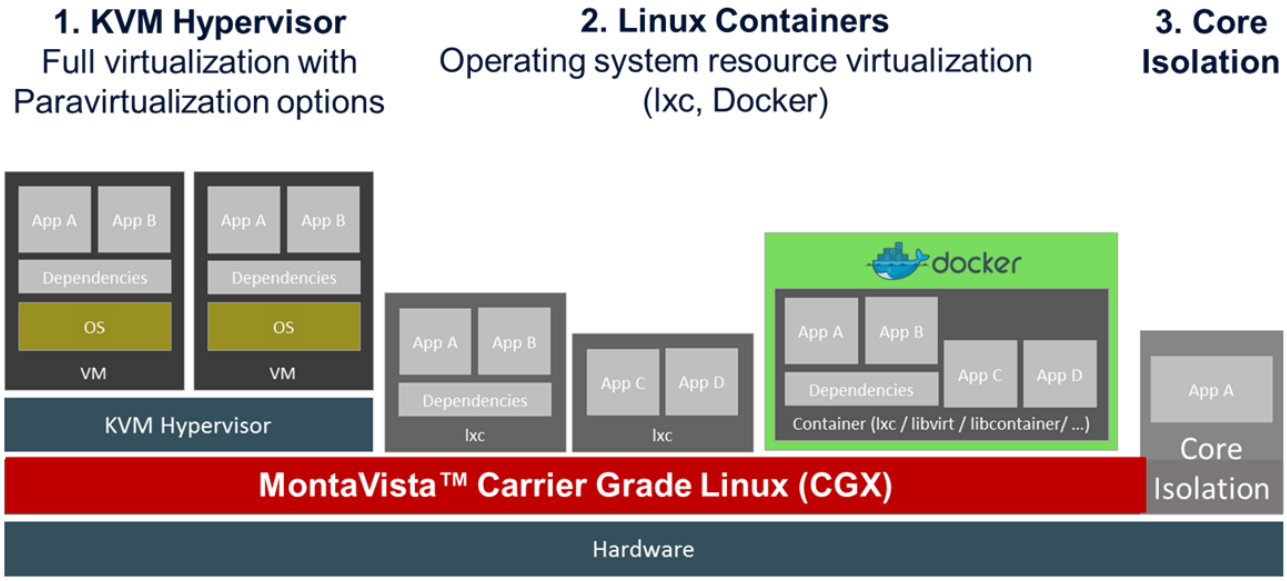


Fig 2: Virtualization solution options with MontaVista™ CGX

Next Generation Networks

Network data centers are undergoing major transformations by introducing virtual network devices to provide the agility and efficiency required today. Until recently, deployment of new services was done with purpose built telecommunications equipment. However, telecommunication service provides are demanding for higher agility with better costs to keep up with the rapid expansion in the user base, the increasing services offerings along with the rate technology innovation.

Multicore processing and virtualization are rapidly becoming ubiquitous in software development. They are widely used in the commercial world, to

- Reduce CAPEX by Isolating application software from hardware and operating systems, enabling different applications to share underutilized computers or processors.
- Improve OPEX through system scalability at a fraction of time and cost along with high reliability and robustness by limiting fault and failure propagation and support failover and recovery.

Network Performance (Throughput)

Network performance can be best understood in terms of cost per packet i.e. both CAPEX & OPEX needed to deliver packet to its destination. The promises of cost reductions from virtualizing a network can only be realized if packet throughput and I/O response times can be achieved and predicted. Linear scalability in performance by the CPU core in a virtualization server environment can be unpredictable. This is because generic Operating systems are not optimized for networking in a virtual environment.

However, it is possible to deliver hardware line rates in a virtualized environment through the use of certain architectures and development kits like the Data Plane Development Kit (DPDK) and Open Data Plane (ODP).

MontaVista is able to provide the optimized architecture used to run a high performance, optimized product. There is an increasing trend to use such user space access to network I/O in host and guest, as a popular way to work-around the native performance challenges of the OS networking stack.

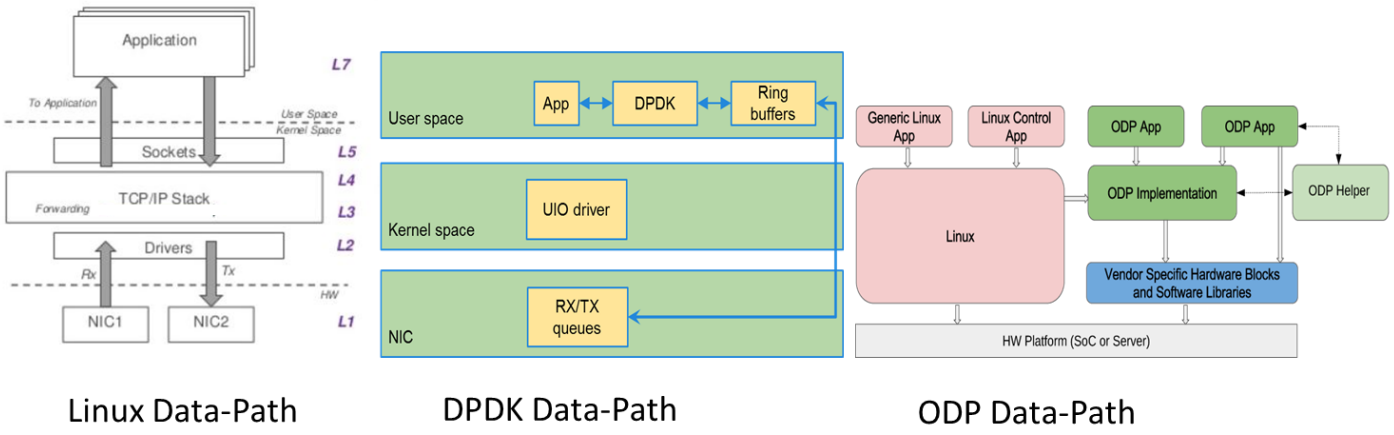


Fig 3: Understanding Data-path & Performance techniques

Let's explore some of these methods,

Packet Processing in Standard Linux

Linux Kernel data path is designed as an internet host that receives direct access to hardware drivers. However, it is very generic and not optimized for forwarding use case. Here is a quick flow for ease of understanding

- Packet arrives on the NIC RX queue.
- DMA copies the packet to a DMA-able memory without CPU intervention
- At this point an interrupt is generated to notify the system that a packet needs to be processed
- Now the driver copies the packet from DMA-able region to the kernel packet buffer
- Later the network stack takes care of passing the received packet to the application via socket interface which involves copying data from kernel space to user space

Packet Processing Architecture

- Single threaded architecture (standard OS)

A standard networking stack uses services provided by the Operating System (OS) running on a single processor (single threaded). This model adds overheads associated with the performance of OS functions such as preemptions, thread management, timers and locking.

- Multi-threaded architecture (multi-processing OS)

The protocol stack processing software supports multiple processors (multi-threaded), either through the use of Symmetrical Multiprocessing (SMP) platforms or multi-core processor architecture. Performance increases are realized for a small number of processors, but fails to scale linearly over larger numbers of processors.

- Fast path architecture (OS by-pass)

In a fast path implementation, the data plane is split into two layers. The lower layer, typically called the fast path, processes the majority of incoming packets outside the OS environment and without incurring any of the OS overheads that degrade overall performance. Only those packets that require complex processing are forwarded to the OS networking stack (the upper layer of the data plane), which performs the necessary management, signaling and control functions.

Accelerated Packet Processing Approach

Special Hardware - Network processors (NPU), Multicore processors with dedicated hardware accelerators for packet processing like crypto-engines, pattern matching engines, hardware queues for QoS, etc.

Software - specialized packet processing like DPDK.

MontaVista

Solution Brief

Data Plane Development Kit (DPDK)

DPDK is a set of libraries and drivers for fast packet processing. It runs mostly in Linux userland. The main libraries are:

- multicore framework
- huge page memory
- ring buffers
- poll-mode drivers for networking, crypto and eventdev

DPDK utilizes the following techniques to achieve the maximum throughput and minimal time for processing packets

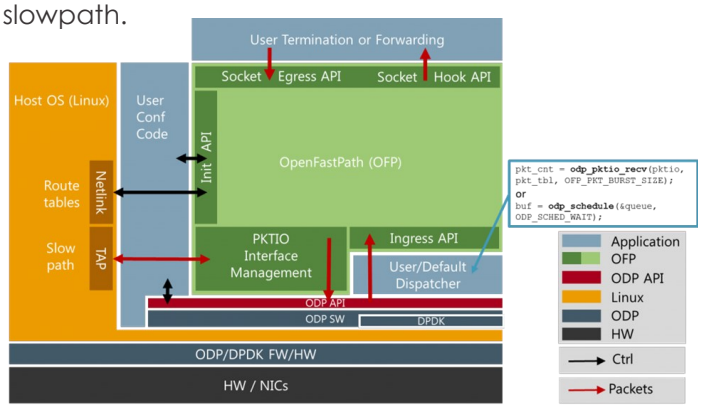
- DMA directly to user-space shared memory
- Polling instead of handling interrupts for each arrived packet;
- SSE instruction set to copy big amount of data effectively;
- Hugepages to decrease the size of TLB that results in a much faster virtual to physical page conversion;
- Thread affinity to bind threads to a specific core to improve cache utilization;
- Lock-free user-space multi-core synchronization using rings;
- NUMA awareness to avoid expensive data transfers between sockets;

Open Data Plane (ODP)

The ODP project has been established to produce an open-source, cross-platform set of application programming interfaces (APIs) for the networking data plane. ODP consists of an API specification and a set of reference implementations that realize these APIs on different platforms. Implementations range from pure software to those that deeply exploit the various hardware acceleration and offload features found on modern networking system-on-Chip (SoC) processors. ODP's goal is to allow implementers of the API great flexibility to exploit and optimize the implementation. This is intended to enable easy platform portability such that an application written to the API can pick up performance gains without needing significant platform knowledge when ported. An ODP application runs as a Linux user space process but makes very limited calls to Linux APIs. Instead it uses ODP APIs (and possibly SDK APIs) to enable accelerated support of underlying hardware features without incurring kernel overhead.

Open Fast Path (OFP)

OpenFastPath is an open source implementation of a high performance TCP/IP stack that provides features that network application developers need to cope with today's fast-paced network. OFP enables accelerated routing/forwarding for IPv4 and IPv6, tunneling and termination for a variety of protocols. Unsupported functionality is provided by the host OS networking stack or slowpath.



(Source: http://www.openfastpath.org/index.php/service/technicaloverview/)

Fig 4: OFP Arch. Overview

FD.io (Fast data - Input/Output)

FD.io is a software-based packet processing technology geared towards the creation of high-throughput, low-latency and resource-efficient IO services suitable to many architectures (x86, ARM, and PowerPC) and deployment environments (bare metal, VM, container).

- Vector Packet Processing (VPP) library is highly modular, allowing for new graph nodes to be easily "plugged in" without changes to the underlying code base making it highly scalable solution.
- Along with VPP, FD-io leverages DPDK capabilities in support of additional projects including NSH_ SFC, Honeycomb, and ONE to accelerate the NFV data planes.

MontaVista

Solution Brief

Software Fast Path in VM with DPDK : Packet Performance

Test setup to measure packet performance improvement

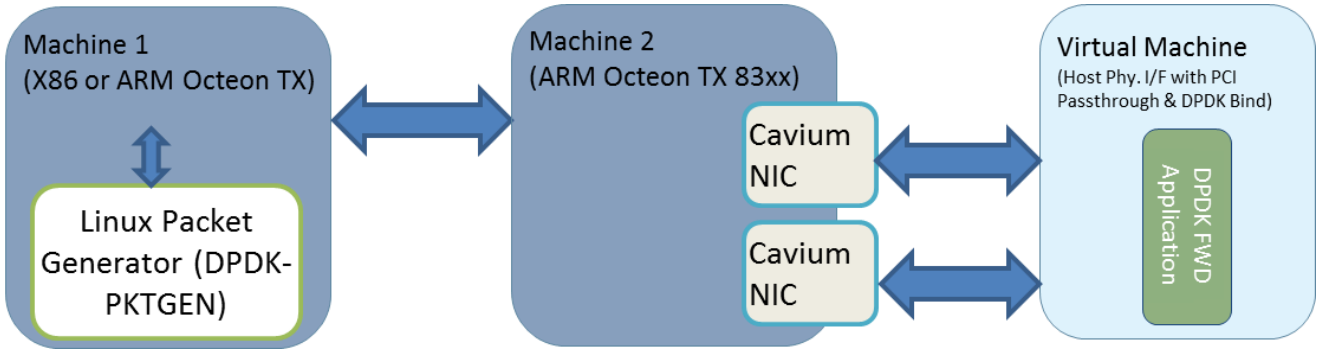


Fig 5: Case Study: Guest VM fast path packet performance

Measuring Packet Performance for Guest VM

The setup shown in Fig 5 shows a ThunderX NICVF PMD (librte_pmd_thunderx_nicvf) provides poll mode driver support for the inbuilt NIC found in the Cavium ThunderX SoC family as well as their virtual functions (VF) in SR-IOV context.

- ThunderX NIC PF/VF kernel modules maps each physical Ethernet port automatically to virtual function (VF) and presented them as PCIe-like SR-IOV device

Pass VF device to VM context (PCIe Passthrough):

- The VF devices may be passed through to the guest VM using qemu or virt-manager or virsh etc.
- Vhost is a kernel acceleration module for virtio qemu backend
- The DPDK extends KNI to support vhost raw socket interface, which enables vhost to directly read/ write packets from/to a physical port.

Performance Measurement

Launch the forwarding application using ports which are binded. As Fig 9 shows we then use the flow—

packet generator->Guest VM DPDK port ->Guest VM DPDK port->packet generator

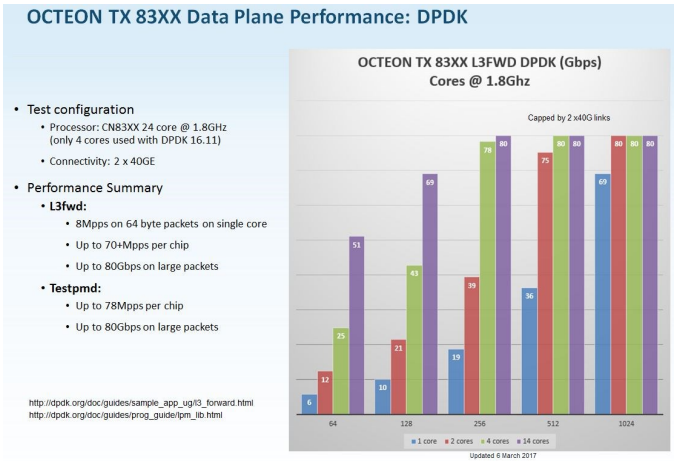


Fig 6: Data path performance (Octeon TX 83xx)

MontaVista

Solution Brief

MontaVista Carrier Grade eXpress (CGX), meets the demands of the interconnected intelligent devices, providing application portability, dynamic configuration, field maintenance, and real-time performance in a single platform that is optimized for 5G, NFV Applications and Internet of Things (IoT) Devices.

eXpress.Connected.Everything.

About MontaVista Software

MontaVista Software, LLC, a wholly owned subsidiary of Cavium Networks (NASDAQ:CAVM) is a leader in embedded Linux commercialization. For over 15 years, MontaVista has been helping embedded developers get the most out of open source by adding commercial quality, integration, hardware en-ablement, expert support, and the resources of the MontaVista development community.

MontaVista All Ways Open

MontaVista Software
2315 North First St, 4th FL
San Jose, CA 95131
Email: sales@mvista.com
Tel: +1-408-943-7451

© 2018 MontaVista Software, Inc. All rights reserved.
LiquidIO, LiquidSecurity, OCTEON Fusion-M, OCTEON TX, ThunderX, Xpliant are trademarks of Cavium, Inc. Linux is a registered trademark of Linus Torvalds. MontaVista is a registered trademark of MontaVista Software, LLC. All other names mentioned are trademarks, registered trademarks or service marks of their respective companies. MVSB11