# Accelerated Data Plane for NFV

*Achieve a 200% VM Density Increase With Low Latency and 50% Better CPU utilization*

*using*

*IBM's IOBricks Engine, Accelerated by Silicom*

## Overview

***IBM's IOBricks Engine*** is a complete programmable Network Function Virtualization (**NFV**) and service chaining infrastructure solution that **redefines the data plane**. Designed using a bottom-up approach, *IBM's IOBricks Engine* employs an innovative new connectivity paradigm to solve many of the challenges created by NFV deployments: impaired VM performance, impaired service chaining, network bottlenecks and/or software switching mechanism inefficiencies that are caused by deficiencies in common data planes (OpenVSwitch-**OVS**, Vector Packet Processing-**VPP**, etc.)

The tight integration of *IBM's IOBricks Engine* with Silicom's **Intel® FM10K**-based network adapters (*PE3100G2DQIRXXX* series) brings a further performance improvement to this unique software solution.

*For more information about IBM, please visit* [https://www.ibm.com](https://www.ibm.com)

## Benefits for NFV

When deploying NFV, the most important considerations are Virtual Machine (VM) density and service chaining/provisioning efficiency. On the one hand, services running as Virtual Network Functions (**VNF**) need enough CPU power to enable services to be provisioned successfully. On the other hand, the denser, the better: the more VNFs that can run on a server, the more economical it becomes to embrace the NFV model. **Therefore, an ideal solution would support dense provisioning with optimal CPU utilization.** ***This is exactly the approach taken by IBM's IOBricks Engine.***

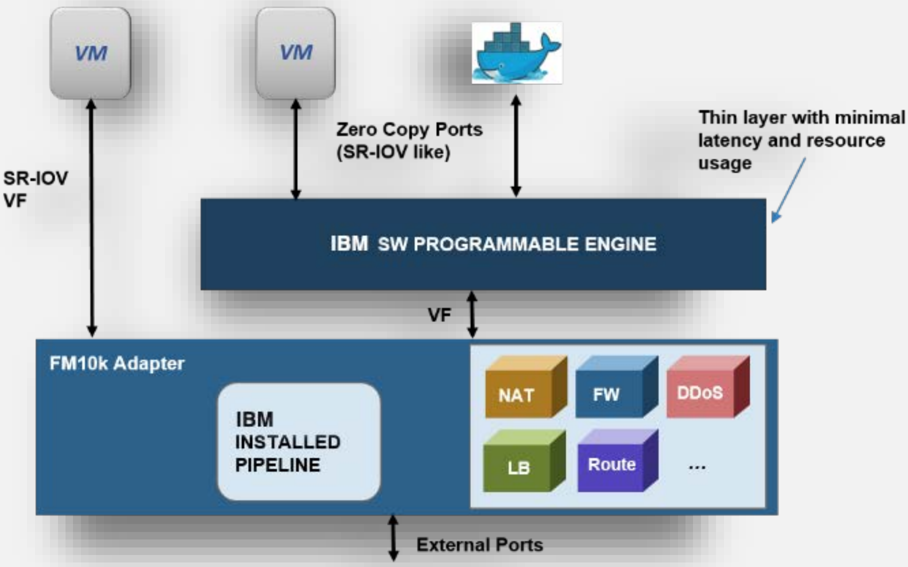


*Figure 1 - IBM's IOBricks Engine*

Beyond dense provisioning and optimal CPU utilization, IBM's IOBricks Engine delivers the following additional benefits:

- Accelerated performance due to the solution's integration with the Silicom Intel®-based FM10K PE3100G2DQIRXXX adapter
- Increased VM capacity per server
- High-bandwidth with low latency network infrastructure
- Reduced latency
- VMs and containers connected directly to the fabric
- <u>**High bandwidth (up to 200GB) support for each server without oversubscription**</u>
- Innovative new fabric design that allows implementation of new connectivity policies and programming of the physical network in addition to the virtual network

- **Efficient service chaining** – enables service chaining and networking services to be built for execution by the FM10K Intel® hardware, reducing server CPU load by >50%
- Full virtualized network services including virtual networking, virtual routing, virtual security and load balancing, NAT, etc. to SR-IOV based VMs

## Performance in Action: Benchmark Testing of IBM's IOBricks Engine

Typically, pure software data planes consume considerable amounts of CPU power, and when operating in a context of a hypervisor, compete against compute VMs for precious CPU cycles. Even user-space data planes such as DPDK OVS or VP, which can be confined to a specific mask of CPU cores, require an increased number of cores as bandwidth increases. The ratio of CPU power allotted to a busy data plane can reach 30%-35%.

The first-order issue stemming from this reality is, of course, reduced availability of CPU power for compute processing. With less CPU power available, it is impossible to achieve desirable VM densities, especially as bandwidth increases. At the same time, for busy compute nodes, the lack of CPU power for networking tasks limits the bandwidth to and from the VMs.

The second-order issue stemming from this reality is that, since both data plane forwarding software and compute VM software are governed by a shared software scheduling mechanism, the true performance of typical hypervisors falls far below the ideal 100% real-time basis. As a result, traffic fluctuations and high traffic rates create severe disturbances to SLA-sensitive applications within VMs.

A test of the effectiveness of the IBM's IOBricks Engine, then, would involve evaluating the following metrics with and without the use of the engine:

1) ***VM density*** – total number of VM instances successfully operating on a server
2) ***CPU utilization*** – relative ratio of server CPU cycles translated into the number of physical CPU cores available for compute task processing
3) ***Traffic latency*** – measurement of absolute time required by the packet forwarding infrastructure to transmit packets and traffic to the destination, i.e., the processing VM.

The benchmark test system included a Dual Xeon E5 2690 v4 server in which each CPU socket was equipped with 14 physical cores, OVS-DPDK, and a VPP. These were installed as a data plane for a KVM hypervisor.

Rather than simple layer 2 forwarding, a set of services were configured on the hypervisor, including (a) a set of virtual routing configurations; (b) stateful security groups for VMs; and (c) tunneling (encap/decap) service; all in order to simulate actual busy setup.

Traffic was injected into this system for forwarding to, from and across VMs, while bandwidth and VMs number were gradually increased across the combined 28 physical cores of both CPU sockets. Measurements were taken with the goal of answering the following questions:

- What CPU core count would be required to support 100Gpbs traffic forwarding?
- What is the Gbps forwarding rate of a single CPU core?
- With 100Gpbs forwarded traffic, how many CPU physical cores are left for compute?
- What is the latency and what affects it?

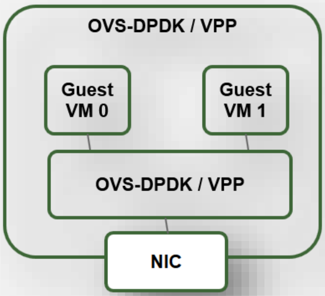


*Figure 2 - Software Data Plane*

As seen in Table 1, the results pre-IOBricks Engine were as expected. To achieve 100Gbps forwarding, 14 CPU cores were needed, leaving space for 56 VMs (assuming 4 VMs per physical core) running on only 14 CPU available cores. The latency reading was considerable as well.

Then, IBM's IOBricks Engine programmable data plane was introduced into the benchmark setup, and the tests were repeated.

As seen in Table 2, ***a vast improvement was immediately visible across all parameters.*** In fact, the improvement was so significant that the entire test was repeated to verify its integrity.

| Data Plane Overhead | Number of Cores required for 100Gbps | Number of cores available for VMs | Number of VMs (4 VMs per core) | Latency |
|---|---|---|---|---|
| **7.6Gbps per core** | 14 | 14 | 56 | 33us |

*Table 1 - Software Data Plane Performance*

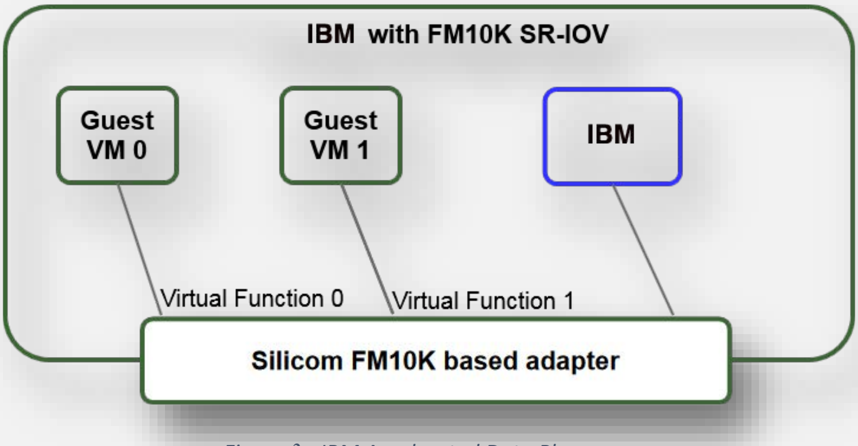

*Figure 3 - IBM Accelerated Data Plane*

When IBM's IOBricks Engine with the Silicom Intel® based FM10K adapter was used, only one (!) CPU core was required to reach 100Gbps forwarding. Moreover, the number of physical cores available for compute VM surged to 108 VMs, a ~200% improvement.

| Data Plane Overhead | Number of Cores required for 100Gbps | Number of cores available for VMs | Number of VMs (4 VMs per core) | Latency |
|---|---|---|---|---|
| **1 core** | 1 | 27 | 108 | ~0us |

*Table 2 - IBM Data Plane Performance*

Latency measurements were also harbingers of extremely good news.

**Much lower latency was measured**. This parameter alone enables us to predict that more coherent and predictable scheduling could be administered across the system as a whole, allowing for SLA-sensitive services to be run as VMs.

As seen from the block diagram in Figure 2, SRIOV came into play as well. Unlike any other software data plane, the IBM data plane is built to take advantage of SRIOV without sacrificing desirable features related to a virtual software-based data path.

## How does it work?

- **IOBricks Engine:** provides modules for most common networking services
- **Silicom Adapter:** provides a seamless smart offload

Most traffic processing tasks are offloaded onto the FM10K Intel® network adapter, freeing up the CPU for Virtual Network Functions (**VNF**). The use of the Intel® FM10K chip provides a higher degree of SLA accuracy and QoS to the provisioned infrastructure.

By enabling a distributed switching fabric *(described in a separate paper),* the solution can dedicate unique traffic pipes to specific traffic and guarantee delivery of the correct SLA set called for by the user policy.

IBM's IOBricks Engine supports a number of deployment scenarios. VMs and containers can be connected directly to the NIC (leveraging SR-IOV) or to the engine itself via zero copy ports. Scenarios can include built-in services (see Figure 1).

## Additional Features

- Modular approach to networking services:
  - Routing
  - Load balancing
  - QinQ tunneling
  - Security (Firewall, DPI, DDoS protection)
- Provides advanced stateful network services with traffic engineering tools
- Enables full programmability into the data plane
- Complete infrastructure for data center connectivity with no need for additional networking equipment (switches, routers, etc.)
- Flexible design: highly adaptable to all data center architectures
- Programmable design: users can create the model that suits them best using common programing languages, with existing modules connected via a user-defined policy graph

## Applications

IBM's IOBricks Engine includes common telecom-oriented modules that are highly useful for deployment in NFV clusters, like MPLS, PPPoE, QinQ, NSH and BGP routing support.

This new inter-connected switchless cluster enables an extremely rapid and efficient innovation cycle, since the network becomes uniform and programmable.

Autoscaling as well as efficient service chaining can be achieved while reducing cost, power consumption, and latency – while still providing 200G of bandwidth to each server.

## SUMMARY

- Provisioning network services close to the edge, as required in NFV deployments, can be leveraged by cloud technology.
- CPU power is often wasted on forwarding tasks.
- Increasing VM density in servers can lead to poor ROI, raising questions about the total benefit achieved.

The coupling of IBM's IOBricks Engine with Silicom's FM10K Intel®-based adapter as a networking front-end accelerator, solves all of these issues and more, enabling a new level of performance to be achieved in Cloud network services.

Silicom Ltd., 14 Atir Yeda St., Kfar Sava, ISRAEL
sales@silicom.co.il www.silicom-usa.com