Poster: Towards In-Network Resource Scaling of VNFs

Francisco Germano Vogt
Universidade Estadual de Campinas (Unicamp)
Campinas, Brazil

Fabricio Rodriguez
Universidade Estadual de Campinas (Unicamp)
Campinas, Brazil

Marcelo Caggiani Luizelli
Federal University of Pampa (Unipampa)
Alegrete, Brazil

Christian Esteve Rothenberg
Universidade Estadual de Campinas (Unicamp)
Campinas, Brazil

ABSTRACT

Intra-server resource orchestration is becoming increasingly challenging. Modern server architectures are now composed of multiple computing units, such as CPUs and DPUs. VNFs should be appropriately provisioned between these computing units, avoiding performance degradation and optimizing server resources. In this work, we introduce In-Network Resource Allocation (InReal), a system to orchestrate containerized VNFs. InReal can provision and manage VNF resources, optimizing CPU usage and enhancing power efficiency while meeting VNF performance requirements.

1 INTRODUCTION

To address the growing computing demands [3], disaggregated computing units like SmartNICs are being deployed in cloud servers. These specialized units offload specific tasks from the main processor, enhancing overall system performance and efficiency. Orchestrating resources within a server (i.e., *intra-server orchestration*) is becoming increasingly challenging. Modern servers now resemble *distributed systems*, composed of multiple computing units such as host CPUs, SmartNIC CPUs (e.g., ARM in SoC), and SmartNIC ASICs. Flow packets must be accurately directed to the computing unit where the corresponding application resides. To keep tail latency within the microsecond scale, intra-server orchestration encompasses three critical tasks: (*i*) request scheduling, (*ii*) load balancing, and (*iii*) core assignment [1].

In the context of Virtual Network Function (VNF) overload [2], an orchestrator is responsible for scaling up by allocating additional CPU quota or scaling out by spawning additional instances and redistributing the load. The orchestrator also manages timely scale down and scale in of VNFs to optimize resources. Current cloud orchestrators, such as Kubernetes (K8s), generally operate with a
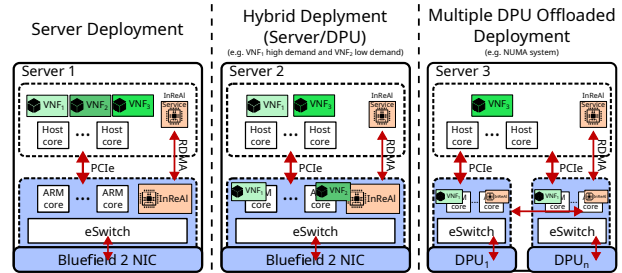
Figure 1: VNFs deployment options.

coarse-grained management cycle. For example, the K8s Horizontal Pod Autoscaler (HPA) typically makes orchestration decisions every 15 seconds. This interval may be insufficient for applications needing responsive scaling, causing over- or under-provisioning and affecting performance and efficiency.

Figure 1 illustrates different VNF deployment options. In a conventional setup, the application runs entirely on the host server. Alternatively, the VNF can be split between the host and Smart-NIC or fully offloaded to the SmartNIC, where the ARM cores of the Nvidia BlueField handle all processing. The choice depends on the application's needs, allowing for optimized resource use and performance tuning.

2 IN-NETWORK RESOURCE ALLOCATION

Offloading intra-server orchestration to SmartNICs allows for precise monitoring of metrics like packet drops and inter-packet latency, enabling responsive orchestration and real-time network management. SmartNICs can optimize traffic flows and resource allocation, quickly detecting and addressing network issues. However, offloading orchestration to SmartNICs poses several challenges. Firstly, ARM cores are relatively low-performance, requiring orchestration to be simple and lightweight. Secondly, VNF metrics like CPU and memory usage must be shared between the host and SmartNIC to ensure efficient resource management. Recent related work includes RingLeader [1] and Horus [3], both of which focus on offloading task orchestration to SmartNICs. Others [2] have scaled up VNFs by intra-server parallelization.

This work presents a preliminary design and experimental insights of **InReal** (**In-Network Resource Allocation**), a system that offloads the orchestration of containerized VNFs to the Smart-NIC. InReal decides *where*, *when*, and *how* VNFs are deployed in a server. InReal maintains both network flow and VNF statistics within the data plane, enabling precise and efficient resource management. For example, it monitors inter-packet latency, and this data, combined with VNF metrics like CPU and memory usage,

(a) Power Consumption.   (b) Throughput.   (c) Throughput varying CPU quota (CPU scaling up/down) of VNFs.   (d) Power Consumption varying CPU quota (CPU scaling up/down) of VNFs.
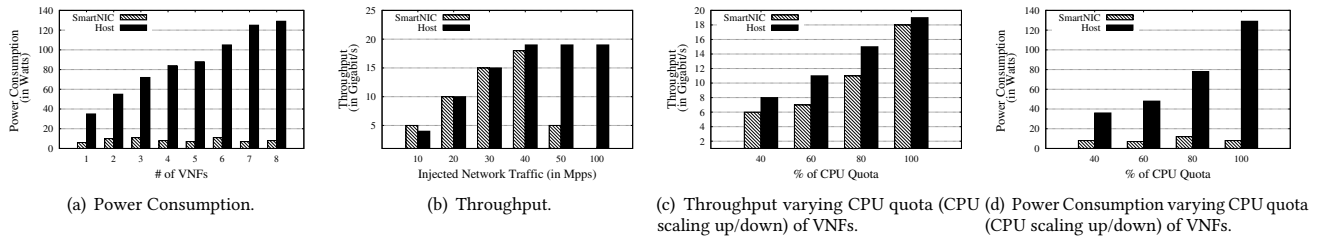
Figure 2: Power consumption and throughput of scaling VNFs in/out and up/down

guides InReal's orchestration of server resources. To facilitate the sharing of VNF metrics with the primary SmartNIC, InReal employs RDMA or DMA reads and writes. This approach minimizes the use of SmartNIC CPU cycles for handling VNF statistics, ensuring that orchestration tasks do not hinder SmartNIC performance. To handle traffic at line rate, InReal dynamically monitors VNF network traffic, adjusting its resource allocation strategies in real-time to maintain optimal performance. Based on the observed metrics, InReal decides to scale in/out or scale up/down VNFs. Fundamentally, the decision is made to increase VNF performance or reduce the resource consumption of host CPUs.

3 PROTOTYPE & INITIAL EVALUATION

Our initial prototype implements straightforward orchestration logic to explore Data Processing Units' (DPUs) limitations and identify scenarios where InReal can benefit from scaling VNFs either in/out or up/down on both CPU and DPU sides. To achieve this, we developed a simple strawman VNF with scaling capabilities and evaluated throughput and energy consumption.

Testbed. Our setup has two servers connected by a Tofino Switch via 100G DAC cables. The Device Under Test (DUT) server has two BlueField-2 SmartNICs, an i9-13900K processor (base 3.0 GHz, boost up to 5.80 GHz), and 128GB of RAM. HyperThreading and energy-efficient cores are disabled, providing eight high-performance cores with DVFS enabled and set to performance mode. The BlueField SoC contains eight ARM cores at a fixed 2.0 GHz (DVFS is unavailable). The second server generates traffic using TRex, sending 64-byte UDP packets at rates from 10 to 100 Mpps.

VNFs. We developed Docker containers to run on either the DUT or DPU, each with two ports: Scalable Functions on the DPU and Virtual Functions on the DUT. Each container hosts a Data Plane Development Kit (DPDK) app that copies packets between RX and TX ports. Containers process packets within specific IP subnets, with traffic managed by the e-Switch, which, based on the configuration, forwards it to the DPU or the host.

Scaling in/out up/down. VNFs are scaled in or out by spawning new container instances, adjusting the number of active VNFs as needed. They are scaled up or down by modifying the CPU quota for specific containers, allowing fine-tuned control over computational resources. This flexibility ensures the system can dynamically adapt to varying workloads and performance requirements.

Power Consumption. Figure 2(a) shows the power consumption of multiple containerized VNFs on both the host and DPU. Running 3 VNFs on the host consumes about 75W, similar to having an NVIDIA BlueField SmartNIC plugged in. This suggests that offloading these containers to the DPU could be beneficial, freeing

up host CPU cores without increasing power consumption. Scaling up to 8 VNFs on the host can raise power consumption to 130W, highlighting the efficiency advantages of offloading to the DPU.

Throughput. Figure 2(b) shows the maximum throughput supported by the system with the maximum number of VNFs deployed. The DPU can handle up to 40 Mpps, after which throughput declines sharply. This highlights the limits of offloading VNFs with InReal, indicating that while the DPU can manage significant traffic independently, performance drops beyond certain thresholds. Understanding these contentions is essential for optimizing resource allocation between the host and DPU to maximize system efficiency.

Scaling In/Out Effects. Figures 2(c) and 2(d) show the impact of adjusting CPU quotas for a given VNF instance. This experiment illustrates the trade-off between throughput degradation and energy savings with reduced CPU quotas. The figures illustrate the scenario with the maximum number of VNFs deployed at a traffic rate of 40 Mpps. Reducing the host's CPU quota from 100% to 60% leads to a 40% decrease in throughput and over a 30% drop in power consumption. This underscores the complexity of offloading decisions, especially when power consumption is paramount. The results suggest that reducing CPU quotas on the host may lead to better power efficiency and throughput than offloading to the SmartNIC in some scenarios.

4 CLOSING REMARKS

This work introduces InReal, a system for offloading VNFs resource orchestration to SmartNICs. InReal optimizes server resources, enhancing power efficiency while maintaining performance and freeing the CPU. The next steps involve investigating scaling decisions and networking contentions in the host/DPU to balance performance and power consumption based on workload conditions.

REFERENCES

[1] Jiaxin Lin et al. 2023. RingLeader: Efficiently Offloading Intra-Server Orchestration to NICs. In *20th USENIX NSDI*. Boston, MA, 1293–1308.
[2] Francisco Pereira et al. 2024. Automatic Parallelization of Software Network Functions. In *21st NSDI*. USENIX, Santa Clara, CA, 1531–1550.
[3] Parham Yassini et al. 2024. Horus: Granular In-Network Task Scheduler for Cloud Datacenters. In *21st NSDI*. USENIX, Santa Clara, CA, 1–22.