# DPU-based system evaluation with End-to-End Modular Simulation

Young Ju Woo, Eui-Young Chung
*Dept. of Electrical and Electronic Engineering*
*Yonsei University*
*Seoul, Republic of Korea*
*wyj@yonsei.ac.kr, eychung@yonsei.ac.kr*

## Abstract

*As the burden on host CPU increased in maintaining the infrastructure of data centers, the emergence of data processing unit (DPU) became inevitable. Deploying DPU reduces the load on host CPU while enabling acceleration of security, network, and storage, which are a part of data center infrastructure. Previous studies have been conducted using commercial products. This approach makes it challenging to explore the internal architecture of DPU. In this study, we propose an idea that allows for fundamental exploration of DPU through easy and quick solutions. To demonstrate its use case of it, we implement one of the primary security functions accelerated by DPU, with modular full-system simulation framework, and evaluate the acceleration achieved. While the overall execution time varied across functions, we observed a reduction of 20% and 2% in execution time for each function, respectively.*

**Keywords:** DPU, IPU, SmartNIC

## 1. Introduction

As data centers continue to expand, the responsibility of host CPU extends beyond computing tasks to encompass the maintenance of data center systems. This transition has led to a significant allocation of computing resources towards infrastructure maintenance, leading to application performance bottlenecks. DPU offers a solution by offloading and accelerating data center infrastructure. DPU efficiently manages and process data movement, optimizing these essential tasks. DPU typically integrates a network interface card (NIC) and logic specifically designed for data center infrastructure maintenance, often incorporating specialized accelerators for data encryption and compression. By offloading processing tasks from host CPU to DPU, computing resources are freed up, resulting in enhanced application performance and reduced infrastructure maintenance costs.

The demand and importance of DPU have increased, leading to a surge in research efforts. Previous research focused on determining the optimal design platform for DPU. This includes investigating the suitability of different design platforms such as FPGA, ASIC, and SoC [1] for DPU implementation. Another area of research aims to identify suitable tasks for DPU to execute. For example, researchers explored the acceleration of packet processing [2] and utilization of DPU in various distributed applications, such as replicated key-value store, distributed transactions [3], and distributed file system (DFS) [4].

Many studies across different research areas have relied on commercial products. However, this reliance on commercial products poses challenges when attempting to conduct an in-depth exploration of the internal architecture of DPU. Additionally, implementing desired functionalities in hardware and evaluating their performance within the system require considerable time and effort.

To address this issue, we propose the method that can rapidly evaluate DPU-based system by modeling DPU functionality with end-to-end modular simulation. By simulating DPU-based system with modular simulation, researchers can quickly evaluate performance and explore different design options, reducing the cost and time required for hardware-based testing.

## 2. DPU-based system modeling

**End-to-end modular simulation.** To evaluate the performance of DPU-based system, the environment setup where multiple hosts using DPU communicate with each other is necessary. This is because our goal is not to observe the single operation of DPU, but to observe the overall system performance in which DPU is used. Our environment setup is based on Simbricks [5] is a framework that connects host simulator, NIC simulator, and network simulator. Simbricks [5] provides various options for

979-8-3503-2641-3/23/$31.00 ©2023 IEEE

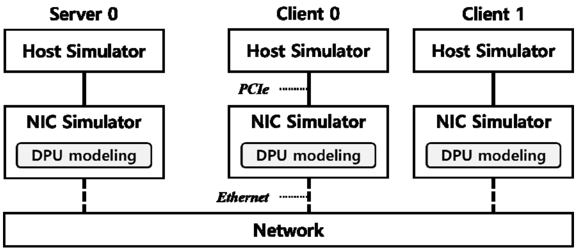


**Figure 1. DPU-based system with end-to-end modular system**

configuring experimental environments, allowing for flexible experiment setups as needed. Simbricks [5] provides various simulation options including QEMU and gem5 as host simulators, i40e behavior model and Corundum RTL model as NIC simulators, and net switches and ns-3 as network simulators.

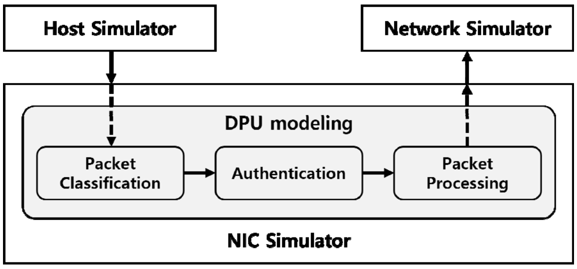


**Figure 2. Packet Processing Procedures**

**DPU modeling.** To build DPU-based system with Simbricks [5], we modeled DPU functionality on NIC simulator as shown in figure 1. It is important to consider what functions are offloaded and in what manner. We implemented authentication functionality using on-path manner.

Authentication is an important part of security and is frequently offloaded to DPU. In addition, the authentication is one of the essential procedures in currently used network security protocols such as Secure Sockets Layer/Transport Layer Security (SSL/TLS), Internet Protocol Security (IPSec), and Secure Shell (SSH). We offload both SHA-256 and SHA-512, which are most frequently used. While many vendors of DPU support SHA-256 acceleration, SHA-512 is often not supported. Through this implementation, we aim to verify its effectiveness.

On-path manner refers to processing all packets that pass through NIC, which is like the operation of traditional NIC and enables easy compatibility with existing NIC.

The overall operation process is shown in figure 2. When a packet is sent, it is first classified based on its total length and header. After that, the authentication function operates and processes the packet based on its results. Once the processing is complete, the packet is sent out through ethernet. On receiving side, packets are processed in the same order. However, the difference is that during the authentication step, the integrity of the packet is verified for assurance. During packet processing step, it removes the information that is needed for authentication to transmit the data in its original state to CPU.

## 3. Evaluation

To evaluate the acceleration performance of the authentication function in DPU-based system, our experimental setup is described in table 1.

**Table 1. Evaluation Configurations**

| | |
|---|---|
| **# of Server** | 1 |
| **# of Client** | 2 |
| **Host Simulator (# of Core)** | QEMU (1) |
| **NIC Simulator** | i40e behavior model |
| **Network Simulator** | Net Switch |
| **Benchmark** | iPerf |

The goal of this experiment is to evaluate multi-DPU environment, therefore, a node configuration is set up as shown in table 1, consisting of 1 server and 2 clients communicating with each other. For fast simulation time, we select QEMU as host simulator and i40e behavior model as NIC simulator. iPerf, a widely used tool for measuring network performance, is used as a benchmark in this study.

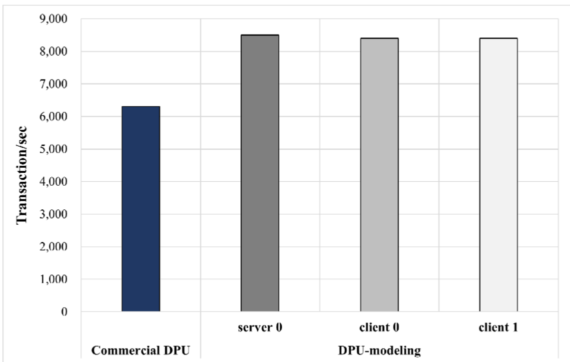


**Figure 3. Comparison with commercial DPU and DPU-modeling applied NIC**

**Comparison with commercial DPU.** To ensure the validity of our DPU-modeling, we compare it to offloading TLS to commercial DPU [6], Mellanox BlueField NIC. TLS is one of the network security protocols that involve authentication and various other security functions. As DPU modeling includes some part of TLS, it should show better performance than performing the full TLS. We evaluate DPU modeling under the same condition as depicted in table 1, with the same number of host CPU cores [6].

As shown in figure 3, while commercial DPU shows 6300 transactions/sec, we confirm that our simulation environment performs as the expectation, with 8497, 8395, and 8397 transactions/sec on server 0, client 0, and client 1, respectively.

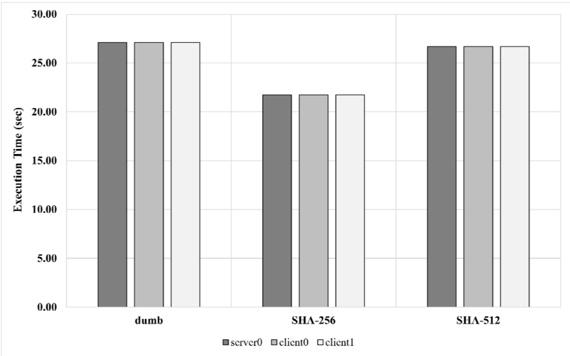


**Figure 4. Evaluation of Authentication Offloading**

**Authentication offloading.** A conventional NIC that only handles packet transmission and reception without performing authentication is referred to as a dumb NIC. The performance time of dumb NIC is compared to DPU-modeling applied NIC. We measured the time it took for the benchmark to complete, and this is repeated 10 times to obtain an average.

Despite the addition of the authentication function, the completion time was reduced by approximately 20% for SHA-256 and approximately 2% for SHA-512. This indicates that, even with additional functionality, both cases saw a decrease in completion time. The time reduction effect is less pronounced for SHA-512 compared to SHA-256 because SHA-512 requires more computation than SHA-256.

## 4. Conclusion

This paper proposes the method for implementing DPU-based system by modeling using end-to-end modular full system framework [5] and evaluates its performance. We offload authentication functions and measure the time savings achieved by using DPU modeling applied NIC. The results show that the time savings vary depending on the authentication function used, ranging from 2% to 20% reduction in execution time compared to performing the same task without security function offloading.

Currently, we only explore limited parts of the functionality and architecture of DPU. We plan to implement and investigate more diverse function offloading and internal architecture support in the future.

## Acknowledgment

This research was supported by the MOTIE(Ministry of Trade, Industry & Energy (1415181094) and KSRC(Korea Semiconductor Research Consortium) (20019956) support program for the development of the future semiconductor device.

## References

[1] Grant, S., Yelam, A., Bland, M., & Snoeren, A. C., “SmartNIC performance isolation with FairNIC: Programmable networking for the cloud”, *SIGCOMM*, pp. 681-693, July 2020

[2] Phothilimthana, P. M., Liu, M., Kaufmann, A., Peter, S., Bodik, R., & Anderson, T. E., “Floem: A Programming System for NIC-Accelerated Network Applications”, *OSDI*, Vol. 18, pp. 663-679, October 2018

[3] Liu, M., Cui, T., Schuh, H., Krishnamurthy, A., Peter, S., & Gupta, K., “Offloading distributed applications onto smartnics using ipipe”, *SIGCOMM*, pp. 318-333, 2019

[4] Kim, J., Jang, I., Reda, W., Im, J., Canini, M., Kostić, D., ... & Witchel, E., “LineFS: Efficient SmartNIC offload of a distributed file system with pipeline parallelism”, *SOSP*, pp. 756-771, October 2021

[5] Li, H., Li, J., & Kaufmann, A., “SimBricks: end┐ to-end network system evaluation with modular simulation”, *SIGCOMM*, pp. 380-396, August 2022

[6] Kim, D., Lee, S., & Park, K., “A case for smartnic-accelerated private communication”, *APNet*, pp. 30-35, August 2020