
AINFV: Analysis of Isolation (memory/packet) in Network Function Virtualization

Abdul Ahad Ayaz

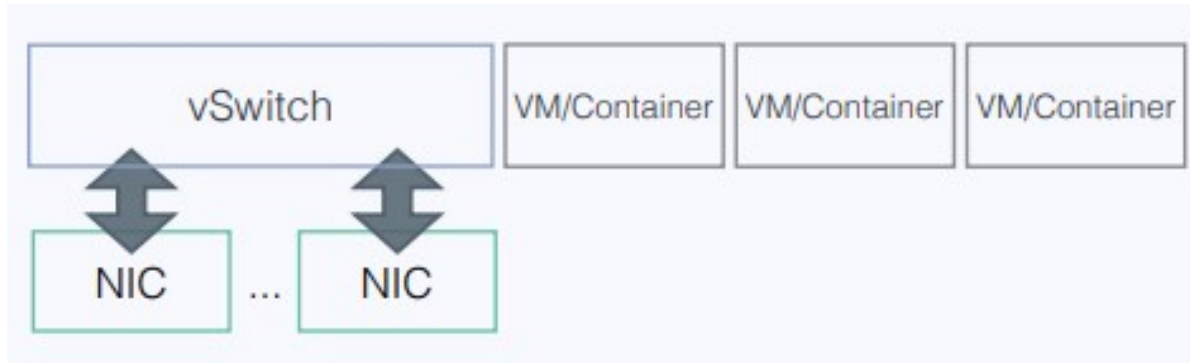


Network Function Virtualization (NFV)

- What is NFV?
- Why we use NFV?
- What are the requirement of NFV?
- Issues of Using NFV
 - Executing NFs
 - Developing NFs

Executing NFs

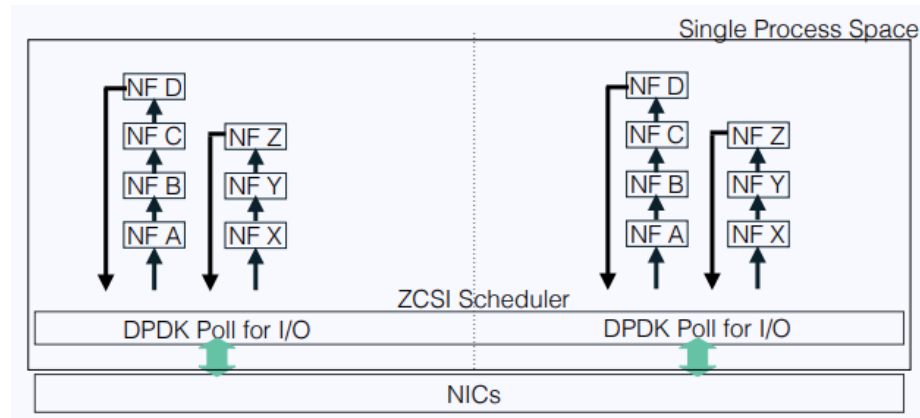
- Isolation vs Performance
- Types of Isolation
 - Memory Isolation
 - Packet Isolation
- Current Approach
 - Using VMs or Containers



[10]

Executing NFs

- NetBricks[1]
- Execution Environment
 - Single Process Space
- Isolation using Zero Copy Soft Isolation (ZCSI)[1]
 - Packet Isolation : Unique Types
 - Memory Isolation : Type checks and array bounds checks



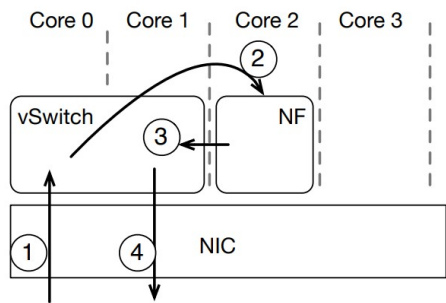
NetBricks Architecture[10]

Developing NFs

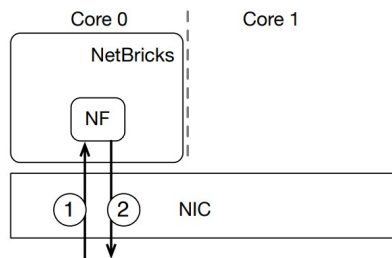
- High-Level Programming Language vs Performance
- Current Approach
 - More focused on Low-level code
 - Spending a lot of time on optimization
- NetBricks
 - Modular code
 - Separate common functionality and User-defined functionality

Isolation Analysis

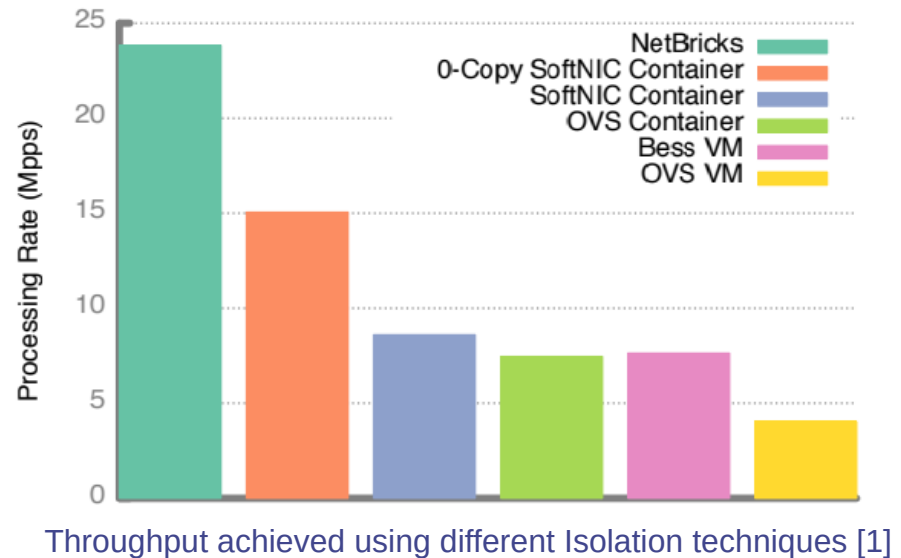
- Simple NF vs NF chains vs Complex NF
- Simple NF
 - Swaps source and destination address of receiving packets.



For VM/Container[1]

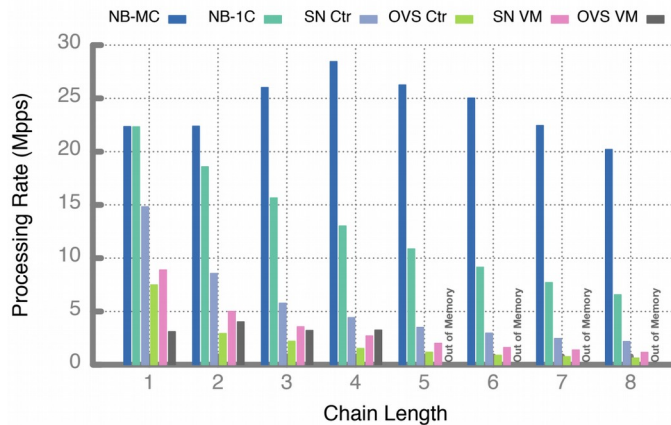


For NetBricks[1]

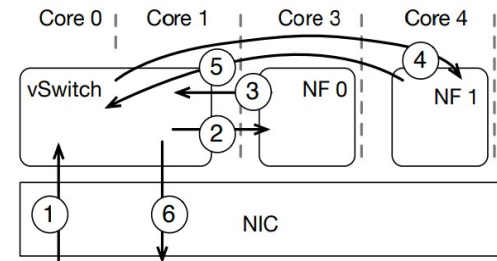


Isolation Analysis

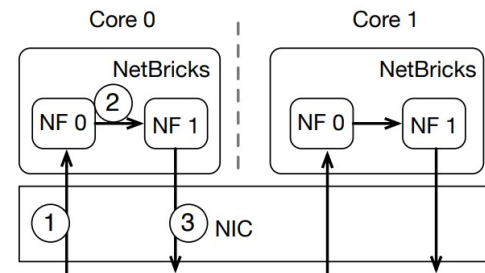
- Simple NF vs NF chains vs Complex NF
- NF chains
 - Multiple instances of Packet TTL (time to live) = 0 NFs.
 - Two cases:
 - Single-core
 - Multi-core



Throughput analysis [1]



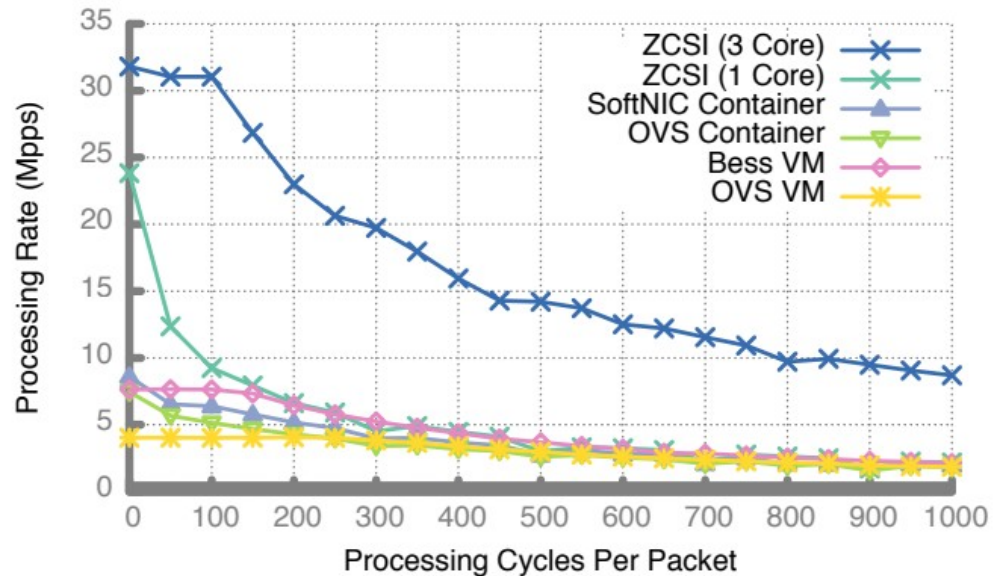
For VM/Container[1]



For NetBricks[1]

Isolation Analysis

- Simple NF vs NF chains vs Complex NF
- Complex NF
 - Simple NF used with increased computation time per packet.
 - Two cases:
 - Single-core
 - Three-cores



Throughput of complex NF using different technologies[1]

Related Frameworks

- Developing NF
 - YANFF[2]
 - libVNF[3]
 - FLICK[4]
- Executing NF
 - NetVm[5] / OpenNetVM[6]
 - HyperNF[7]
 - G-NET[8]
- Isolation
 - SafeBricks[9]

Conclusion & Further Research

- Conclusion

- Isolation is necessary for NFV
- VM/Container ensures isolation but at the cost of Performance degradation
- NetBricks runs NF as a single process
 - Ensures both memory and packet isolation

- Further Research

- Add control plane functionality to NetBricks
- NetBricks integration with MANO systems

References

- [1] A. Panda, S. Han, K. Jang, M. Walls, S. Ratnasamy, and S. Shenker, “NetBricks: Taking the V out of NFV.” in OSDI’16, 2016, pp. 203–216. [Online]. Available: <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/panda>
- [2] I. Philippov and A. Melik-Adamyan, “Novel approach to network function development,” in Proceedings of the 13th Central & Eastern European Software Engineering Conference in Russia on -CEE-SECR ’17. ACM Press, 2017, pp. 1–6. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3166094.3166111>
- [3] P. Naik, A. Kanase, T. Patel, and M. Vutukuru, “libVNF: A Framework for Building Scalable High Performance Virtual Network Functions,” in Proceedings of the 8th Asia-Pacific Workshop on Systems - APSys ’17. ACM Press, 2017, pp. 212–224. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3124680.3124728>
- [4] A. Alim, R. G. Clegg, L. Mai, L. Rupprecht, E. Seckler, P. Costa, P. Pietzuch, A. L. Wolf, N. Sultana, J. Crowcroft, A. Madhavapeddy, A. W. Moore, R. Mortier, M. Koleni, L. Oviedo, M. Migliavacca, and D. McAuley, “FLICK: Developing and running application-specific network services,” in 2016 USENIX Annual Technical Conference (USENIX ATC 16). Denver, CO: USENIX Association, 2016, pp. 1–14. [Online]. Available: <https://www.usenix.org/conference/atc16/technical-sessions/presentation/alim>

References

- [5] J. Hwang, K. K. Ramakrishnan, and T. Wood, “NetVM: High performance and flexible networking using virtualization on commodity platforms,” IEEE Transactions on Network and Service Management, vol. 12, no. 1, pp. 34–47, mar 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/7036139/>
- [6] M. Yurchenko, P. Cody, A. Coplan, R. Kennedy, T. Wood, and K. K. Ramakrishnan, “OpenNetVM: A Platform for High Performance Network Service Chains,” in Proceedings of the 2016 workshop on Hot topics in Middleboxes and Network Function Virtualization. ACM, 2018, pp. 1–2. [Online]. Available: <https://dl.acm.org/citation.cfm?id=2940155>
- [7] K. Yasukata, F. Huici, V. Maffione, G. Lettieri, and M. Honda, “HyperNF: building a high performance, high utilization and fair NFV platform,” pp. 157–169, 2017. [Online]. Available: <http://dl.acm.org/citation.cfm?doid=3127479.3127489>
- [8] “G-NET : Effective GPU Sharing in NFV Systems,” 2018. [Online]. Available: <https://www.usenix.org/conference/nsdi18/presentation/zhang-kai>
- [9] R. Poddar, C. Lan, R. A. Popa, and S. Ratnasamy, “SafeBricks: Shielding Network Functions in the Cloud,” pp. 201–216, 2018. [Online]. Available: <https://www.usenix.org/conference/nsdi18/presentation/poddar>
- [10] A. Panda, S. Han, K. Jang, M. Walls, S. Ratnasamy, and S. Shenker, “NetBricks: Taking the V out of NFV.” in OSDI’16, 2016, (Presentation). [Online]. Available: https://www.usenix.org/sites/default/files/conference/protected-files/osdi16_slides_panda.pdf