Virtualization at the Network Edge: A Performance Comparison

2016 IEEE 17th International Symposium On
A World of Wireless, Mobile and Multimedia Networks (WoWMoM)

Fla´vio Ramalho and Augusto Neto

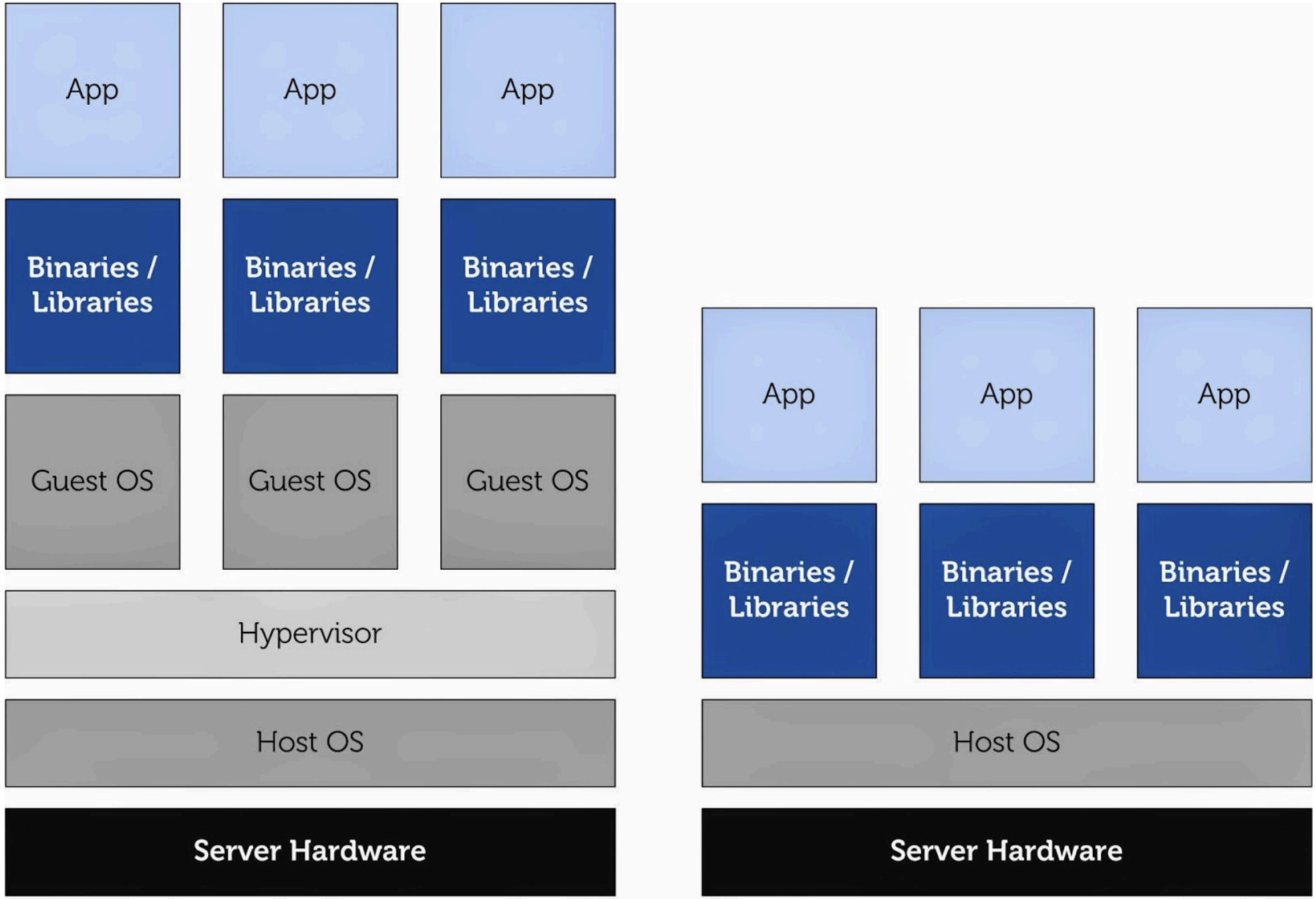
- The study was performed through the execution of several synthetic benchmarks providing an insight in the performance overhead introduced by Docker containers (lightweight-virtualization) and KVM VMs (hypervisor-virtualization) running at network edge devices.

- The intent is to quantify the level of overhead introduced by these techniques compared to a non-virtualized environment, when running on a typical edge device.

VIRTUALIZATION TECHNIQUES OVERVIEW

- *Hypervisor-Based Virtualization*
  - The core of the hypervisor based virtualization is a software technology called hypervisor, which allows several operating systems to run side-by-side on a given hardware.

- *Container-Based Virtualization*
  - Containers are a lightweight approach to virtualization that can be used to rapidly develop, test, deploy, and update IoT applications at scale.
  - Docker is an open platform for container-based virtualization on Linux, it is built on top of facilities provided by the Linux kernel.


App
App
App
Binaries / Libraries
Binaries / Libraries
Binaries / Libraries
Guest OS
Guest OS
Guest OS
Hypervisor
Host OS
Server Hardware
App
App
App
Binaries / Libraries
Binaries / Libraries
Binaries / Libraries
Host OS
Server Hardware

	Virtualization (i.e. kvm, xen)	**LXC Containers**
Footprint	Requires a hypervisor and a full operating system image.	Does not require a hypervisor or a separate operating system image.
OS supported	Any OS supported by the hypervisor	Most Linux distros, uses same kernel as host
Typical server deployment	10 – 100 VMs	100 - 1000 containers
Boot time	Less than a minute	Seconds
Physical resources use (i.e. memory, CPU)	Each VM has resource reserved for its own use	Shared by all containers

PERFORMANCE EVALUATION (1)

- Computer model
  - Cubieboard2
- Platform
  - Native
  - Docker
  - Kernel-based Virtual Machine (KVM)
- The benchmark tools measure
  - CPU
  - Memory
  - Disk I/O
  - Network I/O

PERFORMANCE EVALU

- *CPU Evaluation*
  - NBench
    - a synthetic computing benchmark pr
      measure CPU, FPU, and Memory Syst
  - SysBench
    - a multi-threaded benchmark tool that
      evaluating a variety of low-level system
  - LINPACK benchmarks
    - measure the computer's floating- point rate of execution
    - the algorithm uses a random matrix A (size N),
      and a right hand side vector B that is defined as follows: A * X = B.

TABLE I
CPU BENCHMARK: NBENCH

Platform	Memory Index		Integer Index		Floating-Pt Index	
Native	4.028	%	4.243	%	0.417	%
Docker	4.024	**-0.10%**	4.209	**-0.80%**	0.417	**0.00%**
KVM	3.856	**-4.27%**	4.121	**-2.88%**	0.411	**-1.44%**

Docker	145.918	**+0.25%**	9.695	**+34.77%**
KVM	147.250	**+1.17%**	78.529	**+991.65%**

Fig. 1. The value of Linpack results on each platform over 15 runs. In this case N=2000.

PERFORMANCE EVALUATION (3)

- *Disk I/O Evaluation*
  - Bonnie++
    - an open-source benchmark tool that is suited to perform a number of simple tests of hard drive and file system to characterize the disk performance
    - using a file size of 3 GiB.

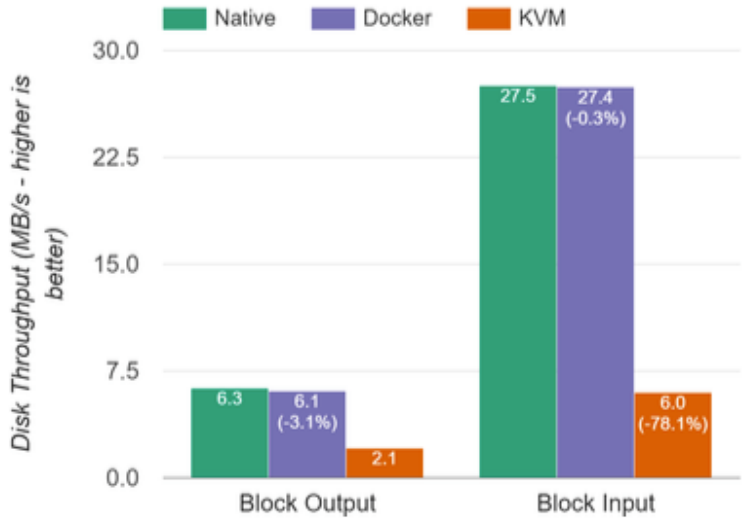


Fig. 2. Disk throughput results from running Bonnie++ using a file size of 3 GiB.

Fig. 3. Disk throughput results from running DD using a file size of 3 GiB.

PERFORMANCE EVALUATION (4)

- *Memory Evaluation*
  - STREAM
    - benchmark the performance of Memory I/O
    - memory bandwidth (in MB/s)
    - four simple kernel different operations:
      - *Copy*, *Scale*, *Add* and *Triad*

TABLE III
MEMORY BENCHMARK: STREAM

Platform	Copy		Scale		Add		Triad	
Native	1759.4	%	806.8	%	654.6	%	534.7	%
Docker	1754.5	**-0.28%**	804.8	**-0.25%**	652.8	**-0.27%**	533.2	**-0.28%**
KVM	1723.4	**-2.05%**	786.3	**-2.54%**	641.2	**-2.05%**	523.7	**-1.78%**

PERFORMANCE EVALUATION (5)

- *Network Evaluation*
  - Netperf
    - a benchmark tool embedded with seve
      measure the performance of different t

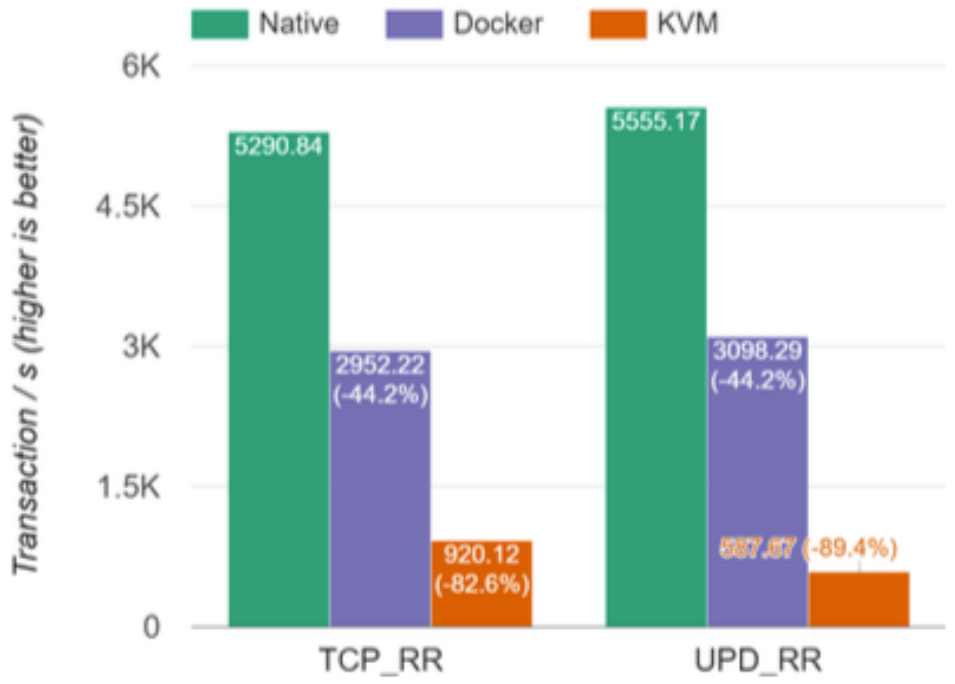


Fig. 5. Network request/response results from running netperf.

CONCLUSION

- While the results on the hypervisor-based solution showed a significant overhead that cannot be easily mitigated, the results of the Docker platform are promising .

- Linux containers seems to take advantage over hypervisor- based virtualization for deploying applications at the network edge.