Hypervisors vs. Lightweight Virtualization: a Performance Comparison

Authors:
Roberto Morabito
Jimmy Kjällman
Miika Komu

Presented By:

Rituja Dange 1750196
Bharathi Manoharan 1750197
Misba Momin 1750151

UNIVERSITY of WASHINGTON

OUTLINE

OVERVIEW

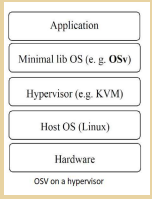
- This paper presents a detailed performance comparison of traditional hypervisor based virtualization and new lightweight solutions (Container based virtualization)
- Several benchmarks tools have been used in order to understand the strengths and weaknesses introduced by these different platforms in terms of processing, storage, memory and network
- Results show that containers achieve generally better performance when compared with traditional virtual machines and other recent solutions
- Virtualization technologies are having predominant role
- The main benefits of virtualization: hardware independence, isolation, secure user environments, and increased scalability, together with the large number of new properties optimized for different use cases

Contd…

- Consequently, the area has become very attractive and competitive, contributing to the raise of novel solutions of the main classes of virtualization technologies, that is container based virtualization and hypervisor-based virtualization
- Further, this has boosted the introduction of hybrid techniques, which promise to combine the advantages of the previous
- First part of paper literature review and a brief description of all the technologies and platforms evaluated is provided
- The methodology used to realize our performance comparison is introduced in second part. The benchmark results are presented

INTRODUCTION OF TECHNOLOGIES

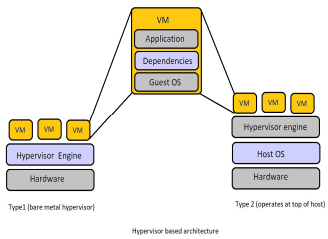
- Container-based Virtualization: it can be considered as a lightweight alternative to hypervisor-based virtualization
- Hypervisors abstract hardware, which results in overhead in terms of virtualizing hardware
- In contrast, containers implement isolation of processes at the operating system level, thus avoiding such overhead.
- Advantage of container-based solutions
  - They can achieve a higher density of virtualized instances
  - Disk images are smaller compared to hypervisor-based solutions
- Disadvantage of container-based solutions
  - Windows containers cannot be run on top of a Linux host
  - Containers do not isolate resources as well as hypervisors

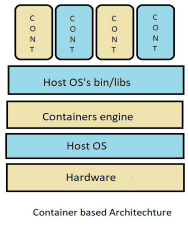


OSV on a hypervisor

Contd…

- In performance analysis, we focus on LXC and Docker





Container based Architechture

Contd…

- Hypervisor-Based Virtualization:
- Contrary to containers, hypervisors operate at the hardware level
- Advantage : Supporting standalone virtual machines that are independent and isolated of the host system
- Disadvantage: A full operating system is installed to virtual machine, which means that the image will be substantially larger
- For hypervisor-Based Virtualization, Linux's Kernel-based Virtual Machine (KVM) is used for benchmark testing which as characteristics of both type1 and type2 hypervisor
- OSV: it achieves the isolation benefits of hypervisor-based systems, but avoids the overhead (and configuration) of a complete guest OS

RELATED WORK

Hwang et al. compared four hypervisors (Hyper-V, KVM, vSphere and Xen) in different use cases

- Elisayed et al. conduct a quantitative and qualitative evaluation of VMware ESXi5, Microsoft Hyper-V2008R2, and Citrix Xen Server 6.0.2 in various scenarios
- Varrette et al. provide a similar analysis, but with some differences
- Toor et al. report a 4% overhead of grid virtualization
- Li et al. measure a commercial (unspecified) hypervisor, Xen and KVM using Hadoop and MapReduce as the use cases
- Recent research literature compares hypervisors with container solutions, including Dua et al., who depict increasing use for containers in PaaS environments
- Felter et al. compare KVM and Docker performance with native environment

HARDWARE AND SOFTWARE PLATFORM USED

Hardware:

- Computer model: Dell Precision T5500
- Processor: Intel Xeon X5560 (8M Cache, 2.80 GHz, 4 cores, 8 threads)
- Memory: 12GB (3x4GB) 1333 MHz DDR3 ECC R
- Disk: OCZ-VERTEX 128GB
- Network: 10Gb/s interface
- OS: Ubuntu 14.04 (64-bit)

Platform	Multi-core Efficiency
Native	98.27%
LXC	98.19%
Docker	98.16%
KVM	97.51%

CPU usaee to calculate the value of PI

CPU PERFORMANCE

- Y-cruncher is a multi-threaded benchmark for multicore systems to calculate the value of Pi.

Platform	Multi-core Efficiency
Native	98.27%
LXC	98.19%
Docker	98.16%
KVM	97.51%

CPU usaee to calculate the value of PI

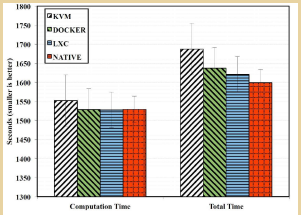
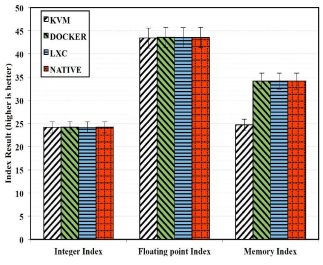


Fig. 2. Results of Y-cruncher over 30 runs. The error bars indicate the Standard Deviation. Container-based solutions perform better than KVM.

Contd…

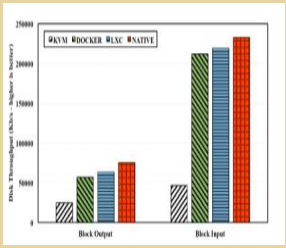
- Nbench is a benchmark tool for CPU, FPU (Floating Point Unit), and memory system performance measurements



DISK I/0 PERFORMANCE

- To measure disk I/O performance author used Bonnie++

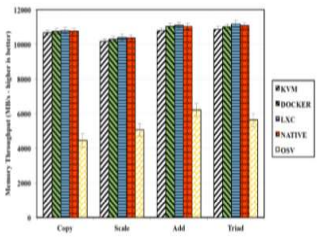
Platform	Random Write speed (Kb/s)		Random Seeks	
Native	48254	%	1706	%
LXC	43172	- 10.53%	1517	- 11.07%
Docker	41170	- 14.68%	975	- 42.84%
KVM	23999	- 50.26%	125.7	- 92.63%



MEMORY PERFORMANCE

- The performance as measured by the tool has a strong dependency to the CPU cache size
- Rule: "each array must be at least 4 times the size of the available cache memory".

Operation	Kernel
Copy	x[i] = y[i]
Scale	x[i] = q * y[i]
Add	x[i] = y[i] + z[i]
Triad	x[i] = y[i] + q * z[i]



NETWORK PERFORMANCE

The configuration for the tests is as follows

- Two identical machines directly connected with 10 Gigabit Ethernet Link
- One host is running netperf client and the other netperf server
- Default values for the Local/Remote socket size and the Message sizes are used
- Test duration time: 60 seconds
- netperf used tests: TCP_STREAM, UDP_STREAM
- IPv4 addressing
- Results represent the average across 15 runs

Platform	TCP_STREAM (Mbps)	%	UDP_STREAM (Mbps)	%
Native	9413.76	%	6907.98	%
LXC	9411.01	−0.00029%	3996.89	−42.14%
Docker	9412	−0.00018%	3939.44	−42.97%
KVM	6739.01	−28.41%	3153.04	−54.35%
OSv	6921.97	−26.46%	3668.95	−46.88%

AUTHOR'S EVALUATION

- Operating system virtualization is the use of software to allow a piece of hardware to run multiple operating system images at the same time
- The concept of containerization basically allows virtual instances to share a single host operating system and relevant binaries, libraries or drivers
- the isolation between the host and the container is not as strong as hypervisor-based virtualization since all containers share the same kernel of the host
- The result shows that the overhead introduced by containers can be considered as almost negligible

CONCLUSION

- Container-based solutions are challenging traditional hypervisor based virtual machines
- The container based solutions are more lightweight
- The level of overhead introduced by containers can be considered almost negligible
- Taking all of the differences into account, authors confirm that containers perform well

CRITIQUE: STRENGTHS

Technology Paper:

- For CPU and Disk I/O performance benchmark, different benchmarking tools has been used to compare the results for reliability
- More number of tests has been conducted to verify the consistency between the different results obtained from benchmark tests

Container based Virtualization:

- Performance
- Light weight alternative
- Portability

CRITIQUE : WEAKNESS

Technology Paper:

- For hypervisor model, only one platform was selected whereas two platforms were selected for container based virtualization
- For Memory and network performance benchmarking, only one tool was used

Container based Virtualization

- Multi-tenant security

CRITIQUE : EVALUATION

- CPU Performance benchmarking: Y-cruncher – No details regarding which constant computation was performed
- Analysis for the results obtained from benchmark testing tools was not done

GAPS

- In Disk I/O performance benchmarking, mismatch between the results of Bonnie++ and Sysbench was reported
- Author failed to provide details on: Strength, weakness, anomalies introduced by different virtualization platform

FUTURE WORK

- To repeat the measurements with the recently announced *"Linux Container Daemon" (LXD)*[40]
- OSv represents an interesting work-in-progress alternative

THANK YOU!!!

QUESTIONS???