Questions Assignment 2

Benchmark questions

1. Look at your benchmark results. Are they consistent with your expectations, regarding the different virtualization platforms? Explain your answer. What are the main reasons for the differences between the platforms? Answer these questions for all benchmarks:

We expected the native platform to perform the best for all benchmarks, which turned out to be true, as they is are no redirections between the experiment host´s hardware (or the Parallels Pro “Hardware”) and the virtual machines.

1. CPU

The QEMU VM without KVM performed the worst, since it does not use Hardware acceleration methods like QEMU with KVM or Linux Containerization (if set up like that) with Docker do.

As most virtualization difficulties are caused by conventional IA-32 design, in which sensitive instructions do not always trap in rings bigger than zero, extended architecture for CPUs (1. generation extension), where the VMM in root mode and Guest OS in guest mode coexist in ring 0 and VMM handles permissions for which instructions guest is allowed to handle, has significantly better CPU performance, as there is less VMM intervention necessary.

1. Memory

The same is valid for the memory benchmark. Since hardware acceleration with KVM (2. generation) introduced nested page tables and tagged TLBs (which improves performance for context switches), there is less VMM intervention required in QEMU with KVM and Docker with hardware acceleration. Also, there is no shadow page table overhead due to the nested page tables and a better scalability on multi-core CPUs.

QEMU performs worse, because of the many traps due to the shadow page table mechanism.

1. Random disk access

As we expected, Docker with OS-assisted virtualization has no overhead in comparison to the native host, but KVM delivers significantly less IOPs, since every IO operation must go through QEMU, where I/O instructions usually trap (full virtualization).

1. Sequential disk access

As we expected, there is no difference between the different hosts, since there is potential even for the native host to have much more IOPs.

Also, with Direct Assignment through Hardware assisted virtualization, guest VMs run the unmodified device drivers and there can be efficient I/O without VMM intervention.

e. Fork

f. Iperf uplink

We expected a worse performance for QEMU with and without KVM, since it uses a virtual network device.

2. Can you think of a flaw in the design of the Iperf-based benchmark? How should a more appropriate benchmark be designed when measuring the network performance of virtualization platforms?