HLL VMs

Objective

- These VMs are designed to enable platform independence. Because modern HLL VMs are intended to support network-based computing and object-oriented programming, the features important for supporting these aspects are emphasized.

Virtual Machine Language

- It's **type of language which can be understood by different operating systems**.
- It is platform-independent.
- Just like to run any programming language (C, python, or java) we need specific compiler that actually converts that code into system understandable code (also known as byte code). The same virtual machine language works.
- If we want to use code that can be executed on different types of operating systems like (Windows, Linux, etc) then virtual machine language will be helpful.

Questions to ponder!

- Is there any limit to no. of virtual machines one can install?
- Can one access the files of one VM from another?

Is there any limit to no. of virtual machines one can install?

- In general there is no limit because it depends on the hardware of your system. As the VM is using hardware of your system, if it goes out of it’s capacity then it will limit you not to install further virtual machines.

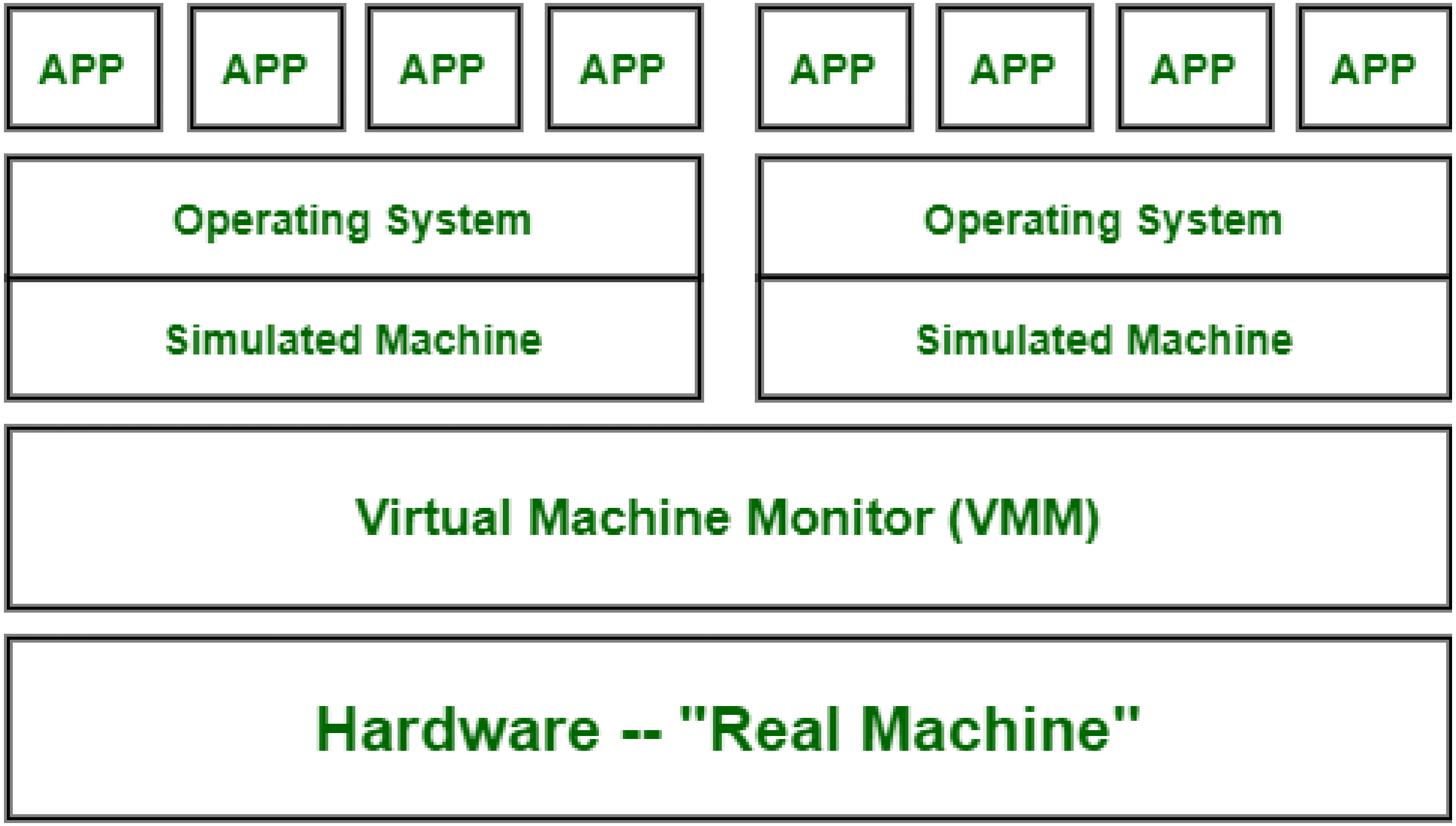
Can one access the files of one VM from another?

- In general No, but as an advanced hardware feature, we can allow the file-sharing for different virtual machines.

Types of Virtual Machines

- **System Virtual Machine:** These types of virtual machines gives us complete system platform and gives the execution of the complete virtual operating system.
- Just like virtual box, system virtual machine is providing an environment for an OS to be installed completely.

System Virtual Machine



APP APP APP APP

APP APP APP APP

Operating System

Simulated Machine

Operating System

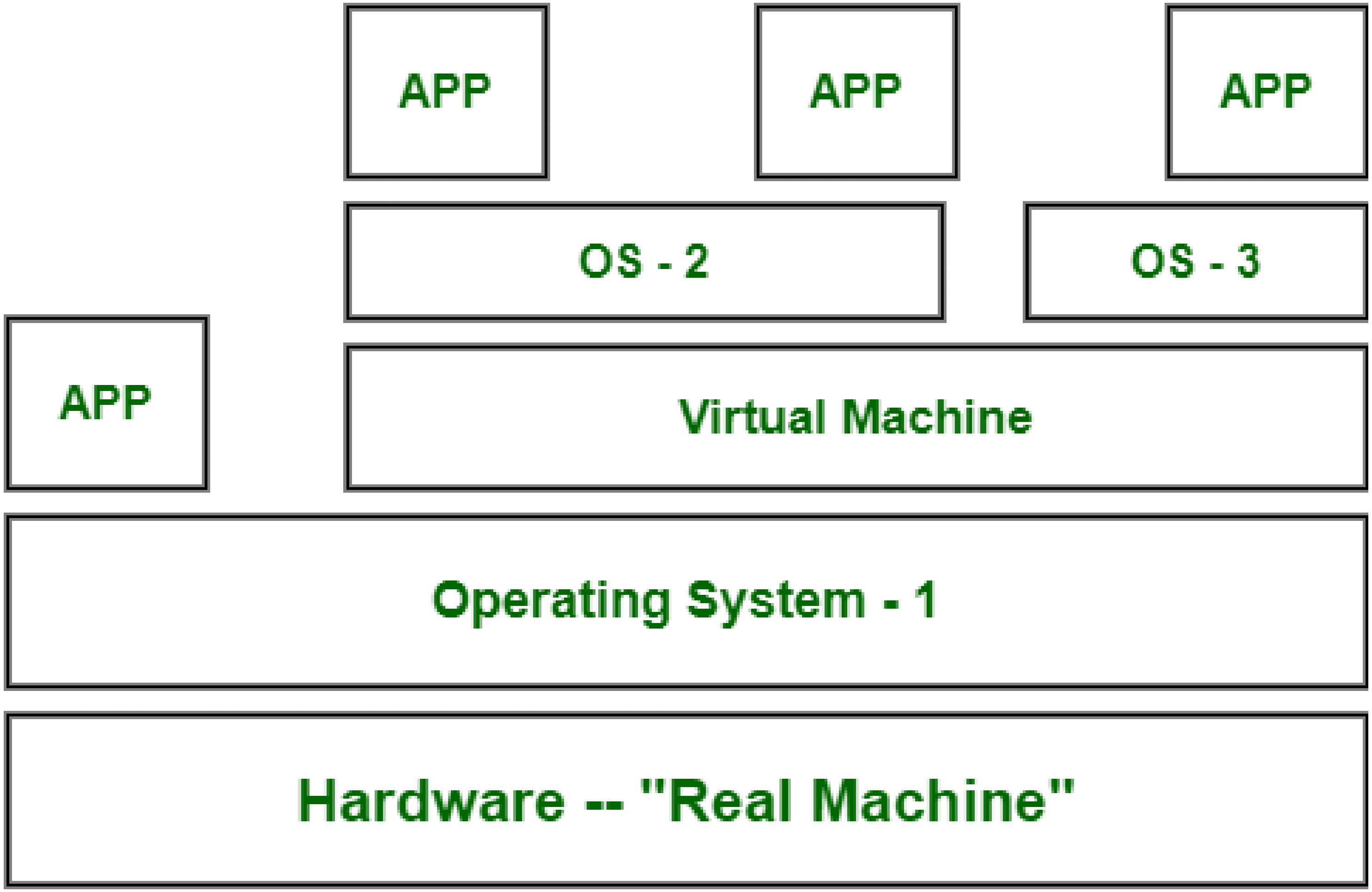
Simulated Machine

Virtual Machine Monitor (VMM)

Hardware -- "Real Machine"

- **Process Virtual Machine :** While process virtual machines, unlike system virtual machine, does not provide us with the facility to install the virtual operating system completely.
- Rather it creates virtual environment of that OS while using some app or program and this environment will be destroyed as soon as we exit from that app.
- There are some apps running on main OS as well some virtual machines are created to run other apps. This shows that as those programs required different OS, process virtual machine provided them with that for the time being those programs are running. **Example –** Wine software in Linux helps to run Windows applications.

Process Virtual Machine



APP

APP

APP

OS - 2

OS - 3

APP

Virtual Machine

Operating System - 1

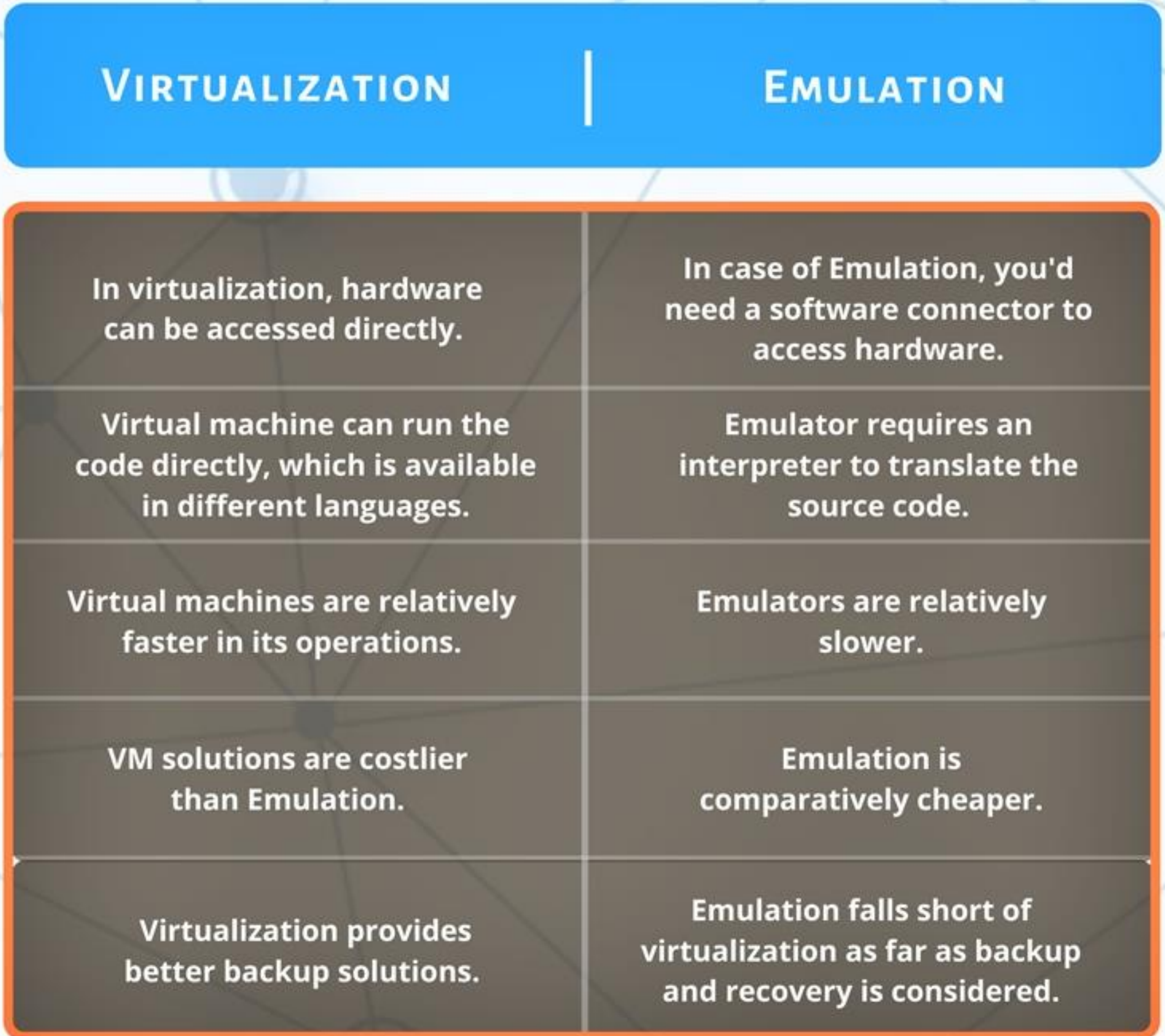
Hardware -- "Real Machine"

Emulation

- Emulation is a concept of creating an environment that imitates the properties of one system onto another.
- An Emulator mimics the qualities and logic of one processor to run in another platform efficiently.
- Emulation is an excellent way to run an OS or software in any other system. Guest Operators need a translation.

Emulation technical definition

- An emulator converts the needed architecture CPU instructions and successfully runs it on another architecture.
- Anyone can access the emulation platforms remotely and is easier to use. It is an excellent ability to have for embedded/OS development, without affecting the underlying OS.
- Emulation can generally handle the size of the design under test (DUT), without considering the host’s capabilities.



Virtualization	Emulation
In virtualization, hardware can be accessed directly.	In case of Emulation, you'd need a software connector to access hardware.
Virtual machine can run the code directly, which is available in different languages.	Emulator requires an interpreter to translate the source code.
Virtual machines are relatively faster in its operations.	Emulators are relatively slower.
VM solutions are costlier than Emulation.	Emulation is comparatively cheaper.
Virtualization provides better backup solutions.	Emulation falls short of virtualization as far as backup and recovery is considered.

Conclusion virtualization Vs Emulation

- In emulation, you use full hardware and software that you want to imitate on top of the host system.
- In virtualization, you mimic the only parts of the hardware according to your requirements with the help of guest OS to run correctly to have the same architecture.

Similarity in virtualization and emulation

- They both are programs that imitate hardware one way or another.
- They both let you mimic and run a program in one environment that is actually meant for the other but with different techniques.