NYU
New York University

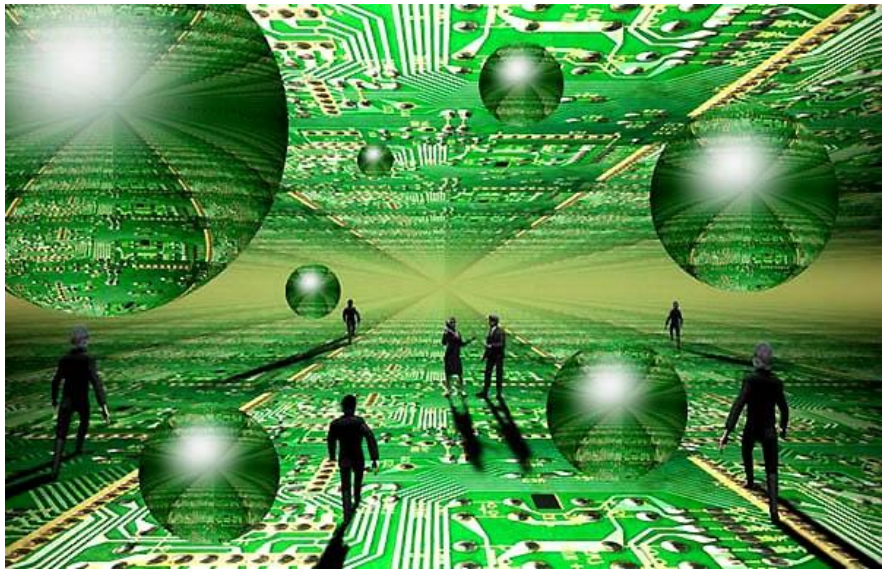
# CSCI-GA.3033-016

# Virtual Machines: Concepts & Applications

## Revision

Mohamed Zahran (aka Z)
mzahran@cs.nyu.edu
http://www.mzahran.com





State a scenario where we still need to use VM even though an application has the same ISA as the hardware and compiled under the same platform

# Answer

- When we need dynamic optimization of the application.
- To gather a lot of statistics about the application's execution profile.
- For logging and replay
- For security (i.e. sandboxing)

What is the difference between virtualization and abstraction?

# Answer

- Abstraction hides the underlying details. Virtualization does not necessarily hide the underlying details. For example, abstraction can show a disk as a file. Virtualization can show a disk as two smaller disks, yet not hide the details of disks (sectors, tracks, …).

Is the compiler affected by virtual machine technology? Explain

# Answer

- For traditional process VMs and system VMs, the compiler does not even realize there is a VM. It just generates the assembly for the guest ISA.
- But for HLL VM, the compiler is affected because the guest ISA is made for non-existent hardware, and hence the back end of the compiler is affected (may be very thin or not needed)

What is code discovery problem? What is the implication of this problem? Do we face it more in RISC or CISC ISAs? Justify?

# Answer

- Code discovery problem is to find out the instructions to be interpreted.
- The implication of this problem is that we may not, especially in CISC ISAs, interpret the whole program offline.
- We face it more in CISC because:
  - instructions of different lengths
  - data insert among instructions
  - padding to ensure alignments

Suppose there is a trap. Does it happen due to an instruction at the source pc? or the target PC? What is the implication of this?

# Answer

- In general it happens at the source PC.
- The implication of this is that the VMM must keep track, especially when executing a translated block, which SPC map to TPC. So to generate traps at the correct sequence and keep the equivalence.

What are profile data that may be of interest to the emulator? and why each type of profile data is important?

# Answer

- Example (not extensive list):
  - block frequency:
    - to decide what to translate and optimize
    - to help in code cache replacement
  - edge frequency:
    - to help forming superblocks
    - to predict jumps/branches
  - Instruction mix: to help in optimization
  - Memory access pattern: to help in memory management
  - …

In HLL VM ISAs, is the portion of privileged instructions more? or less? than traditional ISAs…

# Answer

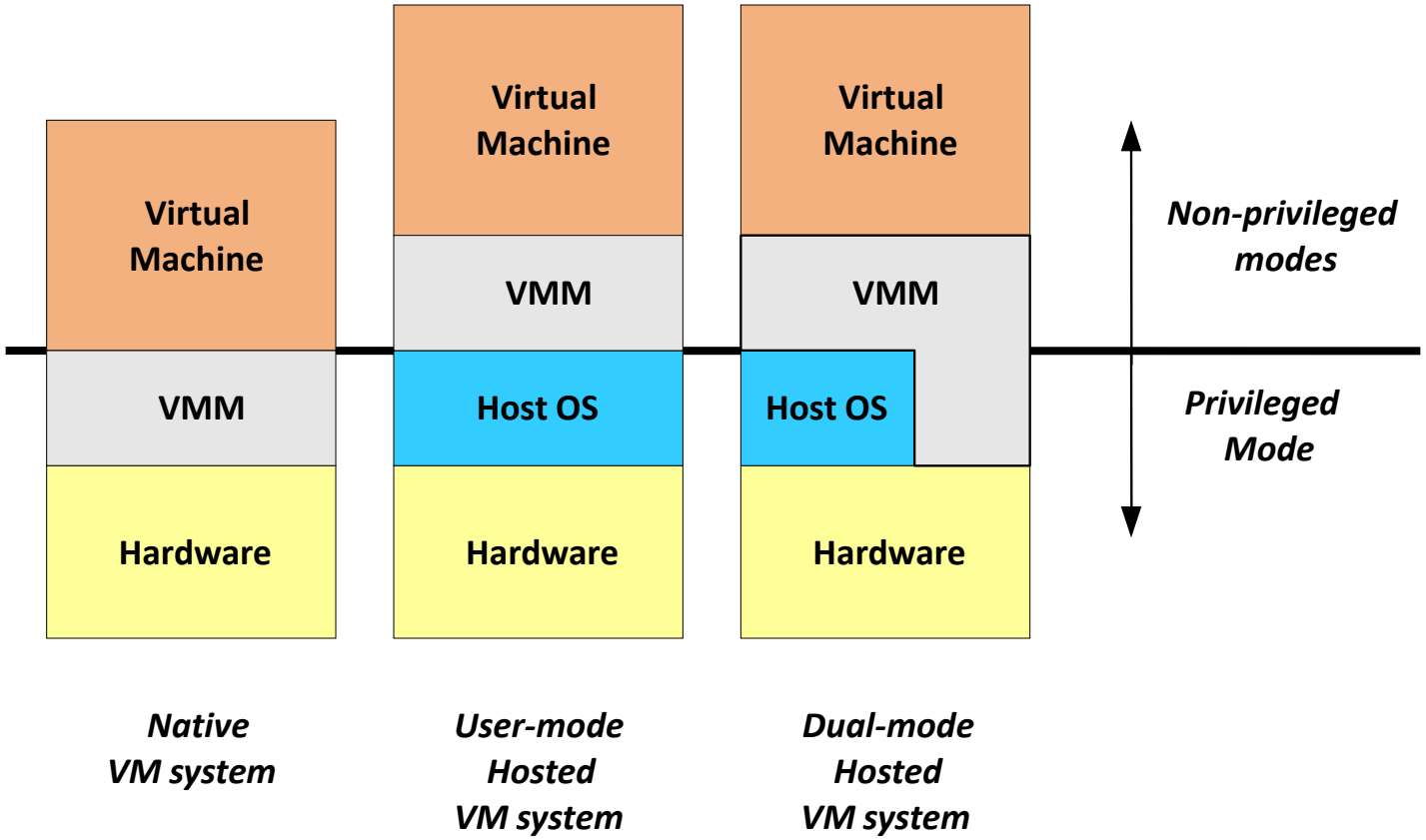
- In HLL VM ISA there are no privileged instructions altogether because the ISA is for a "virtual" hardware.

What is the main source of performance loss in co-designed VM? And how does the VM deal with it?

# Answer

- Indirect jump (especially returns from calls)
- There are hardware methods to deal with it:
  - Jump TLB
  - Dual-Address Return Address Stack
  - Specific ISA instructions like: JTLB_Lookup

# Discuss the pros and cons of each one of the following schemes:



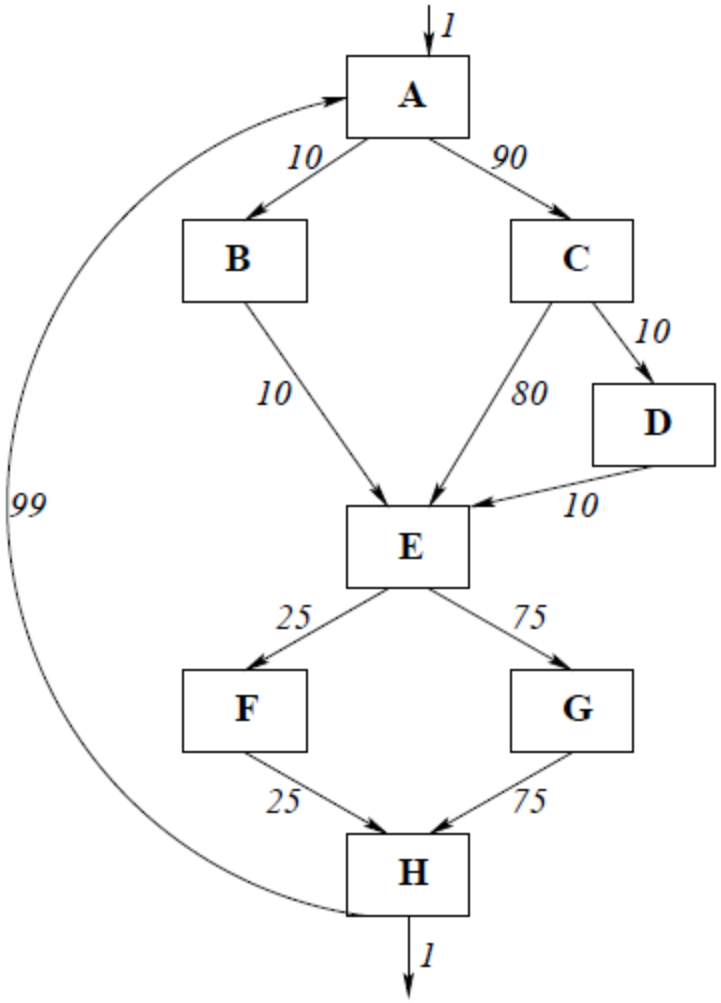
# Answer

- Native:
  - + the highest performance
  - - More sophisticated and need to support many I/O devices
- User-hosted:
  - + Easier to implement and deploy
  - - Performance loss
- Dual mode:
  - + More efficient that user-hosted
  - - Need to modify the host OS

What are the differences between code cache and traditional cache? What are the implications of these differences?

# Answer

- Code cache blocks are of different sizes while traditional cache blocks are of fixed size
  - To bring a new block into code cache we may need to evict more than one block (if they are of smaller sizes) which may affect performance.




Given this control flow graph where a vertex is a static BB. Partition it into superblocks.

# Answer

- A-C-E-G-H
- B
- F
- H

# Done!!