Formal Requirements for Virtualizable Third Generation Architectures

Gerald J. Popek
University of California, Los Angeles
and
Robert P. Goldberg
Honeywell Information Systems and Harvard University

Published: July 1974, Volume 17, Number 7
Communications of the ACM
http://dl.acm.org/citation.cfm?id=361073

Presented by James Owens
Old Dominion University
For CS795 on 11/7/2014
About the Authors

Gerald J. Popek
- Alma Maters:
  - NYU, Nuclear Engineering
  - Harvard, Applied Math
- Notable Works:
  - “Popek and Goldberg Virtualization Requirements”
  - LOCUS, Distributed OS
  - CTO CarsDirect.com
  - CTO NetZero -> Juno
  - DARPA Steering Committee
- Died 20 July 2008

Robert P. Goldberg
- Alma Maters:
  - MIT, Math
  - Harvard, Applied Math
- Notable works:
  - “Popek and Goldberg Virtualization Requirements”
  - Ph.D. thesis, classification for Hypervisors
- Died 25 Feb 1994

Significance

Formally defines a minimal set of conditions which (provably) allow a computer system to support a virtual machine monitor.

Foundational work, cited over 938 times. [Google Scholar, Nov 2014]
Virtual Machine (IN)CAPABLE

IBM 360/67

DEC-PDP 10

Image Sources:
Primary Theorem

For any conventional third generation computer, a virtual machine monitor may be constructed if the set of sensitive instructions for that computer is a subset of privileged instructions.
Approach

1. Define a Third Generation Computer.
   • Identify privileged and sensitive instructions.

2. Define a Virtual Machine Monitor.

3. Discuss examples and extensions.
Third Generation Computer
First Generation Computers
Vacuum Tubes | 1945 -1956

http://campus.udayton.edu/~hume/Computers/first.htm
Image Source
Second Generation Computers
Transistors | 1956 - 1963

http://campus.udayton.edu/~hume/Computers/second.htm
Image Source
Third Generation Computers
Integrated Circuits | 1964 - 1971

http://campus.udayton.edu/~hume/Computers/third.htm
Image Source
Fourth Generation Computers
Microprocessors | 1971 - Present

http://campus.udayton.edu/~hume/Computers/fourth.htm
Image Source
3rd Gen. - Abstract Model

- **Processor** with supervisor and user modes
  - *Supervisor*, may use entire instruction set
  - *User*, may use a subset of instructions

- **Linear, Uniformly Addressable memory**
  - Executable Memory is of size \( Q \)
  - All addresses are a \( \text{base} + \text{offset} < Q \)

- **Arithmetic, look-up, and copy operations exist** while I/O instructions and Interrupts do not.
Primary Theorem

For any conventional third generation computer, a virtual machine monitor may be constructed if the set of sensitive instructions for that computer is a subset of privileged instructions.
3rd Gen. - Abstract Model
State & Linear Memory

**State** $(E, M, P, R)$
- **Executable Memory**
  - Size $Q$
- **Mode of processor**
- **Program Counter**
  - address relative to $R$
    - $0 \leq P < B$
- **Relocation Register** $(L, B)$
  - $L$ – absolute address to a relative 0
  - $B$ – bounds of memory space as size

*Note: All references to memory by the processor are relocated.*

---

**Image Source:**
3rd Gen. - Abstract Model

Traps

If an instruction produces the address $a$, the address development is as follows:

\[
\begin{align*}
\text{if } a + l &\geq q \text{ then } \text{memorytrap } \text{else} \\
\text{if } a &\geq b \text{ then } \text{memorytrap} \\
\text{else } &\text{use } E[a + l].
\end{align*}
\]

All operations which violate constraints or otherwise would cause an undesirable action
trap, then execute some predefined exception handler.

Recall $Q$ is the size of $E$ and $B$ is the size of $R(l,b)$. 
Privileged instructions are those which trap in user mode, do not trap in supervisor mode, AND do not memory trap.

- A function of the physical machines ISA.
- *This definition requires trapping; a NOP does not satisfy the definition.

Examples of privileged instructions in common third generation machines:

1. if \( M = s \) then load_{PSW} else trap; IBM System/360 LPSW
2. if \( M = s \) then load_{R} else trap; {Honeywell 6000 LBAR, DEC PDP-10 DATAQ APR}
• **Sensitive** instructions:
  
1. **Control** Sensitive:
   - Modify resource allocation
   - Modify processor mode

2. **Behavior** Sensitive:
   - The effect of execution depends upon \( R(l,b) \) or the mode.
3rd Gen. - Abstract Model

Instruction Behavior

- **Control** Sensitive:
  - (Potentially) Modify memory allocation.
    - LOAD PSW, LOAD R

- In English: If the MODE or R(l,b) could be different after the execution of some arbitrary instruction, then that instruction is control sensitive.
  - M1 != M2
  - R(l,b)1 != R(l,b)2
3rd Gen. - Abstract Model
Instruction Behavior

- **Behavior** Sensitive:
  - **Location Sensitive:**
    - **LRA:** Load physical address.
    - **Recall S(E,M,P,R) | R(l,b) => P**
      - **E[l + P] | l+P < B && l+P < Q**
  - **Mode Sensitive:**
    - **MFPI:** Move from previous instruction
    - **Effective address depends on mode.**
Primary Theorem

For any conventional third generation computer, a virtual machine monitor may be constructed if the set of sensitive instructions for that computer is a subset of privileged instructions.

Recall:
- Privileged Instructions trap in user mode
  - Sensitive instructions:
    - modify M or R
    - calculate addresses
      - dependent upon M or R
Virtual Machine Monitor
Virtual Machine Monitor

1974 Diagram

Modern diagram

Fig. 1. The virtual machine monitor.

Sources:
Popek Goldberg, 1974
https://www.usenix.org/legacy/event/usenix01/sugerman/sugerman_html/img4.png
Virtual Machine Monitor

Software with three essential characteristics:

1. Provides an environment for programs which is essentially identical to the original machine.

2. Programs (VMs) run in this environment show at worst only minor decreases in speed.

3. The VMM always has complete control of resources.
VMM: Essentially Identical

Provides an essentially identical environment...

Caveats:

1. Availability of system resources
   1. E.g. System Bus, Memory, I/O
2. Timing dependencies due to concurrent virtual machines.
VMM: Efficiency

VMs show only minor decreases in speed...

A majority of instructions must run on bare metal, without software intervention by the VMM.

Non-sensitive, non-privileged instructions are innocuous.
Resources: memory, peripherals, etc.* are entirely controlled by the VMM.

1. No VM may acquire resources without the VMM.
2. The VMM can take resources away.

*Note: This does not include the processor.
VMM Construction
VMM Construction

VMM as a modular control program:
1. Dispatcher
2. Allocator
3. Interpreter(s)
• **Dispatcher**, the top level control module.
  - Dispatcher decides what module to call.
  - All traps lead to the dispatcher.
VMM Construction

- **Allocator**, the system resource manager.
  - e.g. Memory Lookup Table.
  - Ensures against memory violations.
VMM Construction

- **Interpreter(s)**, exception handlers.
  - A set of modules for each trapping instruction
  - One interpreter for each privileged instruction
  - Purpose is to simulate the effect of an instruction which traps.
Primary Theorem

For any conventional third generation computer, a *virtual machine monitor may be constructed* if the set of sensitive instructions for that computer is a subset of privileged instructions.
Virtual Machine (IN)CAPABLE

IBM 360/67

DEC-PDP 10

Image Sources:
Why can’t the PDP-10 support a VM system?

• Answer: PDP-10 Instruction: **JRST 1**

• JRST 1, return to user mode, is a *supervisor control sensitive* instruction which is not a *privileged* instruction.

• What does this mean?
  - It cannot host a VMM as defined
Questions
Recursive Virtualization

A conventional third generation computer is recursively virtualizable if it is:

(a) virtualizable, and
(b) A VMM without any timing dependencies can be constructed for it.
Hybrid Virtual Machines

- The PDP-10 can host a hybrid VM system because all of the user sensitive instructions are privileged.
- An HVM is almost identical to a VMM
  - More instructions are interpreted
    - All instructions in virtual supervisor mode will be interpreted.
Proof

- Existential Proof, non-exclusive.
- State Transition tables are limited by size of theoretical machine.
- Use of Lemmas 1-3 in an inductive proof.
- * Resource Control and Efficiency are addressed by Thm.1.
Lemma 1

• Innocuous instructions, as executed by the virtual machine system, obey the equivalence property.
Lemma 2

- Sensitive instructions, as interpreted by the virtual machine system, obey the equivalence property.
Lemma 3

• Given all single instructions obey the equivalence property, any finite sequence of instructions also obeys the equivalence property.