Virtual CPU Validation

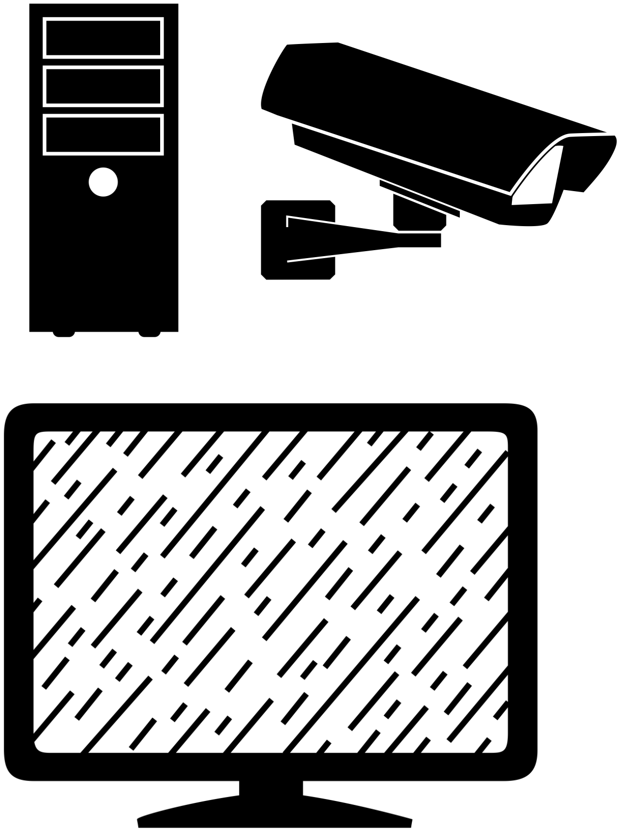
Nadav Amit, Dan Tsafrir, Assaf Schuster

Ahmad Ayoub, Eran Shlomo

Technion
Israel Institute of
Technology

intel®

Question

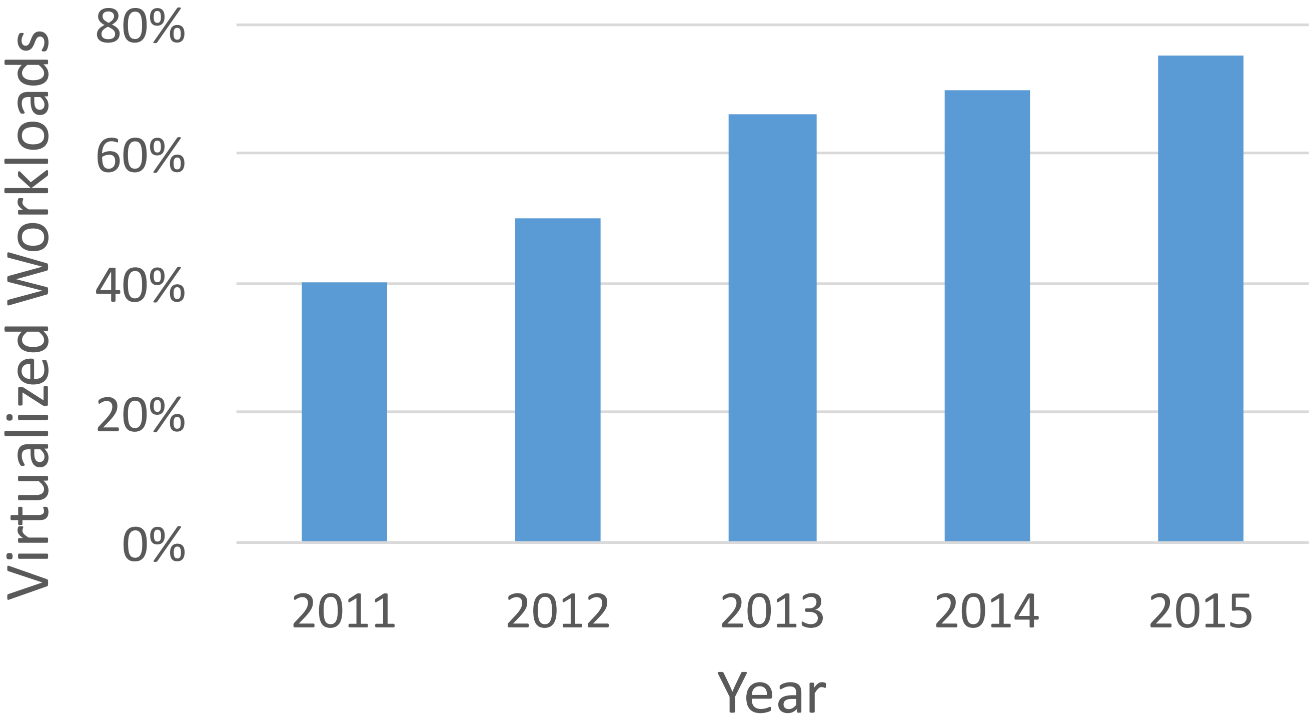


Your video server freezes once a month. Why?

- OS, drivers, BIOS
- CPU, hardware
- Virus / Hack
- Cosmic rays / Power

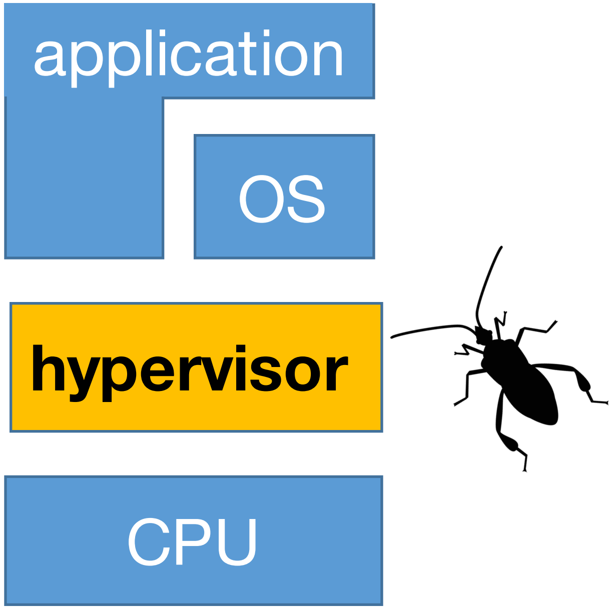
Anything else?

“75% of x86 server workloads are virtualized” [Gartner’15]

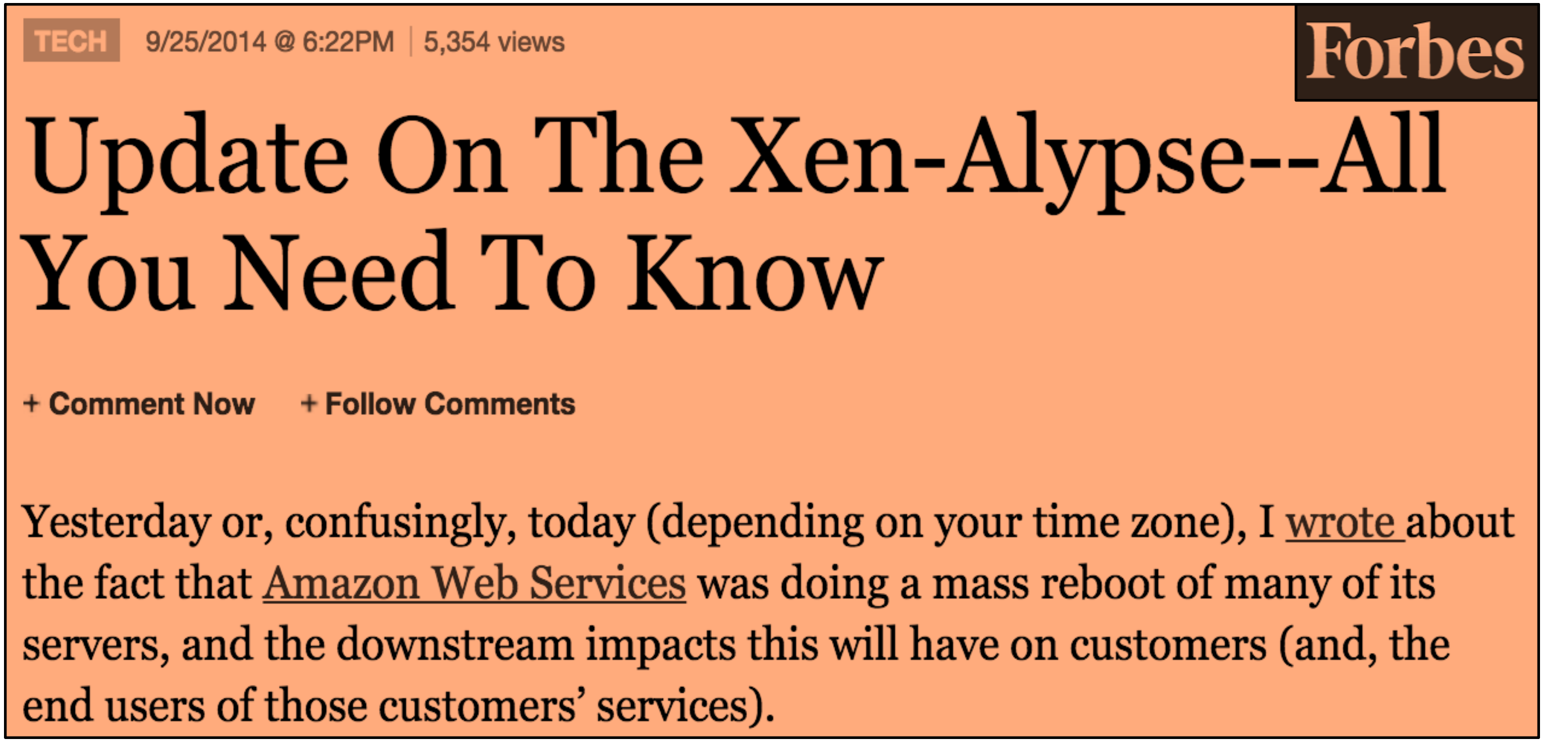


Hypervisor Bugs

- HW assists virtualization, but SW is still there
- Bug implications: security, stability
- CPU virtualization is hardest, and its bugs have the greatest impact



Real-Life Example



TECH 9/25/2014 @ 6:22PM | 5,354 views

Forbes

Update On The Xen-Alypse--All You Need To Know

+ Comment Now + Follow Comments

Yesterday or, confusingly, today (depending on your time zone), I wrote about the fact that Amazon Web Services was doing a mass reboot of many of its servers, and the downstream impacts this will have on customers (and, the end users of those customers' services).

- Non-existent register reads leaked host data
  - Security vulnerability
  - Patching required reboot

Existing Solutions

Micro-hypervisors [Steinberg'10]
Reduced trusted-computing base,
not hypervisor code

Formal Verification [Leinenbach'09]
No formal model of CPU

Fuzzing [Martigonini'12]
No knowledge of CPU semantics

Observation

- CPU vendors invest heavily in developing testing tools
    - 100s of person years or more!

- Physical and virtual CPU should behave similarly

- So tools for testing physical CPUs
  should be able to find bugs in virtual CPUs

Contribution

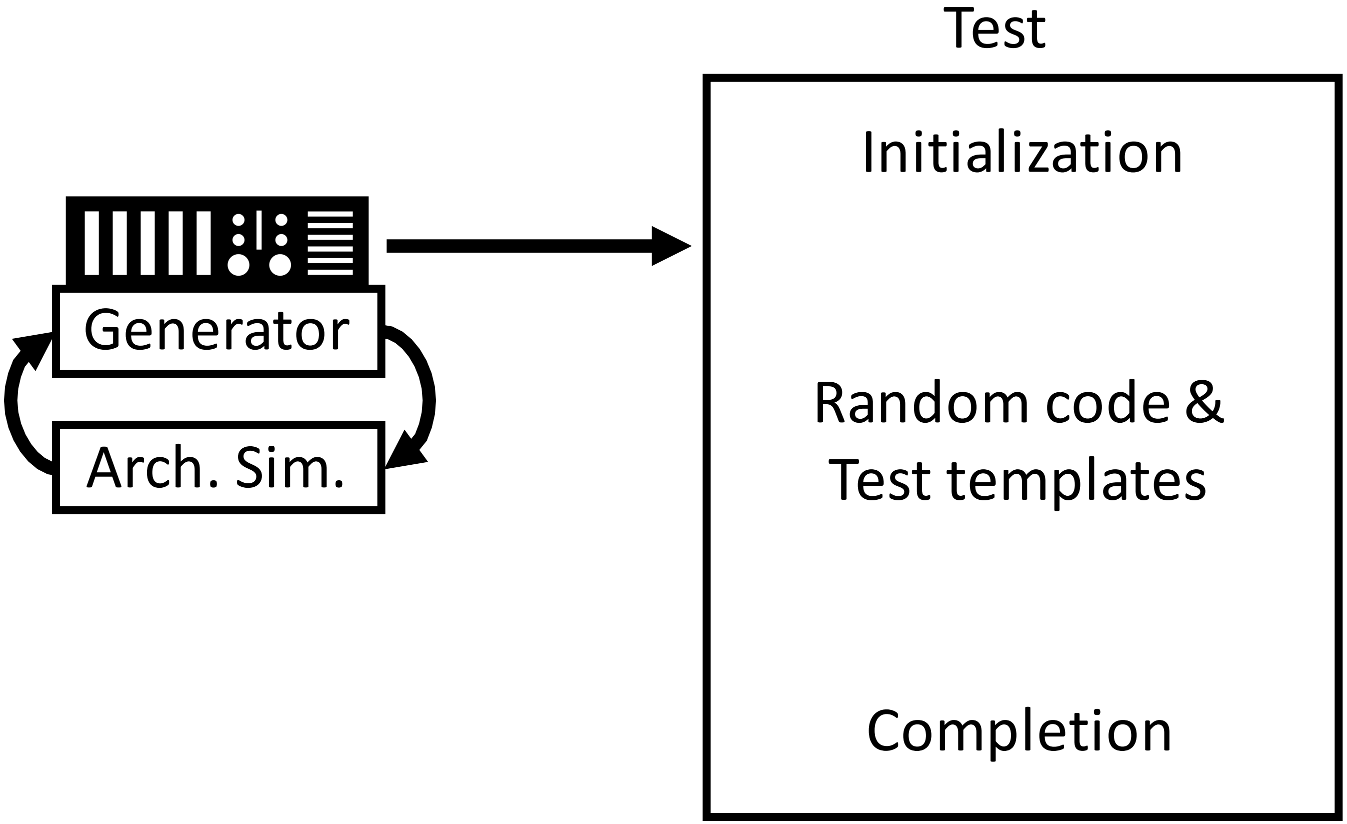
1. Adapt & apply physical CPU testing tools to VCPUs

2. Study hypervisor bugs
    - Found, fixed, and analyzed >100 bugs

Outline

- Motivation
- System
  - Physical CPU testing tools
  - Adapting tools to VCPUs
- Results
  - Causes of bugs
  - Impact of bugs
  - Architectural flaws (as opposed to SW bugs)
- Conclusions

Physical CPU Testing



Generator

Arch. Sim.

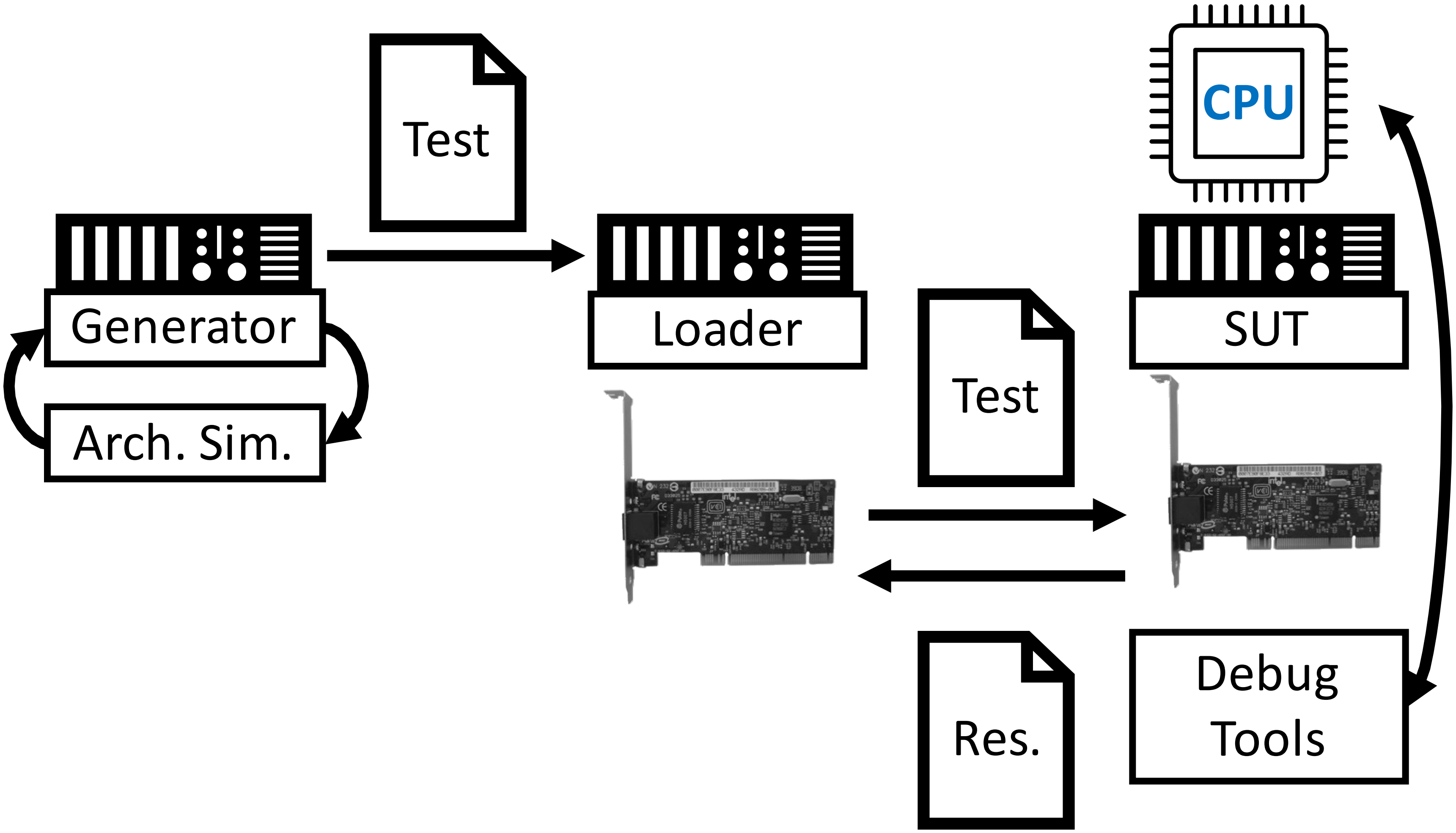
Test

Initialization

Random code &
Test templates

Completion

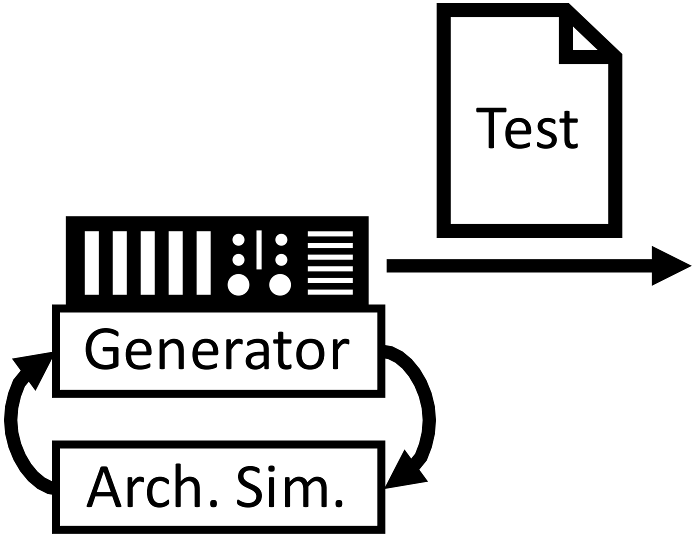
Physical CPU Testing



Benefits

- **High coverage**
  - Due to intimate architecture semantic awareness + effort
- **Low false-positive rate**
  - No undefined results of instructions
  - No nondeterministic results (due to errata or async events)
- **Easy to debug**
  - Interim checks
  - Detailed failure indications
  - Trace of expected architectural execution

Adaptation: Test Generation



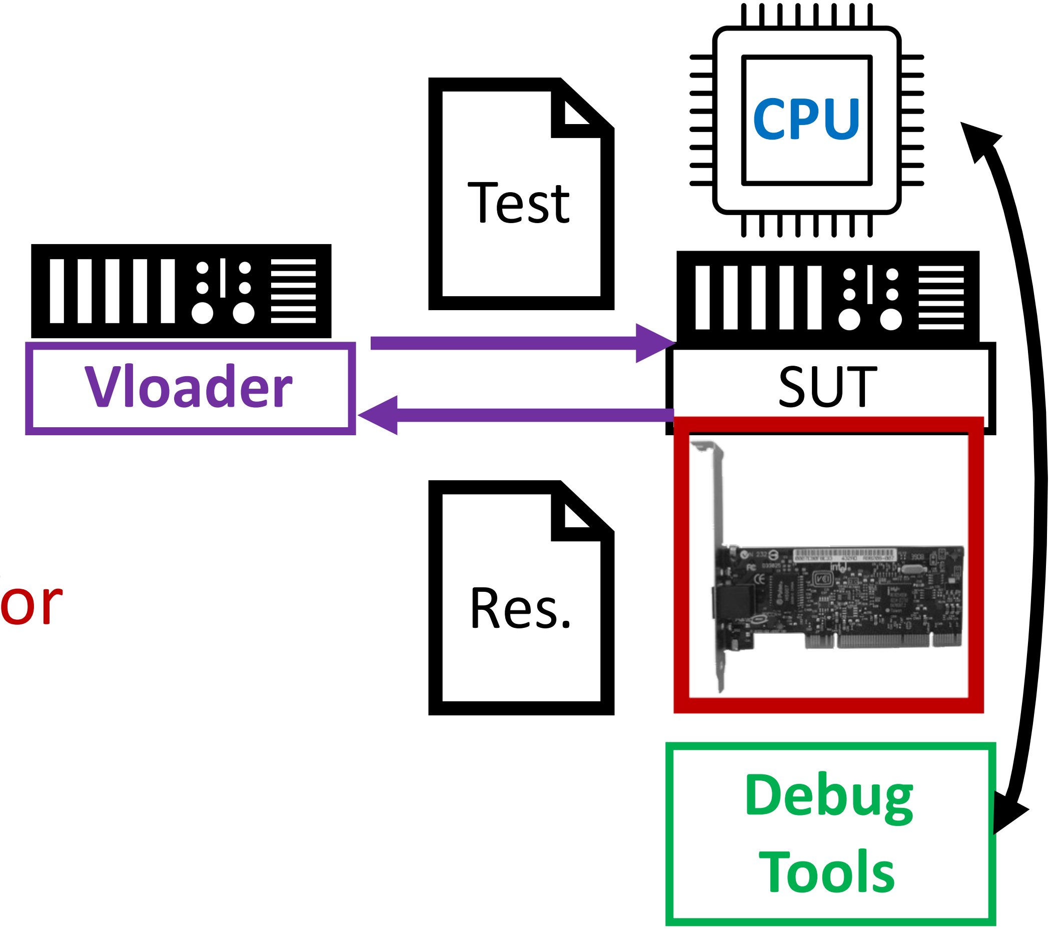
- Broken or missing virtualization features

- Add:
  - Cache-line monitoring
  - Performance Monitor Unit v3
  - …

- Workaround:
  - Nested virtualization
  - Data breakpoints
  - …

Adaptation: Execution and Debug

- Load tests using hypervisor monitor protocol
- Curb OS jitter
- Emulate test device for I/O instructions
- Enhance debug tools



Effort and Testing Time

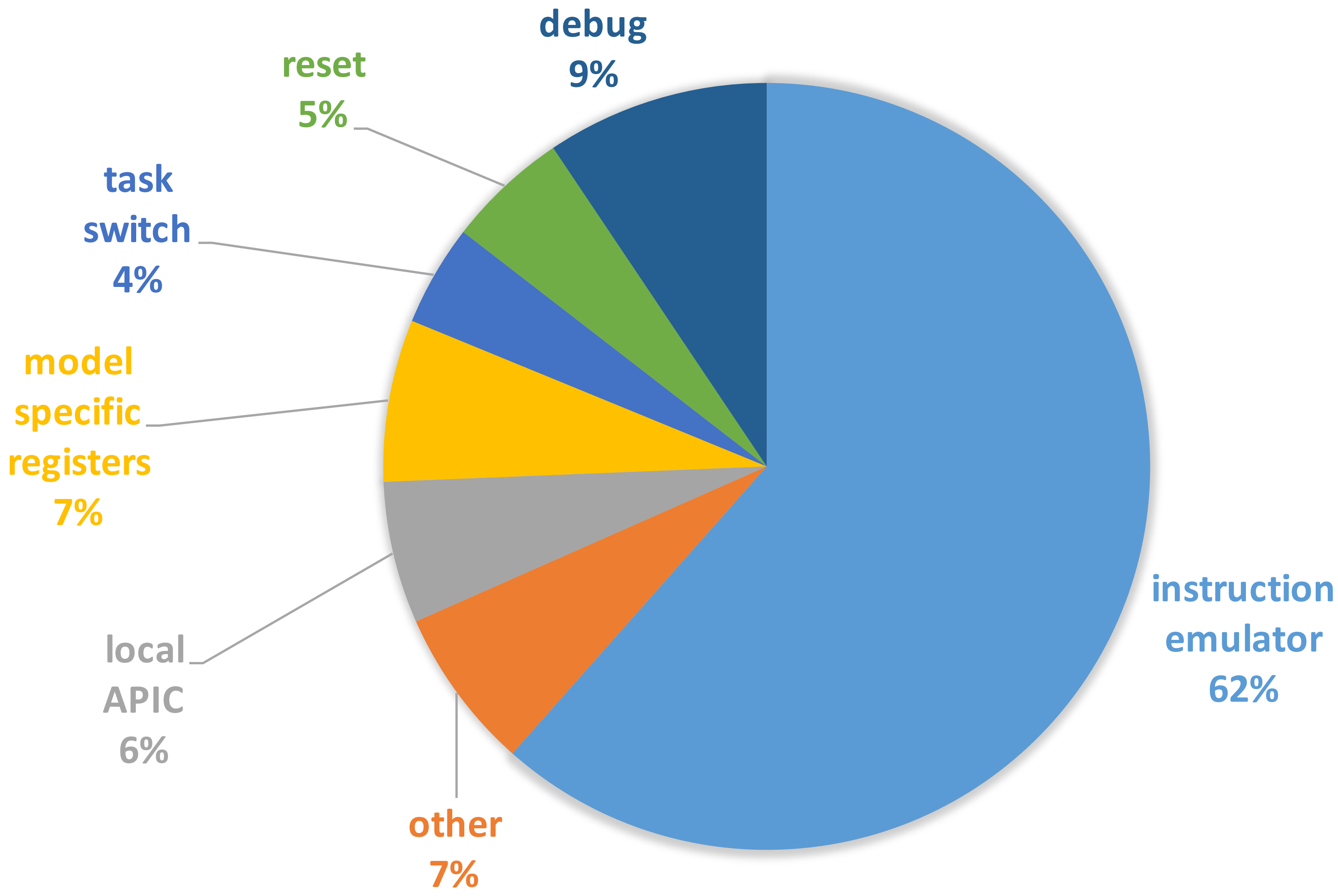
- Bootstrapping effort
  - 2 weeks to run the first empty test
  - 1.5 months to run the first full test

- Per-test time
  - Generation – 5 seconds
  - Execution – less than a second / 1MB
  - Failure debugging avg – ~3 hours (high var)

Outline

- Motivation
- System
  - Physical CPU testing tools
  - Adapting tools to VCPUs
- Results
  - Causes of bugs
  - Impact of bugs
  - Architectural flaws (as opposed to SW bugs)
- Conclusions

Testing KVM: 117 Bugs



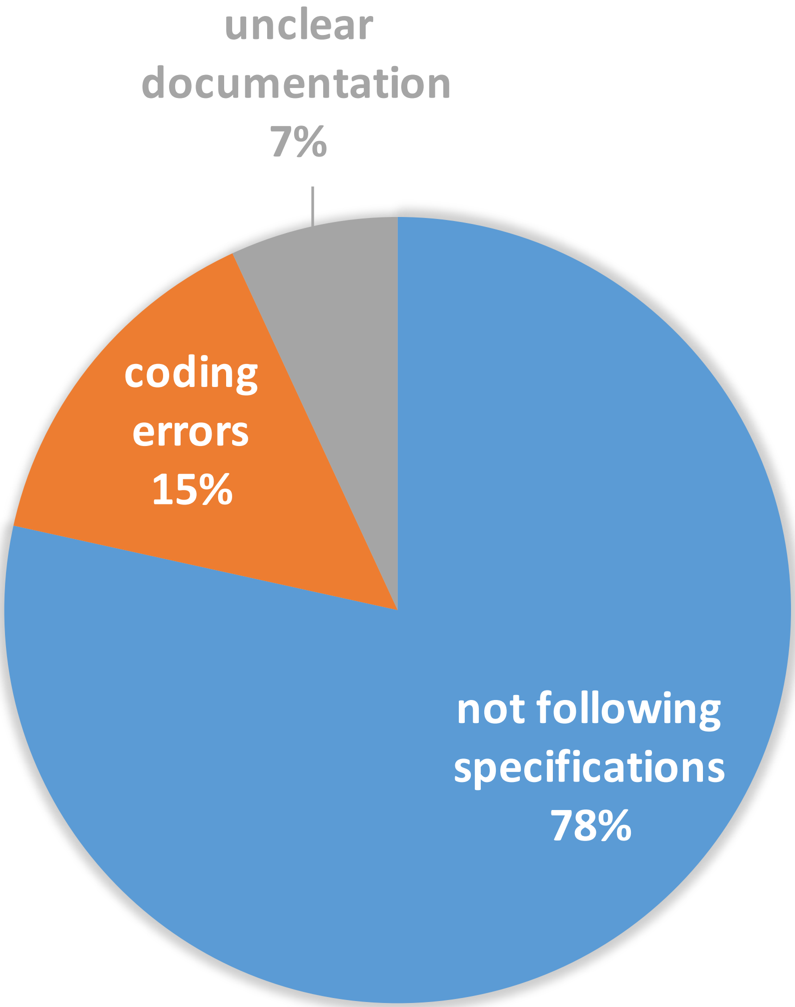
Instruction Emulator

- Why does a hypervisor need an instruction emulator?
  - **Port I/O** and **Memory Mapped I/O (MMIO)**
    Emulating instructions that access emulated devices
  - **Support for old hardware**
    Restricted guest; shadow page tables
  - **Vendor specific instructions**
    Migration between AMD and Intel

- Instruction emulator stress
  - Emulate every instruction
  - Run natively if emulation is unsupported

Bug Causes

- Mostly due to not following specifications
- Documentation can be improved
- Plain coding errors
  - Races
  - Null dereferences
  - Wrong error codes
  - Decimal/Hex



Implications: Security

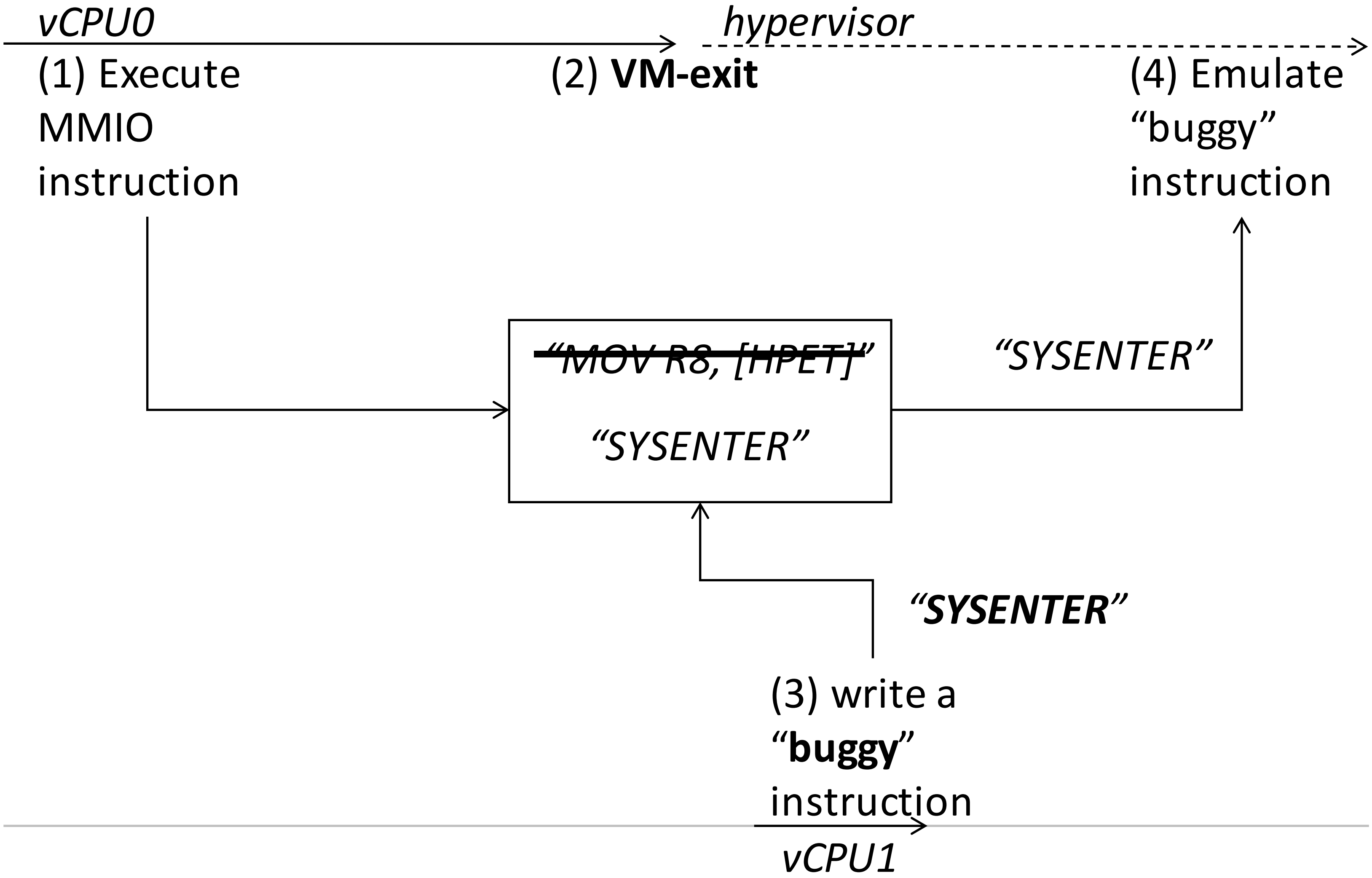
- 6 vulnerabilities

- Impact:
  - **Host** compromised: 3 host DoS
  - **VM** compromised: 2 VM DoS, 1 privilege escalation

- Main cause – **instruction emulator bugs**
  - x86 ISA consists of 800+ instructions
  - Usually, many instructions should not be emulated
  - But the hypervisor can be tricked to emulate them

Implications: Security - Example

Exploiting CVE-2015-0239 – potential privilege escalation



vCPU0

hypervisor

(1) Execute MMIO instruction

(2) **VM-exit**

(4) Emulate “buggy” instruction

~~“MOV R8, [HPET]”~~

“SYSENTER”

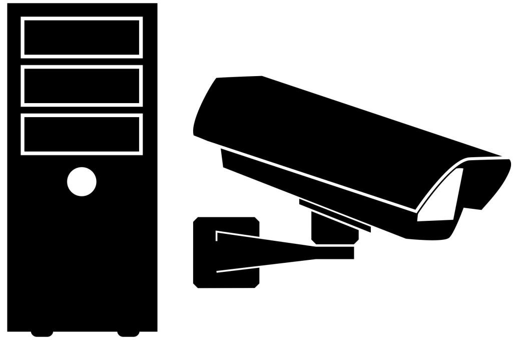
“SYSENTER”

“SYSENTER”

(3) write a “**buggy**” instruction

vCPU1

Implications: Stability



- Hard to quantify
- One bug caused virtual machines to freeze
  - Nontrivial race
  - Turns to be 5-year old bug
  - Was seen number of times over the years
- 4 additional software regressions

Hardware Flaws

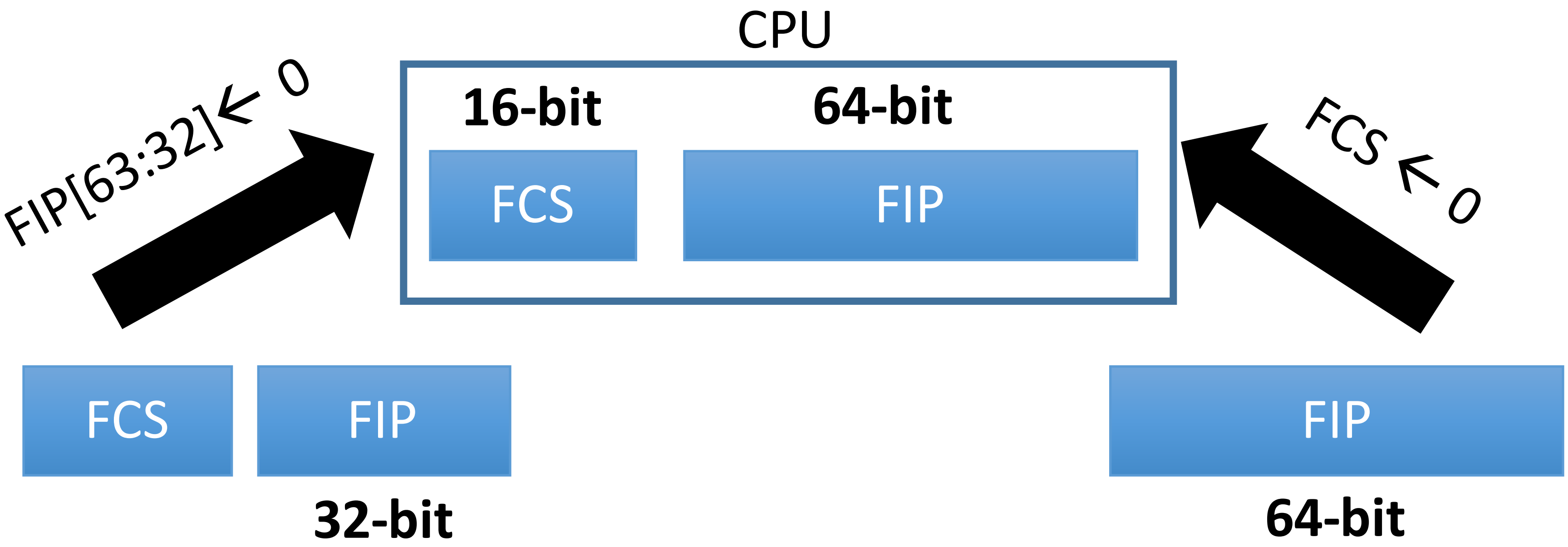
- Found 4 architecture flaws

- Desired virtual machine properties
  - Equivalence
  - Efficiency
  - Resource Control

Both cannot be kept

- Causes:
  - Non-virtualizable state
  - Missing state save/restore facilities
  - Errata

Hardware Flaw: FPU state



- Old CPUs: restore **either** 16-bit FCS or 64-bit FIP
- New CPUs: deprecate FCS save/restore
  New Problem in Real-Mode: **FIP = (FCS << 4) | FIP**

Outline

- Motivation
- System
  - Physical CPU testing tools
  - Adapting tools to VCPUs
- Results
  - Causes of bugs
  - Impact of bugs
  - Architectural flaws (as opposed to SW bugs)
- Conclusions

Conclusions

- Virtualization robustness/security should not be assumed

- CPU vendors are able to test hypervisors efficiently

- And it is in their best interest…

- **Demand it from your CPU vendor!**