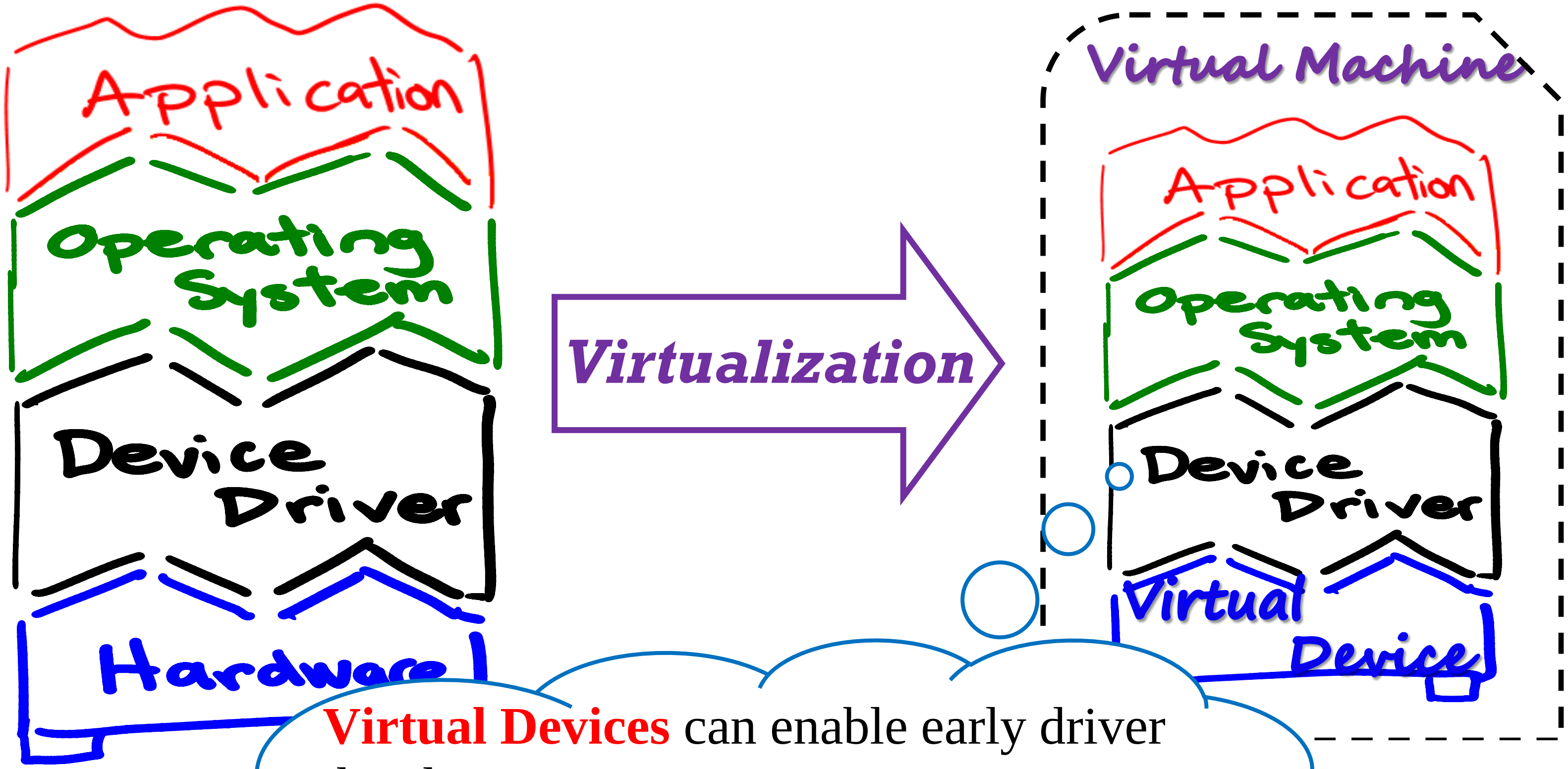
Symbolic Execution of Virtual Devices

Kai Cong

Portland State University

Virtual Devices to the Rescue



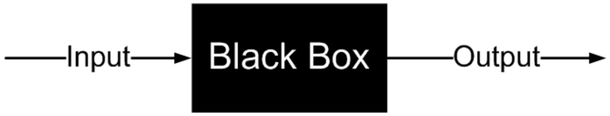
Virtualization

Virtual Devices can enable early driver development.

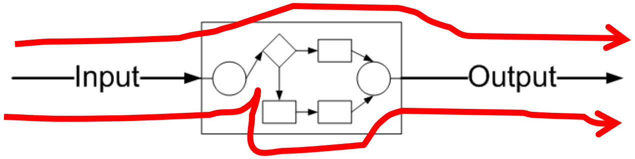
Is it possible to **bring more benefits** with virtual devices to help HW/SW development and validation???

Observability and Traceability

Real Device



Virtual Device

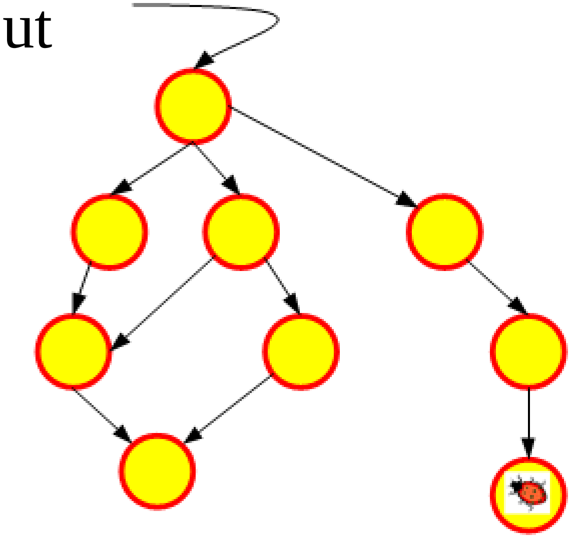


```
……

static uint32_t
e1000_mmio_readl(void *opaque, uint64 addr)
{
    E1000State *s = opaque;
    unsigned int index = (addr & 0x1ffff) >> 2;

    if (macreg_readops[index])
    {
        return macreg_readops[index](s, index);
    }
     return 0;
}

……
```

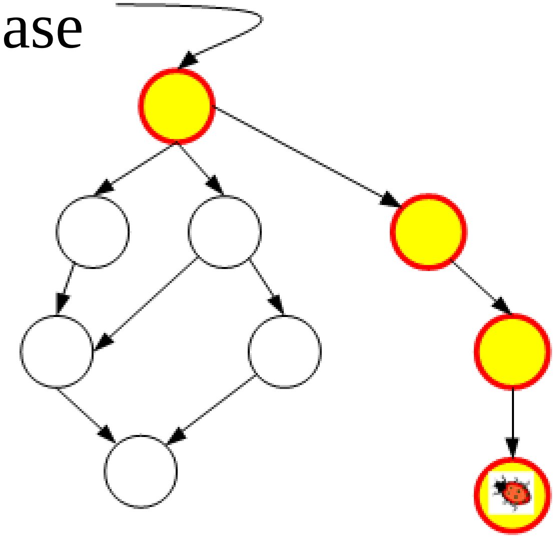
Symbolic Execution of Virtual Devices

- Execute virtual devices **symbolically**
- Enumerate as many **paths** as possible
- Generate and replay **concrete test cases**

Symbolic input



A concrete test case



Evaluation

- Applied to five QEMU virtual devices
- Most popular network adapters

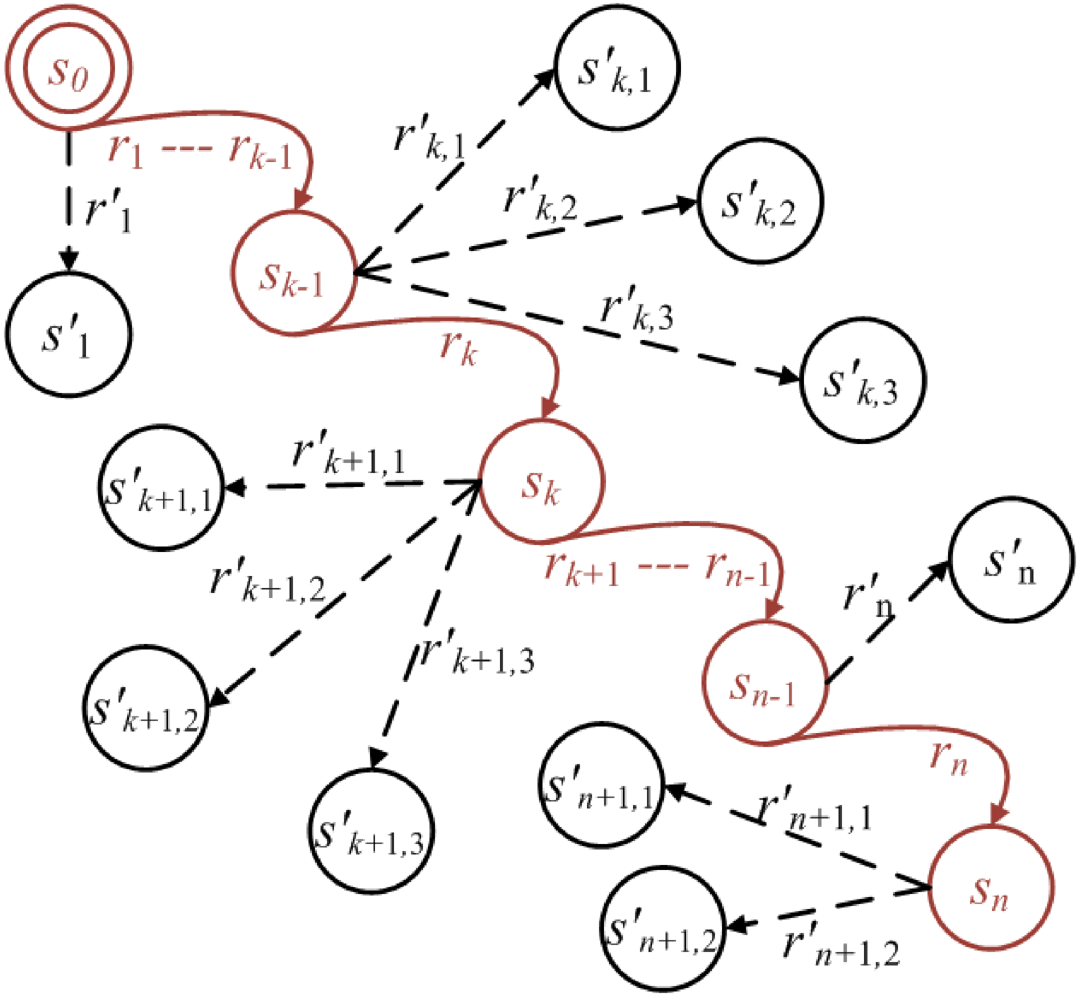
Virtual Device	Vendor	Description
E1000	Intel	Pro/1000 Gigabit Ethernet Adapter
EEPro100	Intel	Pro/100 Ethernet Adapter
PCNet	AMD	PCNet32 10/100 Ethernet Adapter
RTL8139	Realtek	PCI Fast Ethernet Adapter
Tg3	Broadcom	BCM57xx-based Gigabit Ethernet Adapter

Evaluation

- Experiment setup: 8-core i7 CPU, 8 GB of RAM, and Ubuntu Linux 64-bit
- Five configurations with different loop bounds and time bounds

	Config 1 Loop bound: 1 Time: 150 sec		Config 2 Loop bound: 1 Time: 300 sec		Config 3 Loop bound: 1 Time: 600 sec		Config 4 Loop bound: 2 Time: 300 sec		Config 5 Loop bound: 3 Time: 600 sec	
Device	Paths	Memory (MB)	Paths	Memory (MB)	Paths	Memory (MB)	Paths	Memory (MB)	Paths	Memory (MB)
E1000	289	401	440	721	671	1614	406	803	505	1622
EEPro100	183	131	499	254	1539	508	468	238	1005	483
RTL8139	371	66	402	131	408	238	404	131	414	262
PCNet	279	74	424	139	646	262	417	139	601	262
Tg3	252	811	391	1196	556	4104	398	1581	569	3162

Application Example: Test Generation



Concrete execution

+

Symbolic execution

Automatic Concolic Test Generation with Virtual Prototypes for Post-silicon Validation. In *ICCAD*, 2013.

Conclusions and Future Work

- Presented an approach to symbolic execution of virtual devices, central to achieving observability and traceability.

- Application example
  - Concolic Test Generation for Post-silicon Validation

- Future work
  - Algorithms for setting loop bounds adaptively
  - Utilization of symbolic execution of virtual devices in run-time fault injection and test coverage computation

Thanks!