CompSC: Live Migration with Pass-through Devices

ZHENHAO PAN [&], YAOZU DONG [*], ***YU CHEN*** [&],

LEI ZHANG[&], ZHIJIAO ZHANG[&],

[&]Tsinghua University, [*]Intel Asia-Pacific Research and Development Ltd.

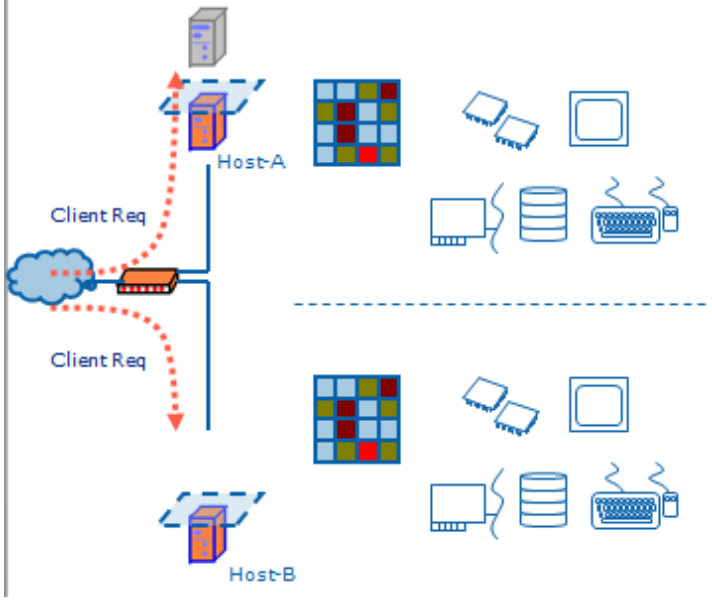
VEE 2012

Outline

- **Introduction**
- CompSC solution
- Experiments
- Conclusion

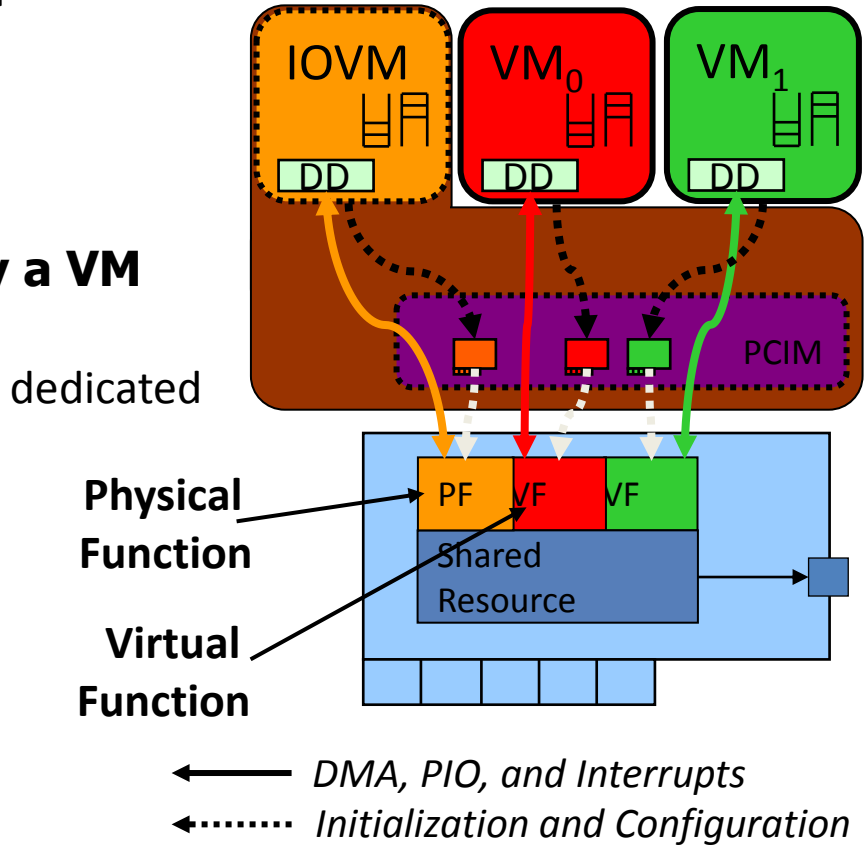
Introduction

- Background
  - Live migration
  - Pass-through device
  - SR-IOV spec
- Experimental result
  - Liv migration with SR-IOV NIC
    - 282.66% more throughput
    - 42.9% less downtime



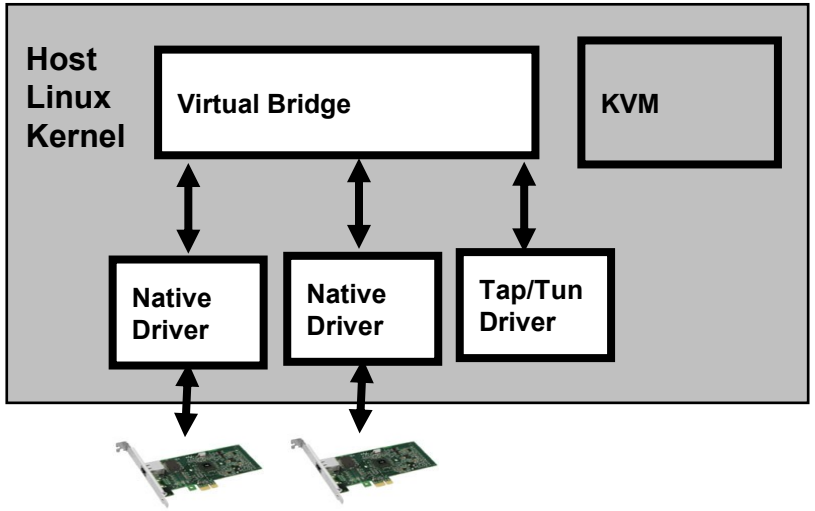
Introduction

- SR-IOV Specification
  - **Start with a single function device**
    - **HW under the control of privileged SW**
    - **Includes an SR-IOV Extended Capability**
    - **Physical Function (PF)**
  - **Replicate the resources needed by a VM**
    - **MMIO for direct communication**
    - **RID to tag DMA traffic**
    - **Minimal configuration space**
    - **Virtual Function (VF)**
  - **Introduces PCI Manager (PCIM)**
    - **Conceptual SW entity**
    - **Completes the configuration model**
    - **Translates VF into a full function**
    - **Configures SR-IOV resources**

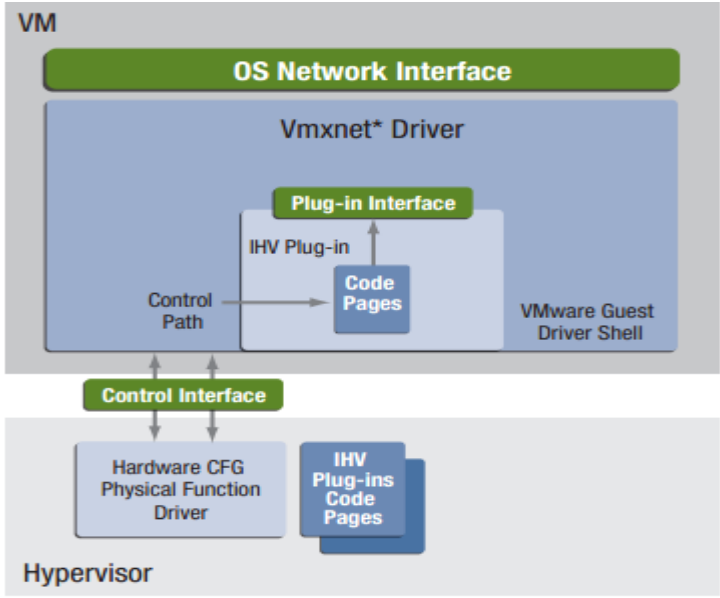


Related work

- Bonding driver [Linux Ethernet Bonding Driver HOWTO]
  - Failover/Load balance
- NPIA (Network Plug-in Architecture)



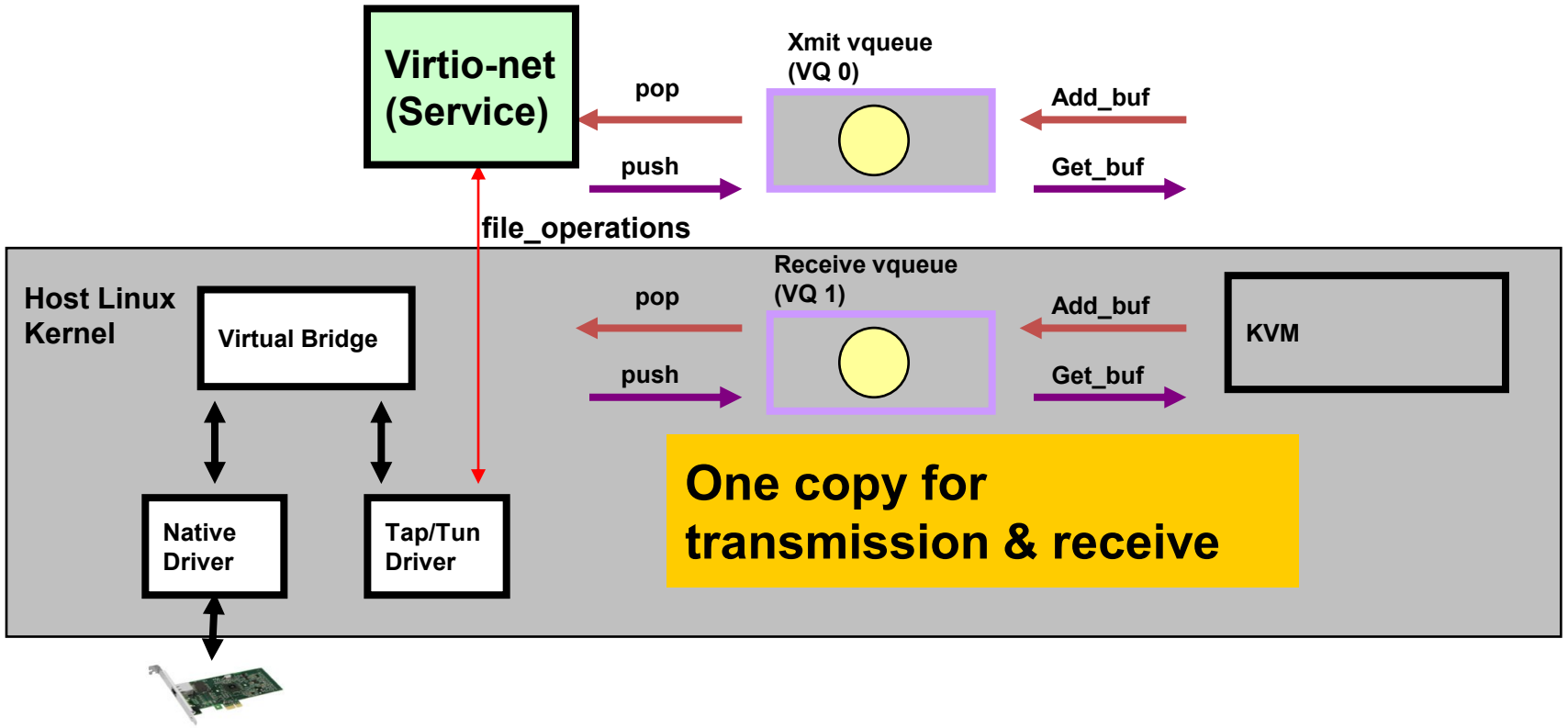
Bonding driver



NPIA

Related work

- VMDq (Virtual Machine Device Queue)
  - Multiple queue pairs for partitioning



Why not store/restore device states directly?

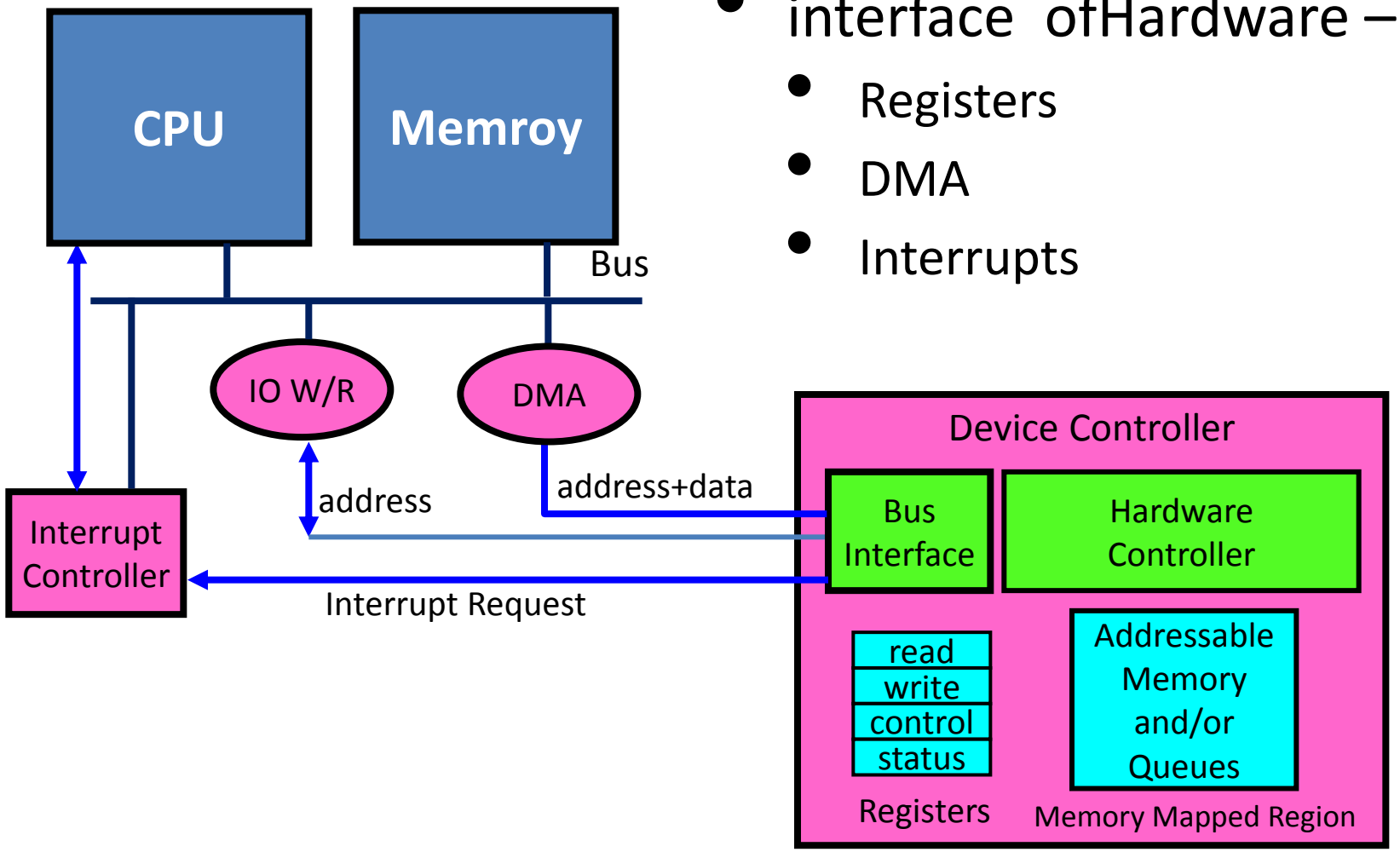
Outline

- Introduction
- **CompSC solution**
- Experiments
- Conclusion

CompSC Approaches

- Requirement challenges
  - The state (such as registers) of the device needs to be efficiently read and written to support device state replication;
  - The dirty memory written by the device Direct Memory Access (DMA) needs to be efficient and tracked for lazy memory state transmission.

CompSC Approaches



- interface ofHardware – OS
  - Registers
  - DMA
  - Interrupts

CompSC Approaches

- Requirement of I/O Register migration :
  - Most parts: Read/write, No side effect
  - Some special: RO/WO, RC/WC, etc., with side effect

Register type	Description
read-write	If written since reset, the value read reflects the value written.
read-only	Writes to this reg have no effect.
write-only	Reading this reg returns no meaningful value.
read-write-clear	A register can be read and written. However, a write of a 1b clears the corresponding bit.
write-clear	Writing 1b to register clears an event possibly reported in another register.
read-clear	A register bit with this attribute is cleared after read. Writes have no effect on the bit value.
read-write-set	Register that is set to 1b by software, and cleared to 0b by hardware.
reserved	Reserved field can return any value on read access and must be set to its initial value on write access.

CompSC Approaches

State replay for side effect

- Method
  - Record every hardware access (Recording stage)
  - Replay them on the target device (Replaying stage)
- Optimization 1
  - Record last reg writing when this writing brings no side effect
- Optimization 2
  - Define operation sets (op set), the op sets is Critical Section

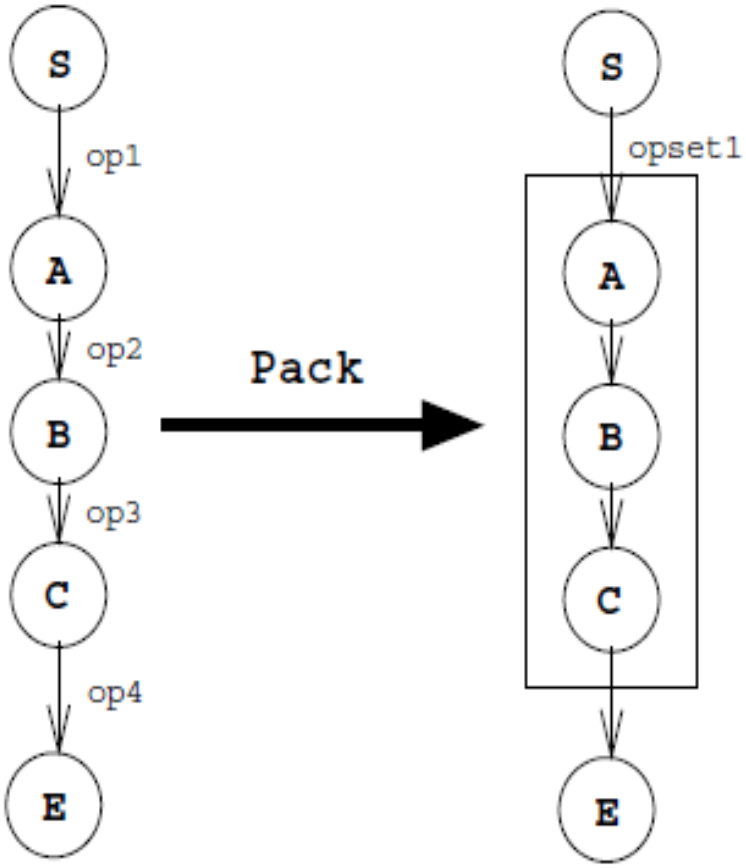
CompSC Approaches

State replay – with op set

Op sets in Intel 82576/82599 NIC

- All **initializing** operations
- All **Sending** operations
- All **Receiving** operations
- other remaining op states include only {**uninitialized**, **up**, **down**}

In this kind of set up, only the latest operations on each setting register and whether or not the interface is up need to be tracked.



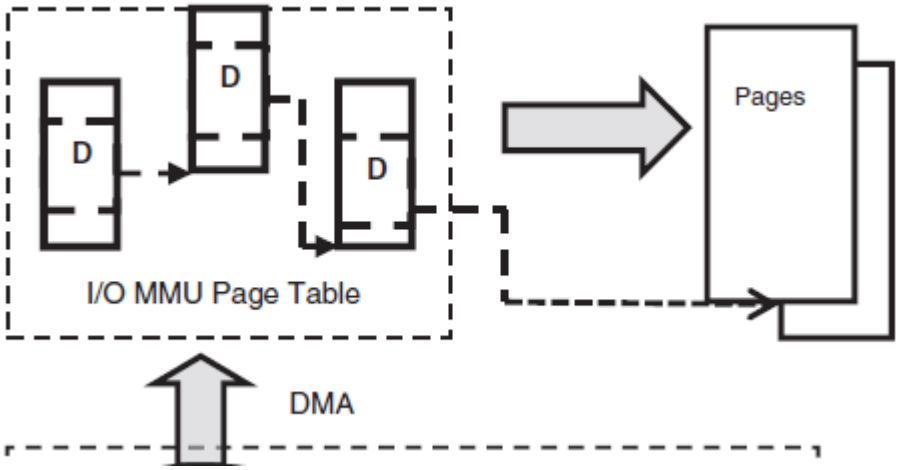
CompSC Approaches

Self-emulation for Read-only, etc. Regsiters

- Design for statistic registers (read-only/read-clear)
- Require mathematical attributes (monotonicity)
- Example: dropped packets counter
  - = n before migration
  - initialized to 0 when migration
  - = m now (after migration)
  - correct value= n + m

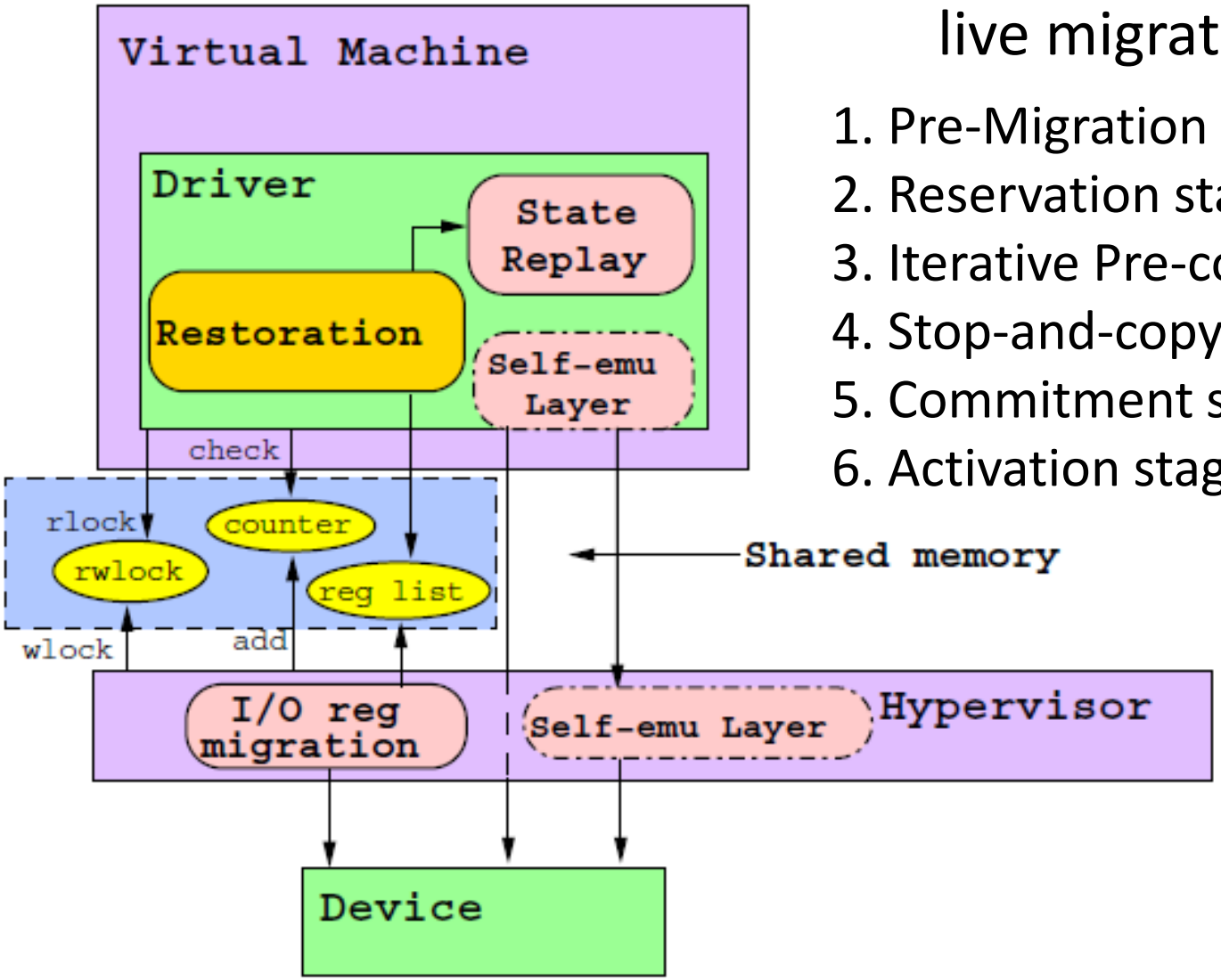
CompSC Approaches

- Dummy writing for DMA dirty page
  - ***DMA dirty page tracking***. To replicate the I/O state, memory pages modified by the device DMA operations must be efficiently tracked for efficient live migration. Unfortunately, DMA dirty page tracking is not supported in the existing I/O MMU.
  - Dummy write the DMAed page after DMA process finished.



CompSC Architecture

Design & Implemenatation



live migration

1. Pre-Migration stage
2. Reservation stage
3. Iterative Pre-copy stage
4. Stop-and-copy stage
5. Commitment stage
6. Activation stage

CompSC Implementation

Xen and SR-IOV NIC drivers

Implementation complexity ~2000 LoC

	Line of code
Xen hypervisor	362
Xen tools	446
VF driver(common)	153
IGBVF driver	344
IGB driver	215
IXGBEVF driver	303
IXGBE driver	233

CompSC Implementation

Xen and SR-IOV NIC drivers

- Intel 82576 Gbps NIC & 82599 10Gbps NIC
  - PF/VF drivers
- Driver changes on IGBVF/IXGBEVF
  - Rlock every hardware operation
  - Pack igbvf_up/igbvf_down and ixgbevf_up/ixgbevf_down into operation sets
  - Restoration after migration

CompSC Implementation

Xen and SR-IOV NIC drivers

- Shared memory for sync
  - rw-lock and version counter
  - List of registers for I/O register migration
  - List of registers for self-emulation
- Synchronization for Live Migration
  - Acquire w-lock before suspending
  - Increase version counter
  - Release w-lock after migration
  - Invoke driver restoration at first r-lock

CompSC Implementation

Xen and SR-IOV NIC drivers

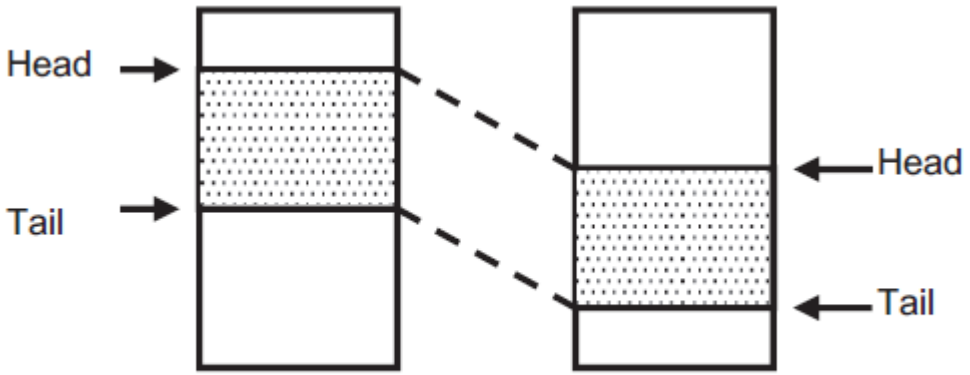
- Pages dirtied by DMA
  - In x86/x64, memory access by DMA cannot be tracked on page tables by MMU, IOMMU
  - In CompSC, driver performs dummy writes to descriptor/buffer when receive an interrupt
  - May cause packet miss/packet duplication during migration

	Dup	Miss
No workload	0	0
scp	0	0
SPECweb	0	3

CompSC Implementation

Xen and SR-IOV NIC drivers

- Descriptor ring
  - Descriptor ring head index is in read-only register
  - Altering head index is hard (hard for state replay)
  - CompSC introduces an offset between the ring in hardware's view and software's view
  - During migration, increase the offset to make sure ring head index on target hardware is 0



Outline

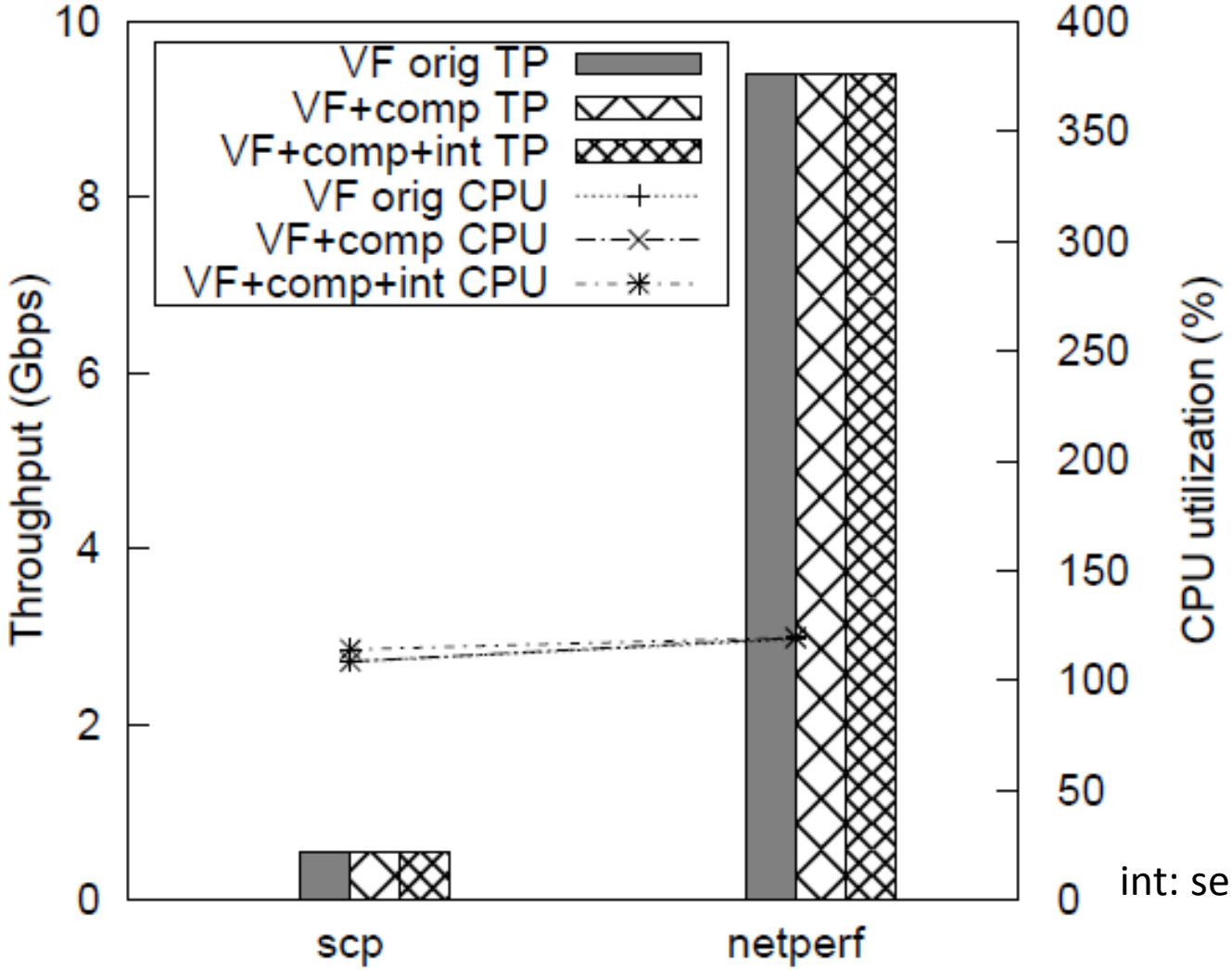
- Introduction
- CompSC solution
- **Experiments**
- Conclusion

Experiments

- Physical Environment
  - Intel Core i5 670 (with VT-x, VT-d, VT-c features)
  - 4GB memory, 1TB hard disk
  - Intel 82576 & Intel 82599 NICs
- Virtual Environment
  - 4 vCPU
  - 3GB memory
  - PF/VF of Intel 82576 or Intel 82599 NIC

Experiments
Evaluation - Throughput

Intel 82599

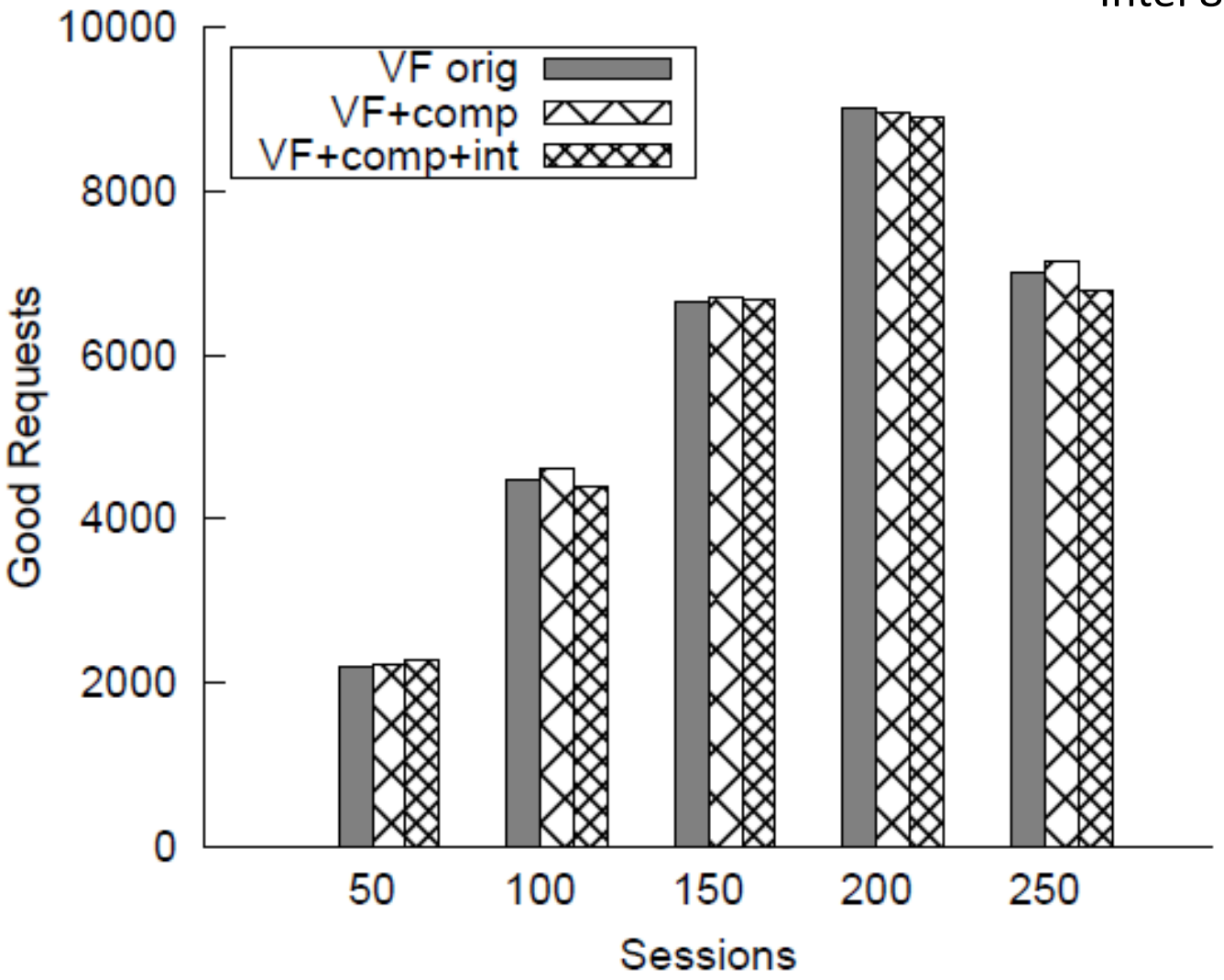


9. 4 Gbps

int: sel f-emulation

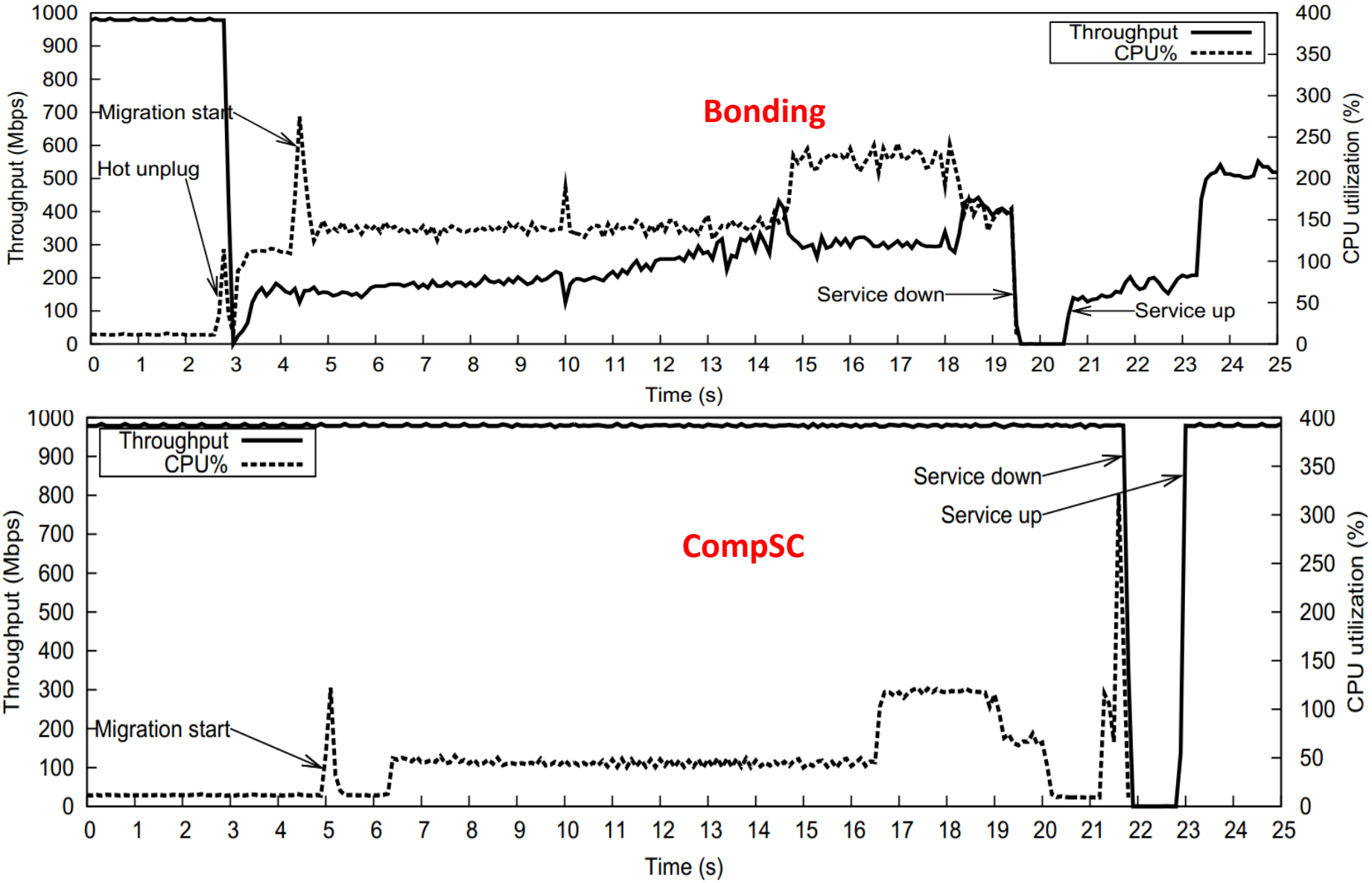
Experiments
Evaluation - SPECweb2009

Intel 82599



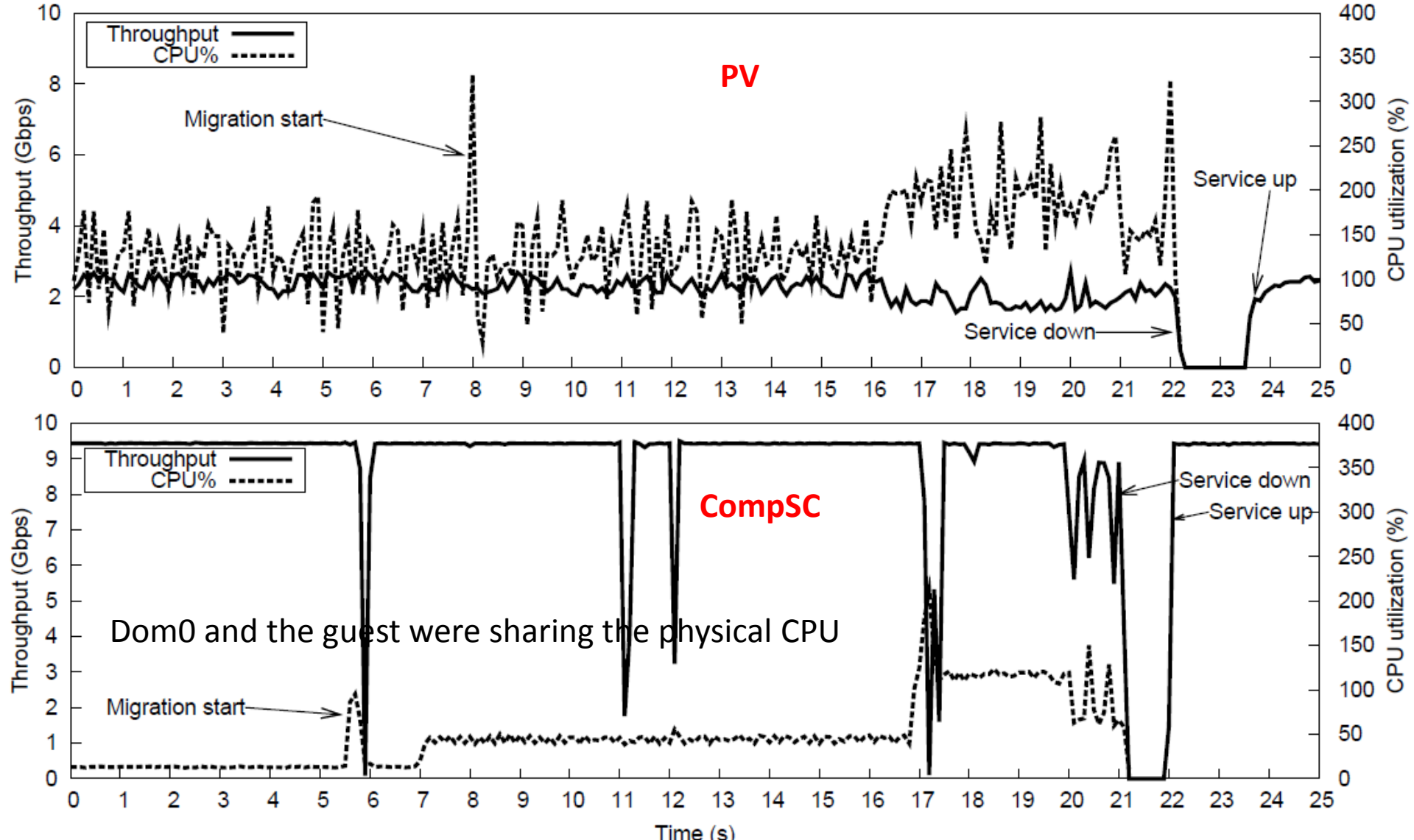
Experiments

Evaluation - Live migration(Bonding v.s. CompSC), Netperf 82576 NIC



Experiments

Evaluation - Live migration(PV v.s. CompSC), Netperf 82599 NIC



Consolution

- Proposed a directly solutons for live migration of pass-through device : CompSC
  - Support Live Migration with SR-IOV NIC
- Future
  - Evaluate NPIA method
  - Support Checkpoint (such as Remus in XEN)
  - Other SR-IOV devices
  - …

Thank you for your attention!

Questions?