# DQEMU: A Scalable Emulator with Retargetable DBT on Distributed Platforms

**Ziyi Zhao**, Zhang Jiang, Ximing Liu, Xiaoli Gong*
Nankai University

Wenwen Wang
University of Georgia

Pen-Chung Yew
University of Minnesota



# Dynamic Binary Translation(DBT)

*"A Key Enabling Technology"*

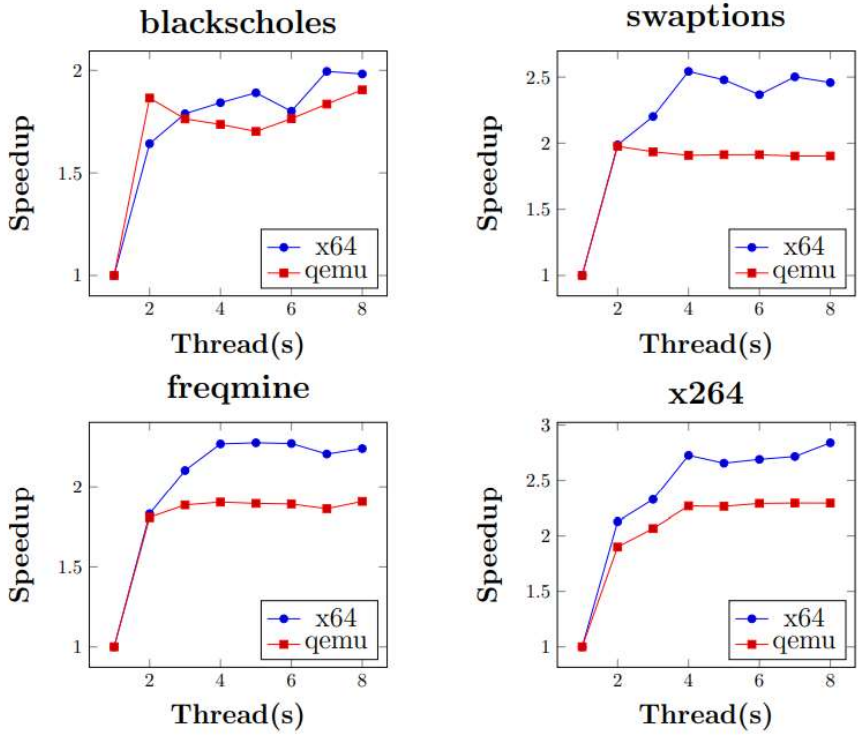
---

**Cross-ISA Virtualization** **Dynamic Instrumentation**

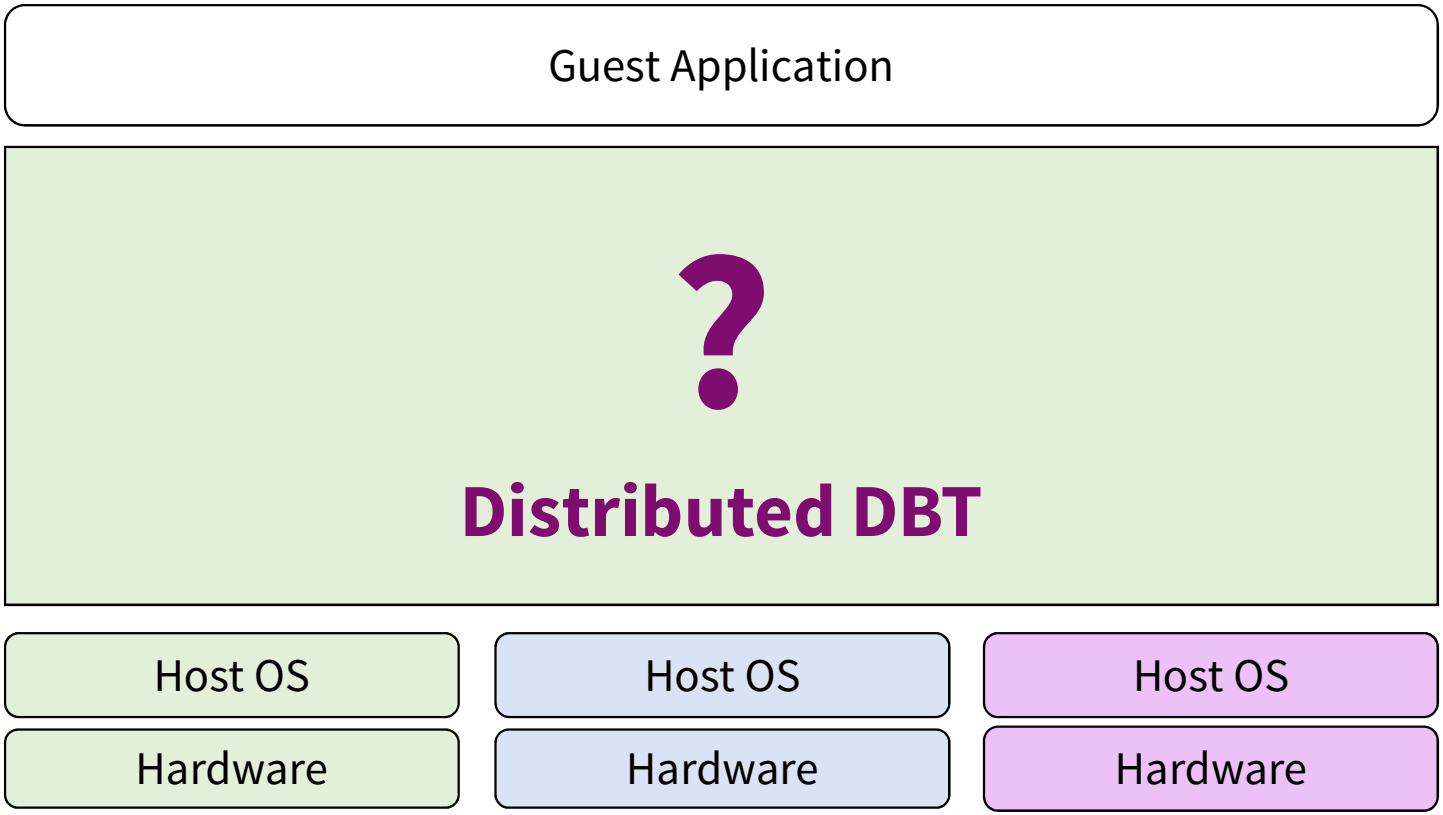
# The scalability of DBT is limited by computing resources

- QEMU is a trending DBT
- Parallel programs from PARSEC
- On x64 dual-core machine

Saturate around speedup of 2.0x



# Goal: Enable DBT to utilize compute resources across nodes

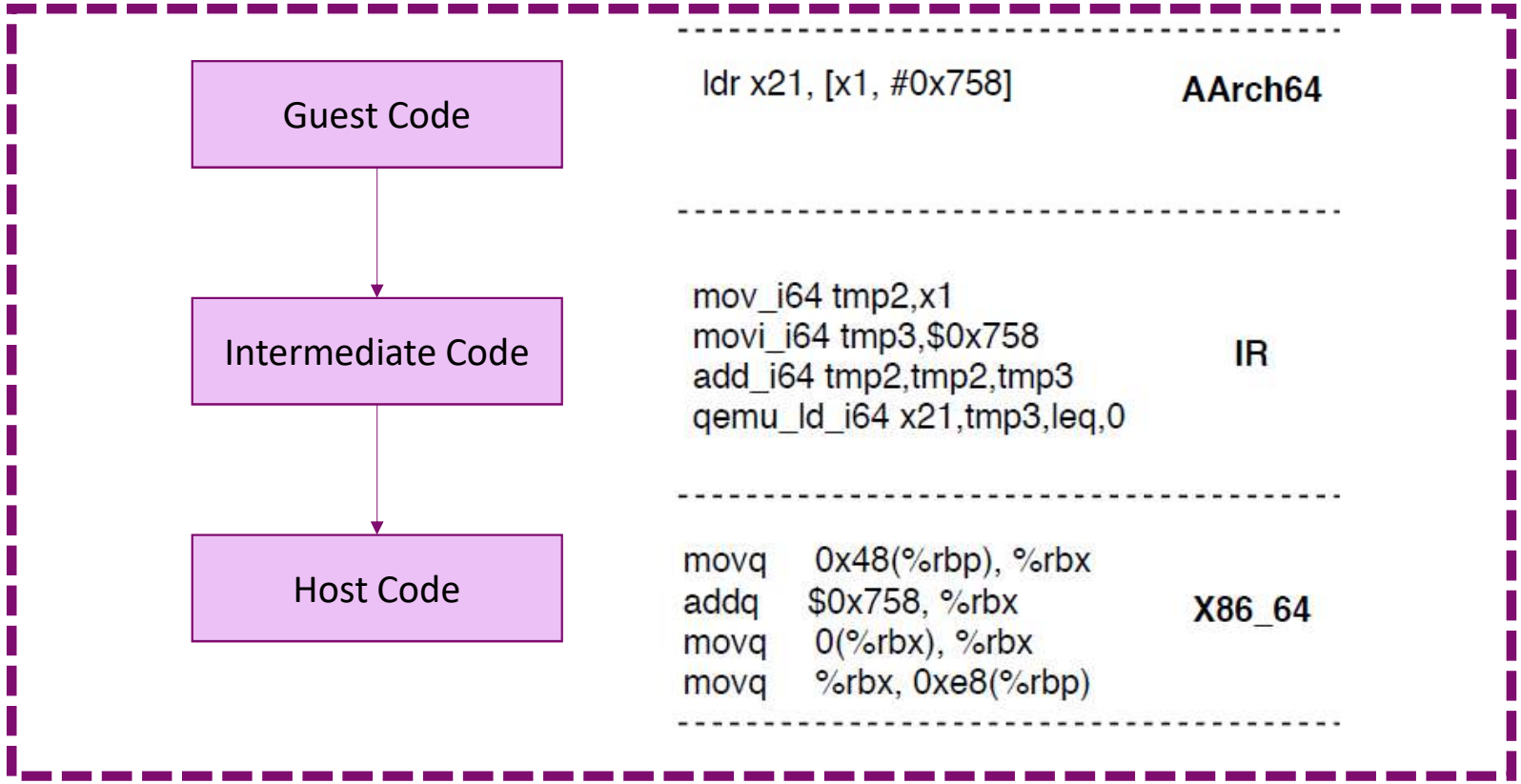


# Goal: Enable DBT to utilize compute resources across nodes

In a **distributed** emulator...

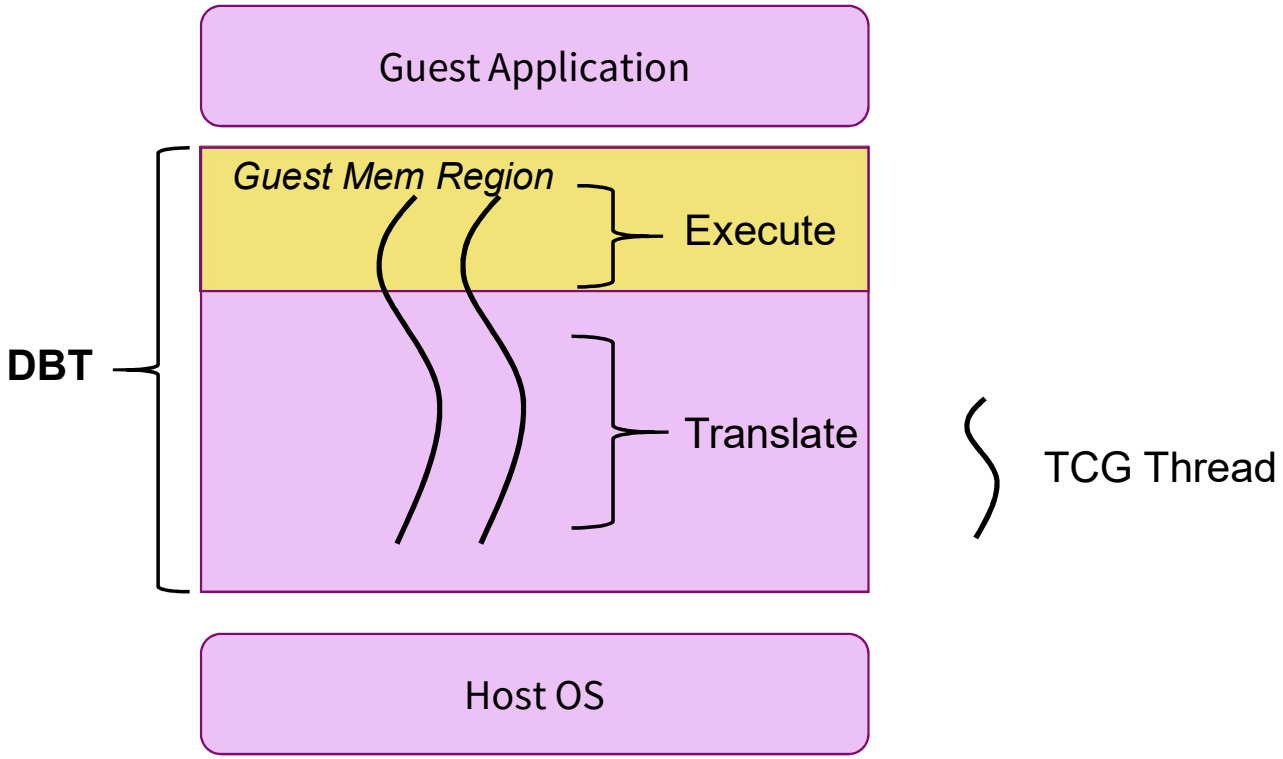
- How to maintain *guest* **cache coherence**?
  - Transparently
- How to emulate *guest* **system calls**?
  - Side effect to host kernel
- How to emulate *guest* **atomic operations**?
  - Equivalent atomic sematic between RISC←→CISC

# How does DBT work?

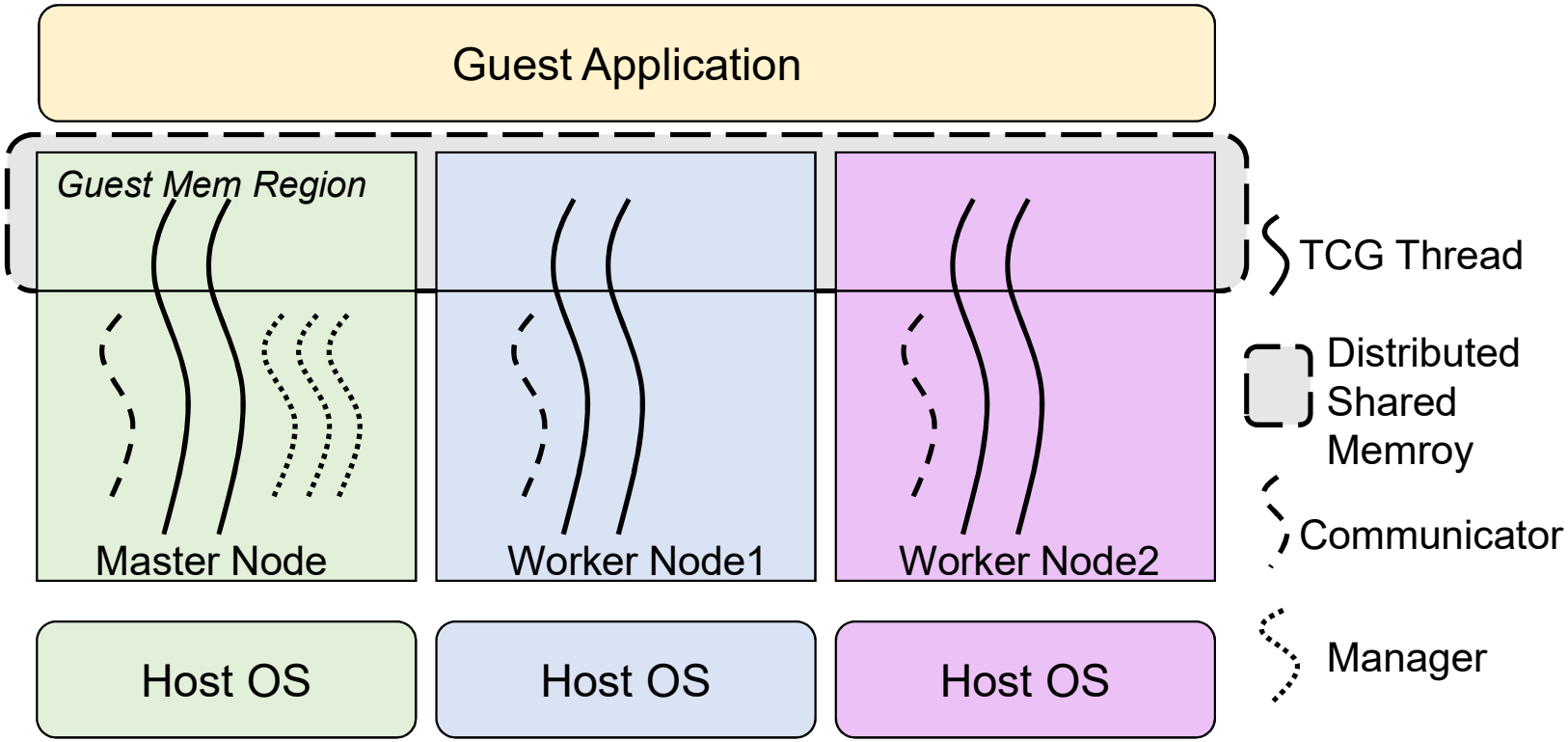



**T**iny **C**ode **G**enerator (TCG)

# How does DBT work?



# What should Distributed DBT looks like?

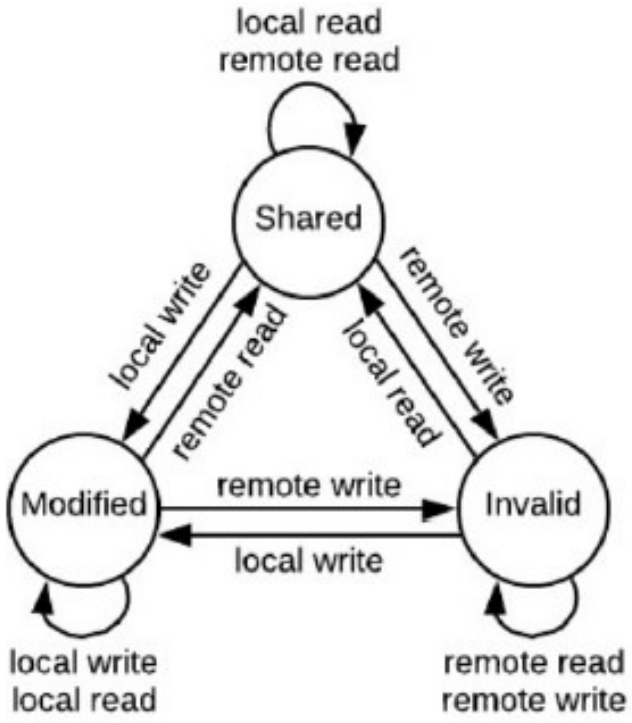


# How to keep cache coherence?

For the Distributed Shared Memory Region...

- At what granularity?
  - Cache line size? Page size? Larger?
- How to check privilege?
  - Software-based instrumentation: check on every memory access
  - Hardware-based: MMU – host page level check
- Which type of protocol?
  - Distributed / Centralized
  - MSI

# How to keep cache coherence?

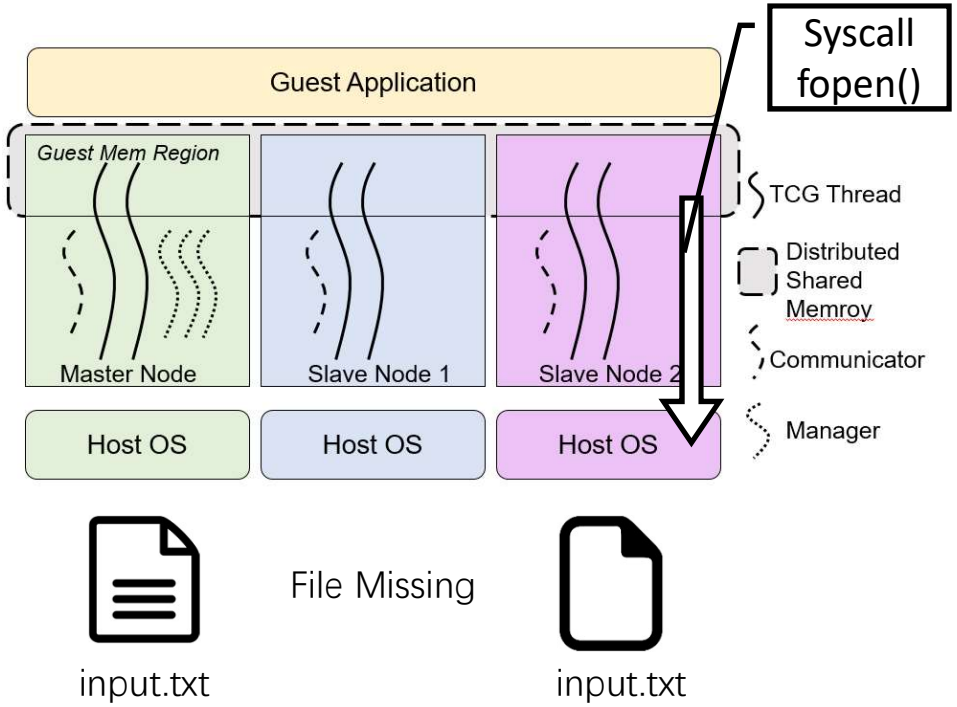



- Utilize host MMU to do state check

| State | Page Protection |
|---|---|
| **M**odified | RW |
| **S**hared | R- |
| **I**nvalid | -- |

- Synchronize granularity = 4K(host page size)

# The problem of system calls




- Eg. *fopen*() by a worker thread at node#2 affects
  - User-space file descriptor
  - Kernel-space resource manager
- Syscalls also affects host kernel

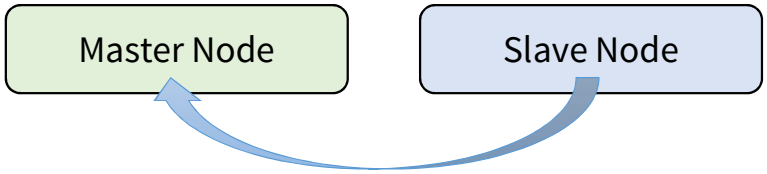
# The problem of system calls – Syscall Delegate

## Global Syscall

read, write, openat, open, fstat, close, stat64, lstat64, fstat64, futex, writev, brk, mmap2, mprotect, madvise, mumap, **clone**, vfork, **futex**

## Local Syscall

gettimeofday, clock_gettime, exit, nanosleep, ... all the rest




- Syscall parameters
- Guest CPU state

# The emulation of atomic operations

RISC

ARM, MIPS...

**LL(Load-linked)**
**SC(Store-conditional)**

Translate?

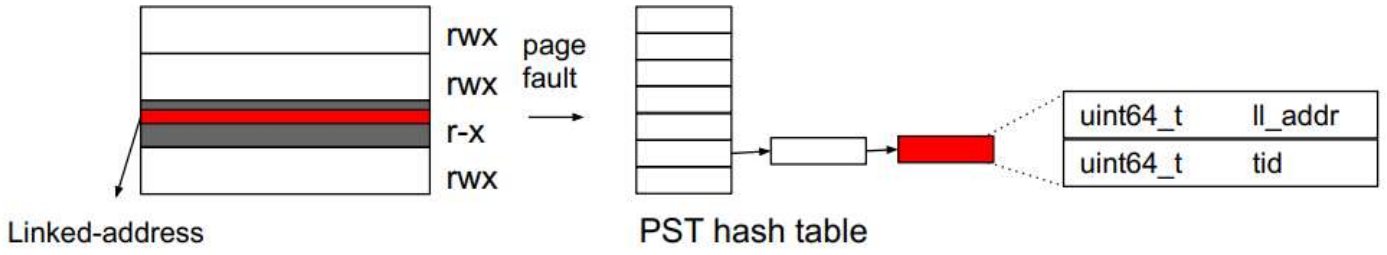
CISC

x86

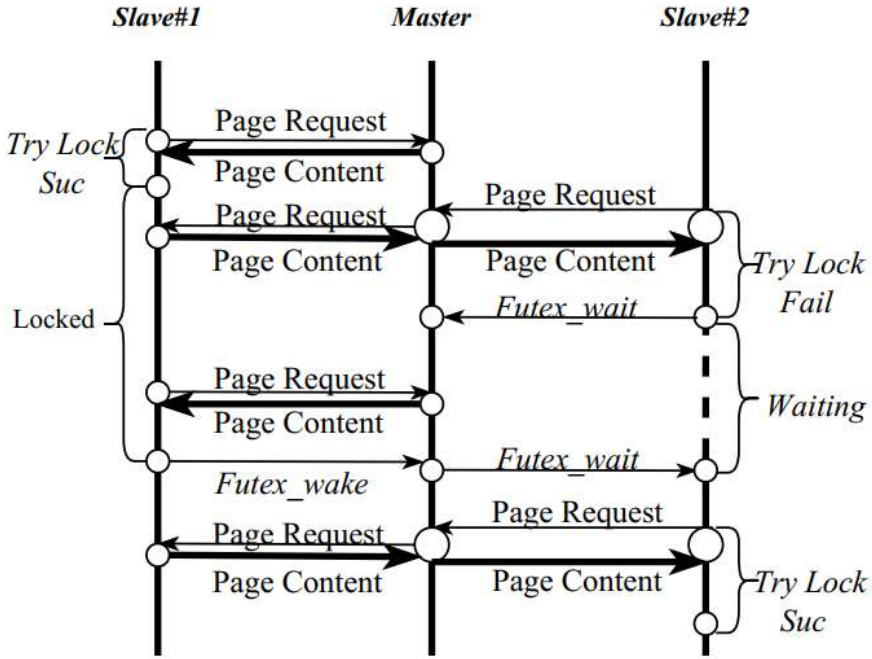
**CAS(Compare and Swap)**

(1)Page privilege check by MMU

(2)Check if ll_addr equals pf_addr, clean the entry if equal, else it is a false alarm.



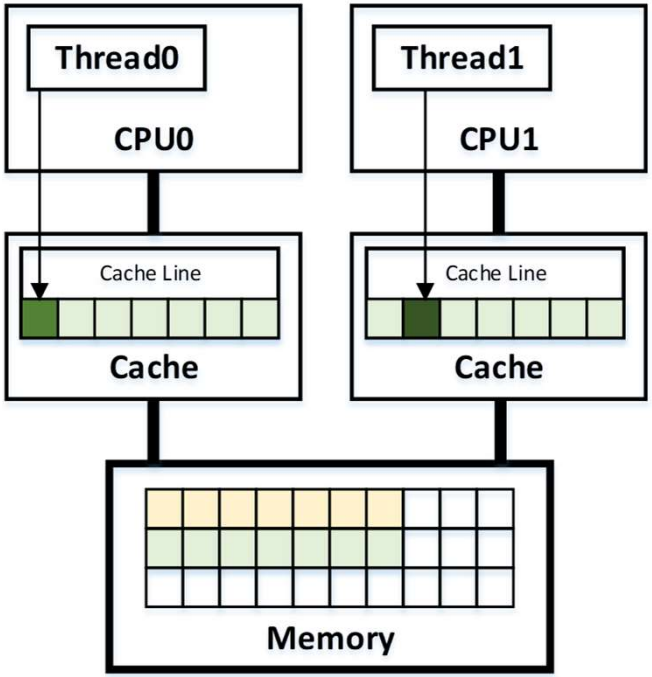
# The emulation of atomic operations




**Hierarchical lock**

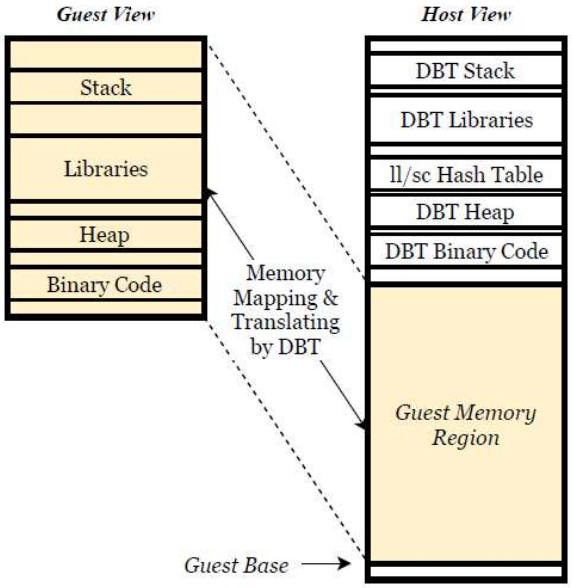
1. Intra-node: Consistency model translation[ArMOR]
2. Inter-node: MSI Coherence Protocol – Sequential

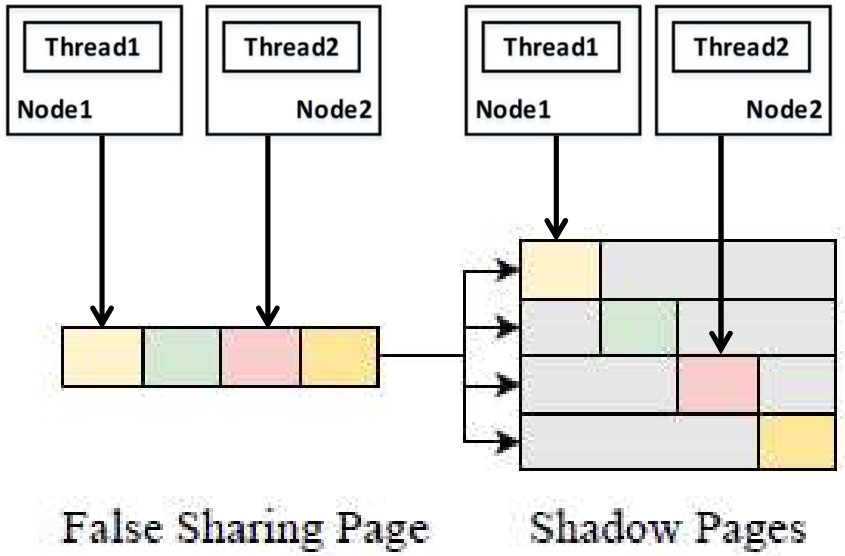
# Page Split: The false sharing overhead




- **Probability**: cache line size **64B** → page size **4096B**
- **Cost**: cache miss **23 cycles** → network + pagefault >= **120000cycles**

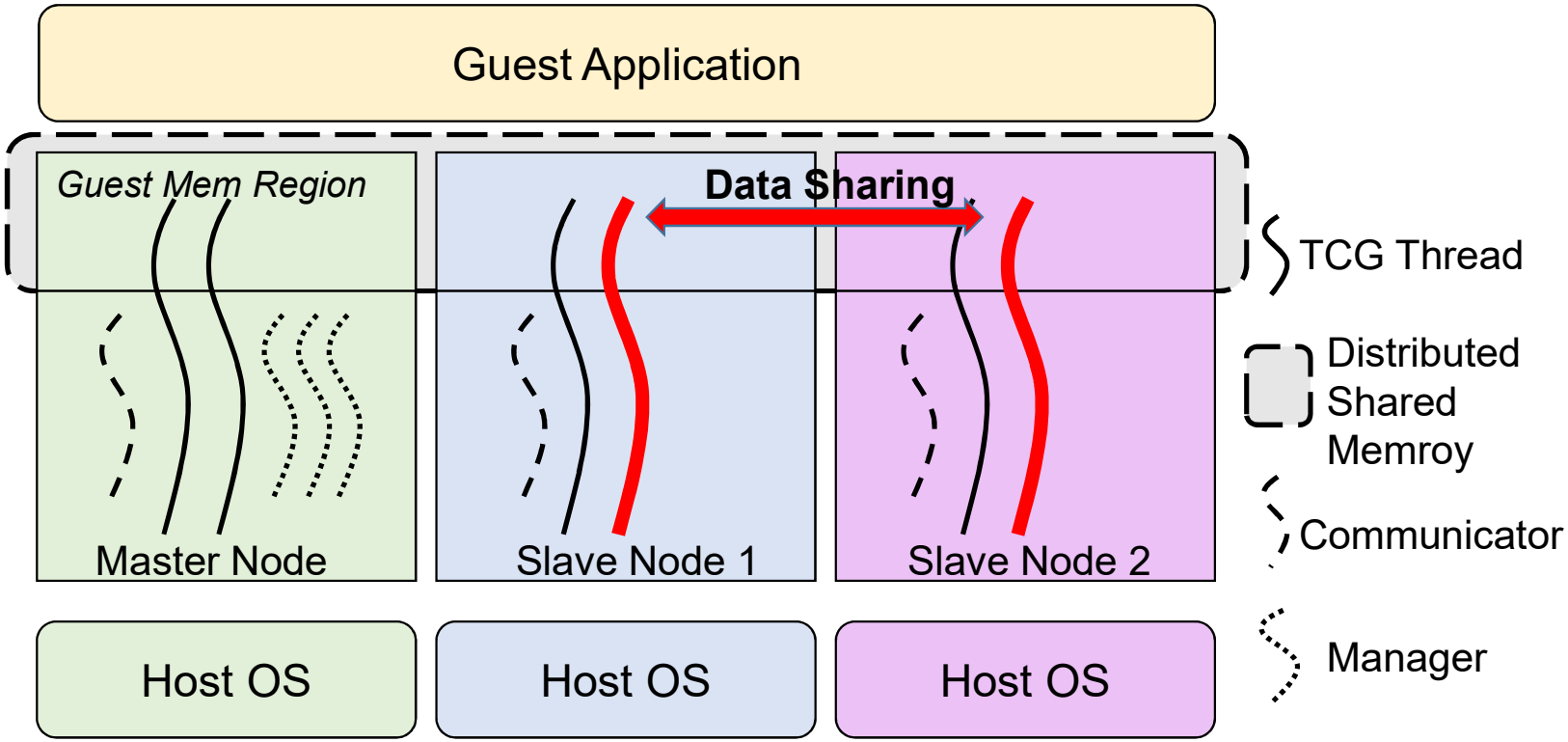
# Page Split: The false sharing overhead




- Reduce false sharing possibility
- Compatible with cache coherence protocol

# Hint-based thread scheduling: data sharing among nodes



# Hint-based thread scheduling: data sharing among nodes

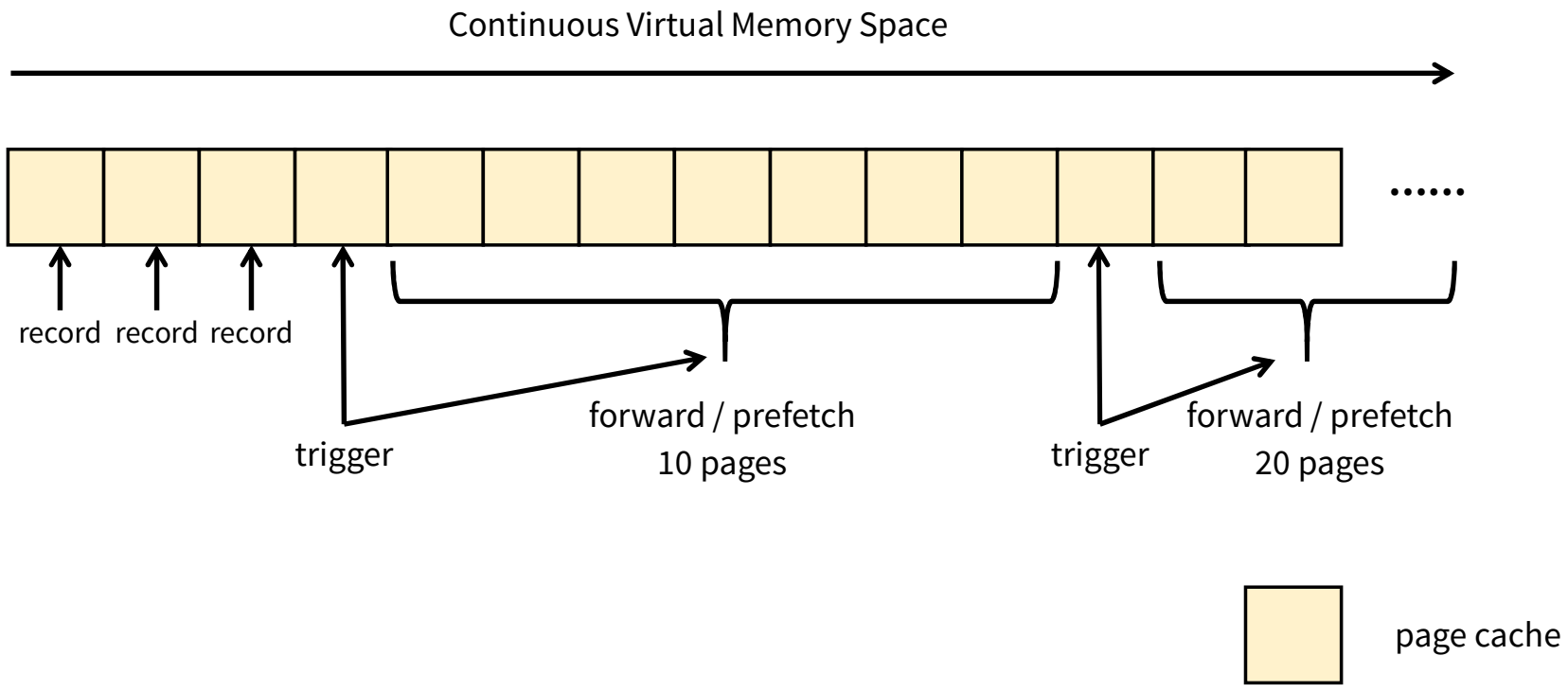
```
for (int i =0; i<N; i++) {
    DQEMU_schedule();
    // do_something
}
```

Source Code Hint

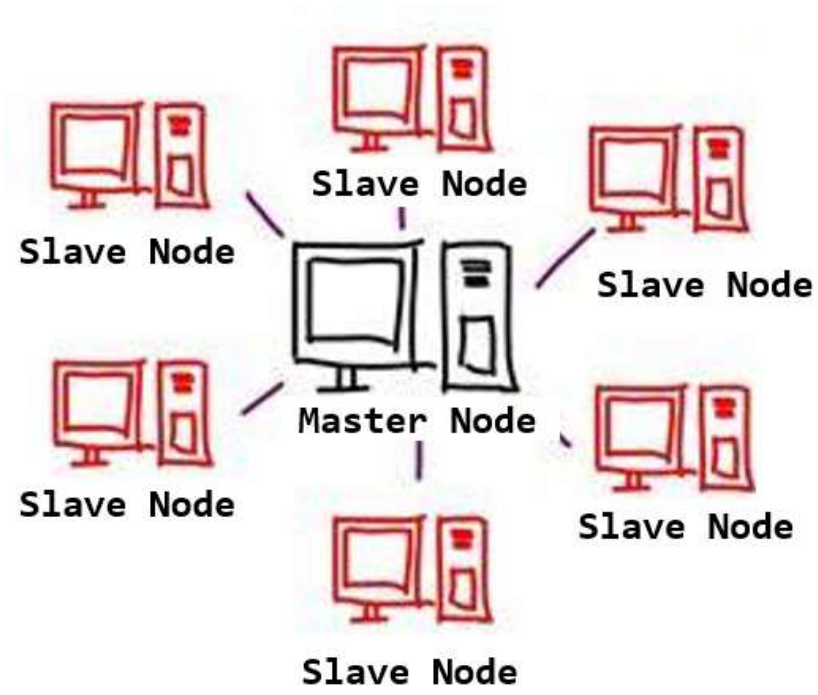
```
movl    $0x0,-0x8(%rbp)
jmp     0x41664
nop
xorl    $0x2333,-0xc(%rbp)
```

Means “call DQEMU_scheduler” to DBT

# Page forwarding: to cover the network latency



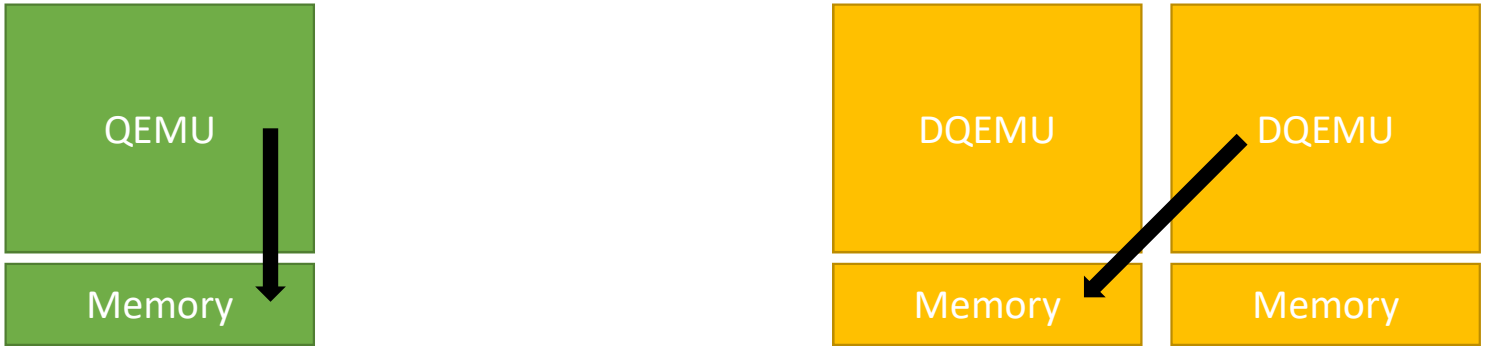
# Experiment Setup




| | |
|---|---|
| **Network** | TP-Link TL-SG1024DT Gigabit Switch |
| **Processor** | Quad-core Intel i5-6500@3.30GHz CPU |
| **Memory** | 12GB |
| **Kernel** | Linux 4.15.0 Ubuntu 18.04 |
| **Workload** | micro bench, PARSEC-3.0 |
| **ISA** | Guest: ARM → Host: X64 |
| **Baseline** | QEMU-4.2.0 |

# Memory Access Performance

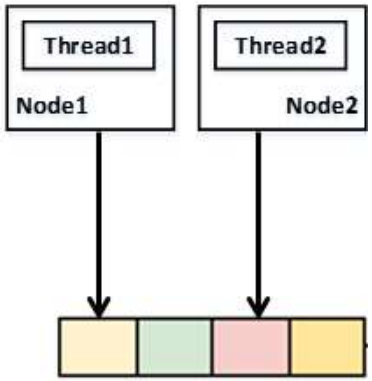
**Sequential memory access**




| Access Type | Throughput(MB/s) | Latency(us) |
|---|---|---|
| QEMU Sequential Access | **173.06** | - |
| Remote Sequential Access | 7.88 | 410.5 |
| Page forwarding Enabled | 108.01 | 83.2 |

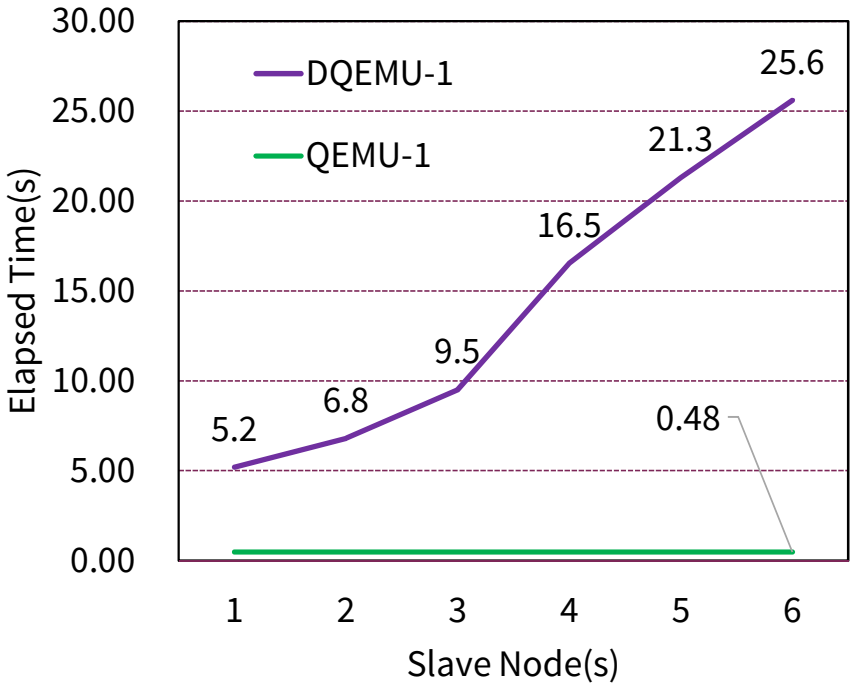
# Memory Access Performance

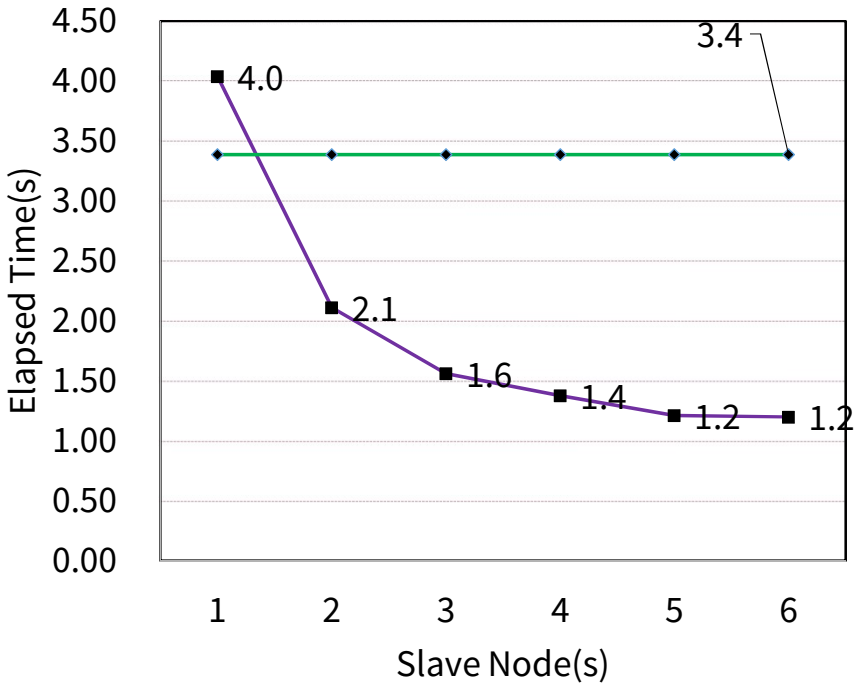
**False sharing**



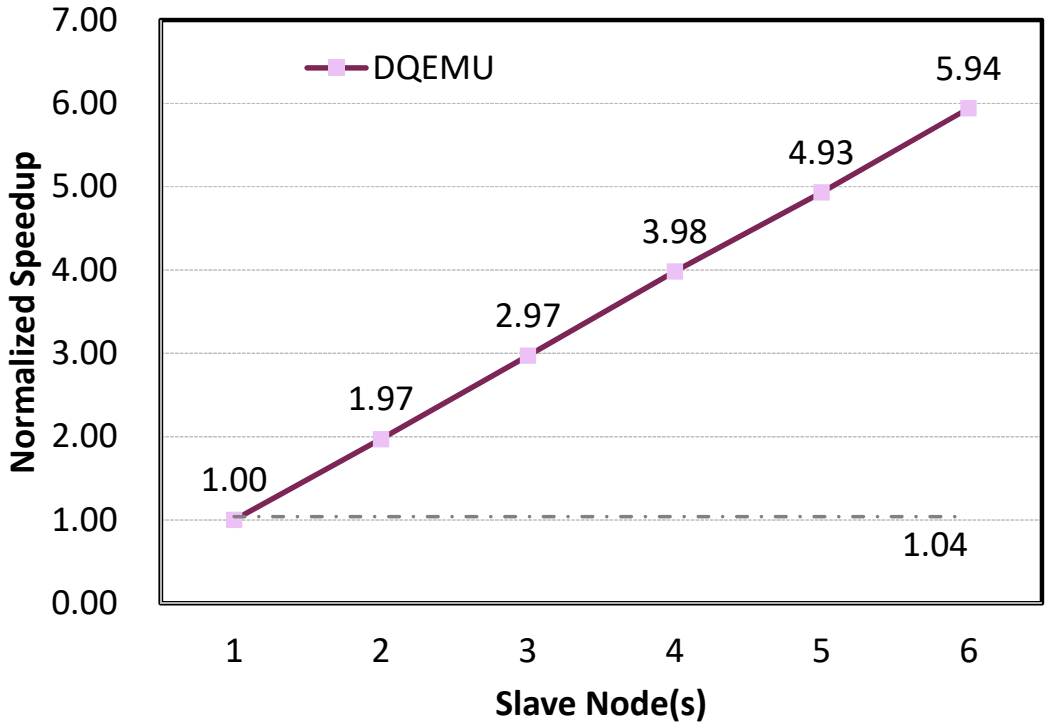

| Access Type | Throughput(MB/s) |
|---|---|
| QEMU Access of 128 bytes | **20,259** |
| False Sharing of 1 Page | 2,216 |
| Page Splitting Enabled | 75,294 |

# Atomic Operation Performance

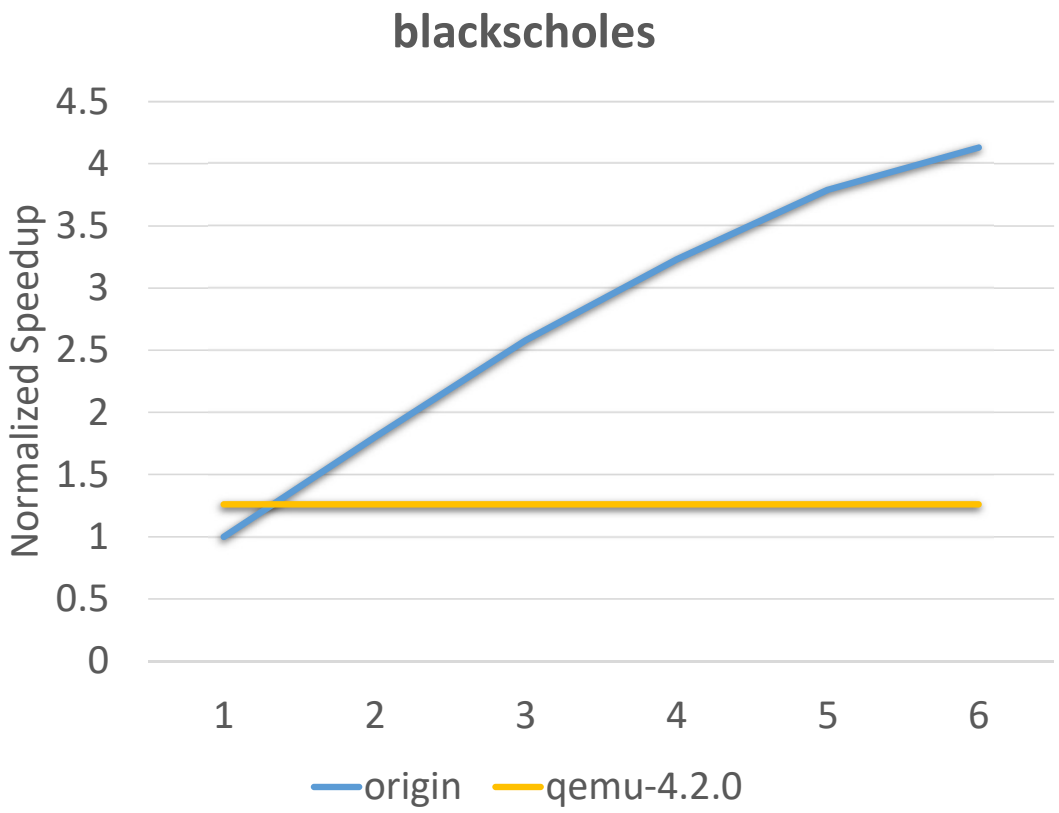


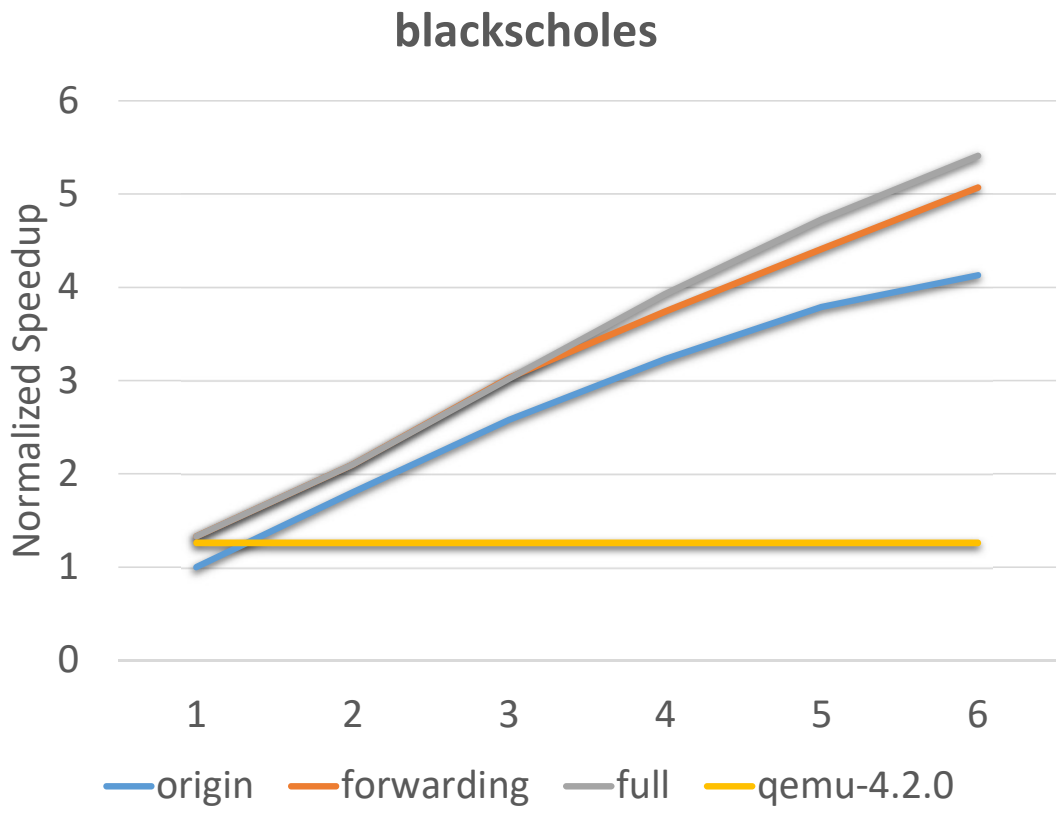
# Scalability - Ideal



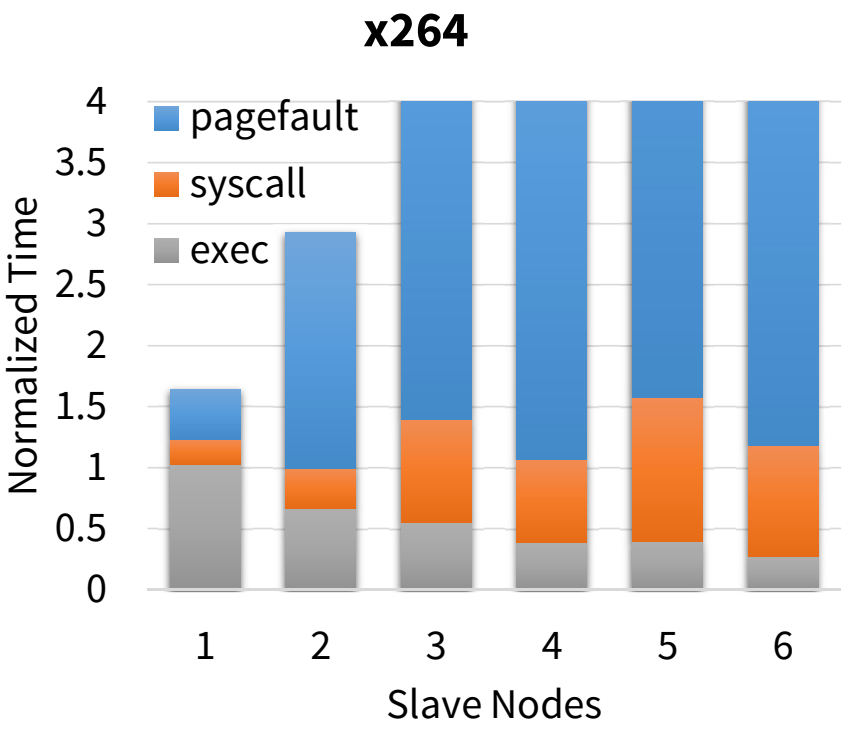
# Scalability – Parallel Programs

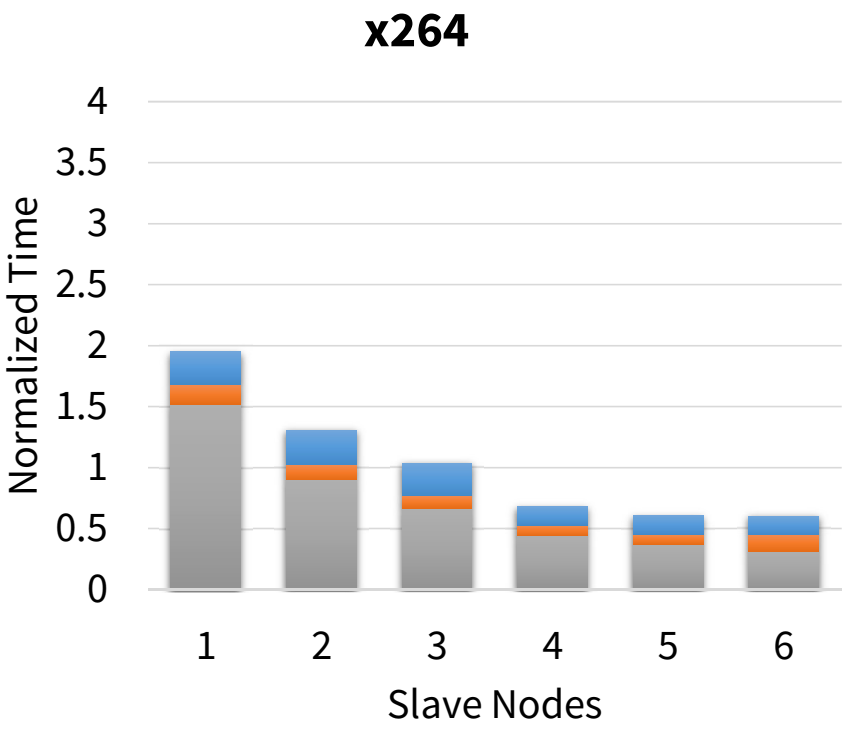


# Scalability – Parallel Programs



# Scalability – Heavy data sharing program



# Discussion

- A more scalable coherence protocol?
- Random memory access hurts DSM.
- What kind of program suits DQEMU? How to recognize?
- Support various host ISA → Heterogeneous computing?

# Thank you!
# Q&A