The CASE of FEMU: Cheap, Accurate, Scalable and Extensible Flash Emulator

Huaicheng Li, Mingzhe Hao, Michael Hao Tong, Swaminatahan Sundararaman*, Matias Bjørling+, Haryadi S. Gunawi
What SSD platforms are used?

**Simulator**
- DiskSim+SSD
- SSDSim
- FlashSim
  - Simple
  - Time-saving
- Trace driven
- Internal-research only

**Emulator**

**Hardware Platform**

**Trends**
- Software-Defined Flash
- Split-Level Architecture
**Simulator**
- DiskSim+SSD
- SSDSim
- FlashSim
- Simple
- Time-saving
- Trace driven
- Internal-research only

**Emulator**
- Simple
- Time-saving
- Trace driven
- Internal-research only

**Hardware Platform**
- OpenSSD
- OpenChannel-SSD
- Full-stack Research
- Accurate
- Expensive
- Complex to use
- Wear-out

- 20%
- 19% Single SSD
- 1% Distributed SSDs
**Simulator**
- DiskSim+SSD
- SSDSim
- FlashSim

- Simple
- Time-saving
- Trace driven
- Internal-research only

**Emulator**
- LightNVM’s QEMU
- VSSIM
- FlashEm

- Fullstack Research
- Cheap
- Full-stack Research
- Accurate

- Poor Scalability
- Poor Accuracy
- Expensive
- Complex to use

**Hardware Platform**
- OpenSSD
- OpenChannel-SSD
- Wear-out
The “CASE” of FEMU

FEMU: QEMU/Software based Flash Emulator

- **Cheap**: $0, https://github.com/ucare-uchicago/femu
- **Accurate**: 0.5-38% error rate in latency
  - 11% average at microsecond level
- **Scalable**: support 32 channels/chips
- **Extensible**
  - modifiable interface
  - modifiable FTL
What is FEMU?

Typical Fullstack Research

- App
- Host OS
- Hardware Platform

FEMU Fullstack Research

- App
- Guest OS
- VM
- NVMe
- FEMU
- QEMU

Supported research:
- Kernel changes
- Interface changes
- FTL changes
QEMU Scalability

Guest OS

QEMU

IO

IO

IO

IO

Expected

# of threads

IO Latency (us)

1  2  4  8  16  32  64

0  50  100  150  200  250  300  350  400

...
QEMU IDE Scalability

1 IO thread

IO

Guest OS

QEMU

1 2 4 8 16 32 64

# of threads

IO Latency (us)

0 50 100 200 300 400

Expected

- Expected IO latency for different numbers of threads.
Expected 2 IO threads
Guest OS
QEMU

Represent VSSIM
QEMU NVMe Scalability

Represent LightNVM’s QEMU

Guest OS

QEMU
QEMU Scalability

QEMU and existing emulators are NOT Scalable!

FEMU is Scalable!
Scalability Root Causes & Solutions (1)

- QEMU NVMe Emulation
- Guest OS
- App
- NVMe driver
  - Submission Queue
  - Completion Queue
  - Tail DoorBell
  - Head DoorBell
- thousands of cycles
- interrupt overhead

- VM-exit

- Submission Queue
- Completion Queue
- Shadow DoorBell
- polling
- ZERO VM-exit
- QEMU NVMe Emulation

13
Scalability Root Causes & Solutions (2)

- NVMe Emulation
- Block Driver
- DMA Emulation
- Image Format Driver
- Raw Device Driver
- AIO Queue
- Thread Pool
- Host File System
- Host Block IO Layer
- Host Device Driver

- DMA Emulation
- FEMU Heap Storage
- DMA from/to heap storage

More than 20us latency reduction
FEMU Accuracy

\[ \text{Error} = \frac{|L_{\text{femu}} - L_{\text{oc}}|}{L_{\text{oc}}} \]
Single-Register model (S-Reg)

Double-Register model (D-Reg)

\[ \text{OLTP} \]

\[ \text{Error (')} \]

\[ \text{D-Reg S-Reg} \]
FEMU Accuracy

Latency Error: 11-57% ⇒ 0.5-38%

Single Register Model (S-Reg) Double Register Model (D-Reg)

X: # of channels
Y: # of planes per channel
FEMU Limitations

- Further optimizations to support higher parallelism (more scalable)
- Accuracy can be improved
- Not able to emulate large-capacity SSD
- No persistence
Installing, using and debugging FEMU can cause side effects including headache, nausea, agitation, and depression. If your research condition does not improve after using FEMU for a week, please talk to your advisor or us right away.

- Cheap
- Accurate
- Scalable
- Extensible
Thank you!

Questions?

FEMU: https://github.com/ucare-uchicago/femu

http://ucare.cs.uchicago.edu