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

THE UNIVERSITY OF CHICAGO

Parallela

CNEXLABS
What SSD platforms are used?

**Simulator**
- DiskSim+SSD
- SSDSim
- FlashSim

**Emulator**

**Hardware Platform**

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

- 57%

**Trends**
- Software-Defined Flash
- Split-Level Architecture
**Simulator**

- DiskSim+SSD
- SSDSim
- FlashSim

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

**Emulator**

- 20%
  - 19% Single SSD
  - 1% Distributed SSDs

**Hardware Platform**

- OpenSSD
- OpenChannel-SSD

- Full-stack Research
- Accurate

- Expensive
- Complex to use
- Wear-out
**Simulator**
- DiskSim+SSD
- SSDSim
- FlashSim

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

**Hardware Platform**
- OpenSSD
- OpenChannel-SSD

---

**Guest OS**
- Fullstack Research
- Accurate

**Simulator**
- Simple
- Time-saving

**Emulator**
- Fullstack Research
- Cheap
- Trace driven
- Internal-research only

**Hardware Platform**
- Full-stack Research
- Accurate
- Expensive
- Complex to use
- 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

FEMU Fullstack Research

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

Guest OS

QEMU

IO

IO

IO

IO

IO

Expected

# of threads
QEMU IDE Scalability

1 IO thread

Guest OS

QEMU

IO Latency (us)

# of threads

Expected

100 100 200 300 400

0
2 IO threads

Guest OS

QEMU

Expected

IO Latency (us)

# of threads
Guest OS

QEMU

Represent VSSIM

IO Latency (us)

# of threads

Expected
QEMU NVMe Scalability

Represent LightNVM’s QEMU

IO

Guest OS

QEMU

IO

IO

IDE

NVMe

Expected

# of threads

IO Latency (us)
QEMU Scalability

QEMU and existing emulators are NOT Scalable!

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

- QEMU NVMe Emulation
- Tail DoorBell
- Head DoorBell
- Submission Queue
- Completion Queue

- App
- Guest OS
- NVMe driver

- Thousands of cycles interrupt **overhead**

- ZERO VM-exit

- Shadow DoorBell

- POLLING

- Submission Queue
- Completion Queue

- VM-exit

- QEMU NVMe Emulation
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

- FEMU Heap Storage
- DMA from/to heap storage

More than 20us latency reduction
FEMU Accuracy

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

Double-Register model (D-Reg)

OLTP

Error (%)
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
Conclusion

Installing, and using 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 us, your advisor, or your doctor immediately.

- Cheap
- Accurate
- Scalable
- Extensible

https://github.com/ucare-uchicago/femu
Thank you!

Questions?

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

Huaicheng Li, huaicheng@cs.uchicago.edu

http://ucare.cs.uchicago.edu