Designing a Storage Software Stack for Accelerators

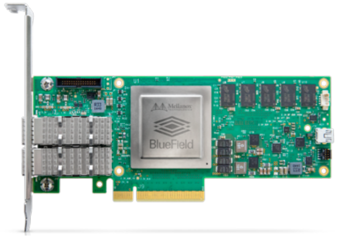
Shinichi Awamoto[1], Erich Focht[2], Michio Honda[3]

<shinichi.awamoto@gmail.com>

[1]NEC Labs Europe; [2]NEC Deutschland; [3]Univerisity of Edinburgh

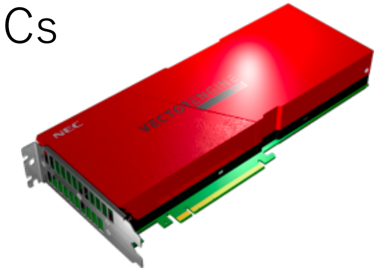
SoC-based Accelerators

execute the entire application code unlike GPUs.

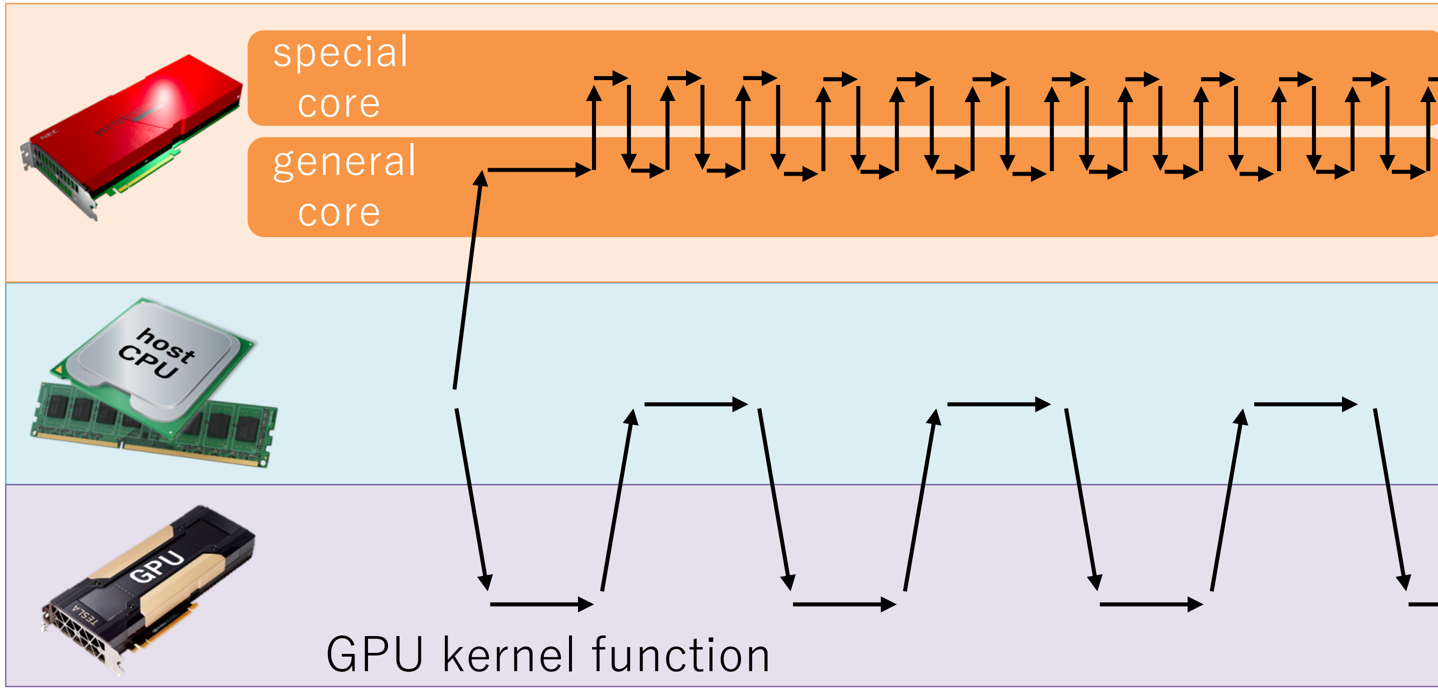


SmartNICs

Xilinx Zynq



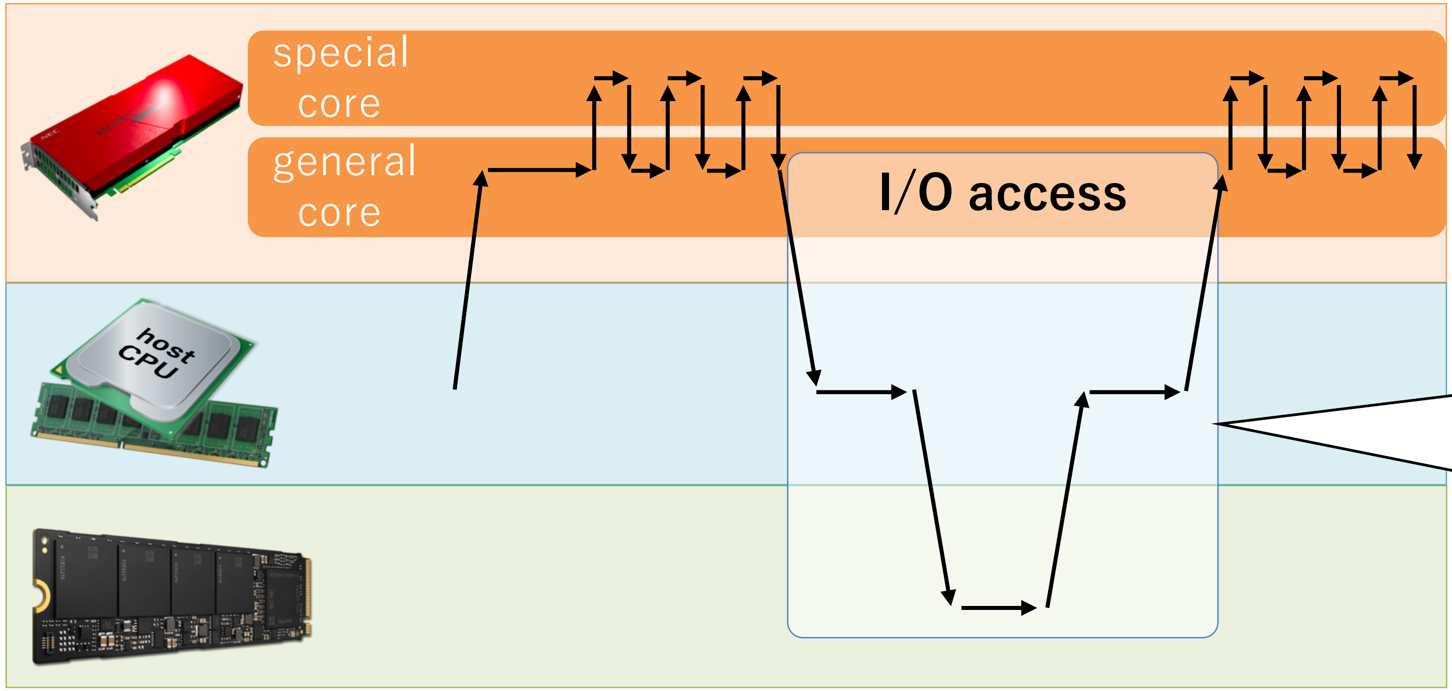
NEC SX-Aurora TSUBASA
Vector Engine

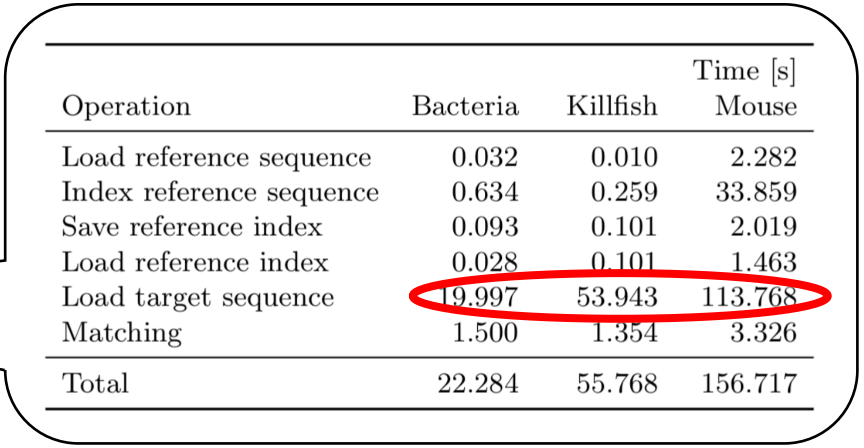


photos from: xilinx.com, www.mellanox.com, uk.nec.com, nvidia.com

The I/O problem in the accelerator

But still, the host system mediates data access, which incurs overhead.

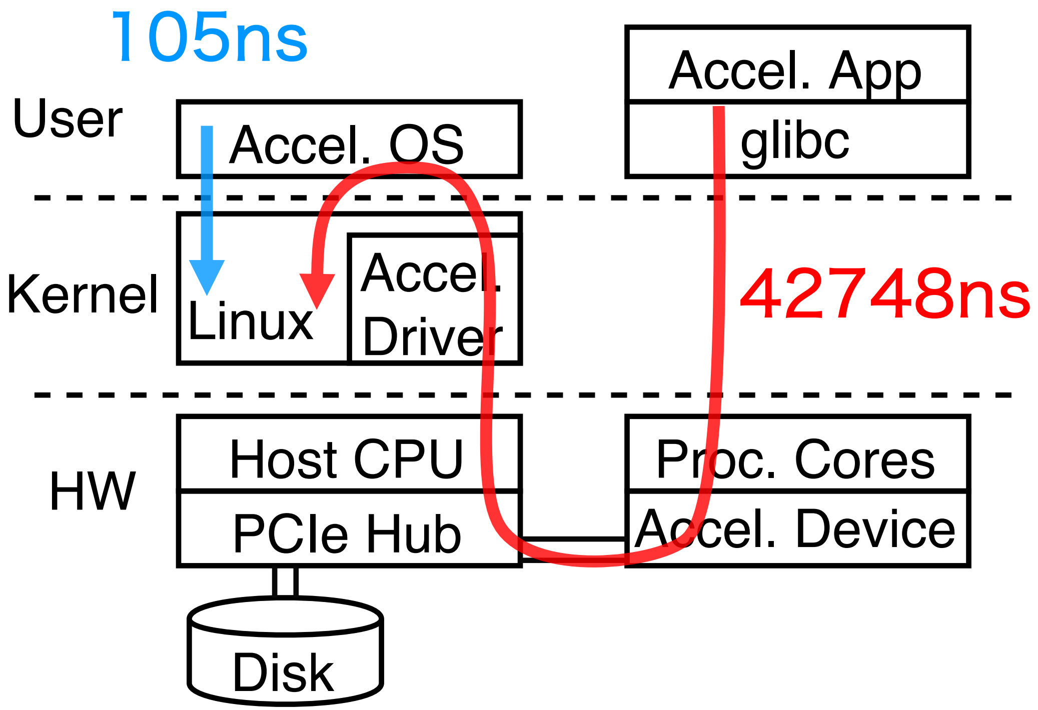




Operation	Bacteria	Killfish	Time [s] Mouse
Load reference sequence	0.032	0.010	2.282
Index reference sequence	0.634	0.259	33.859
Save reference index	0.093	0.101	2.019
Load reference index	0.028	0.101	1.463
Load target sequence	19.997	53.943	113.768
Matching	1.500	1.354	3.326
Total	22.284	55.768	156.717

How does this problem happen?

Multiple data copies and dispatch inside of redirected system calls increase latency.



105ns

User

Accel. OS

Accel. App

glibc

Kernel

Linux

Accel. Driver

42748ns

HW

Host CPU

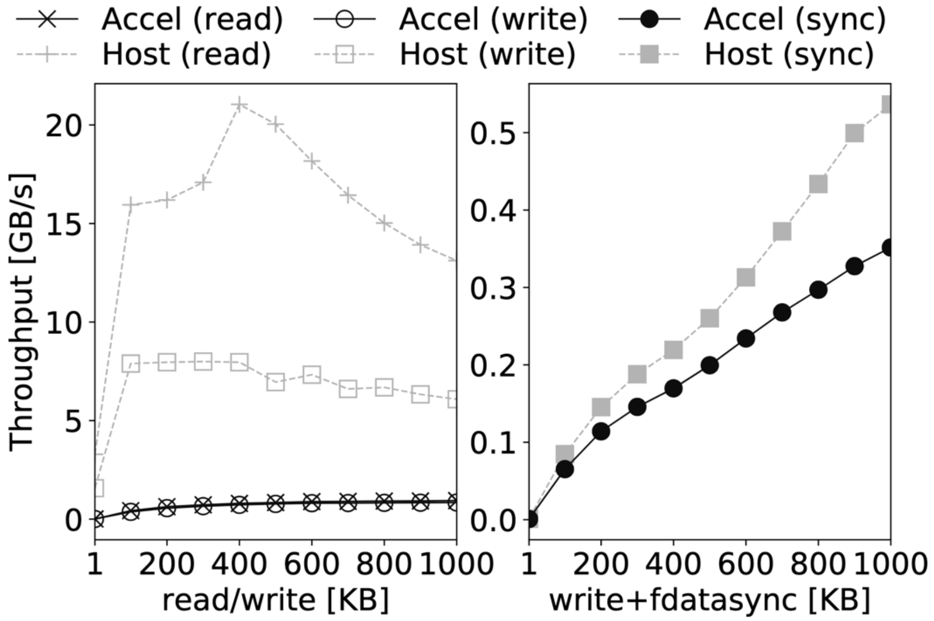
PCIe Hub

Proc. Cores

Accel. Device

Disk

Even on microbenchmarks, the overhead is obvious.

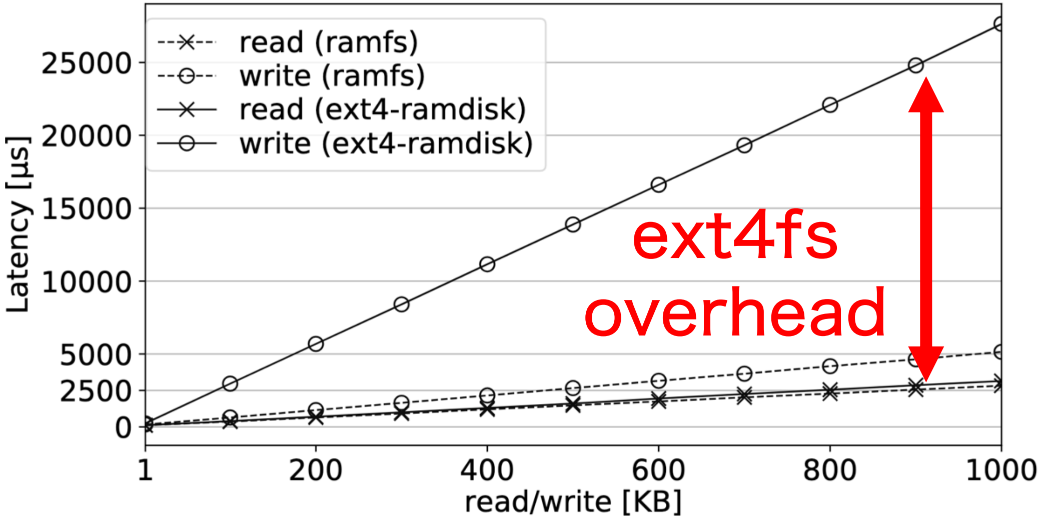


Designing a Storage Stack

Goal: Fast storage access in the accelerator applications.

Design options

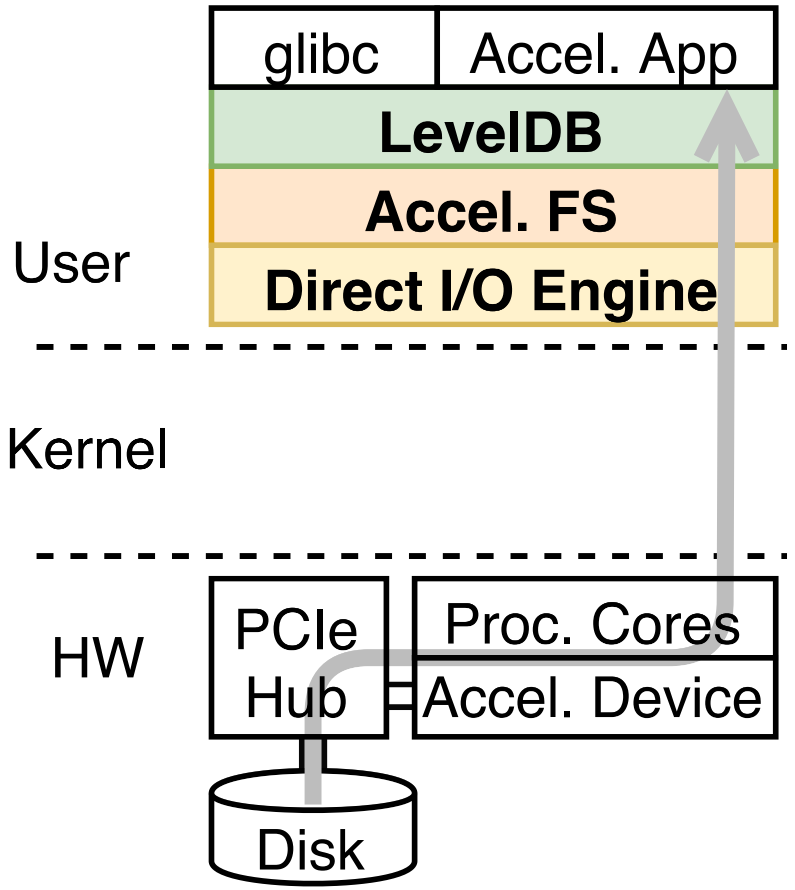
- **Linux kernel library** (RoEduNet ‘10)
  Expensive kernel emulation
- Buffer cache sharing
  e.g. GPUfs (ASPLOS ‘13), SPIN (ATC ‘17)
  Only DMAs are mitigated.
- Heterogeneous-arch kernels
  e.g. Multikernel (SOSP ‘09), Popcorn Linux (ASPLOS ‘17)
  No kernel context in the accelerator



Conventional system software does not perform well on wimpy accelerator cores.

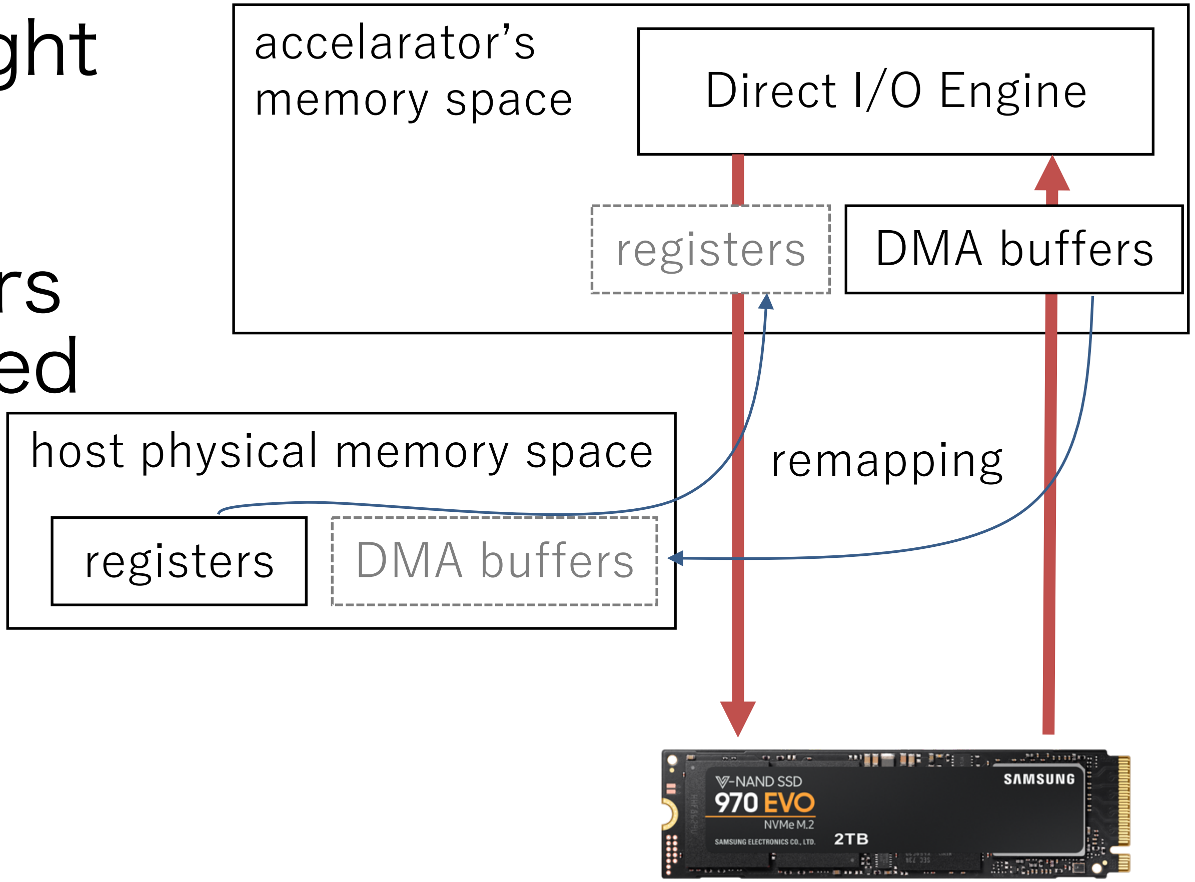
Accelerator storage stack

- NVMe device driver within the accelerator user-space
- File system for data organization and buffer caches
- LevelDB for KVS interface



How Direct I/O Engine controls SSDs?

- UIO manages a device access right on the host side.
- Device registers and DMA buffers are remapped using APIs provided by the accelerator vendor.
- No host side intervention is needed throughout the entire process.



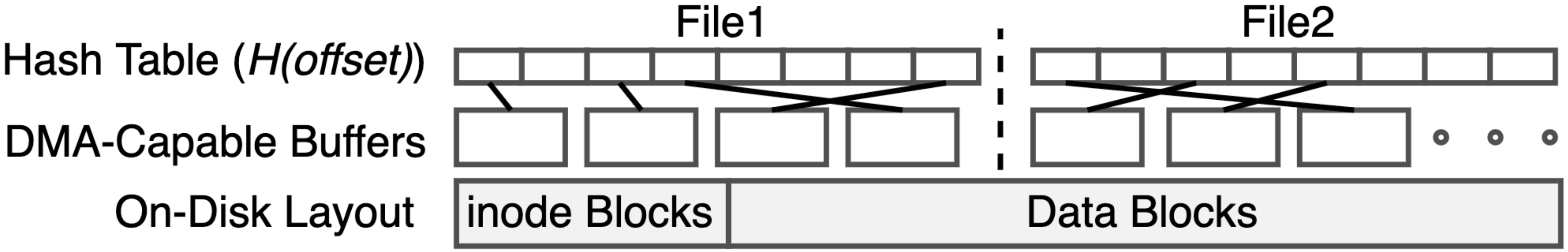
AccelFS & LevelDB

- AccelFS provides file names, organized data, buffer caches without sacrifice of performance.

  The current ext2-like design would be replaced with accelerator-aware implementation.

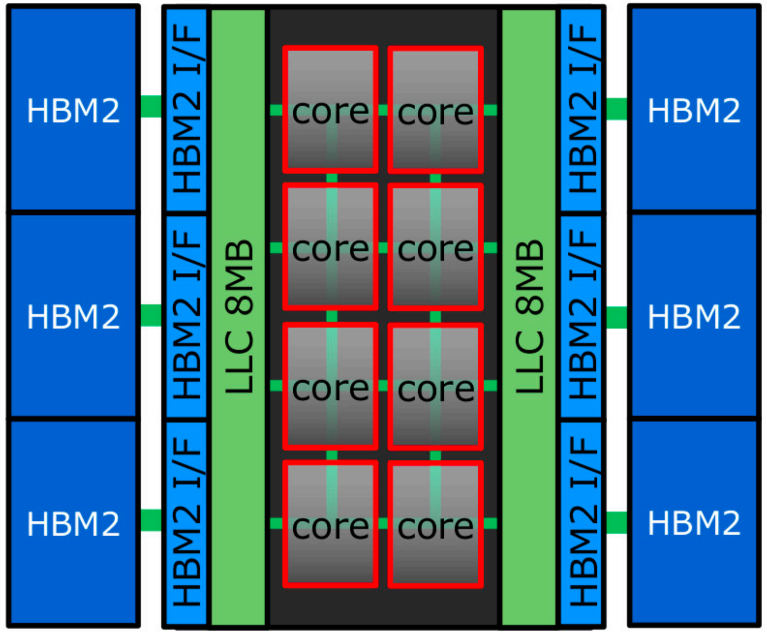
- LevelDB is also ported on top of AccelFS.

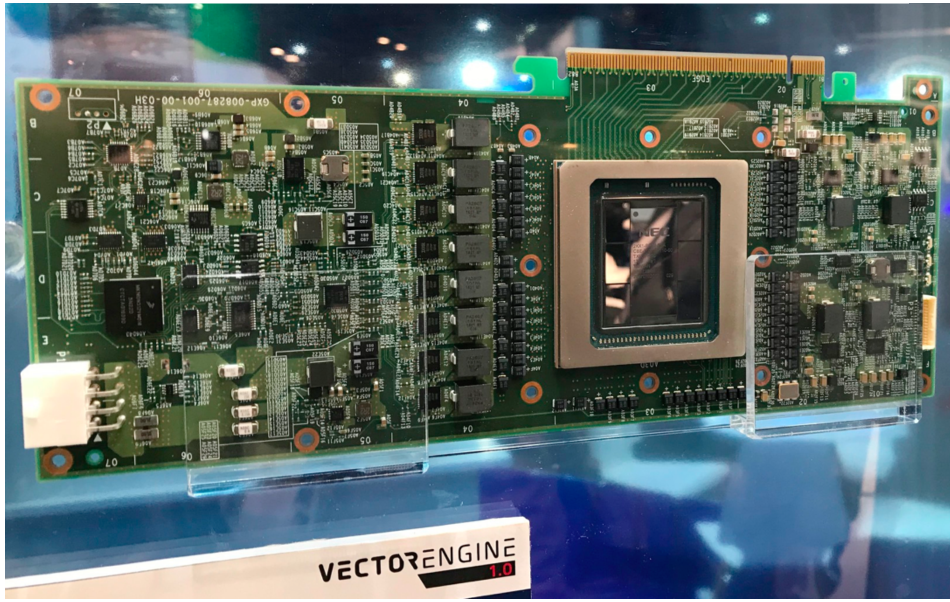
  As a further step, we are exploring accelerator specific optimizations. (e.g. vectorized LSM-tree compactions)



Evaluation

- Host:
  Intel Xeon Gold 6126 (2.60GHz, 12-core)
  96GB RAM
- Accelerator: NEC SX-Aurora TSUBASA
- NVMe SSD: Samsung EVO 970



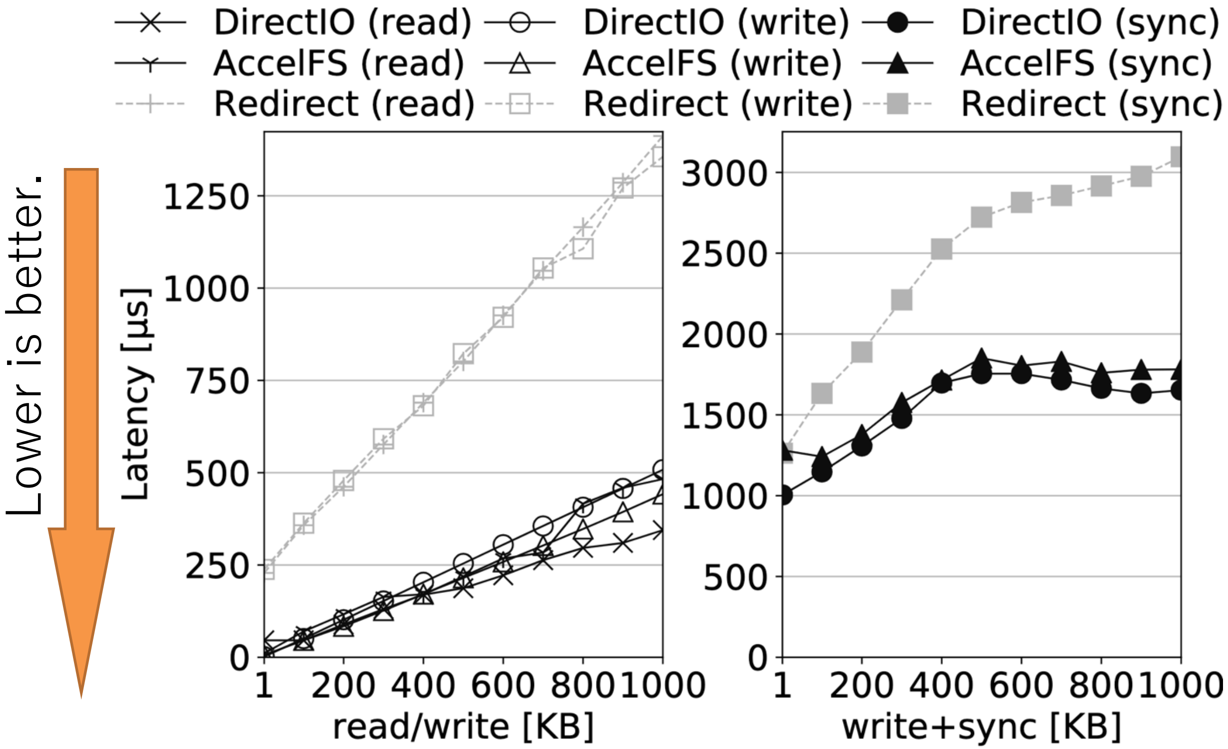


photos from: sx-aurora.github.io

Microbenchmarks

How much does HAYAGUI improve file operations?

- read, write and write+sync
  - In baselines, read(2), write(2), fdatasync(2) are used.
- 20-99% reduction in latency

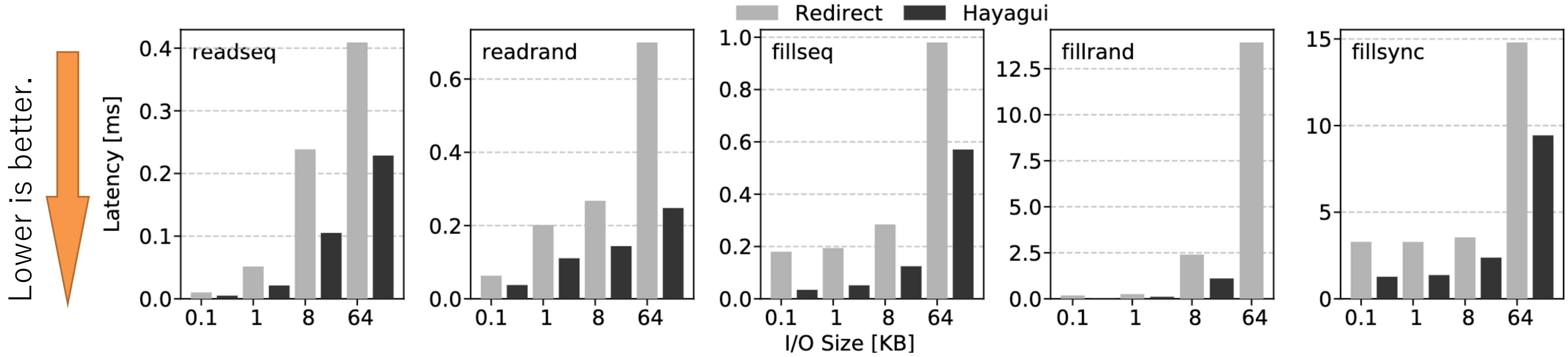


Lower is better.

LevelDB evaluation (db_bench)

How much does HAYAGUI improve KVS workloads?

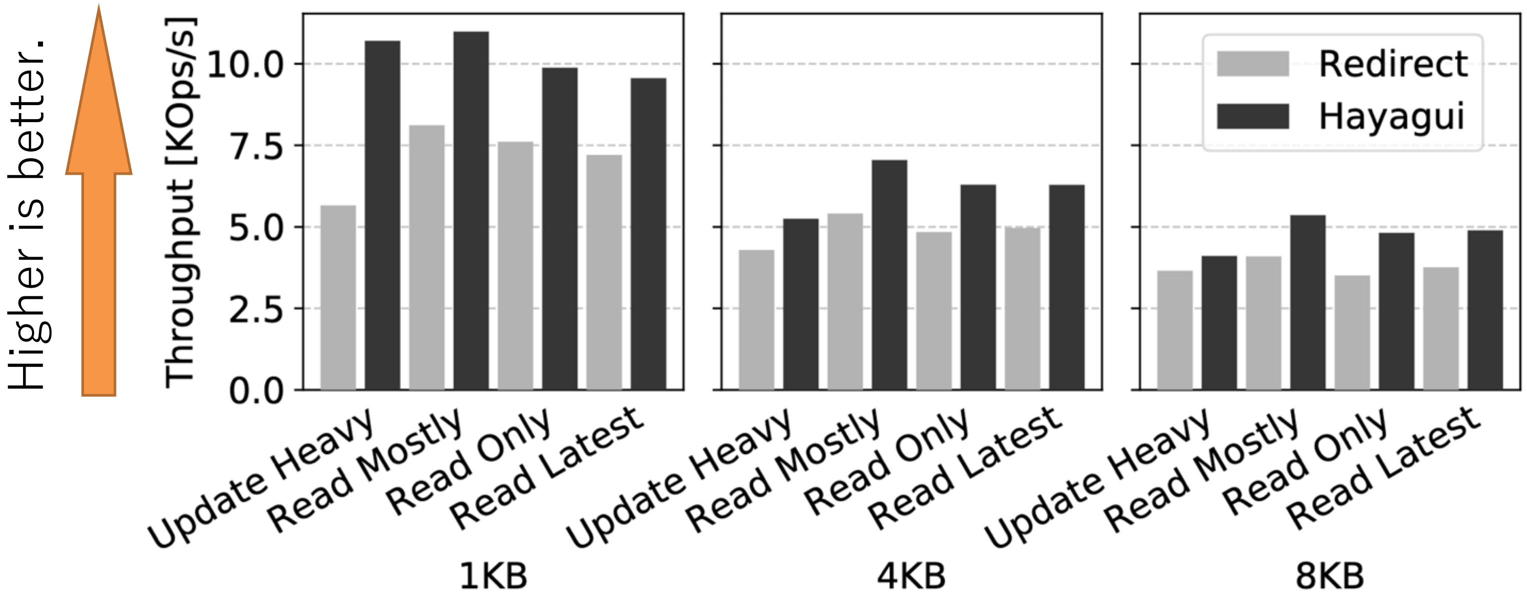
- sequential and random access workloads via db_bench
- 33-81% latency improvements



LevelDB evaluation (YCSB)

Realistic KVS workloads

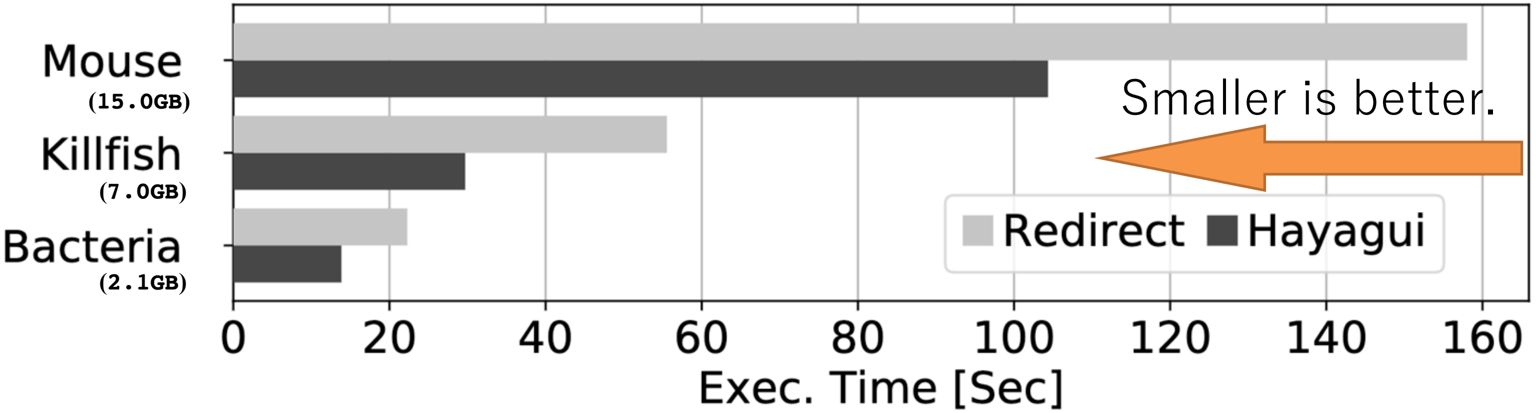
- Small- or medium-sized data
- 12-89% throughput improvements



Genome sequence matching app

How does Hayagui improve realistic apps?

- an accelerator application analyzing DNA sequences
- Bulk-read workloads
- 33-46% reduction in execution time



Summary

- On SoC-based accelerators, I/O access matters.
- HAYAGUI: an accelerator storage stack
  - reads and writes the storage medium directly.
  - provides various interfaces: raw I/O, file system and KVS
  - outperforms the system call redirection baselines

- Ongoing work
  - Is the direct access architecture feasible in other accelerators?
  - How do we overcome the weakness of general-purpose cores in accelerators?
  - How could we exploit specialized engines in accelerators?
  - Is it possible to build a generic, one-size-fit-all storage stack for accelerators?

Designing a Storage Software Stack for Accelerators

Thank You Q&A

Shinichi Awamoto[1], Erich Focht[2], Michio Honda[3]
<shinichi.awamoto@gmail.com>

[1]NEC Labs Europe; [2]NEC Deutschland; [3]Univerisity of Edinburgh