Venice

Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses

Onur Mutlu

omutlu@gmail.com

https://people.inf.ethz.ch/omutlu

7 August 2024

FMS: the Future of Memory and Storage

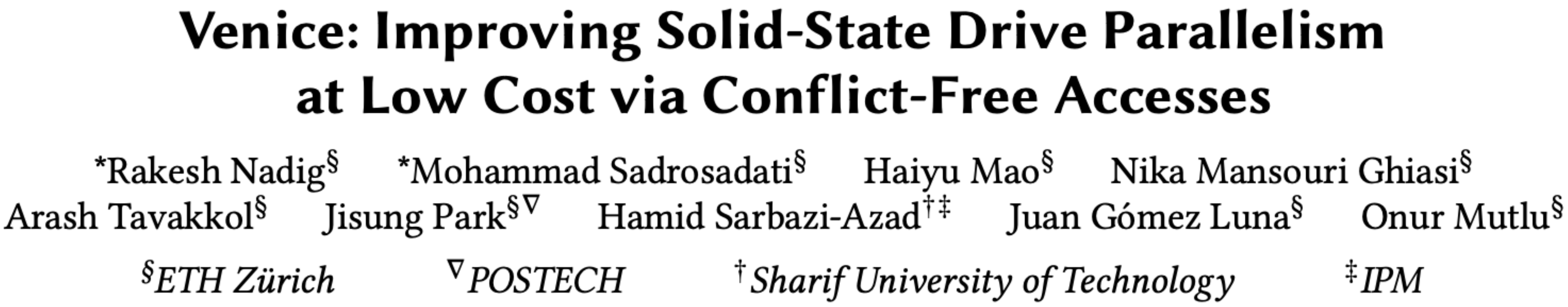
the Future of Memory and Storage

SAFARI

ETH zürich

Venice Paper, Slides, Video [ISCA 2023]

- Rakesh Nadig, Mohammad Sadrosadati, Haiyu Mao, Nika Mansouri Ghiasi, Arash Tavakkol, Jisung Park, Hamid Sarbazi-Azad, Juan Gómez Luna, and Onur Mutlu,
"Venice: Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses"
Proceedings of the 50th International Symposium on Computer Architecture (***ISCA***),
Orlando, FL, USA, June 2023.
[arXiv version]
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (3 minutes)]
[Talk Video (14 minutes, including Q&A)]



Venice: Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses

*Rakesh Nadig[§] *Mohammad Sadrosadati[§] Haiyu Mao[§] Nika Mansouri Ghiasi[§]
Arash Tavakkol[§] Jisung Park[§∇] Hamid Sarbazi-Azad[†‡] Juan Gómez Luna[§] Onur Mutlu[§]

[§]*ETH Zürich* [∇]*POSTECH* [†]*Sharif University of Technology* [‡]*IPM*

https://arxiv.org/pdf/2305.07768

Venice

Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses

Rakesh Nadig*, Mohammad Sadrosadati*, Haiyu Mao,
Nika Mansouri Ghiasi, Arash Tavakkol, Jisung Park,
Hamid Sarbazi-Azad, Juan Gómez Luna, and Onur Mutlu

Presented at ISCA 2023

Talk Outline

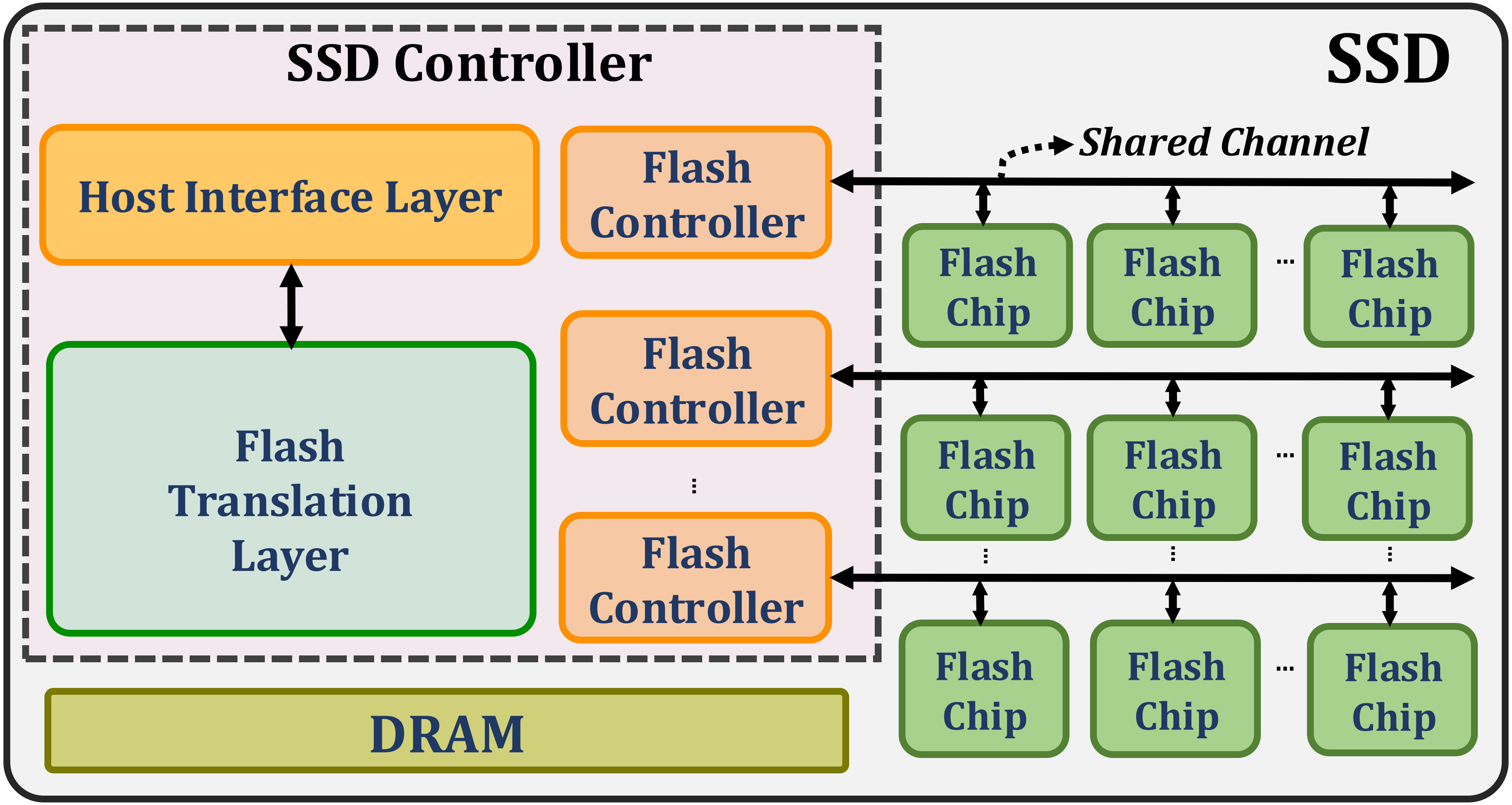
Motivation

Venice

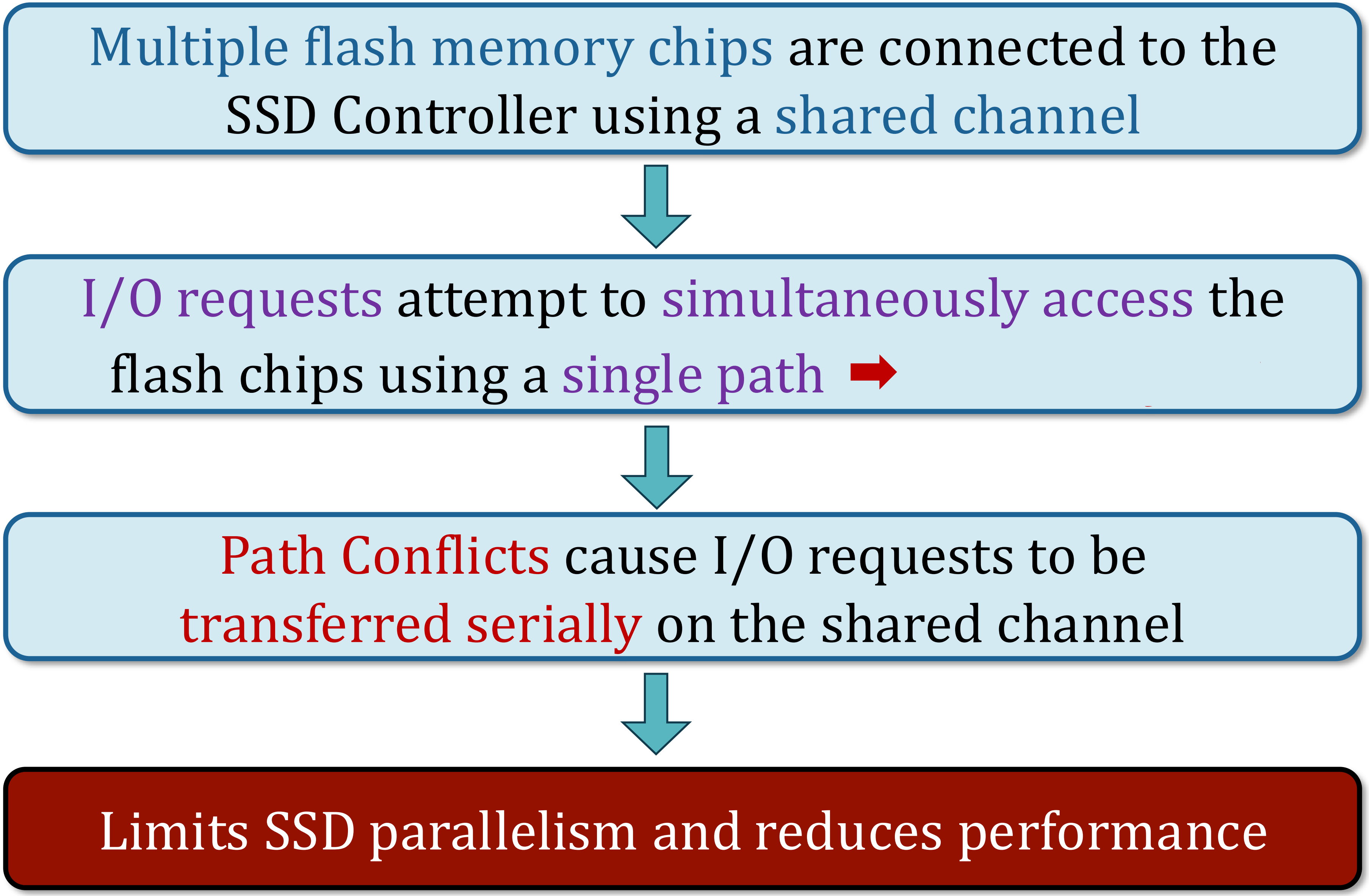
Evaluation

Summary

Overview of a Modern Solid-State Drive



Key Problem: Path Conflicts in Modern SSDs



Multiple flash memory chips are connected to the SSD Controller using a shared channel

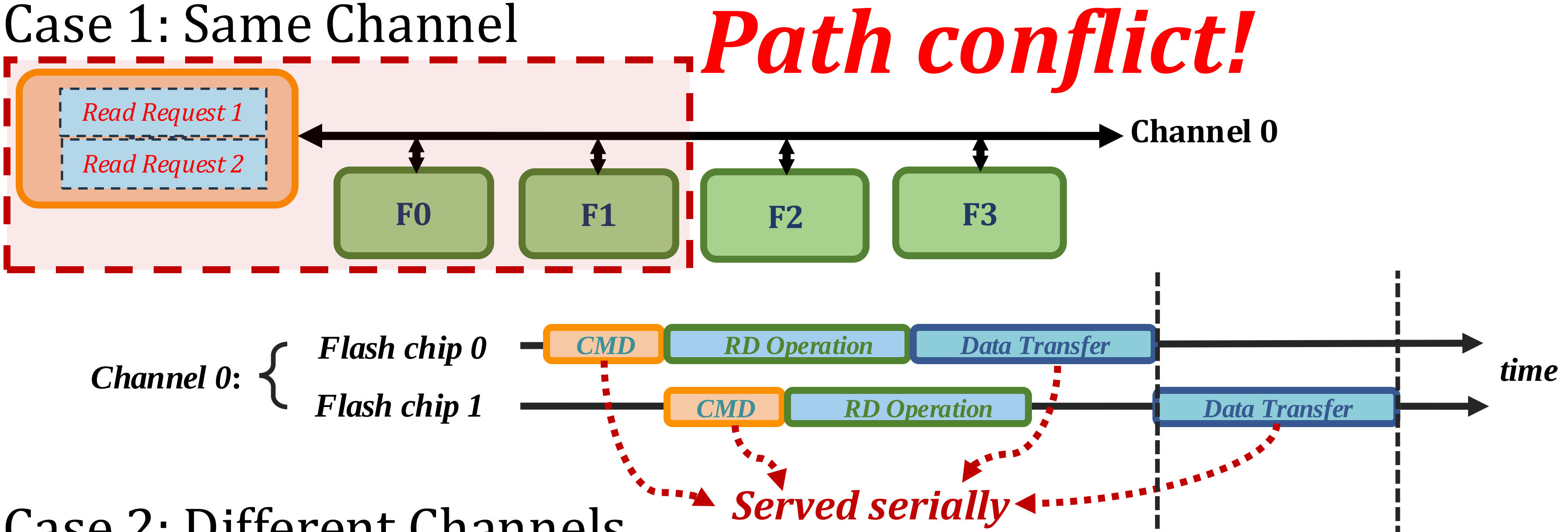
I/O requests attempt to simultaneously access the flash chips using a single path ➡

Path Conflicts cause I/O requests to be transferred serially on the shared channel

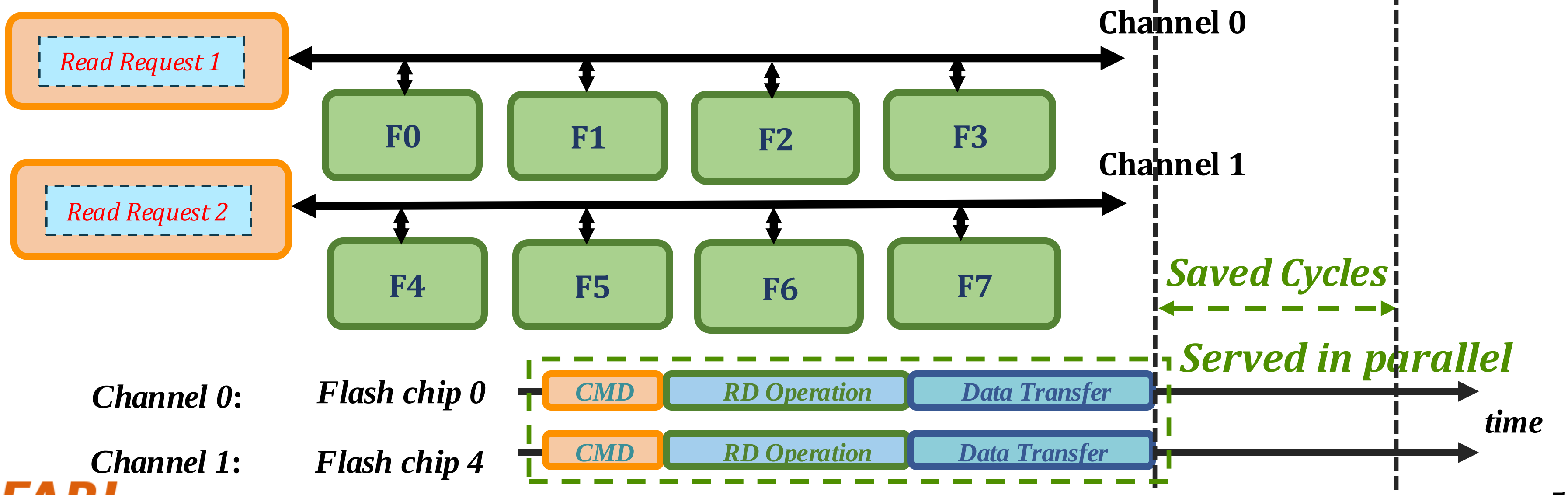
Limits SSD parallelism and reduces performance

Delay Caused by Path Conflicts

- Case 1: Same Channel



- Case 2: Different Channels

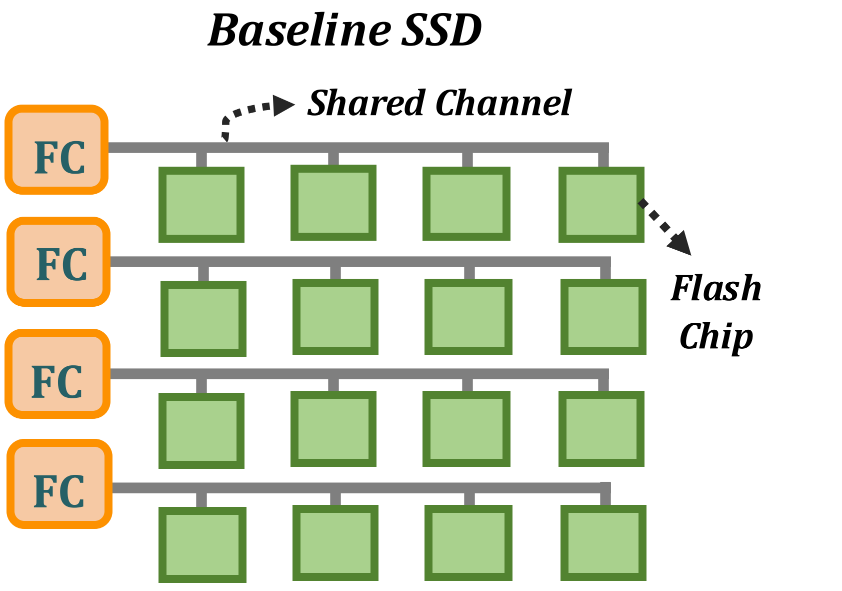


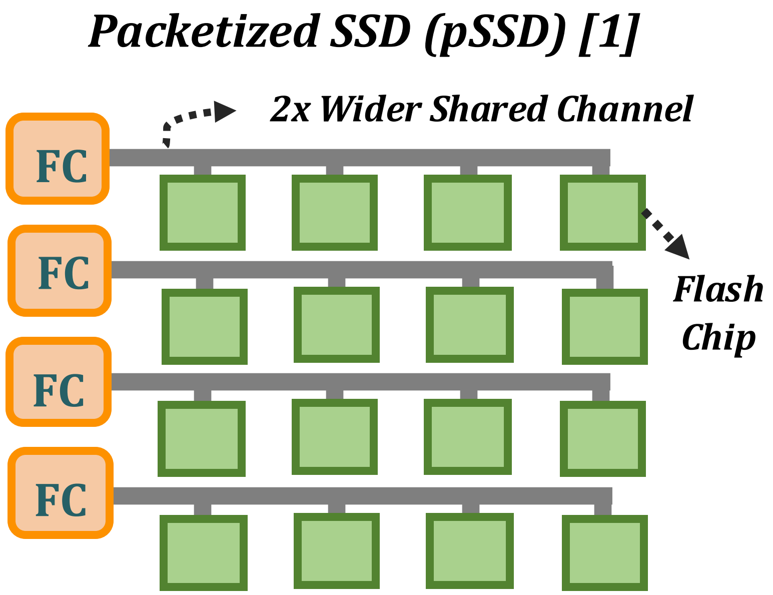
Performance Impact of Path Conflicts

Path conflicts increase the average I/O latency by 57% in our experiments on a performance-optimized SSD

The performance overhead of path conflicts increases by 1.6x in our experiments for high-I/O-intensity workloads

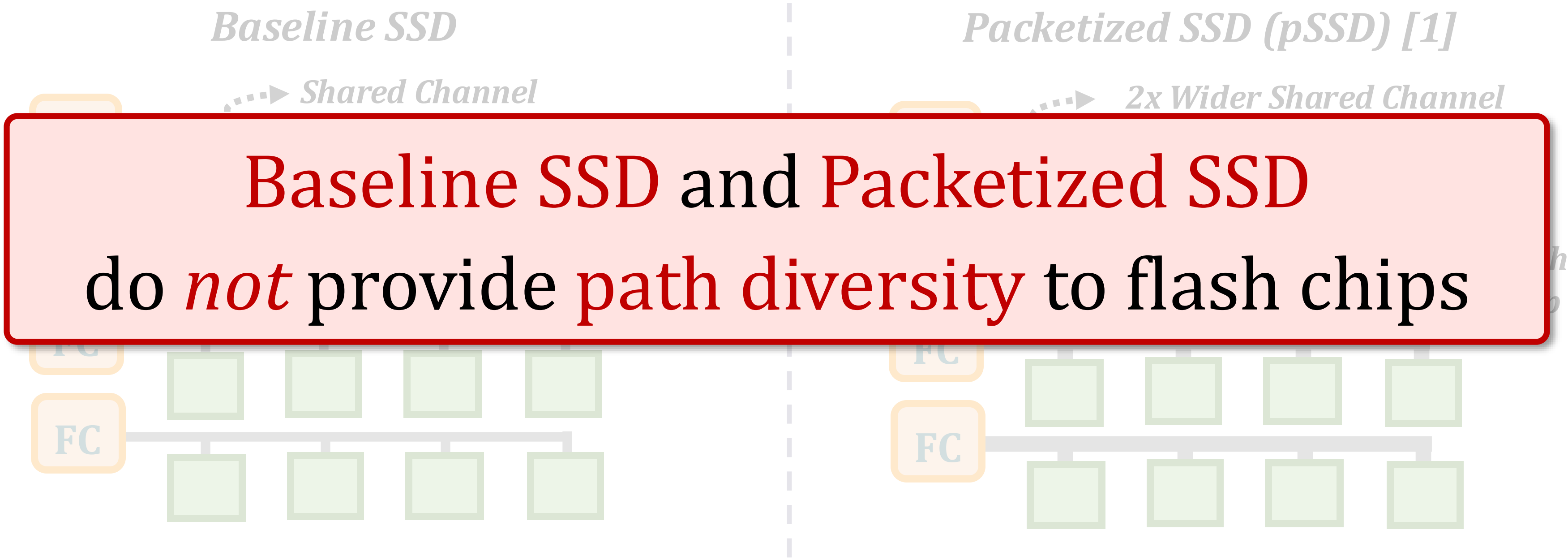
Prior Approaches





[1] Kim+, "Networked SSD: Flash Memory Interconnection Network for High-Bandwidth SSD", MICRO 2022
[2] Tavakkol+, "Network-on-SSD: A Scalable and High-Performance Communication Design Paradigm for SSDs", IEEE CAL 2012

Prior Approaches



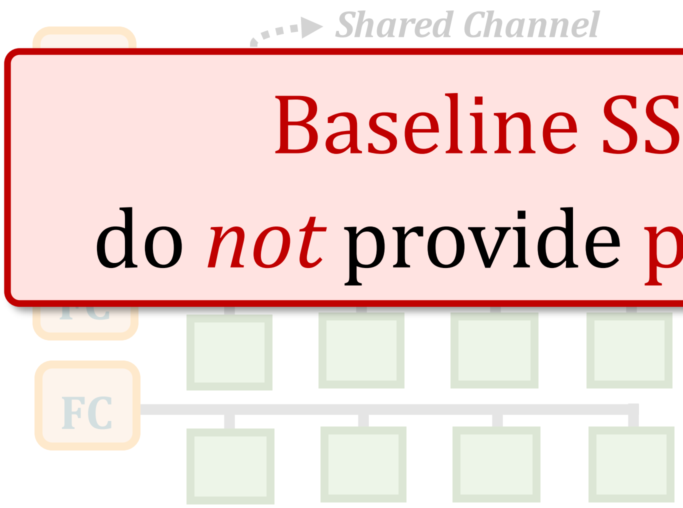
Baseline SSD and Packetized SSD

do *not* provide path diversity to flash chips

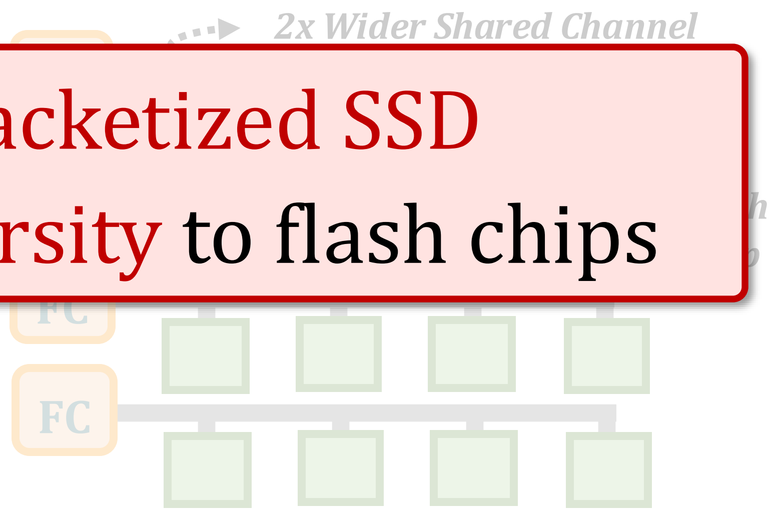
[1] Kim+, "Networked SSD: Flash Memory Interconnection Network for High-Bandwidth SSD", MICRO 2022
[2] Tavakkol+, "Network-on-SSD: A Scalable and High-Performance Communication Design Paradigm for SSDs", IEEE CAL 2012

Prior Approaches

Baseline SSD



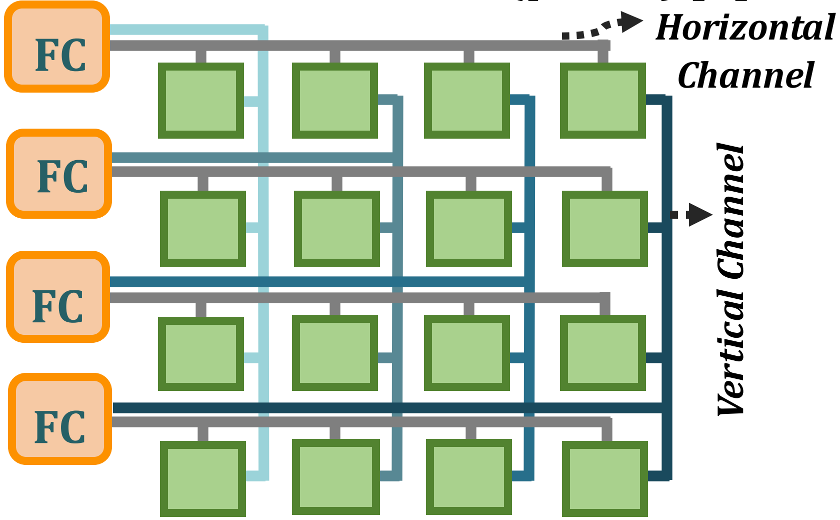
Packetized SSD (pSSD) [1]



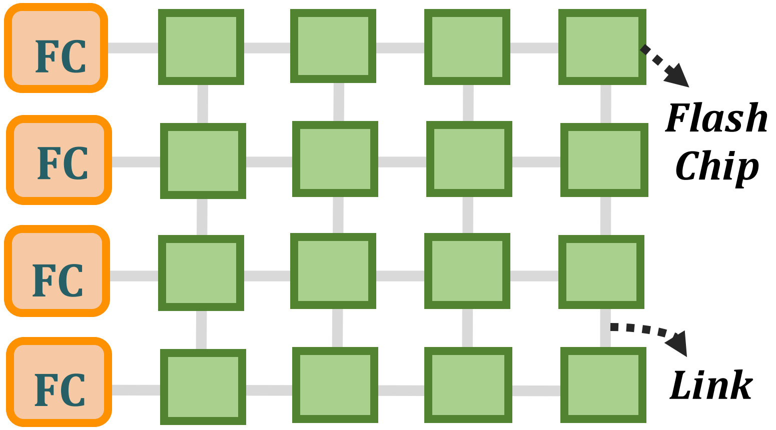
Baseline SSD and Packetized SSD

do *not* provide path diversity to flash chips

Packetized Network SSD (pnSSD) [1]

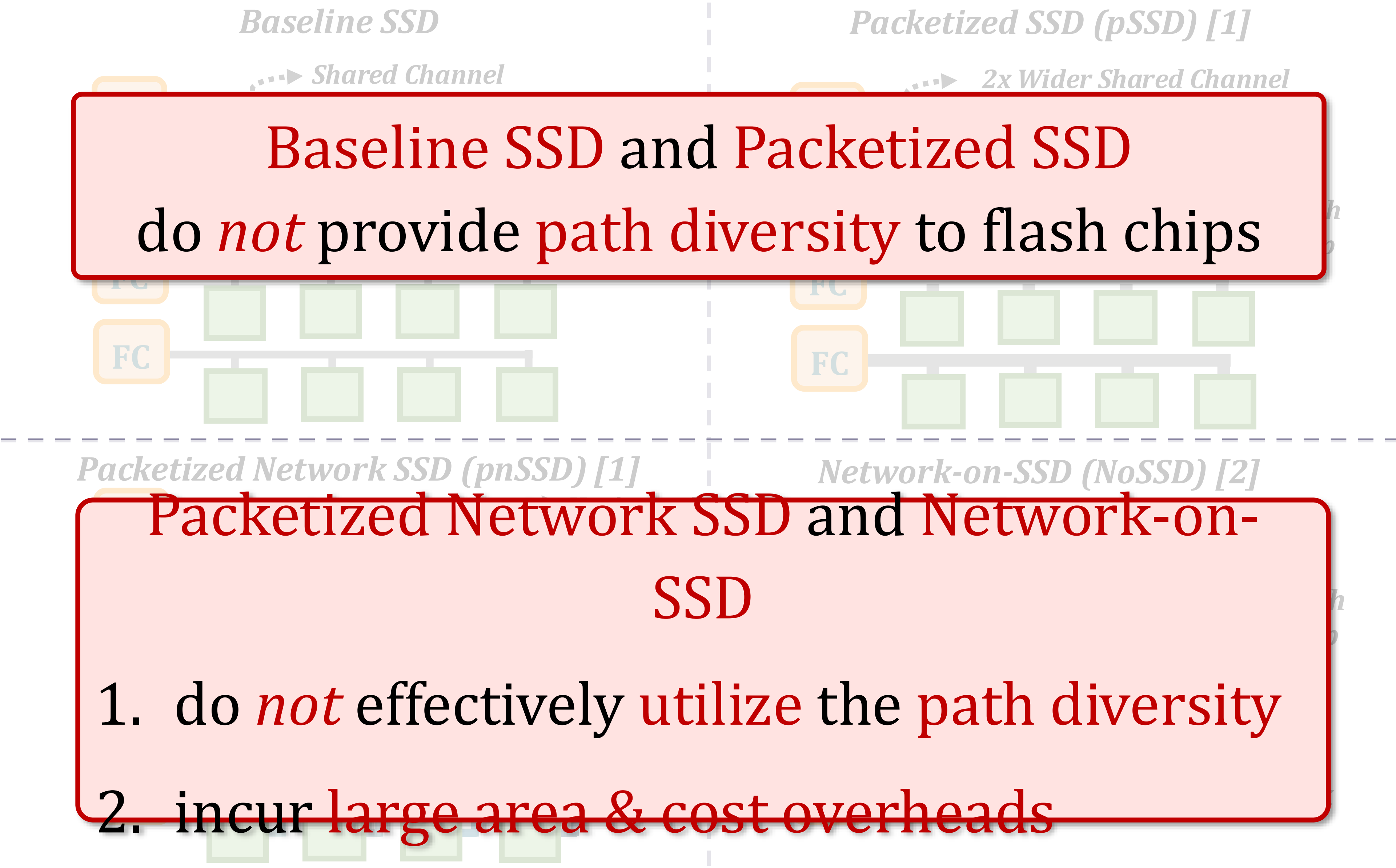


Network-on-SSD (NoSSD) [2]



[1] Kim+, "Networked SSD: Flash Memory Interconnection Network for High-Bandwidth SSD", MICRO 2022
[2] Tavakkol+, "Network-on-SSD: A Scalable and High-Performance Communication Design Paradigm for SSDs", IEEE CAL 2012

Prior Approaches


Baseline SSD
Shared Channel
FC
FC
Packetized SSD (pSSD) [1]
2x Wider Shared Channel
FC
FC
Packetized Network SSD (pnSSD) [1]
Network-on-SSD (NoSSD) [2]

Baseline SSD and Packetized SSD

do *not* provide path diversity to flash chips

Packetized Network SSD and Network-on-SSD

1. do *not* effectively utilize the path diversity
2. incur large area & cost overheads

[1] Kim+, "Networked SSD: Flash Memory Interconnection Network for High-Bandwidth SSD", MICRO 2022

[2] Tavakkol+, "Network-on-SSD: A Scalable and High-Performance Communication Design Paradigm for SSDs", IEEE CAL 2012

Our Goal

To fundamentally address the path conflict problem in SSDs by

1. increasing the number of paths to each flash chip (i.e., path diversity) at low cost

2. effectively utilizing the increased path diversity for communication between the SSD controller and flash chips

Talk Outline

Motivation

Venice

Evaluation

Summary

Our Proposal

Venice

A low-cost interconnection network of flash chips in the SSD

Conflict-free path reservation for each I/O request

A non-minimal fully-adaptive routing algorithm for path identification

Named after the network of canals in the city of Venice
https://en.wikipedia.org/wiki/Venice

Our Proposal

Venice

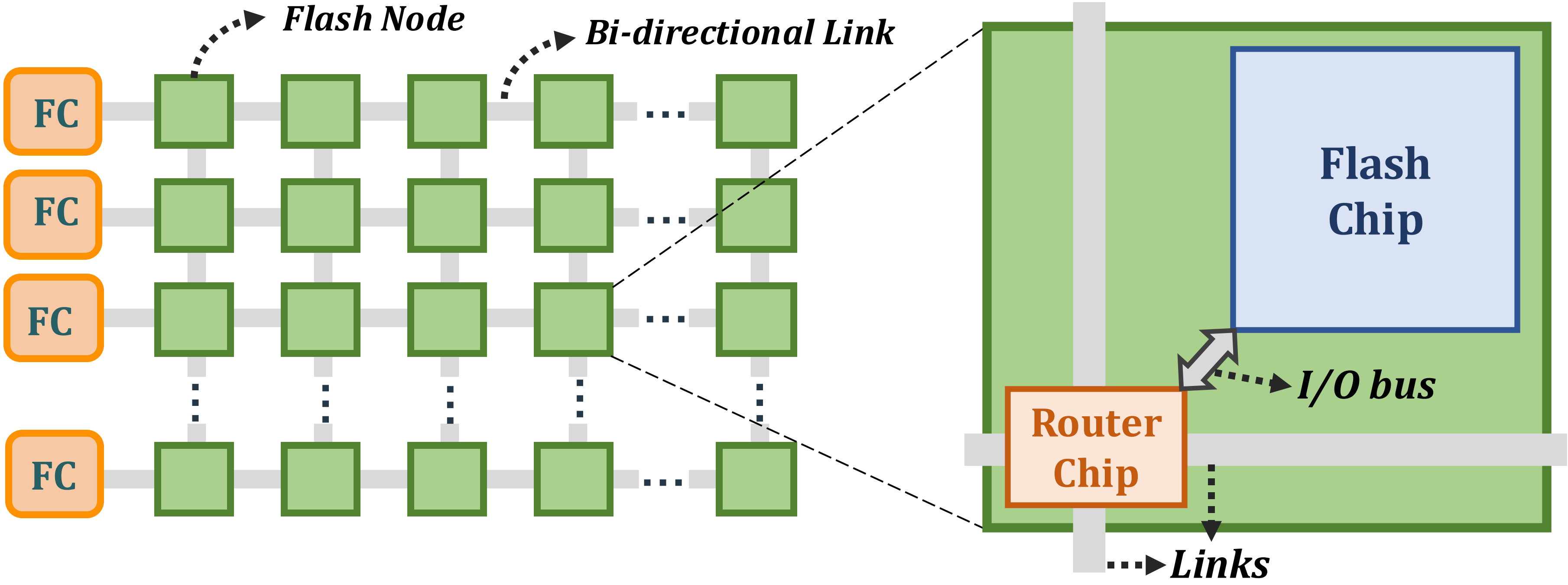
A low-cost interconnection network of flash chips in the SSD

Conflict-free path reservation for each I/O request

A non-minimal fully-adaptive routing algorithm for path identification

Named after the network of canals in the city of Venice
https://en.wikipedia.org/wiki/Venice

Venice: Architecture



Venice provides increased path diversity at low cost

No modifications to existing flash chips in Venice

Our Proposal

Venice

A low-cost interconnection network of flash chips in the SSD

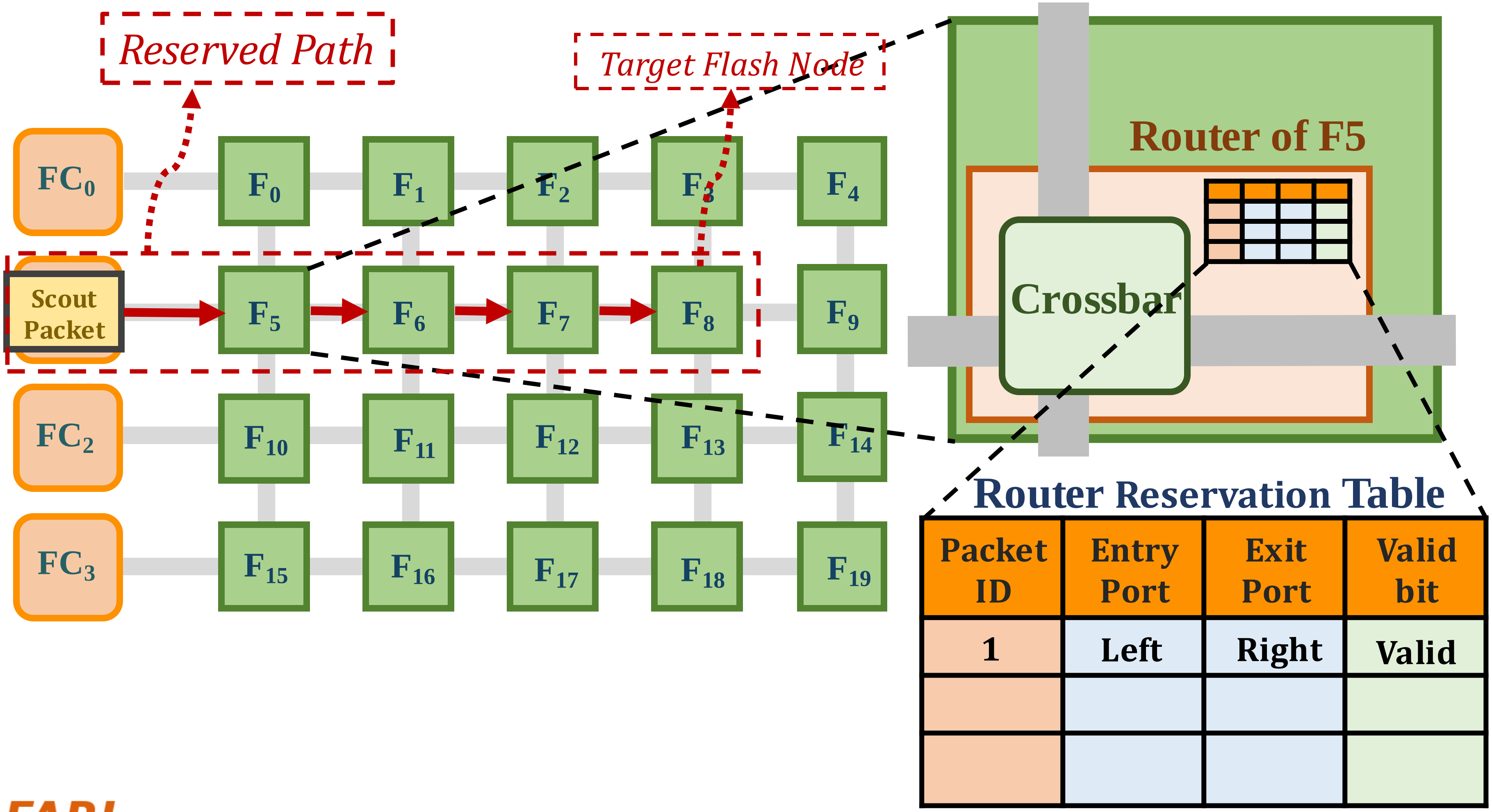
Conflict-free path reservation for each I/O request

A non-minimal fully-adaptive routing algorithm for path identification

Named after the network of canals in the city of Venice
https://en.wikipedia.org/wiki/Venice

Venice: Path Reservation (I)

- Venice uses a small *scout packet* to reserve a conflict-free path for each I/O request

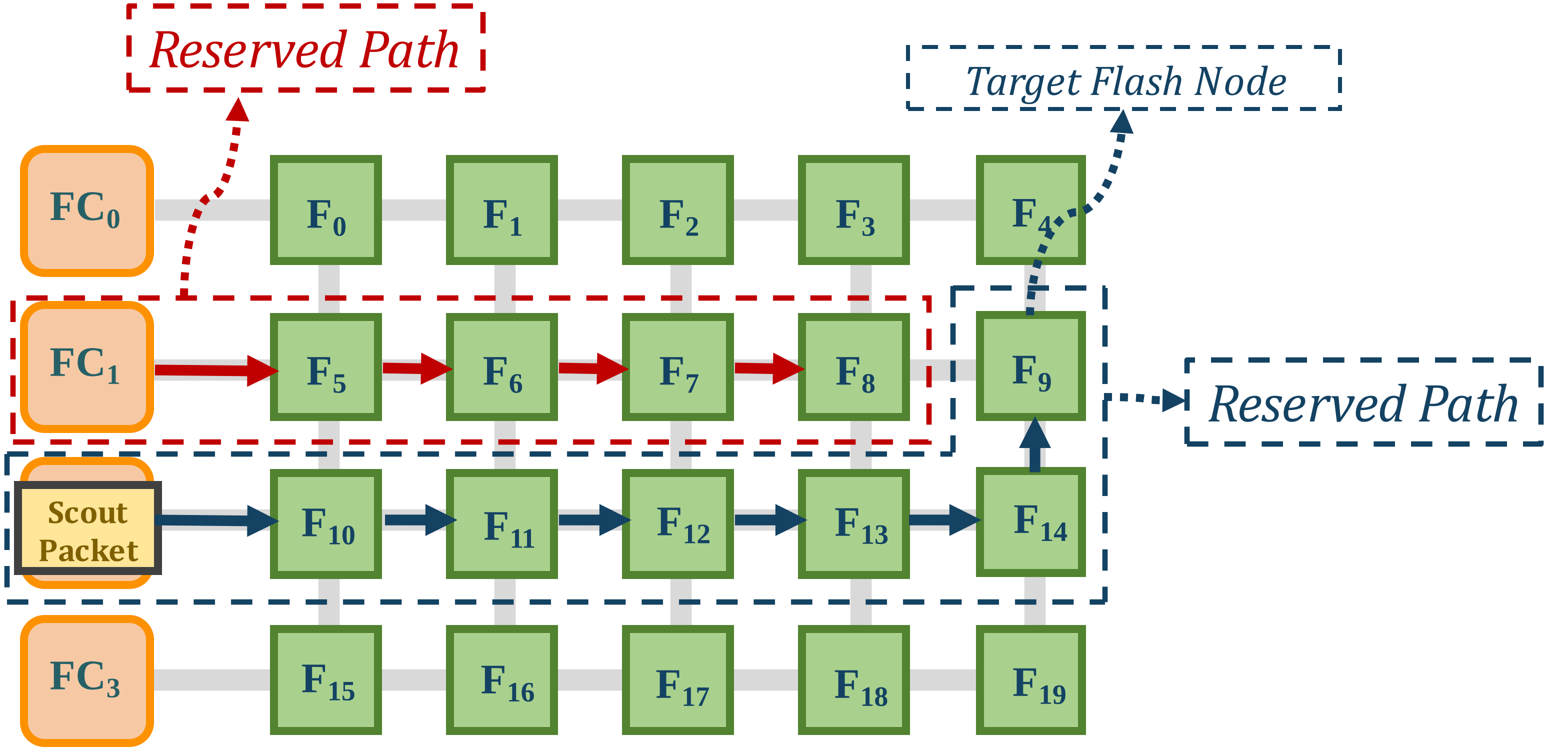


Router Reservation Table

Packet ID	Entry Port	Exit Port	Valid bit
1	Left	Right	Valid

Venice: Path Reservation (II)

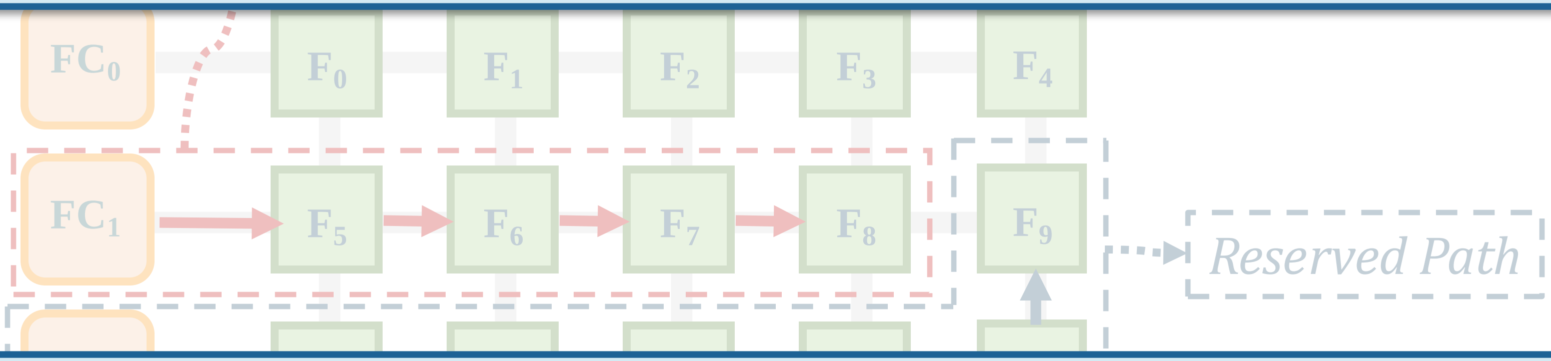
- Venice uses a small *scout packet* to reserve a conflict-free path for each I/O request



Venice: Path Reservation

- Venice uses a small *scout packet* to reserve a

Path reservation eliminates path conflicts by enabling conflict-free I/O transfer



The overhead of path reservation is negligible due to the small size of the scout packet

Our Proposal

Venice

A low-cost interconnection network of flash chips in the SSD

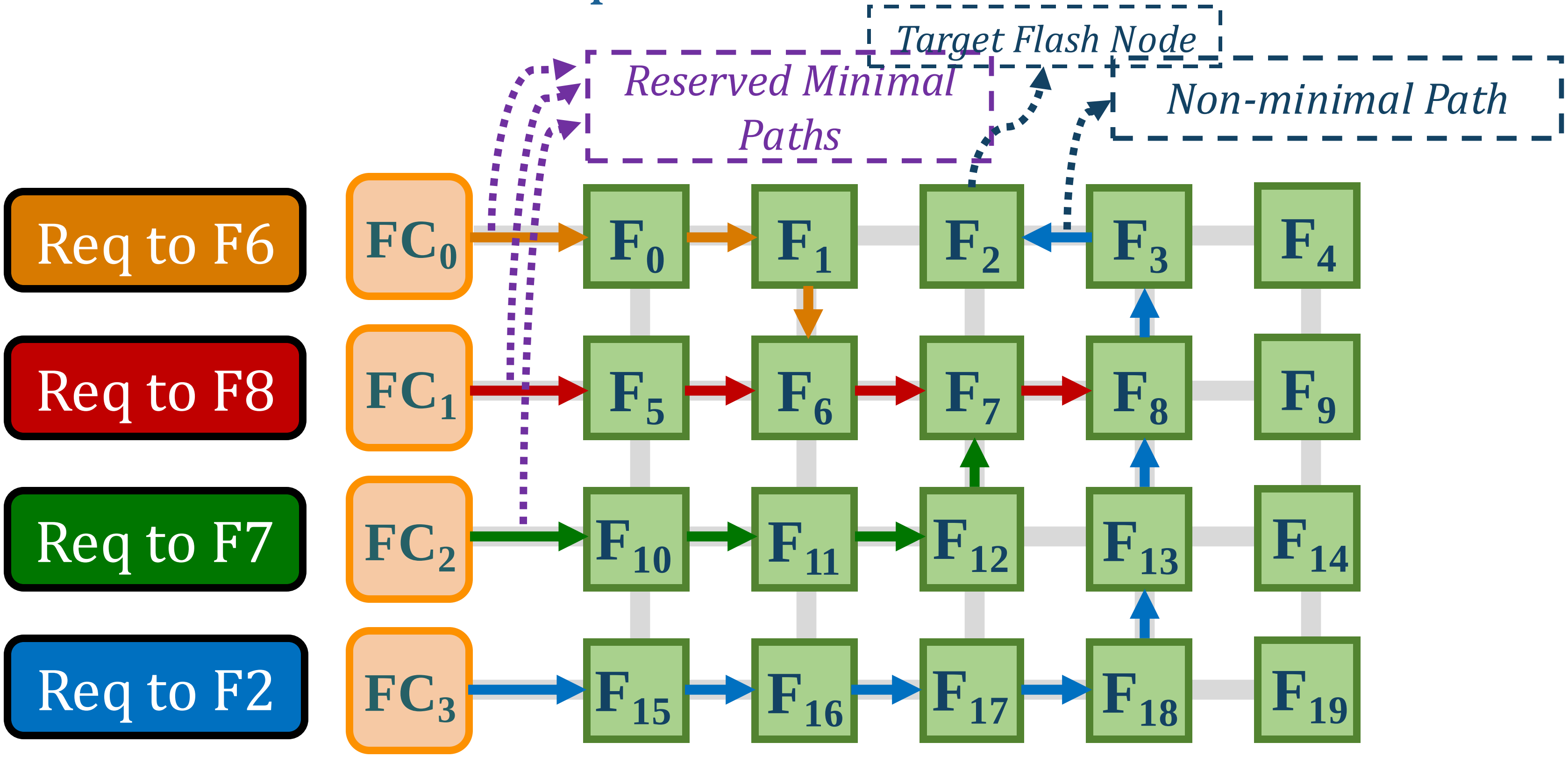
Conflict-free path reservation for each I/O request

A non-minimal fully-adaptive routing algorithm for path identification

Named after the network of canals in the city of Venice
https://en.wikipedia.org/wiki/Venice

Venice: Non-Minimal Fully Adaptive Routing

- Venice uses a non-minimal fully-adaptive routing algorithm to route *scout packets* when a minimal path is unavailable
- Effectively utilizes the idle links in the interconnection network to find a conflict-free path



More in the Paper

- Venice's non-minimal fully-adaptive routing algorithm

- Handling deadlock and livelock scenarios

- Overhead of exercising a non-minimal path

- Analysis of prior architectures proposed to mitigate the path conflict problem

- Detailed background on modern SSD architecture

Talk Outline

Motivation

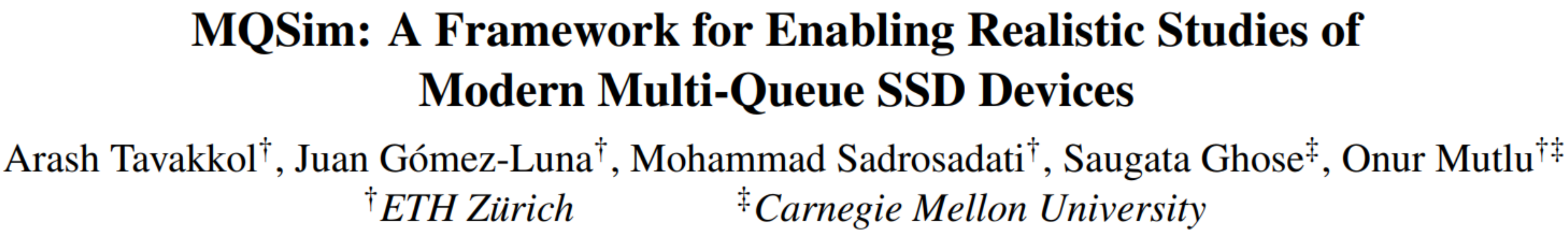
Venice

Evaluation

Summary

Simulating SSDs: MQSim [FAST 2018]

- Arash Tavakkol, Juan Gomez-Luna, Mohammad Sadrosadati, Saugata Ghose, and Onur Mutlu,
"MQSim: A Framework for Enabling Realistic Studies of Modern Multi-Queue SSD Devices"
*Proceedings of the 16th USENIX Conference on File and Storage Technologies (**FAST**)*, Oakland, CA, USA, February 2018.
[Slides (pptx) (pdf)]
[Source Code]



MQSim: A Framework for Enabling Realistic Studies of Modern Multi-Queue SSD Devices

Arash Tavakkol[†], Juan Gómez-Luna[†], Mohammad Sadrosadati[†], Saugata Ghose[‡], Onur Mutlu[†‡]
[†] *ETH Zürich* [‡] *Carnegie Mellon University*

https://github.com/CMU-SAFARI/MQSim

https://people.inf.ethz.ch/omutlu/pub/MQSim-SSD-simulation-framework_fast18.pdf

Simulating Memory: Ramulator 2.0

- Haocong Luo, Yahya Can Tugrul, F. Nisa Bostanci, Ataberk Olgun, A. Giray Yaglikci, and Onur Mutlu,
  "Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"
  *Preprint on **arxiv***, August 2023.
  [arXiv version]
  [Ramulator 2.0 Source Code]

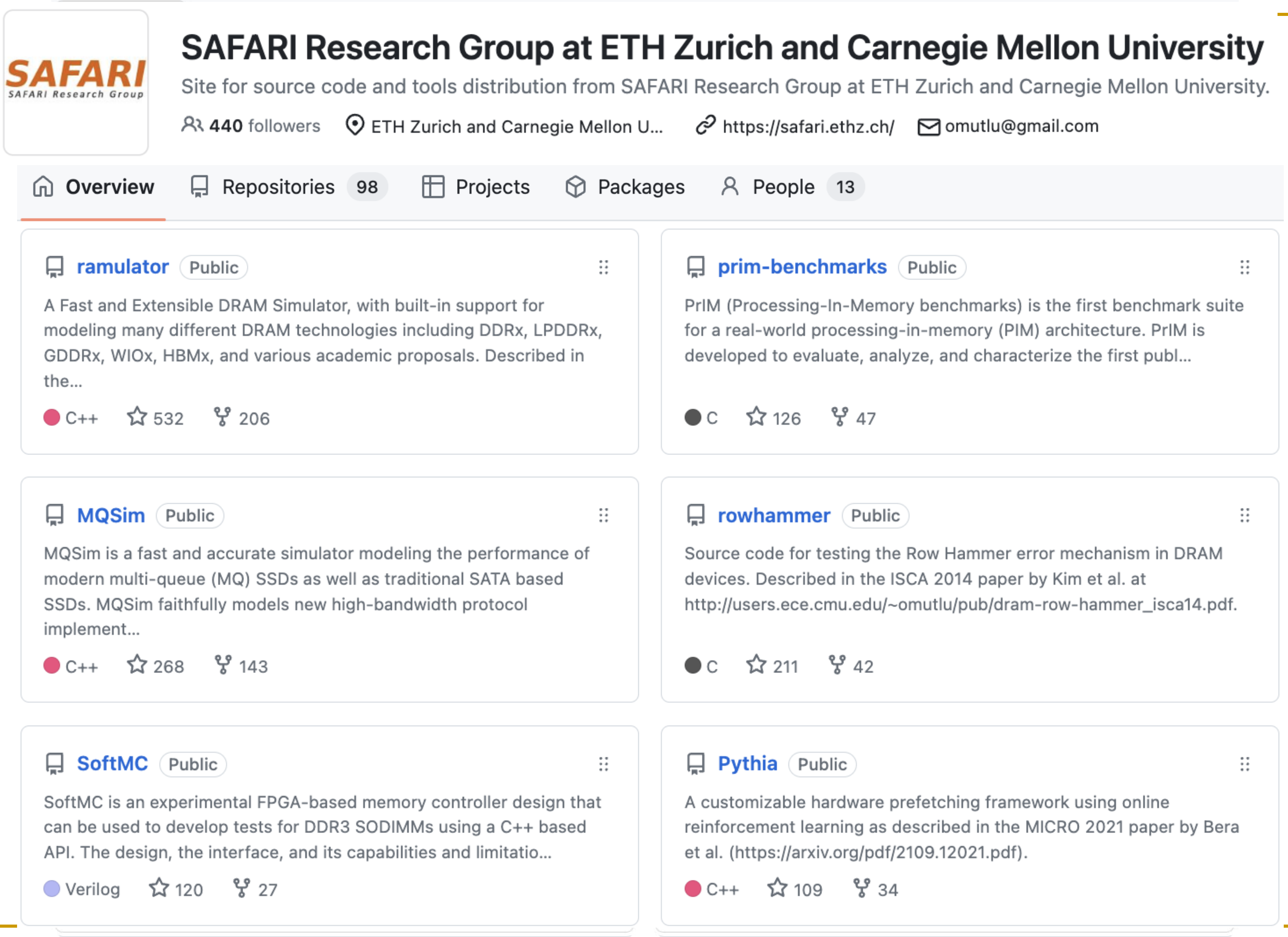
Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

https://arxiv.org/pdf/2308.11030.pdf

https://github.com/CMU-SAFARI/ramulator2

Open Source Tools: SAFARI GitHub



SAFARI
SAFARI Research Group

SAFARI Research Group at ETH Zurich and Carnegie Mellon University

Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University.

440 followers · ETH Zurich and Carnegie Mellon U... · https://safari.ethz.ch/ · omutlu@gmail.com

ramulator Public

A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the...

● C++ ☆ 532 206

prim-benchmarks Public

PrIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publ...

● C ☆ 126 47

MQSim Public

MQSim is a fast and accurate simulator modeling the performance of modern multi-queue (MQ) SSDs as well as traditional SATA based SSDs. MQSim faithfully models new high-bandwidth protocol implement...

● C++ ☆ 268 143

rowhammer Public

Source code for testing the Row Hammer error mechanism in DRAM devices. Described in the ISCA 2014 paper by Kim et al. at http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer_isca14.pdf.

● C ☆ 211 42

SoftMC Public

SoftMC is an experimental FPGA-based memory controller design that can be used to develop tests for DDR3 SODIMMs using a C++ based API. The design, the interface, and its capabilities and limitatio...

● Verilog ☆ 120 27

Pythia Public

A customizable hardware prefetching framework using online reinforcement learning as described in the MICRO 2021 paper by Bera et al. (https://arxiv.org/pdf/2109.12021.pdf).

● C++ ☆ 109 34

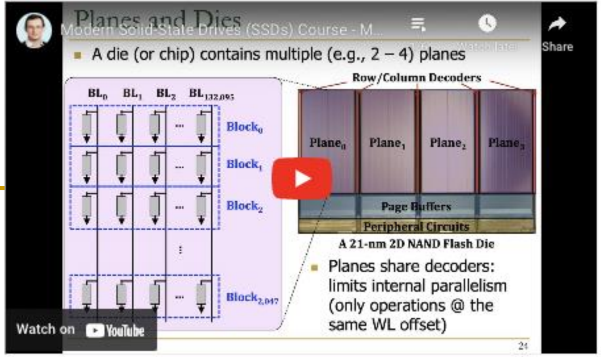
https://github.com/CMU-SAFARI/

SSD Course (Spring 2023)

- **Spring 2023 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2023/doku.php?id=modern_ssds
- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=modern_ssds

- **Youtube Livestream (Spring 2023):**
  - https://www.youtube.com/watch?v=4VTwOMmsnJY&list=PL5Q2soXY2Zi_8qOM5Icpp8hB2SHtm4z57&pp=iAQB
- **Youtube Livestream (Fall 2022):**
  - https://www.youtube.com/watch?v=hqLrd-Uj0aU&list=PL5Q2soXY2Zi9BJhenUq4JI5bwhAMpAp13&pp=iAQB

- Project course
  - Taken by Bachelor's/Master's students
  - SSD Basics and Advanced Topics
  - Hands-on research exploration
  - Many research readings

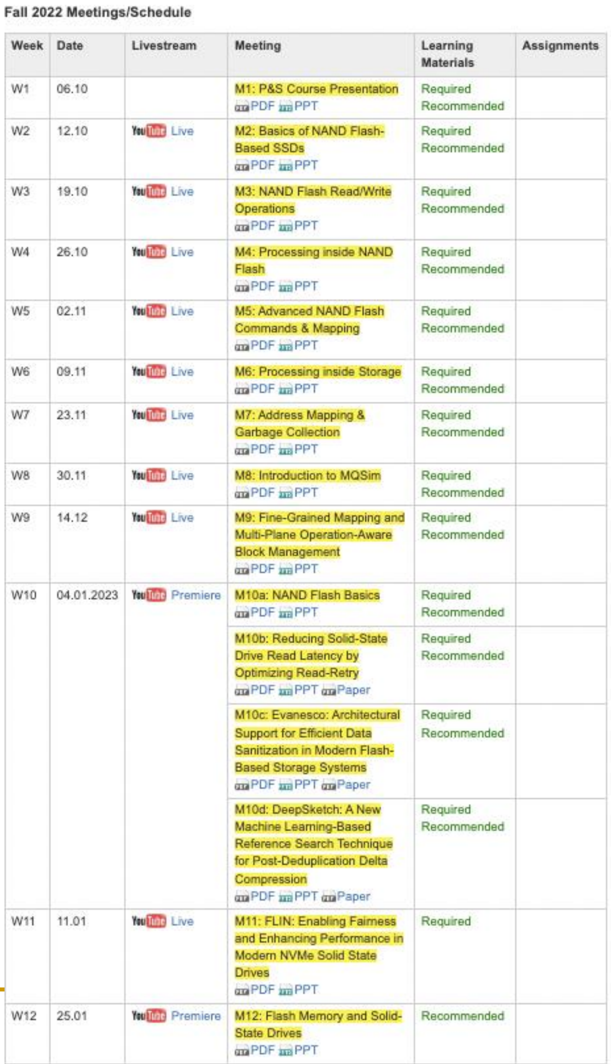


Planes and Dies

Modern Solid-State Drives (SSDs) Course - M...

- A die (or chip) contains multiple (e.g., 2 – 4) planes
- Planes share decoders: limits internal parallelism (only operations @ the same WL offset)

A 21-nm 2D NAND Flash Die



Fall 2022 Meetings/Schedule

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	06.10		M1: P&S Course Presentation PDF PPT	Required Recommended	
W2	12.10	YouTube Live	M2: Basics of NAND Flash-Based SSDs PDF PPT	Required Recommended	
W3	19.10	YouTube Live	M3: NAND Flash Read/Write Operations PDF PPT	Required Recommended	
W4	26.10	YouTube Live	M4: Processing inside NAND Flash PDF PPT	Required Recommended	
W5	02.11	YouTube Live	M5: Advanced NAND Flash Commands & Mapping PDF PPT	Required Recommended	
W6	09.11	YouTube Live	M6: Processing inside Storage PDF PPT	Required Recommended	
W7	23.11	YouTube Live	M7: Address Mapping & Garbage Collection PDF PPT	Required Recommended	
W8	30.11	YouTube Live	M8: Introduction to MQSim PDF PPT	Required Recommended	
W9	14.12	YouTube Live	M9: Fine-Grained Mapping and Multi-Plane Operation-Aware Block Management PDF PPT	Required Recommended	
W10	04.01.2023	YouTube Premiere	M10a: NAND Flash Basics PDF PPT	Required Recommended	
			M10b: Reducing Solid-State Drive Read Latency by Optimizing Read-Retry PDF PPT Paper	Required Recommended	
			M10c: Evanesco: Architectural Support for Efficient Data Sanitization in Modern Flash-Based Storage Systems PDF PPT Paper	Required Recommended	
			M10d: DeepSketch: A New Machine Learning-Based Reference Search Technique for Post-Deduplication Delta Compression PDF PPT Paper	Required Recommended	
W11	11.01	YouTube Live	M11: FLIN: Enabling Fairness and Enhancing Performance in Modern NVMe Solid State Drives PDF PPT	Required	
W12	25.01	YouTube Premiere	M12: Flash Memory and Solid-State Drives PDF PPT	Recommended	

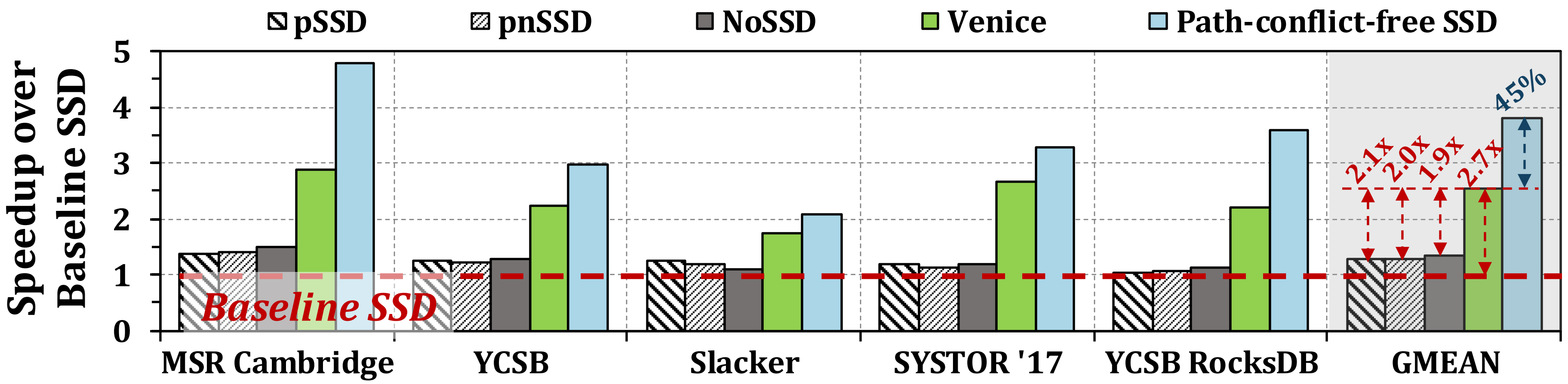
https://www.youtube.com/onurmutlulectures

Evaluation Methodology

- **Using MQSim [Tavakkol+, FAST'18]**, a state-of-the-art SSD simulator
- **Two SSD configurations**
  - Performance-Optimized (Samsung Z-NAND SSD)
  - Cost-Optimized (Samsung PM9A3)
- **Nineteen data-intensive workloads from**
  - MSR Cambridge, YCSB, Slacker, SYSTOR '17 and RocksDB
- **Prior Approaches**
  - Baseline SSD: A typical multi-channel shared bus SSD
  - Packetized SSD (pSSD) [Kim+, MICRO'22]: Uses packetization to double the flash channel bandwidth
  - Packetized Network SSD (pnSSD) [Kim+, MICRO'22]: Increases path diversity by introducing vertical channels
  - Network-on-SSD (NoSSD) [Tavakkol+, CAL 2012]: Proposes an interconnection network of flash chips with simple deterministic routing
  - Path-conflict-free SSD: An *ideal SSD* with no path conflicts

Results: Performance Analysis (I)

- Performance-Optimized SSD

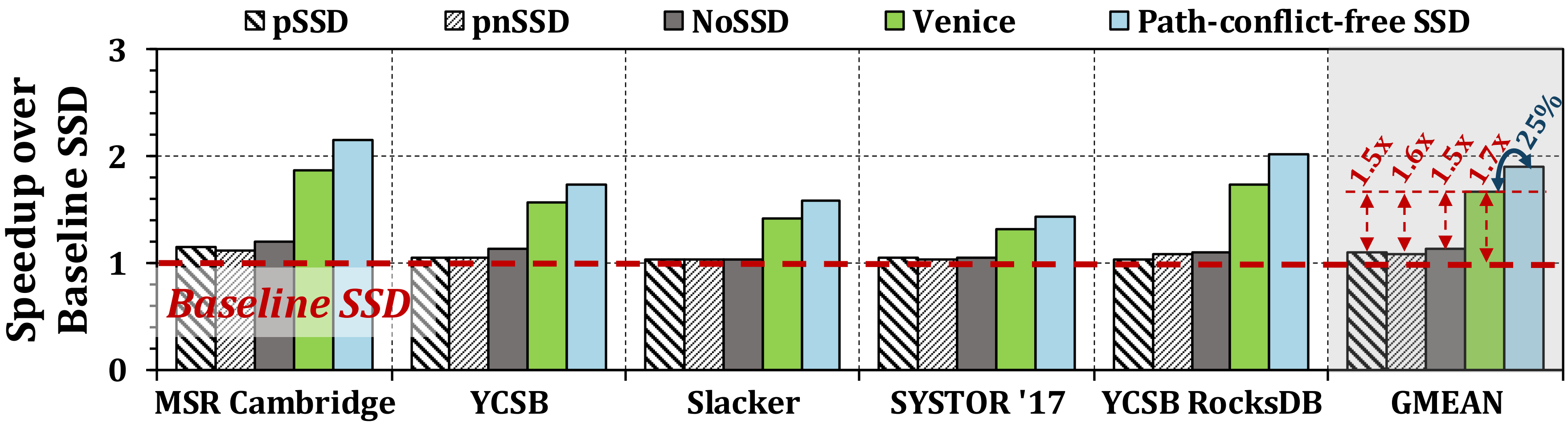


Venice improves SSD performance by 1.9x
on average over the best-performing prior work

Venice's performance is within 45%
of the performance of a Path-conflict-free SSD

Results: Performance Analysis (II)

- Cost-Optimized SSD

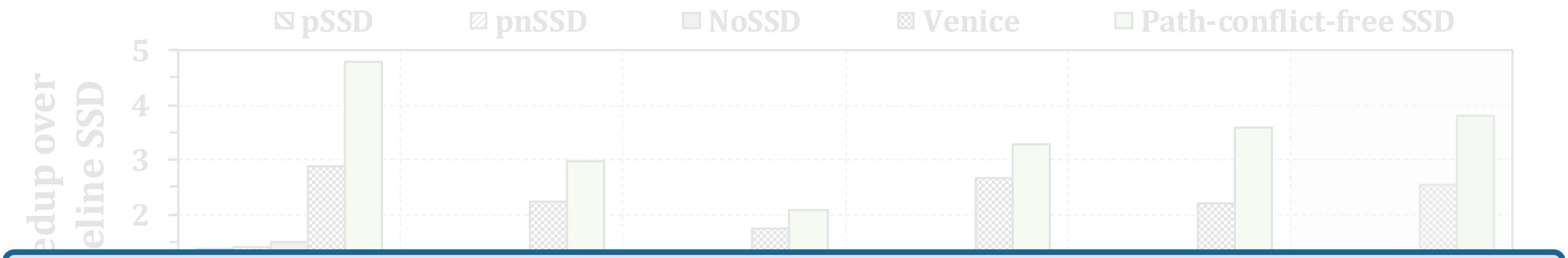


Venice improves SSD performance by 1.5x on average over the best-performing prior work

Venice's performance is within 25% of the performance of a Path-conflict-free SSD

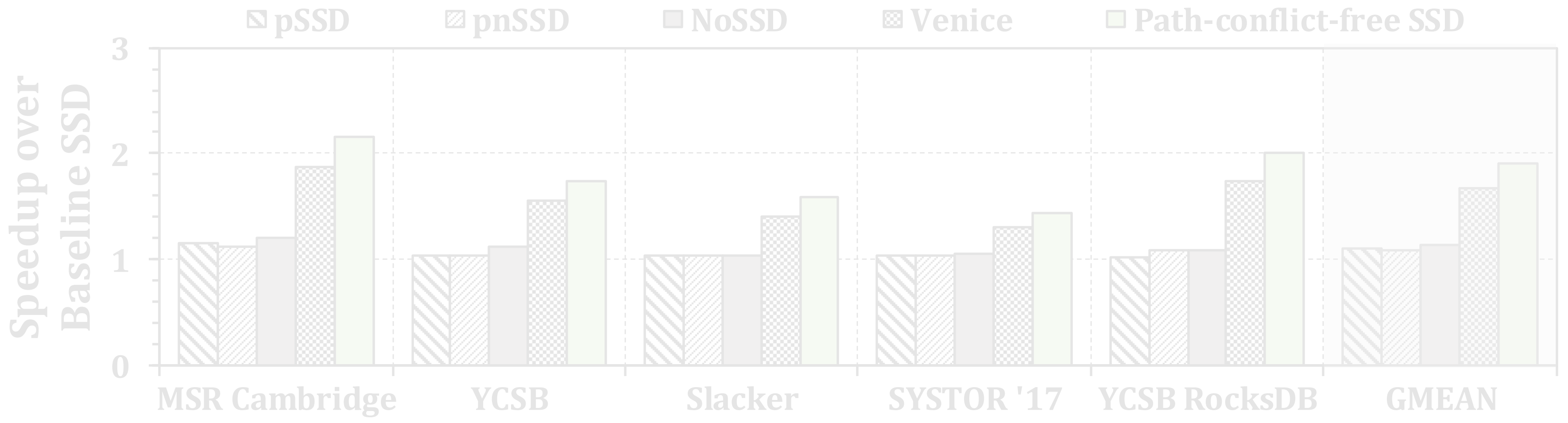
Results: Performance Analysis (III)

- Performance-Optimized SSD

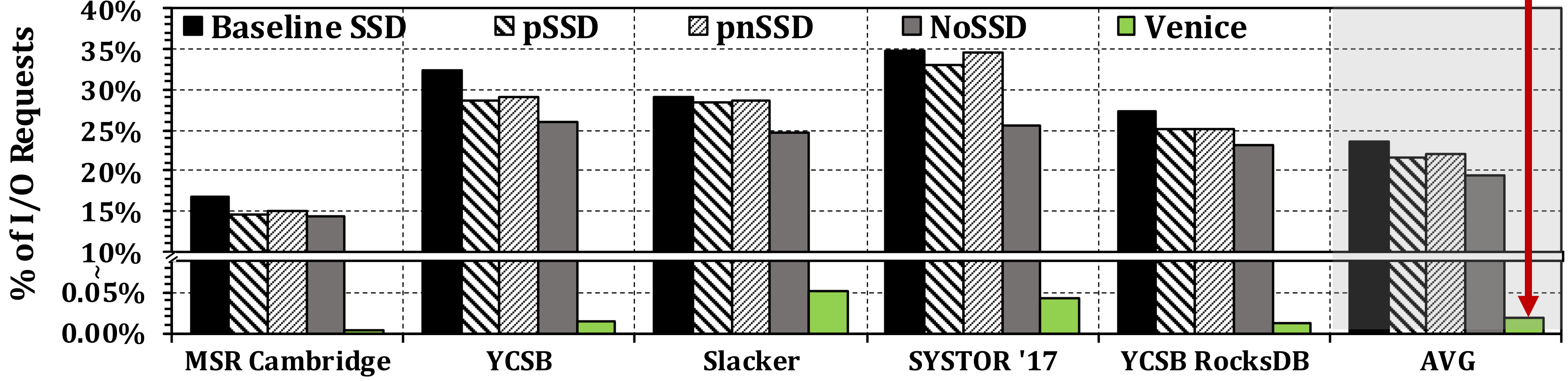


Venice provides significant improvement in performance over all prior approaches

- Cost-Optimized SSD



Results: Reduction in Path Conflicts

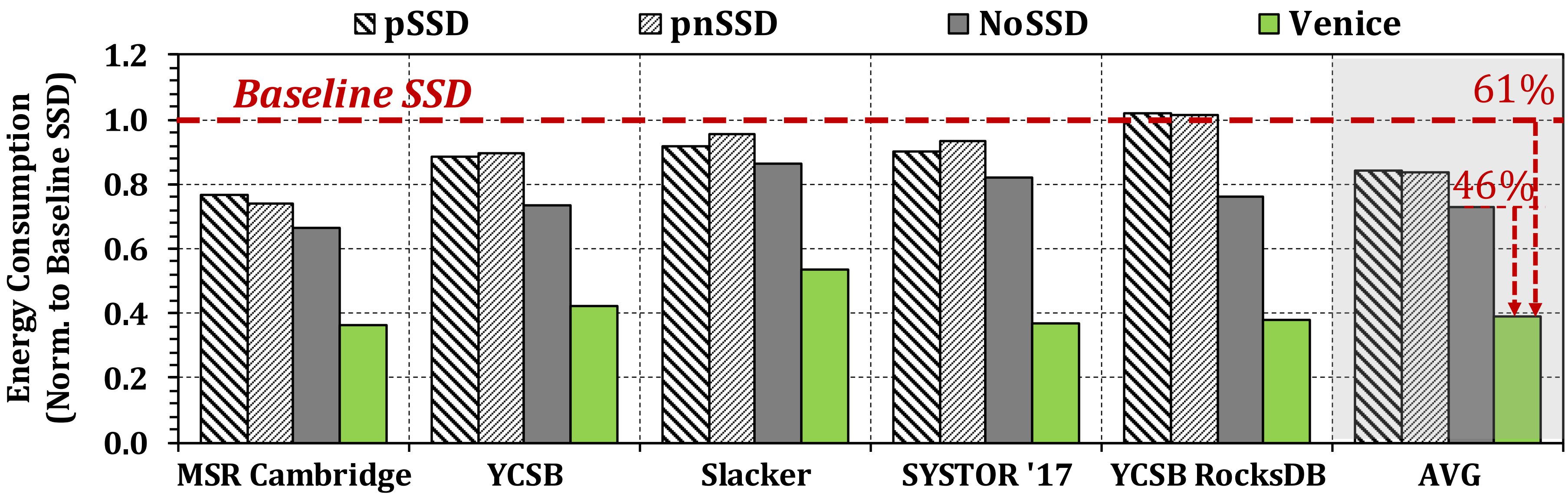


99.98% of I/O requests do not experience path conflicts

Venice mitigates path conflicts by using path reservation and effective utilization of path diversity

SAFARI

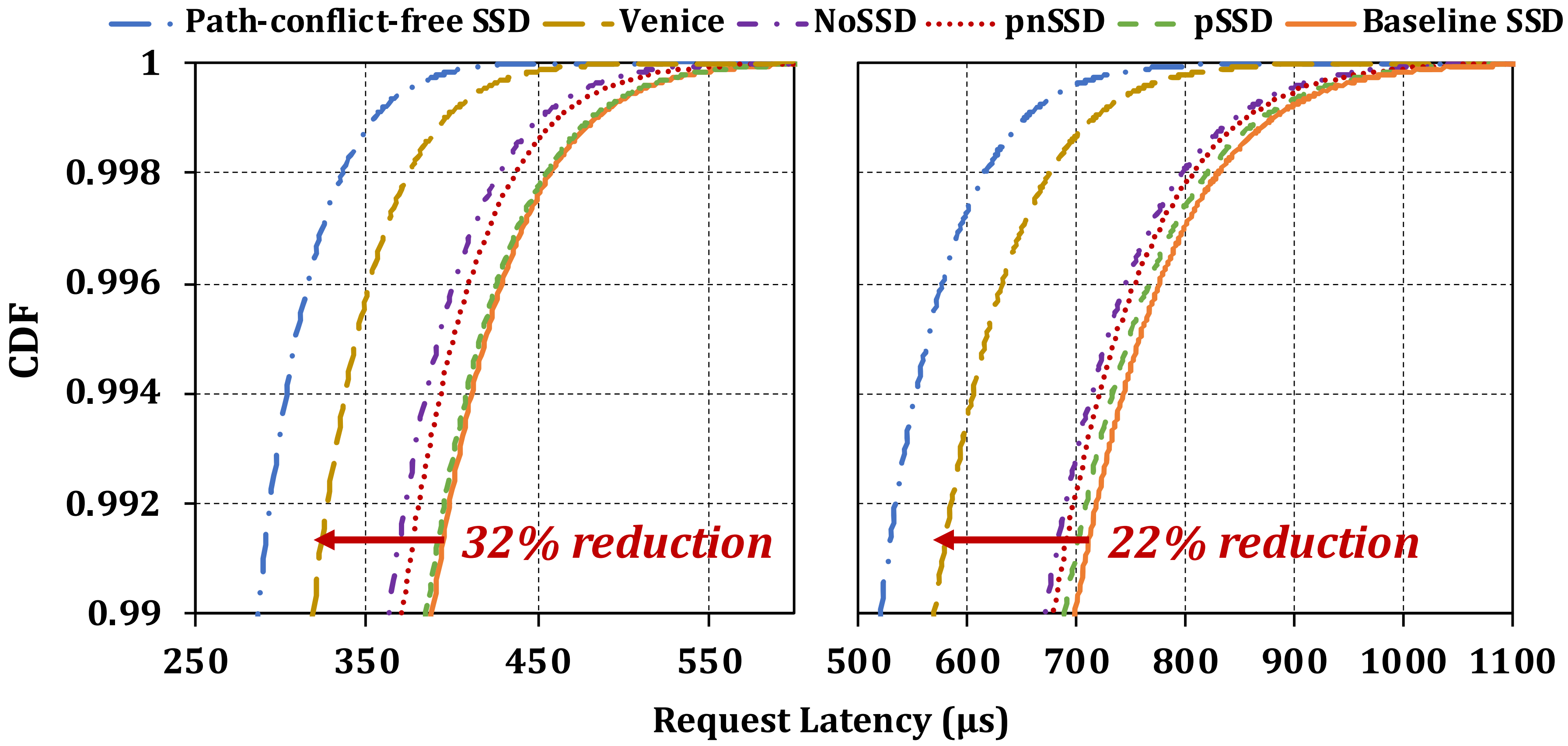
Results: SSD Energy Consumption



Venice reduces the SSD energy consumption
by 46% on average
over the most efficient prior work

Tail Latency

- Comparison of tail latencies in the 99th percentile of I/O requests

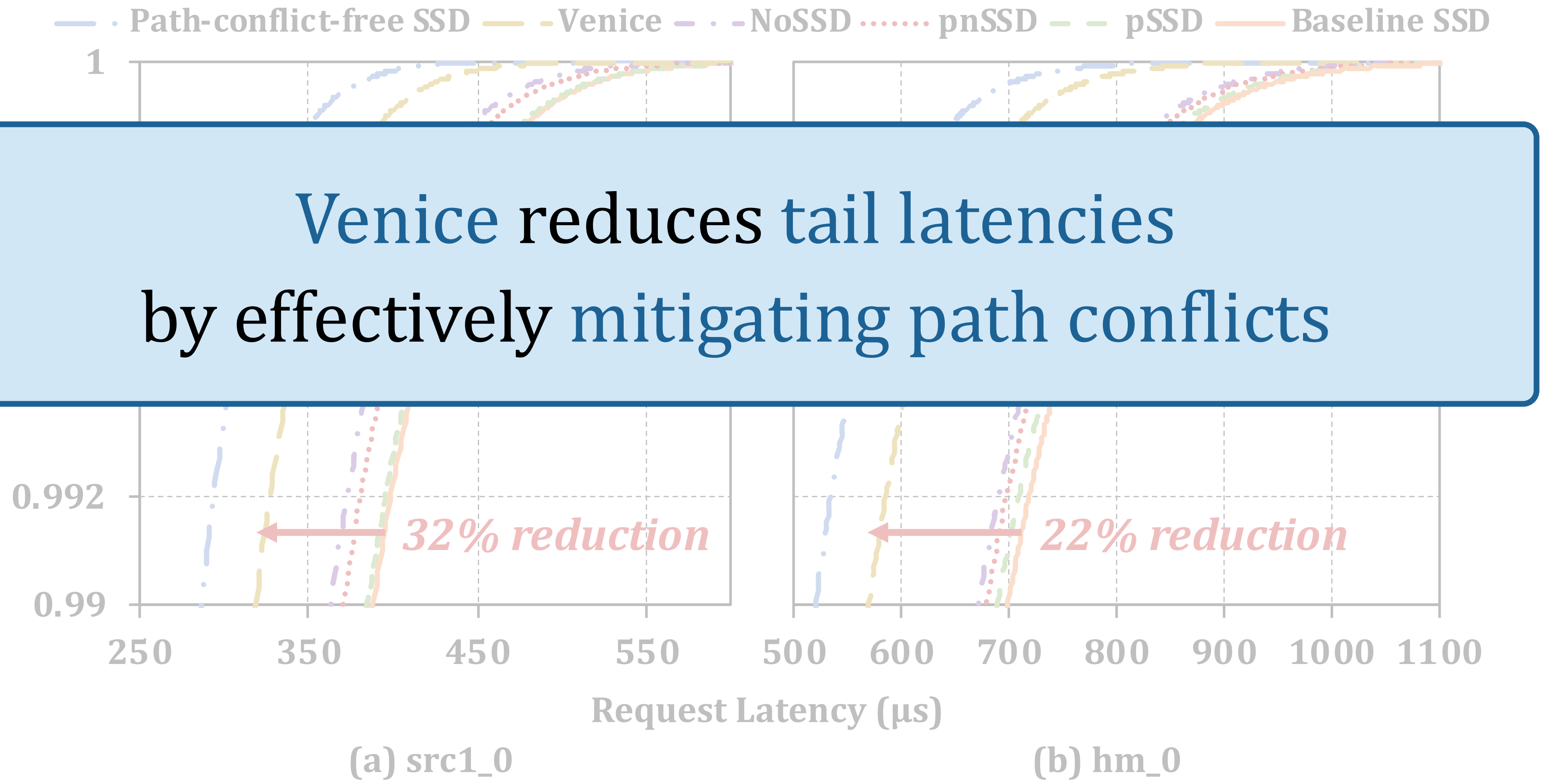


(a) src1_0

(b) hm_0

Tail Latency

- Comparison of tail latencies in the 99th percentile of I/O requests

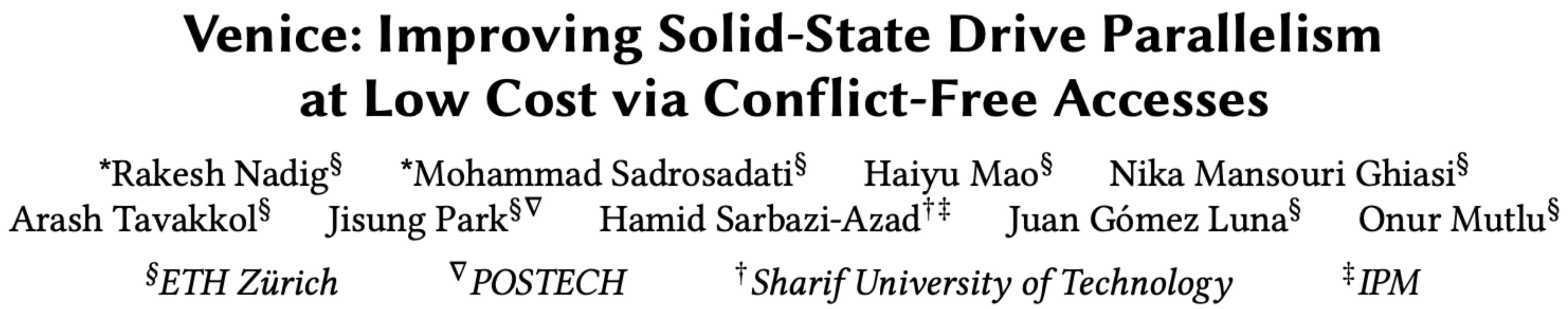


Venice reduces tail latencies
by effectively mitigating path conflicts

More in the Paper

- Power and area overhead analysis
- Tail latency analysis
- Sensitivity to interconnection network configurations
- Performance on mixed workloads
- Detailed evaluation methodology

More in the Paper



Venice: Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses

*Rakesh Nadig[§] *Mohammad Sadrosadati[§] Haiyu Mao[§] Nika Mansouri Ghiasi[§]
Arash Tavakkol[§] Jisung Park[§∇] Hamid Sarbazi-Azad[†‡] Juan Gómez Luna[§] Onur Mutlu[§]

[§]*ETH Zürich* [∇]*POSTECH* [†]*Sharif University of Technology* [‡]*IPM*

https://arxiv.org/abs/2305.07768

Talk Outline

Motivation

Venice

Evaluation

Summary

Venice: Summary

Mitigates path conflicts by efficiently utilizing the path diversity of the SSD interconnection network

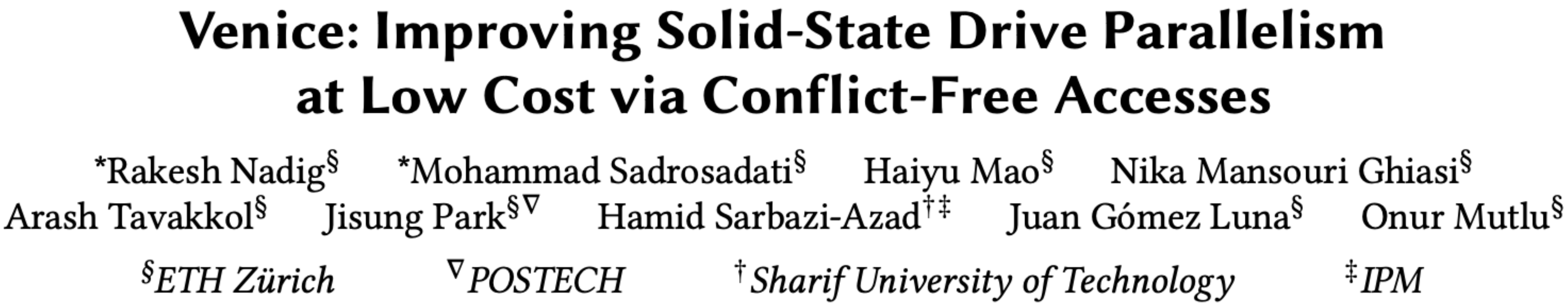
Improves performance
by 1.9x/1.5x over the best-performing prior work
on performance-optimized/cost-optimized SSD

Reduces energy consumption
by 46% on average over the most efficient prior work

Low-cost and requires
no changes to commodity flash chips

Venice Paper, Slides, Video [ISCA 2023]

- Rakesh Nadig, Mohammad Sadrosadati, Haiyu Mao, Nika Mansouri Ghiasi, Arash Tavakkol, Jisung Park, Hamid Sarbazi-Azad, Juan Gómez Luna, and Onur Mutlu,
"Venice: Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses"
Proceedings of the 50th International Symposium on Computer Architecture (***ISCA***),
Orlando, FL, USA, June 2023.
[arXiv version]
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (3 minutes)]
[Talk Video (14 minutes, including Q&A)]



Venice: Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses

*Rakesh Nadig[§] *Mohammad Sadrosadati[§] Haiyu Mao[§] Nika Mansouri Ghiasi[§]
Arash Tavakkol[§] Jisung Park[§∇] Hamid Sarbazi-Azad[†‡] Juan Gómez Luna[§] Onur Mutlu[§]

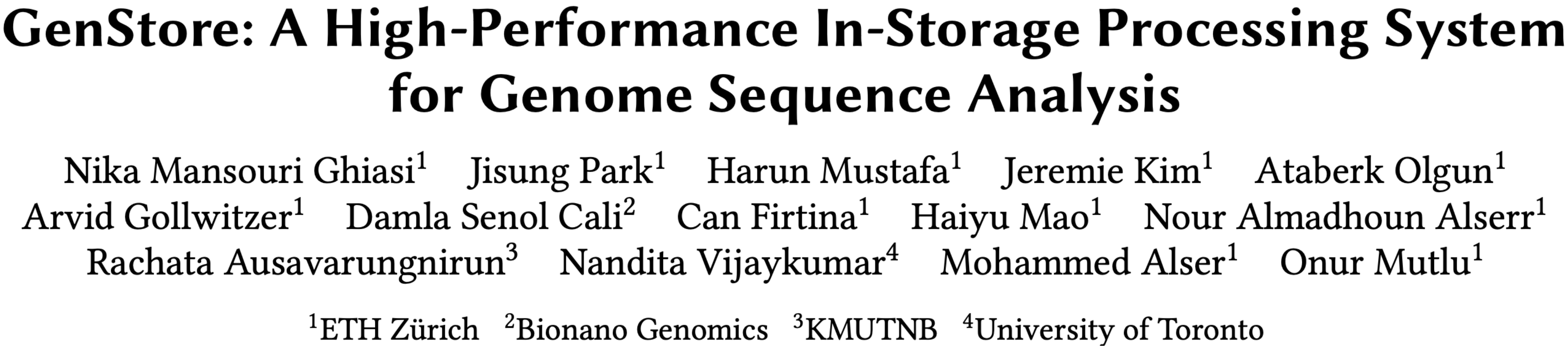
[§]*ETH Zürich* [∇]*POSTECH* [†]*Sharif University of Technology* [‡]*IPM*

https://arxiv.org/pdf/2305.07768

Venice
Can Enable More Effective
In-Storage Processing

In-Storage Genomic Data Filtering [ASPLOS 2022]

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu, **"GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"**
  Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (***ASPLOS***), Virtual, February-March 2022.
  [Lightning Talk Slides (pptx) (pdf)]
  [Lightning Talk Video (90 seconds)]



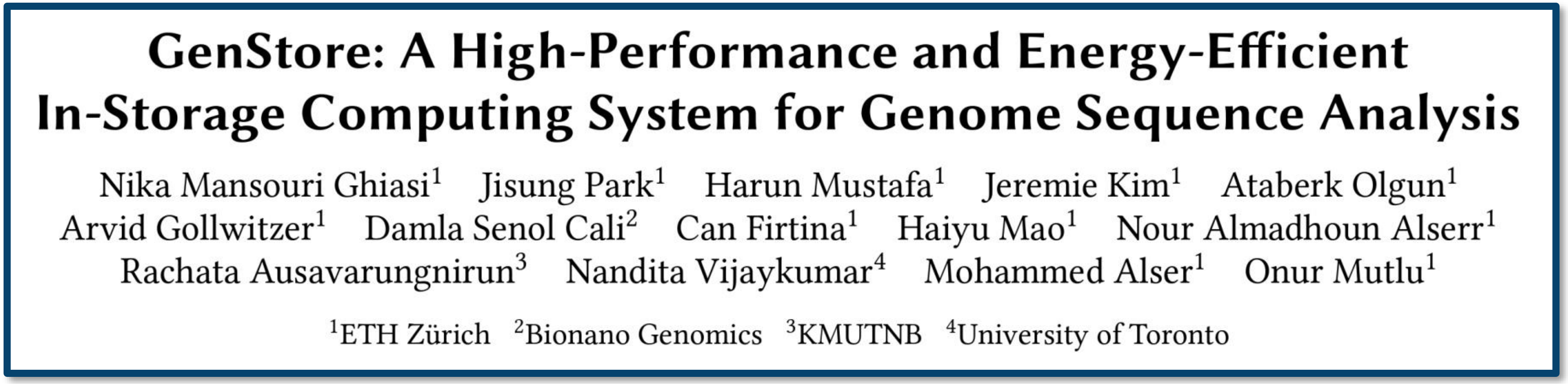
GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi[1] Jisung Park[1] Harun Mustafa[1] Jeremie Kim[1] Ataberk Olgun[1]
Arvid Gollwitzer[1] Damla Senol Cali[2] Can Firtina[1] Haiyu Mao[1] Nour Almadhoun Alserr[1]
Rachata Ausavarungnirun[3] Nandita Vijaykumar[4] Mohammed Alser[1] Onur Mutlu[1]

[1]ETH Zürich [2]Bionano Genomics [3]KMUTNB [4]University of Toronto

https://arxiv.org/abs/2202.10400

GenStore



GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis

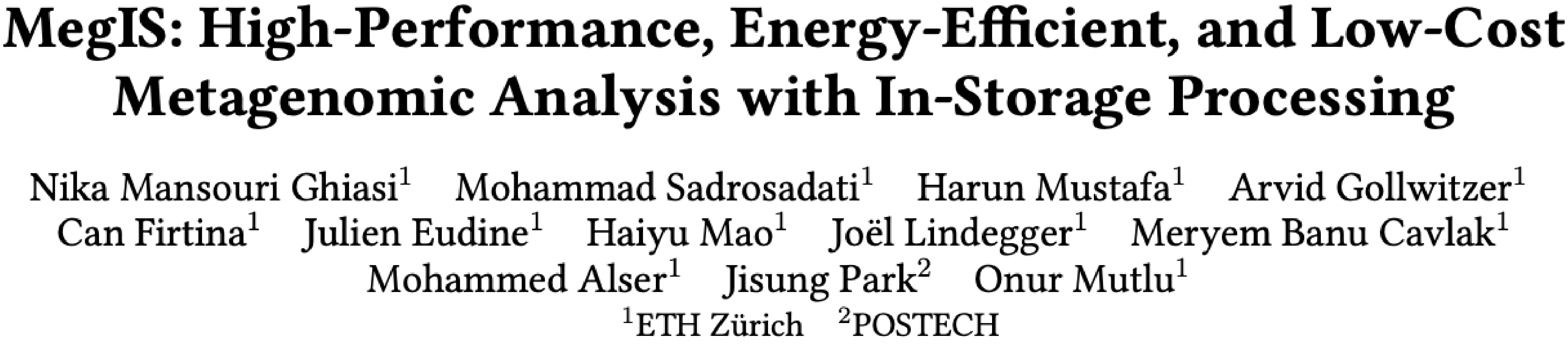
Nika Mansouri Ghiasi[1] Jisung Park[1] Harun Mustafa[1] Jeremie Kim[1] Ataberk Olgun[1]
Arvid Gollwitzer[1] Damla Senol Cali[2] Can Firtina[1] Haiyu Mao[1] Nour Almadhoun Alserr[1]
Rachata Ausavarungnirun[3] Nandita Vijaykumar[4] Mohammed Alser[1] Onur Mutlu[1]

[1]ETH Zürich [2]Bionano Genomics [3]KMUTNB [4]University of Toronto

https://arxiv.org/abs/2202.10400

In-Storage Metagenomics [ISCA 2024]

- Nika Mansouri Ghiasi, Mohammad Sadrosadati, Harun Mustafa, Arvid Gollwitzer, Can Firtina, Julien Eudine, Haiyu Mao, Joel Lindegger, Meryem Banu Cavlak, Mohammed Alser, Jisung Park, and Onur Mutlu,
  "MegIS: High-Performance and Low-Cost Metagenomic Analysis with In-Storage Processing"
  *Proceedings of the 51st Annual International Symposium on Computer Architecture (**ISCA**)*, Buenos Aires, Argentina, July 2024.
  [Slides (pptx) (pdf)]
  [arXiv version]

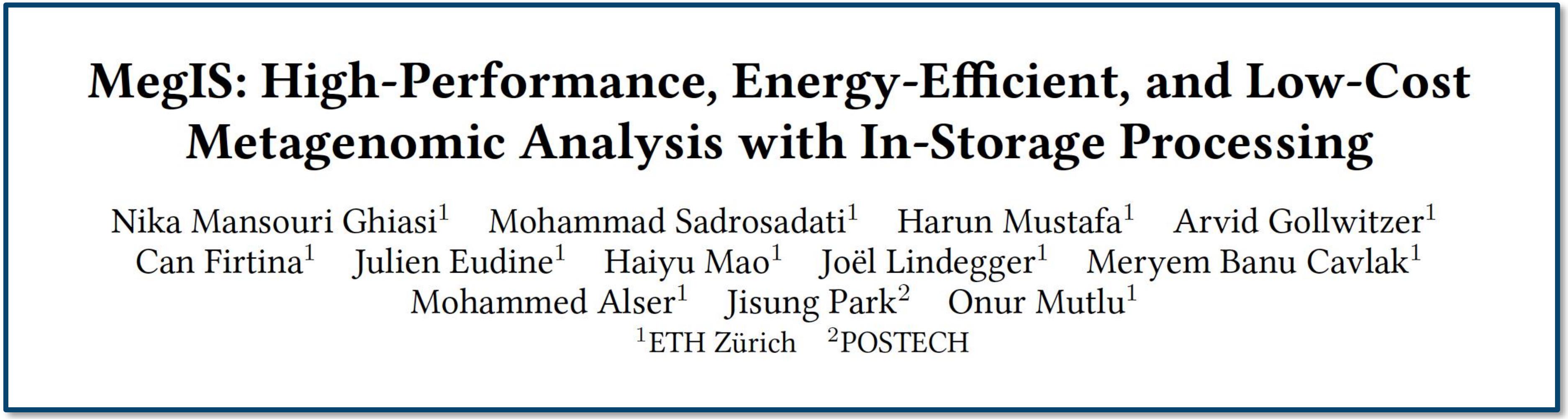


MegIS: High-Performance, Energy-Efficient, and Low-Cost Metagenomic Analysis with In-Storage Processing

Nika Mansouri Ghiasi[1] Mohammad Sadrosadati[1] Harun Mustafa[1] Arvid Gollwitzer[1]
Can Firtina[1] Julien Eudine[1] Haiyu Mao[1] Joël Lindegger[1] Meryem Banu Cavlak[1]
Mohammed Alser[1] Jisung Park[2] Onur Mutlu[1]
[1]ETH Zürich [2]POSTECH

https://arxiv.org/pdf/2406.19113

MegIS

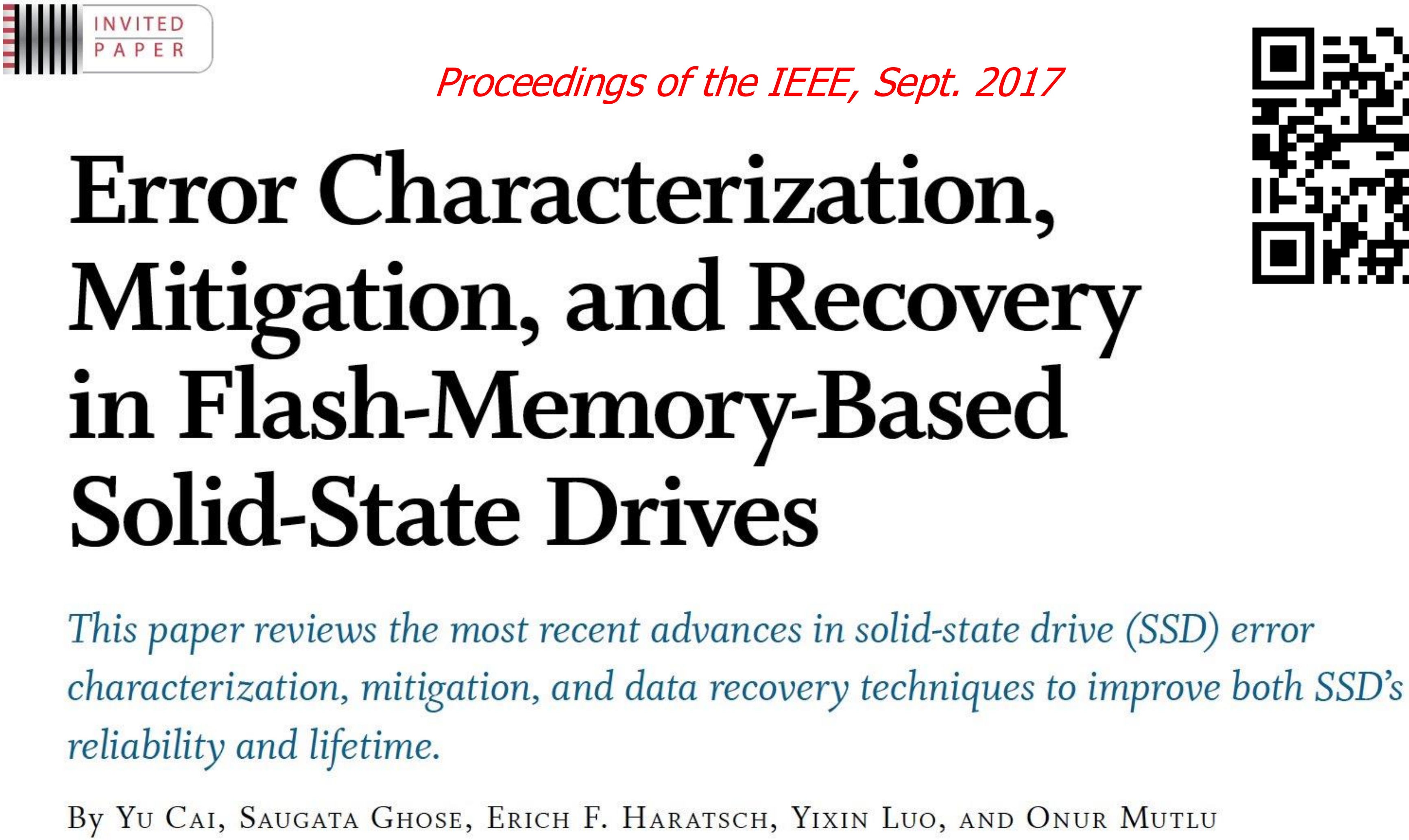


MegIS: High-Performance, Energy-Efficient, and Low-Cost Metagenomic Analysis with In-Storage Processing

Nika Mansouri Ghiasi[1] Mohammad Sadrosadati[1] Harun Mustafa[1] Arvid Gollwitzer[1]
Can Firtina[1] Julien Eudine[1] Haiyu Mao[1] Joël Lindegger[1] Meryem Banu Cavlak[1]
Mohammed Alser[1] Jisung Park[2] Onur Mutlu[1]
[1]ETH Zürich [2]POSTECH

https://arxiv.org/abs/2406.19113

Can Enable Better Error Handling in SSDs



INVITED PAPER

Proceedings of the IEEE, Sept. 2017

Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

This paper reviews the most recent advances in solid-state drive (SSD) error characterization, mitigation, and data recovery techniques to improve both SSD's reliability and lifetime.

By Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu

https://arxiv.org/pdf/1706.08642

Venice

Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses

Rakesh Nadig*, Mohammad Sadrosadati*, Haiyu Mao,
Nika Mansouri Ghiasi, Arash Tavakkol, Jisung Park,
Hamid Sarbazi-Azad, Juan Gómez Luna, and Onur Mutlu

https://arxiv.org/abs/2305.07768

SAFARI Research Group
safari.ethz.ch

Venice

Improving Solid-State Drive Parallelism at Low Cost via Conflict-Free Accesses

Onur Mutlu

omutlu@gmail.com

https://people.inf.ethz.ch/omutlu

7 August 2024

FMS: the Future of Memory and Storage

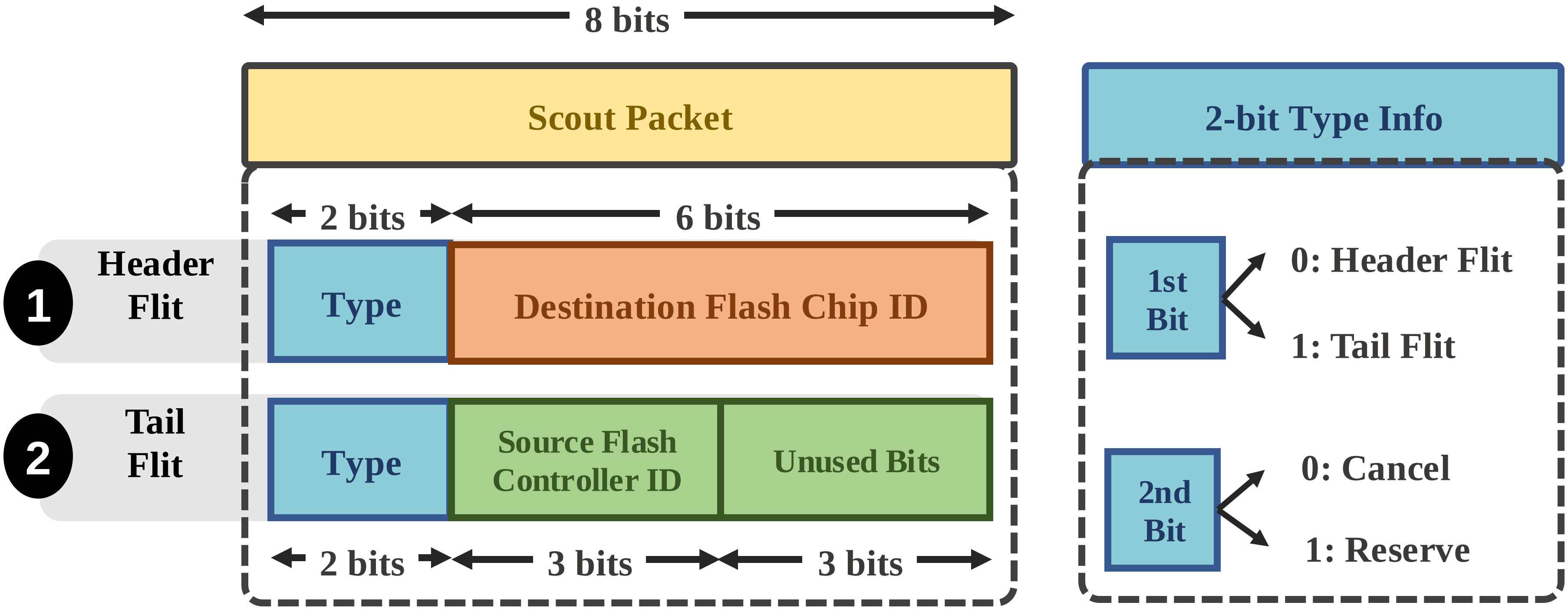
Backup Slides

FAQ

- Scout Packet Structure
- Power Overhead
- Area Overhead
- Evaluated Configurations
- Workload Characteristics
- Mixed Workloads
- Results
  - Throughput Analysis
  - Tail Latency
  - Power Consumption
  - Performance on Mixed Workloads
  - Sensitivity to Interconnection Network Configuration

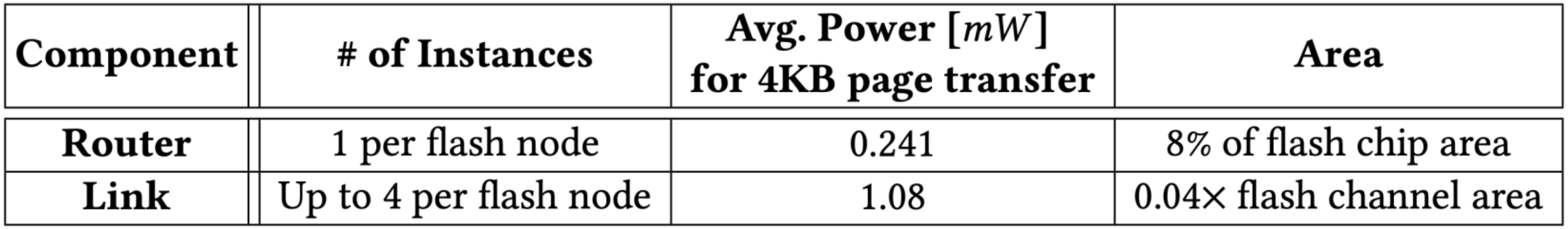
Structure of a Scout Packet

- A *scout packet* consists of two 8 bit flits, a header flit and a tail flit
- The flash controller sends a scout packet to identify a conflict-free path for the I/O request



Power Consumption (I)

- **Router**
  - We implement the HDL and synthesize it using UMC 65nm technology node
  - Router consumes 0.241mW for a 4KB page transfer
- **Network Link**
  - ORION 3.0 power model tool
  - Each network link consumes about 1.08mW for a 4KB page transfer
  - **Link capacitance is lower than bus capacitance** -> 90% less power than that of the shared channel bus
    - Links are shorter and thinner than a shared bus
    - Two drivers in links compared to several drivers in a bus



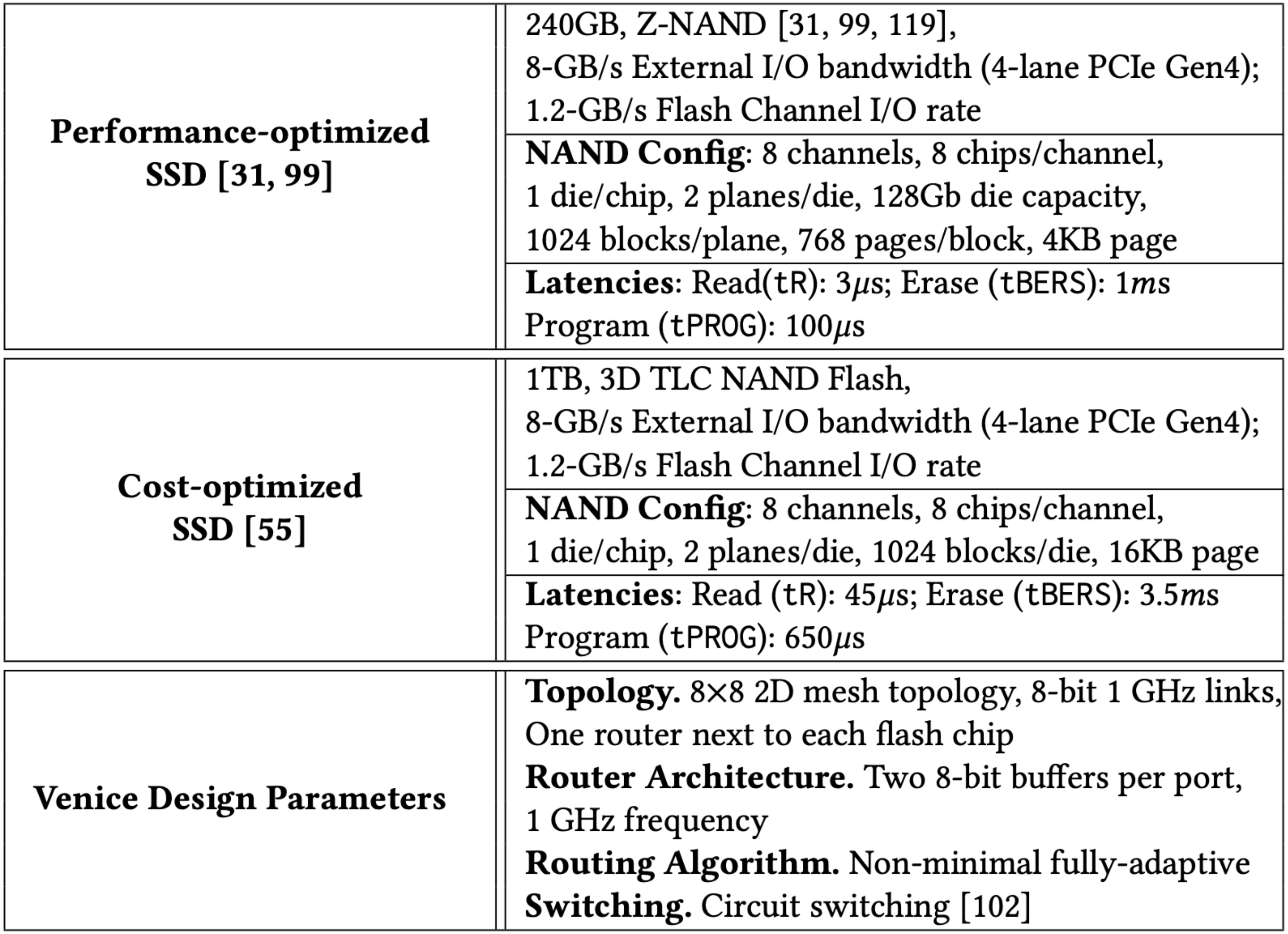
Component	**# of Instances**	**Avg. Power [mW] for 4KB page transfer**	**Area**
Router	1 per flash node	0.241	8% of flash chip area
Link	Up to 4 per flash node	1.08	0.04× flash channel area

Area Overhead

- Router
  - Area overhead estimated using router's HDL model
  - Each router has
    - an area of 614 μm^2 + 40 I/O
    - A total area of 8mm^2 -> 8% of a typical 100mm^2 flash chip

- Network Link
  - ORION 3.0 model for area analysis of network links
  - 112 network links for a 8x8 flash array configuration
  - 44% lower area than a baseline multi-channel shared bus architecture
  - **Links are thinner and require lower pitch sizes**

Evaluated Configurations

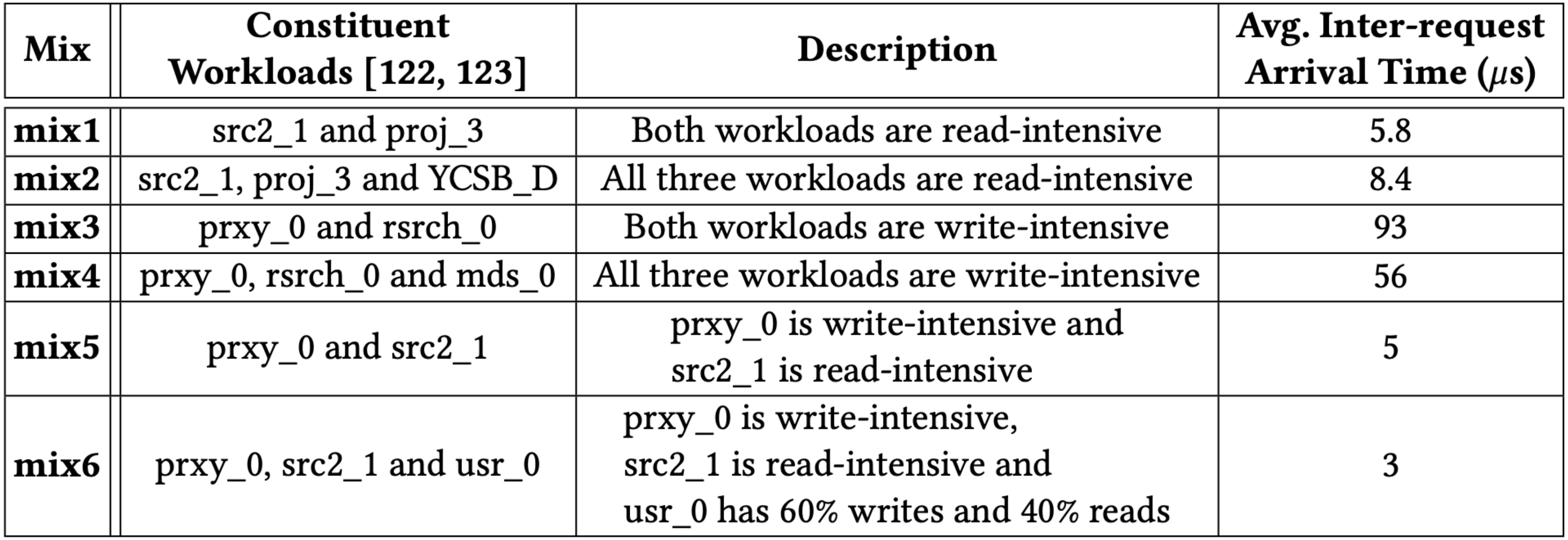


Performance-optimized SSD [31, 99]	240GB, Z-NAND [31, 99, 119], 8-GB/s External I/O bandwidth (4-lane PCIe Gen4); 1.2-GB/s Flash Channel I/O rate
	NAND Config: 8 channels, 8 chips/channel, 1 die/chip, 2 planes/die, 128Gb die capacity, 1024 blocks/plane, 768 pages/block, 4KB page
	Latencies: Read(tR): 3μs; Erase (tBERS): 1ms Program (tPROG): 100μs
Cost-optimized SSD [55]	1TB, 3D TLC NAND Flash, 8-GB/s External I/O bandwidth (4-lane PCIe Gen4); 1.2-GB/s Flash Channel I/O rate
	NAND Config: 8 channels, 8 chips/channel, 1 die/chip, 2 planes/die, 1024 blocks/die, 16KB page
	Latencies: Read (tR): 45μs; Erase (tBERS): 3.5ms Program (tPROG): 650μs
Venice Design Parameters	**Topology.** 8×8 2D mesh topology, 8-bit 1 GHz links, One router next to each flash chip **Router Architecture.** Two 8-bit buffers per port, 1 GHz frequency **Routing Algorithm.** Non-minimal fully-adaptive **Switching.** Circuit switching [102]

Workload Characteristics

	Traces	Read %	Avg. Request Size (KB)	Avg. Inter-request Arrival Time (μs)
MSR Cambridge [122]	hm_0	36	8.8	58
	mds_0	12	9.6	268
	proj_3	95	9.6	19
	prxy_0	3	7.2	242
	rsrch_0	9	9.6	129
	src1_0	56	43.2	49
	src2_1	98	59.2	50
	usr_0	40	22.8	98
	wdev_0	20	9.2	162
	web_1	54	29.6	67
YCSB [123]	YCSB_B	99	65.7	13
	YCSB_D	99	62	14
Slacker [124]	jenkins	94	33.4	615
	postgres	82	13.3	382
SYSTOR '17 [125]	LUN0	76	20.4	218
	LUN2	73	16	320
	LUN3	7	7.7	3127
YCSB RocksDB [126]	ssd-00	91	90	5
	ssd-10	99	11.5	2

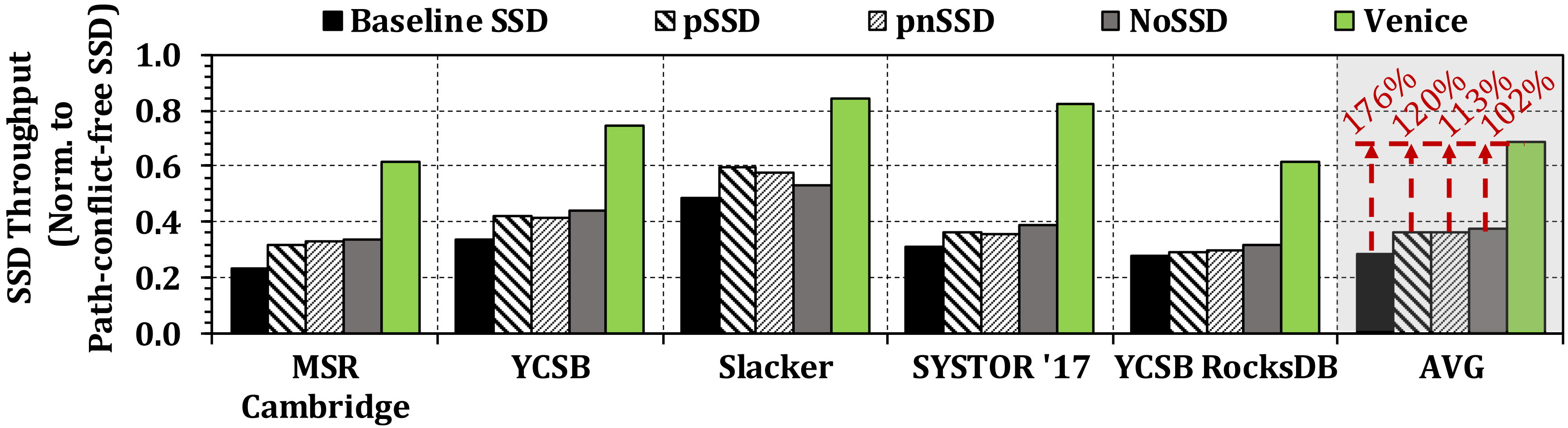
Mixed Workloads



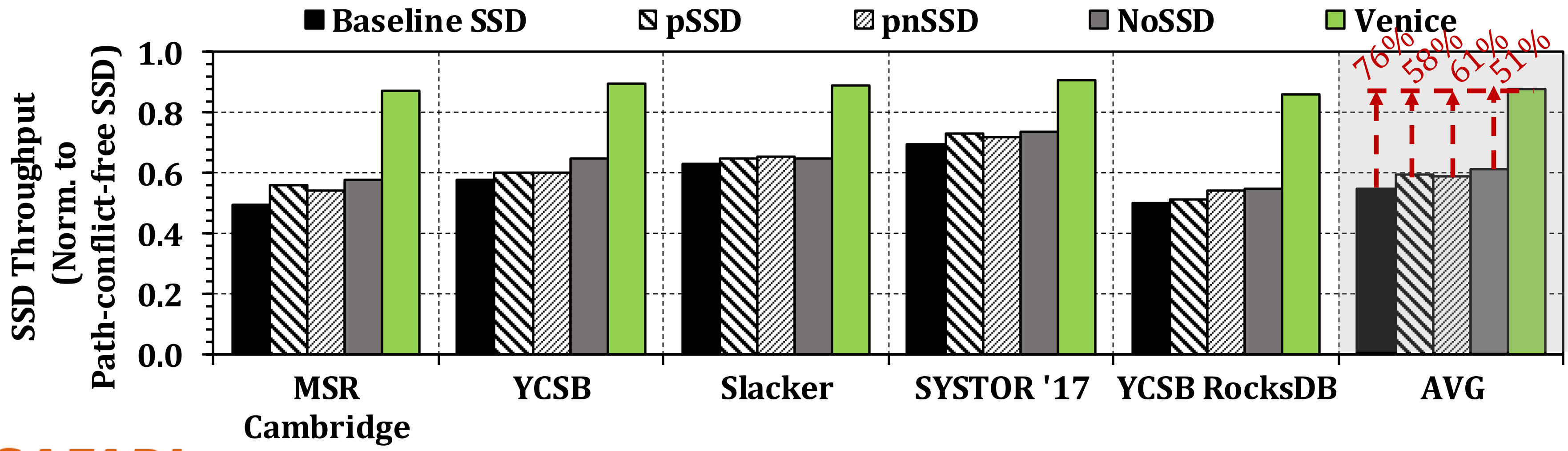
Mix	Constituent Workloads [122, 123]	Description	Avg. Inter-request Arrival Time (μs)
mix1	src2_1 and proj_3	Both workloads are read-intensive	5.8
mix2	src2_1, proj_3 and YCSB_D	All three workloads are read-intensive	8.4
mix3	prxy_0 and rsrch_0	Both workloads are write-intensive	93
mix4	prxy_0, rsrch_0 and mds_0	All three workloads are write-intensive	56
mix5	prxy_0 and src2_1	prxy_0 is write-intensive and src2_1 is read-intensive	5
mix6	prxy_0, src2_1 and usr_0	prxy_0 is write-intensive, src2_1 is read-intensive and usr_0 has 60% writes and 40% reads	3

SSD Throughput Analysis

- Performance-Optimized SSD

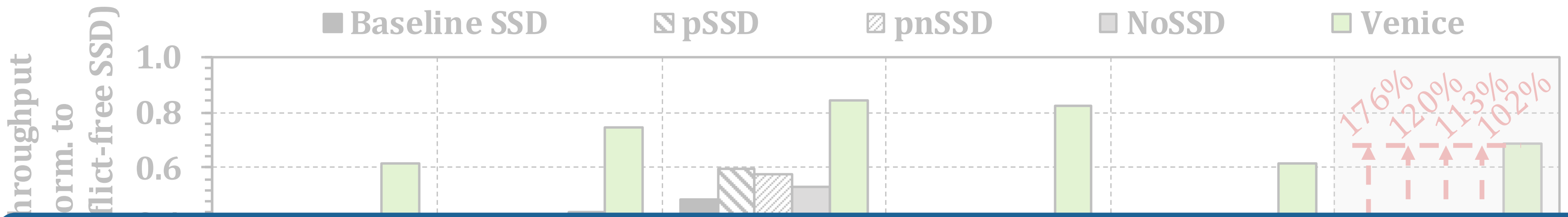


- Cost-Optimized SSD

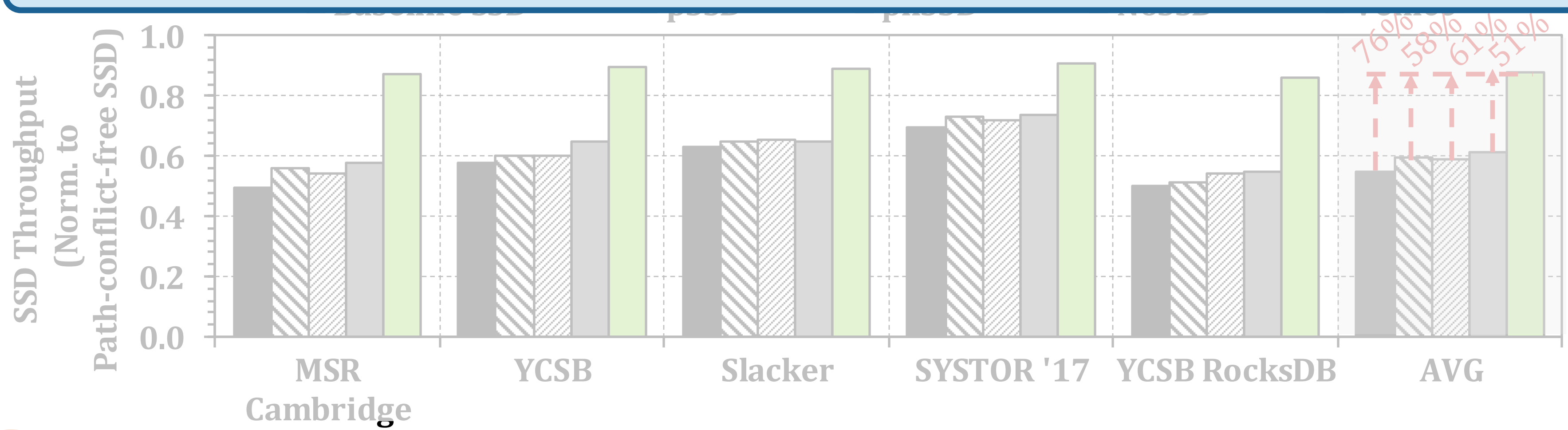


SSD Throughput Analysis

- Performance-Optimized SSD

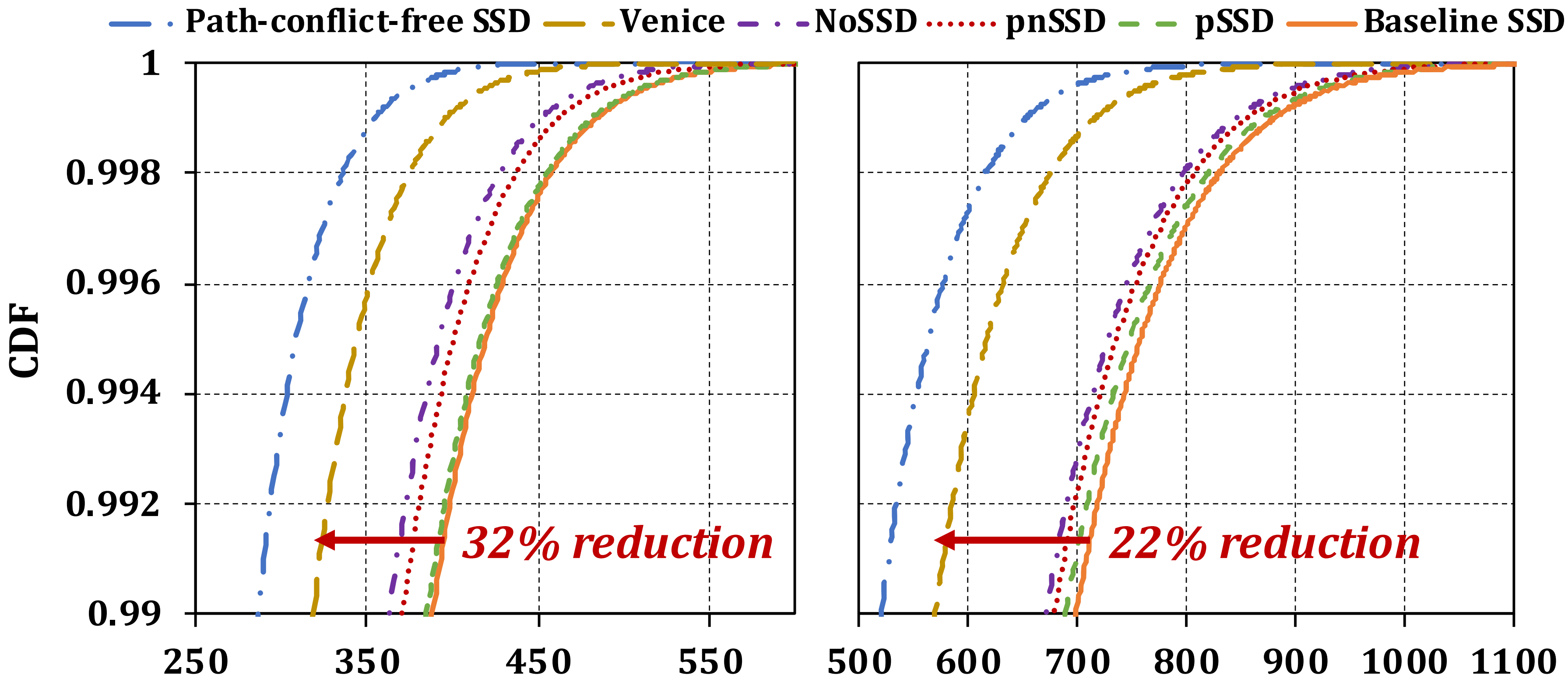


Venice improves SSD throughput over prior approaches by effectively mitigating path conflicts



Tail Latency

- Comparison of tail latencies in the 99th percentile of I/O requests



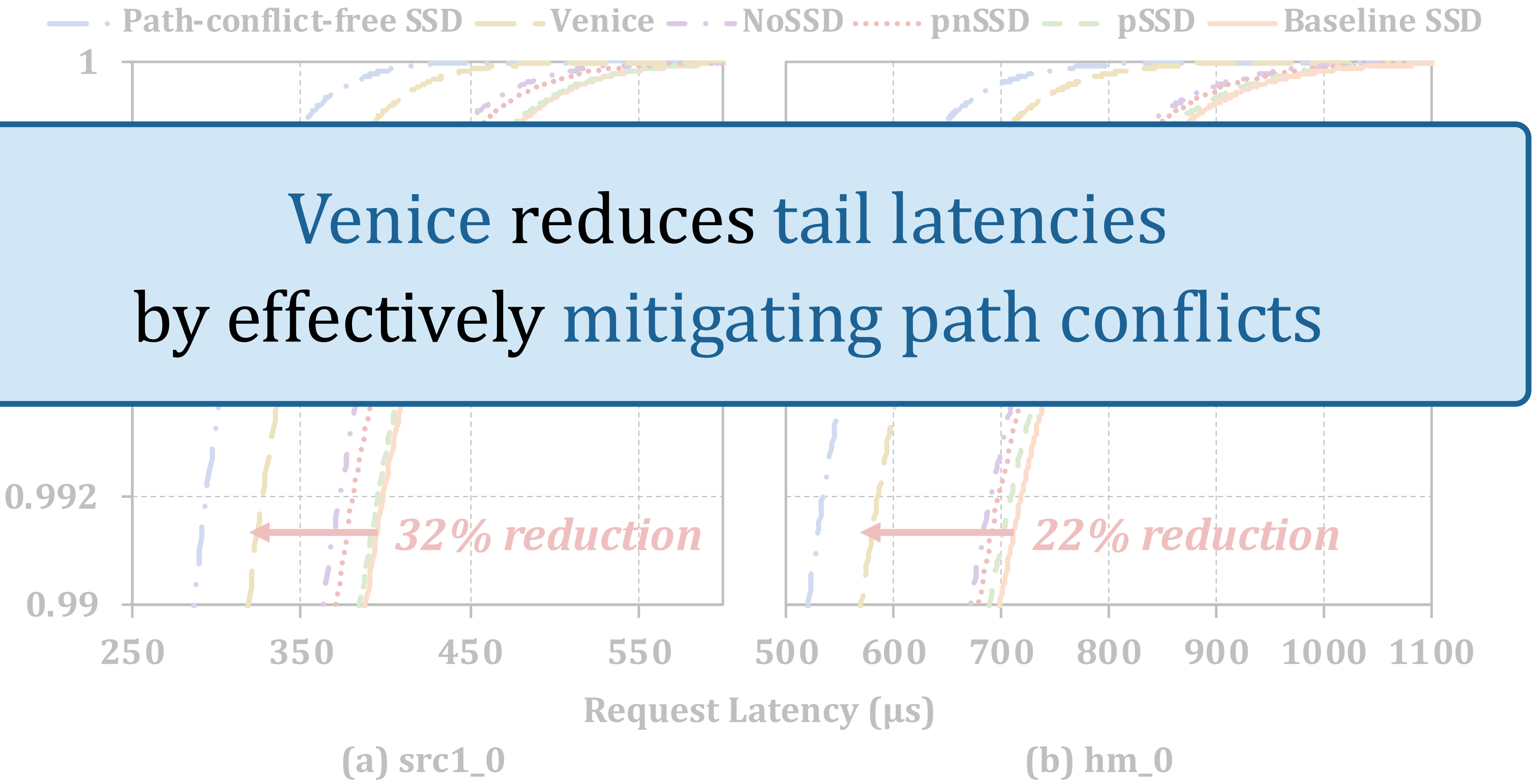
(a) src1_0

(b) hm_0

SAFARI

Tail Latency

- Comparison of tail latencies in the 99th percentile of I/O requests

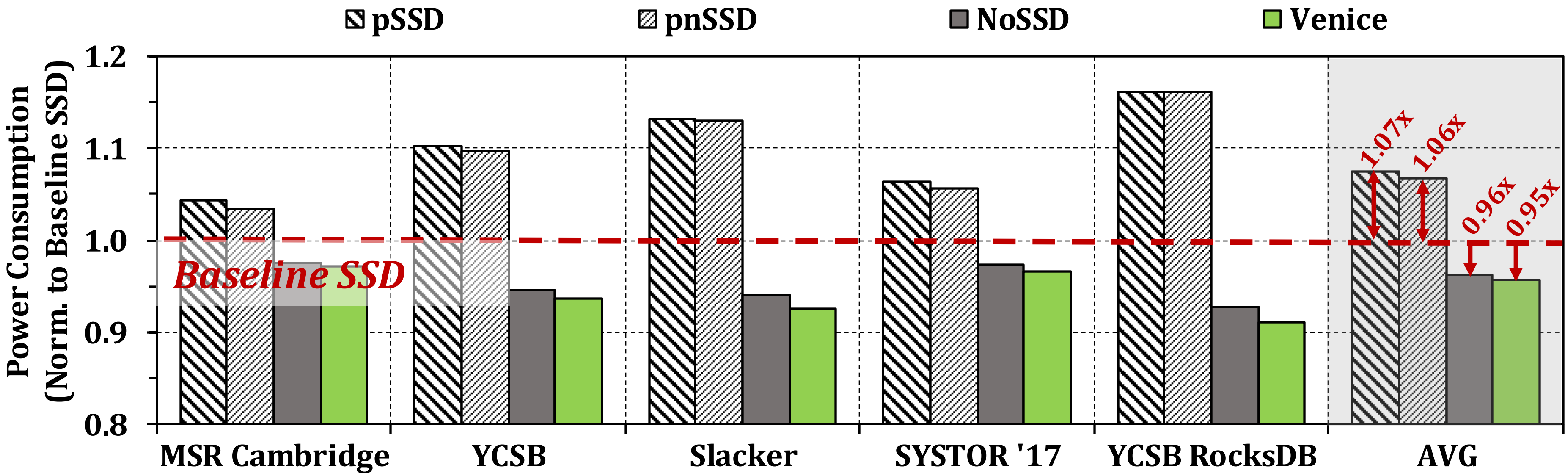


(a) src1_0

(b) hm_0

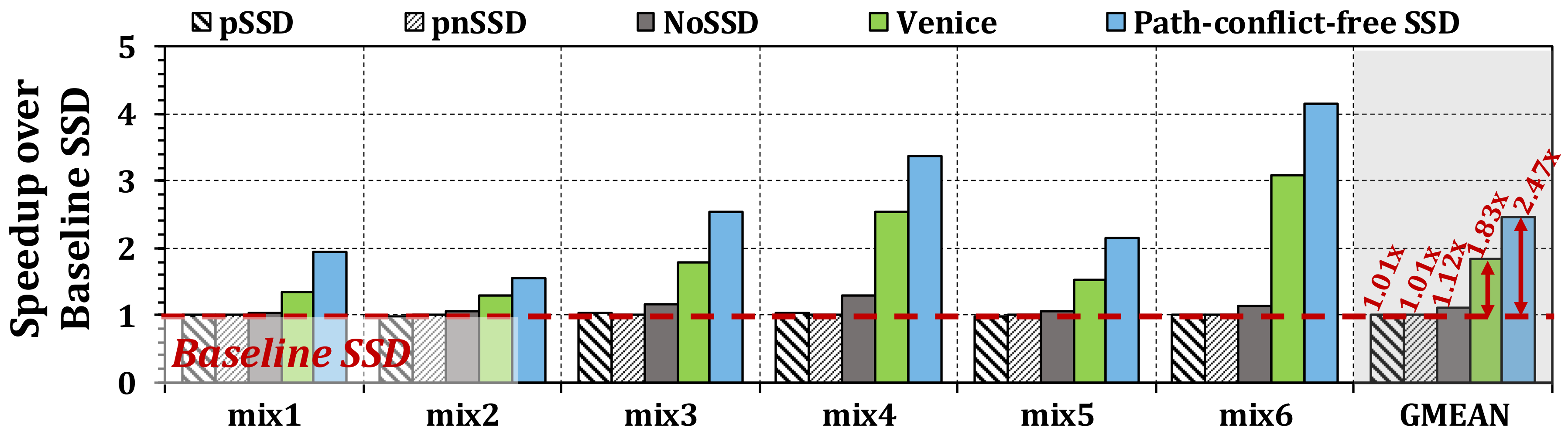
Venice reduces tail latencies
by effectively mitigating path conflicts

Power Consumption (II)



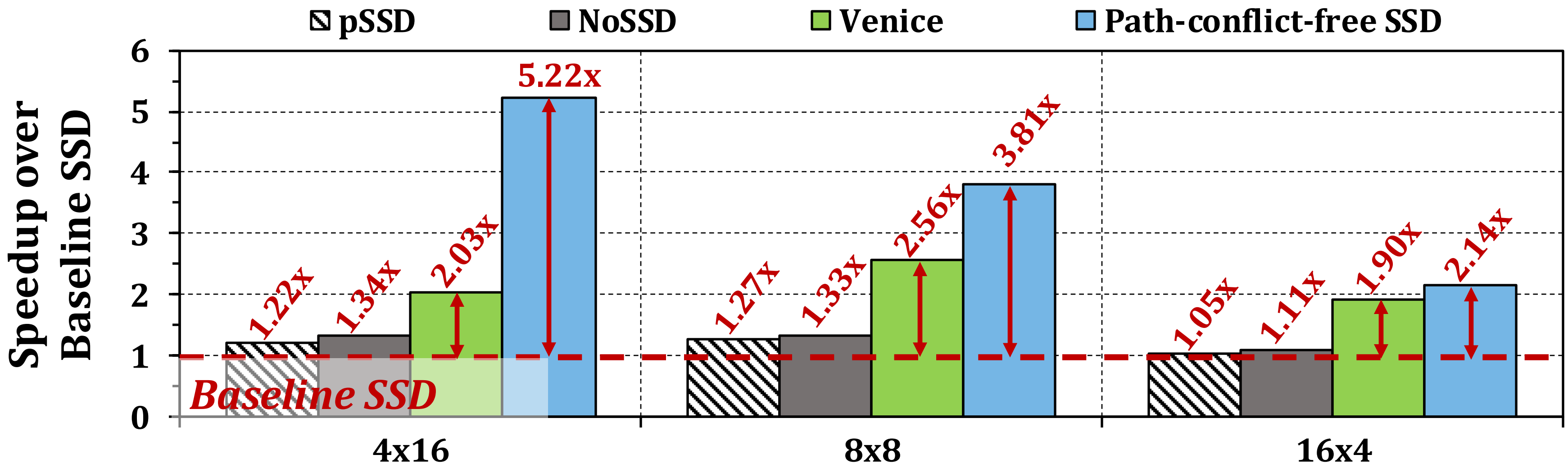
Venice reduces the average power consumption by 4% over Baseline SSD

Performance on Mixed Workloads



Venice outperforms prior approaches on high-intensity mixed workloads by effectively mitigating path conflicts

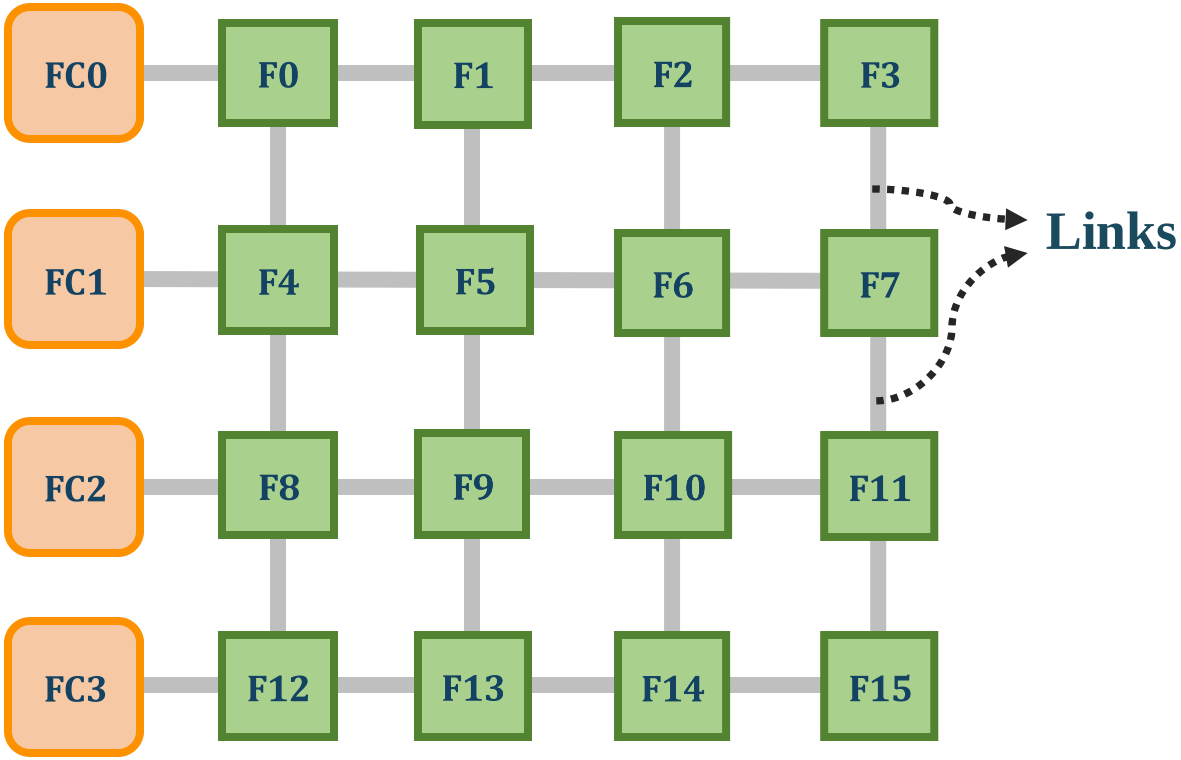
Sensitivity to Network Configurations



Venice provides higher performance improvement for 8x8 compared to 4x16 and 16x4

Prior Approaches to Address Path Conflicts

- **Network-On-SSD [2]**
  - Replaces a multi-channel shared bus architecture with an interconnection network of flash chips
  - Significantly increases path diversity than a typical SSD

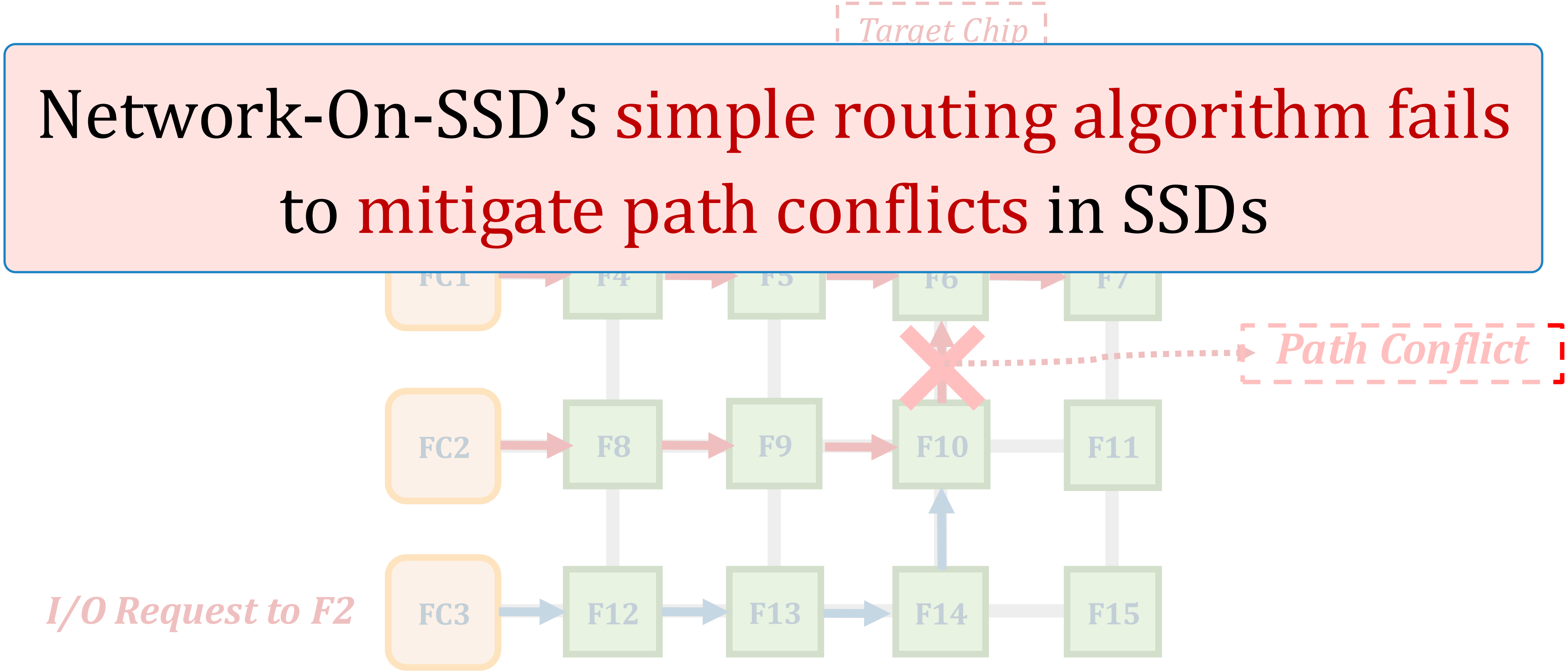


[2] Tavakkol+, “Network-on-SSD: A Scalable and High-Performance Communication Design Paradigm for SSDs”, IEEE CAL 2012

Prior Approaches to Address Path Conflicts

- **Network-On-SSD [2]**
  - Replaces a multi-channel shared bus architecture with an interconnection network of flash chips
  - Significantly increases path diversity than a typical SSD

Network-On-SSD's simple routing algorithm fails to mitigate path conflicts in SSDs



[2] Tavakkol+, "Network-on-SSD: A Scalable and High-Performance Communication Design Paradigm for SSDs", IEEE CAL 2012