#### KVAccel: A Novel Write Accelerator for LSM-Tree-Based KV Stores with Host-SSD Collaboration

<u>Kihwan Kim</u><sup>1</sup>, <u>Hyunsun Chung</u><sup>1</sup>, Seonghoon Ahn<sup>1</sup>, Junhyeok Park<sup>1</sup>, Safdar Jamil<sup>1</sup>, Hongsu Byun<sup>1</sup>, Myungcheol Lee<sup>2</sup>, Jinchun Choi<sup>2</sup>, Youngjae Kim<sup>1</sup>







The 39th IEEE International Parallel and Distributed Processing Symposium (IPDPS), Milan, Italy, June 3-7, 2025

Background	Motivation	Design	Evaluation	Conclusion
Conten	ts			

- Background
- Motivation
- Design
- Evaluation
- Conclusion

DISCOS

Background	Motivation	Design	Evaluation	Conclusion

## Background

Background	Motivation	Design	Evaluation	Conclusion
LSM-tre	e based K	<b>Key-Value</b> S	Stores (LSM	-KVS) 🎵



- Log-Structured Merge-Tree(LSM-tree)
  - Designed for write-intensive workloads
  - Optimized for large-scale data
  - Out-of-place updates
  - Sequential batch operations







RocksDB [1]

[1]: Facebook, "RocksDB" https://rocksdb.org, 2012

[2]: Google, "LevelDB" https://github.com/google/leveldb, 2017

[3]: Meta, "ZippyDB" https://engineering.fb.com/2021/08/06/core-infra/zippydb/, 2021

Background	Motivation	Design	Evaluation	Conclusion
LSM-tre	e based K	Key-Value S	Stores (LSM	-KVS)

- LSM KVS(e.g. RocksDB) stores data in an append-only manner in the active MemTable
- Data in MemTable is moved to and managed on disk through background jobs(Flush, Compaction)



Fig. 1: An architecture of LSM-tree.

Background	Motivation	Design	Evaluation	Conclusion
Write S	tall Proble	m		

- Write Stall: write operation blocked, due to bottlenecks in Flush, Compaction
- In RocksDB, Write stall occurs under these 3 scenarios<sub>[4][5]</sub>
  - Incoming Writes > Flush
  - Flush > Level 0 to Level 1 Compaction
  - Pending deep level compaction size becomes heavier

[4]: SILK: Preventing Latency Spikes in Log-Structured Merge Key-Value Stores, Oana Balmau et al., USENIX ATC'19

[5]: ADOC: Automatically Harmonizing Dataflow Between Components in Log-Structured Key-Value Stores for Improved Performance, Jinghuan Yu et al. (USENIX FAST'23)

Background	Motivation	Design	Evaluation	Conclusion
Existing	g Work: Al	<b>DOC</b> <sub>[5]</sub>		

- In three types of overflow scenarios, ADOC alleviates write stalls by adjusting two tuning knobs
- Two tuning knobs: # of Compaction threads, MemTable size

	# of Compaction Threads	MemTable Size
Incoming Writes > Flush	Ļ	ſ
Flush > Level 0 to Level 1 Compaction	1	
Pending deep level compaction size becomes heavier	1	↓

Background	Motivation	Design	Evaluation	Conclusion
Existing	g Work: Al	<b>DOC</b> <sub>[5]</sub>		
<ul> <li>In three ty adjusting</li> </ul>	/pes of overflow two tuning knob	scenarios, ADO s	C alleviates write	stalls by

Two tuning knober # of Composition throads MamTable size

1. Not an immediate remedy  $\rightarrow$  Write stalls still occur

#### 2. Tuning knobs does not stop write slowdowns from occurring.

Pending deep level compaction size becomes heavier



Background	Motivation	Design	Evaluation	Conclusion	
				DISCOS	

## **Motivation**

Background	Motivation	Design	Evaluation	Conclusion
Observatio Slowdow	on 1. Ins <sub>161</sub> : The Ine	efficient Write	e Stall Soluti	ion Discos 🖉

- RocksDB uses the *slowdown*<sub>[6]</sub> method to prevent user writes from becoming completely blocked
- The state of the art solution ADOC<sub>[5]</sub> also uses *slowdowns*

Both RocksDB and ADOC<sub>[5]</sub> ultimately fall back to using *slowdown* to avoid a write stall

<sup>[5]:</sup> ADOC: Automatically Harmonizing Dataflow Between Components in Log-Structured Key-Value Stores for Improved Performance, Jinghuan Yu et al. (USENIX FAST'23) [6]: https://github.com/facebook/rocksdb/wiki/Write-Stalls

Background	Motivation	Design	Evaluation	Conclusion
Observatio	on 1.			

Slowdowns<sub>[6]</sub>: The Inefficient Write Stall Solution

 Slowdowns, while preventing a complete write stall from occurring, harms overall performance







Background	Motivation	Design	Evaluation	Conclusion
Observatio Under-u	n 2. Itilization of	PCIe Band	dwidth	Discos

• PCIe Traffic drop sharply during a write stall, implying inefficient device resource usage



Background	Motivation	Design	Evaluation	Conclusion
Observatio	on 2.			

## Under-utilization of PCIe Bandwidth



- PCIe Traffic drop sharply during a write stall, implying inefficient device resource usage
  - RocksDB is shown to leave up to 90% of available PCIe bandwidth around 50% of the time during a write stall







VS

Slowdowns

- Maintains I/O service at all times
- Overall throughput and latency penalty due to said slowdowns

- Allowing Write Stalls
- Overall throughput and latency conserved
- Complete interrupts in I/O service as write stalls are allowed to occur
- Observation 2. reveals an unexploited resource to help mitigate write stalls and increase performance without sacrificing system resources: underutilized PCIe and device bandwidth during write stalls



underutilized PCIe and device bandwidth during write stalls.

Background	Motivation	Design	Evaluation	Conclusion
				DISCOS

## **Proposed Solution:** *KVAccel*

Background	Motivation	Design	Evaluation	Conclusion
Pronose	d Solution	KVAccel		

 KVAccel's design is based on two key factors: Disaggregation and Aggregation

#### Disaggregation

- Division of SSD into hybrid interface (block and keyvalue) and its required I/O paths
- Maintenance of each interface's separate LSM-Tree

### Aggregation

- Manage data from each interface as if it was one database instance
- Unify separate I/O commands and database state with rollback

Background	Motivation	Design	Evaluation	Conclusion
Overvie	w of <b>KVAc</b>	cel		

- Co-Design of Hardware & Software provides 2 I/O paths
- Different I/O paths taken based on the presence of a write stall



Background	Motivation	Design	Evaluation	Conclusion
Overvie	w of <b>KVAc</b>	cel		

- Co-Design of Hardware & Software provides 2 I/O paths
- Different I/O paths taken based on the presence of a write stall



Background	Motivation	Design	Evaluation	Conclusion
Overvie	w of <b>KVAc</b>	cel		

- Co-Design of Hardware & Software provides 2 I/O paths
- Different I/O paths taken based on the presence of a write stall



Background	Motivation	Design	Evaluation	Conclusion
Hybrid [	Dual-Interfa	ce SSD		

- Hybrid interface SSD achieved by logical NAND flash address disaggregation via a specified address boundary
  - $\circ~$  SSD issues different commands for each interface



DISCOS

Background	Motivation	Design	Evaluation	Conclusion
Softwar	e Modules(	1)		



#### Detector

- Detects write stalls checking 3 components
  - # of Level 0 SSTs
  - Memtable size
  - Pending compaction size

#### • Controller

 Directs I/O commands to the correct interface based on the Detector's output.

DISCO

Background	Motivation	Design	Evaluation	Conclusion
Softwar	e Modules(	2)		



#### • Metadata Manager

- Keeps track of KV pairs located in Dev-LSM via a hash table for membership testing
- Rollback Manager
  - Initiates and performs the rollback operation based on the rollback scheduling policy and the Detector's output

Background	Motivation	Design	Evaluation	Conclusion
Rollhack	Concration	. Schedulir	าต	

## Rollback Operation: Scheduling



- Rollback refers to return the KV pairs in Dev-LSM back to Main-LSM into one LSM-KVS instance
- Rollback operation can be scheduled *eagerly* or *lazily* based on workload characteristics

#### Eager Rollback

- Perform rollback as soon as there are enough resources available (by using L<sub>0</sub> file count threshold)
- Ideal for a read orientated workload to avoid slow Dev-LSM read operations

#### Lazy Rollback

- Delay rollback until the current write workload is completely finished
- Ideal for a write intensive workload to lower interference of rollback with write operations

Background	Motivation	Design	Evaluation	Conclusion
Rollbacl	<pre>&lt; Operatior</pre>	1		

- To accelerate rollback, KV pairs are read in bulk using a range scan operation
- Iterator reads Dev-LSM in its entirety and serializes the KV pairs
- KV pairs are then sent to the host by performing DMA multiple times



DISCOS

Background	Motivation	Design	Evaluation	Conclusion
				DISCOS

## **Evaluation**

Background	Motivation	Design	Evaluation	Conclusion

# Evaluation Setup Testbed:

KV-SSD on Cosmos+ OpenSSD Platform<sub>[7]</sub>





SoC	Xilinx Zynq-7000 with ARM Cortex-A9 Core
NAND Module	1TB, 4 Channel & 8 Way
Interconnect	PCIe Gen2 ×8 End-Points

TABLE II: Specifications of the host system.

CPU	Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz (32 cores), CPU usage limited to 8 cores.
Memory	384GB DDR4
OS	Ubuntu 22.04.4, Linux Kernel 6.6.31

[7]: Cosmos+ OpenSSD Platform: http://www.openssd-project.org/platforms/cosmospl/

DISCOS

Background	Motivation	Design	Evaluation	Conclusion
LSM-K∖	/S and Ben	chmark Co	onfiguration	S 🎵 🕅

TABLE III: LSM-KVS configurations. For all figures, the numbers next to each LSM-KVS refer to compaction thread count. For KVACCEL, the settings refer to the Main-LSM.

LSM-KVS	Compaction Threads $(n)$	MT Size
	1	
KVACCEL(n)	2	]
	4	1
	1	1
RocksDB(n)	2	128 MB
	4	1
	1	1
ADOC(n)	2	]
	4	

TABLE IV: *db\_bench*<sup>8</sup> workload configurations. Each benchmark was run with a 4 B key and 4 KB value size. Workload A,B,C were run for 600 seconds, and Workload D performed 60K read operations.

Name	Туре	Characteristics	Notes (write/read ratio)
A	fillrandom	1 write thread	No write limit
В	readwhilewriting	1 write thread	9:1
C	readwintewriting	+ 1 read thread	8:2
D	seekrandom	1 range query thread	Run after initial
	seekrandom	(Seek + 1024 Next)	20GB fillrandom

DISCOS

Background	Motivation	Design	Evaluation	Conclusion
Write St	tall Avoidan	ice		

 Throughput minimum values greatly increased, as *KVAccel* is designed to allow as much throughput as the SSD and system allows without slowdowns



Background	Motivation	Design	Evaluation	Conclusion
Perform	ance Evalu	uation		

• (a) Throughput, (b) P99 Latency, (c) Efficiency





10

Thread 1

Thread 2 Thread 4

(b)

Thread 1

P99

Thread 2 Thread 4

(a)

Thread 1

Thread 2 Thread 4

(c)











Background	Motivation	Design	Evaluation	Conclusion
				Discos

## Conclusion

Background	Motivation	Design	Evaluation	Conclusion
Conclus	sion			

- Prior work addresses write stalls to a limited extent
   Hardware and software are treated in isolation
- *KVAccel* achieved a 17% improvement in throughput and a 20% reduction in latency compared to ADOC.
- **KVAccel** demonstrates the effectiveness of hardware-software codesign
  - Alleviates write stalls by utilizing:
    - Under-used PCIe bandwidth
    - Computational capabilities within SSDs

Background	Problem Definition	Motivation	Design	Evaluation
T				



Conclusion

## Thank you!

- Contact
  - Kihwan Kim / lewis461@sogang.ac.kr
  - Hyunsun Chung / hchung1652@sogang.ac.kr
  - Seonghoon Ahn / ok10p@sogang.ac.kr
  - Data-Intensive Computing & AI Systems ٠

Laboratory https://discos.sogang.ac.kr/



<Camera-ready paper> Can be found on Google Scholar

> KVACCEL: A Novel Write Accelerator for LSM-Tree-Based KV Stores with Host-SSD Collaboration

> > Kilmus Kim14, Hyunus Chang14, Sronghous Alus14, Jushyook Pach1, Saliar Junif Hongra Byer<sup>1</sup>, Myongcherd Lee<sup>1</sup>, Znichen Chet<sup>1</sup>, Youngjae Kim<sup>1,1</sup> Tour, of Computer Science and Engineering, Seging University, Second, Republic of Konra WIRL Dates, Republy of Kores

whe initiality for a why when her performants digitality for a why when her induities. For induities, why or equipping while the organization for addition. Abcounterful, hardware based whether here is a wKhab, bob downstrate comparatio performance. Judio Down-Key Yuliar Hare, Lap Neucland Merge Teo, field Nam Extra, Webs Hall Mitigation

#### In Instantonic Vision

Log-Structured Margar (LEM) mon-based Key-Value Struct (KVS) icinities, such as Ricks/08 [7] and LevelD8 [7], and commonly used in write-intensive applications due to their ability to handle high-throughput writes efficiently. However, LNM based KVIh (LIM KVIn) often experience performance dependence due to write stalls that some during comparison [15-[35]. These write staffs black meaning artir operations, resulting is a significant enducion in throughput and as increase in tell latency, which andormous system reliability in morunation workloads.

To alleriant series stalls, many software based solutions have horn replaced and deployed. Buckel/8 [1], one of the most wishty used LIM-KVS, suplements a mechanism known as structures [75]. This should not standard an anticipates presented write stafts and presentively reduces the write pressure on the LSM KVS. White cheedeway can percent write stads, it may assessmently decrease the throughput of Rock-DB by Loning the write pressure directed to the LOM KVE. Additionally, the state of the art solution ADOC [1] estigates write staffs by dynamically increasing back sizes and die norder of

"Plot an lite or attlent and have constituted spinits. T. Etc. of the contraposition without

shinest-day linestend Marge (LSM: involued Key Nater comparison-fictuals during a solar sizedness, florefry tollacing more AVia an addit adopted for their high performance in compaction deputies. Memore ADEX increases load CPL Alternatively, hardware based solutions have been invest Both an opposing to under to straig multiple comparison finals, can some second darge is through an intervent built (PT, and PH), and a strategy approaches using PPLA, built revision in PDI before finaling than to be LSM-text. additional horizon cash, is the indig to proper EVECUS, or sent landscare database undergo intervents that Experiments from angle its loweraging a distribution from 50th EVECUS, advances links Excerning on the indigeneous biolic Statistics, advances indigeneous advances and advances and advances and advances and advances intervention of the indigeneous advances and advances and advances intervention of the indigeneous advances and advances and advances and advances advances and advances and advances advances and advances adva makers with an invarie many many many methanism. Our whinding sary performance degradations due to maccurate perdictance or aparhenas ABOC in ap is 17% in torus of throughput and increased host CPU usage, while hardware adultions require efformance to CPU utilization efformers. For extend read worth additioned hardware remond costs, In this study, not propose a grandwisking approach that provide actor malls without compromoting KVS performance, minimizes how CPU utilization, and requires an additional hardware costs. One method suprants a see paradign that is fundamentally different from entering approaches, by actively inversiging affer resolution in stating merage devices to avoid write such while minimizing And CPU involvement.

admare proletige framework that immeges a new ibalinterface KHD architecture to millipate active stalls and optimize the self-sation of storage bandwidth. &'CACCO, is built on the observation that during bost-cale withit malls, the andorlying staniale Atolice's analytic MD blandscidth semaints underset/less despite to prevental to handle additional UO operations. EVA:: (11) then incorporates a dynamic 10 rollination mechanics that measures the status of host-outs LEM-RVS and, space shreeting a write stall, shifts writen littee the LSM AVS to the deside outs has aske write hother

KONCERL provents a disappropriate of the SMP's logical NAND flash address space toto year regimes year for the traditional block interface, which is managed by the loss state LIM KVL and another for the late raise interface impired by lie KY-58D, which serves as a temporary write hafter to serve pending sinis mapping in braining the multilinual LUM based sheni mish shering malle.

To maintain consistency between the main LOM on the hand and the write haffer on the dovice, KNISTERS introduces

In his paper, we present EVD/VIU, a second laybrid family are