19th USENIX Conference on File and Storage Technologies (FAST '21)

Behemoth: A Flash-centric Training Accelerator for Extreme-scale DNNs

Shine Kim^{1,2,*} Yunho Jin^{1,*} Gina Sohn¹ Jonghyun Bae¹ Tae Jun Ham¹ Jae W. Lee¹



¹ Seoul National University

SAMSUNG

² Samsung Electronics

*Equal Contributions

Explosive expansion of DNNs

- Deep Neural Networks have become widespread in various application domains
 - Natural language processing, computer vision, recommendation, and so on
- Increasing the model size is crucial to improve accuracy of DNNs
 - Extreme-scale models demand a tremendous amount of computation and memory capacity



DNN training is a repetitive process of matrix operation



- DNN training is a repetitive process of matrix operation
 - Forward path: multiply activation and weights to generate expected value
 - Calculate the difference (loss) between expected value and ground truth
 - Backward path: propagate the loss in backward order and update weights

A: Activation W: Weight G: Gradient



- DNN training is a repetitive process of matrix operation
 - Forward path: multiply activations and weights to generate expected value
 - Calculate the difference (loss) between expected value and ground truth
 - Backward path: propagate the loss in backward order and update weights



- DNN training is a repetitive process of matrix operation
 - Forward path: multiply activations and weights to generate expected value
 - Calculate the difference (loss) between expected value and ground truth
 - Backward path: propagate the loss in backward order and update weights



- DNN training is a repetitive process of matrix operation
 - Forward path: multiply activation and weights to generate expected value
 - Calculate the difference (loss) between expected value and ground truth
 - Backward path: propagate the loss in backward order and update weights



- Memory capacity wall
 - DNN model size exceeds memory capacity of a single GPU
 - Forces users to partition the model and distribute to HBM DRAM on GPU (Model Parallelism)



*figure borrowed from http://jalammar.github.io/how-gpt3-works-visualizations-animations/

- Memory capacity wall
 - DNN model size exceeds memory capacity of a single GPU
 - Forces users to partition the model and distribute to HBM DRAM on GPU (Model Parallelism)



- Memory B/W underutilization
 - As a DNN model (matrix) size increased, each value in the matrix is reused more often
 - The memory B/W requirement does not increase as the computation amount increases



1) Training with batch size of 16 on 840 TFLOPs compute core

- Memory B/W underutilization
 - As a DNN model (matrix) size increased, each value in the matrix is reused more often
 - The memory B/W requirement does not increase as the computation amount increases



50GB/s¹⁾ (required B/W)

Total 28TB/s (HBM B/W)

Significant memory B/W underutilization occurs!

1) Training with batch size of 16 on 840 TFLOPs compute core

Scaling of DNNs necessitates *a new memory system with high-capacity and low-cost* (replacing low-capacity, high-cost HBM)

USENIX FAST'21, Behemoth: A Flash-centric Training Accelerator for Extreme-scale DNNs









• Compute Core: compute tensors and transfer data between Tensor Buffer and NANDs



- Compute Core: compute tensors and transfer data between Tensor Buffer and NANDs
- Tensor Buffer: keep tensors in DDR DRAM serving as a staging (prefetching/offloading) area for NANDs



- Compute Core: compute tensors and transfer data between Tensor Buffer and NANDs
- Tensor Buffer: keep tensors in DDR DRAM serving as a staging (prefetching/offloading) area for NANDs
- NANDs: store tensors like HBMs in conventional training system

Behemoth adopts a **two-level** memory architecture using DDR DRAM and NAND flash to reduce the DNN training cost



- Compute Core: compute tensors and transfer data between Tensor Buffer and NANDs
- Tensor Buffer: keep tensors in DDR DRAM serving as a staging (prefetching/offloading) area for NANDs
- NANDs: store tensors like HBMs in conventional training system





• Activation Node: compute and store activations



- Activation Node: compute and store activations
- Weight Node: update and store weights



- Activation Node: compute and store activations
- Weight Node: update and store weights
- Host system: transfer training command sequence to Behemoth



- Activation Node: compute and store activations
- Weight Node: update and store weights



- Activation Node: compute and store activations
- Weight Node: update and store weights



- Activation Node: compute and store activations
- Weight Node: update and store weights



- Activation Node: compute and store activations
- Weight Node: update and store weights



- Activation Node: compute and store activations
- Weight Node: update and store weights



- Activation Node: compute and store activations
- Weight Node: update and store weights



- Activation Node: compute and store activations
- Weight Node: update and store weights



- Activation Node: compute and store activations
- Weight Node: update and store weights

Flash Memory System (FMS) is the main storage in Behemoth to meet the bandwidth and endurance requirements of extreme scale DNN training



Flash Memory System (FMS) is the main storage in Behemoth to meet the bandwidth and endurance requirements of extreme scale DNN training

>50GB/s Read and Write Bandwidth

5-year Endurance

SSD firmware has become a bottleneck for scalable performance



- SSD firmware has become a bottleneck for scalable performance
- H/W implemented (automated) data-path can be a solution



- SSD firmware has become a bottleneck for scalable performance
- H/W implemented (automated) data-path can be a solution
- Complex functions of FTL make data-path automation difficult
 - Garbage Collection (GC), Wear-leveling (WL), Metadata management for persistency, and so on



FMS separates data types and adopts lightweight FTL to implement H/W automated data path



36

FMS separates data types and adopts lightweight FTL to implement H/W automated data path

#: Stream name			Access po	ermission
(Act. Node / Weight Node)	Persistency	Retention	Host	Behemoth
1: NV-Stream (Training inputs / –)	Non-volatile	Years	Append-only seq. write	Read only
2: V-Stream (Activations / Interm. weights)	Volatile	Minutes	N/A	Read & Append-only seq. write
3: NV-Stream (- / Trained weights)	Non-volatile	Years	Read only	Read & Append-only seq. write

Multi-stream support for data seperation

FMS separates data types and adopts lightweight FTL to implement H/W automated data path

#: Stream name			Access pe	ermission
(Act. Node / Weight Node)	Persistency	Retention	Host	Behemoth
1: NV-Stream (Training inputs / –)	Non-volatile	Years	Append-only seq. write	Read only
2: V-Stream (Activations / Interm. weights)	Volatile	Minutes	N/A	Read & Append-only seq. write
3: NV-Stream (- / Trained weights)	Non-volatile	Years	Read only	Read & Append-only seq. write

Multi-stream support for data seperation



Strict append-only seq. write for lightweight FTL



H/W automated write data path of FMS

(a) write command pipeline: transfers data from TSB to an SRAM buffer in the FMS controller

(b) NAND program pipeline: programs data in the SRAM to NANDs

Improving Endurance of FMS

Endurance of SSD relies on the Program/Erase (P/E) cycles for NAND block



Improving Endurance of FMS

- Endurance of SSD relies on the Program/Erase (P/E) cycles for NAND block
- DNN training workloads cause frequent P/E operation



Behemoth reduces the data retention time and maintains very low WAF (~1)

USENIX FAST'21, Behemoth: A Flash-centric Training Accelerator for Extreme-scale DNNs



42

Behemoth reduces the data retention time and maintains very low WAF (~1)



- Max. data lifespan: 41 sec.
- 1 year retention → 3 days
 - P/E cycle can be increased by at least 40x ^{1, 2)}
 - e.g., 50K P/E cycle \rightarrow 200K P/E cycle

Tensor lifespan for a training iteration of GTP-3 on Behemoth

1) Yu cai et al, ICCD'12, Flash correct-and-refresh: Retention-aware error management for increased flash memory lifetime 2) Ren-Shuo Liu et al, FAST'12, Optimizing NAND flash-based SSDs via retention relaxation



Behemoth reduces the data retention time and maintains very low WAF (~1)

- Only performs monotonic sequential writes and reads
 - No garbage collection \rightarrow WAF 1



Evaluation Methodology

- We evaluate our platform's effectiveness by
 - 1) Comparing the memory cost of Behemoth against the conventional TPU-based DNN training system
 - 2) Comparing the training throughput of FMS against conventional SSDs



Evaluation Methodology

- We evaluate our platform's effectiveness by
 - 1) Comparing the memory cost of Behemoth against the conventional TPU-based DNN training system
 - 2) Comparing the training throughput of FMS against conventional SSDs

gatech edu 🖉	♦ → C a gthub.com/CMU-SAVAR/MOSH		
ARESTRO Here GitHub Docs Resources	Why Globub? - Team Enterprise Explore - Marketplace Pricing -	Search	/ Signin S
REAL PROPERTY AND A VIEW	G CMU-SAFARI / MQSim		⊕menh N ☆St
MAESTRO:	Primantar + Pthranch ©Φtags	Go to file 👲 Code •	About MQSim is a fast and accurate
	areama werge pus request also mon consumater init sature sature teaching teaching teaching teaching teaching teaching teaching teaching	3 uses and	of modern multi-queue (MQ) SSD
an Open-source infrastructure for	and Correct minor typos	9 months ago	MQSim faithfully models new high
Indeling Detefleure within Deen	traces Code refactored	3 years ago	steady-state SSD conditions, and
hodeling Datanows within Deep	D gitignore gitignore	2 years ago	full end-to-end latency of reque modern SSDs. It is described in
corning Accolorators	D LUCENSE First source commit	3 years ago	in the FAST 2018 paper by A
earning Accelerators	D MQSimain Windows complation files	3 years ago	people.int.ethz.ch/omutiu/pub;
이 이 가슴 다는 것은 것을 알고 있어요. 바람들은 것을 것을 가지 않는 것을 가지 않는 것을 했다.	MQSImuceprej Added a separate OOO scheduler that supports NVMe priv	rities 9 months ago	D Readme
그는 것 같아요. 그는 것은 것이 것에서 귀엽을 걸려야 한다. 말했는 것이 많이 많이 많이 많이 했다.	MdSimucoproj.fitters Added a separate OCO scheduler that supports NMMe priv	rties 9 months ago	MIT License
	Modern coprojutor Denuit execution arguments in visual studio	3 years ago	Beleases
건설 경기 사이는 지수는 것을 위해 집에 가지 않는 것이 가지 않는 것이 많다.	D READARE.md Added new option for transaction scheduling	9 months ago	No releases published
승규는 영상에 대한 것을 다 가지 않는 것을 다 주셨다. 것을 다 가지 않는 것을 다 하는 것을 다 나는 것을 수 있다. 말 수 있는 것을 다 나는 것을 수 있다. 말 수 있는 것을 다 나는 것을 수 있다. 말 수 있는 것을 다 나는 것을 다 나는 것을 다 나는 것을 다 나는 것을 것을 다 나는 것을 수 있다. 않는 것을 다 나는 것을 수 있다. 않는 것을 다 나는 것을 다 나는 것을 수 있다. 않는 것을 다 나는 것을 수 있다. 않는 것을 다 나는 것을 수 있다. 않는 것을 다 나는 것을 수 있다. 않는 것을 다 나는 것을 수 있다. 않는 것을 다 나는 것을 수 있다. 않는 것을 다 나는 것을 다 나는 것을 다 나는 것을 수 있다. 않는 것을 다 나는 것을 수 있다. 않는 것을 다 나는 것을 수 있다. 않는 것을 것을 것을 수 있다. 않는 것을 수 있다. 않는 것을 것 같이 않는 것을 수 있다. 않는 것을 것 같이 않는 것을 수 있다. 않는 것을 것 같이 않는 것을 것 같이 않는 것 않는	B ssdoanfig.xml Added a separate OOO scheduler that supports NMMe priv	ettes 9 months ago	
김 것동 사람이 잘 맛있게 뭐야가지요. 신영에 관정할	workload.ami Added a separate OOO scheduler that supports NVMe private private and private	etties 9 months ago	Packages
지 않는 것은 것 않는 것 같은 것 같은 것 같은 것	RADDRE ond MQSim: A Simulator for Modern NVMe and SA	ATA SSDs	No packages published Contributors (?)
verview	Usage in Linux		Languages
	Run following commands:		
-learning truchniques, especially convolutional neural networks (SNR), have perveded vision explications across image classification, face pation, vision precessing, and so on ult to the high depend efficiency (http://www.fit.Bith.linkshyr.end.com/ acrossings). Reference and well as a merid by a planess and the hypothese face. Other and purple	5 make 5 ./MQSim -i <550 Configuration Files -w -deralead Definition Files		Differ 0.2%

NPU Simulator: MAESTRO¹⁾ SSD Simulator: MQ-Sim²⁾

1) https://maestro.ece.gatech.edu/ 2) https://github.com/CMU-SAFARI/MQSim



Evaluation Methodology

- We evaluate our platform's effectiveness by
 - 1) Comparing the memory cost of Behemoth against the conventional TPU-based DNN training system

N T 1 1

2) Comparing the training throughput of FMS against conventional SSDs

SSD Simulator:

MQ-Sim²⁾

E&TRO Horse GitHub Docs Resources	Why Github? ~ Team Enterprise Explore ~ Marketplace	Pricing - Search	🖉 Sign is
	CI CMU-SAFARI / MQSim	ecurity 🖂 Insights	© Watch 24
AAESTRO.	P maater + P 1 branch ⊗ 0 tags	Go to file 📃 ± Code -	About MDSim is a fast and accura
ALOTINO.	💝 arashta Merge puli request #33 from civila/naster 🖃	1d1a2r7 on May 15, 2020 3147 commits	simulator modeling the per of modern multi-ousue (MI
Open-source Infrastructure for	fast18 Merge remote-tracking brand	ch 'origin/master' 3 years ago	well as traditional SATA ba
	erc Correct minor typos	9 months ago	bandwidth protocol imple
eling Dataflows within Deep	traces Code refactored	3 years ago	steady-state SSD conditio full end-to-end latency of
MEAN NOT 2015 MEAN AND AND AND AND AND AND AND AND AND A	D LICENSE First source commit	3 years ago	modern SSDs. It is descri in the FAST 2018 paper b
ng Accelerators	D MQSim.ain Windows complation files	3 years ago	& people.inf.ethz.ch/one
법입니다. 이번 방법은 것 같아요. 정말 방법은 것이 같아요. 이는 사람들이 없다.	D MQSimucoproj Added a separate OOO sche	duler that supports NMMe priorities 9 months ago	D Readme
ALC NO. S.	MQSimucoproj.filters Added a separate OOO sche	duler that supports NVMe priorities 0 months ago	MIT License
Moth Deno	MQSimucxprojuser Default execution arguments Modeline variability	in Visual Studio 3 years ago	Deleases
	D READARD md Added new option for transa	ction scheduling 9 months ago	No releases published
	ssdconfig.xml Added a separate OOO sche	duler that supports NMMe priorities 9 months ago	
	Monitoadiami Added a separate OOD schere	duler that supports NVMe priorities 9 months ago	Packages
한 정도 가 없는 것이 같은 것이다.	RELEVANCE AND MUSIC A Simulator for Modern	NVMe and SATA SSDs	Contributors (2)
view	Usage in Linux Ban following commands:		Languages
ing techniques, especially convolutional neural networks (2006, have pervaded vision explications across image classification, face vision precessing, and us on due to the high degree of accuracy they previde, their holenty and academic are exploring specialized	5 make 5 ./MDSim -1 <ssd configuration="" file=""> -w -akurklaad Def</ssd>	initian File>	C++ BBON INTEL 10 Other 0.2%

Model	Size	(GB)	(GB)	PFLOP
	1×1	44	350	2.15
	1×2	88	698	4.42
BERT/GPT3-like	1×4	175	1393	8.56
DEI(I/OI TO IIIC	2×1	88	1395	8.56
	2×2	175	2786	17.12
	2×4	349	5569	34.21
	1×1	40	305	0.62
	1×2	80	609	1.25
	1×4	160	1218	2.49
I S-IIKE	2×1	80	1218	2.49
	2×2	160	2436	4.99
	2×4	319	4871	9.97
— • • • • • • •				

Total act.

Evaluation workloads

https://maestro.ece.gatech.edu/
https://github.com/CMU-SAFARI/MQSim

NPU Simulator:

MAESTRO¹⁾

2) https://github.cor	n/CMU-SAFARI/MQSim
LISENIX EAST'21	Behemoth: A Flash-centric Training Accelerator for Extreme-scale DNN

Total weight



Platform configurations for the cost comparison

Memory cost¹⁾ comparison between TPU V3 and Behemoth

• To maintain the same training throughput, TPUv3-like platform costs up to 3.65x the memory cost

1) HBM: \$20/GB, SLC NAND: \$0.67/GB, DDR DRAM: \$4/GB

Training Throughput Evaluation

- We compare Behemoth and baseline system utilizing the commodity SSDs
 - Behemoth with 2TB FMS
 - Behemoth core with 500GB of 4x SSDs (RAID 0)

Storage Parameters				
	Behemoth FMS	Baseline SSD		
	2ТВ,	500GB,		
NAND	64 channels,	16 channels,		
Configurations	2 chips/channel,	2 chips/channel,		
	1 die/chip	1 die/chip		
Channel	1200	MT/s		
Speed Rate	(MT/s: Mega Transf	ers per Second [20])		
NAND	128Gb SLC / die	e: 8 planes / die,		
Structure	683 blocks / plane, 768 j	pages / block, 4KB page		
NAND	Paad: 246 Program: 10046 Plack areas: 5mg			
Latency	Keau. 5μ s, Flogram. IV	Jours, block erase: Jins		
	SRAM 16MB	DRAM 512GB:		
Buffer	6MB for FTL metadata, 10MB for I/O buffer	FTL metadata		
Configurations		SRAM 8MB:		
		I/O buffer, GC Buffer		
FTL	Block mapping	Page mapping,		
Schemes	Block mapping	Preemtible GC [38]		
OP ratio	N/A	7%		
Firmware	N/A	Write:		
Latency	IN/A	1.45µs / a page (4KB)		
	Read:			
Contoller	1.93µs / an NVMe Cmd,	Read:		
Latency	Write:	1.93µs / an NVMe Cmd		
	1.18µs / an NVMe Cmd			

Training Throughput Evaluation



- Behemoth is close to the ideal case
- Conventional SSDs show much lower training throughput (up-to 2.05x)
 - SSD firmware bottleneck is major cause for performance degradation

Behemoth enables efficient data-parallel training of extreme-scale DNN models

- Analyze the memory capacity problem for extreme-scale DNN model training
- Identify new opportunities to leverage NAND flash devices to hold those models
- Present a novel flash-centric DNN training accelerator
- Show 3.65x memory cost savings over TPUv3 and 2.05x training throughput impr ovement over conventional SSDs

19th USENIX Conference on File and Storage Technologies (FAST '21)

Thank you !

Additional details in the paper:

- Analysis of Transformer: a key enabling primitive for extreme-scale DNNs
- Discussion of architectural decisions
- Coverage analysis for various DNN models
- Endurance evaluation