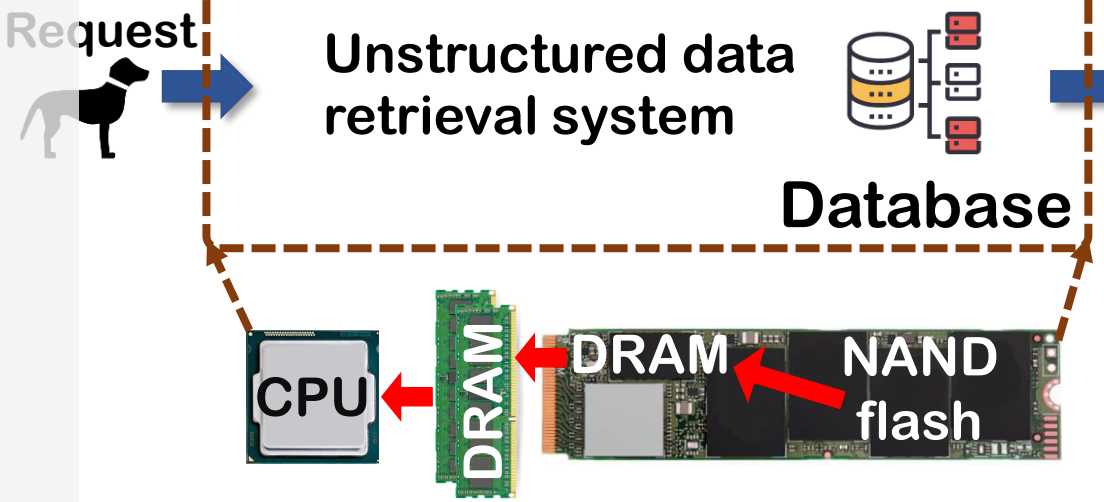
Cognitive SSD: A Deep Learning Engine for In-Storage Data Retrieval

Shengwen Liang[1,2], Ying Wang[1,2], Youyou Lu[3], Zhe Yang[3]
Huawei Li[1,2], Xiaowei Li[1,2]

[1]State Key Laboratory of Computer Architecture,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing
[2]University of Chinese Academy of Sciences
[3]Tsinghua University

Outline – Cognitive SSD

Inaccuracy

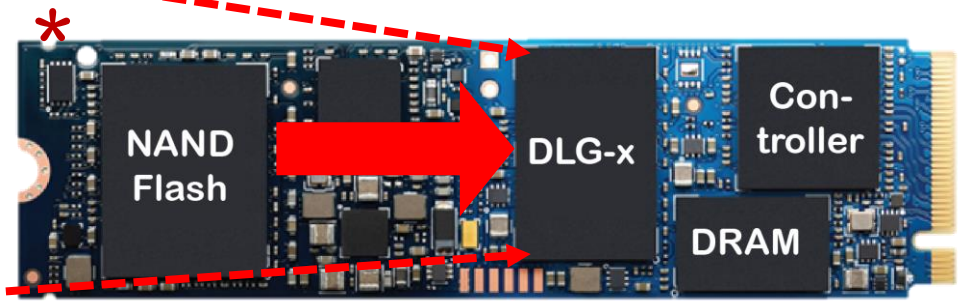


Results

Power inefficiency

Deep learning

Graph search



Cognitive SSD

Near data processing

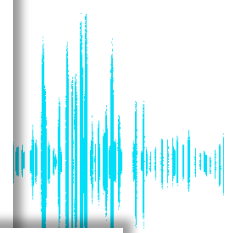
Accuracy

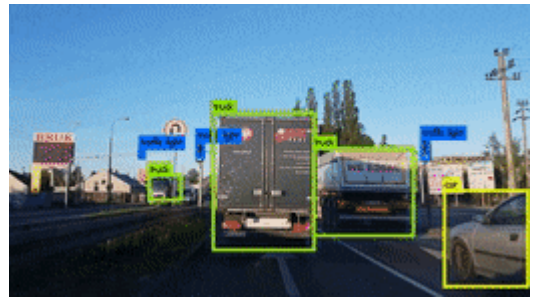
Power efficiency

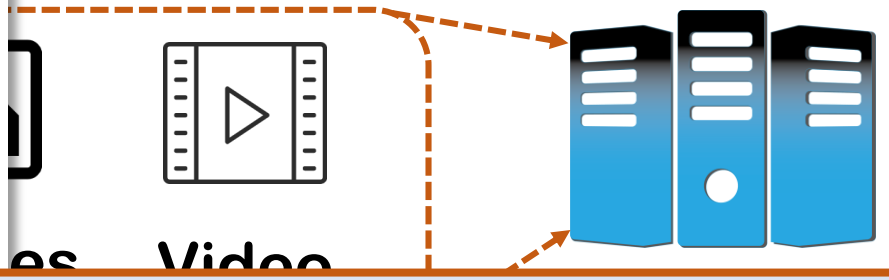
Outline

- Background and Motivation
- Cognitive SSD System
- DLG-x Accelerator
- Evaluation
- Conclusion

Unstructured Data



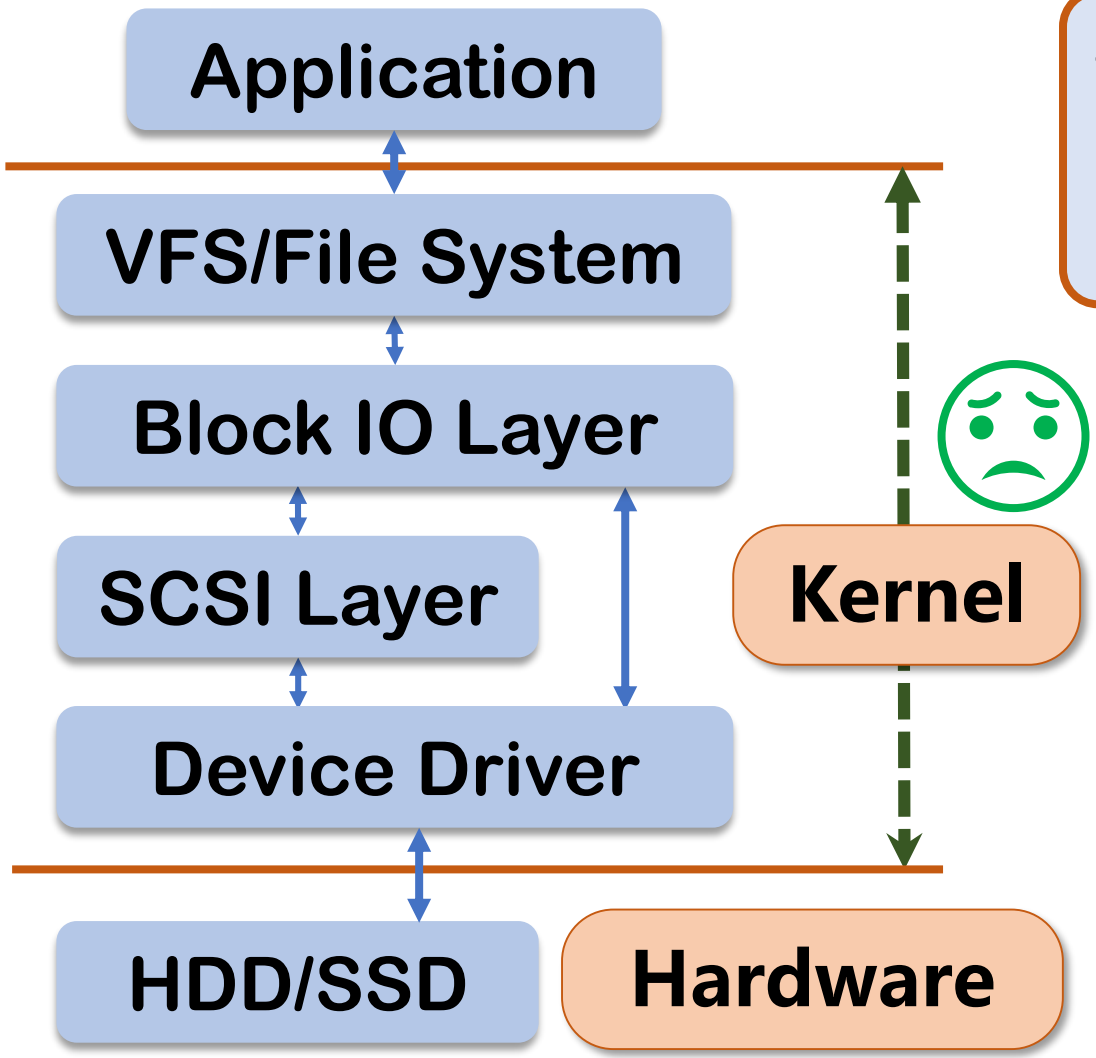




80%

- Unstructured data occupies to 80% of storage capacity in data centers [1].
- Intensive retrieval/analysis requests.
- Fast and energy-efficient data retrieval solution.

Problem- Software



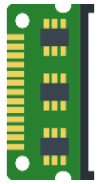
- Performance bottleneck migrates from hardware (SSD(75-50us[2])) to software (IO Stack (60.8us[3])).

HDD	NAND FLASH SSD	OPTANE SSD
		
2-5ms	75-50us[2]	10us
		5x

Problem-Hardware

HDD

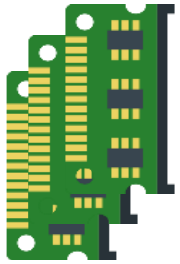
Cache



IO Interface
(SATA,PCIe)

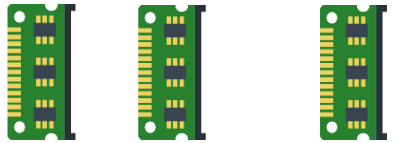
SATA
PCIe-4lane

DRAM

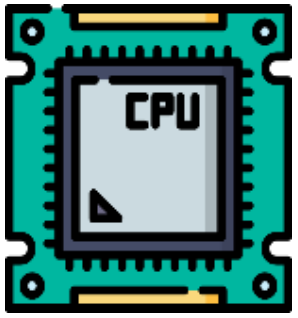


Cache Level

L3 L2 L1



Compute Unit



- Massive data movement incurs energy and latency overhead in the conventional memory hierarchy.

Showcase

Content Based Unstructured Data Retrieval System

Retrieval Request

Feature representation

Data Preprocessing

Feature Mapping

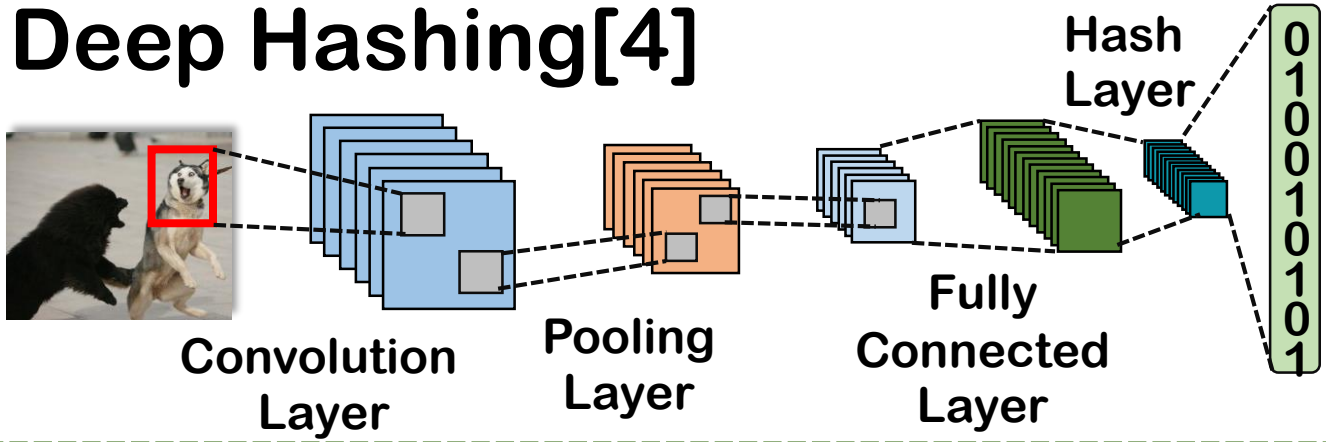
Database indexing

Feature Matching

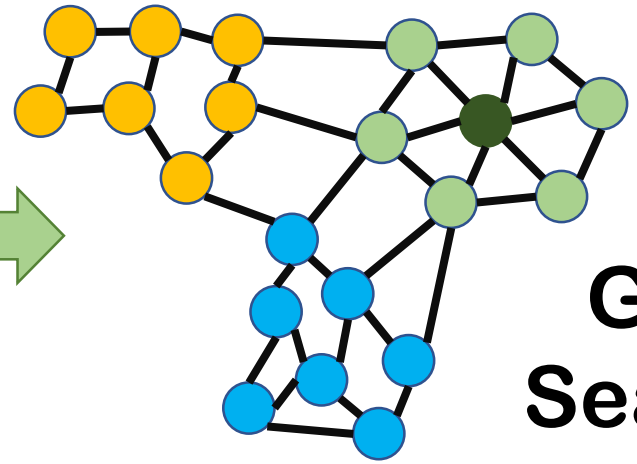
Ranking

Bad retrieval accuracy

Deep Hashing[4]



Better feature representation



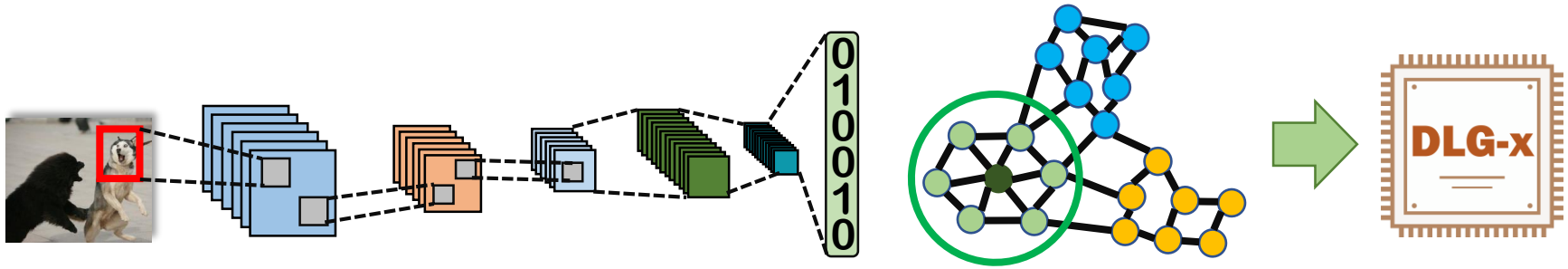
Graph Search[5]

Fast and high accurate retrieval performance

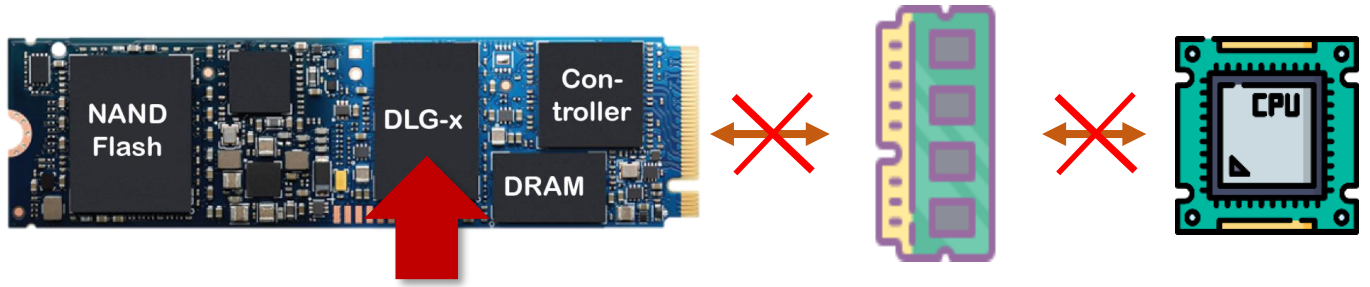
Solutions

☑ Deep Learning Hashing + Graph Search = DLG - accuracy

☑ Simplify the software stack



☑ Near-data processing
Shorten data path



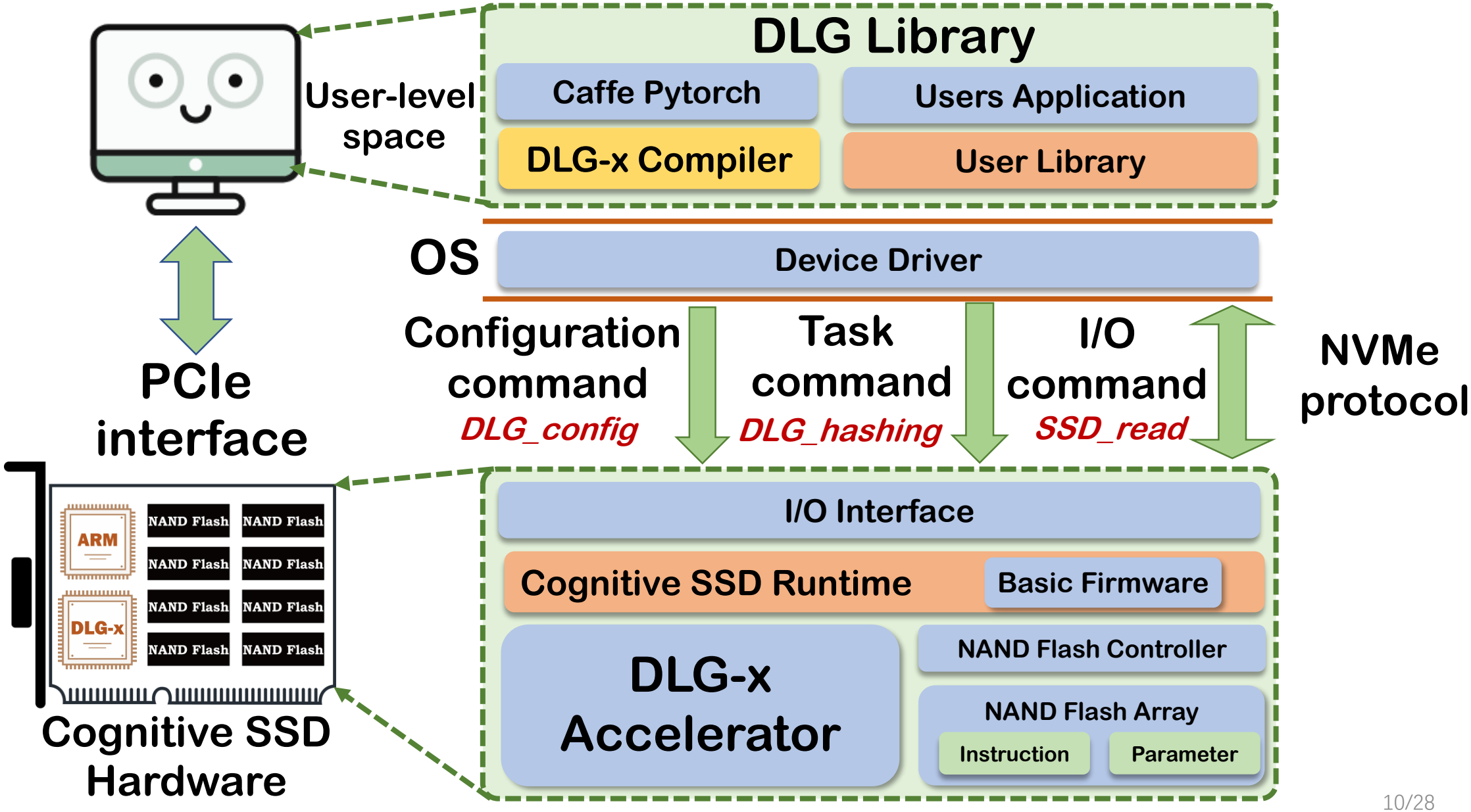
The internal bandwidth of SSD can be 16x higher than the external SSD bandwidth[6]

☑ User-visible Software abstraction → Scalability Different applications

Outline

- Background and Motivation
- **Cognitive SSD System**
  - **Overview**
  - **High-level library**
  - **Firmware and hardware**
- DLG-x Accelerator
- Evaluation
- Conclusion

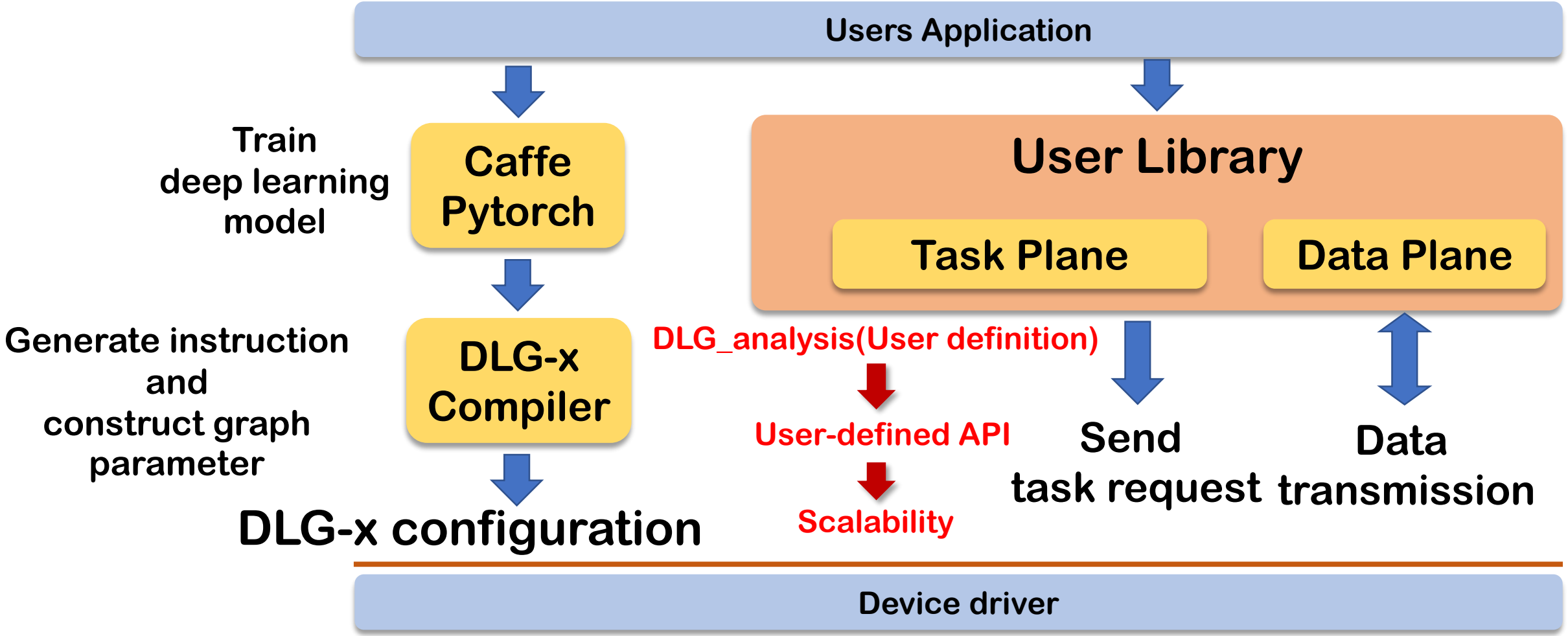
Cognitive SSD System--Overview



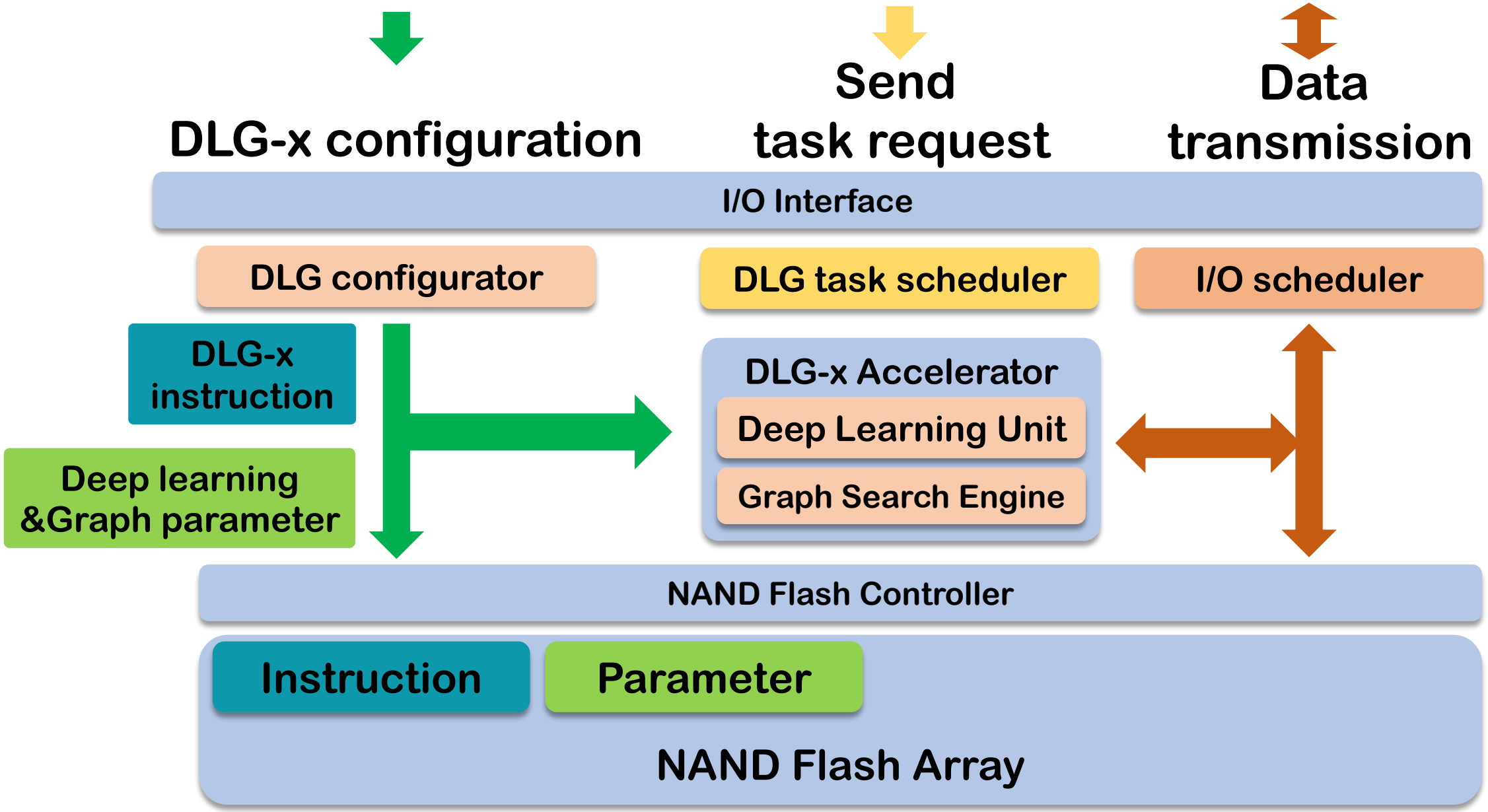
Cognitive SSD System—High Level Library

Challenge: Model and parameter configurable & scalability?

- How to update deep learning model and graph parameter?
- How to dispatch request?



Cognitive SSD System - Firmware and hardware

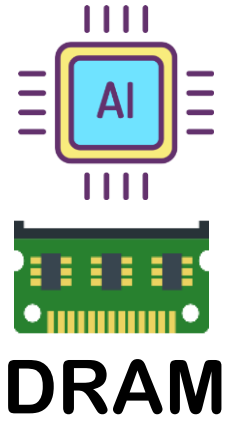


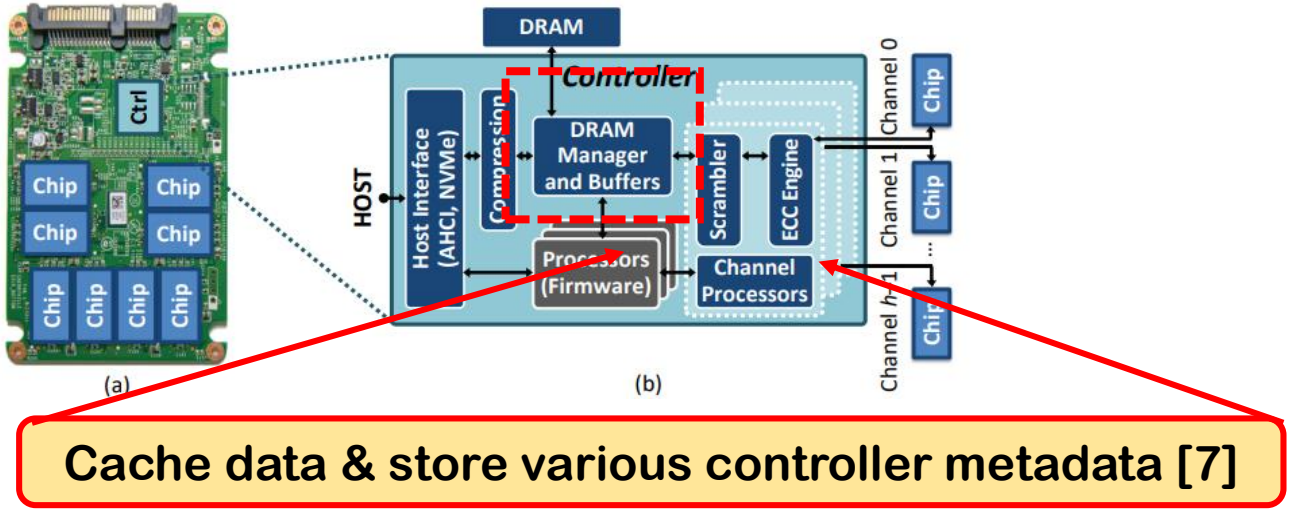
Outline

- Background and Motivation
- Cognitive SSD System
- **DLG-x Accelerator**
  - **Deep learning unit**
  - **Graph search unit**
- Evaluation
- Conclusion

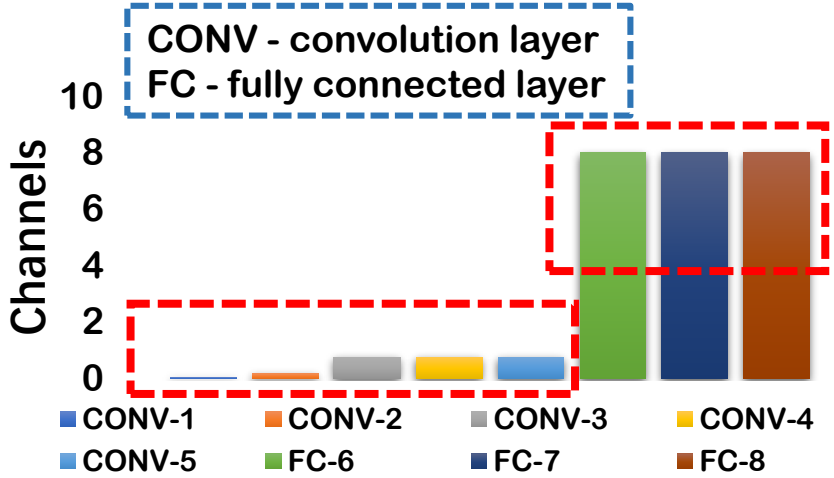
DLG-x Accelerator – Deep learning unit

Challenge: How to supply data to accelerator without DRAM?





Cache data & store various controller metadata [7]



NAND Flash | Controller

NAND Flash | Controller

NAND Flash | Controller

Directly access NAND flash

Weight Buffer-0

Weight Buffer-1

Instruction Queue

Neural Processing Engine

Convolution | Pooling | Activation

Graph Search Engine

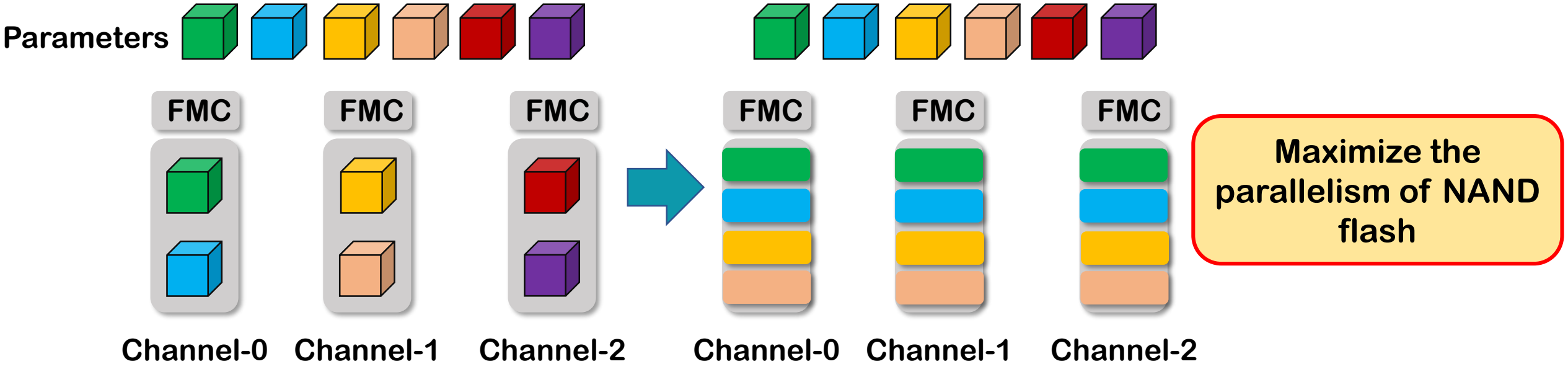
Vertex Detector | Vertex Arbitrator | Address Generator

InOut Buffer-0

InOut Buffer-1

DLG-x Accelerator-Data Layout

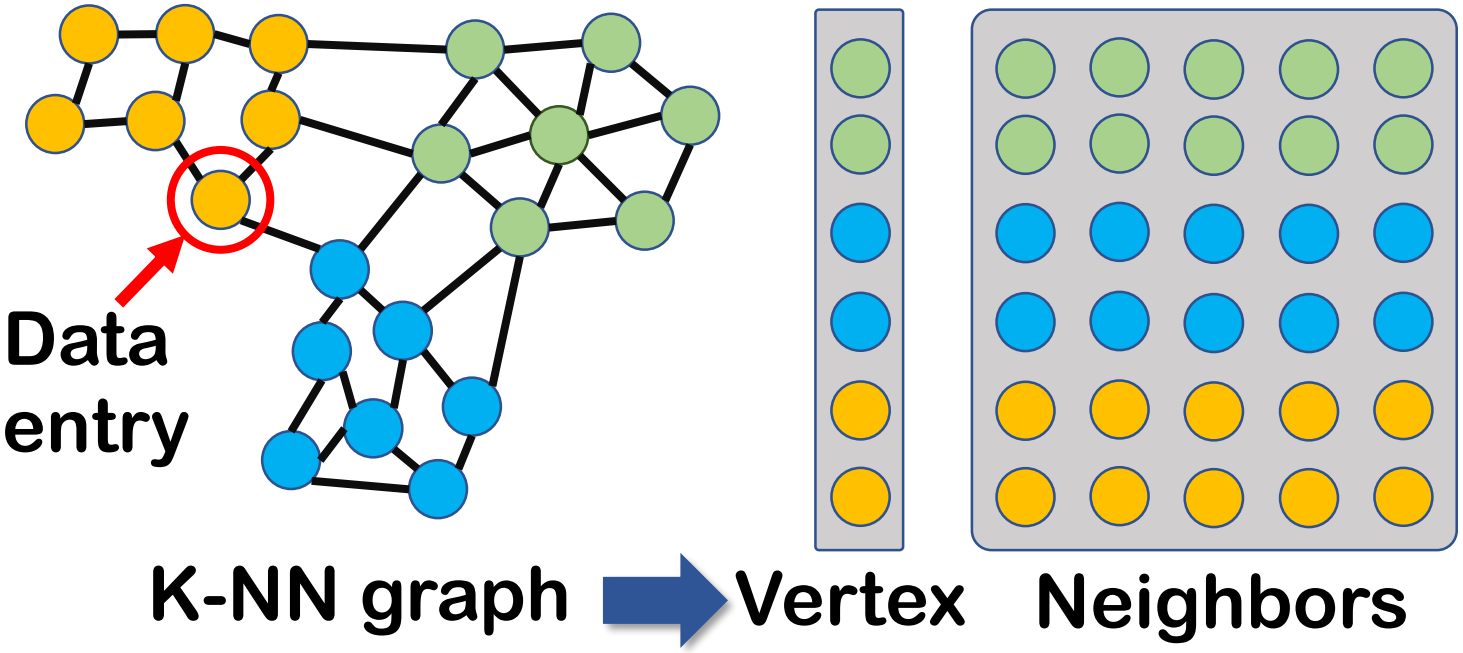
Challenge: How to fully utilize the internal bandwidth of flash?



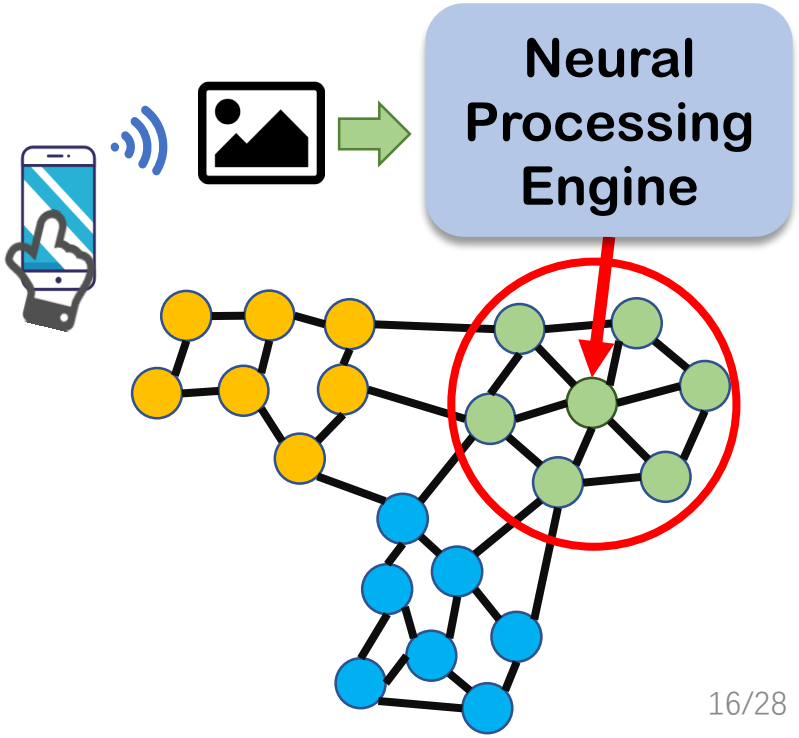
DLG-x Accelerator-Graph search unit

NAND Flash
Controller
NAND Flash
Controller
Weight Buffer
Weight Buffer
Graph Search Engine
Vertex Detector
Vertex Arbitrator
Address Generator
InOut Buffer
InOut Buffer

Offline stage: K-NN graph construction



Online stage: graph search



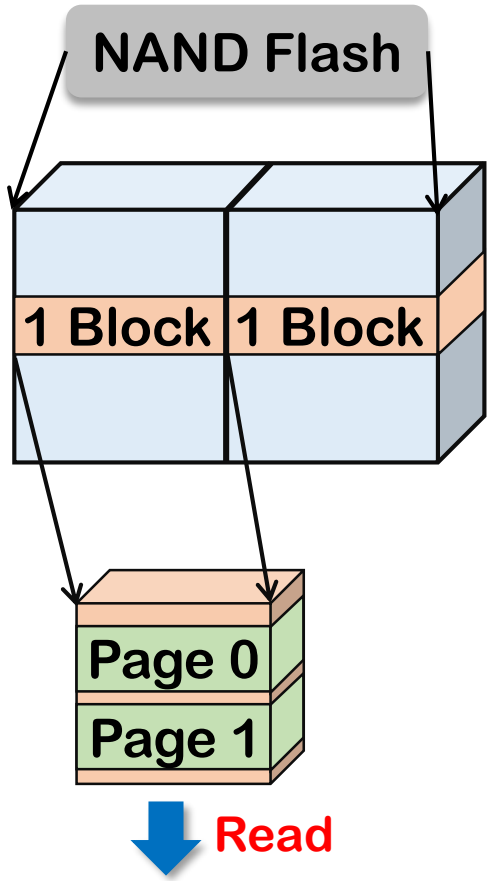
DLG-x Accelerator-Data Layout

Challenge: How to avoid bandwidth waste?

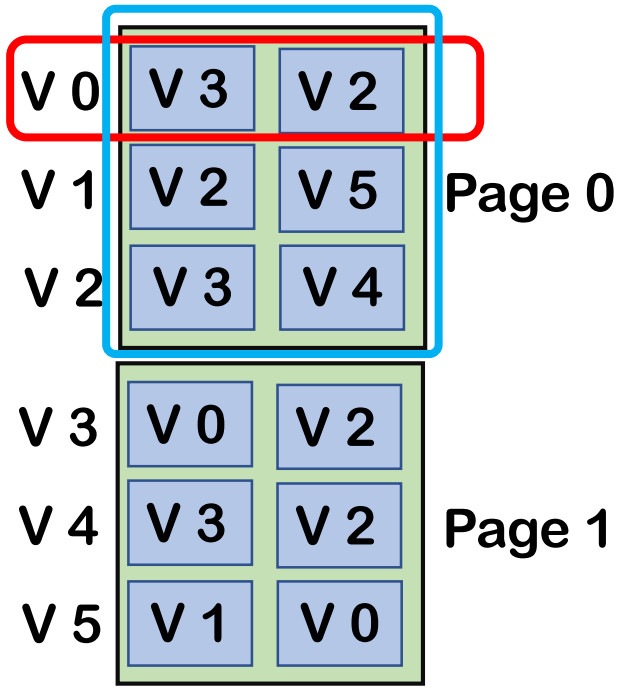
Read Amplification ☹ **Low bandwidth utilization**

ID (32) + Hash Code (48-512) = 80-544bits << 16KB

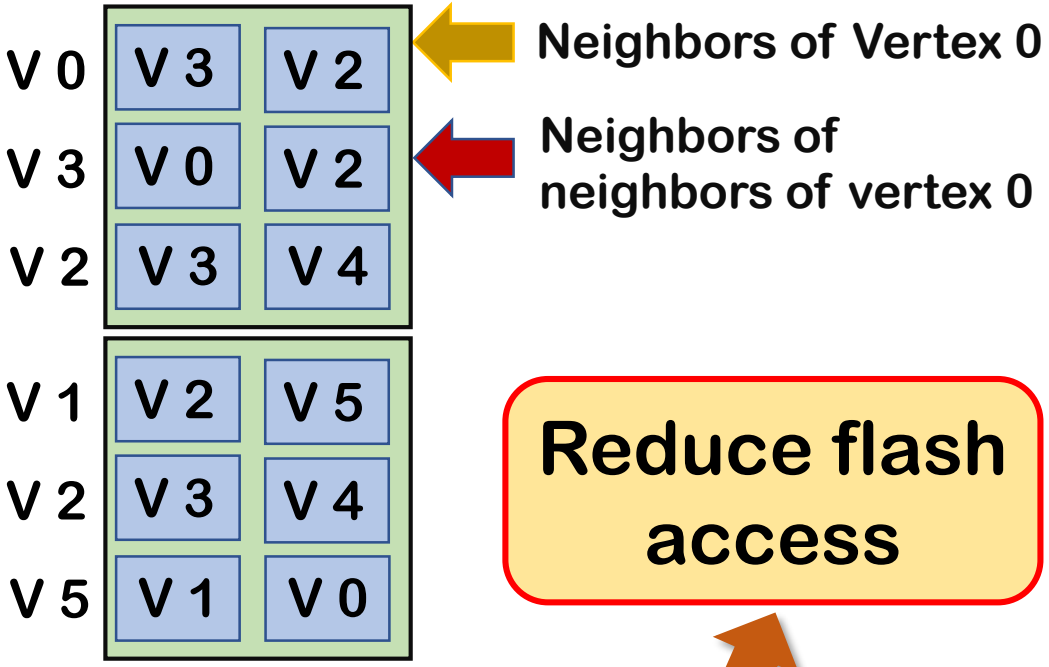
25 neighbors = 25*(80-544) bits = (250-1700) Bytes << 16KB



Page granularity



Data layout



Reduce flash access

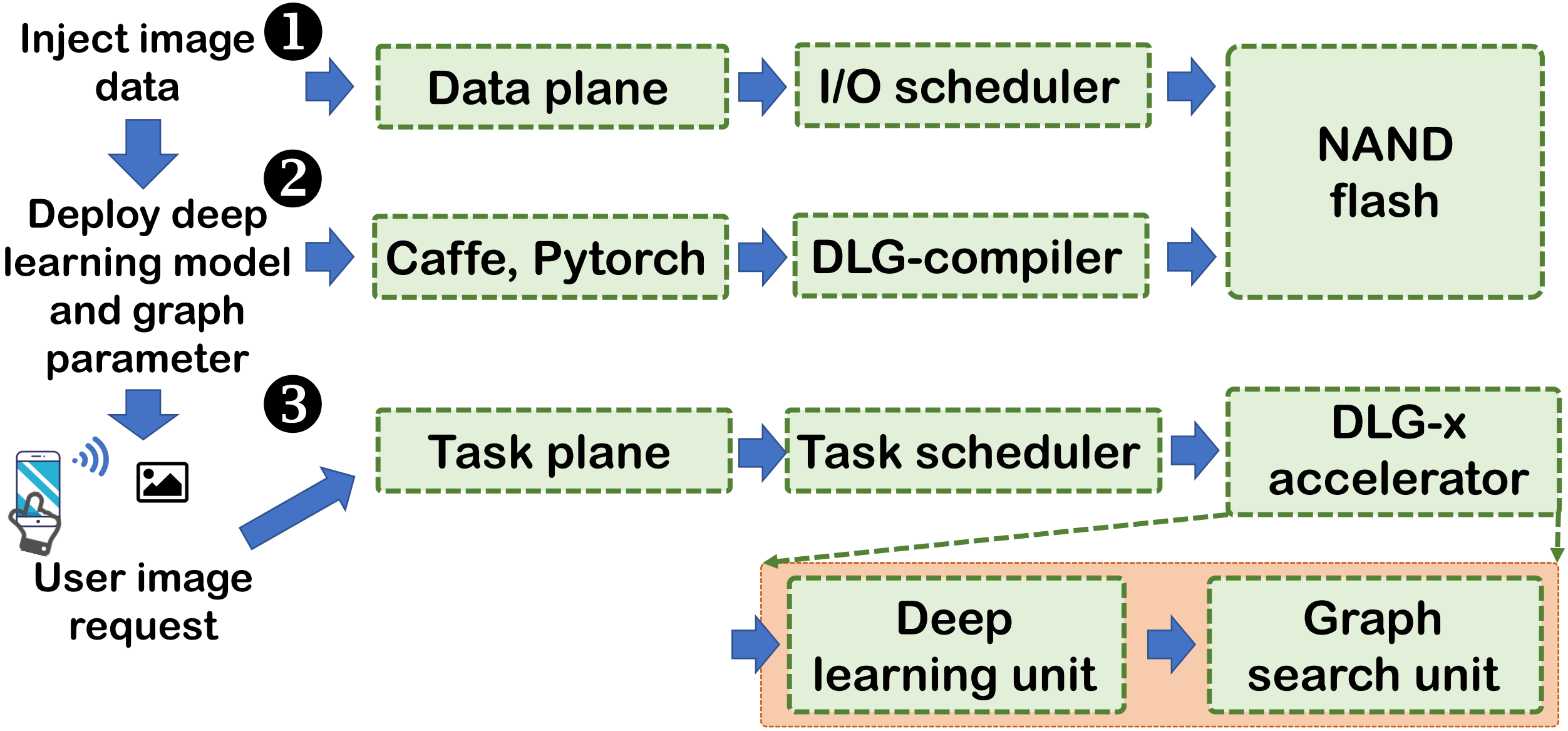
Read Vertex 0
Read Vertex 3
Read Vertex 2

Read Page 0
Read Page 1
Read Page 0

Read Vertex 0
Read Vertex 3
Read Vertex 2

Read Page 0

Cognitive System – Case study



Outline

- Background and Motivation
- Cognitive SSD System
- DLG-x Accelerator
- **Evaluation**
- Conclusion

Evaluation Setup

Hardware

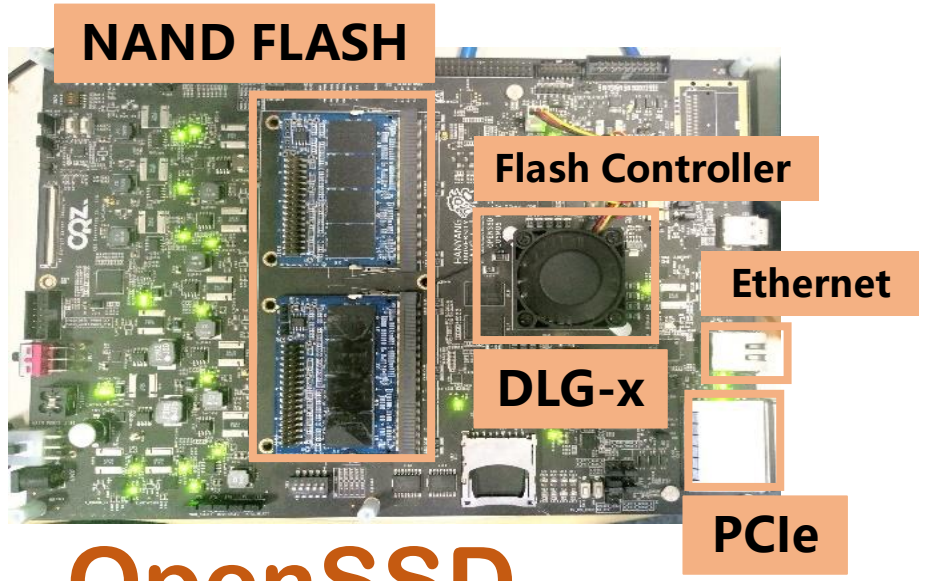
	CPU	DRAM	SSD	GPU	FPGA
B-CPU	2*Xeon E5-2630	32GB	4* 1TB PCIe SSD	-	-
B-GPU	2*Xeon E5-2630	32GB	4* 1TB PCIe SSD	NVIDIA GTX 1080Ti	-
B-FPGA	2*Xeon E5-2630	32GB	4* 1TB PCIe SSD	-	ZC706 Board
B-DLG-x	2*Xeon E5-2630	32GB	4* 1TB PCIe SSD	-	ZC706 Board
Cognitive SSD + CPU	2*Xeon E5-2630	32GB	3* 1TB PCIe SSD	-	OpenSSD
Cognitive SSD	ARM Dual Cortex A9	2GB	1TB NAND flash	-	OpenSSD

Software

Ubuntu 14.04, Caffe[9], Crow web framework[10].

Workload

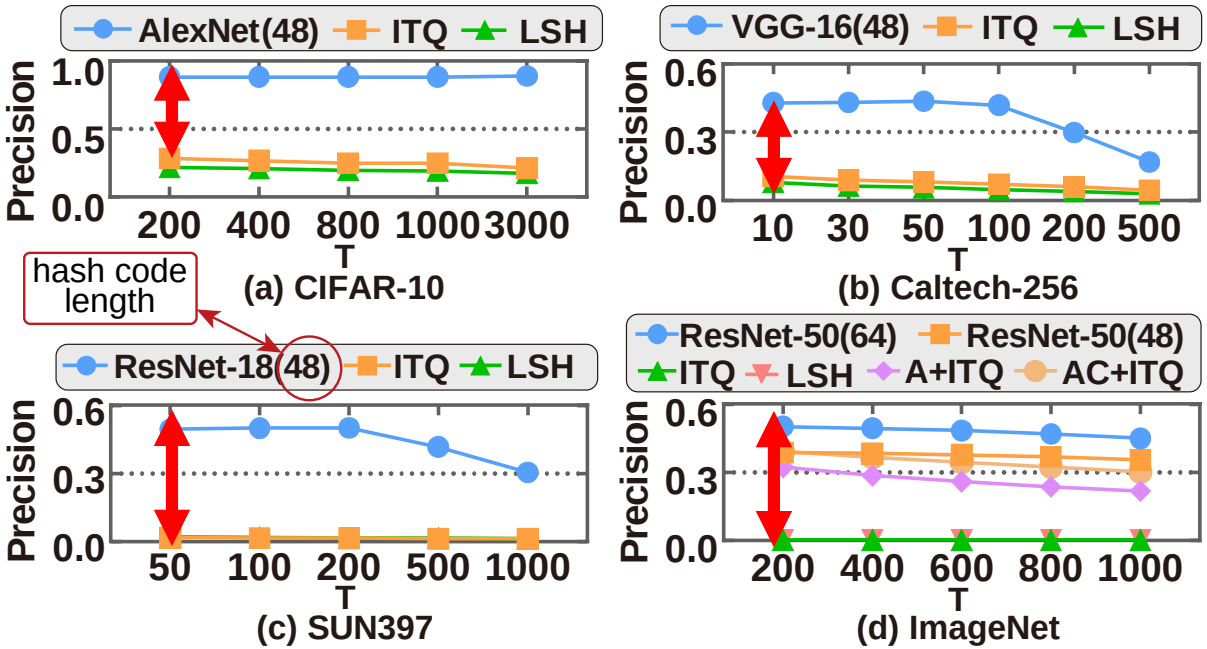
Content-Based Image Retrieval System (CBIR)



OpenSSD

1. Zynq FPGA Chip – DLG-x and flash controller
   1. Dual Cortex A9 -- Firmware
2. 1GB DRAM
3. 8-channels NAND flash
4. Ethernet
5. PCIe Gen 2 (maximum lane = 8)

Evaluation-DLG algorithm



Dataset

Dataset	Total	Train/Validate	Labels
CIFAR-10	60000	50000/10000	10
Caltech256	29780	26790/2990	256
SUN397	108754	98049/10705	397
ImageNet	1331167	1281167/50000	1000

- DLG solution performs **better retrieval accuracy** regardless of the choice of T value when compared to the conventional hash solutions.
- DLG solution shows the **robustness** of the DLG solution when deployed on a real-world system.

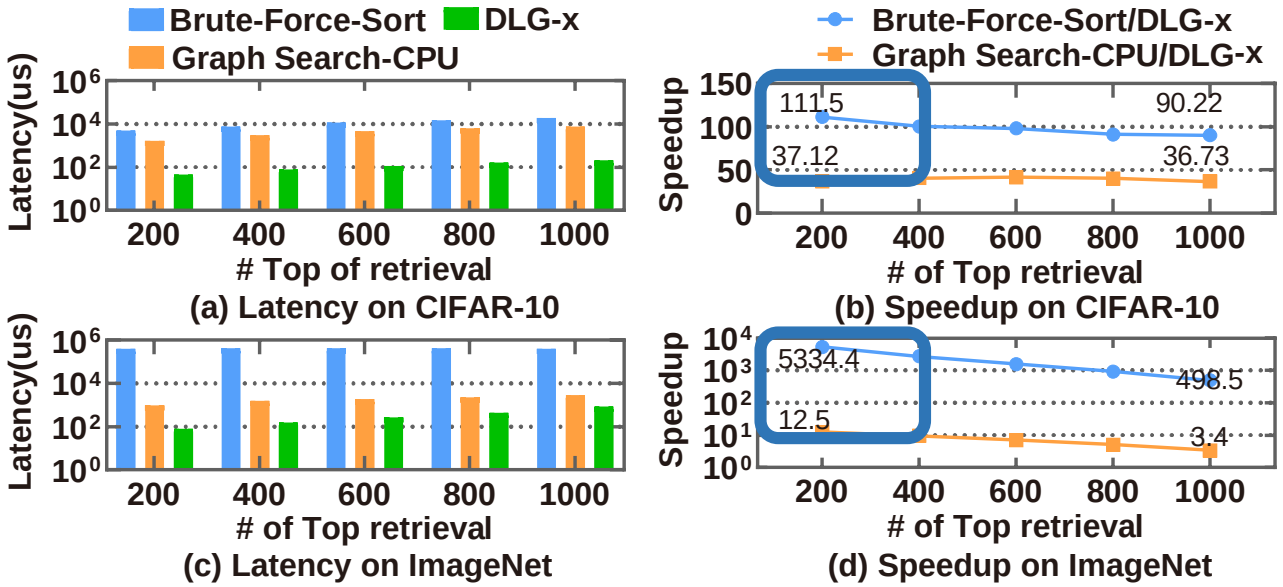
Evaluation-DLG-x

Performance of deep hashing on DLG-x

Model	-	Latency(ms)	Power(w)
Hash AlexNet	DLG-x	38	9.1
	CPU	114	186
	GPU	1.83	164
Hash ResNet-18	DLG-x	94	9.4
	CPU	121	185
	GPU	7.13	112

- Faster than CPU solution
- More power-efficiency than GPU solution

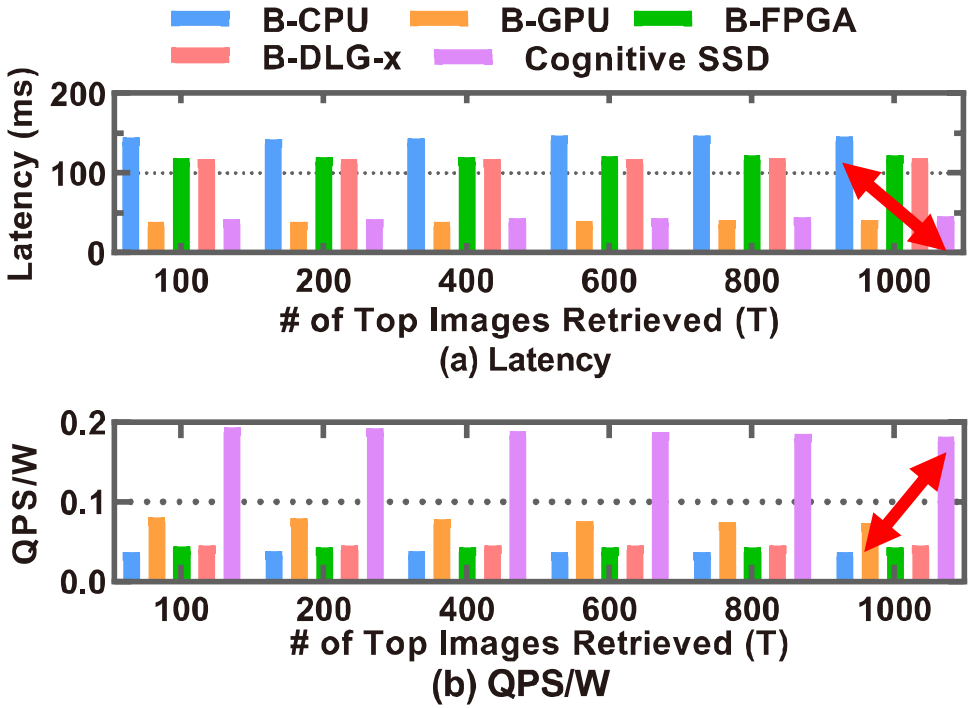
Performance of graph search on DLG-x



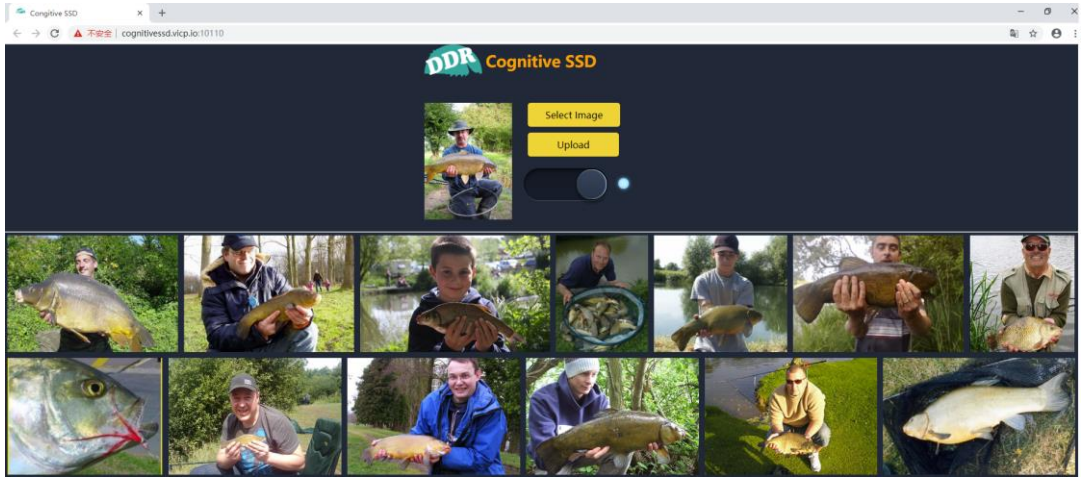
- Outperform than brute force sort method
- Up to 37.12 x and 12.5 x speedup over CPU solution

Evaluation-Cognitive SSD System

Performance of Cognitive SSD system on ImageNet

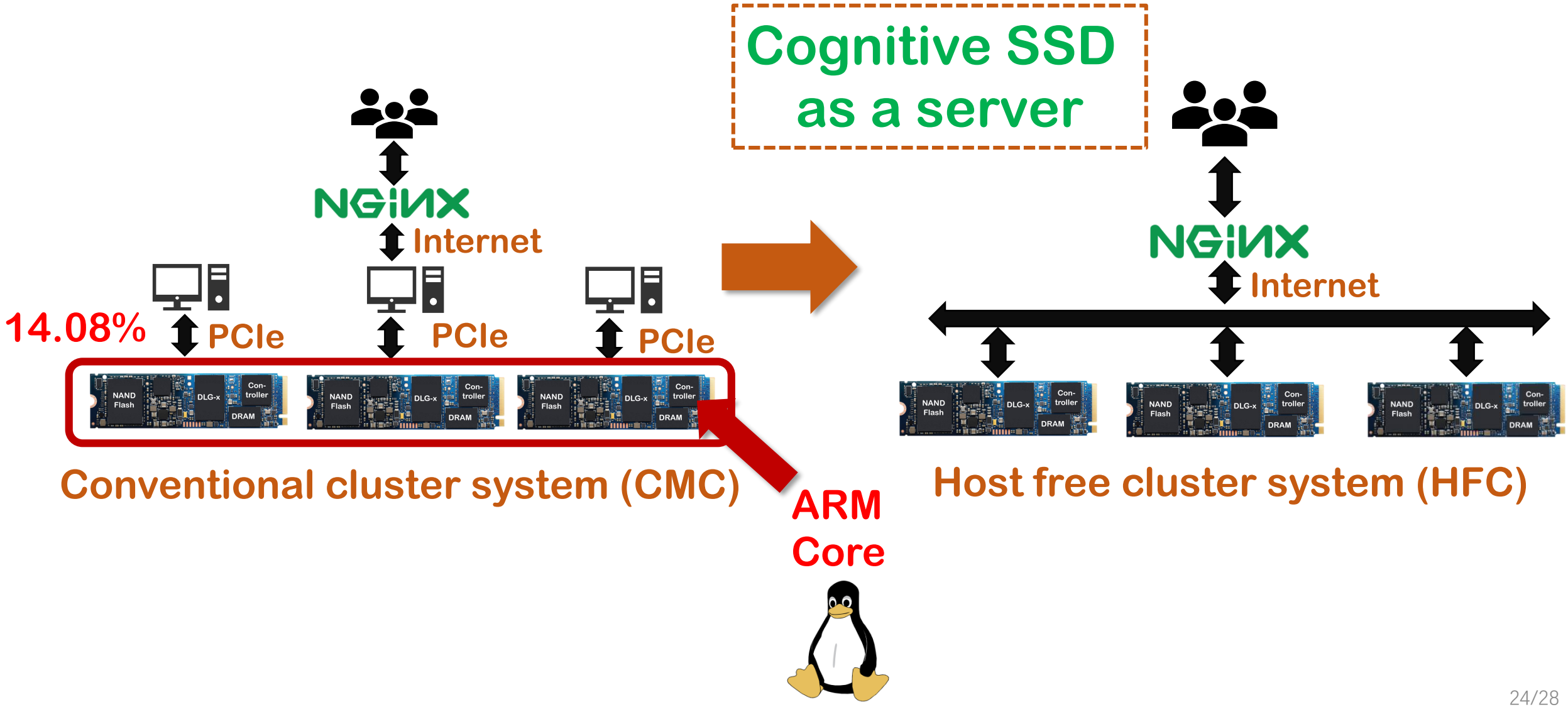


Web Demo

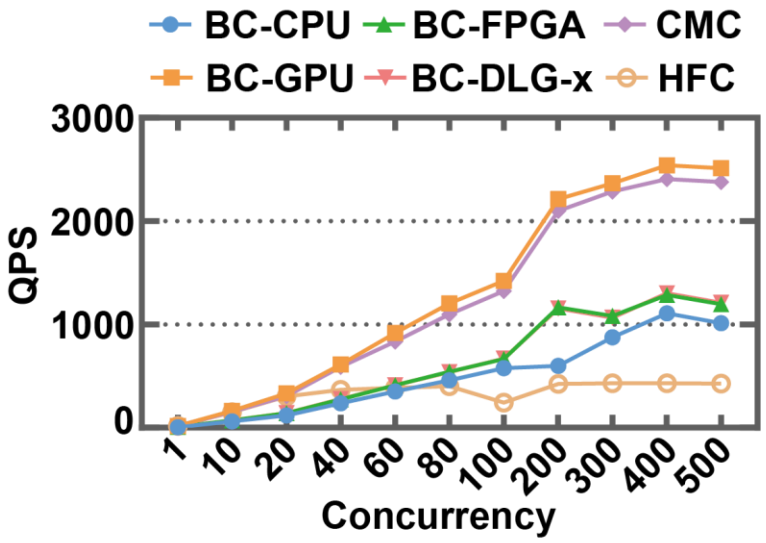


- Compared to B-CPU, Cognitive SSD system reduces latency by 69.9% on average.
- Cognitive SSD achieves higher power-efficiency than B-GPU system by 2.44 x.

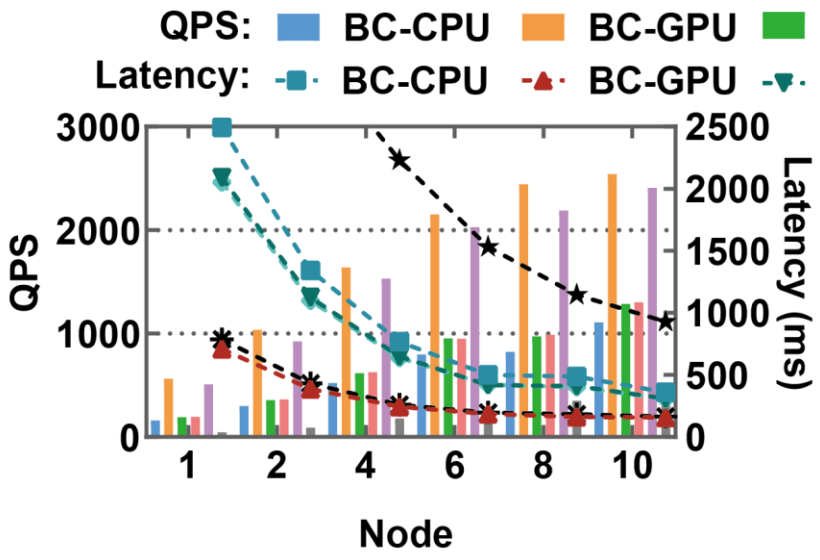
Evaluation-Cognitive SSD Cluster



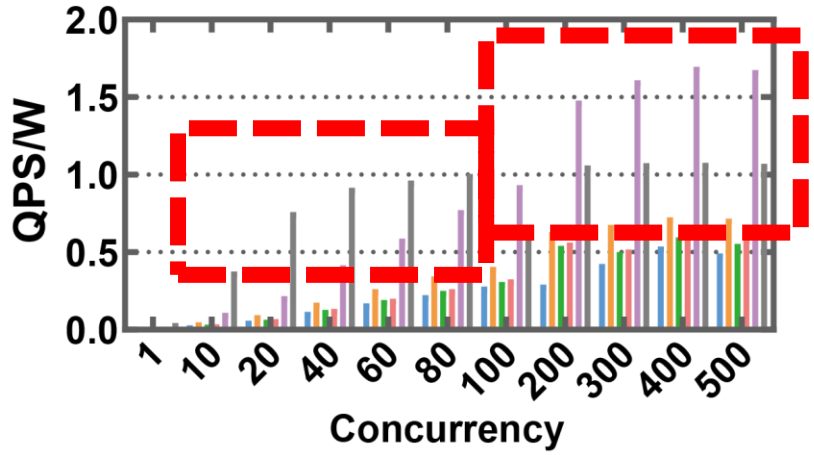
Evaluation-Cognitive SSD Cluster



(a) QPS w.r.t Concurrency



(b) QPS&Latency under different nodes



(c) QPS/W w.r.t Concurrency

- The power-efficiency of the HFC system is better than other baselines when users requests are low.
- HFC system will perform better power-efficiency if the Cortex-A9 processor is replaced by the latest Cortex-A series processor.

Conclusion

- Cognitive SSD provides a more power-efficient solution for unstructured data retrieval.
- The DLG-x accelerator integrates deep learning and graph search into one chip and directly accesses data from NAND flash without crossing multiple memory hierarchies.
- FPGA-based prototype evaluations show that Cognitive SSD outperforms other solutions on power-efficiency.

Q&A

Cognitive SSD: A Deep Learning Engine for In-Storage Data Retrieval

Shengwen Liang[1,2], Ying Wang[1,2], Youyou Lu[3], Zhe Yang[3]
Huawei Li[1,2], Xiaowei Li[1,2]

[1]State Key Laboratory of Computer Architecture,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing
[2]University of Chinese Academy of Sciences
[3]Tsinghua University

中国科学院大学
University of Chinese Academy of Sciences

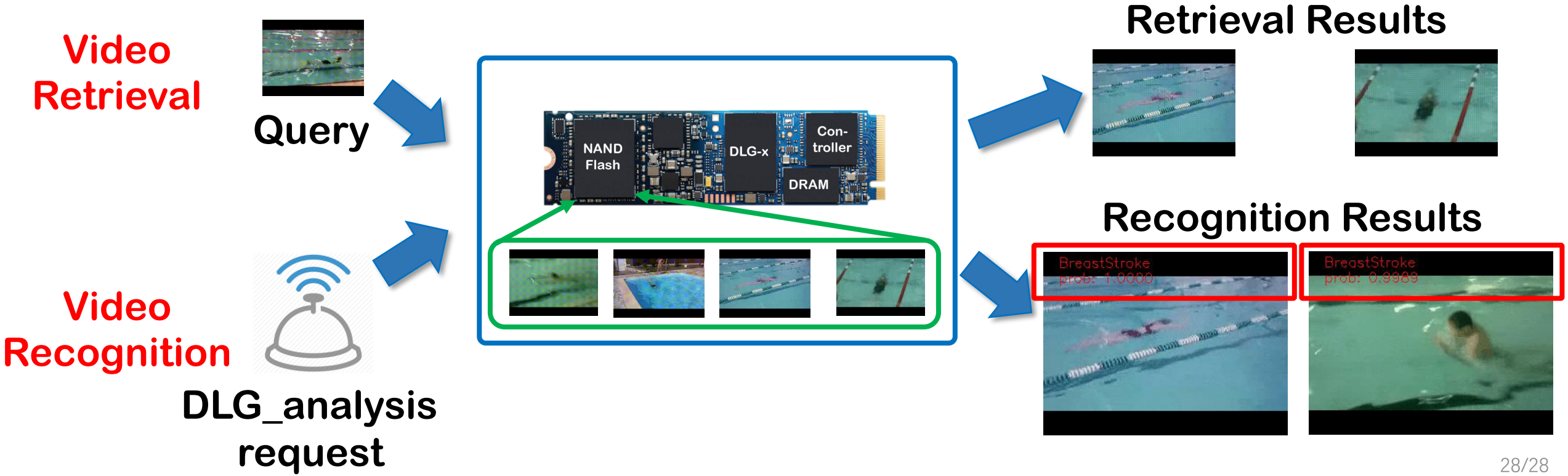
清華大學
Tsinghua University

If you have any questions, please contact us
Email: liangshengwen@ict.ac.cn

Cognitive System - Scalability

- Cognitive SSD system also supports **other applications** and not be limited by the image data retrieval!

- The task plane provides the **user-defined API (*DLG_analysis*)** interface to enable users to deploy other applications without bigger modification.



Reference

[1] The biggest data challenges that you might not even know you have, May 2016. https: //www.ibm.com/blogs/watson/2016/05/ biggest-data-challenges-might-not-even-know/.

[2] “MLC 128Gb to 512Gb Async/Sync NAND,” p. 239, 2017.

[3] Y. Son, N. Y. Song, H. Han, H. Eom, and H. Y. Yeom, “A User-Level File System for Fast Storage Devices,” in *2014 International Conference on Cloud and Autonomic Computing*, 2014, pp. 258–264.

[4] H. Liu, R. Wang, S. Shan, and X. Chen, “Deep Supervised Hashing for Fast Image Retrieval,” 2016, pp. 2064–2072.

[5] C. Fu, C. Wang, and D. Cai, “Fast Approximate Nearest Neighbor Search With The Navigating Spreading-out Graph,” *ArXiv170700143 Cs*, Jul. 2017.

[6] E. Doller, A. Akel, J. Wang, K. Curewitz, and S. Eilert, “DataCenter 2020: Near-memory acceleration for data-oriented applications,” in *2014 Symposium on VLSI Circuits Digest of Technical Papers*, 2014, pp. 1–4.

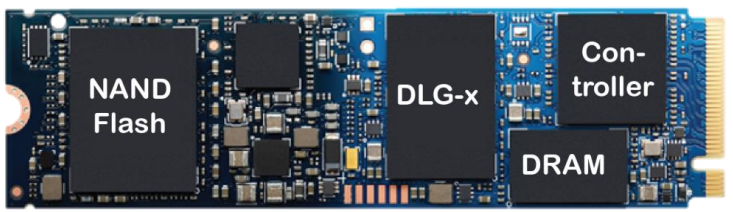
[7] Y. Cai, S. Ghose, E. F. Haratsch, *Y. Luo, and O. Mutlu, “Errors in Flash-Memory-Based Solid-State Drives: Analysis, Mitigation, and Recovery,” ArXiv171111427 Cs, Nov. 2017.*

[8] R. Micheloni, L. Crippa, and A. Marelli, Eds., *Inside NAND Flash memories*. Heidelberg ; New York: Springer, 2010.

[9] Y. Jia *et al.*, “Caffe: Convolutional Architecture for Fast Feature Embedding,” *ArXiv14085093 Cs*, Jun. 2014.

[10] J. Ha, *crow: Crow is very fast and easy to use C++ micro web framework (inspired by Python Flask)*. 2018.

[11] *http://www.thessdreview.com/ces-2019/intel-teases-h10-ssd-intel-optane-memory-with-qlc-3d-nand-in-single-m-2-module-ces-2019-udate/*



* This picture is modified from the web [11] and just for display, not actual Cognitive SSD system. The actual Cognitive SSD system is shown in page 20.