A Study of Sorting Algorithms on Approximate Memory

Shuang Chen[12]*, Shunning Jiang[12]*,
Bingsheng He[2]**, Xueyan Tang[2]

[1]Shanghai Jiao Tong University, China
[2]Nanyang Technological University, Singapore

*Now at Cornell University
**Now at National University of Singapore

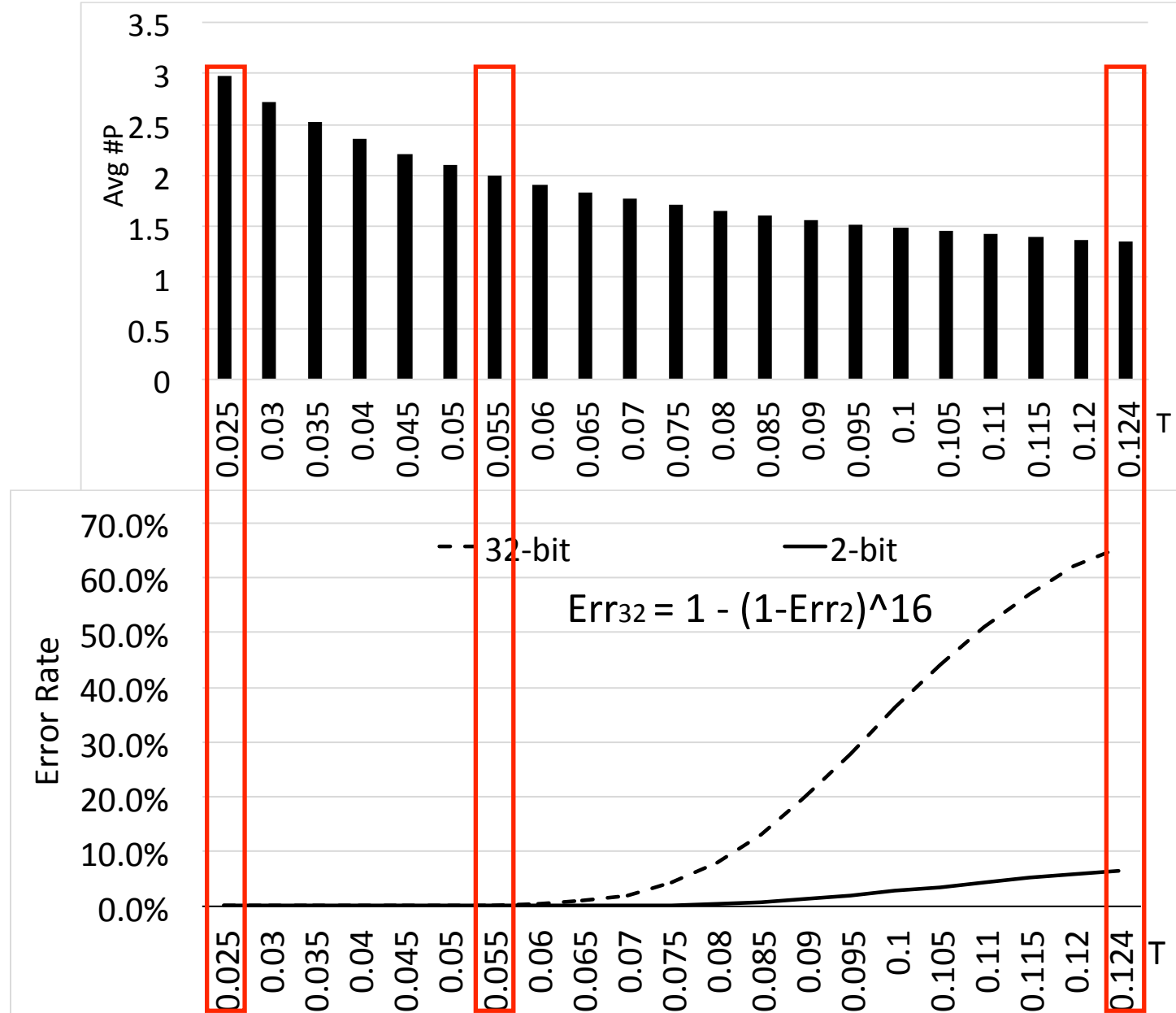# Hardware Evolution Drives Database Architectural Innovations

- Speed: Disk -> Memory
- Size: KB -> GB
- Persistency: DRAM -> NVRAM

NVRAM-based In-memory Database

Approximate NVRAM

- Trade off the accuracy of results for increased write performance and/or reduced energy consumption

Behavior of Approximate Memory

$$Err_{32} = 1 - (1-Err_2)^{16}$$

4

Approximate NVRAM

- Trade off the accuracy of results for increased write performance and/or reduced energy consumption

- Existing studies focus on how to offer approximate computing based on approximate NVRAM.

    - Ideal for apps with intrinsic tolerance to inaccuracies

        - Machine Learning

        - Image/Video processing

- We ask one *radical* question: can we use approximate NVRAM to accelerate **precise** computing?
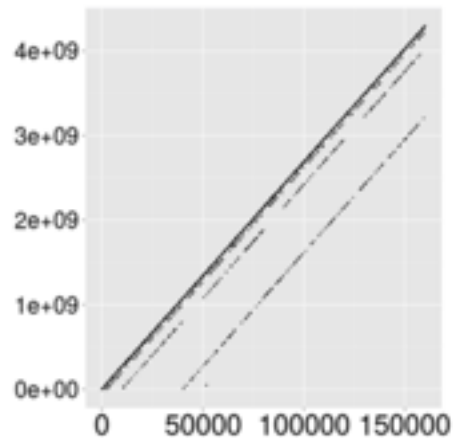
Database on Approximate NVRAM

- Opportunities
  - Better performance and/or less energy consumption from approximate storage
  - Tolerance of inaccuracies in intermediate results

An Example: Running Sorting Algorithms on Approximate Memory
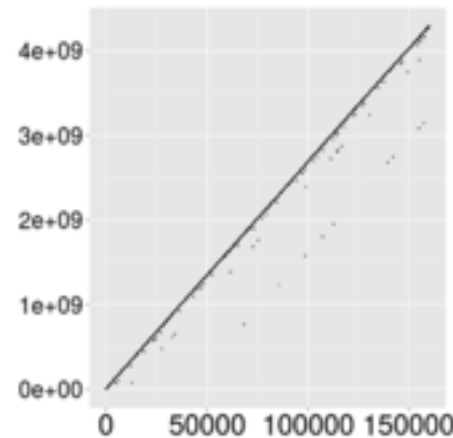
- 1.5x performance speedup
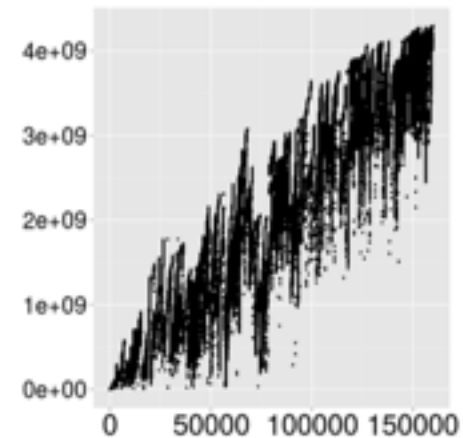- Get nearly sorted sequences
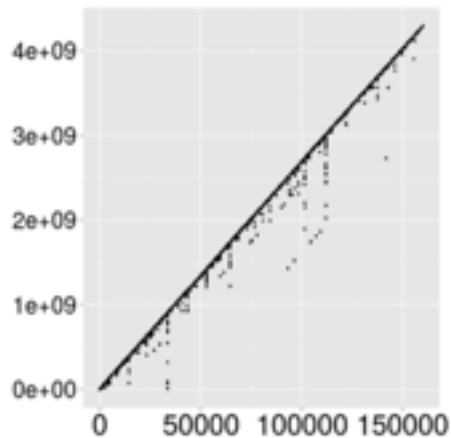
(a) Quicksort   (b) LSD   (c) MSD   (d) Mergesort
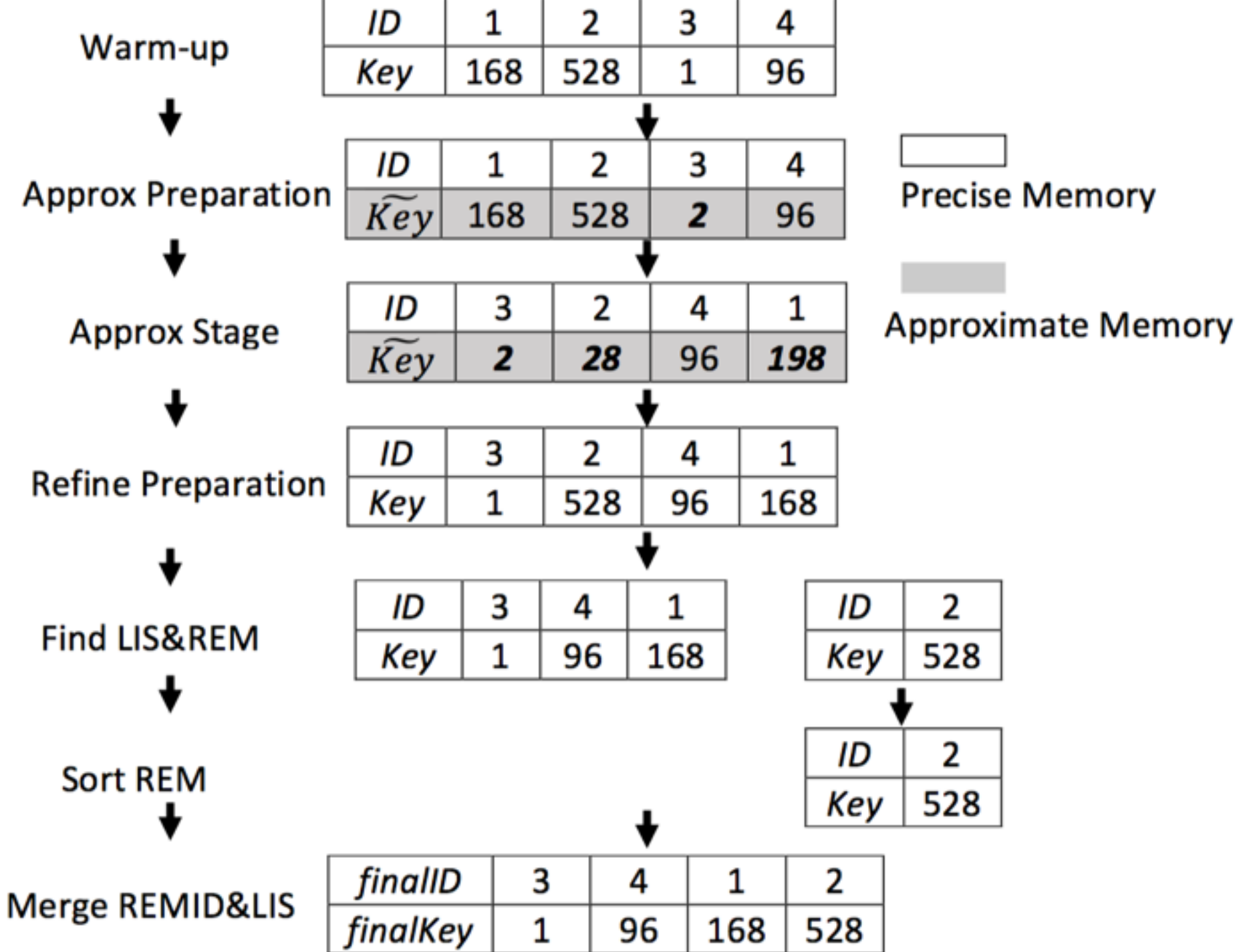
Database on Approximate NVRAM

- Opportunities
  - Better performance and/or less energy consumption from approximate storage
  - Tolerance of inaccuracies in intermediate results
- Challenges
  - Guarantee precise final results
  - Still need precise storage

A Radical Idea: ApproxiDB [VLDB'14]

- A radical data management system with its design, implementation and optimization aware of approximate storage.

- ApproxiDB will run on a **hybrid** machine consisting of both approximate and precise storage.

    - Approximate storage: acceleration

    - Precise storage: backup and recovery

- Start from sorting

Bingsheng He. When Data Management Systems Meet Approximate Hardware: Challenges and Opportunities. Proceedings of the VLDB Endowment, Volume 8 Issue 10, 2014, pages = {877--880}. (VLDB 2014).
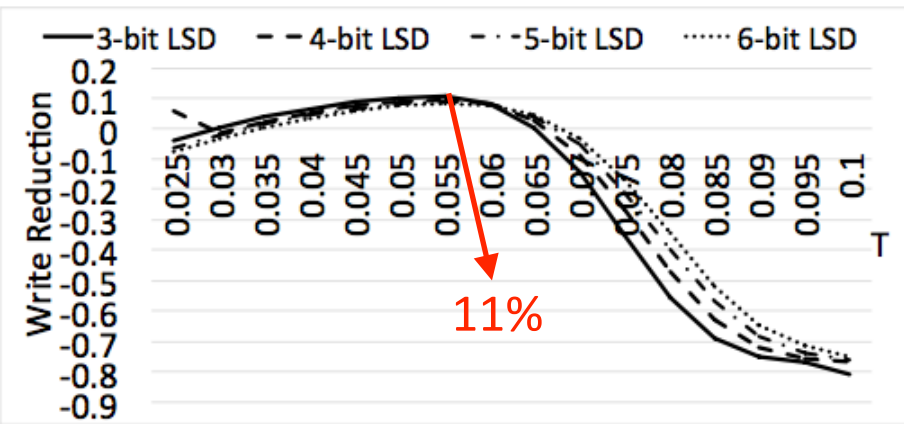
Approx-Refine

- Approx
  - Copy only data to approximate memory
  - Accelerate main sorting algorithm
  - Get a nearly sorted sequence
- Refine
  - Find LIS and REM
  - Sort REM
  - Guarantee a strictly sorted sequence

Warm-up

ID	1	2	3	4
Key	168	528	1	96

Approx Preparation

ID	1	2	3	4
$\widetilde{Key}$	168	528	**2**	96

	Precise Memory
	Approximate Memory

Approx Stage

ID	3	2	4	1
$\widetilde{Key}$	**2**	**28**	96	**198**

Refine Preparation

ID	3	2	4	1
Key	1	528	96	168

Find LIS&REM

ID	3	4	1
Key	1	96	168

ID	2
Key	528

Sort REM

ID	2
Key	528

Merge REMID&LIS

finalID	3	4	1	2
finalKey	1	96	168	528

Evaluation

- Trace-driven memory simulator
  - Detailed model of approximate NVRAM
- N = 1600, 16000, ..., 16M
  - random 32-bit integers
- T = 0.025, 0.03, ..., 0.124
  - Approximation degree of memory
- Write Reduction
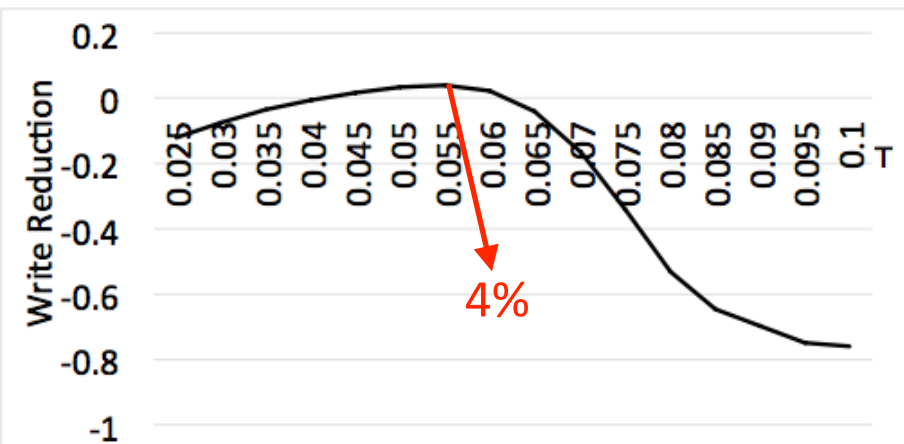 - reduced memory write latency

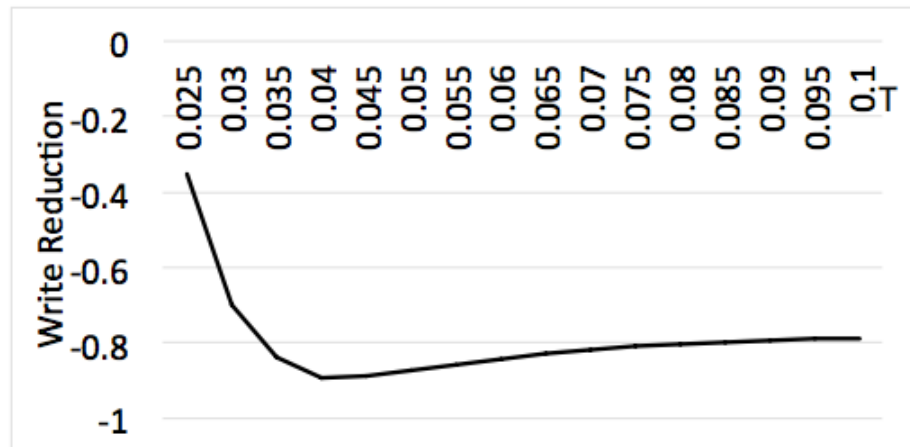# Selected Experimental Results

(a) LSD
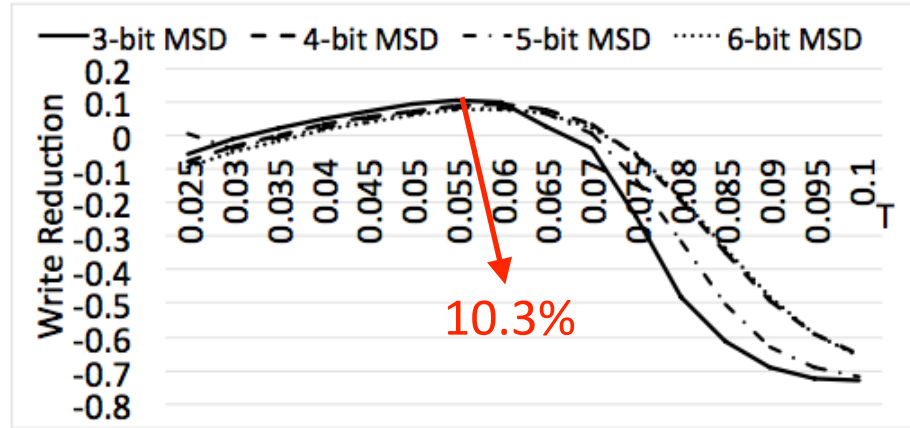
(b) MSD

(c) Quicksort

(d) Mergesort

Contributions

- The first to leverage precise computing on approximate storage

- A novel approx-refine mechanism to guarantee precise sorting on hybrid machines

- Experimental results show that approximate storage can improve performance by 1.1x

- Q&A