ApproxPIM: Exploiting Realistic 3D-stacked DRAM for Energy-Efficient Processing In-memory

Yibin Tang, Ying Wang, Huawei Li, Xiaowei Li

State Key Laboratory of Computer Architecture

Institute of Computing Technology

Chinese Academy of Science

Jan 17th, 2017

Executive Summary

- Observation: memory wall has become a bottleneck of computer systems
- Problem: existing solutions have practical problems
- Solution: approxPIM

  off-the-shelf 3D-stacked DRAM + approximate computing

Hybrid Memory Cube CONSORTIUM

- Results:

  — improves performance more than 30%, while reducing energy consumption in about 13%

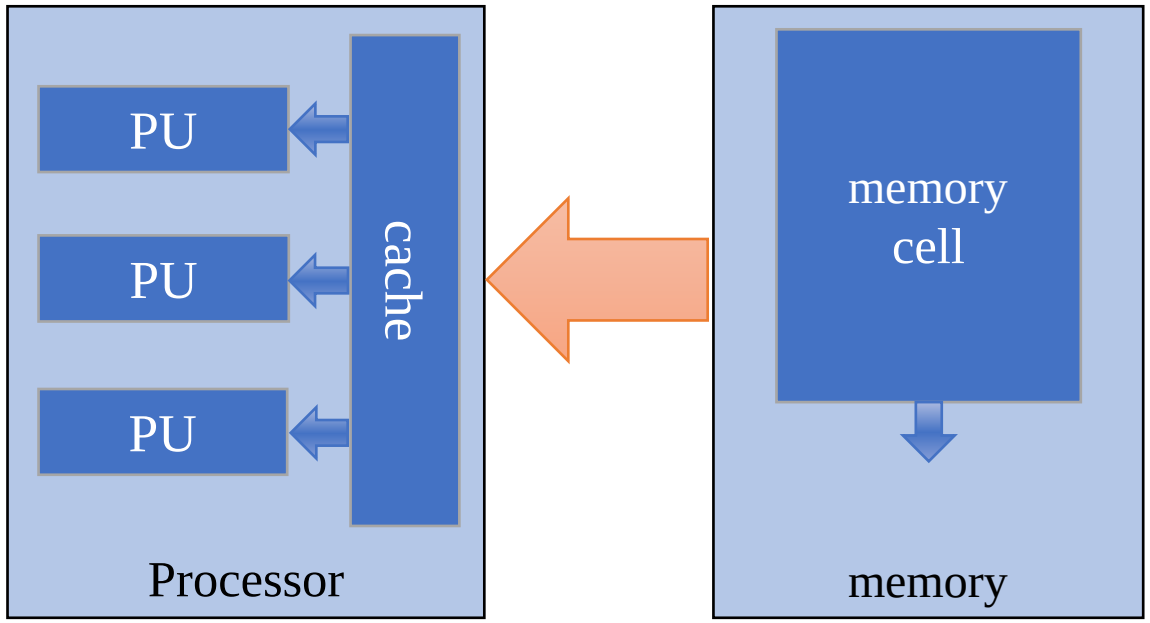
  — more efficient as more aggressive approximate computing strategy is processing

CONTENTS

Processing in-Memory

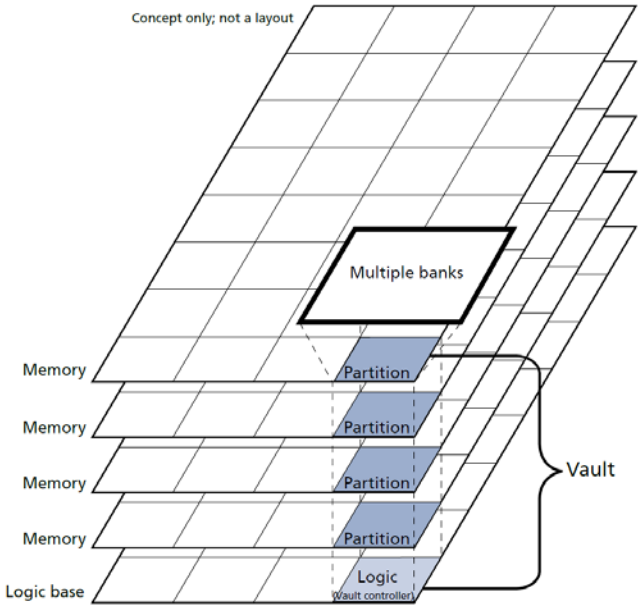
- motivation: address the aggravating memory wall issue

- methods: integrate logics and memory into a single chip
  - application-specific accelerator, GPU, general-purpose multi-processors



Processing in-Memory

- novel 3D-stacked DRAM has been proposed
- integrate basic logic resource into 3D-stack DRAM
  — vault controller and limited computation power
- possibly good start to deploy PiM!



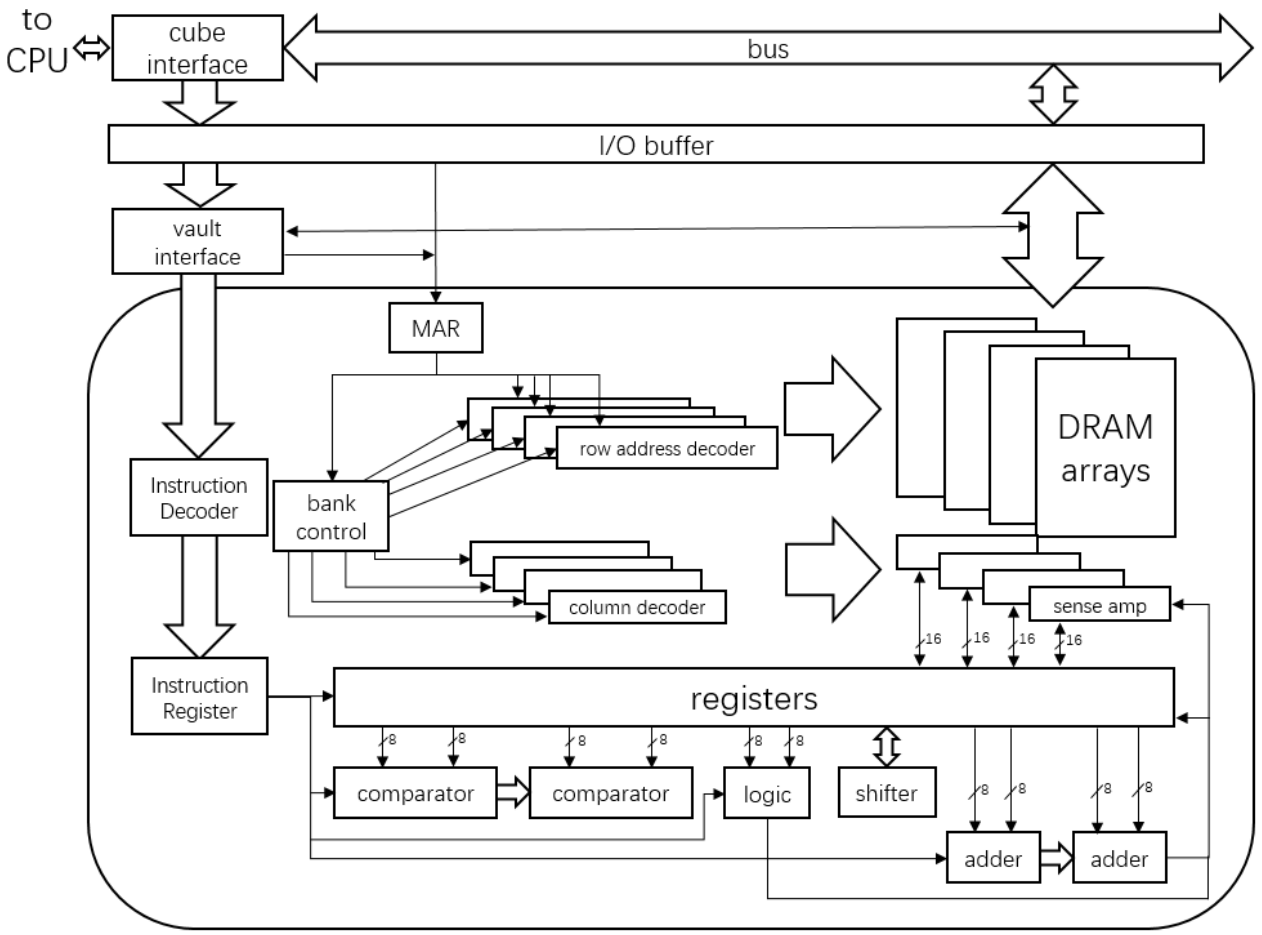
Hybrid Memory Cube
CONSORTIUM

1 Motivation

- existing solutions with PIM
  - ➢using third-party DRAM wafers to design 3D memory with PIM feature?
  - ➢extending computation command + standard DDRx protocol?
- the logic atomic set offered by the HMC is powerless for most applications
- combine PIM with approximate computing techniques

approximate PIM architecture

- approximate PIM architecture



- computing instruction in HMC

Arithmetic	Bitwise	Boolean	Comparison
2ADD8: Dual 8-byte signed add immediate	SWAP16: 16-byte swap	AND16: 16-byte AND	CASEQ8: 8-byte CAS if equal
2ADD8SR: Dual 8-byte signed add immediate with return	BWR: 8-byte bit write	NAND16: 16-byte NAND	CASZERO16: 16-byte CAS if zero
ADD16: Single 16-byte signed add immediate	BWR8R: 8-byte bit write with return	OR16: 16-byte OR	CASGT8/16: CAS if greater than
ADD16SR: Single 16-byte signed add immediate with return	-	NOR16: 16-byte NOR	CASLT8/16: CAS if less than
INC8: 8-byte increment	-	XOR16: 16-byte XOR	EQ8/16: Compare if equal

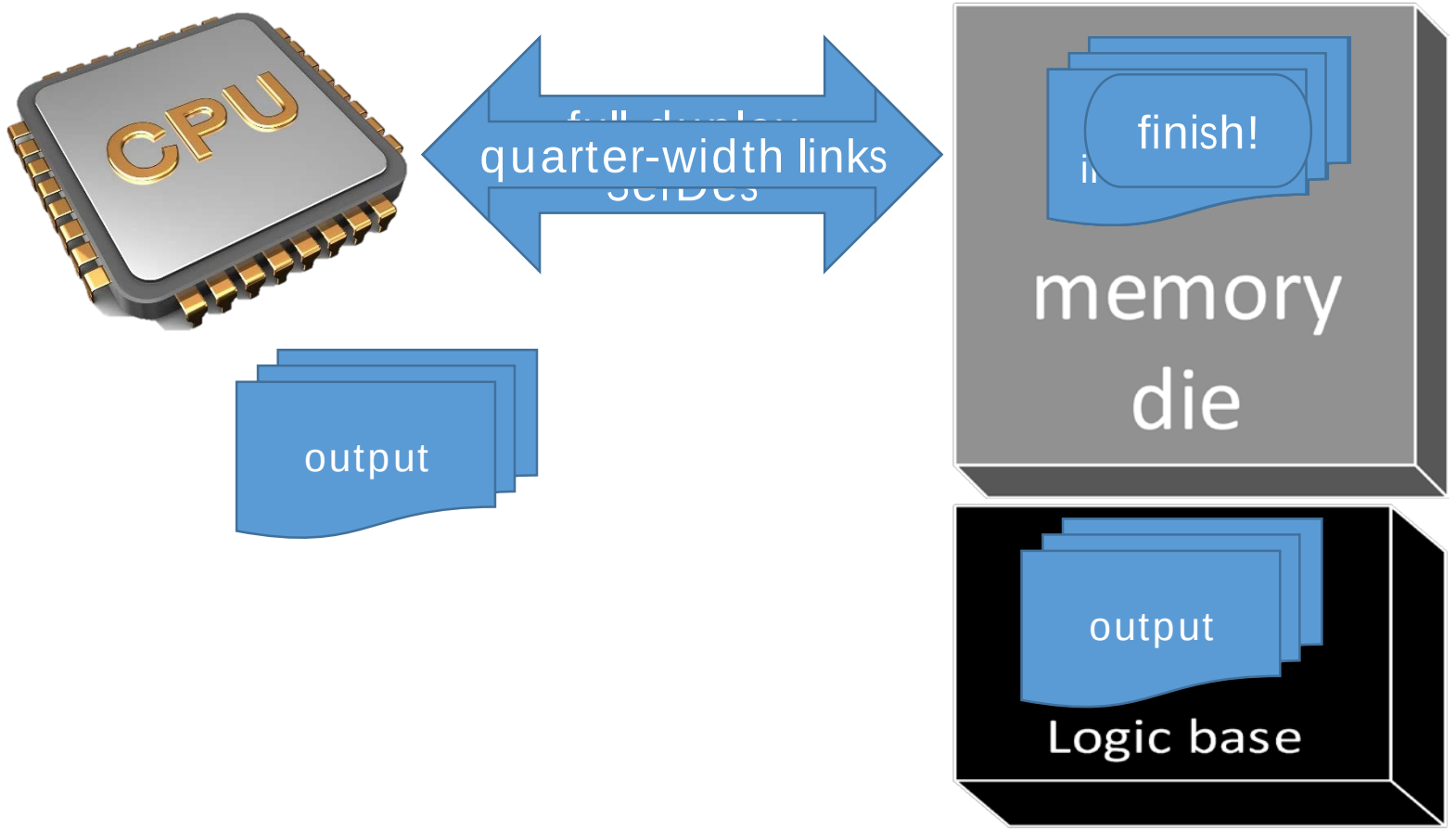
- advanced computing instruction

```
1  decompose multiplier into ∑2^{x_i}, (∀i, x_i > x_{i+1})
2  while (multiplier != 0 ) {
3        if (x_i > 0)
4              product += shift multiplicand x_i times to the left
5        else product += shift multiplicand x_i times to the right;
6        multiplier -= 2^{x_i};
7        i++;
8  }
```

```
1  find an i makes 2^i × divisor < dividend < 2^{i+1} × divisor
2  while (dividend ! = 0) {
3        if dividend ≥ divisor {
4              if i > 0
5                    consult += shift 1 i times to the left
6              else consult += shift 1 i times to the right;
7              dividend -= 2^i × divisor;
8        }
9        i--;
10 }
```

approximate PIM architecture

- normal model



- case study

workloads	approximate computing	most frequent instructions	HMC instruction
BFS	-	AND	AND16
BC	sample and predict	AND & XOR	AND16 & XOR16
BS	-	compare & exchange	CASLT8
SS	sample and predict	compare	CASEQ8
K-means	operation & algorithm	mul & add	2ADD8 & shifting
KNN	operation & algorithm level	mul & div &add	2ADD8 & shifting
NF	operation & algorithm-level	mul & add	2ADD8 & shifting

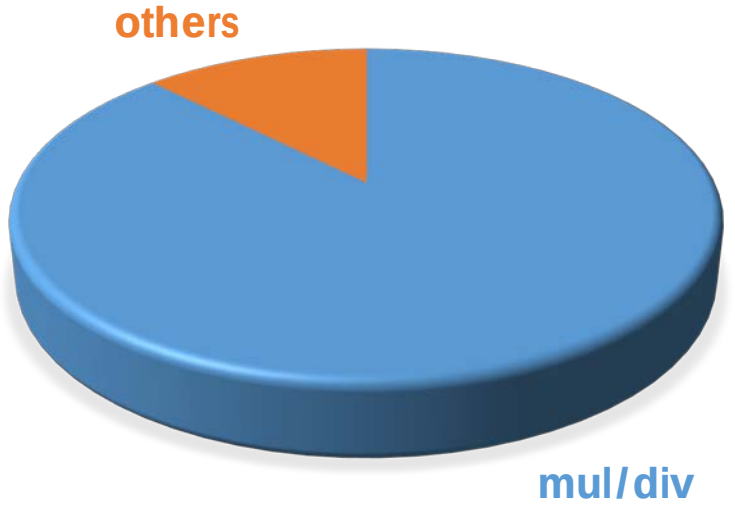
approximate PIM architecture

- large cost while processing computation-intensive applications!

- approximate computing

➢architecture-level approximate computing

— tunable parameter for multiplication/division



➢algorithm-level approximate computing

➢input-level approximate computing

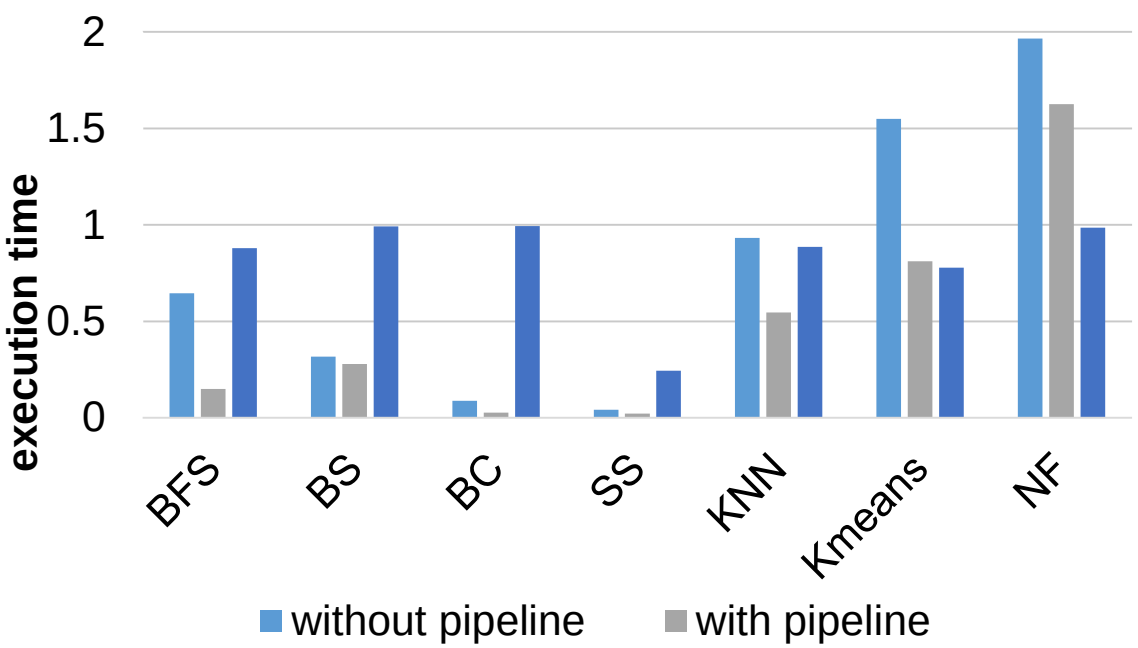
— sampling-estimation

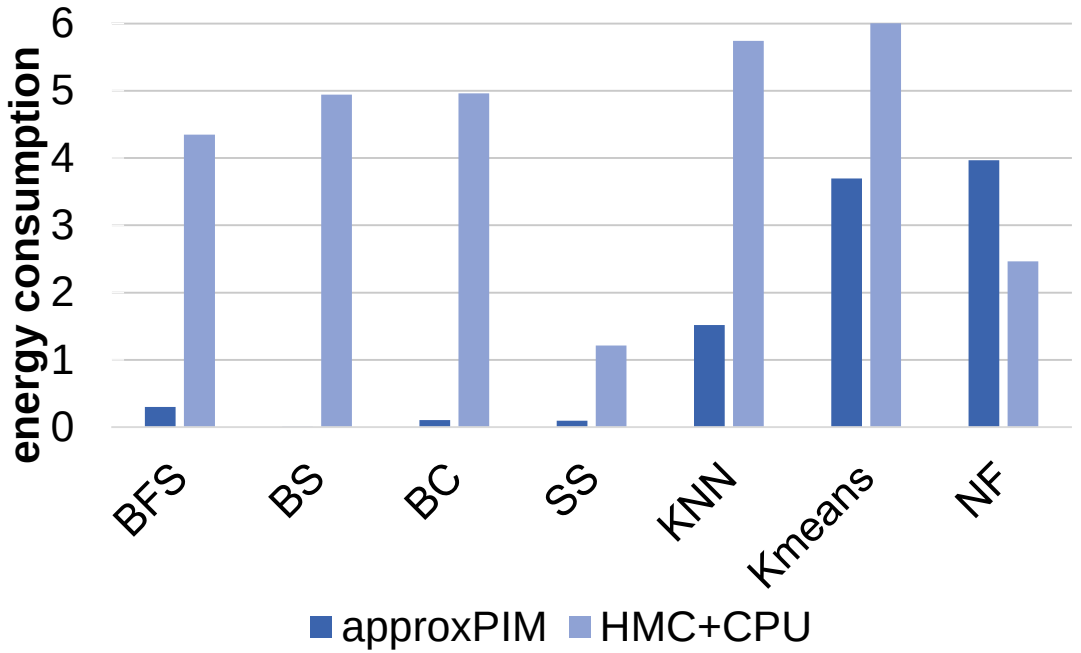
approximate PIM architecture

- platform and configuration
- multi2sim, Catti-3DD and HMC-Sim simulator for performance
- McPAT simulator for energy efficiency

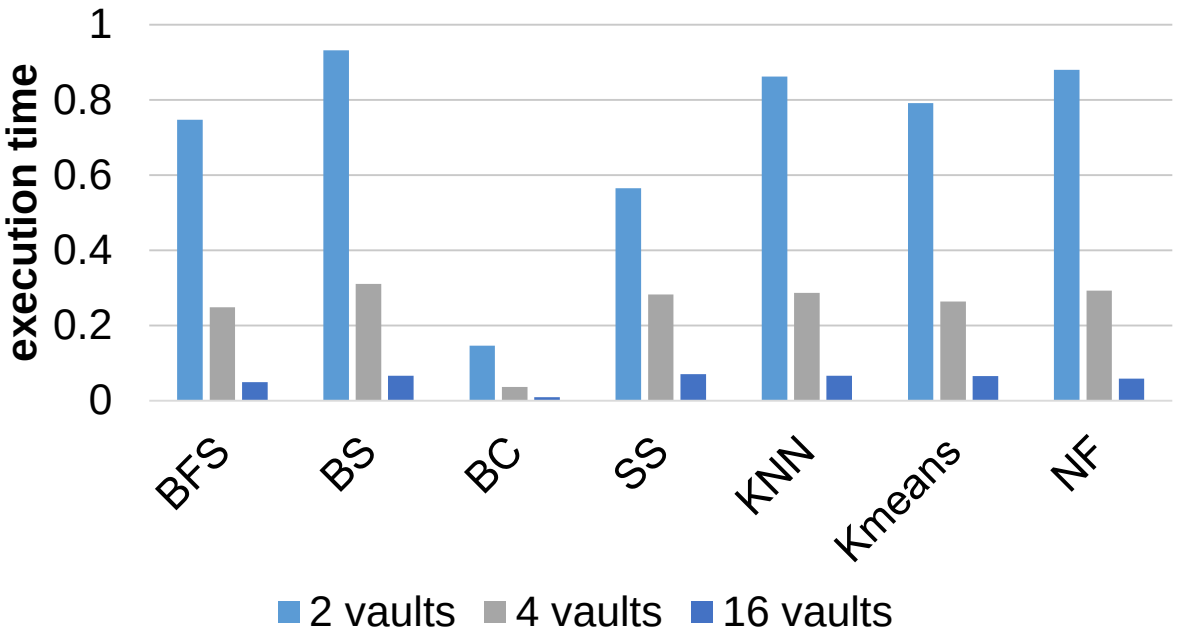
Component	Parameters
Host Processor	Process Technology: 45nm, Frequency: 2GHz, Cores: 4, OoO issue: 4 operations/cycle, Re-order Buffer Size: 64, Threads per Core: 2, Supply Voltage: 0.75V, L1 Cache: 64KB, L2 Cache: 2MB, shared
Memory	Frequency: 1333MHz, Page Size: 4KB, Memory Size: 4GB, Number of Banks: 8, Burst Length: 4, Protocol: DDR3, Channel: 1, Rank: 2
HMC	50nm, 1 cube, 4-layered Memory Dies: Page Size: 4KB, Memory Size: 4GB, Number of Banks: 256, Vault: 32, Bus Width: 128 bit Logic Dies: Process Technology: 50nm, Frequency: 0.67GHz, Supply Voltage: 0.9V,

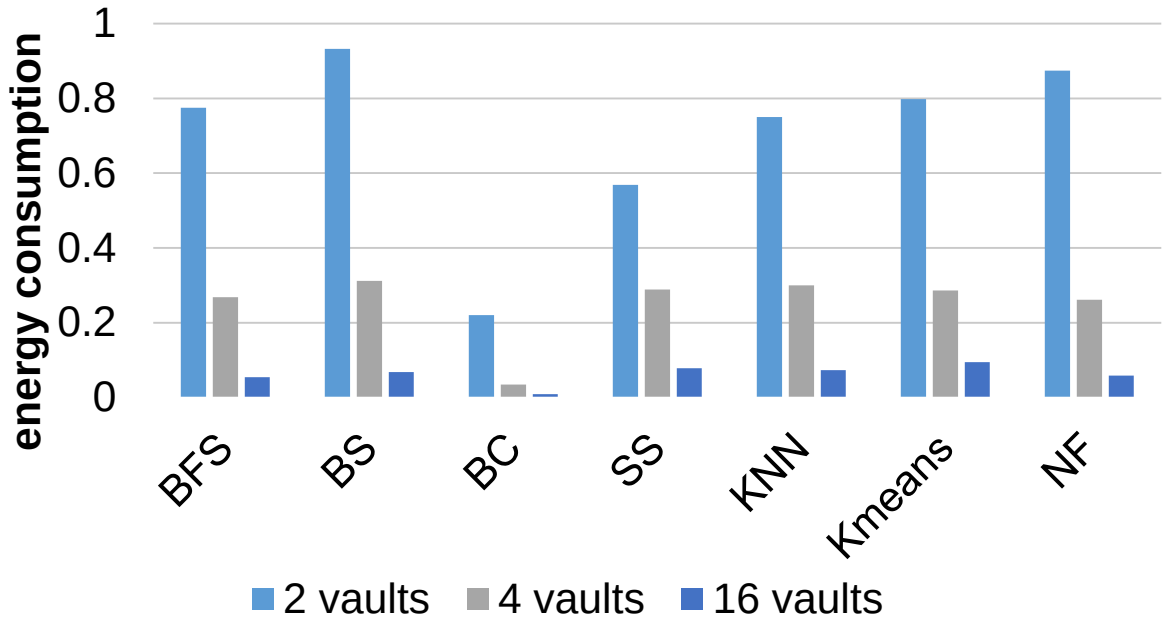
- sequential execution
  - ➢ 21% speedup over the baseline
  - ➢ 14% energy consumption (non-computation workload)





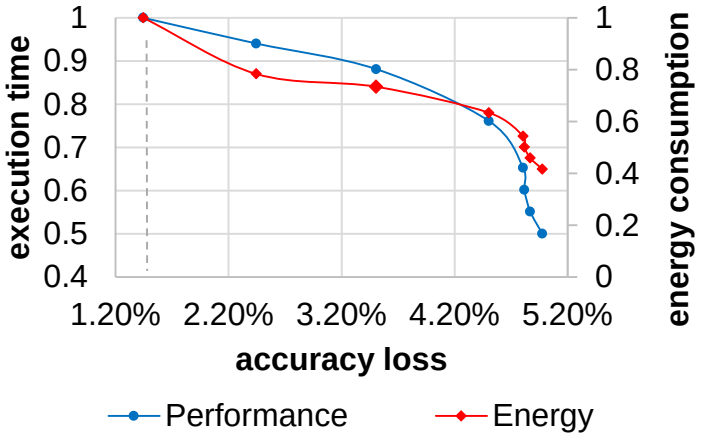
- parallel execution
  - ➢ more than 30% performance improvement
  - ➢ more than 25% energy reduction compare to the sequential situation



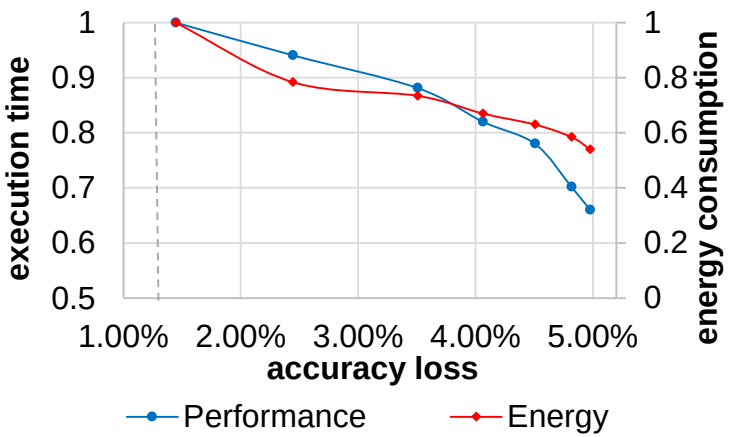


- architecture-level approximate computing
- algorithm-level approximate computing

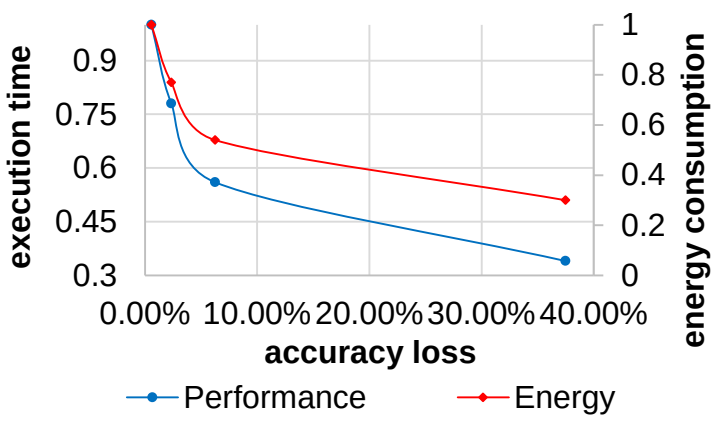
- 20% speedup and 40% energy reduction while 5% inaccuracy loss



K-means



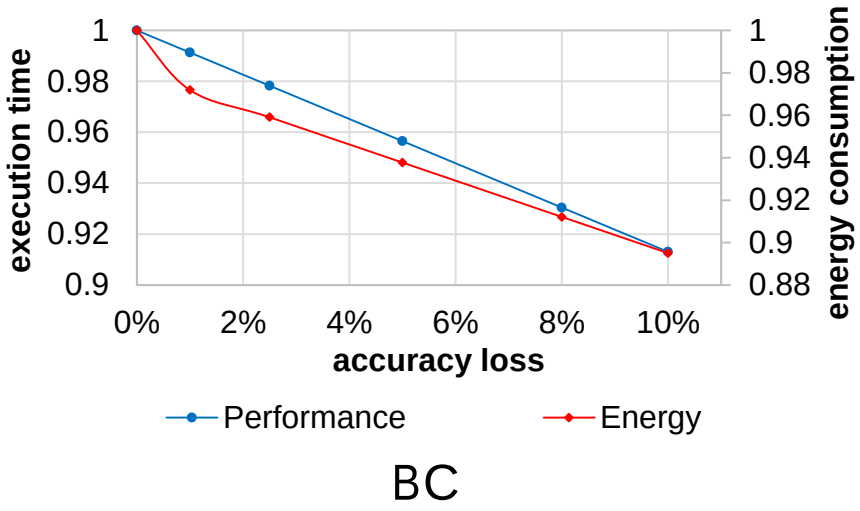
KNN



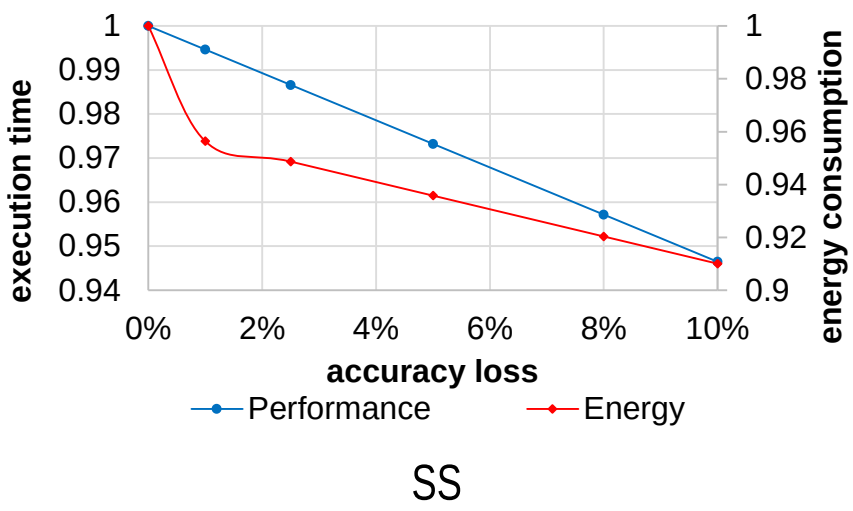
NF

- input-level approximate computing

- 5%~8% speedup and 10% energy reduction while 10% inaccuracy loss



BC



SS

4 Summary

- Problem:
  - — memory wall is a key point of computer systems
  - — existing PIM solutions have practical problems
- Solution:
  - — PIM based on off-the-shelf 3D-stacked DRAM
  - — approximate computing approach to improve its performance
- Results
  - — improves performance more than 30%, while reducing energy consumption about 13%
  - — more efficient as more aggressive approximate computing strategy is processing

THANKS