From Processing-in-Memory to Processing-in-Storage (PRinS)

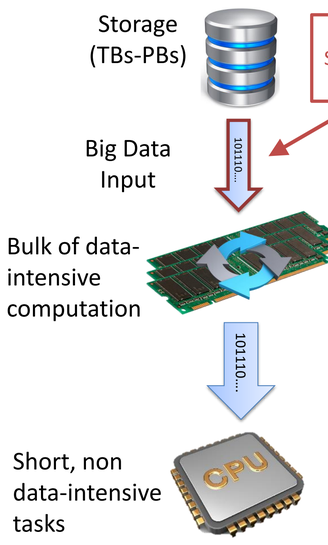
Roman Kaplan

Advisor: Prof. Ran Ginosar — In collaboration with: Dr. Leonid Yavits

Technion Israel Institute of Technology

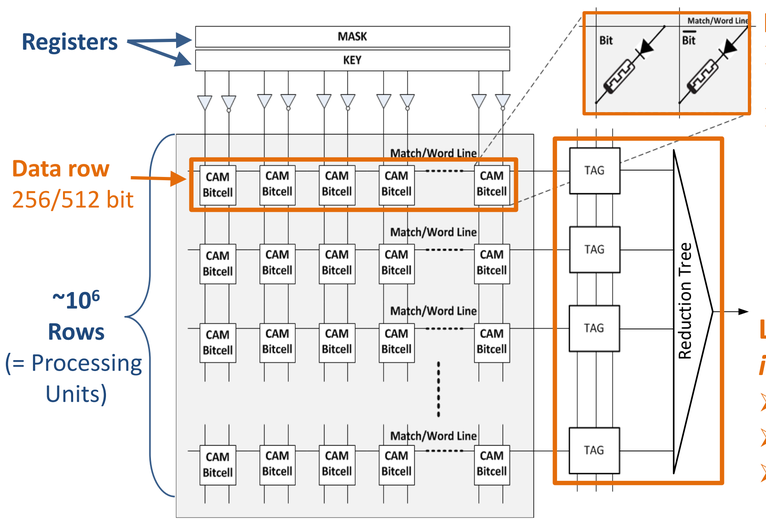
Problem and Solution Overview

Storage ⇔ Memory Bottleneck Problem



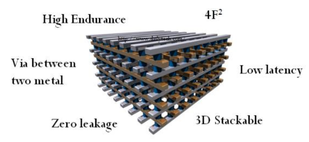
- PiM performs the bulk of computation in main memory
  - ✓ Reduces Mem→CPU transfers
- **Storage → Mem data transfers are the main bottleneck**
- Processing in-storage can increase performance and reduce energy by 100×

Resistive Content Addressable Memory (ReCAM): A PRinS Device



Resistive Bitcells

- Based on resistive elements (e.g., memristors)
- High density & non-volatile

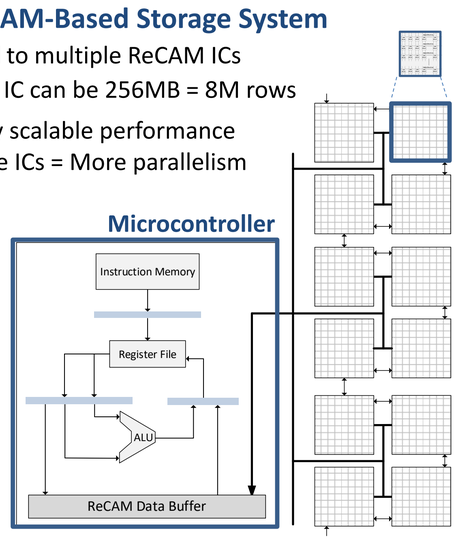


Logic next to data: *in-situ* processing

- Processing unit (PU) per row
- All PUs can operate in parallel
- **Instruction are based on bit-serial associative processing**

Full ReCAM-Based Storage System

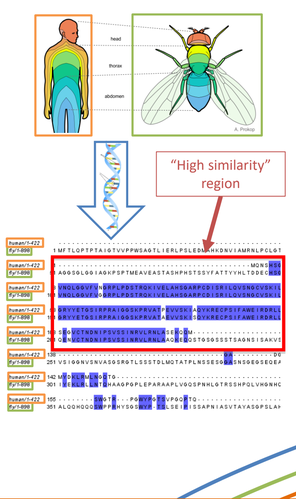
- Divided to multiple ReCAM ICs
  - ✓ E.g., IC can be 256MB = 8M rows
- Linearly scalable performance
  - ➔More ICs = More parallelism



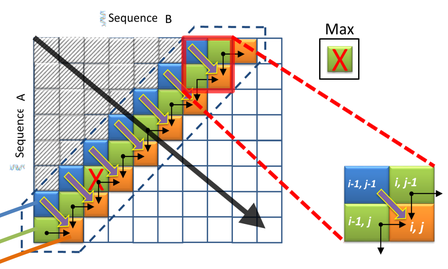
PRinS Application
DNA Local Sequence Alignment

Smith-Waterman Algorithm

- Finds regions with "highest similarity" between two sequences (DNA or protein)
  - ✓ Proven to be optimal
- Every "match", "mismatch" & "gap" has a score
- S-W is based on dynamic programming
  - ✓ Fills a $m \times n$ matrix
  - ✓ Has a quadratic computational complexity $O(m \cdot n)$
- We focus on the computationally-heavy scoring step



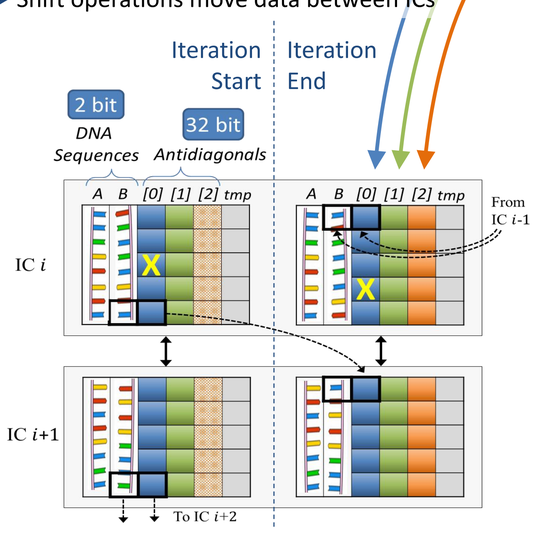
- Matrix-fill order is on the main diagonal
- Entire antidiagonal is calculated in parallel
  - ✓ Antidiagonal per group of ReCAM columns
- Search for the maximal score
- Only 3 antidiagonals are required in each iteration



PRinS Implementation & Performance Comparison

In-Storage Computation

- IC = Integrated Circuit
- Shift operations move data between ICs



Performance Comparison

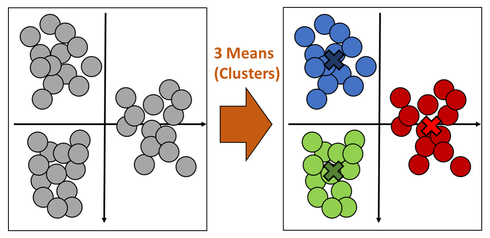
- Cycle-accurate simulator: 8GB of storage running at 500MHz
- Compared to multi-accelerator state-of-the-art solutions: FPGA, Xeon Phi and GPU
- ReCAM shows **4.7×** higher performance than a 384-GPU solution

Accelerator	FPGA	Xeon Phi	GPU	ReCAM
Performance (TCUPS)	6.02	0.23	11.08	**52.68**
# of ICs	128	4	**384**	32
Publication	L.Wienbrandt. **The FPGA-based High-Performance Computer RIVYERA for Applications in Bioinformatics**. In *Language, Life, Limits: 10th CiE*, pages 383-392. Springer 2014.	Y. Liu and B. Schmidt. 2014b. **SWAPHI: Smith-Waterman protein database search on Xeon Phi coprocessors**. In *IEEE ASAP*, 184–185.	Sandes, E. F. O., et al. **"CUDAlign 4.0: Incremental Speculative Traceback for Exact Chromosome-Wide Alignment in GPU Clusters."** IEEE	

PRinS Application
K-Means Clustering

K-Means Clustering Algorithm

- Common Machine Learning algorithm
- For each sample, finds its group among possible K
- Widely used in many fields, including:
  - ✓ Image processing
  - ✓ Anomaly detection
- Data intensitve
  - ➢ Multiple iterations over entire dataset



Two Main Loops

1. For each Sample:
   For each Center:
   Calculate |Sample – Center|
2. For each Center:
   For each Sample:
   Calculate Sum(Sample)

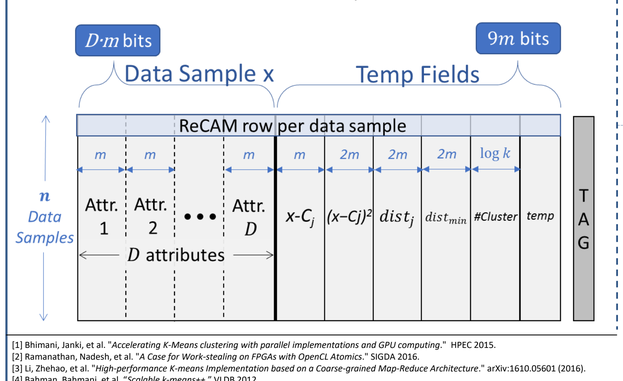
$O(n)$ in a von Neumann machine

$O(1)$ in ReCAM

PRinS Implementation & Performance Comparison

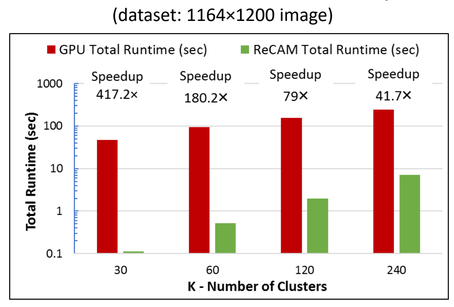
In-Storage Computation

- In large datasets: #Clusters ≪ #Data Samples
- Computation over **ALL** samples in one instruction
- Each sample attributes requires multiple temp fields
  - Difference between center's attribute
  - Squared difference
  - Total distance between sample and center



Comparison to GPU

GPU[1] and ReCAM Total Runtime Comparison
(dataset: 1164×1200 image)



Comparison to Other Works

Work Ref.	Dataset: Samples	Dataset: Size	Dataset: Clusters	Platform	Ref. iteration runtime (msec)	ReCAM iteration runtime (msec)	ReCAM Speedup
[2]	1M	32MB	128	FPGA	21.8	4.7	4.6×
[3]	2M	250MB	4	Map-Reduce	8.5	0.56	15×
[4]	4.8M	700MB	100	Map-Reduce	1.54	$12 \cdot 10^{-3}$	120.7×
[5]	2.5M	345MB	64	Intel i7	2.17	0.04	53.3×
			10000		432.9	6.4	67.8×

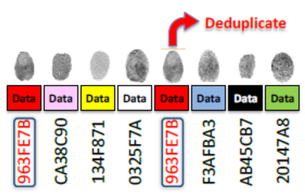
[1] Bhimani, Janki, et al. "*Accelerating K-Means clustering with parallel implementations and GPU computing.*" HPEC 2015.
[2] Ramanathan, Nadesh, et al. "*A Case for Work-stealing on FPGAs with OpenCL Atomics.*" SIGDA 2016.
[3] Li, Zhehao, et al. "*High-performance K-means Implementation based on a Coarse-grained Map-Reduce Architecture.*" arXiv:1610.05601 (2016).
[4] Bahman. Bahmani, et al. "*Scalable k-means++.*" VLDB 2012.
[5] Ding, Yufei, et al. "*Yinyang k-means: A drop-in replacement of the classic k-means with consistent speedup.*" ICML 2015.

PRinS Application: **In-Storage Deduplication**

What is Deduplication?

- Deduplication is a technique for storing a single copy of each data block in storage
  - ✓ Can reach 10x reduction in data volume
- How it works:
  1. Data is broken into fixed blocks

  2. A fingerprint (FP) is calculated for each block
  3. Only pointers are stored for identical blocks

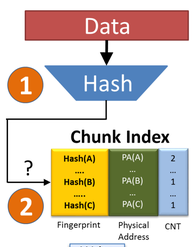


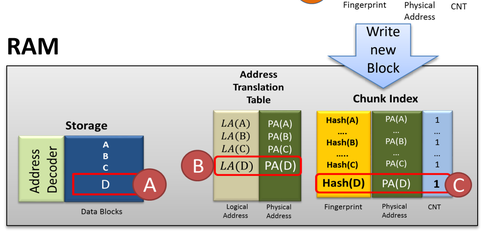
Traditional Systems vs. ReCAM

Traditional (RAM+CPU) Systems

New block write:

1. Hash (create key)
2. Search in key table
3. Write to three tables in RAM (A B & C)

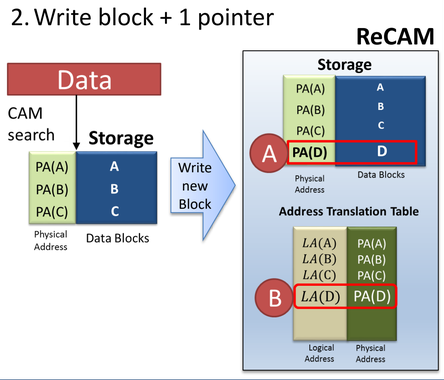




In-ReCAM Deduplication

Use CAM operations:

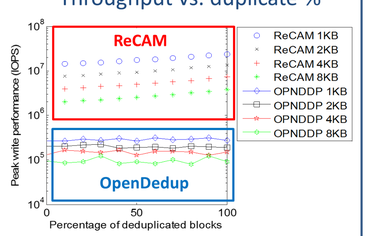
1. CAM search → No need to hash
2. Write block + 1 pointer



Performance Evaluations

- ReCAM was simulated with a cycle-accurate simulator
- ReCAM Parameters: 256GB @1GHz
- Opendedup executed on high-end server: 4×8 octa-core CPU, 64GB RAM, 800GB SSD drive
- **ReCAM has 100× higher throughput than deduplication with RAM+CPU**
- Energy consumption is similar or lower for the common block sizes (4 & 8KB)

Throughput vs. duplicate %



Energy vs. duplicate %