Low Latency, High Throughput Similarity Search with an In-Memory Associative Processor

Dr. Avidan Akerib, VP Associative Computing BU

June 2nd 2019

GSI TECHNOLOGY

In-Memory Computing SUMMIT EUROPE 2019

Agenda

01 About GSI

02 Big Data Similarity Search

03 What Is Associative Computing

04 Architecture

05 K-nearest Neighbors For Big Data

06 Big Data Classification

07 SW Tools

08 Use Case Examples

About GSI

CORPORATE SUMMARY

1 **FOUNDED IN 1995**

2 **PUBLIC COMPANY**
Consistent profitability & zero debt

3 **~150 EMPLOYEES WORLDWIDE.**
Design / R&D in Sunnyvale, CA & Israel; Operations in Taiwan

APU

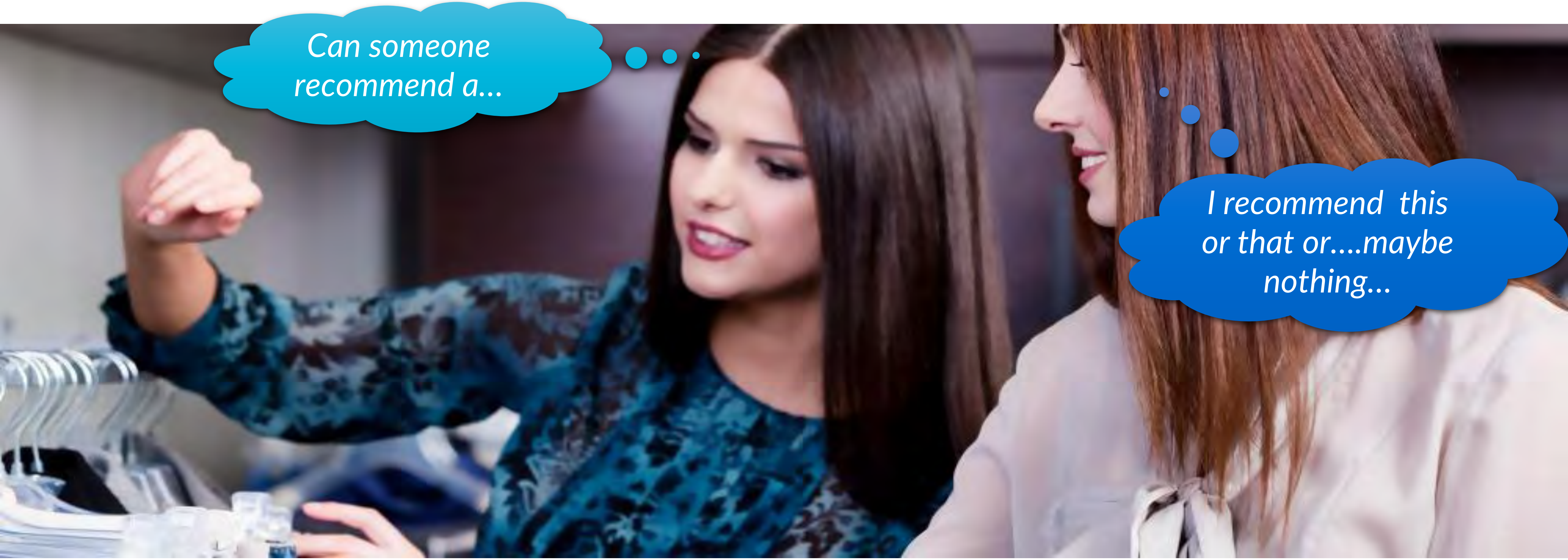
Developed the APU, Massively Parallel Processor for big data similarity search, based on Computational Memory technology.

HIGH PERFORMANCE

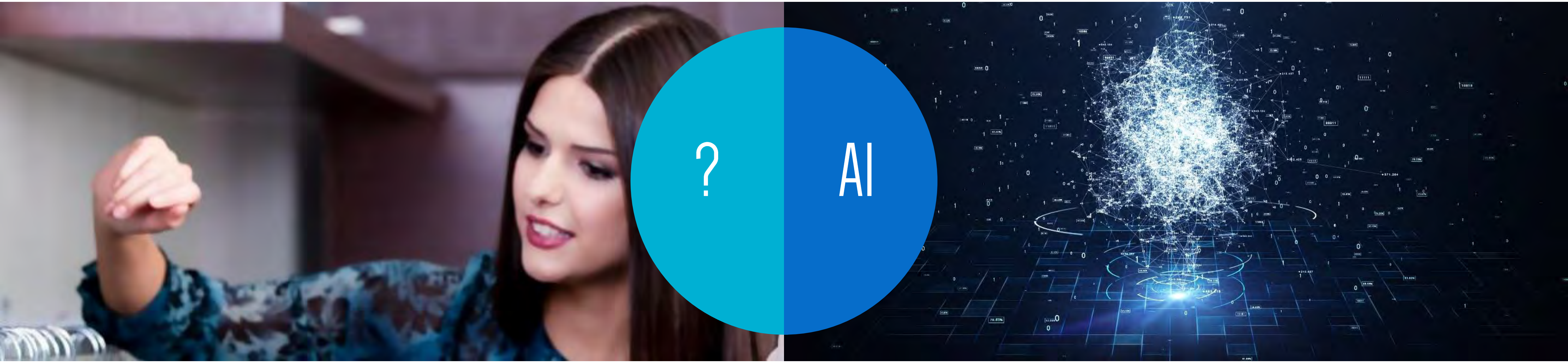
Leader in supplying high performance memories to demanding industries such as aerospace, defense and high performance datacenters.
Acqu MikaMonu and its Associative Computing IP in 2015.

Big Data
Similarity Search

Once Upon a Time There Was a Fashion Store...



Today's Trend



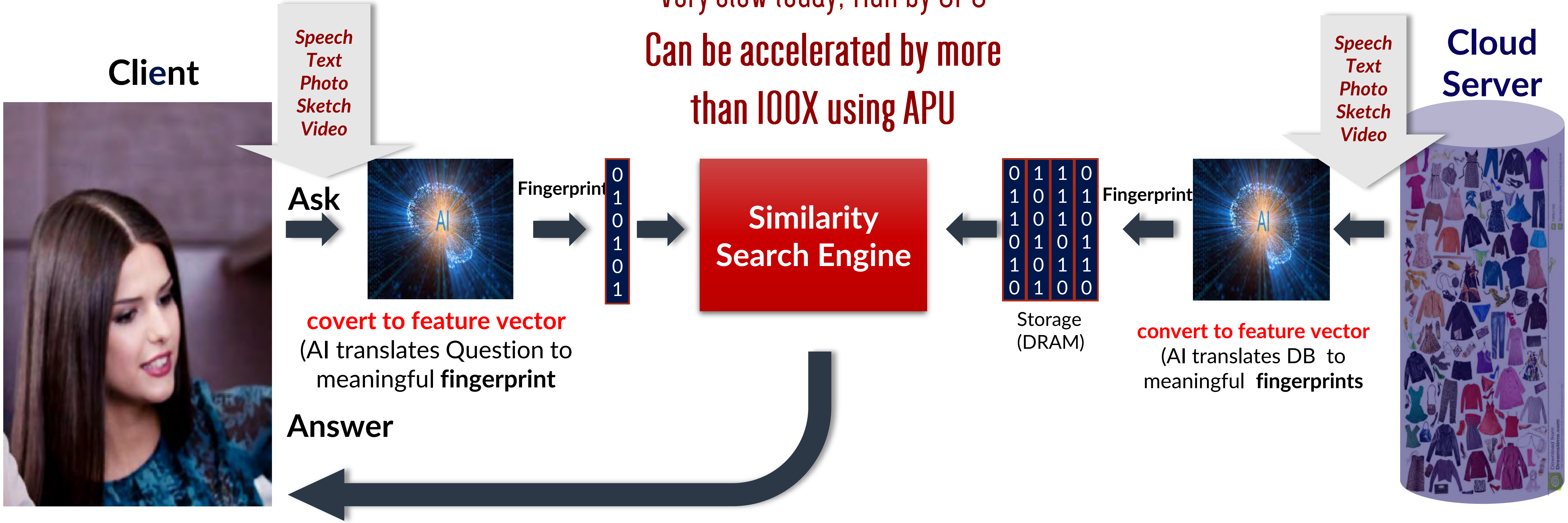
For doing that, Machine Learning is not enough.

LETS UNDERSTAND THE CONCEPT FIRST

The Concept

Very slow today, Run by CPU

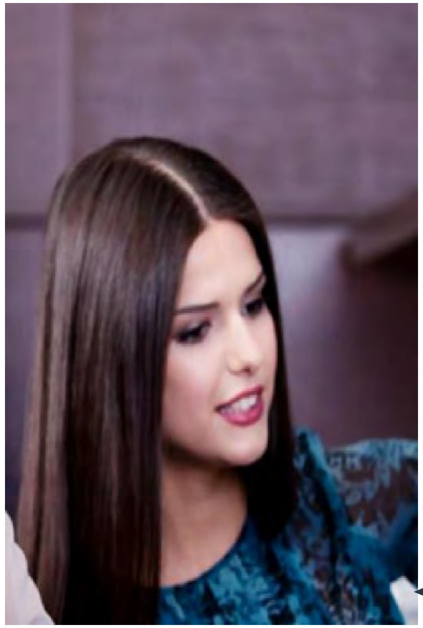
Can be accelerated by more than 100X using APU



The Concept

In-Memory AI Server

Client



Ask

ON LINE COMPUTING

APU
Associative Computing Unit

OFF LINE COMPUTING

DRAM

Embedding

CPU

In-Memory
Storage
< 1 TByte

convert to
feature vector

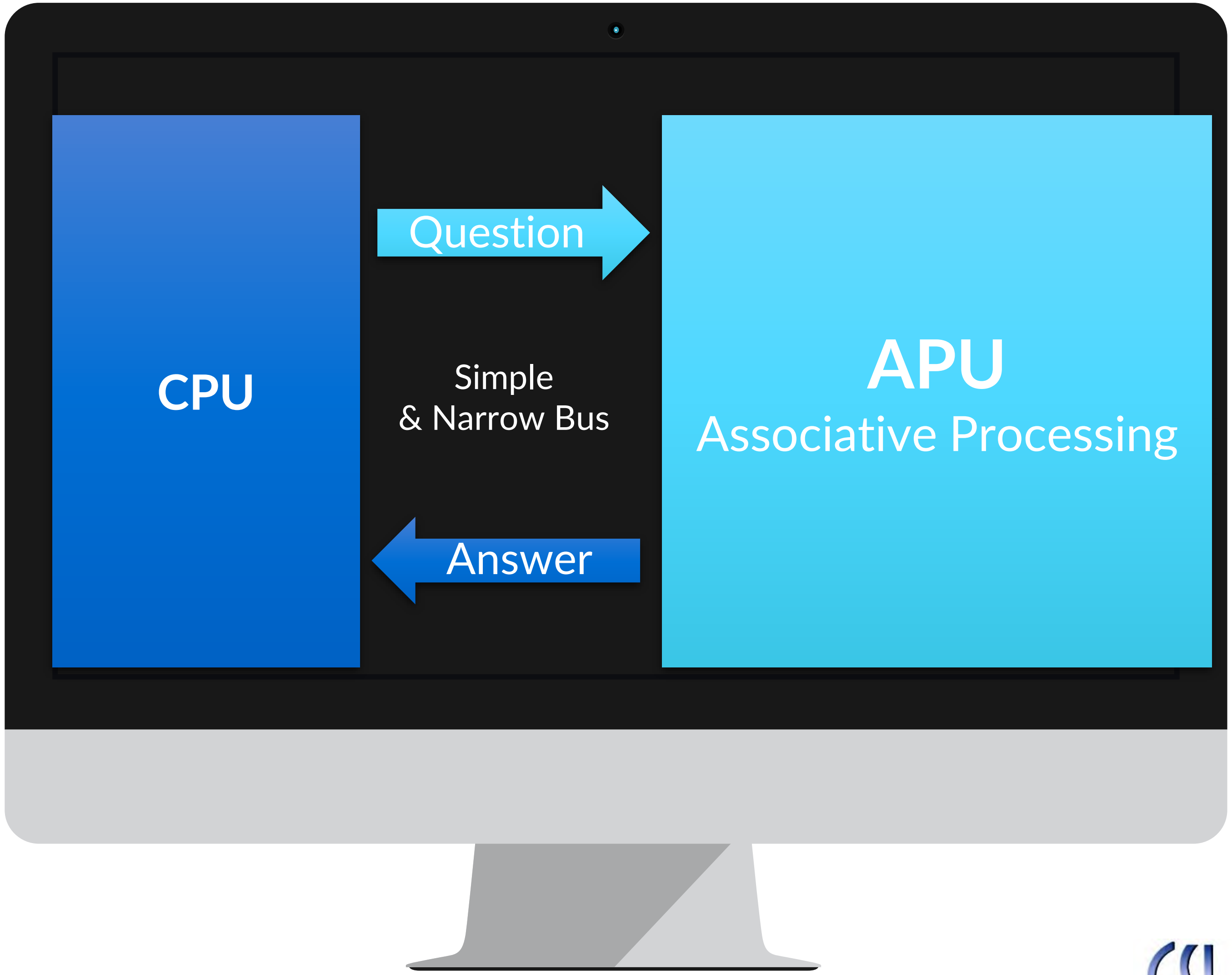
Cloud Server

Answer

The Associative Processing Unit (APU)

Computes in-place, directly in the memory array, removing the I/O bottleneck

- Significantly increases performance
- Reduces power consumption
- Data compression (Binary Reduction)
- Query parallelism for production system



FOR TODAY'S DEMANDING WORLD WE CAN'T RELY ON CPU AND GPGPU ALONE

Associative Computing fundamentals

STORAGE MUST BE MORE "INTELLIGENT"

The current state is that storage simply holds the data. The need for intelligent cache that preprocesses for the main processor (CPU or GPGPU) tedious tasks and replace the main processor with an associative processor

ESSENTIAL PART

Calculations within the memory unit with lower latency and lower voltage is making it an essential part of any architecture of any datacenter

Similarity Search | Visual Search

CRITICAL COMPONENT ACROSS APPS

Similarity search is a critical component for many applications

- As it becomes common large scale similarity search
- Similarity is in Visual Search, Voice, Text apps
- Across applications in all industries – consumer, bioinformatics, cyber, automotive

*“**The future of online product research: visuals and voice.***

***The rise of voice searches fueled by technology** like Google Home and Amazon’s Alexa has been well-documented.*

***But visual searches are also on the rise.** Products like Pinterest Lens use machine learning to aid in brand and product discovery”*

Our User Experience

WERE EXPERIENCING **SIMILARITY** AND **VISUAL** SEARCH

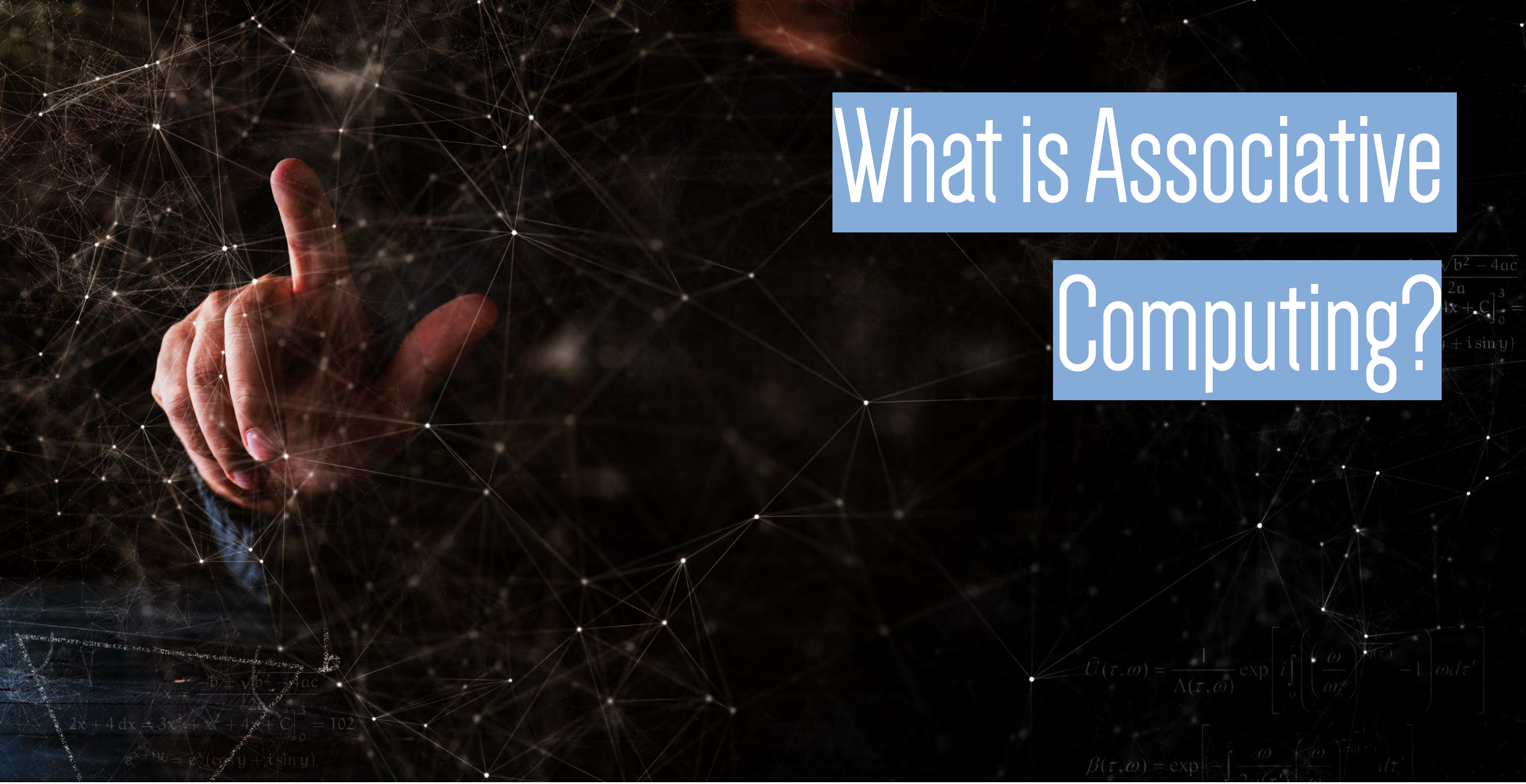
Netflix
Uses similarity search to figure out our taste in TV to retain us by offering personal content

Facebook
Tries to tailor our newsfeed to our interests

Spotify
Builds our playlists according to what we listen to

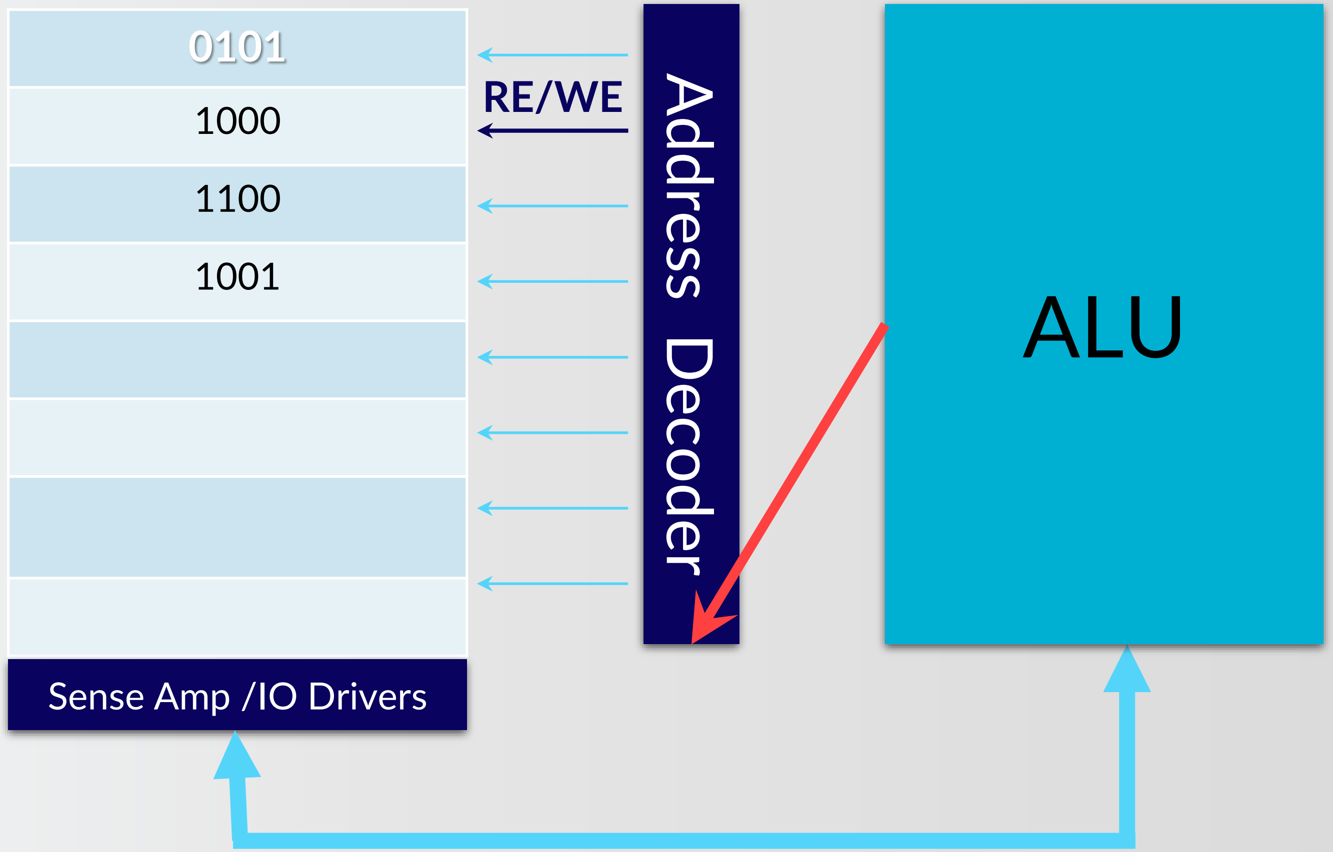
Pinterest
Lets us upload a picture and offer us similar products

Google
Tries to constantly improve its visual search to be more relevant in search results

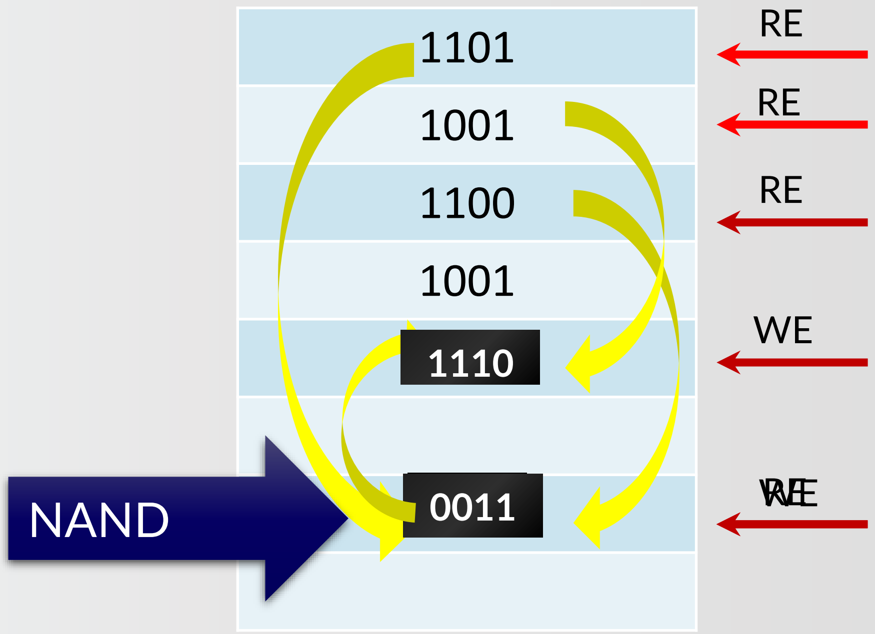


What is Associative Computing?

How Computers Work Today



Lets Look Different
Accessing Multiple Rows Simultaneously



Bus Contention is not an error !!!

It's a simple NOR/NAND satisfying De-Morgan's law

Truth Table Example

A	B	C	D
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

C \ AB	00	01	11	10
0	1	1	0	0
1	0	1	1	0

!A!C + BC =

!!(!A!C + BC) = ! (!(!A!C)!(BC))

= **NAND(NAND(!A,!C),NAND(B,C))**

1 CLOCK

Read (!A,!C) ; WRITE T1

Read (B,C) ; WRITE T2

1 CLOCK

Read (T1,T2) ; WRITE D

- **Every minterm takes one clock**
- **All bit lines execute Karnaugh tables in-parallel**

Vector Add Example

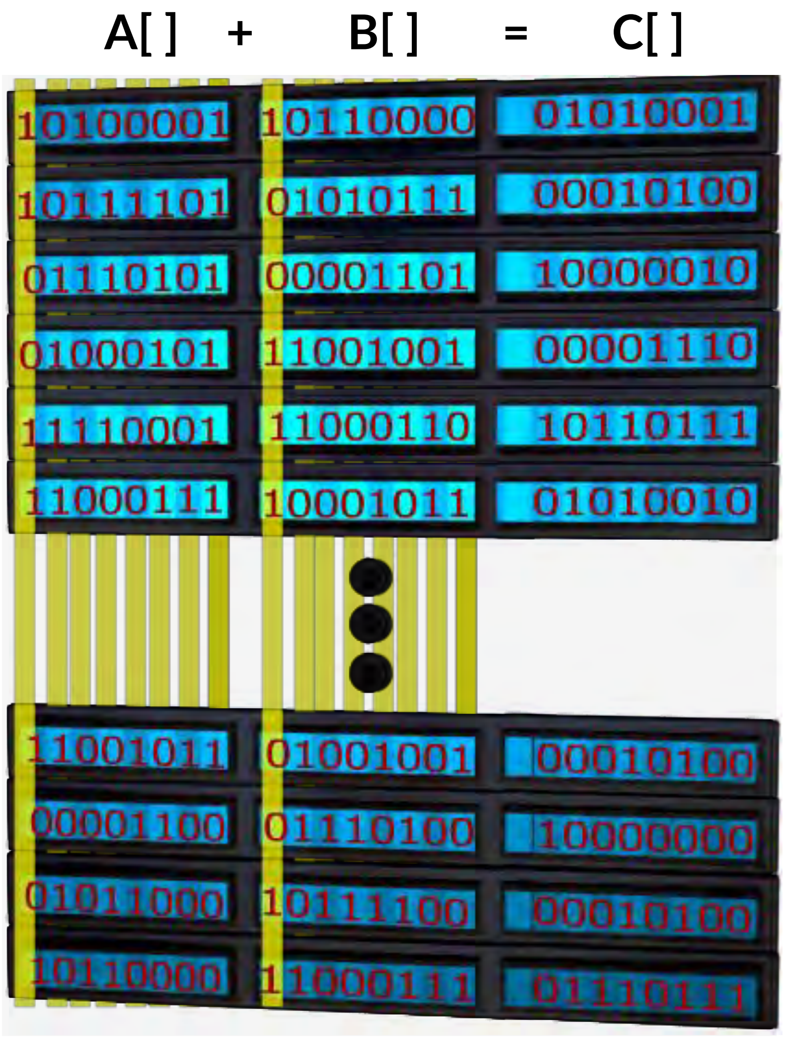
vector A(8,32M)
vector B(8,32M)
Vector C(9,32M)
C = A + B

No. Of Clocks = 4 * 8 = 32

Clocks/byte= 32/32M=1/1M

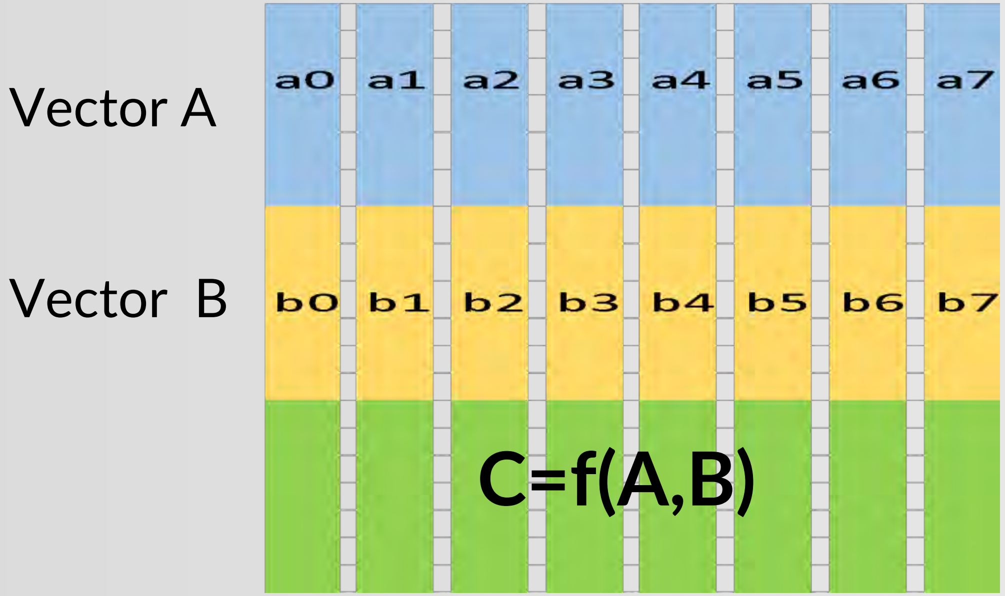
OPS = 1Ghz X 1M

= 1 PetaOPS



Single APU chip has 2M Bit Line Processors – 64 TOPS or >> 2 TOPS/Watt

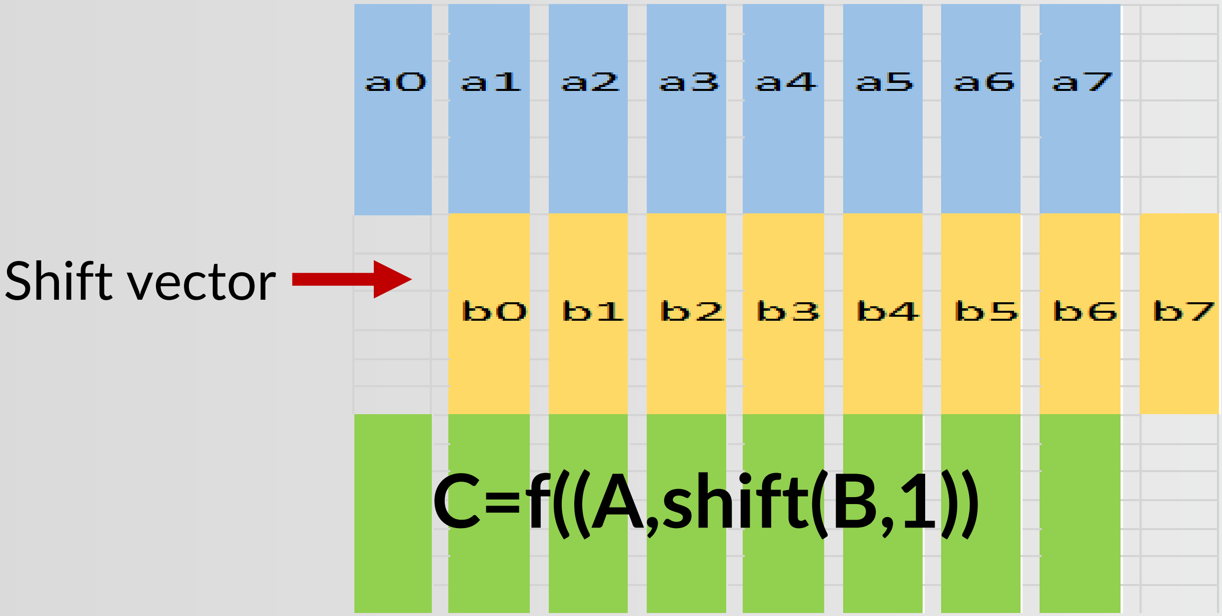
Computing in the Bit Lines



Each bit line becomes a processor and storage

Millions of bit lines = millions of processors

Computing in the Bit Lines



Parallel shift of bit lines @ 1 cycle sections

Enables neighborhood operations such as convolutions

Cosine Similarity Example

Query

4
3
-1
4
0
-1

Associative Memory DB

6	7	-3	-1	4	8	2	8	2	-1
-3	7	4	-2	-1	-1	-2	6	6	-2
0	0	5	-2	2	-3	3	-1	7	4
2	1	0	2	-2	3	-3	2	4	-2
1	1	8	6	4	5	5	1	1	-3
1	6	0	5	1	-1	2	0	-1	0

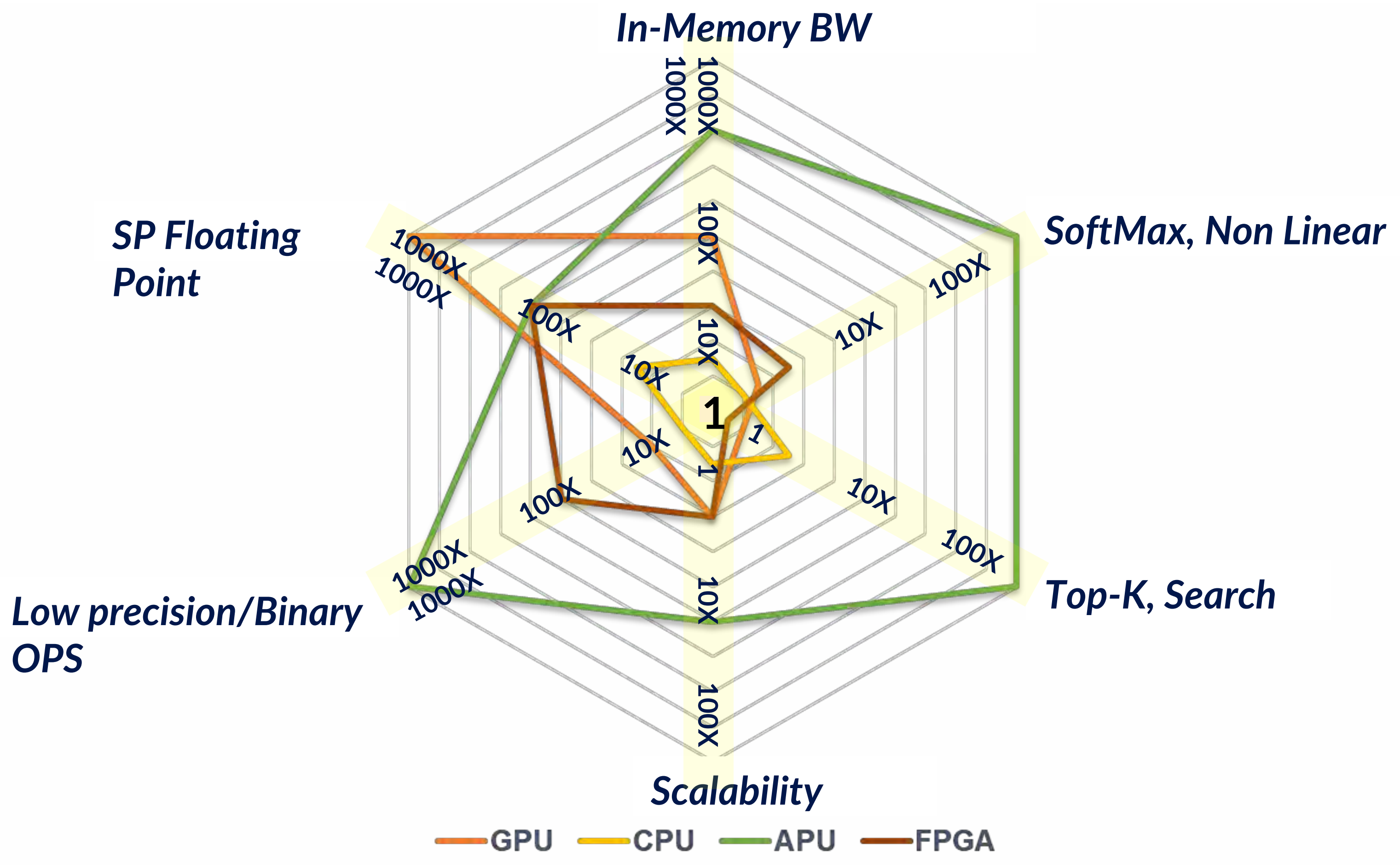
In Memory Compute Cosine Distance

22	47	-5	-5	2	45	-15	59	36	-22

TOP K=3

> 100,000 Quires/sec , any K size, 128K Records, Sigle chip@10Watts

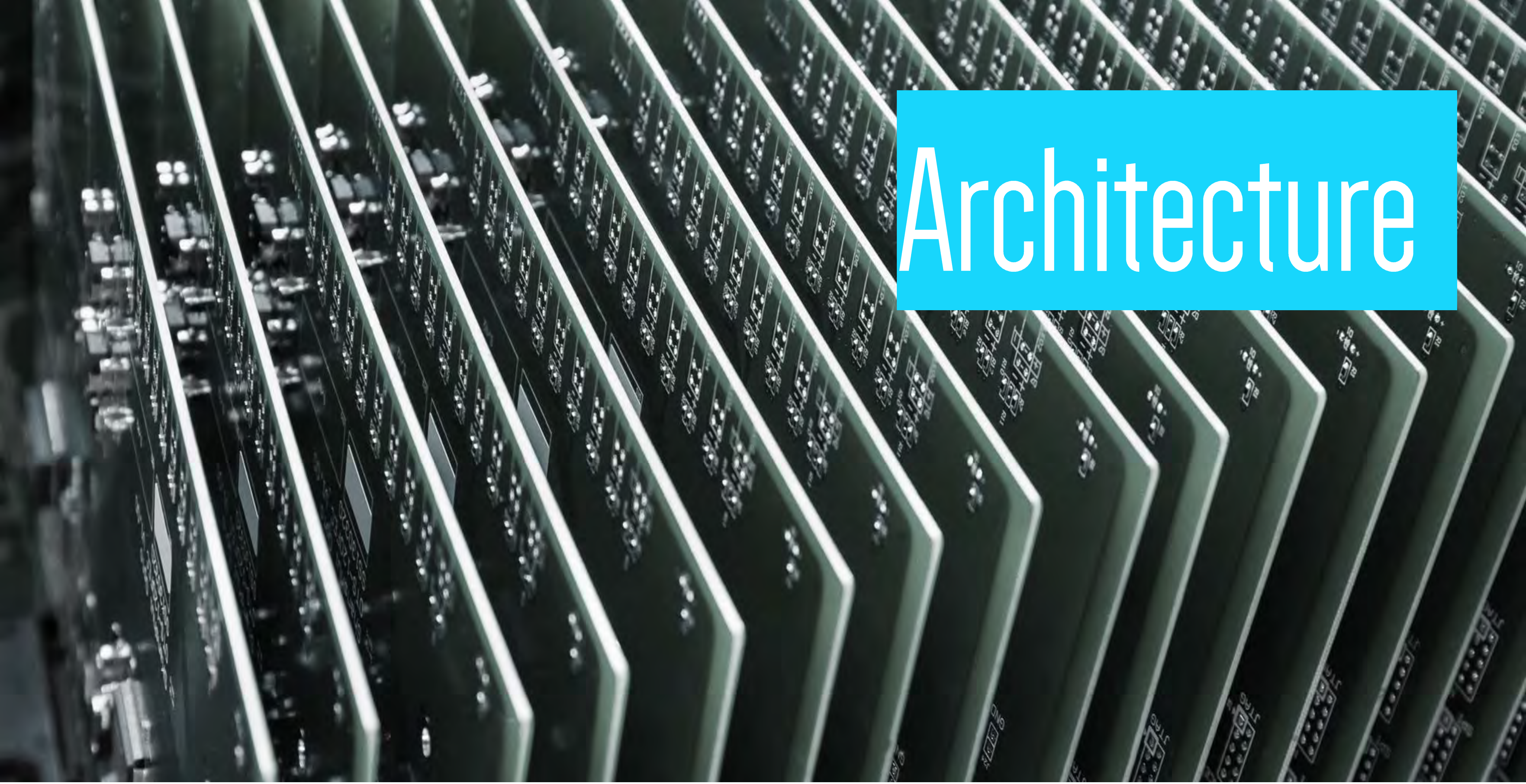
CPU vs GPU vs FPGA vs APU



CPU/GPGPU vs APU

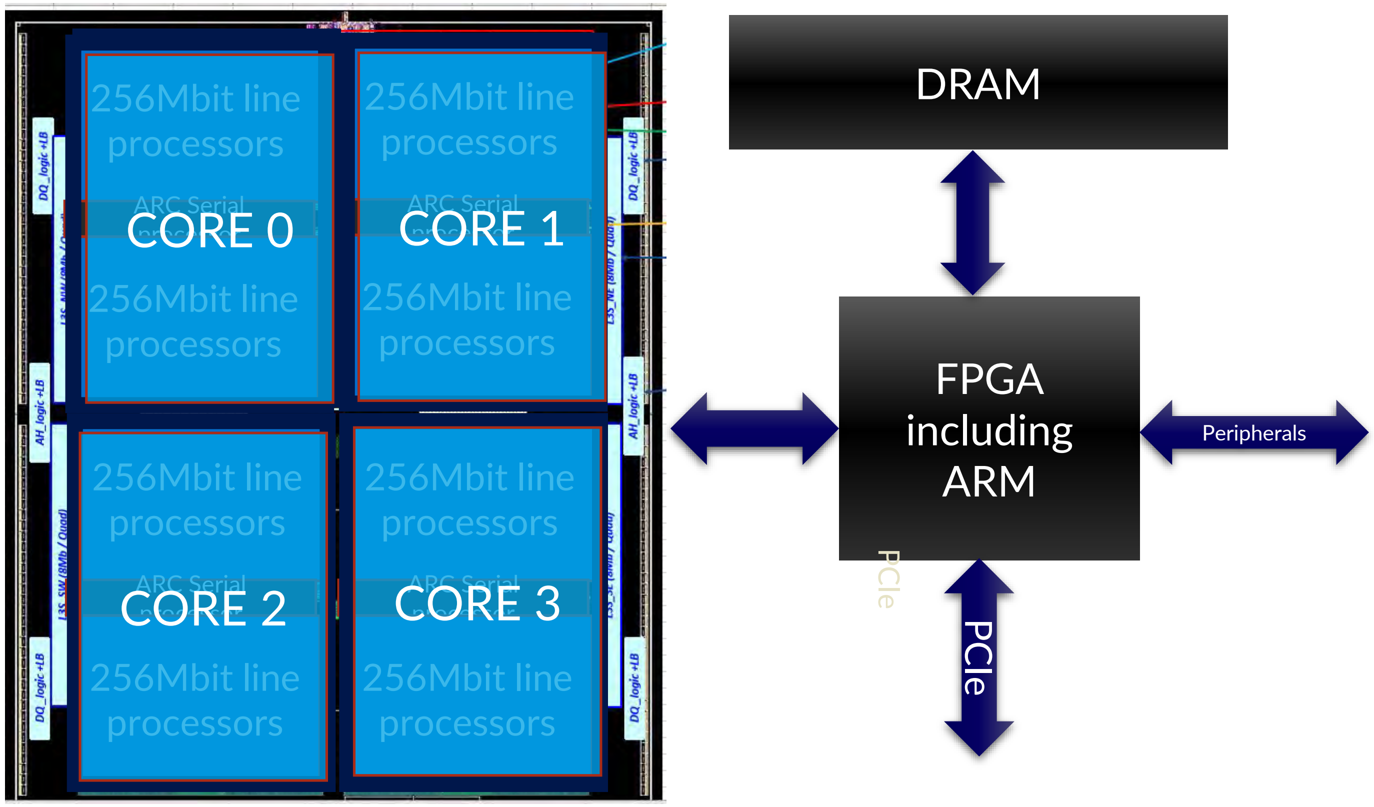
CPU/GPGPU (Current Solution)	(In-Place Computing (APU
Send an address to memory	Search by content
Fetch the data from memory and send it to the processor	Mark in place
Compute serially per core (thousands of cores at most)	Compute in place on millions of processors (the memory itself becomes millions of processors
Write the data back to memory, further wasting IO resources	No need to write data back—the result is already in the memory
Send data to each location that needs it	If needed, distribute or broadcast at once

Architecture

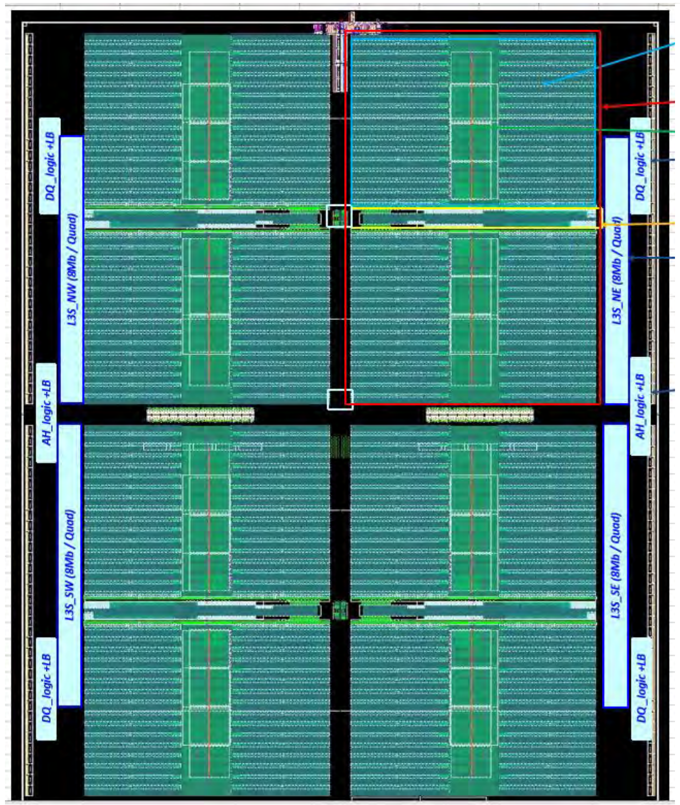


APU Chip Layout

2M bit processors or 128K vector processors runs at 1G Hz From 2 Tera Flops to 2 Peta Ops



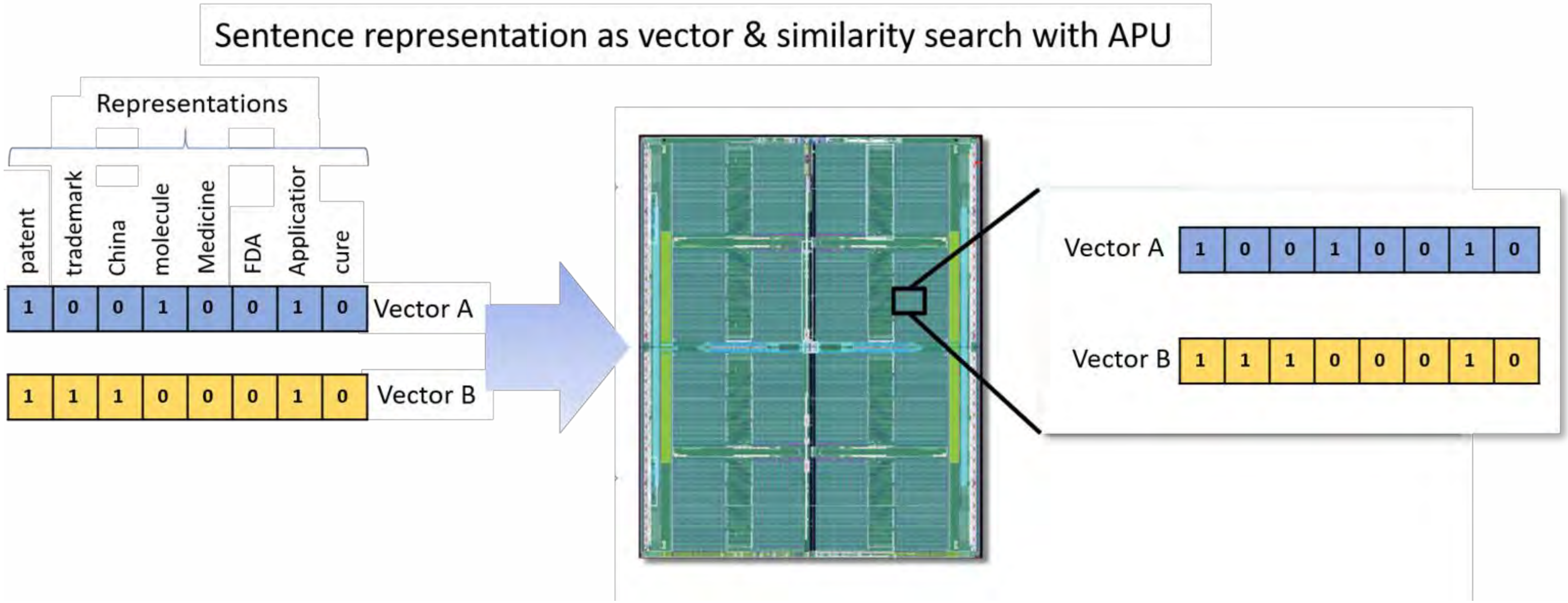
APU Layout vs GPU Layout



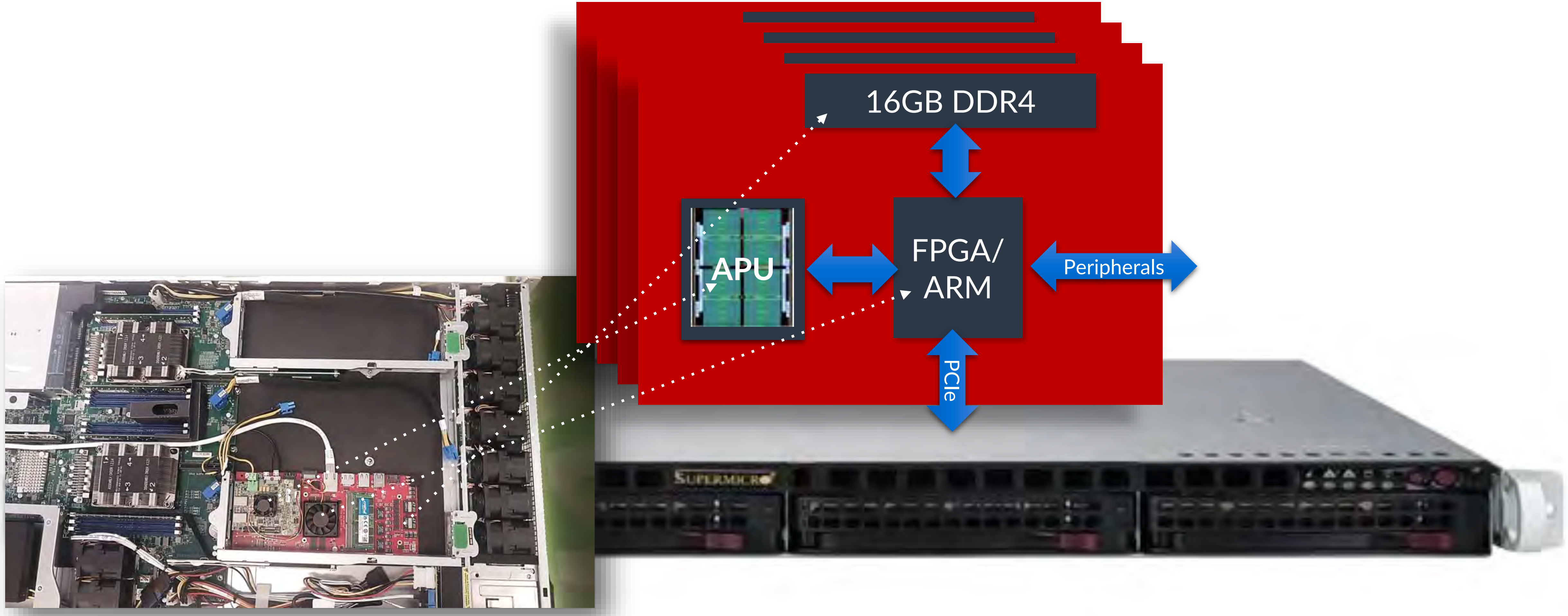
Multi-Functional, Programmable Blocks

Acceleration of FP operation Blocks

In-Memory Compute Example



APU board/System Architecture



K- Nearest Neighbors for Big Data

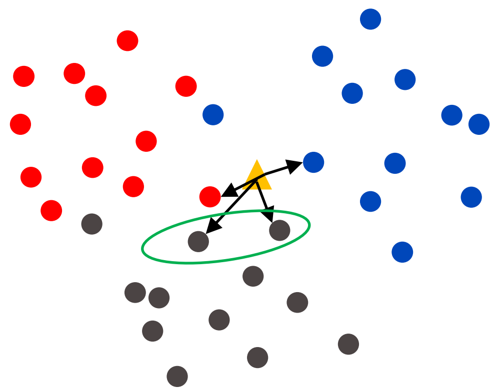
K-Nearest Neighbors (k-NN)

Simple example:

N = 36, 3 Groups

2 dimensions (D = 2) for X and Y

K = 4



Group **Green** selected as the majority.

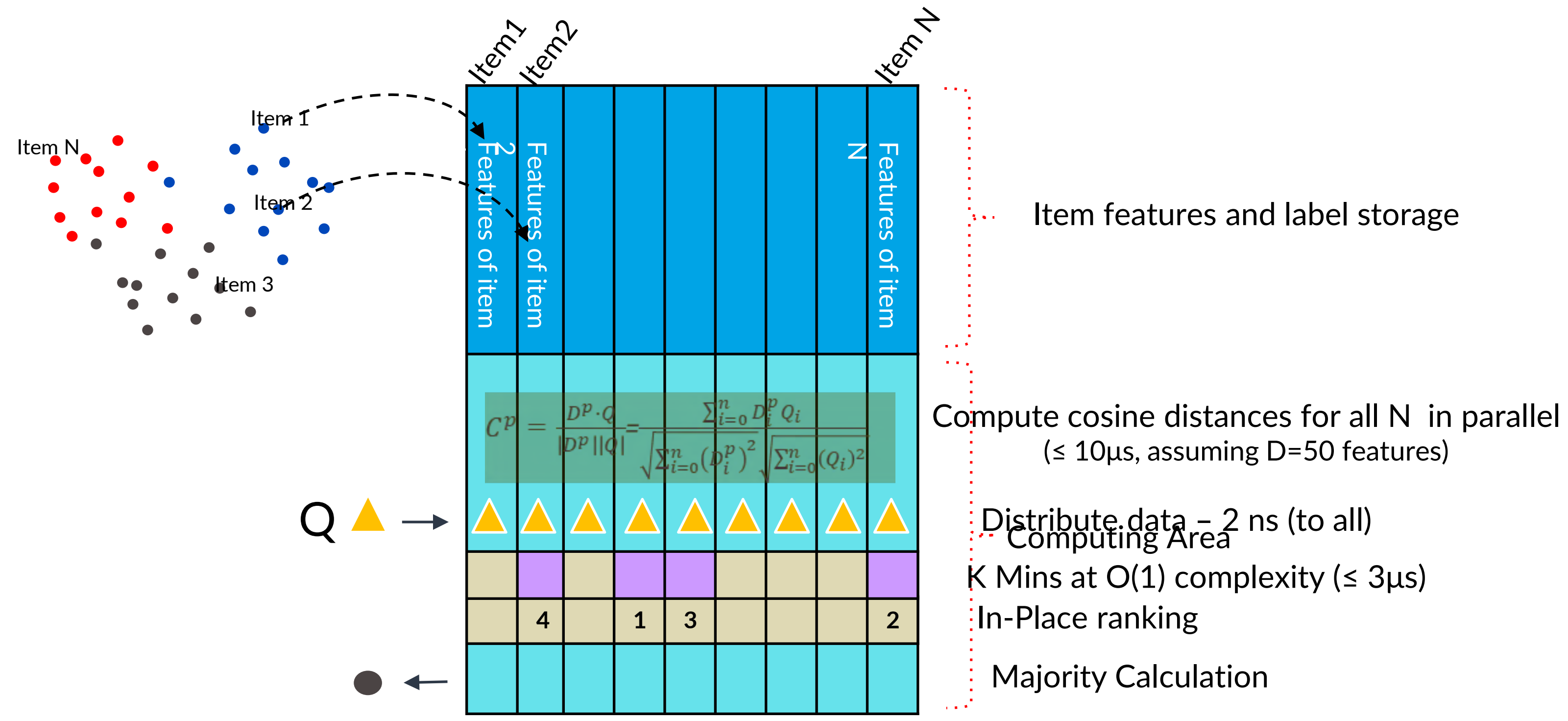
For actual applications:

N = Billions

D = Tens

K = Tens of thousands

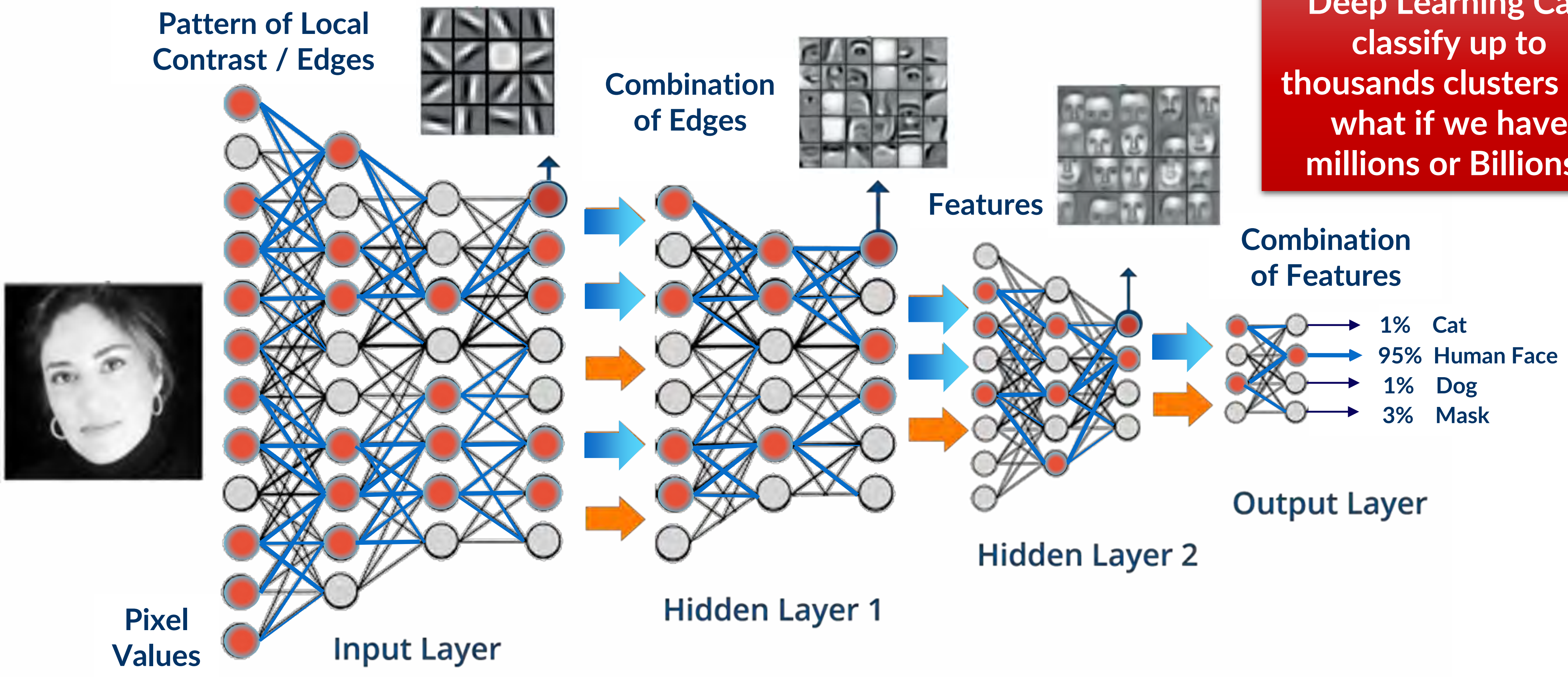
k-NN Use Case in an APU



With the data base in an APU, computation for all N items done in ≤ 0.05 ms, independent of K (1000X Improvement over current solutions)

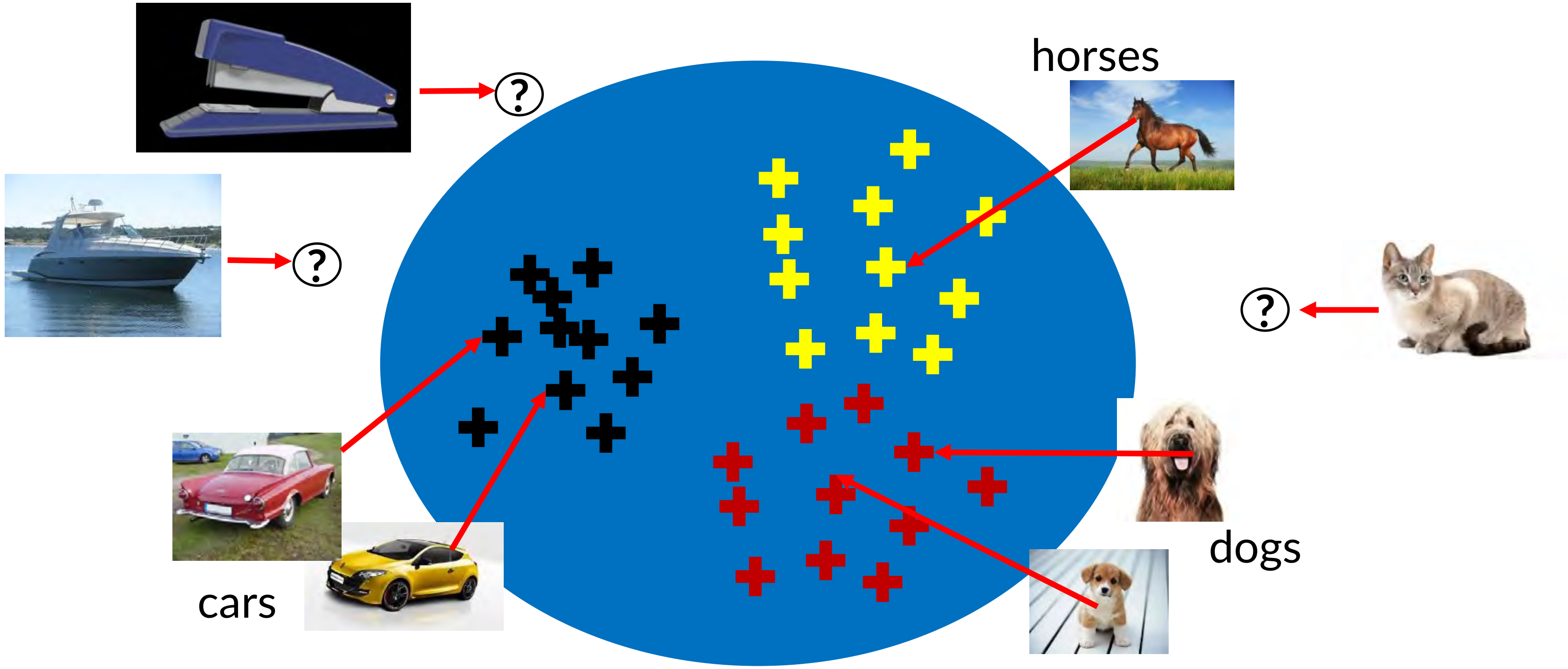
Big Data Classification

The Problem In Deep Learning



Deep Learning Can classify up to thousands clusters but what if we have millions or Billions?

What About New Updates



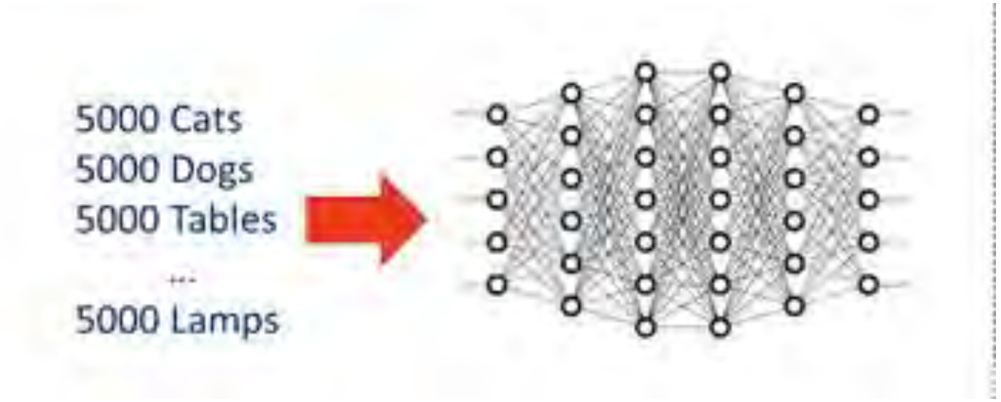
Updates unlabeled images requires new training – that consume latency, power, performance

DEEP LEARNING IN NOT ENOUGH

Associative Computing for Zero/Low Learning

Gradient-Based Optimization has achieved impressive results on supervised tasks such as image classification

These models need a lot of data



ASSOCIATIVE COMPUTING

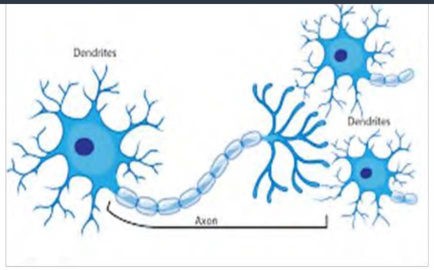
Like people, can measure similarity to features stored in memory

Can also create a new label for similar features in the future

Visual search, Face recognition and NLP are some of used cases showing on next slides

People can learn efficiently from few examples

1 Cat
1 Dog
1 Table
...
1 Lamp



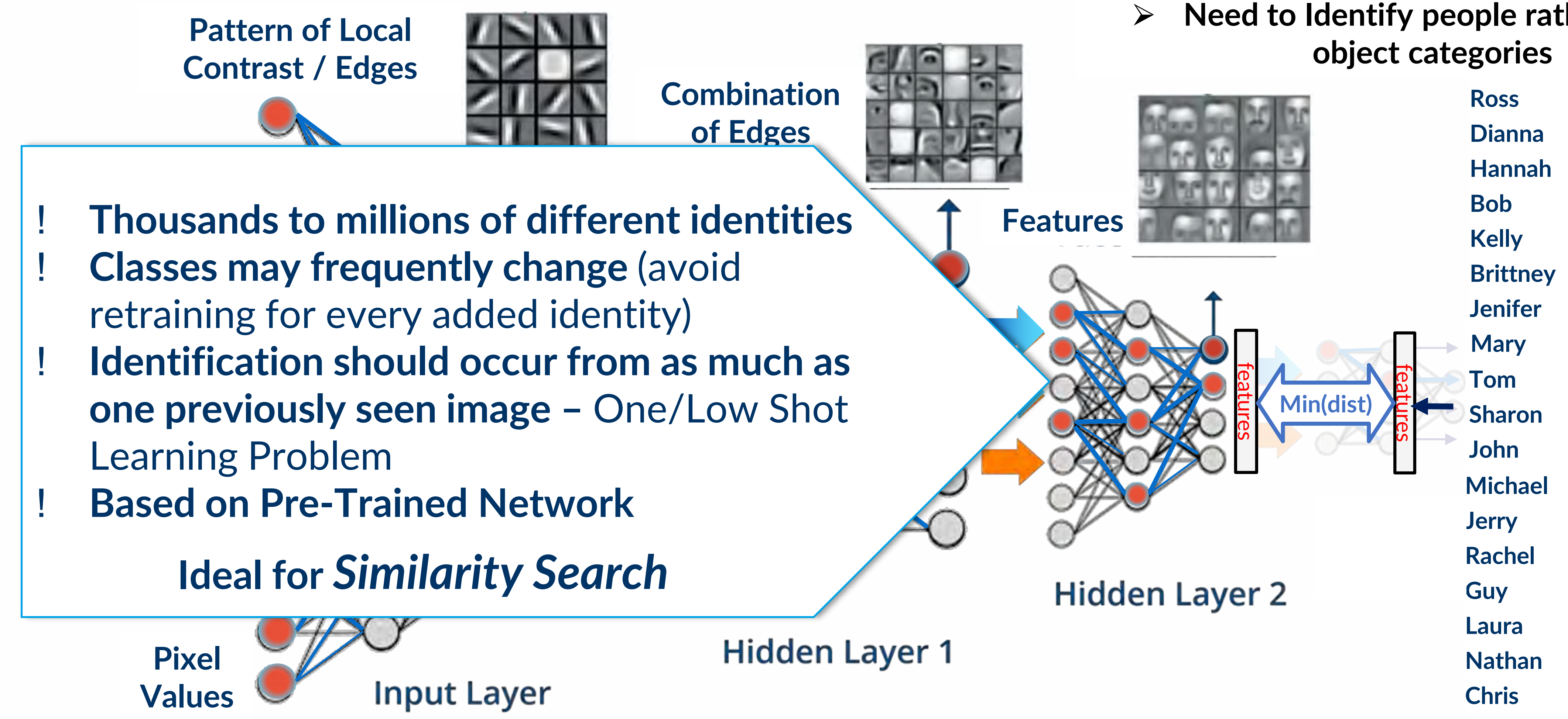
Millions to Billions Categories

Neural Network as Feature Extractor

- Need to Identify people rather than object categories

- ! **Thousands to millions of different identities**
- ! **Classes may frequently change** (avoid retraining for every added identity)
- ! **Identification should occur from as much as one previously seen image –** One/Low Shot Learning Problem
- ! **Based on Pre-Trained Network**

Ideal for *Similarity Search*

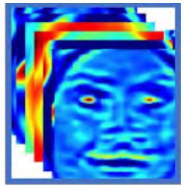


Face Recognition Pipe Line

Identification/Retrieval

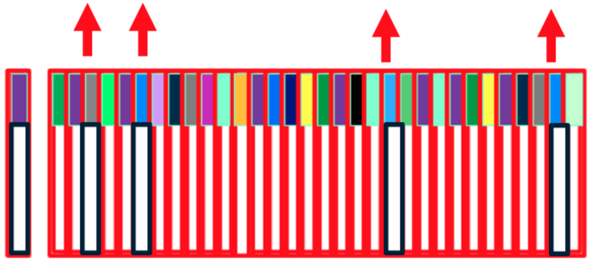
Face Detection (MTCNN) → Face Alignment → Deep Feature Extraction → Hashing (LSH) → Similarity Search (Hamming Distance + TopK)

Query Image



128 float32

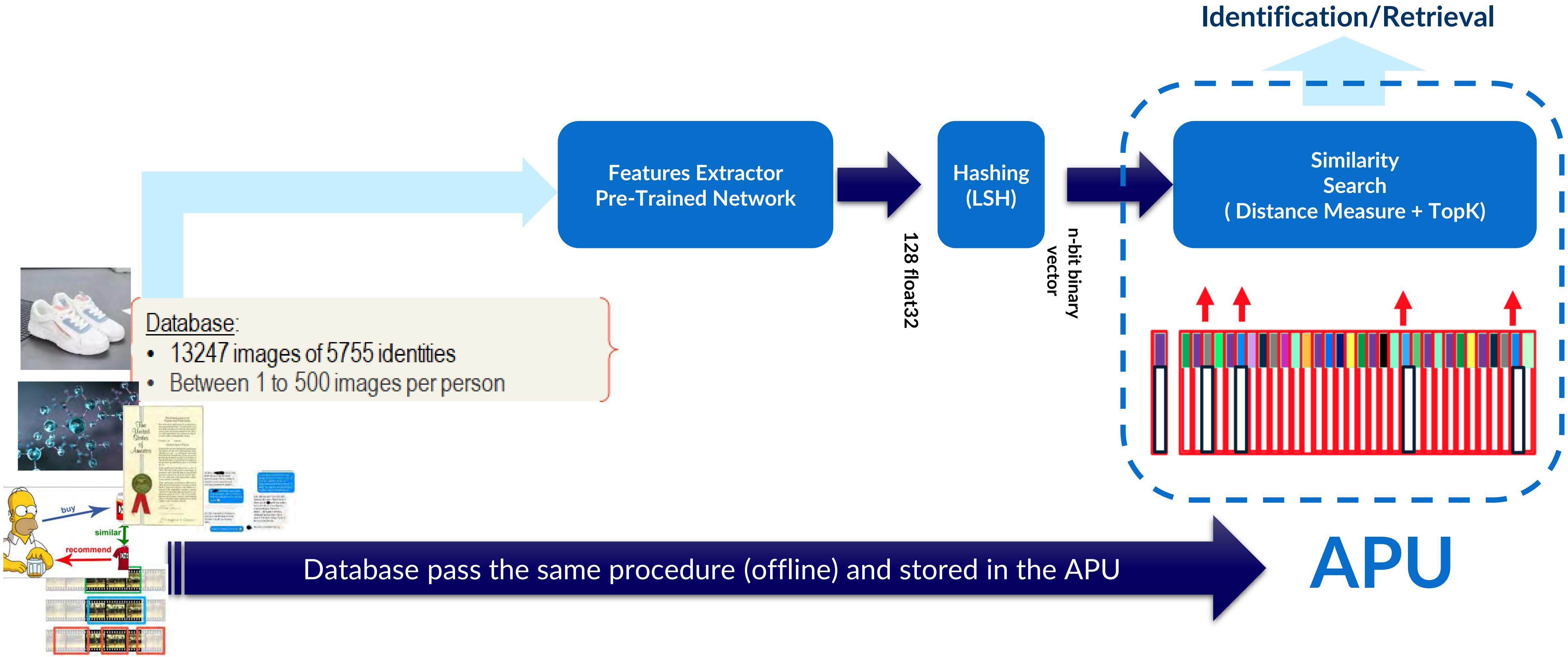
n-bit binary vector



Database faces pass the same procedure (offline) and stored in the APU

APU

Same Concept for Any Big Data Item



Face Recognition Example

Database:
- 13247 images of 5755 identities
- Between 1 to 500 images per person

Query Images

Face Feature Extraction

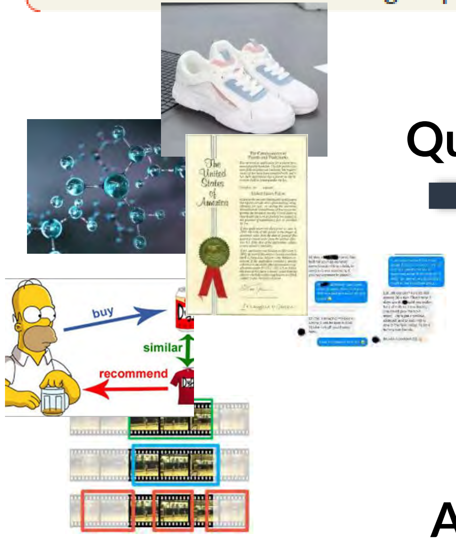
```
MTCNN found 4 faces:
```

```
====== Facenet Embeddings:
(4, 128) float32
total time: 0:00:01.954904 / 4 images - 4 faces
detection time: 0:00:00.622483 / 4 images
face embedding time: 0:00:00.070863 / 4 faces
```

Similarity Search

In Memory Big Data Similarity Search

Database:
- 13247 images of 5755 identities
- Between 1 to 500 images per person



In-Memory AI Server

Query

ON LINE COMPUTING	OFF LINE COMPUTING	
APU Associative Computing Unit	DRAM	CPU
	In-Memory Storage < 1 TByte	convert to feature vector

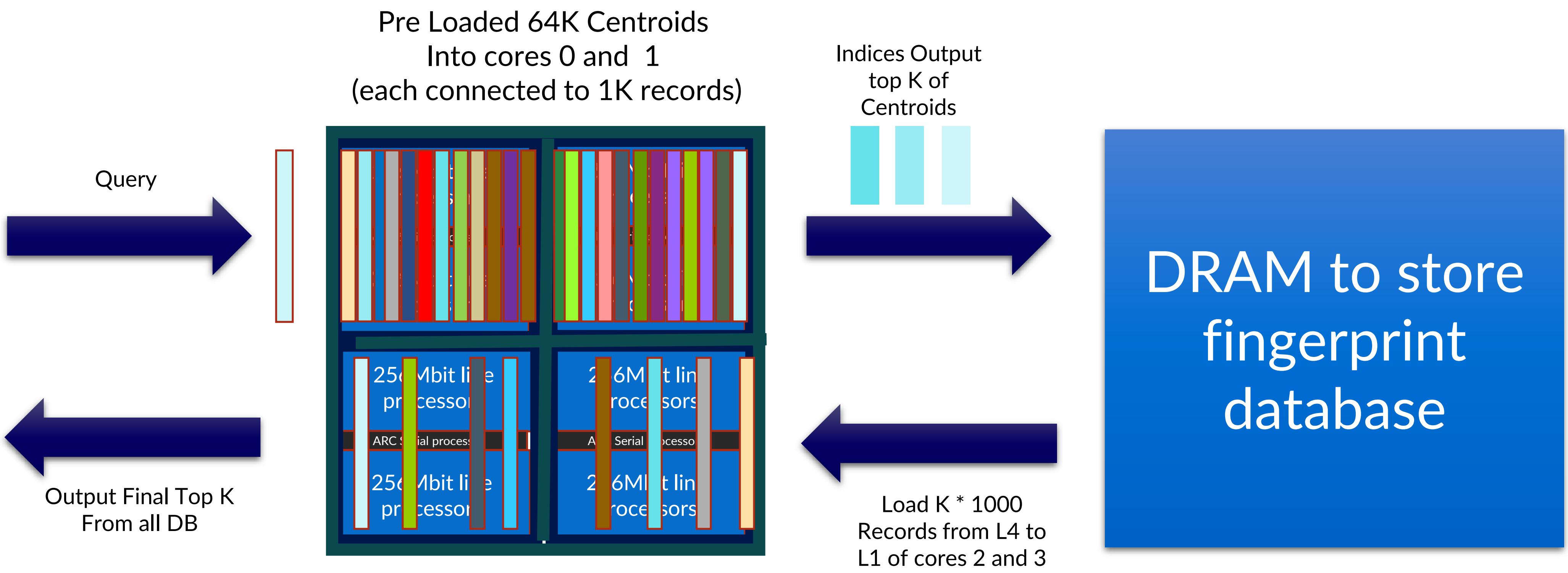
Embedding

Answer

Cloud Server

Searching Concept on APU

Example: Searching 64M records in a single APU chip



Example: 64 M records = 64 K Centroids X 1000 records each

Up to 100,000 queries/sec

Single Server 256M Records Similarity Search

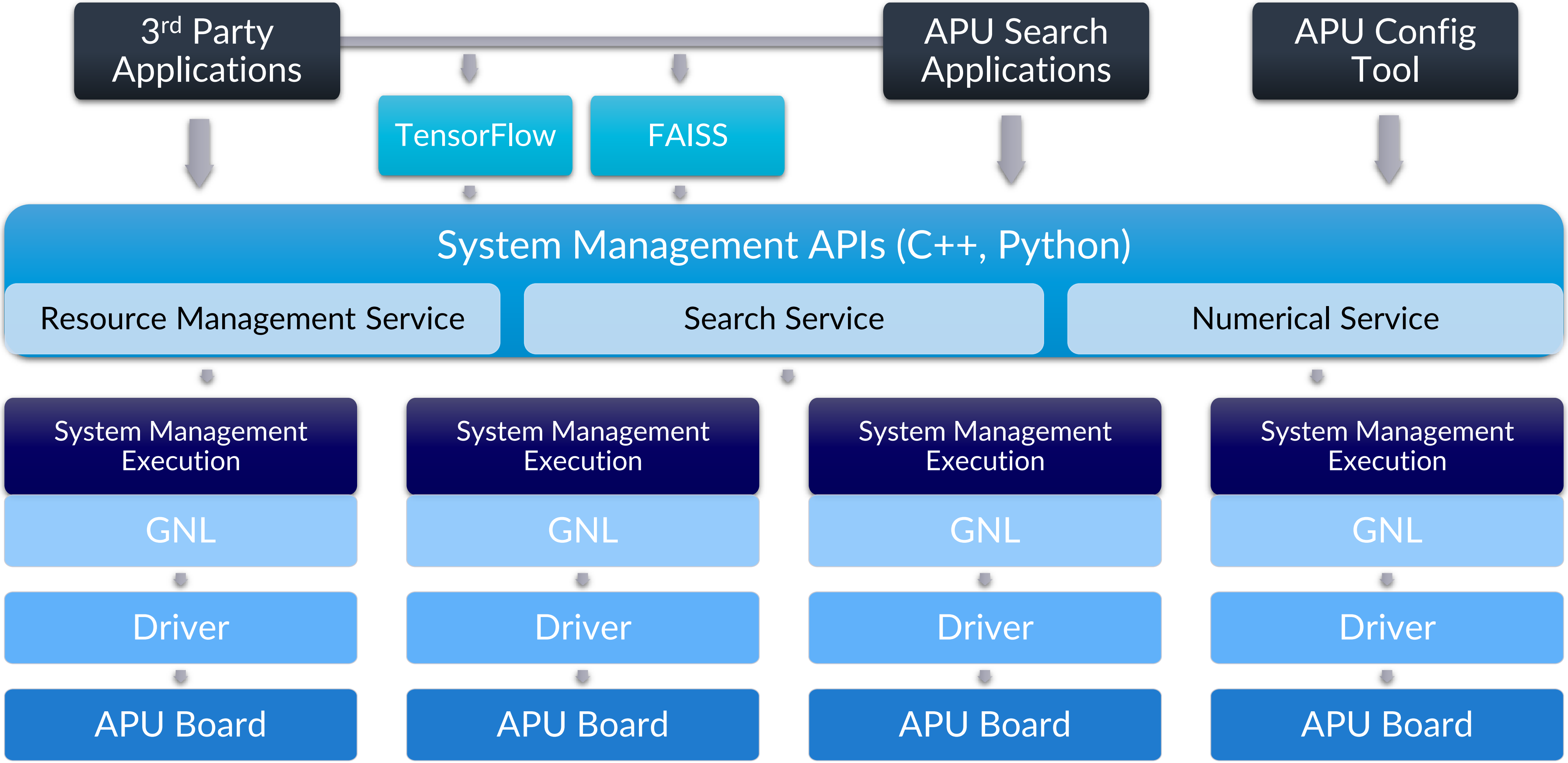
- **HW:**
  1 Sever with 4 APU Boards
  (One APU 1.1 ASIC Per Board)

- **Data Base:**
  - 256 Million Images
  - 256M Binary Vectors with nBit=512 ---
    Total: 16GB

- **Pre Search Preparation:**
  - DB Clustering
    - 256K Clusters x 1000 Records in each Cluster
    - Cluster Size: 16MB
    - Total Records size: 16GB
  - 2 APU's will be use for TOP-K clusters and
    2 APU's will be use for TOP-K Records

SW Tools

Software Stack Layout



APU- Supported Functionalities

- In memory DB
- GSI API
- GSI Numeric Library
- GSI APU Driver
- GSI APU

- Comprehensive list of numerical function algorithms supported
- Wide range of algorithms
- **Multiple clustering techniques**
- Interfacing supported
- Range of interfaces

Use Case Example

Weizmann Institute of Science

Molecule Similarity Structure Search

DB Size for the Pilot:

- 38M Compounds

Vector Size:

- 512 Bits, Search time 12 sec.
  Instead of 6 Minutes
- 1024 Bits, Search Time 24 Sec.
  Instead of endless time
- The performance based on GSI prototype chip.
  For commercial search time is 0.4 sec for 512 bits per 100 queries , or 0.8 sec for 1K bits per 100 queries.

Solution is scalable for any size of DB any size of fingerint and any type of search algorithm.

Search:

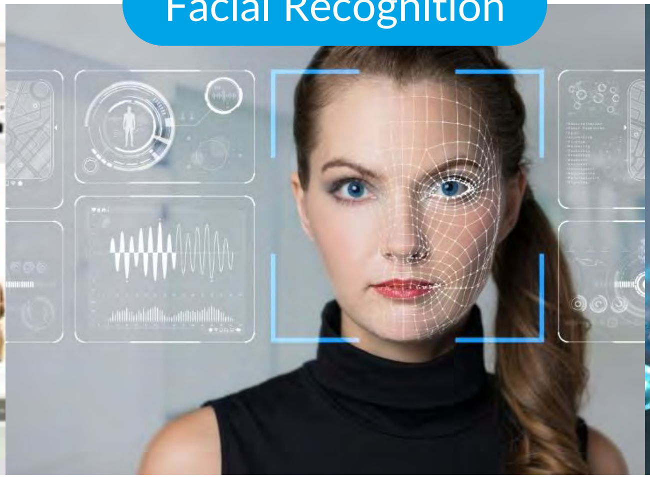
- **Algorithm**: Tanimoto
- Support Threshold Search
- **K- Nearest Neighbors** (KNN) K=1,10,100,1000

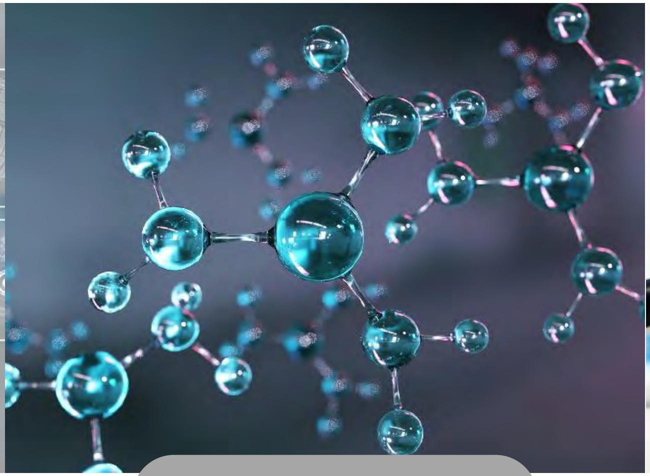
מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

GSI Current Applications

Visual Search

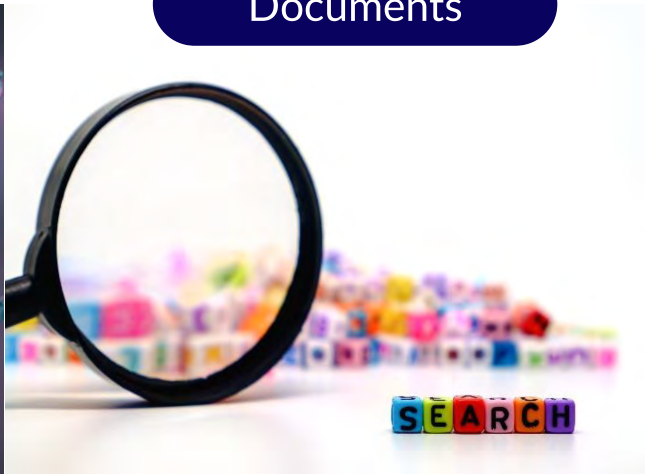
Facial Recognition



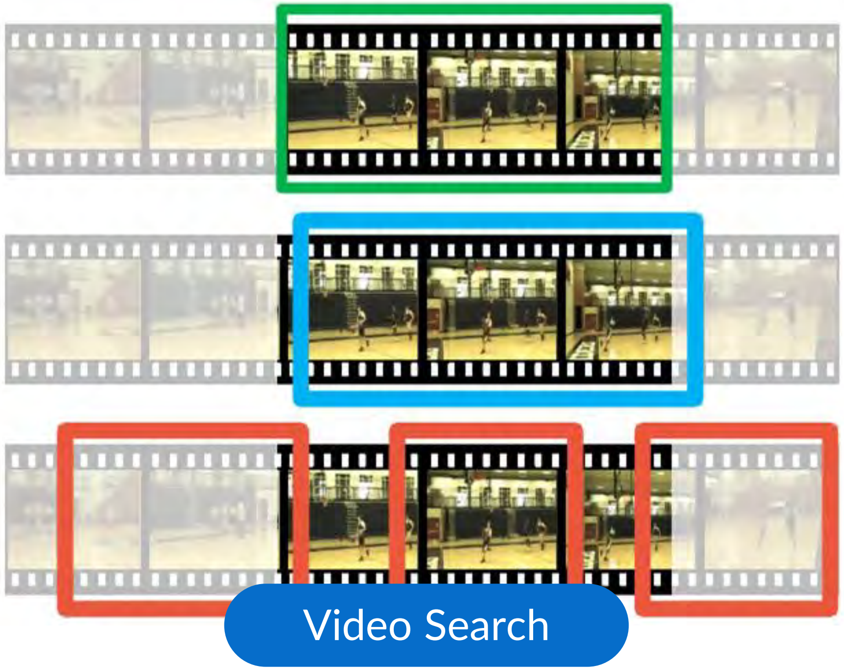


Molecules Search

Documents



In Research



Thank You

QUESTIONS?