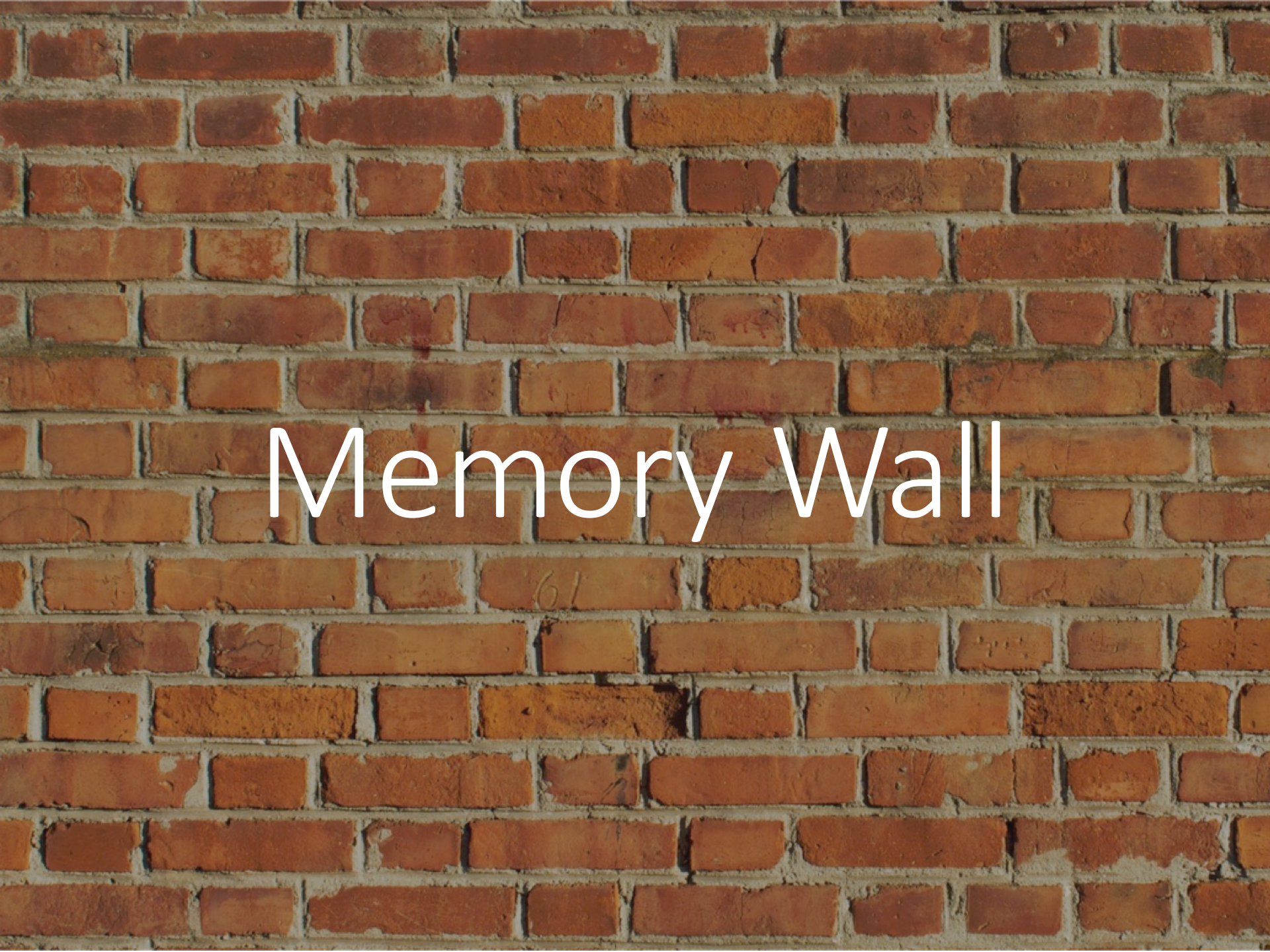
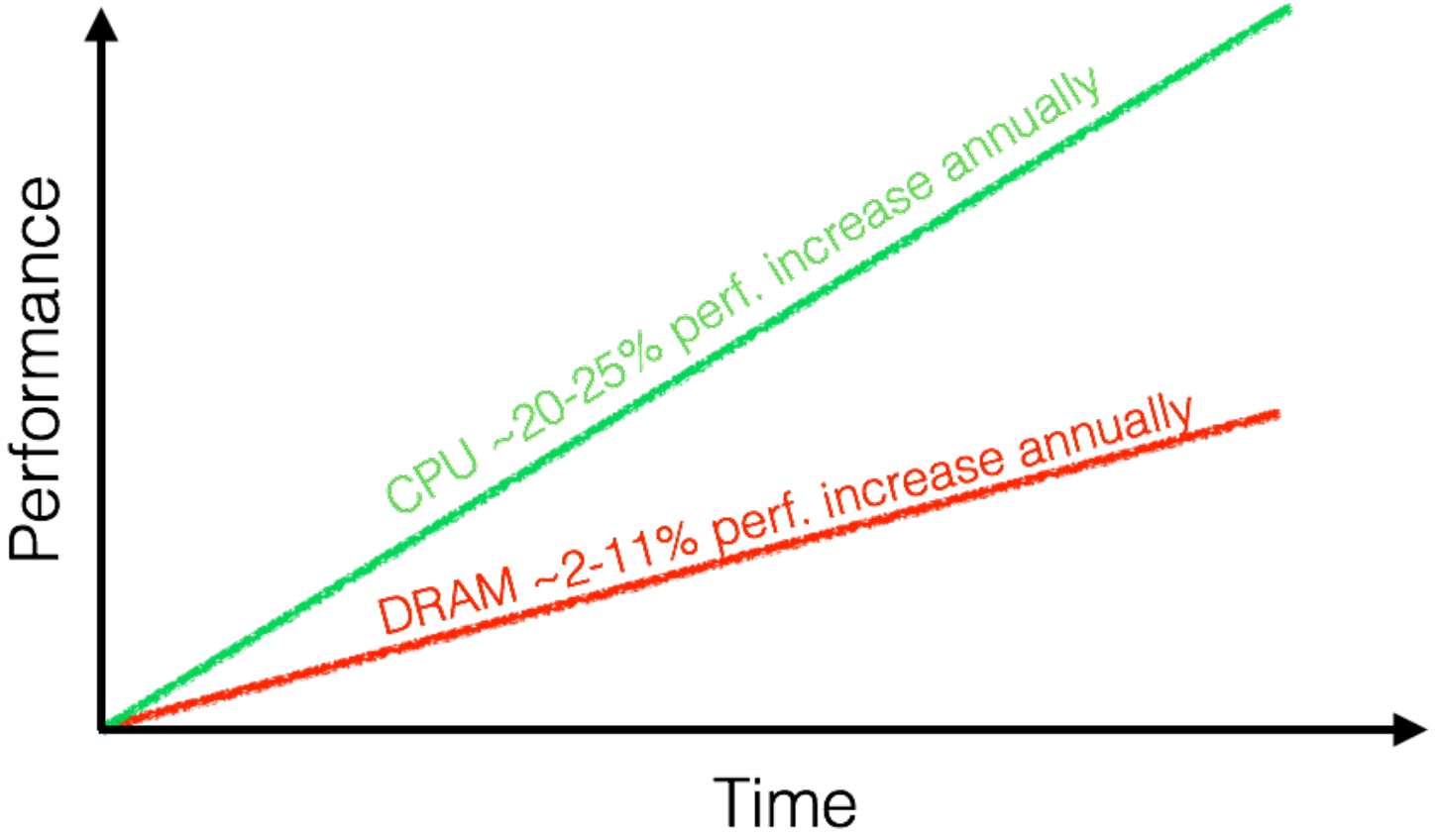
# Beyond the Wall:
## Near-Data Processing for Databases

*Sam Xi*, Ore Babarinsa, *Manos Athanassoulis*, Stratos Idreos

HARVARD
UNIVERSITY

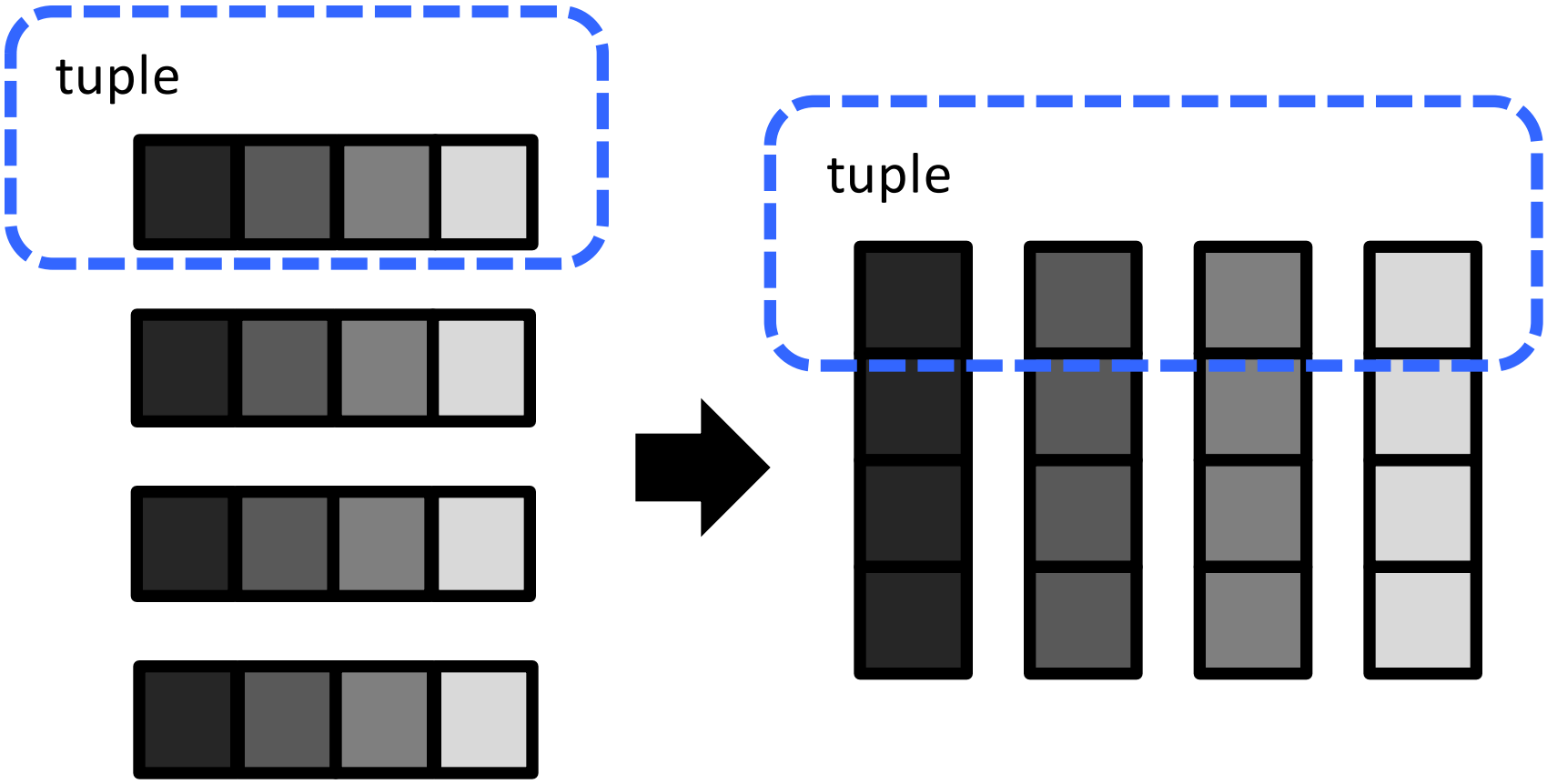

Memory Wall

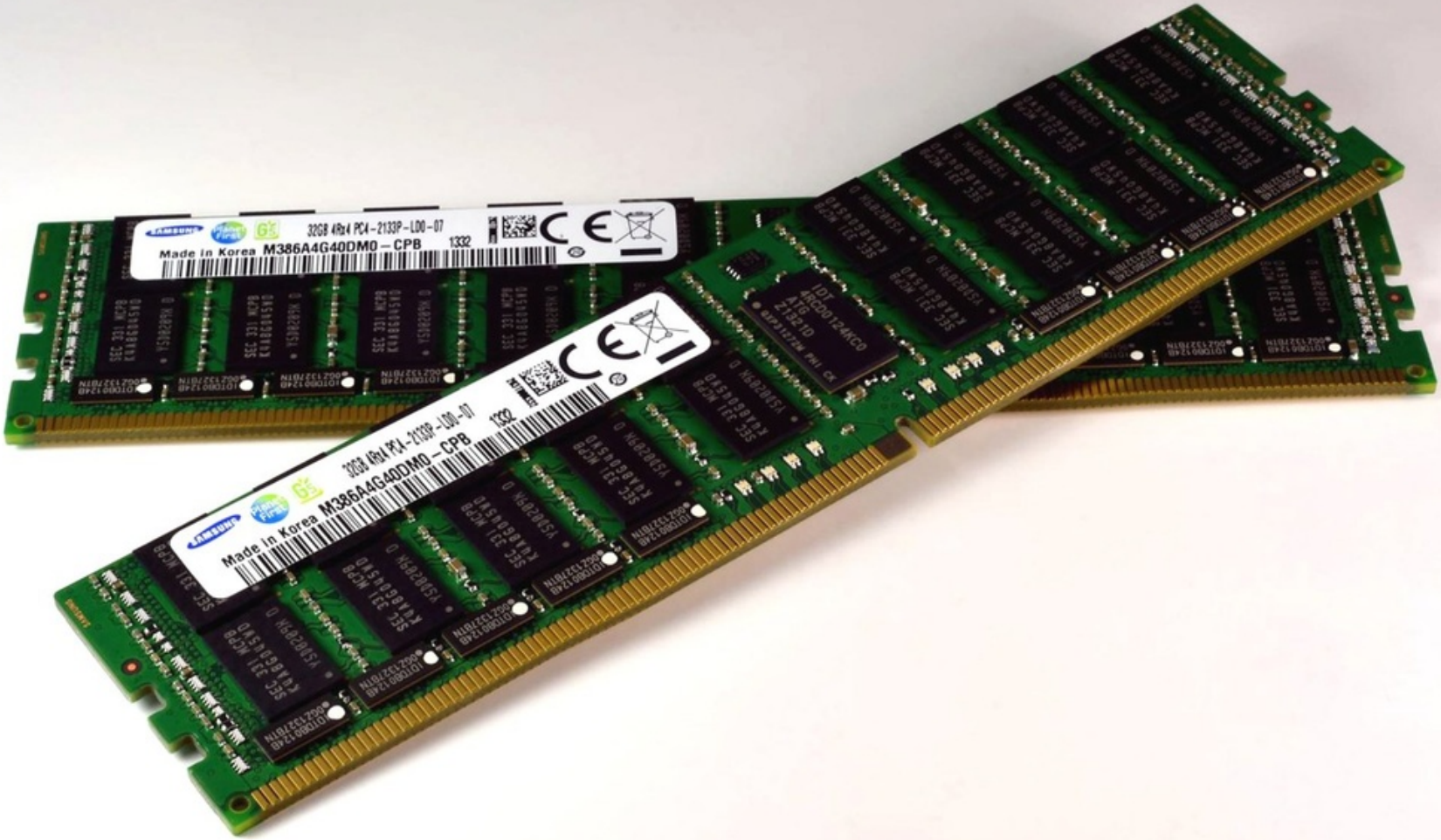
# Memory Wall


Performance
CPU ~20-25% perf. increase annually
DRAM ~2-11% perf. increase annually
Time

## Row store

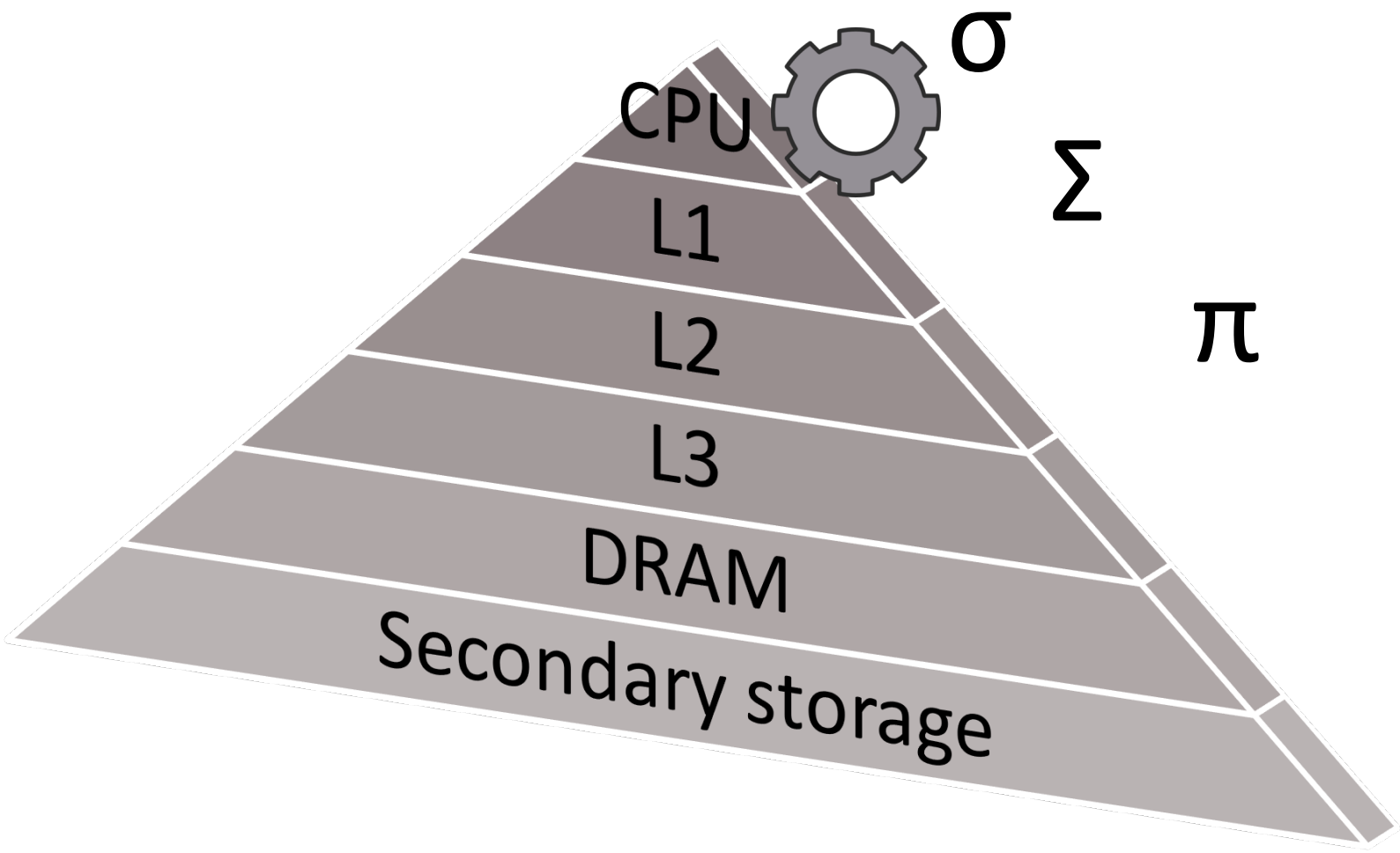
## Column store



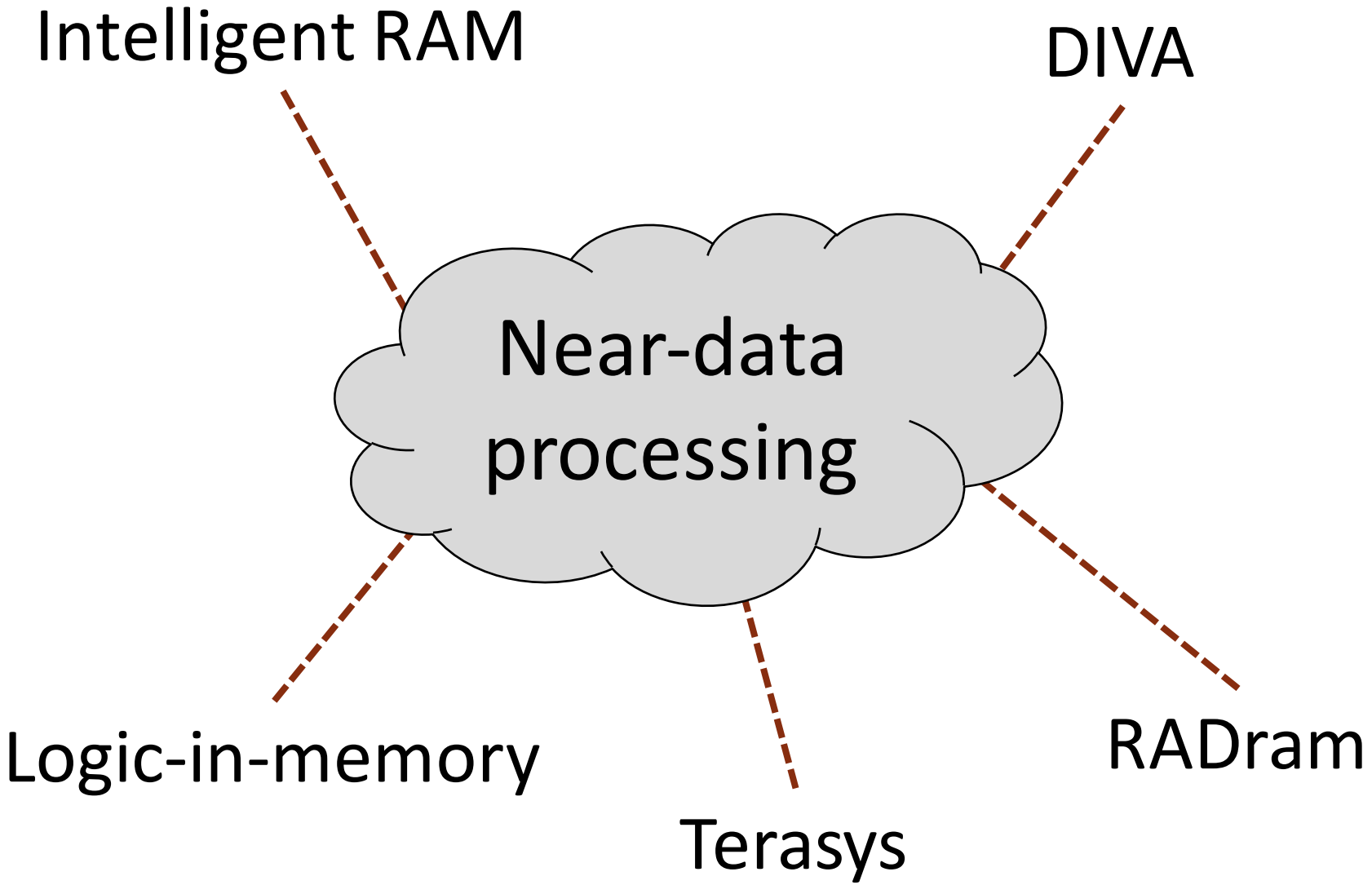



# Memory-optimized data systems

# Data access *remains* the bottleneck


σ
CPU
Σ
L1
L2
π
L3
DRAM
Secondary storage

We are not the first to visit this pyramid!

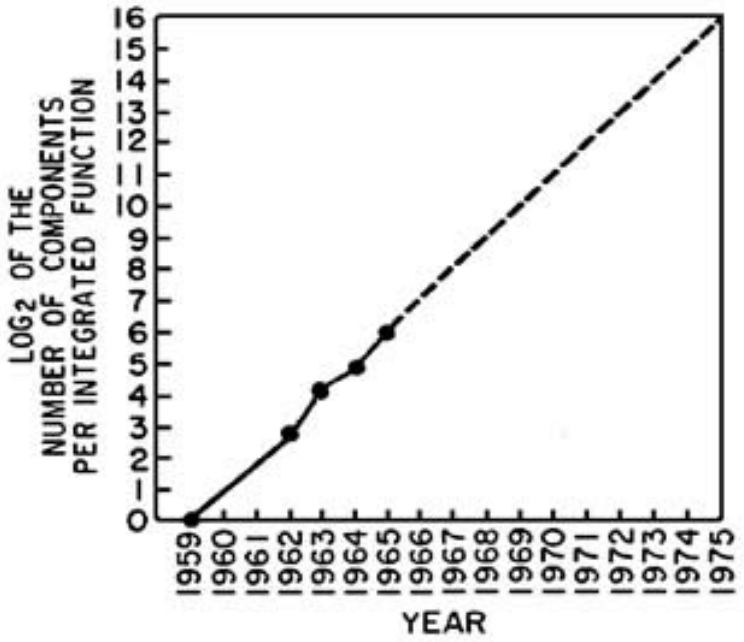

Intelligent RAM
DIVA
Near-data processing
Logic-in-memory
Terasys
RADram

# Why did NDP not take off?

| | DRAM | Logic |
|---|---|---|
| **Leakage** | Low | High |
| **Switching speed** | Slow | Fast |

**Fabrication processes are incompatible**

# Moore's Law + Dennard scaling
## provided consistent performance scaling for years

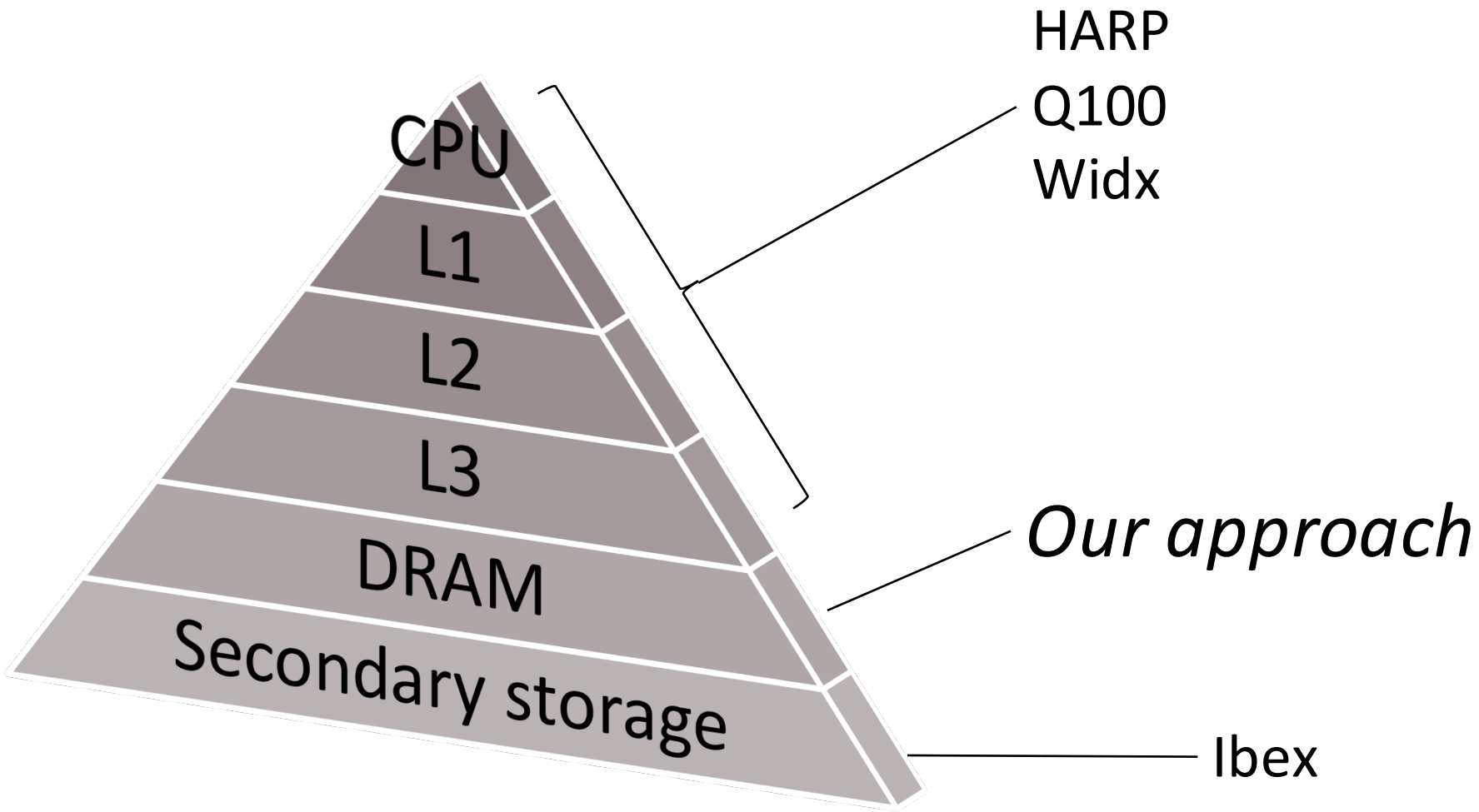



Moore's Law.

| Metric | Scaling factor |
|---|---|
| Area | $1/\kappa^2$ |
| Delay | $1/\kappa$ |
| Power | 1 |

Dennard scaling.

Not the case anymore!


HARP
Q100
Widx
CPU
L1
L2
L3
DRAM
Secondary storage
Our approach
Ibex

# Outline

Intro

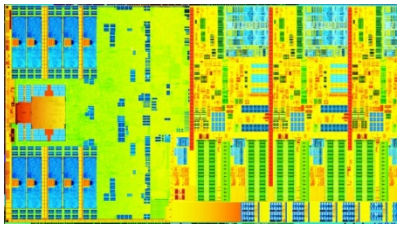
NDP for data systems: Past and present

The architecture of JAFAR

Experimental results

Conclusion

# Opportunity for NDP



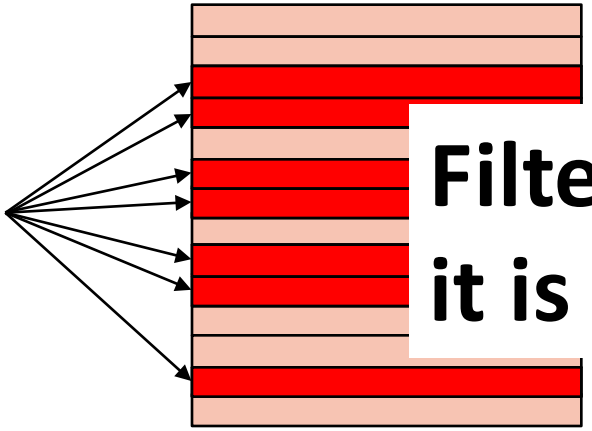
Host server

Query

**Lots of data**

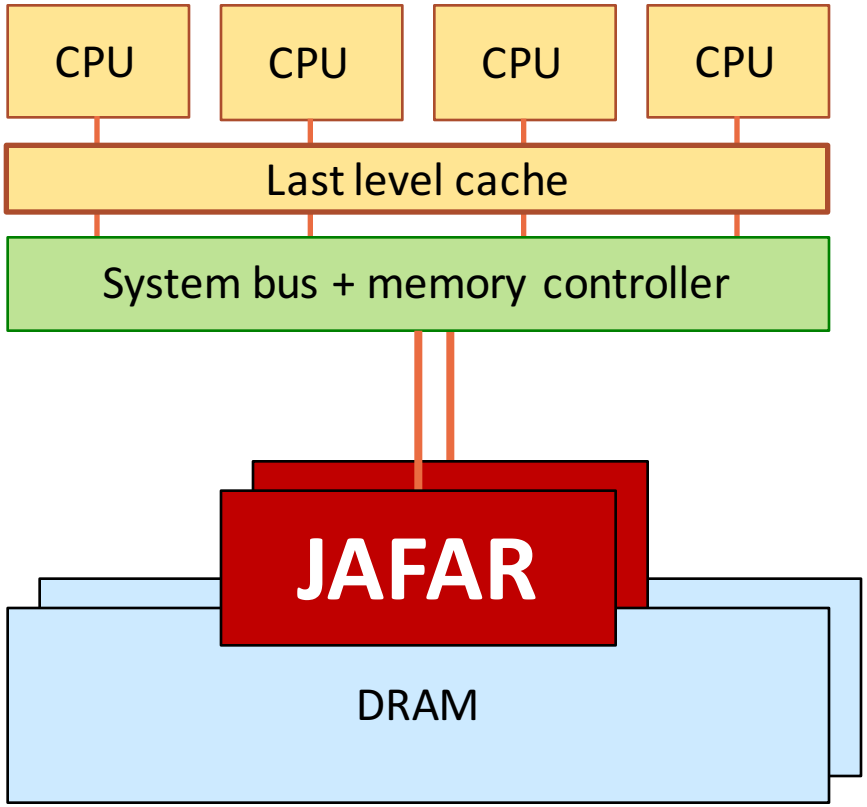
Database

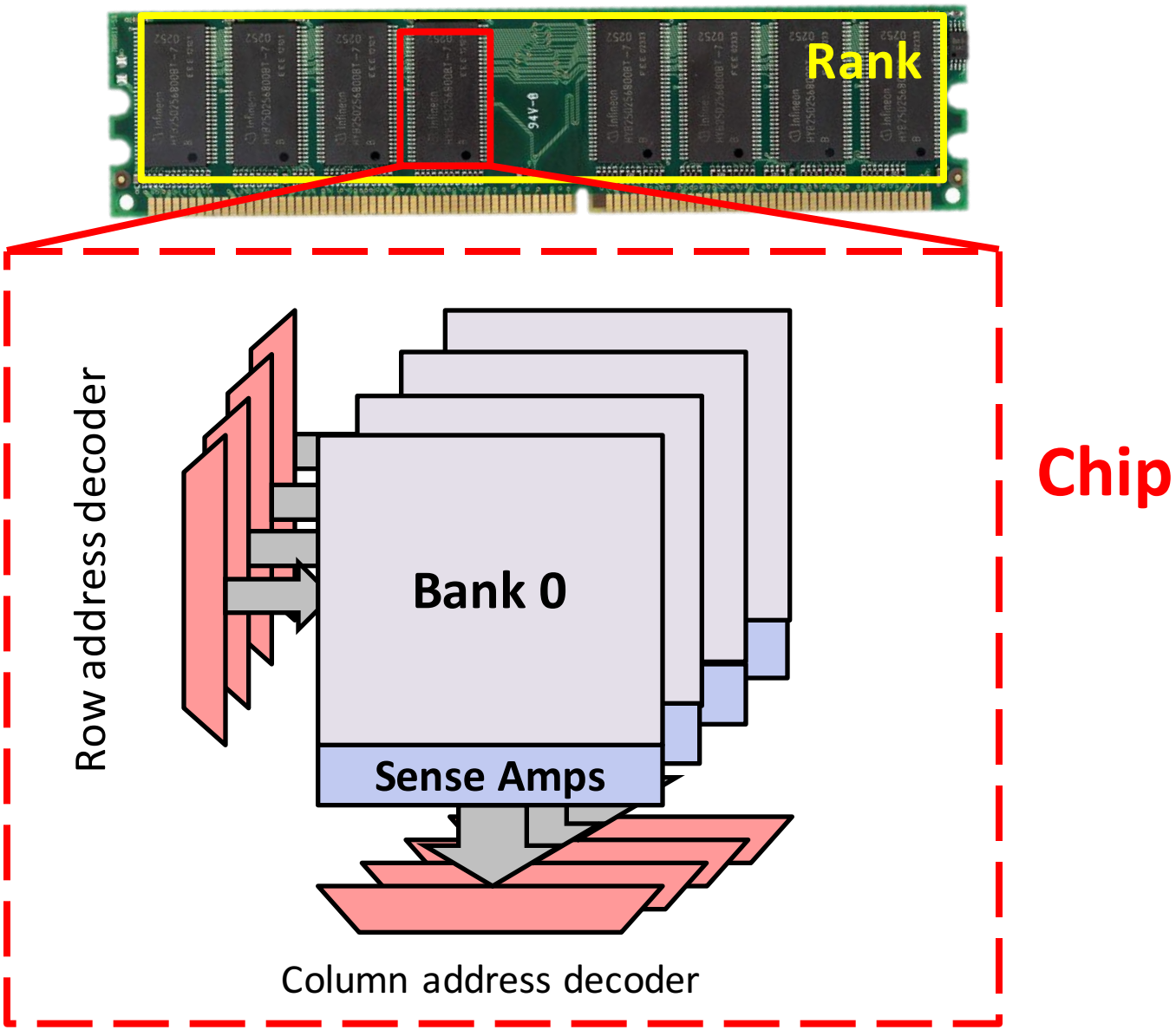
Many rows fail the query predicate and are discarded.

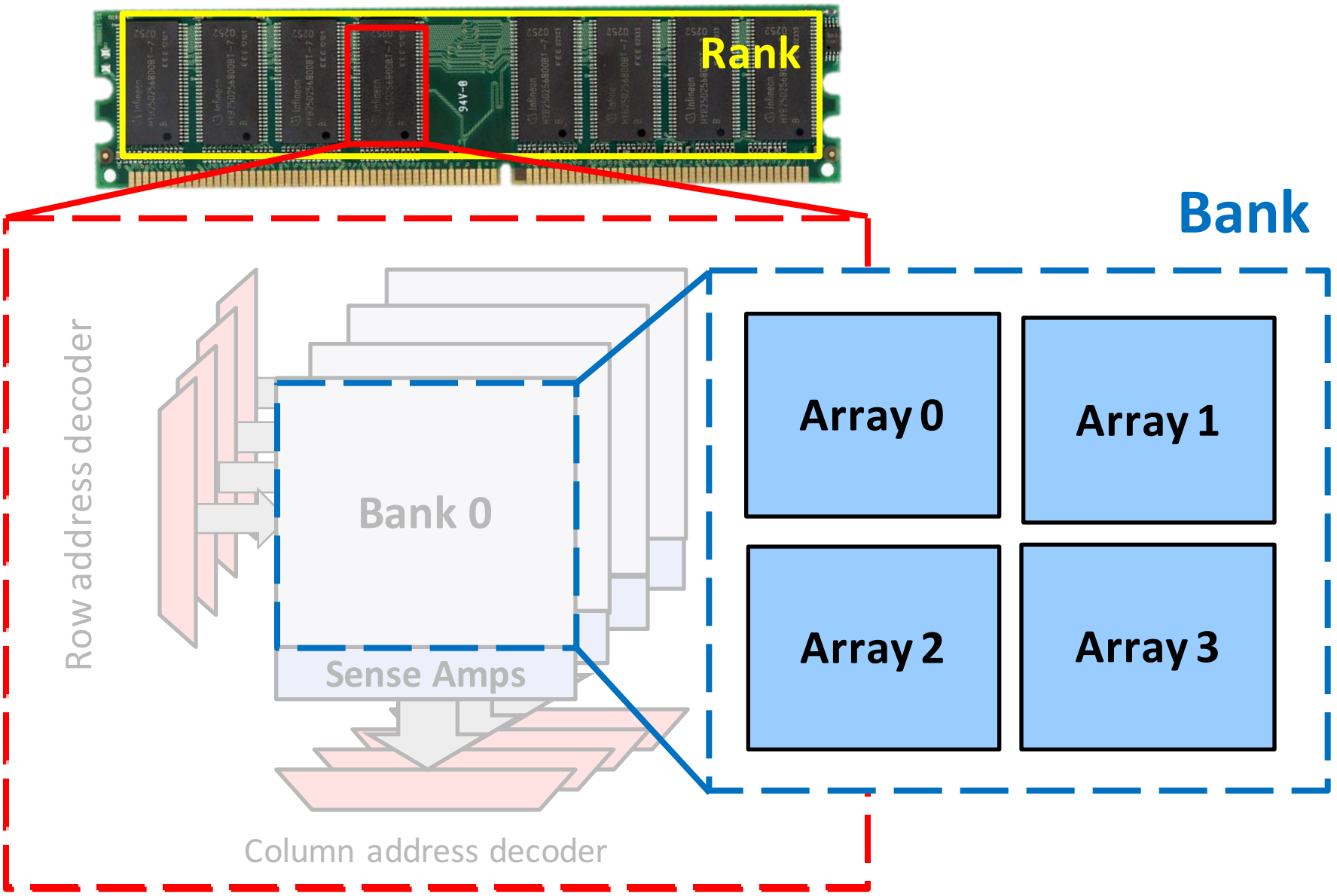


**Filter data before it is sent to CPU.**

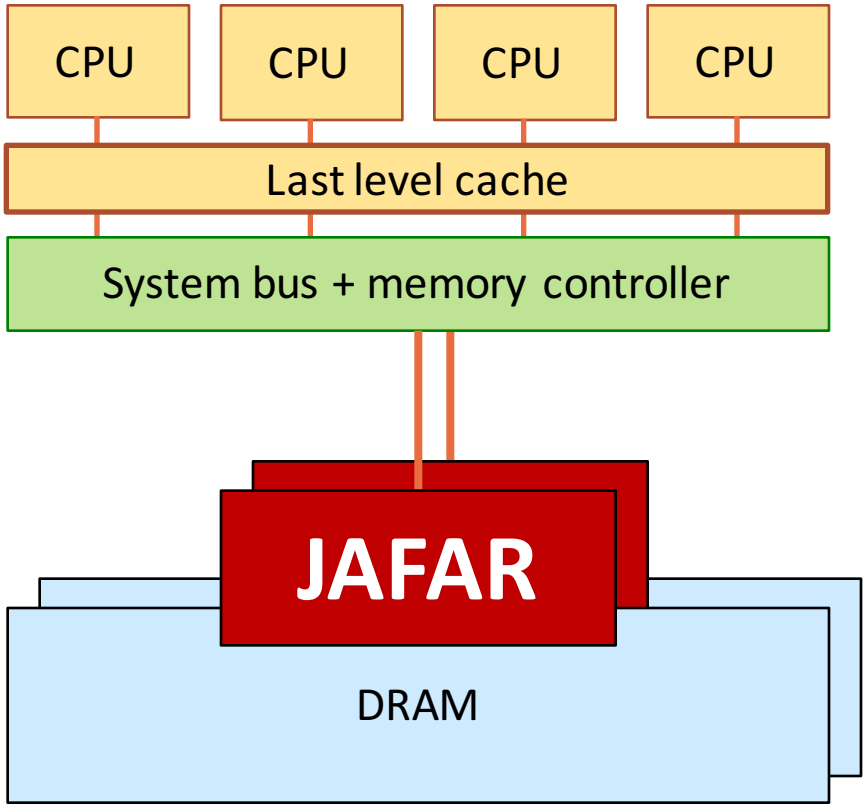
# JAFAR: “Just” A Filtering Accelerator on Relations



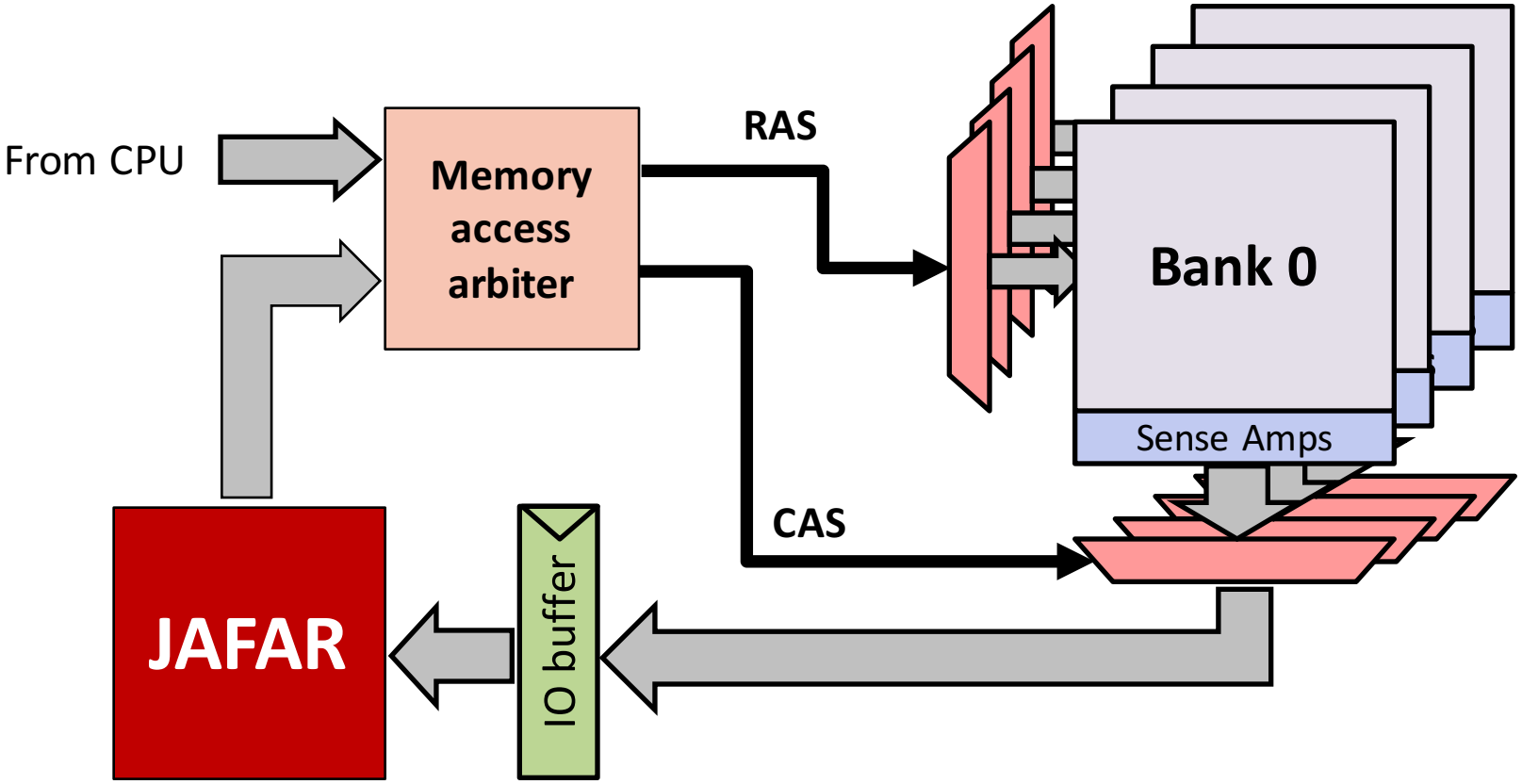

Rank
Row address decoder
Bank 0
Sense Amps
Column address decoder
Chip


Rank
Bank
Row address decoder
Bank 0
Sense Amps
Column address decoder
Array 0
Array 1
Array 2
Array 3

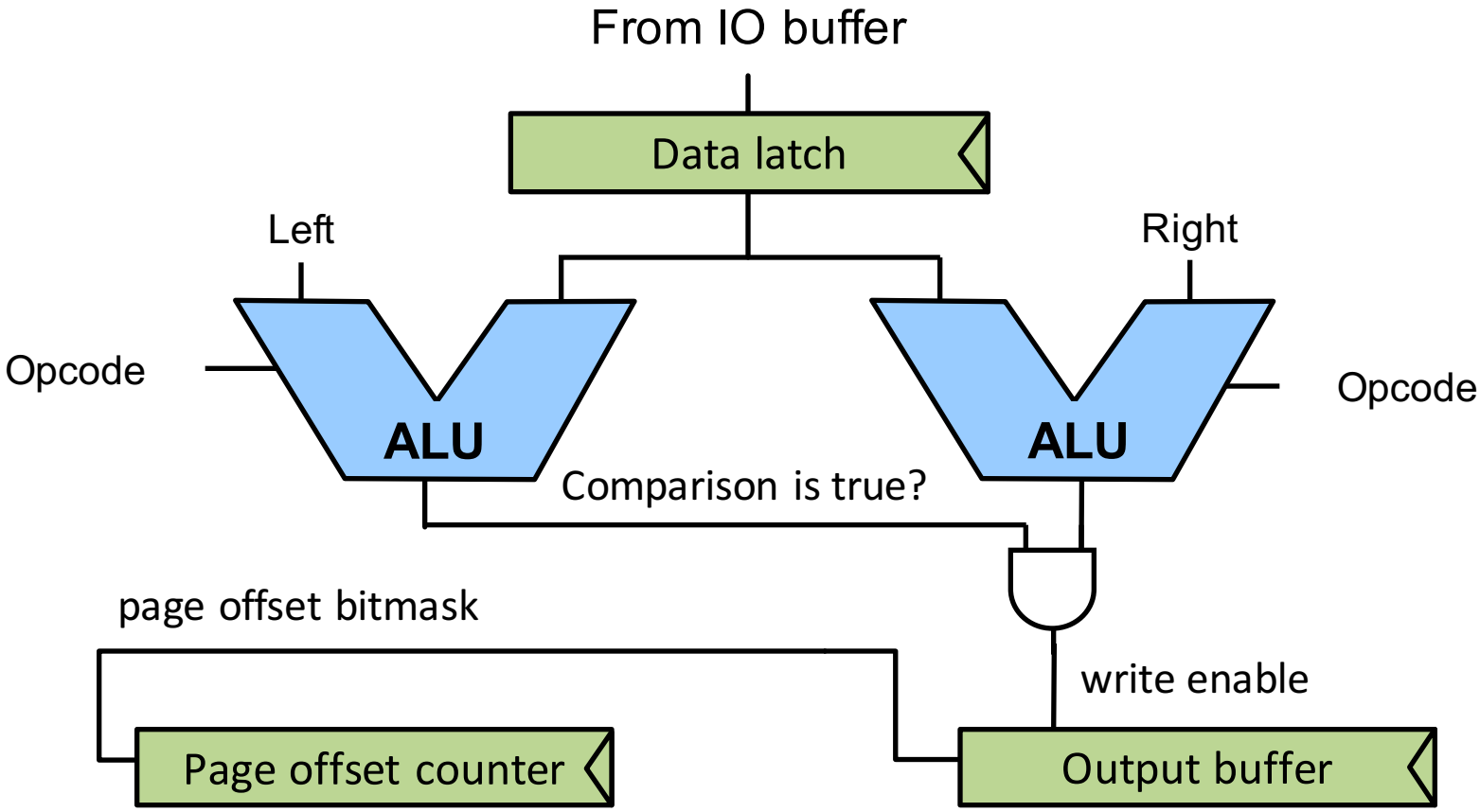
# JAFAR: Overall design



# JAFAR context



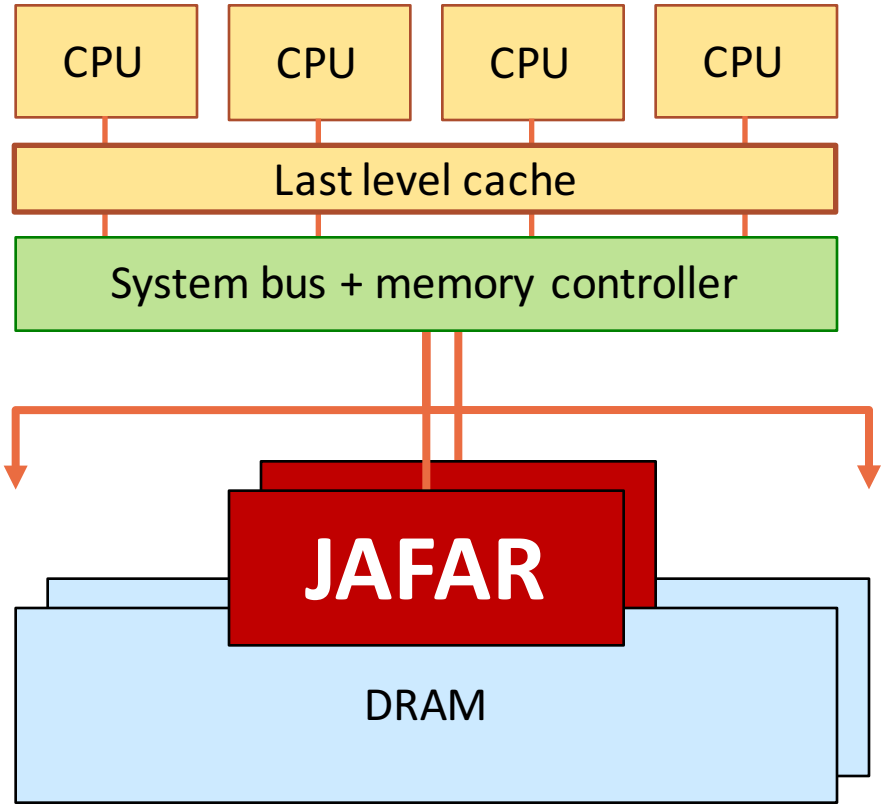
# JAFAR architecture



# Programming JAFAR

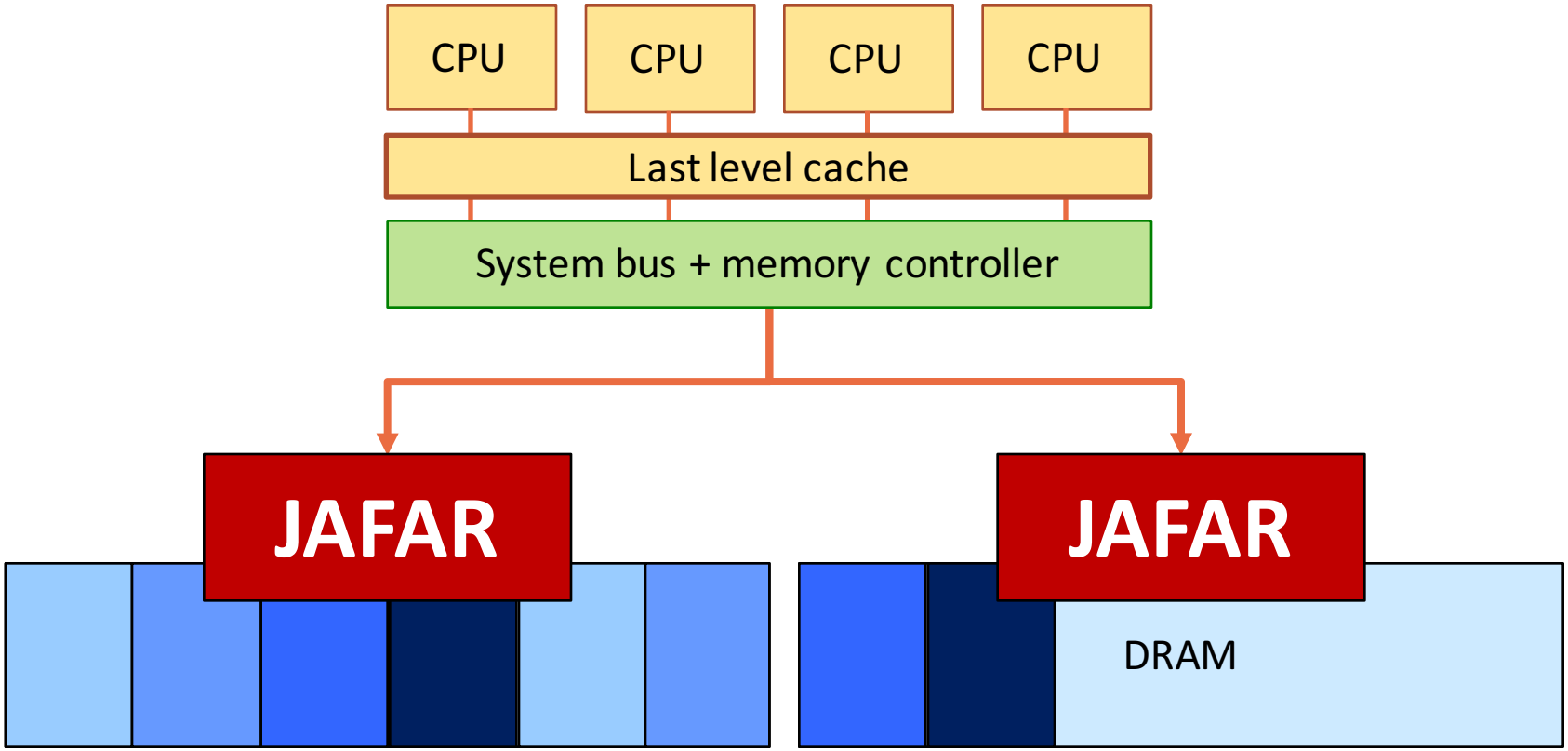
```
int errno =  select_jafar(
       void*          col_data,
       int            range_low,
       int            range_high,
       uint8_t*       out_buf,
       size_t         num_input_rows,
       size_t*        num_output_rows);
```

# Handling multiple modules



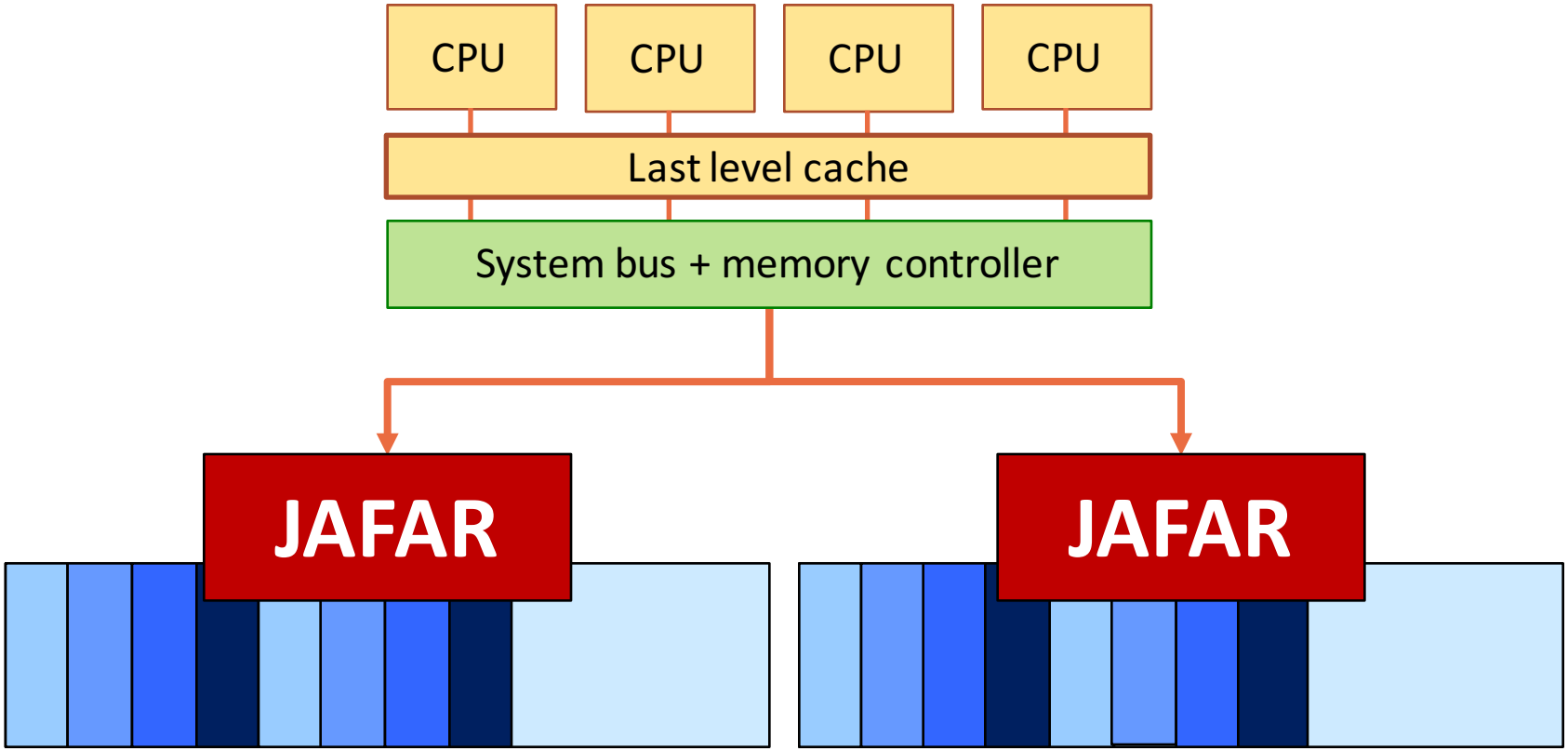
# Handling multiple modules

## Fill up each module first



# Handling multiple modules

## Interleave data across modules



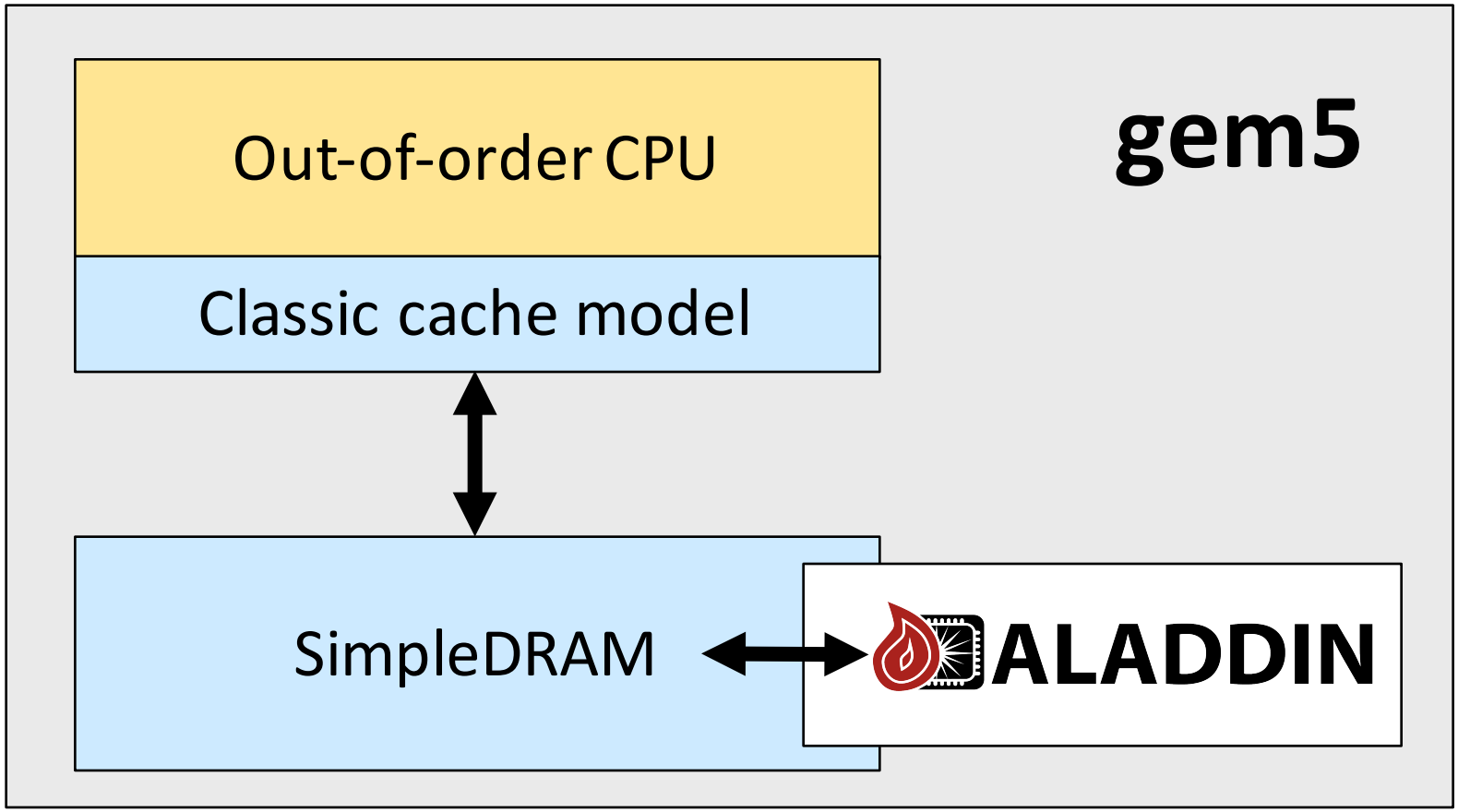
# Coordinating memory access

The CPU and JAFAR cannot simultaneously attempt to access memory.

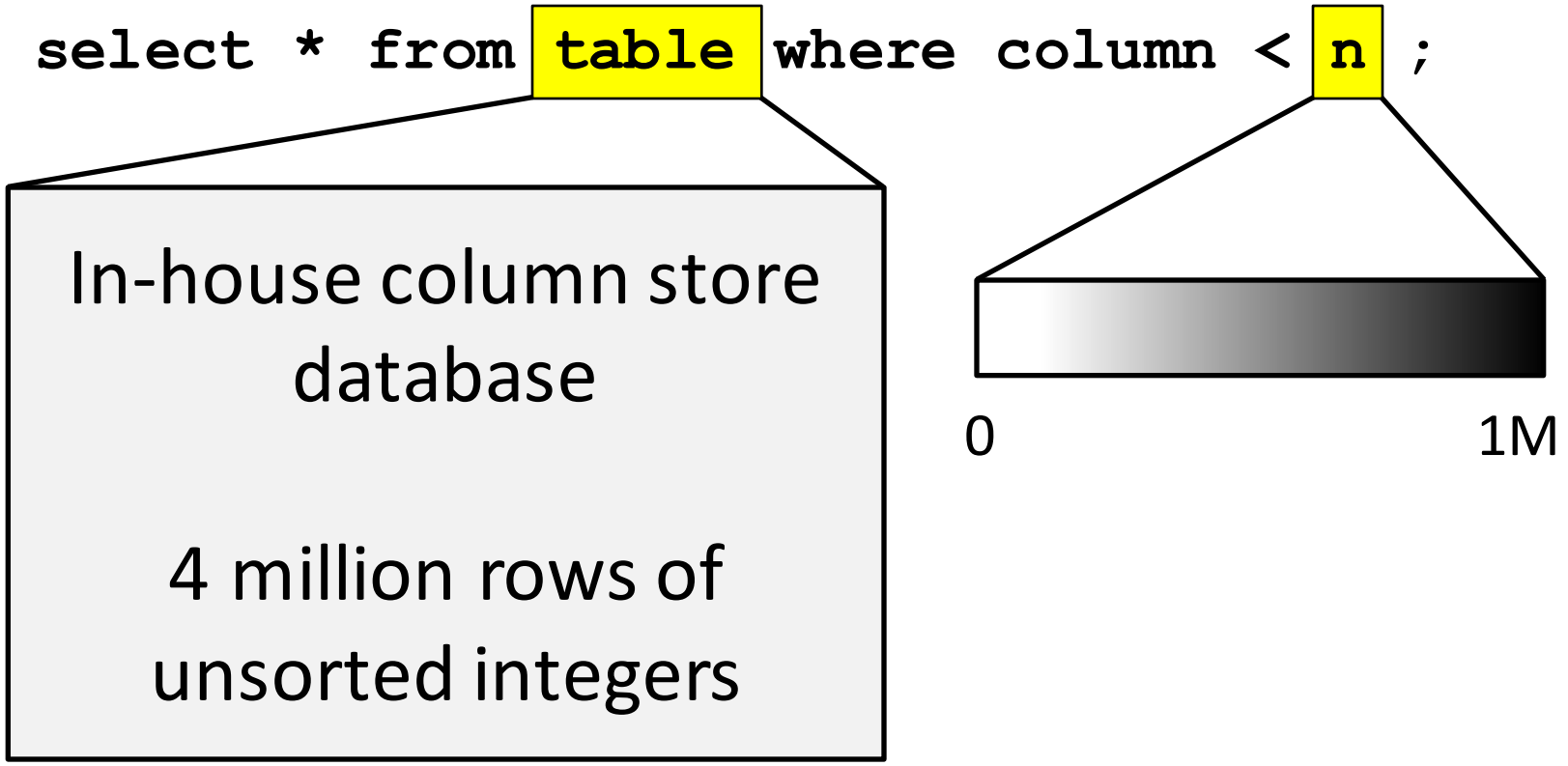
CPU grants JAFAR ownership to a DRAM rank for a period of time.

Possible mechanism: DRAM mode registers

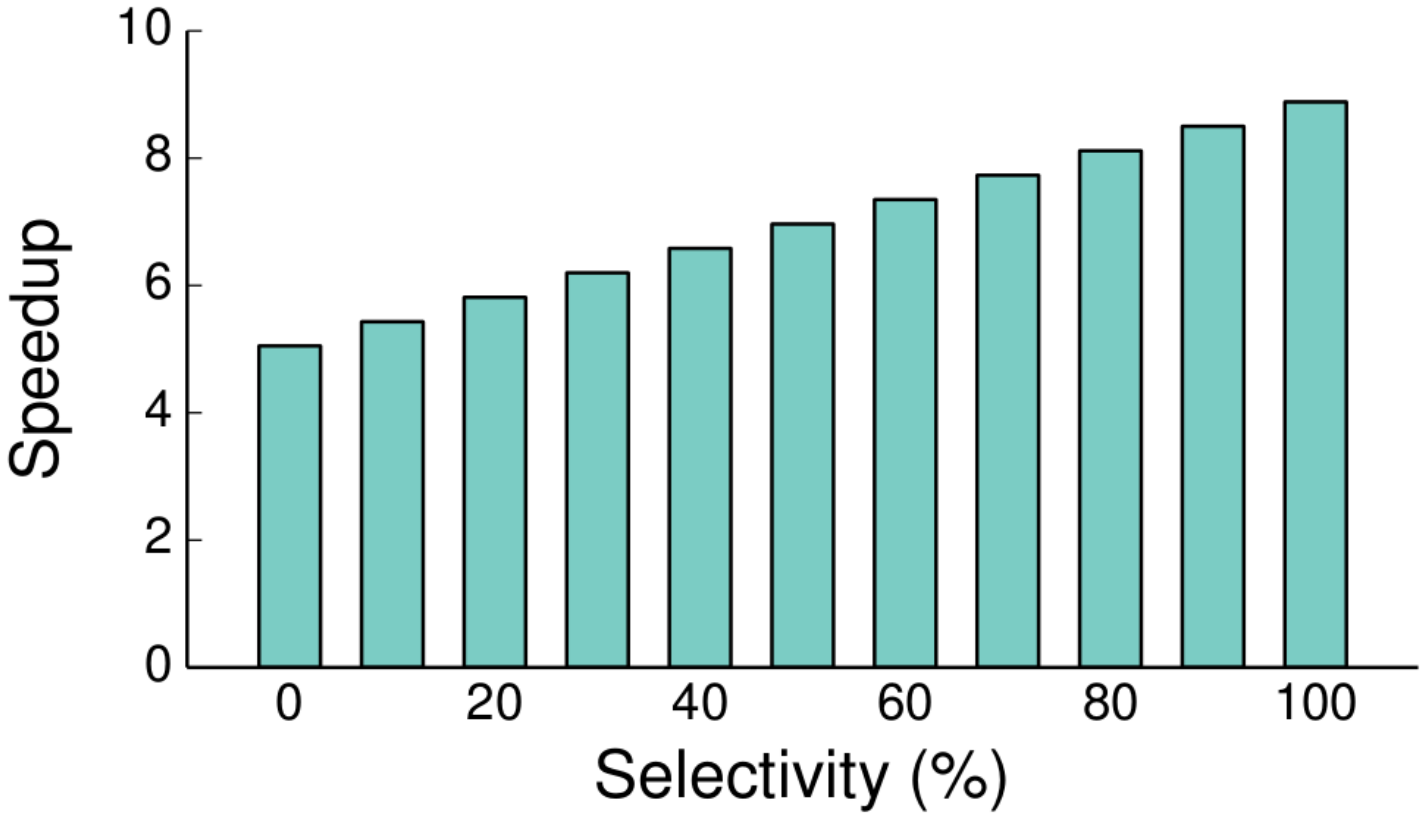
# Experimental setup
## Simulation framework



# Experimental setup
## Queries, input data, and database



# Experimental results

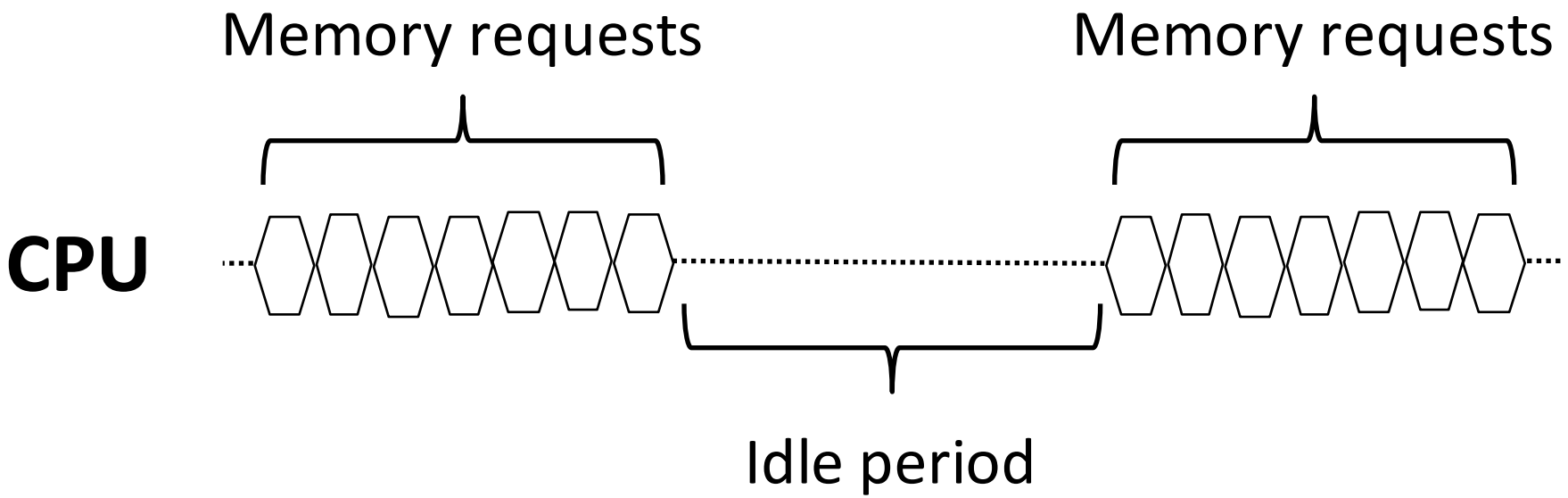


# Memory contention

Scheduling of ownership transfers will be important

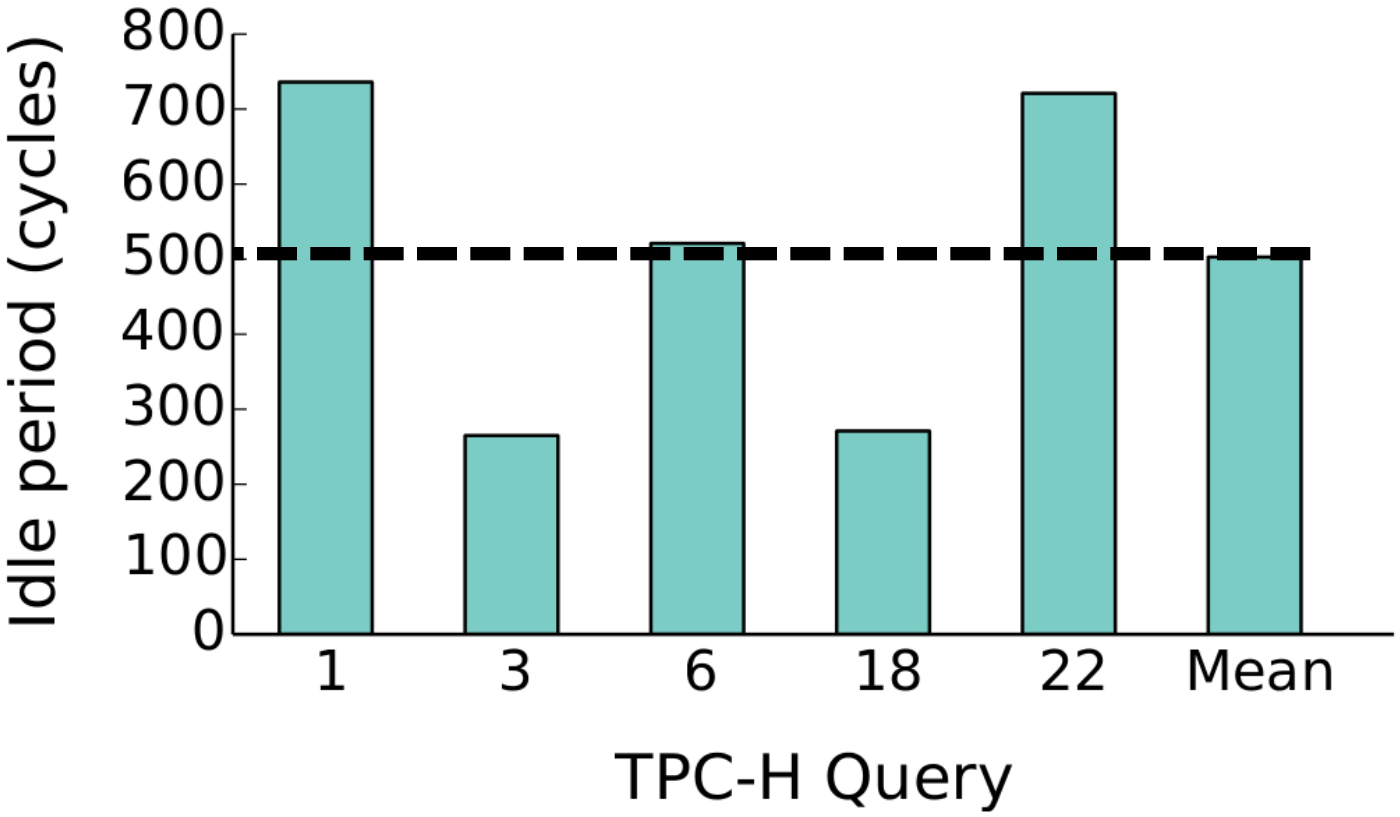
What would JAFAR's performance look like *without* a scheduler?

# Memory contention




JAFAR can execute

# Idle periods on TPC-H



# JAFAR as a framework
## More operators

Aggregations ✓

Projections ✓

Sort ✓

Joins ?

# JAFAR as a framework
## Data types and layouts

Row-stores and hybrids

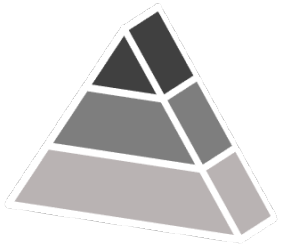
*Multiple filters per row*

*Efficient projections*

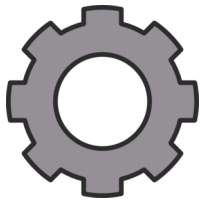
Variable length datatypes

*Process on CPU?*

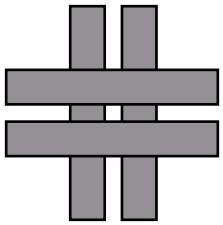
NDP is an exciting opportunity for innovation in data systems



NDP is a promising solution to the memory wall for data systems.



JAFAR provides up to 9x speedup on simple select queries.



JAFAR is built on an extensible framework for accelerating data systems.

Thank you