CS 839: Design the Next-Generation Database

Lecture 13: Smart SSD

Xiangyao Yu

3/3/2020

Announcements

Upcoming deadlines:

- Proposal due: **Mar. 10**

Fill this Google sheet for course project information

- https://docs.google.com/spreadsheets/d/1W7ObfjLqjDChm49GqrLg49x6r4B28-f-PBpQPHX01Mk/edit?usp=sharing

Project Proposal

Use VLDB 2020 format

- https://vldb2020.org/formatting-guidelines.html

The proposal is **1-page** containing the following

- Project name
- Author list
- Abstract (1-2 psaragraphs about your idea)
- Introduction (Why is the problem interesting; what's your contribution)
- Methodology (how do you plan to approach the problem)
- Task-list (Who works on what tasks of the project)
- Timeline (List of milestones and when you plan to achieve them)

Submit proposal by March 10 to https://wisc-cs839-ngdb20.hotcrp.com

Discussion Highlights

Why HBM more successful with GPU than CPU?

- GPU has more computation to saturate HBM bandwidth
- GPU workloads are throughput-bound, not latency bound

Future of storage hierarchy?

- HBM becomes the new DRAM
- Need a universal interface to control the hardware
- Customizable storage solutions
- Another layer: Smart memory
- Some may disappear (e.g., HDD)

APU for database?

- Depends on the price
- Promising because the bandwidth between CPU and GPU increases
- Maybe hard to program

Today's Paper

Query Processing on Smart SSDs: Opportunities and Challenges

Jaeyoung Do[+,#], Yang-Suk Kee*, Jignesh M. Patel[+],
Chanik Park*, Kwanghyun Park[+], David J. DeWitt[#]

[+]University of Wisconsin – Madison; *Samsung Electronics Corp.; [#]Microsoft Corp.

ABSTRACT

Data storage devices are getting "smarter." Smart Flash storage devices (a.k.a. "Smart SSD") are on the horizon and will package CPU processing and DRAM storage inside a Smart SSD, and make that available to run user programs inside a Smart SSD. The focus of this paper is on exploring the opportunities and challenges associated with exploiting this functionality of Smart SSDs for relational analytic query processing. We have implemented an initial prototype of Microsoft SQL Server running on a Samsung Smart SSD. Our results demonstrate that significant performance and energy gains can be achieved by

caches). Various areas of computer science have focused on making this data flow efficient using techniques such as prefetching, prioritizing sequential access (for both fetching data to the main memory, and/or to the processor caches), and pipelined query execution.

However, the boundary between persistent storage, volatile storage, and processing is increasingly getting blurrier. For example, mobile devices today integrate many of these features into a single chip (the SoC trend). We are now on the cusp of this hardware trend sweeping over into the server world. The focus of this project is the integration of processing power and non-volatile

SIGMOD 2013

Today's Agenda

Computation in Memory/Storage

Solid State Drive (SSD)

Query processing on Smart SSDs

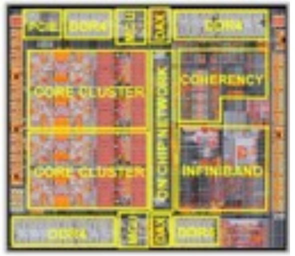
Computation vs. Memory/Storage

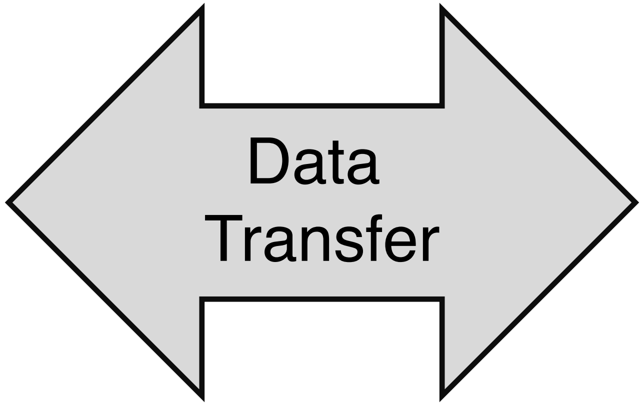
Multicore

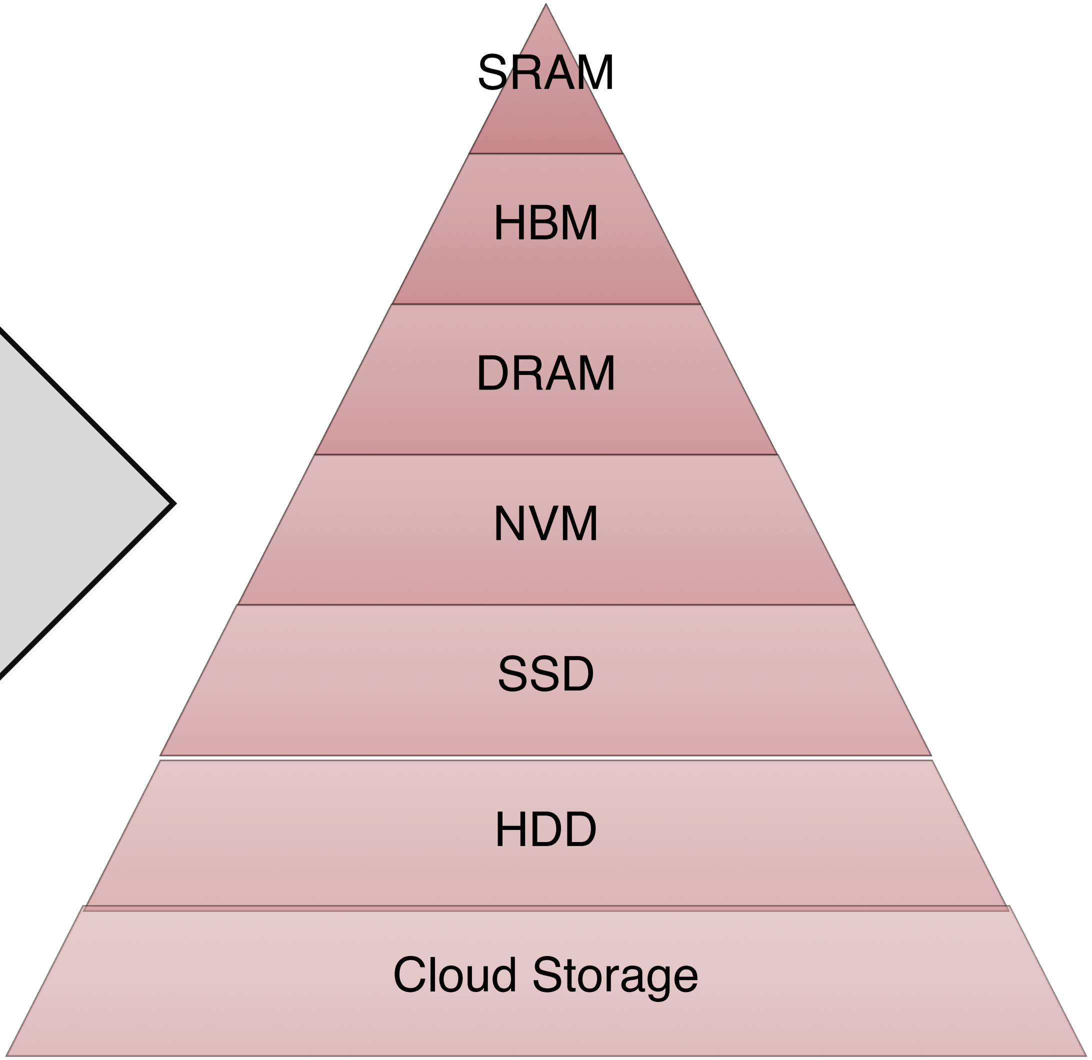
GPU

FPGA

Accelerator



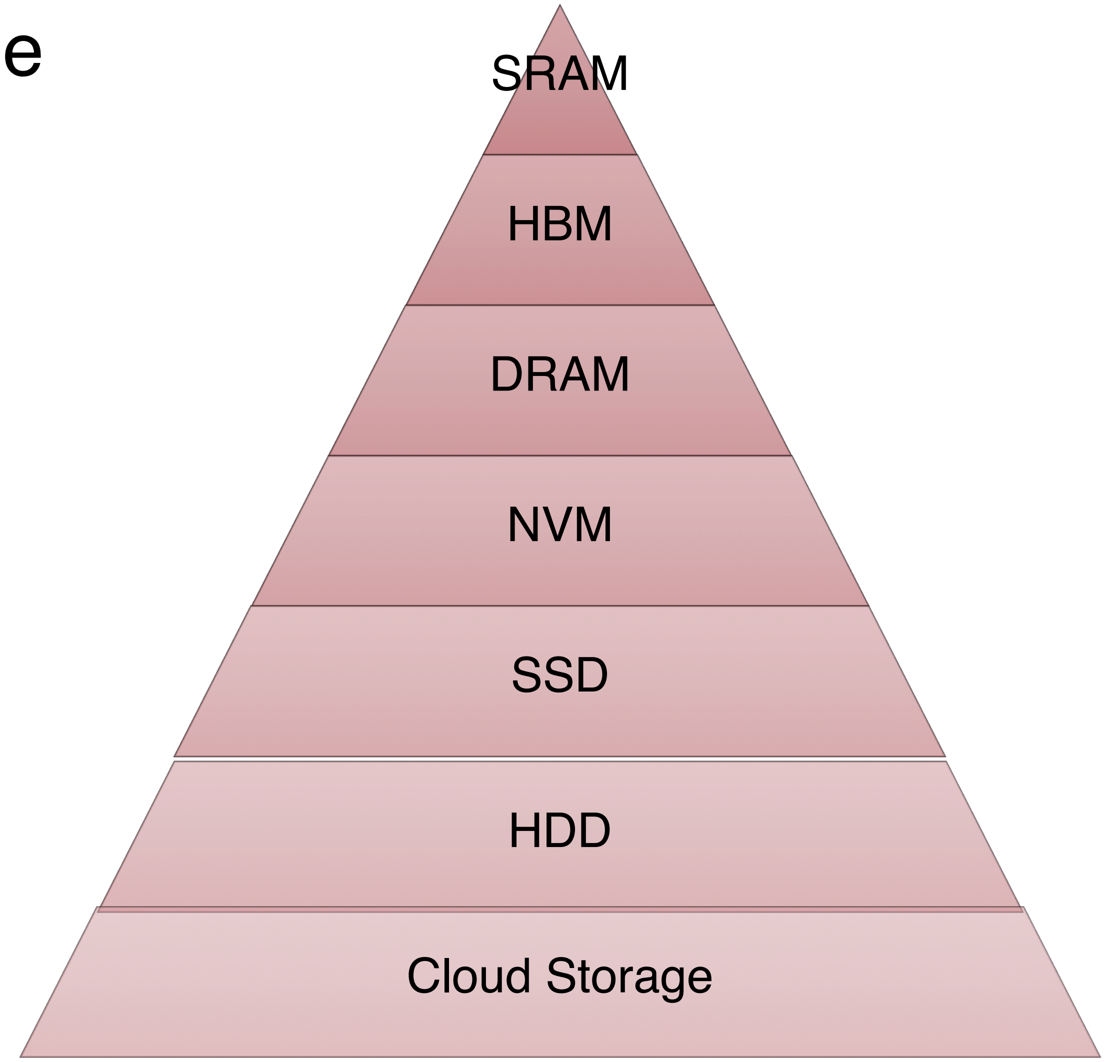




Smart Memory/Storage

Pushing computation to memory/storage

- Process in memory (PIM)
- Smart SSD
- Active Disk
- Intelligent Disk
- AWS S3 Select



Active Disk (CMU), 1998

Embed low-powered processor into each storage device

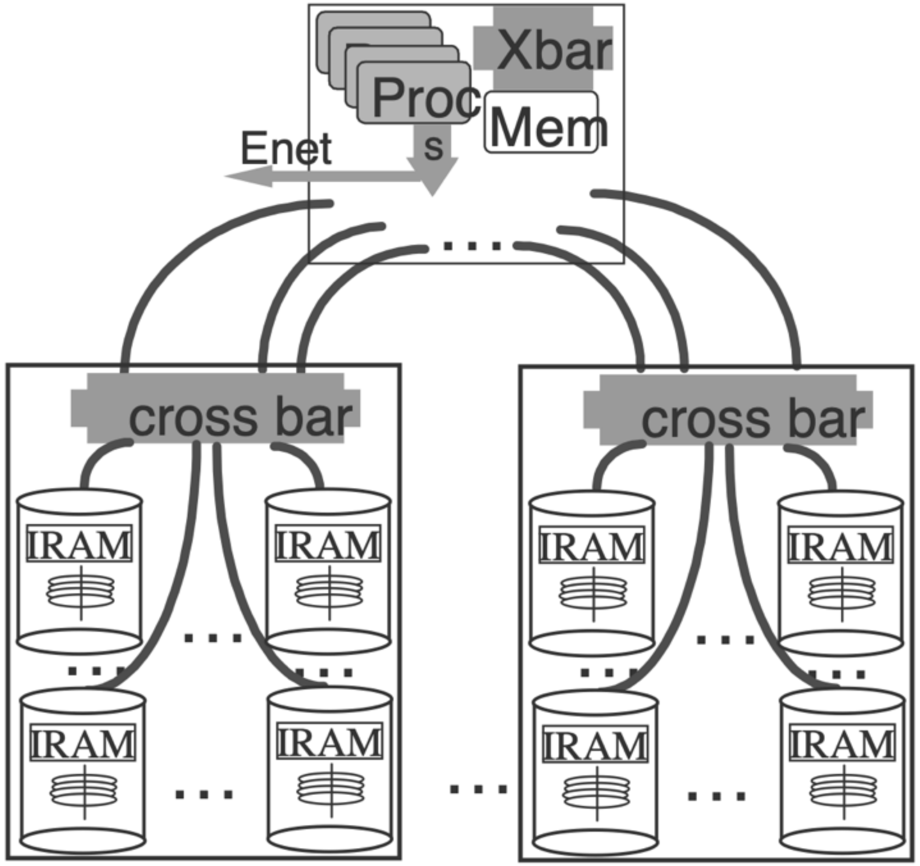
- dramatically reducing data traffic
- exploiting the parallelism in large storage systems

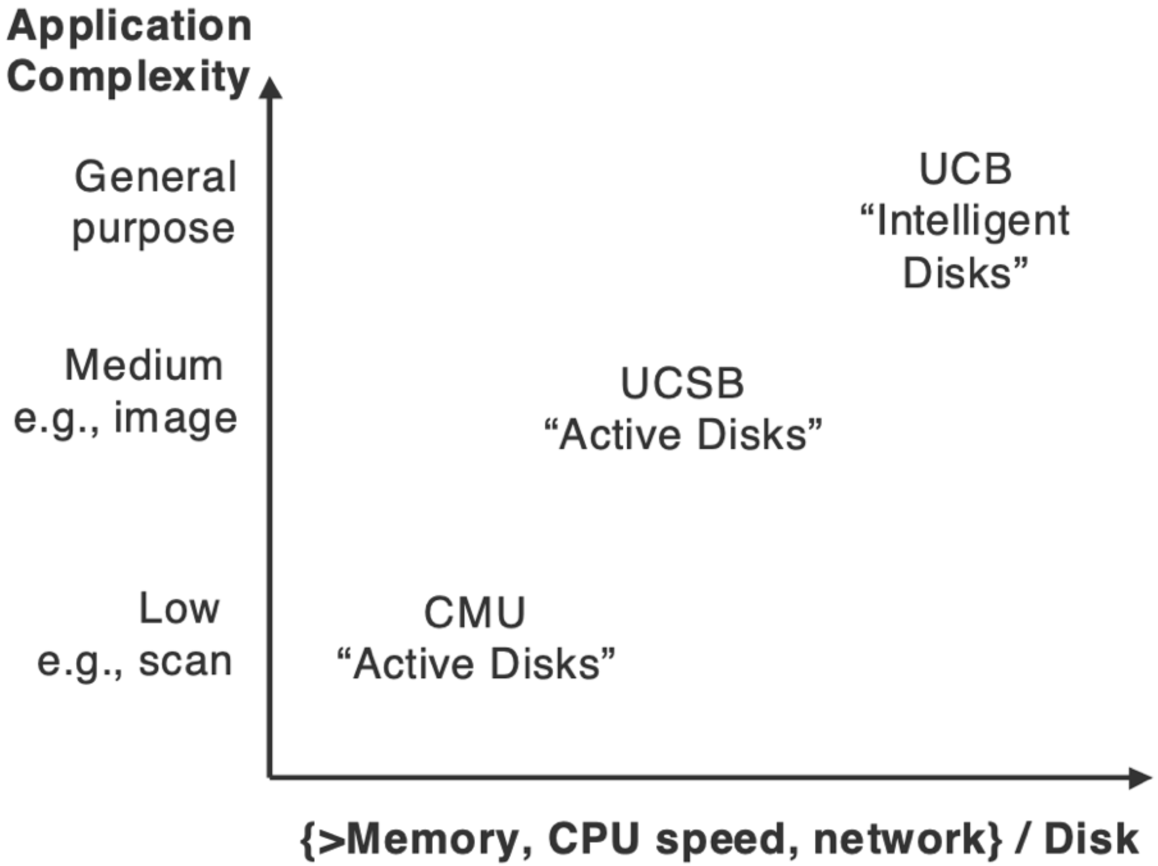
Database operators

- Scan
- Aggregation
- Bloom join

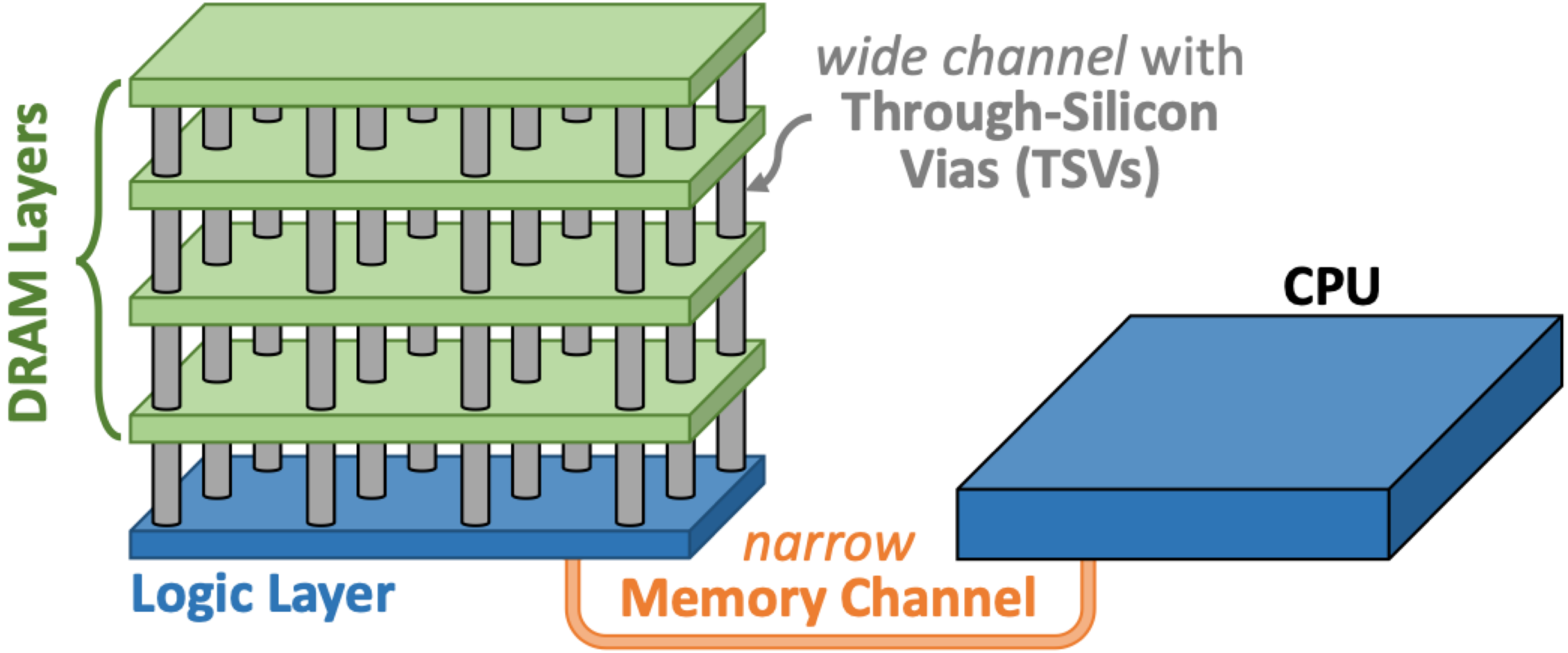
2x speedup based on a prototype

Intelligent Disk (Berkeley), 1998



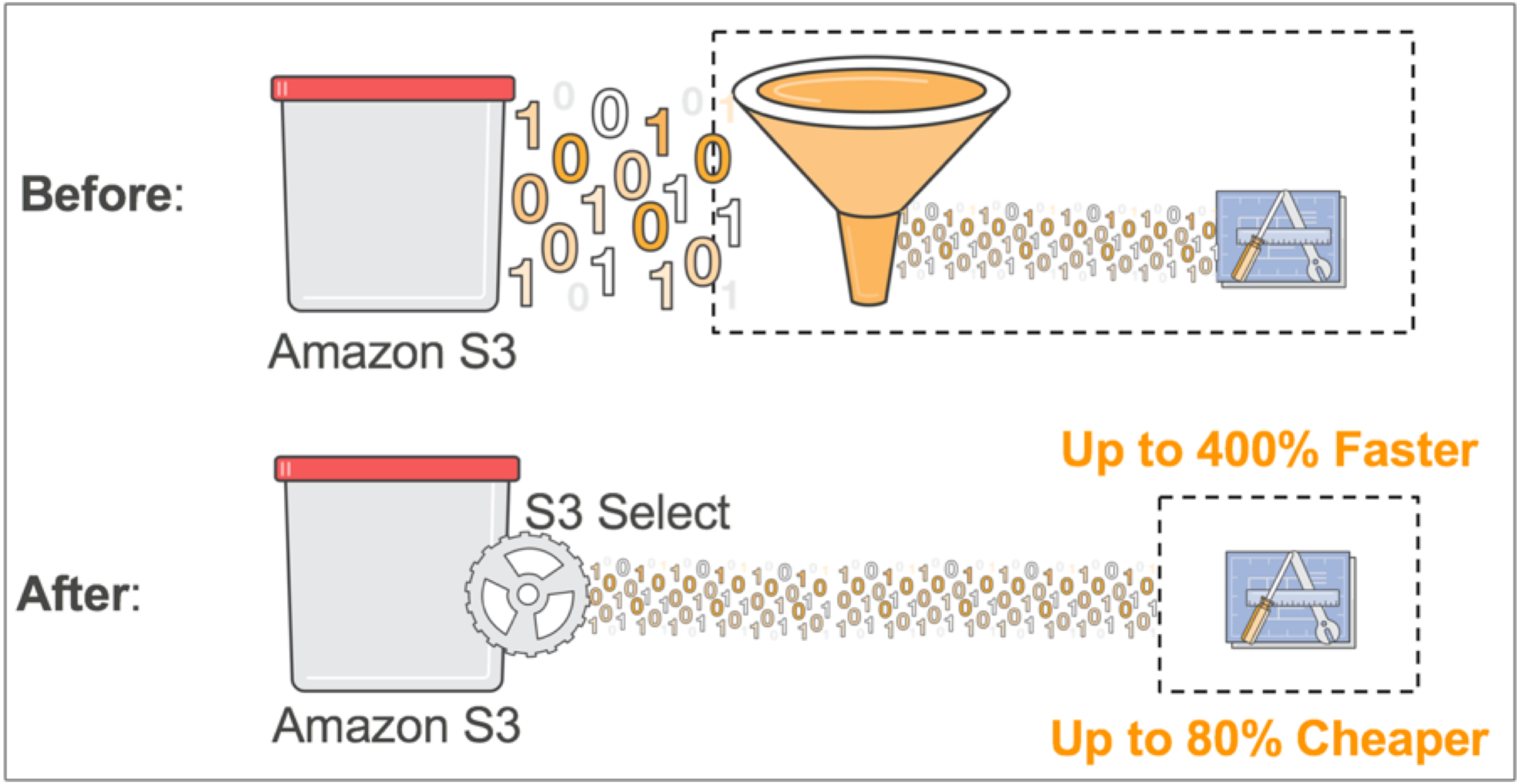


Process in Memory

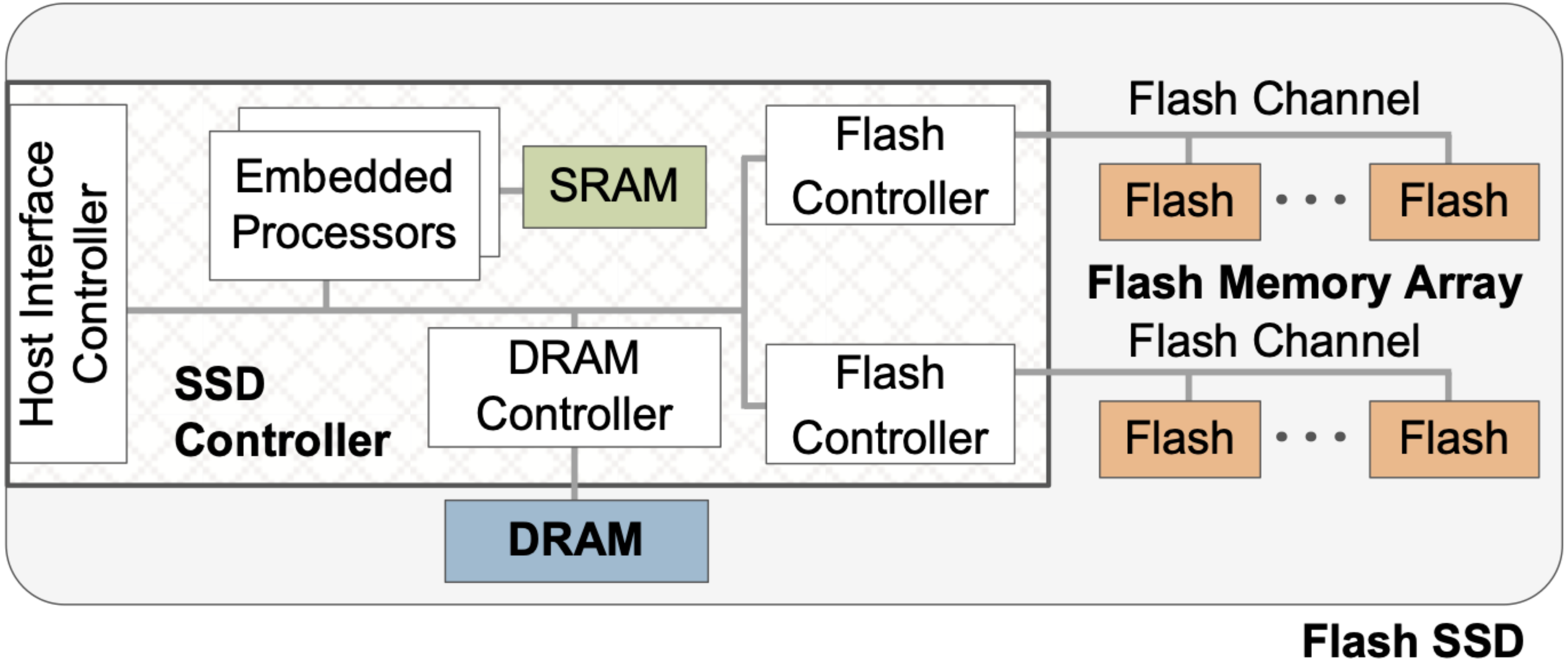


More on this next lecture

AWS S3 Select



Solid State Drive (SSD)



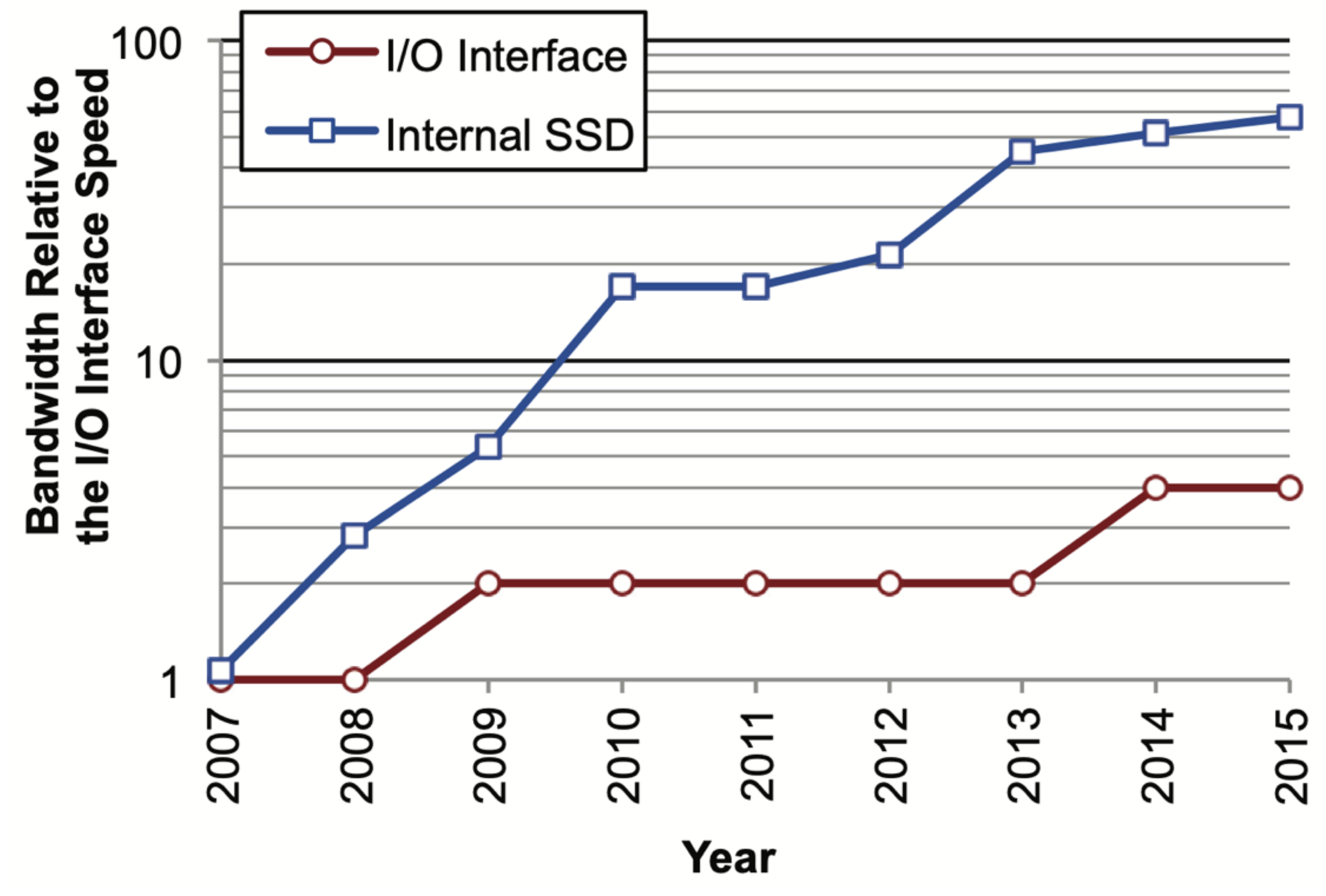
Flash Translation Layer (FTL)

- Bad Block Management
- Map logical addresses with physical addresses
- Wear-levelling
- Garbage Collection

SSD Performance

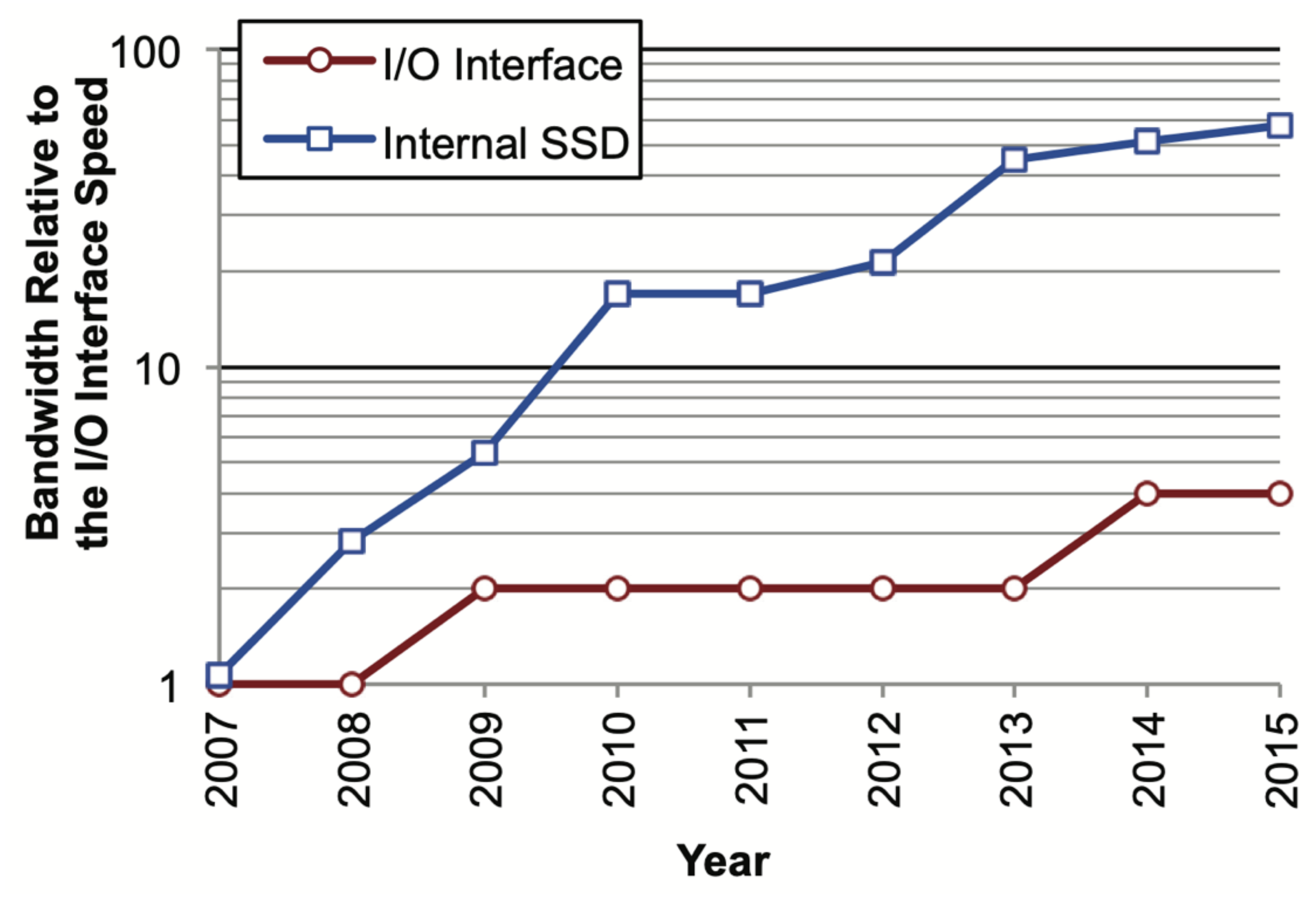
	SATA SSD	NVMe SSD	Optane	DDR4 DRAM
Read Bandwidth	530 MB/s	2150 MB/s	6600 MB/s	25.6 GB/s
Write Bandwidth	500 MB/s	1550 MB/s	2300 MB/s	25.6 GB/s

Query processing on Smart SSDs



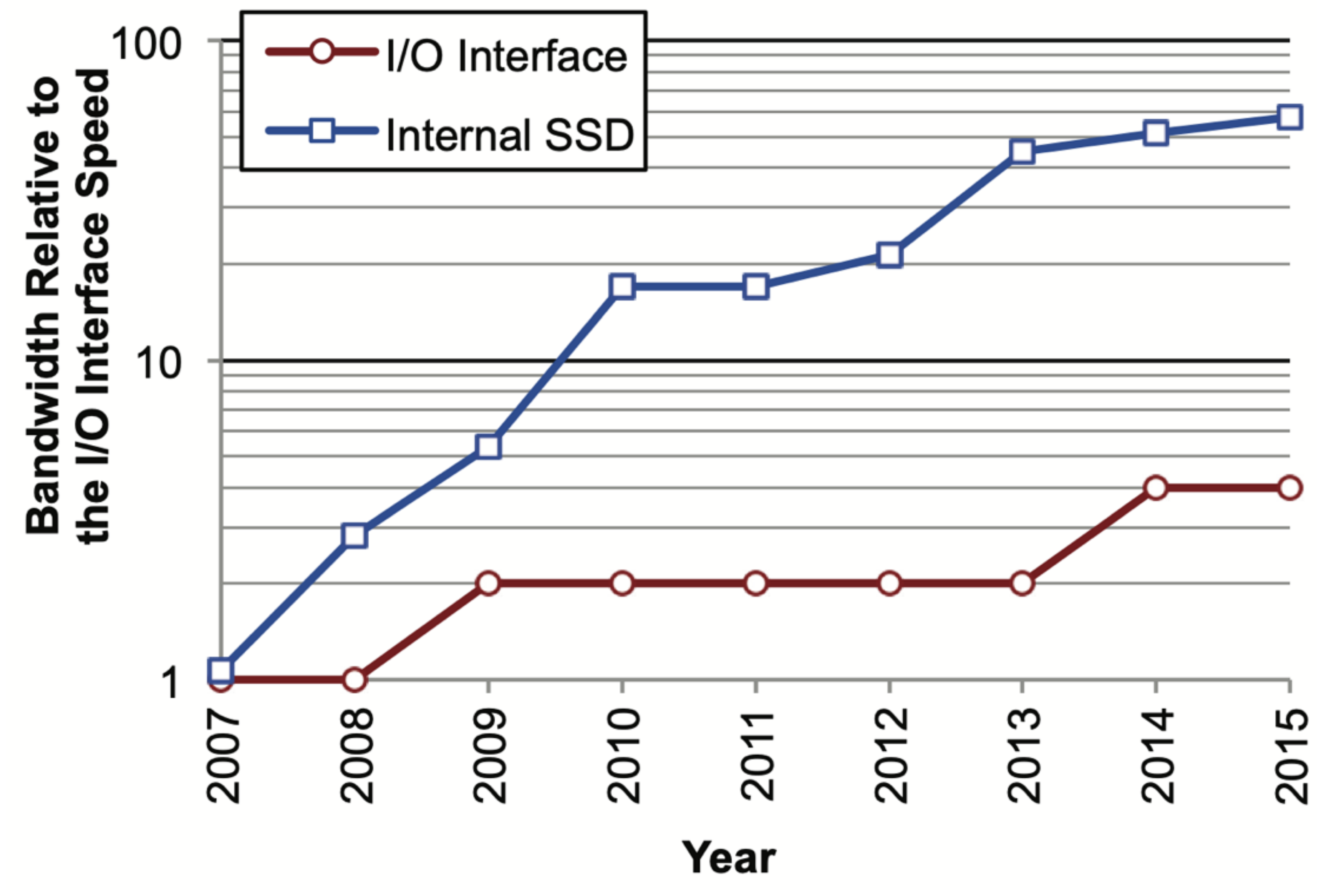
- Internal bandwidth larger than external bandwidth

Query processing on Smart SSDs



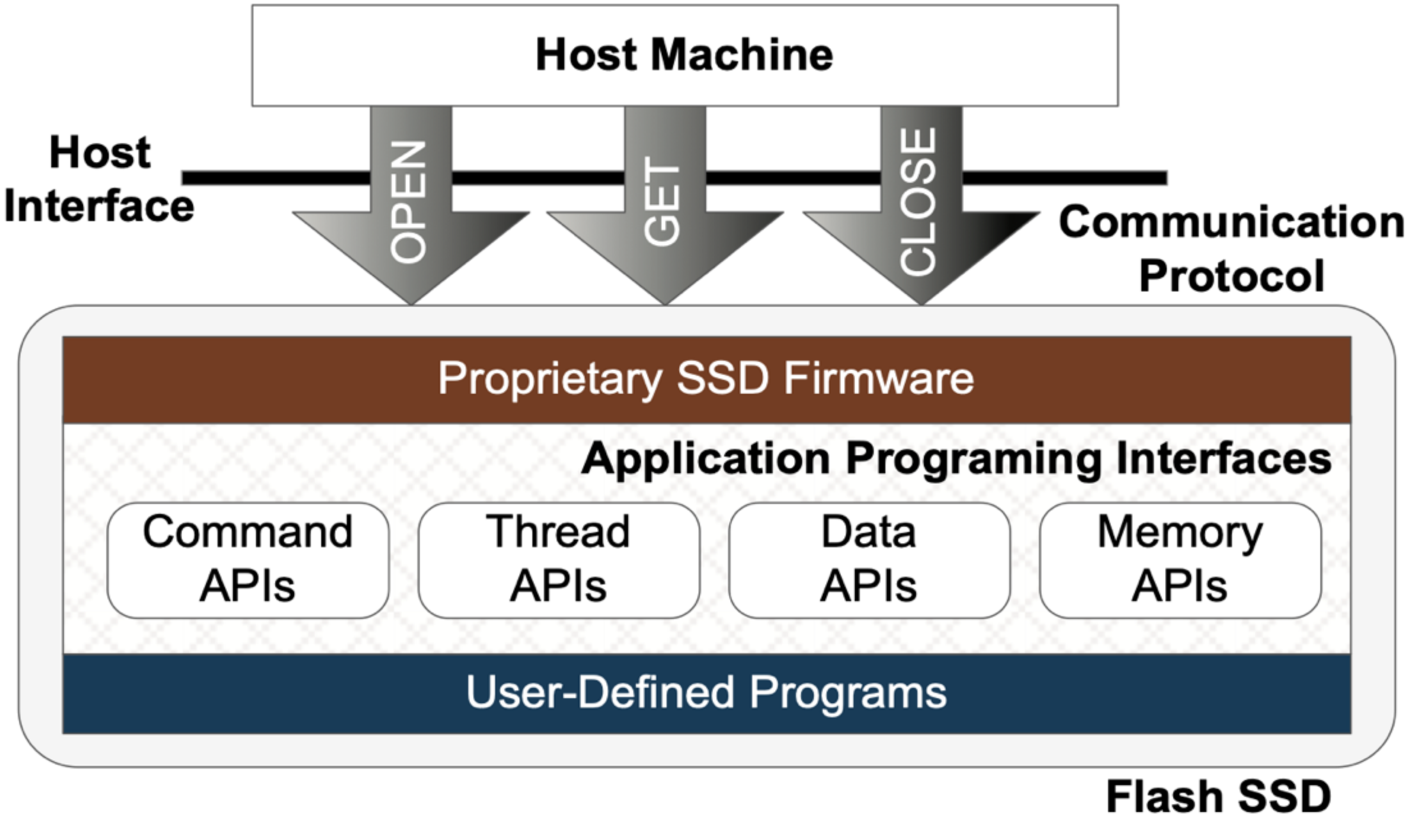
- Internal bandwidth larger than external bandwidth
- In-SSD processor is less powerful and cheaper, Smart SSD may improve overall cost/performance

Query processing on Smart SSDs



- Internal bandwidth larger than external bandwidth
- In-SSD processor is less powerful and cheaper, Smart SSD may improve overall cost/performance
- Reduce energy consumption

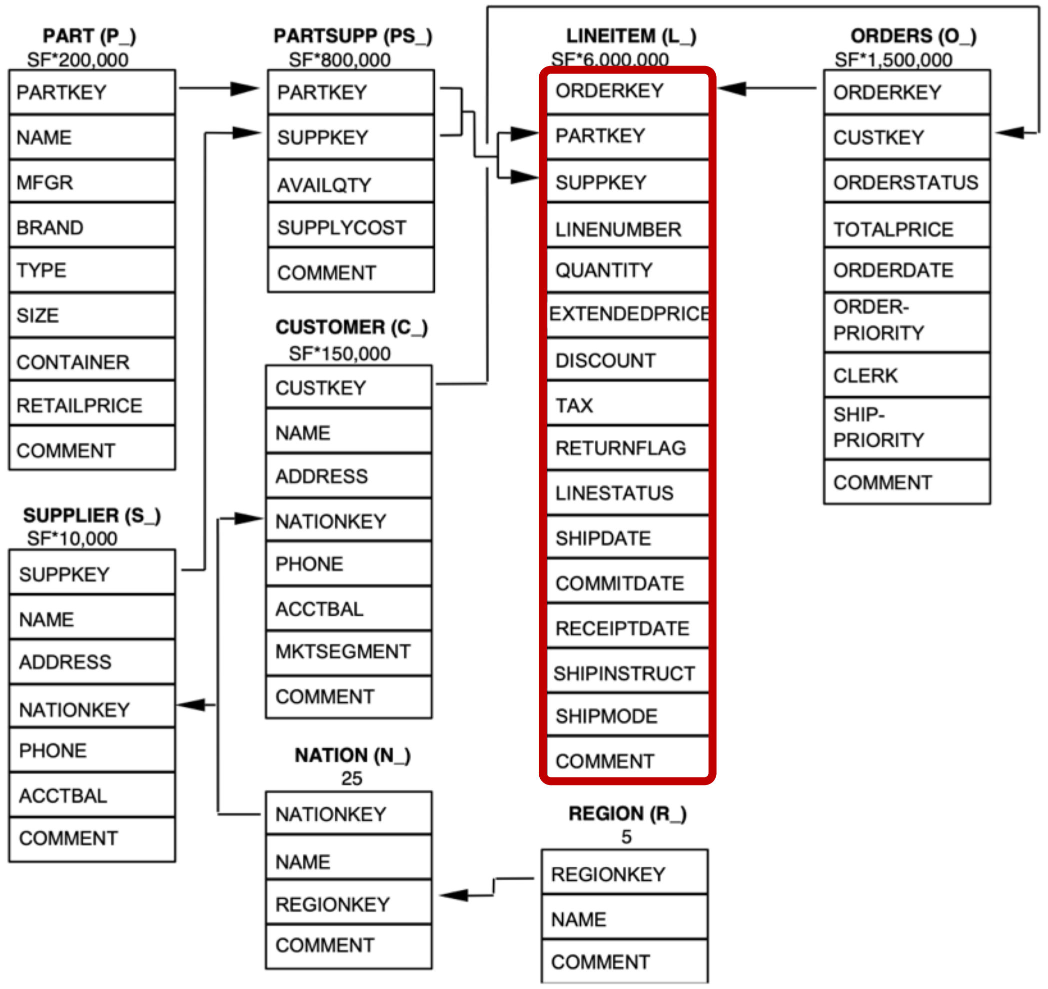
Runtime Framework



- OPEN/CLOSE to start/end a session
  - Allocate threads and memory
- Get
  - Monitor the status of the program and retrieve results
  - 10ms polling interval

Evaluation – Data Set (TPC-H)

TPC-H (lineitem)



- Fixed-length char string for the variable-length column (L_COMMENT)
- All decimal numbers were multiplied by 100 and stored as integers
- All dates converted to numbers of days

Evaluation – Data Set (Synthetic)

Synthetic4: 4 integer columns

Synthetic16: 16 integer columns

Synthetic64: 64 integer columns

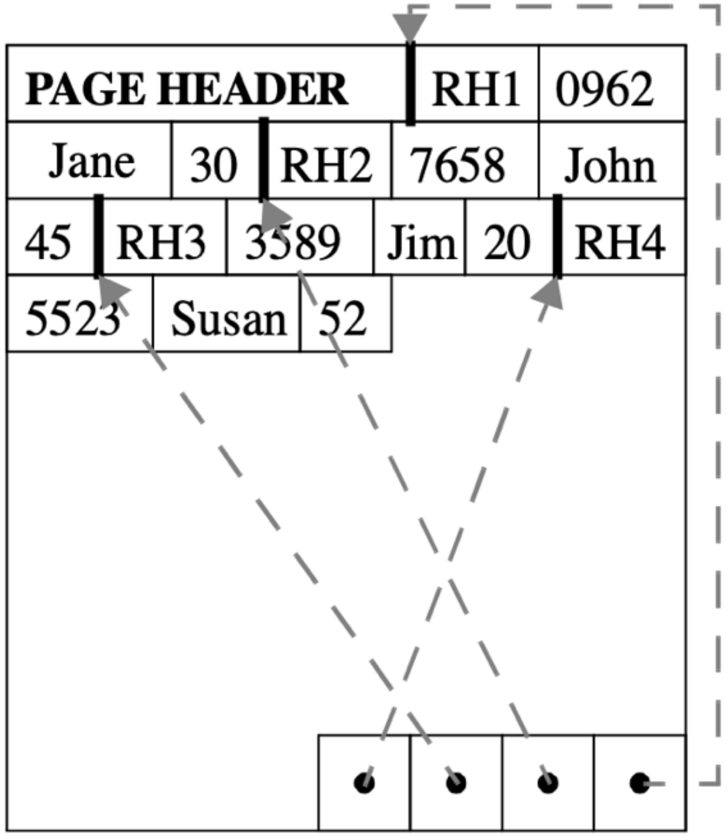
Page Layout – NSM

- N-ary Storage Model (NSM)

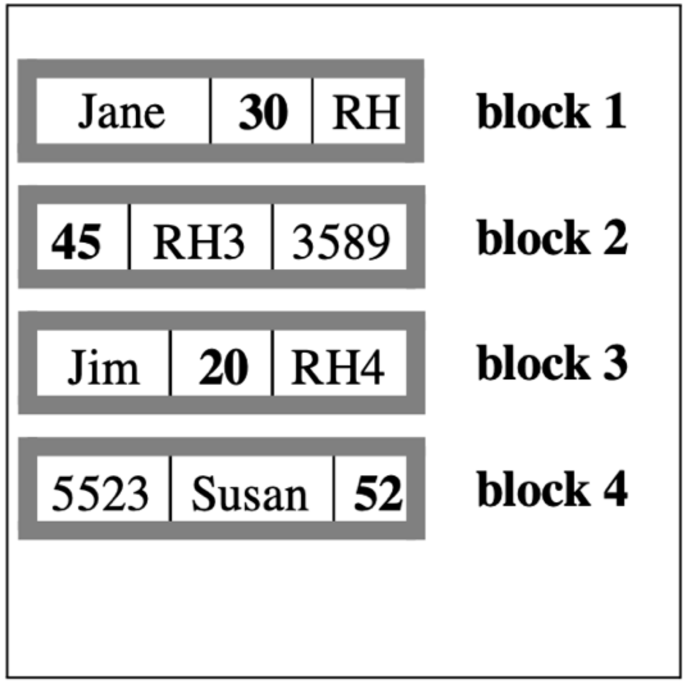
RELATION R

RID	SSN	Name	Age
1	0962	Jane	30
2	7658	John	45
3	3859	Jim	20
4	5523	Susan	52
5	9743	Leon	43
6	0618	Dan	37

NSM PAGE



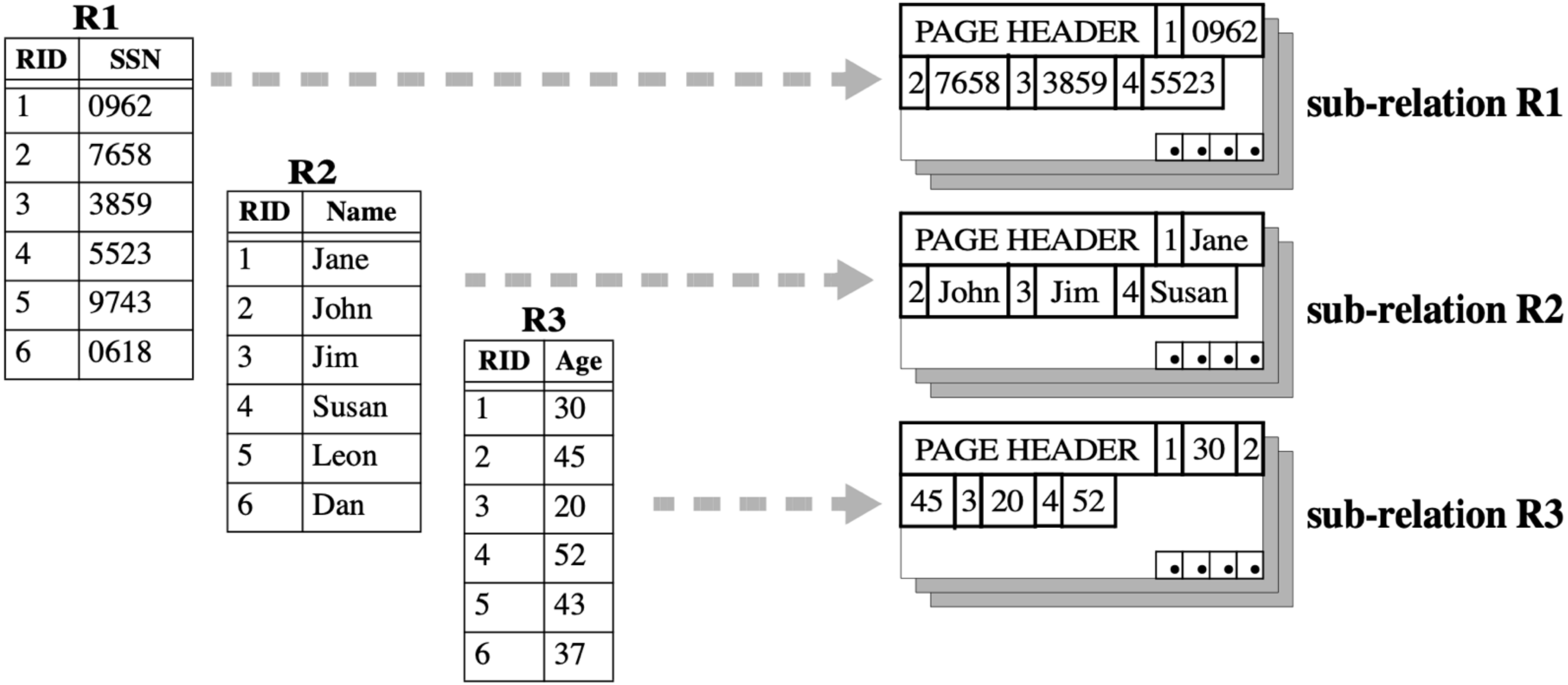
CACHE



Source: **Data Page Layouts for Relational Databases on Deep Memory Hierarchies, VLDB Journal, 2002**

Page Layout – DSM

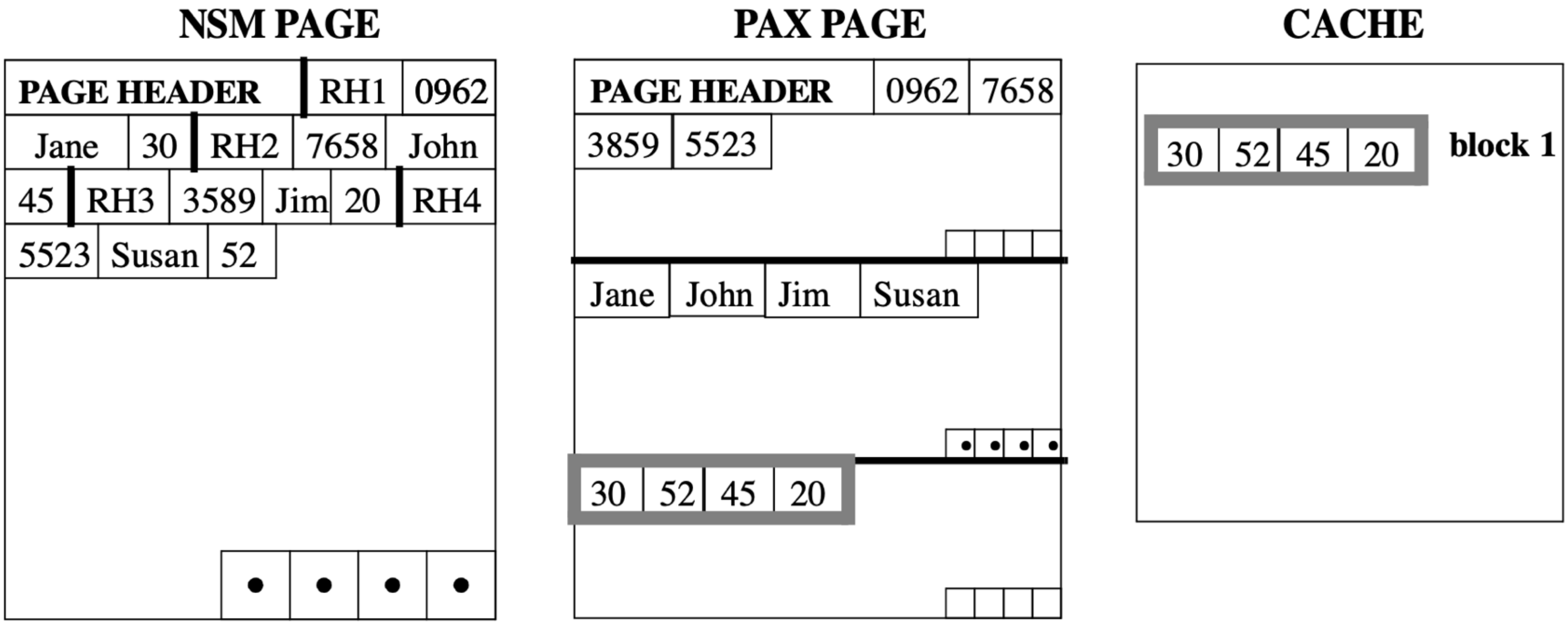
Decomposition Storage Model (DSM)



Source: **Data Page Layouts for Relational Databases on Deep Memory Hierarchies, VLDB Journal, 2002**

Page Layout – PAX

Partition Attributes Across (PAX)



Source: **Data Page Layouts for Relational Databases on Deep Memory Hierarchies, VLDB Journal, 2002**

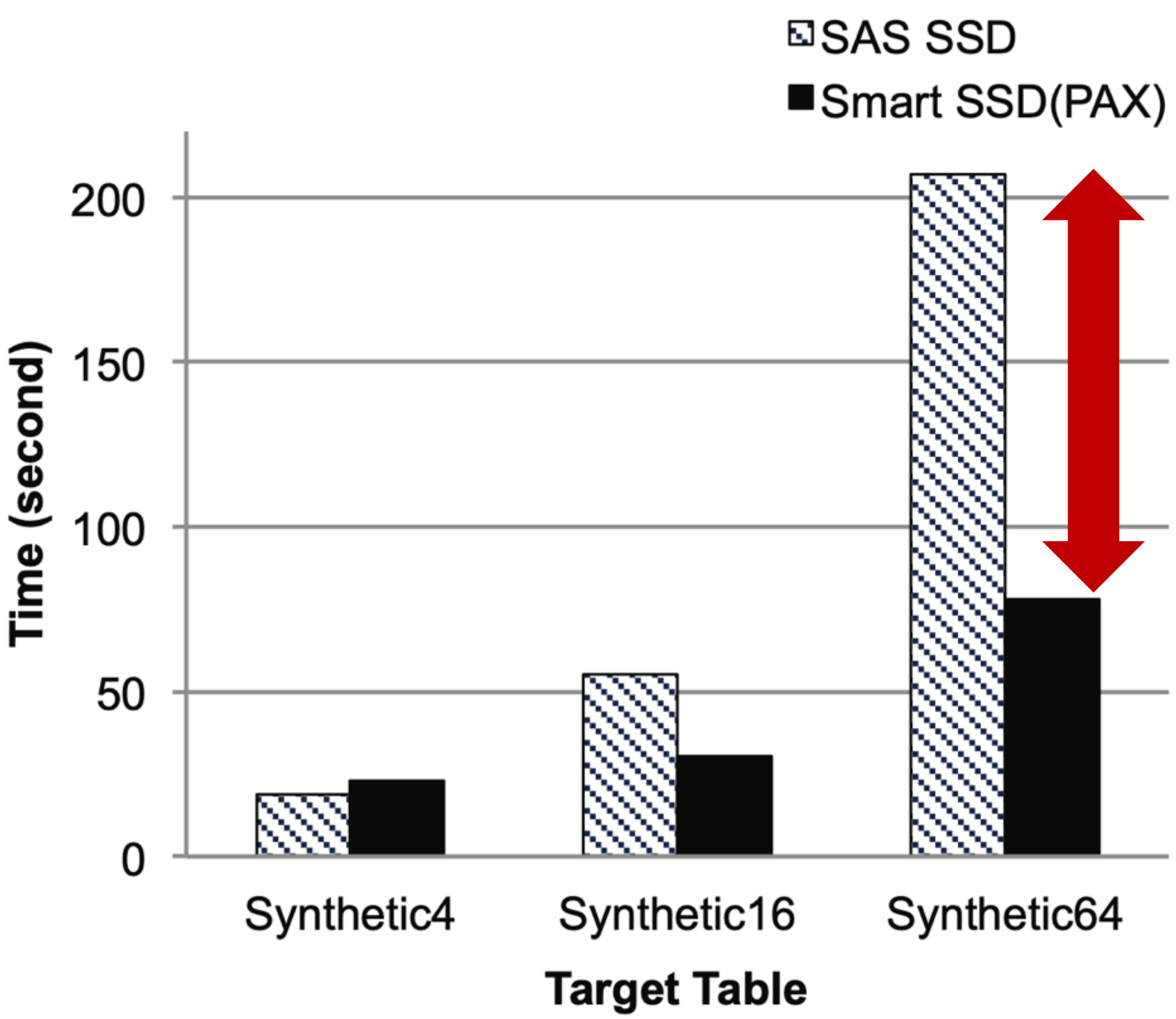
Maximum Sequential Bandwidth

Table 1: Maximum sequential read bandwidth with 32-page (256KB) I/Os.

	SAS HDD	*SAS SSD*	*(Internal) Smart SSD*
Seq. Read (MB/sec)	80	550	1,560

Maximum potential gain is 1560 / 550 = **2.8**

Selection Query

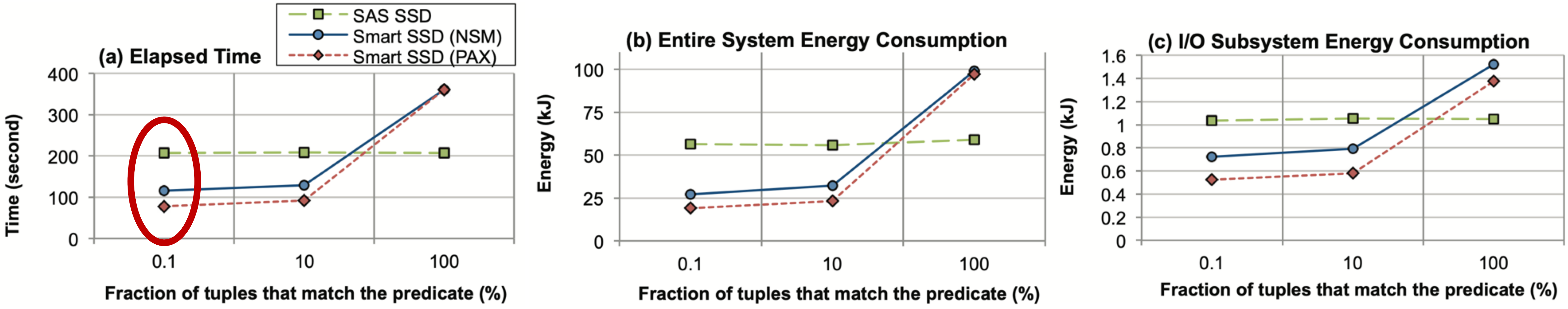


SELECT SecondColumn
FROM SyntheticTable
WHERE FirstColumn < [VALUE]

Selectivity = 0.1%

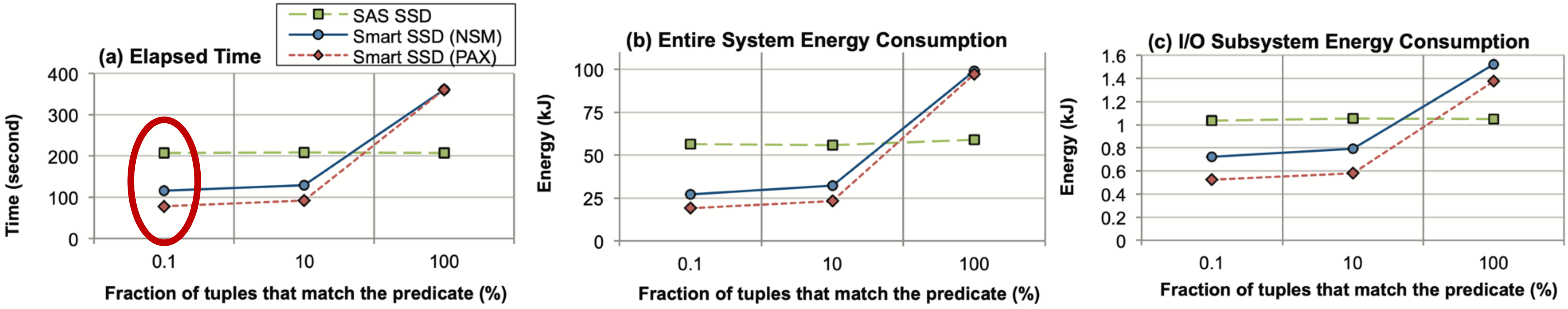
Speedup = 2.6X

Selection Query (Synthetic64)



PAX better than NSM

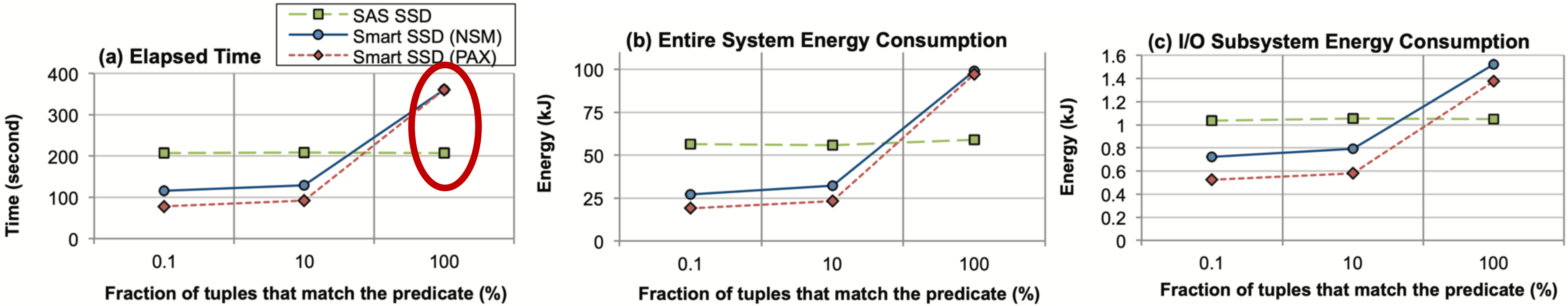
Selection Query (Synthetic64)



PAX better than NSM

2.6x speedup

Selection Query (Synthetic64)



PAX better than NSM

2.6x speedup

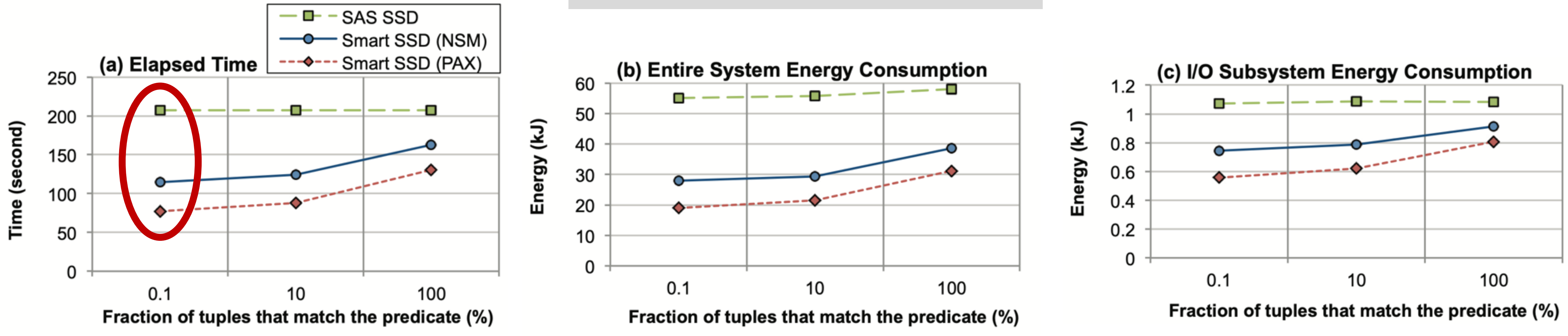
Embedded CPU becomes bottleneck

Selection with Aggregation (Synthetic64)

```
SELECT AVG (SecondColumn)
FROM SyntheticTable
WHERE FirstColumn < [VALUE]
```

Selection with Aggregation (Synthetic64)

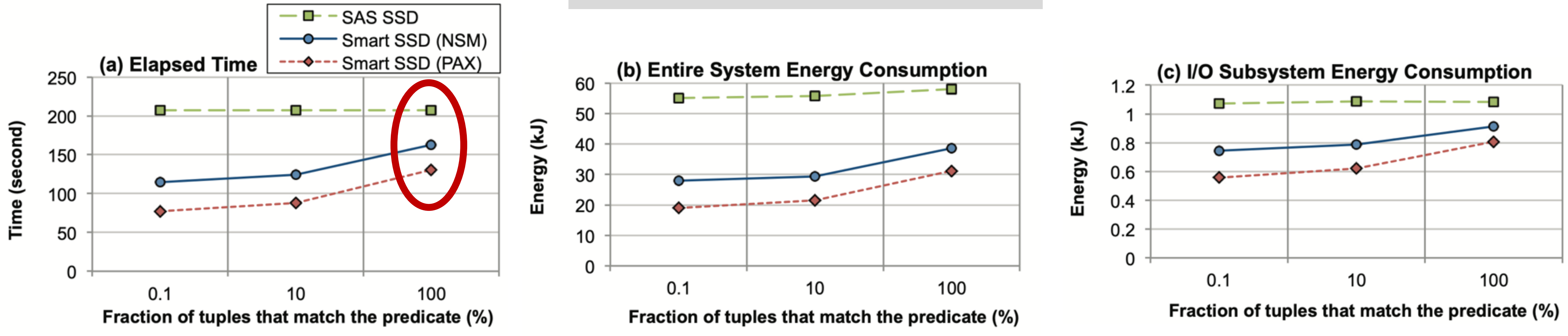
SELECT **AVG** **(**SecondColumn)
FROM SyntheticTable
WHERE FirstColumn < [VALUE]



2.7x speedup

Selection with Aggregation (Synthetic64)

SELECT AVG (SecondColumn)
FROM SyntheticTable
WHERE FirstColumn < [VALUE]

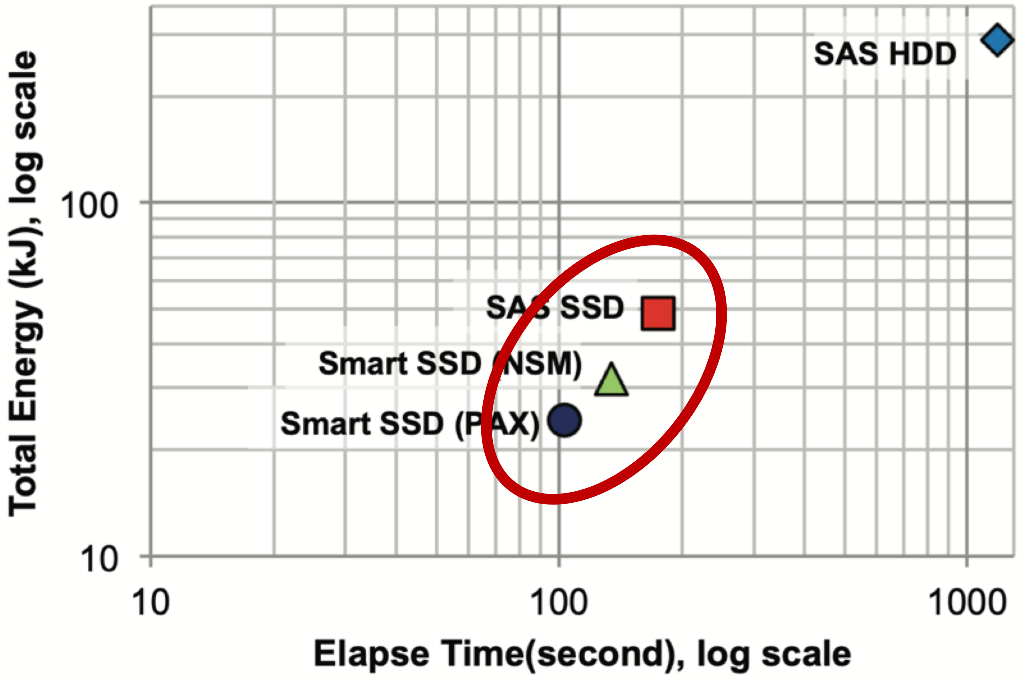


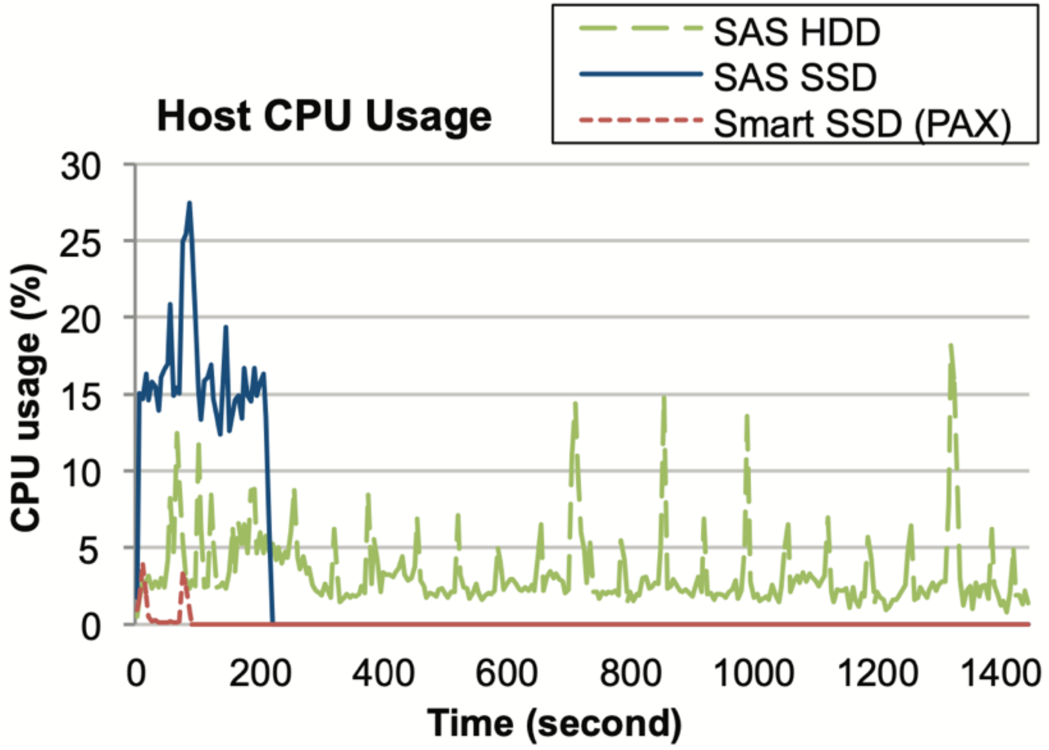
2.7x speedup

Less data transfer with aggregation

TPC-H Query 6

```
SELECT SUM (EXTENDEDPRICE*DISCOUNT)
FROM LINEITEM
WHERE SHIPDATE >= 1994-01-01 AND
      SHIPDATE < 1995-01-01 AND
      DISCOUNT > 0.05 AND
      DISCOUNT < 0.07 AND
      QUANTITY < 24
```





Discussion

1. Processing capabilities inside the Smart SSD becomes a performance bottleneck
2. Needs better development environment
3. Handle dirty data in buffer pool
4. Database internals (e.g., query optimization, caching vs. pushdown)

Samsung SmartSSD Today

Source: https://www.nimbix.net/wp-content/uploads/2020/02/Digital_SmartSSD_Solution_Brief_03.pdf

Samsung SmartSSD Today

SmartSSD Computational Storage Drive Specification*

Form Factor	2.5" (U.2)	
Storage Capacity	3.84TB (other capacities coming soon)	
Host Interface	Single-port PCIe Gen 3x4	
Spec Compliance	NVMe spec rev. 1.3, PCIe base specification rev. 3.0, NVMe Management Interface (NVMe MI) 1.0	
Programmable Hardware Accelerator (FPGA)	Xilinx Kintex Ultrascale+ KU15P FPGA	
	System Logic Cells	1.143 Million
	Available LUTs for acceleration tasks	Approx. 300k
	DSP Slices	1,968
	Internal Distributed RAM	34.6 Mbit
	Internal UltraRAM	36.0 Mbit
	Accelerator-dedicated DRAM	4 GByte DDR4 SDRAM @ 2400 Mbps
	Speed Grade	-2LE

Apply for the free SmartSSD Cloud trial

SmartSSD Cloud Trial

https://www.nimbix.net/samsungsmartssd

Source: https://www.nimbix.net/wp-content/uploads/2020/02/SmartSSD_ProductBrief_12.pdf

Summary

Gap between external and internal bandwidth determines the potential performance improvement

Smart SSD prototype used in the paper delivers 2.7x speedup

Summary

Gap between external and internal bandwidth determines the potential performance improvement

Smart SSD prototype used in the paper delivers 2.7x speedup

> *“ The history of DBMS research is littered with innumerable proposals to construct hardware database machines to provide high performance operations. In general these have been proposed by hardware types with a clever solution in search of a problem on which it might work.”*
>
> – Michael Stonebraker

[1] M. Stonebraker, editor. *Readings in Database Systems*, second edition, Morgan Kaufmann Publishers, San Francisco, 1994, p. 603.

Smart SSD – Q/A

How to support join?

Support for UDF? How hard to program Smart SSD?

Follow-up work on Smart SSD?

Are Smart SSDs widely deployed today?

Why modify LINEITEM?

Does Smart SSD still make sense with fast IO?

Group Discussion

What's your opinion on Prof. Stonebraker's comment?

"The history of DBMS research is littered with innumerable proposals to construct hardware database machines to provide high performance operations. In general these have been proposed by hardware types with a clever solution in search of a problem on which it might work."

How does fast IO/network affect the design of smart memory/storage in general?

Besides filter and aggregation, how can other operators benefit from smart SSD? (E.g., Join, group-by, sort, etc.)

Before Next Lecture

Submit discussion summary to https://wisc-cs839-ngdb20.hotcrp.com

- **Deadline: Wednesday 11:59pm**

Submit review for

- Database Processing-in-Memory: An Experimental Study
- [Optional] The Mondrian Data Engine