Funded by
the European Union

CHORYS

Near-Data Processing: Early Insights from CHORYS

Philippe Bonnet, DIKU, bonnet@di.ku.dk

UNIVERSITY OF COPENHAGEN

An old idea

Put Everything in Future (Disk) Controllers (it's not "if", it's "when?")

Jim Gray

http://www.research.Microsoft.com/~Gray



Acknowledgements:

Dave Patterson explained this to me a year ago

Kim Keeton, **Erik Riedel** } **Helped me sharpen these arguments**

Catharine Van Ingen

1

Talk at NASD, 6/9/98



NEAR-DATA PROCESSING: INSIGHTS FROM A MICRO-46 WORKSHOP

Rajeev Balasubramonian, University of Utah
Jichuan Chang, Google
Troy Manning, Micron
Jaime H. Moreno, IBM Thomas J. Watson Research Center
Richard Murphy, Micron
Ravi Nair, IBM Thomas J. Watson Research Center
Steven Swanson, University of California, San Diego

AFTER A DECADE-LONG DORMANCY, INTEREST IN NEAR-DATA PROCESSING (NDP) HAS SPIKED. A WORKSHOP ON NDP WAS ORGANIZED AT MICRO-46 AND WAS WELL ATTENDED. GIVEN THE INTEREST, THE ORGANIZERS AND KEYNOTE SPEAKERS HAVE ATTEMPTED TO CAPTURE THE KEY INSIGHTS FROM THE WORKSHOP FOR WIDER DISSEMINATION. THIS ARTICLE DESCRIBES WHY NDP IS COMPELLING TODAY AND IDENTIFIES UPCOMING CHALLENGES IN REALIZING ITS POTENTIAL.

Processing in large-scale systems is shifting from the traditional computing-centric model successfully used for many decades into one that is more data centric. This transition is driven by the evolving nature of computing, which is no longer dominated by the execution of arithmetic and logic calculations but instead by the handling of large data volume and the cost of moving data to the locations where computations are performed. The computing-centric model where data lives on disk, or even tape, and moves as needed to a central computing engine across a deep storage hierarchy is sufficient when computational aspects dominate data movement aspects. In contrast, in the data-centric model, data lives in different storage levels within the hierarchy, with processing engines surrounding the data and operating on such data without moving it across the system.

The trend toward big data is leading to changes in the computing paradigm, and in particular to the notion of moving computation to data, in what we call the *near-data processing* (NDP) approach. Data movement impacts performance, power efficiency, and reliability, three fundamental attributes of a system. NDP seeks to minimize data movement by computing at the most appropriate location in the hierarchy, considering the location of the data and the information that needs to be extracted from that data. Thus, in NDP, computation can be performed right at the data's home, either in caches, main memory, or persistent storage. This is in contrast to the movement of data toward a CPU independent of where it resides, as is done traditionally. Examples of NDP already exist in systems that execute computations close to the disk, filtering or preprocessing the data streaming from the disks so that a minimal number of items are transferred for processing at other parts of the system. Conceptually, the same principle can be applied at other levels of a system's memory and storage hierarchy by placing computing resources close to where data is located, and restructuring applications to exploit the resulting distributed computing infrastructure.

At the MICRO-46 conference, a workshop was held to bring together experts from academia and industry, who presented recent advances in the development of large systems employing NDP principles (www.cs.utah.

Published by the IEEE Computer Society 0272-1732/14/$31.00 © 2014 IEEE

Authorized licensed use limited to: Copenhagen University. Downloaded on September 14,2026 at 06:54:03 UTC from IEEE Xplore. Restrictions apply.



FP 14.1: Intelligent RAM (IRAM): Chips that Remember and Compute

David Patterson, Thomas Anderson, Neal Cardwell, Richard Fromm, Kimberley Keeton, Christoforos Kozyrakis, Randi Thomas, Kathy Yelick

Computer Science, EECS, University of California, Berkeley CA

Division of the semiconductor industry into microprocessor and memory camps provides many advantages: fabrication lines can be tailored to a device, packages are tailored to the pinout and power of a device, and the number of memory chips in a computer is independent of the number of processors.

The split has disadvantages as well. While microprocessors have been improving performance by 60% per year, DRAM access time has been improving by 7% per year. This processor-memory performance gap limits many applications. For example, one microprocessor spends 75% of its time in the memory hierarchy for data base and matrix computations [1]. These delays occur despite tremendous resources being spent trying to bridge this gap. Table 1 shows that up to 60% of the area and 90% of the transistors of recent microprocessors are dedicated to the growing "memory-gap penalty"--on-chip memory latency-hiding hardware such as caches.

The DRAM industry also has difficulties. The number of DRAMs for the minimum memory of PCs is shrinking--from 32 1Mb DRAMs in 1986 to 2 64Mb DRAMs today--because the growth rate of the minimum memory size is half the growth rate of DRAM. A challenge for wide DRAMs is that some customers want parity protection and some do not. Finally, today's cache-oriented microprocessors need lower latency but instead are offered higher bandwidth with higher latency. Hence customers may no longer automatically switch to the larger capacity DRAM because the minimum memory capacity may be too large. The larger capacity DRAM will need to be in a wider configuration that is more expensive per bit than the narrow version of a smaller DRAM. The wider capacity does not match the width needed for error checking, or memory latency is higher. Thus the 256Mb or 1Gb DRAM may be greeted with indifference.

It is time to reconsider unifying logic and memory. Since most of the transistors on this merged chip will be devoted to memory, it is called "Intelligent RAM." IRAM is attractive because the gigabit DRAM chip has enough transistors for both a powerful processor and a memory big enough to contain whole programs and data sets. It contains 1024 memory blocks each 1kb wide. It needs more metal layers to accelerate the long lines of 600mm² chips [2]. It may require faster transistors for the high-speed interface of synchronous DRAM. Potential advantages of IRAM include lower memory latency (~0.1x), higher memory bandwidth (~10x), lower system power, adjustable memory width and size, and less board space. Challenges for IRAM include high chip yield given processors have not been repairable via redundancy, high memory retention rates given processors usually need higher power than DRAMs, and a fast processor given logic is slower in a DRAM process.

One microprocessor is described in sufficient detail to allow estimation of performance of an IRAM using a similar organization [1]. Given the breakdown of where time is spent, performance of each piece in an IRAM is estimated. Table 3 shows the performance factor used to scale the Alpha performance parameters to estimate the IRAM speed. Rather than a single number for each category, optimistic and pessimistic factors are used. The latency to IRAM main memory should be 5 to 10 times faster (factor of 0.1 to 0.2) than the 200-300ns latency of typical computers.

Table 2 shows optimistic and pessimistic performance for an IRAM organized like a recent microprocessor [1]. The small SPECI92 benchmarks are the poorest performers in IRAM, being 1.2 to 1.8 times slower. The database varies from a little slower to a little faster, while linpack varies from 1.2 to 1.8 times faster. These programs are more representative than SPECI92, replaced in part due to limited memory traffic.

An alternative computing style is vector processing that works on linear arrays of numbers. Vector processors do not need caches, but rely instead on low-latency memory, often made from SRAM, and high bandwidth using 100s of memory banks. Thus a gigabit IRAM memory system naturally matches the needs of a vector processor. An IRAM vector microprocessor might look like Figure 1. In a 0.18μm DRAM process with a 600mm²chip, an IRAM could have 16 add-multiply units running at 500MHz and 16, 1024b-wide memory ports at 50MHz giving a collective 100GB/s of memory bandwidth. It could run linpack at 8GFLOPS, more than five times faster than the fastest Cray vector supercomputer processor (Cray T-90). The popularity of IRAM is limited by the amount of memory on-chip. If IRAMs succeed, IRAM products should increase as memory size expands from graphics today (10 Mb) to the game and embedded markets (32Mb), and to network computers and portable PCs (128-256Mb). The semiconductor industry may soon see head-to-head competition between its currently segregated logic and memory camps.

Acknowledgements:

This research was supported by DARPA (DAIT63-C-0056), the California State MICRO program, and by research grants from Intel and Sun Microsystems.

References:

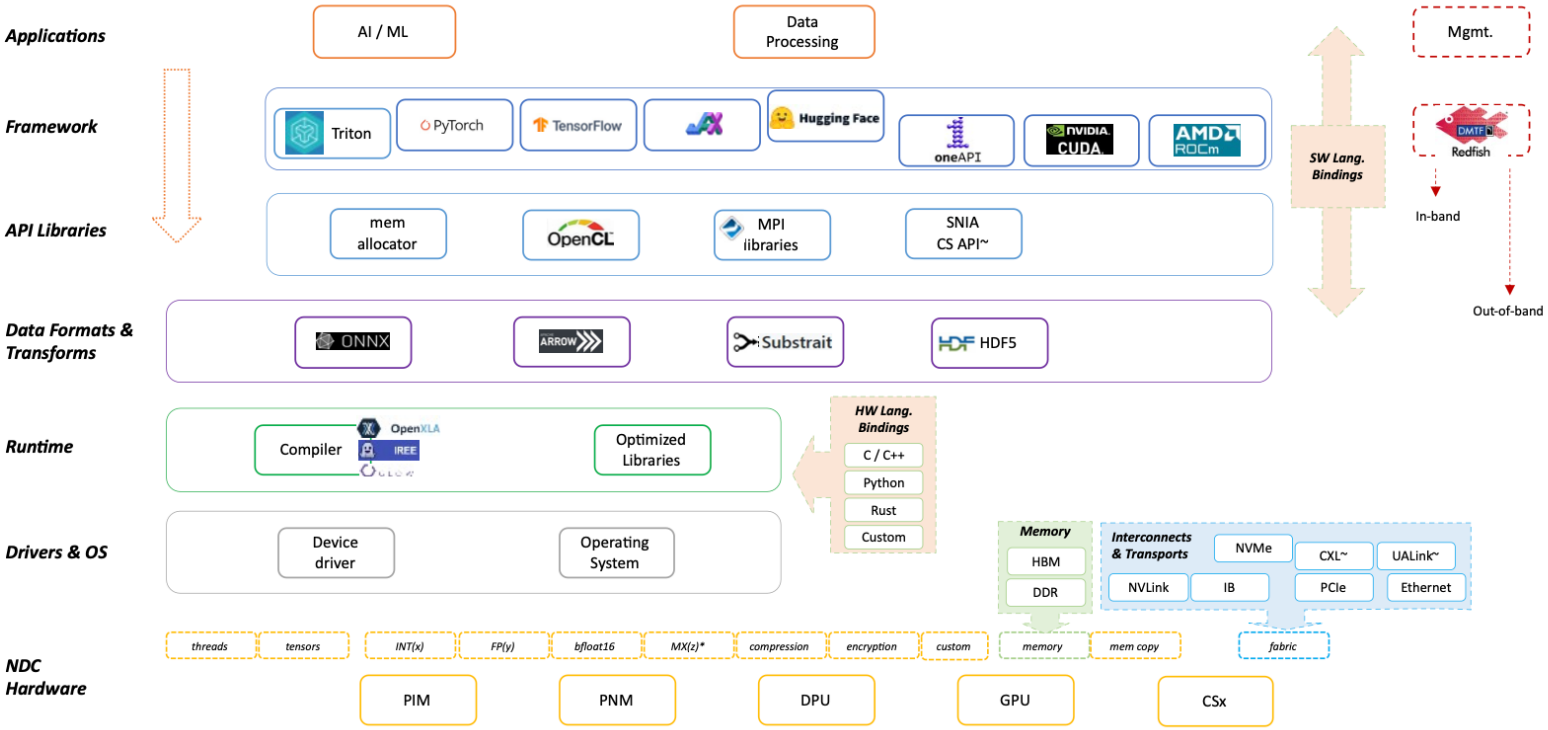
[1] Cvetanovic, Z., D. Bhandarkar, "Performance Characterization of the Alpha 21164," Proc. High Performance Computer Architecture, San Jose, CA, pp. 270-80, Feb., 1996.

[2] Yoo, H. J., et al., "A 32-Bank 1Gb DRAM with 1GB/s Bandwidth," ISSCC Digest of Technical Papers, pp. 378-379, Feb., 1996.

Authorized licensed use limited to: Copenhagen University. Downloaded on September 14,2026 at 07:04:36 UTC from IEEE Xplore. Restrictions apply.

This talk focuses on storage.
Departure from traditional memory abstraction.

OCP Near-Data Computing Framework (draft v0.1)



OCP Future Technologies Initiative/Data Centric Computing

An old idea whose time has come?

SAMSUNG

White Paper

Scaling AI Inference with KV Cache Offloading

Samsung preps PCIe 5.0 QLC SSD with a controller based on open-source RISC-V architecture — BM9K1 delivers speeds up to 11.4 GB/s for 'personal AI workloads'

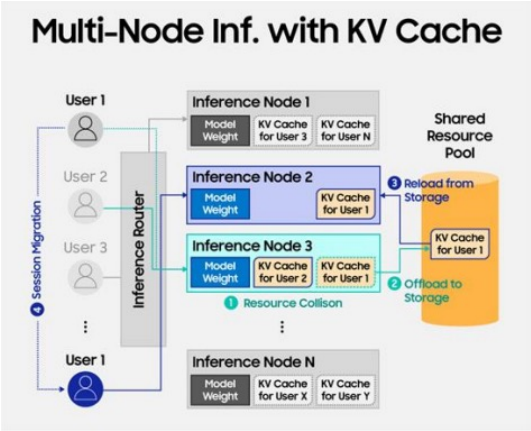
News By Luke James | Published March 28, 2026

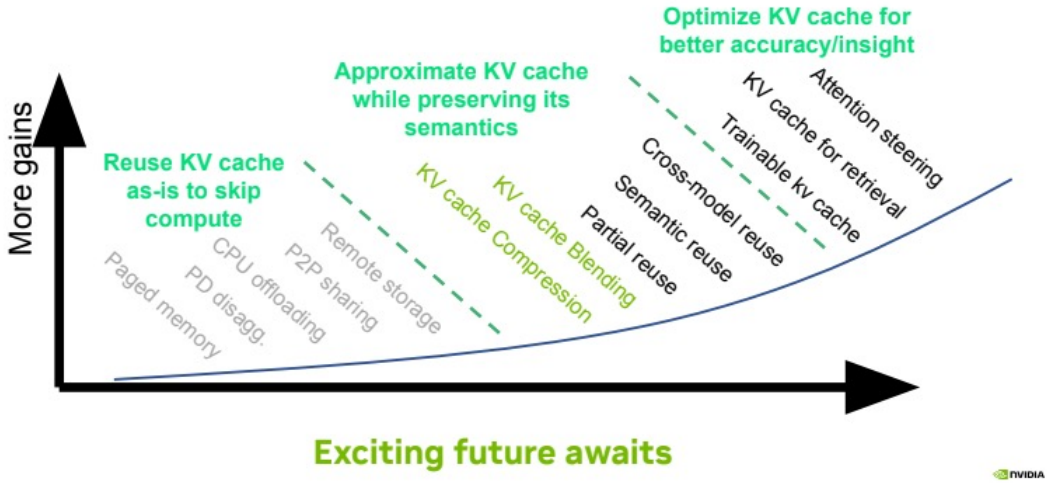
KV cache already departs from traditional caching in three fundamental ways.

First, it is persistent and multi-tier. It is stored across GPU, CPU, SSD, and cloud, survives beyond a single request, and is managed more like a data pipeline than a local buffer.

Second, it is not a black box. It can be compressed, quantized, and transformed. Its internal structure is exposed and exploitable by the systems that manage it.

Third, it is semantically mutable. It can be edited to improve model outcomes. It encodes meaning, not just computation.





https://download.semiconductor.samsung.com/resources/white-paper/scaling_ai_inference_with_kv_cache_offloading.pdf

https://blog.lmcache.ai/en/2026/04/28/stop-calling-it-kv-cache-its-something-much-bigger/

An old idea whose time has come?

ALL THINGS DISTRIBUTED NOW GO BUILD! ARTICLES @WERNER

DuckDB and the changing physics of analytics

August 26, 2026 • 3554 words

arXiv

Computer Science > Databases

[Submitted on 1 Dec 2025]

DuckDB on xNVMe

Marius Ottosen, Magnus Keinicke Parlo, Philippe Bonnet

Asynchronous I/O in DuckDB: Work, Thread, Work

Pedro Holanda

2026-07-31 | 21 min

Efficient Parquet Parsing on FPGAs

Authors: Si Jun Kwon, Zsolt Istvan, Daniel Ritter, Norman May, Christian Faerber

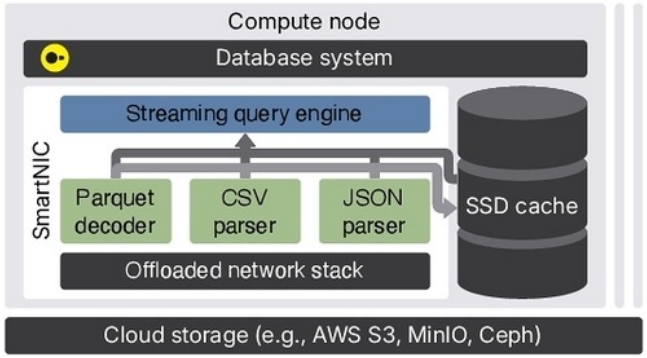
Claims

DaMoN '26: Proceedings of the 22nd International Workshop on Data Management on New Hardware
Article No.: 3, Pages 1 - 9 • https://doi.org/10.1145/3789237.3809124

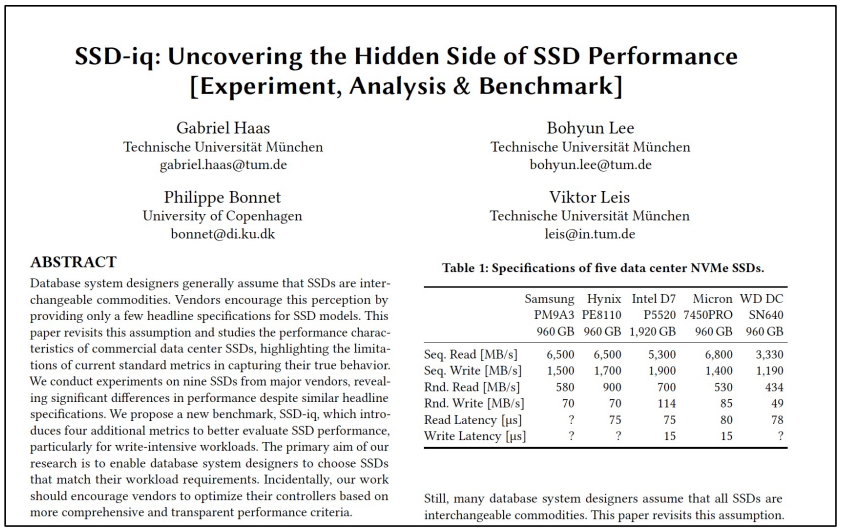
Should I Hide My Duck in the Lake?

Authors: Jonas Dann, Gustavo Alonso Authors Info & Claims

DaMoN '26: Proceedings of the 22nd International Workshop on Data Management on New Hardware
Article No.: 12, Pages 1 - 5 • https://doi.org/10.1145/3789237.3809130



An old idea whose time has come?



SSD-iq: Uncovering the Hidden Side of SSD Performance [Experiment, Analysis & Benchmark]

Gabriel Haas
Technische Universität München
gabriel.haas@tum.de

Bohyun Lee
Technische Universität München
bohyun.lee@tum.de

Philippe Bonnet
University of Copenhagen
bonnet@di.ku.dk

Viktor Leis
Technische Universität München
leis@in.tum.de

ABSTRACT

Database system designers generally assume that SSDs are interchangeable commodities. Vendors encourage this perception by providing only a few headline specifications for SSD models. This paper revisits this assumption and studies the performance characteristics of commercial data center SSDs, highlighting the limitations of current standard metrics in capturing their true behavior. We conduct experiments on nine SSDs from major vendors, revealing significant differences in performance despite similar headline specifications. We propose a new benchmark, SSD-iq, which introduces four additional metrics to better evaluate SSD performance, particularly for write-intensive workloads. The primary aim of our research is to enable database system designers to choose SSDs that match their workload requirements. Incidentally, our work should encourage vendors to optimize their controllers based on more comprehensive and transparent performance criteria.

Table 1: Specifications of five data center NVMe SSDs.

	Samsung PM9A3 960 GB	Hynix PE8110 960 GB	Intel D7 P5520 1,920 GB	Micron 7450PRO 960 GB	WD DC SN640 960 GB
Seq. Read [MB/s]	6,500	6,500	5,300	6,800	3,330
Seq. Write [MB/s]	1,500	1,700	1,900	1,400	1,190
Rnd. Read [MB/s]	580	900	700	530	434
Rnd. Write [MB/s]	70	70	114	85	49
Read Latency [µs]	?	75	75	80	78
Write Latency [µs]	?	?	15	15	?

Still, many database system designers assume that all SSDs are interchangeable commodities. This paper revisits this assumption.



FADU

Design

CONTROLLERS & CUSTOMIZED FIRMWARE

CONTROLLERS & CUSTOMIZED FIRMWARE

FADU's advanced controllers and customizable firmware are available to power your own SSD designs. Work with us to customize the firmware to make your solution unique.

Design Your Own SSD with FADU Inside

Controller

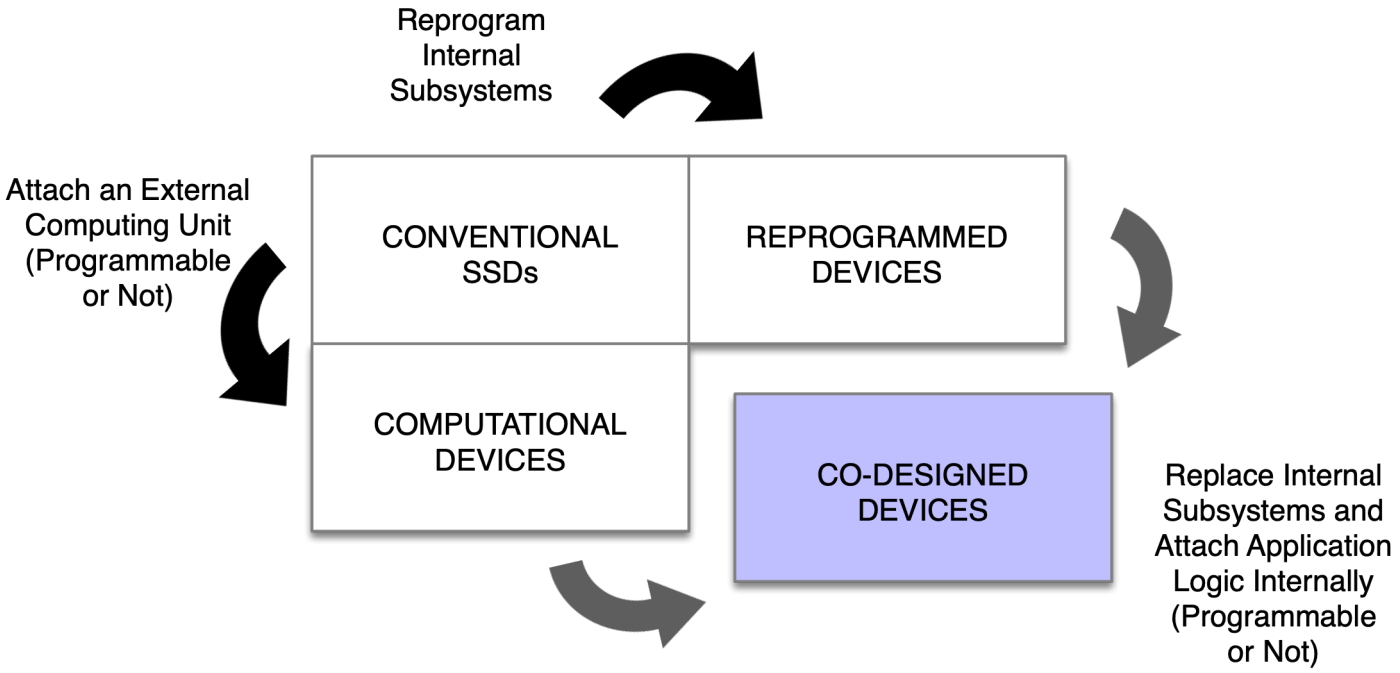
Firmware

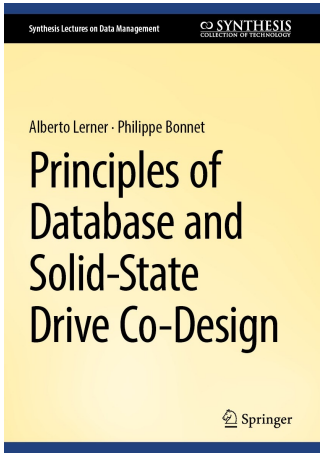
Trends

- Opportunities for cross-layer optimizations
  - End-to-end organizational control
  - Managing trade-offs rather than guessing
- Node-local processing matters
- What you read might differ from what you wrote
  - Volume – e.g., filtering
  - Representation – e.g., layout transformation
  - Access Patterns – e.g., pointer chasing

It is worth exploring data systems–SSDs co-design

Towards programmed SSDs

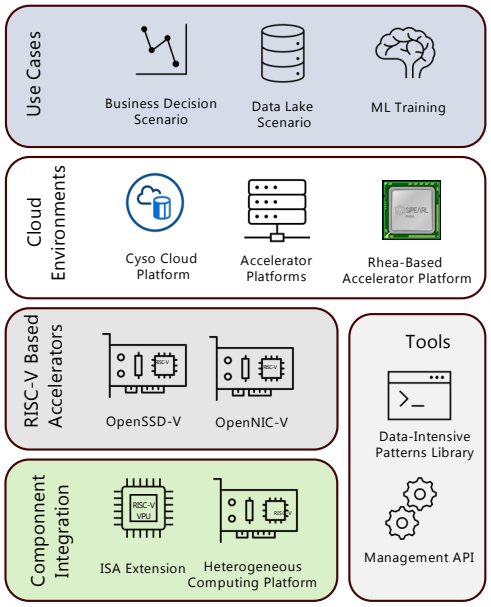




chorys.eu

CHORYS

Open and Programmable Accelerators
for
Data-Intensive Applications on the Cloud





KØBENHAVNS UNIVERSITET
Dresden University of Technology
menta
POLITECNICO MILANO 1863
TECHNISCHE UNIVERSITÄT DARMSTADT
Codasip
inesc id lisboa
CYSO

CHORYS Publications on near-data processing



Datenbank-Spektrum
https://doi.org/10.1007/s13222-026-00564-6

FACHBEITRAG

Near-Data Processing in CHORYS

Philippe Bonnet[1] · Alberto Lerner[2] · Davide Zoni[3] · Dirk Habich[3] · Erwann Kervennic[4] · Gianni Antichi[5] · Giovanni Grandi[6] · Johannes Pietrzyk[3] · Martijn Wokke[7] · Noa Omer[7] · Nuno Neves[9] · Nuno Roma[9] · Pedro Tomas[9] · Wolfgang Lehner[2] · Zsolt István[4]

Received: 1 June 2026 / Accepted: 20 September 2026
© The Author(s) 2026

Abstract
Data-intensive applications in the cloud are expected to service an ever-increasing number of users and handle exponentially growing volumes of data. Meeting these expectations is a challenge because they require balanced performance across the full memory hierarchy, from processor caches to disks accessed over the network. The problem is compounded by the cloud's shift toward disaggregated architectures, where memory and storage are pooled across fabrics, thus lengthening the data path. CHORYS addresses this challenge through near-data processing with open, programmable accelerators embedded in storage and network devices. CHORYS considers an accelerator-based cloud model in which Solid State Drives (SSDs) and Network Interface Cards (NICs) execute data-intensive operations such as filtering, aggregation, and parsing directly on the data path, thus reducing latency, energy consumption, and resource usage. The project contributes both hardware platforms (OpenSSD-V and OpenNIC-V, built around RISC-V cores) and a software stack including data-intensive pattern libraries, APIs, and runtime systems for the seamless integration of near-data execution into existing cloud infrastructures. CHORYS use cases illustrate near-data processing in the context of data lakes, object stores, and AI inference pipelines.

Keywords Near-data processing · Solid-state drives · SmartNIC · Hardware accelerator · Cloud

1 Introduction

Data-intensive applications in the cloud are expected to service an ever-increasing number of users and handle exponentially growing volumes of data. This raises several system problems linked to the traditional issue of incommensurate scaling [1]. First, compute and storage scale at different paces. Second, the multiple layers of the storage hierarchy do not scale together in terms of bandwidth or latency. As a result, data-intensive applications in the cloud are increasingly bottlenecked not by compute but by the cost of moving data. Energy, latency, and bandwidth budgets are dominated by the traffic between processors and the memory and storage tiers that feed them.

Disaggregation is a solution to the first problem. Disaggregated memory and storage are moved off the server into pooled tiers accessible over a fabric such as CXL or RDMA. This way, capacity and compute can scale independently. In addition, resources do not remain unused on servers where they are not needed. However, disaggregation makes the data movement problem worse, not better. A simple LOAD instruction that used to take around tens of nanoseconds to local DRAM now takes a few hundred nanoseconds over CXL or microseconds over a network fabric. Elasticity and utilization are gained, but at the cost of a longer data path.

Near-data processing is a way to reduce that cost. Near-data processing leverages processing elements located in or next to memory and storage—processing-in-memory, computational SSDs, smart NICs, accelerators on memory expanders—so that operations execute close to the data they touch, thus reducing data movement to a host

✉ Philippe Bonnet
bonnet@di.ku.dk

1 University of Copenhagen, Copenhagen, Denmark
2 Computing Flows, Fribourg, Switzerland
3 TU Dresden, Dresden, Germany
4 TU Darmstadt, Darmstadt, Germany
5 Politecnico di Milano, Milano, Italy
6 Menta, Valbonne, France
7 Cyso, Alkmaar, The Netherlands
8 Codasip, Munich, Germany
9 INESC-ID, Lissabon, Portugal

Published online: 26 October 2026

Springer

RESEARCH-ARTICLE | OPEN ACCESS |

Model-Driven Right-Sizing of Offloading in Data Processing Pipelines

Authors: Faeze Faghih, Maximilian Hüttner, Florin Dinu, Zsolt István Authors Info & Claims

DaMoN '25: Proceedings of the 21st International Workshop on Data Management on New Hardware
Article No.: 9, Pages 1 - 8 • https://doi.org/10.1145/3736227.3736239

Published: 10 July 2025 Publication History

Check for updates

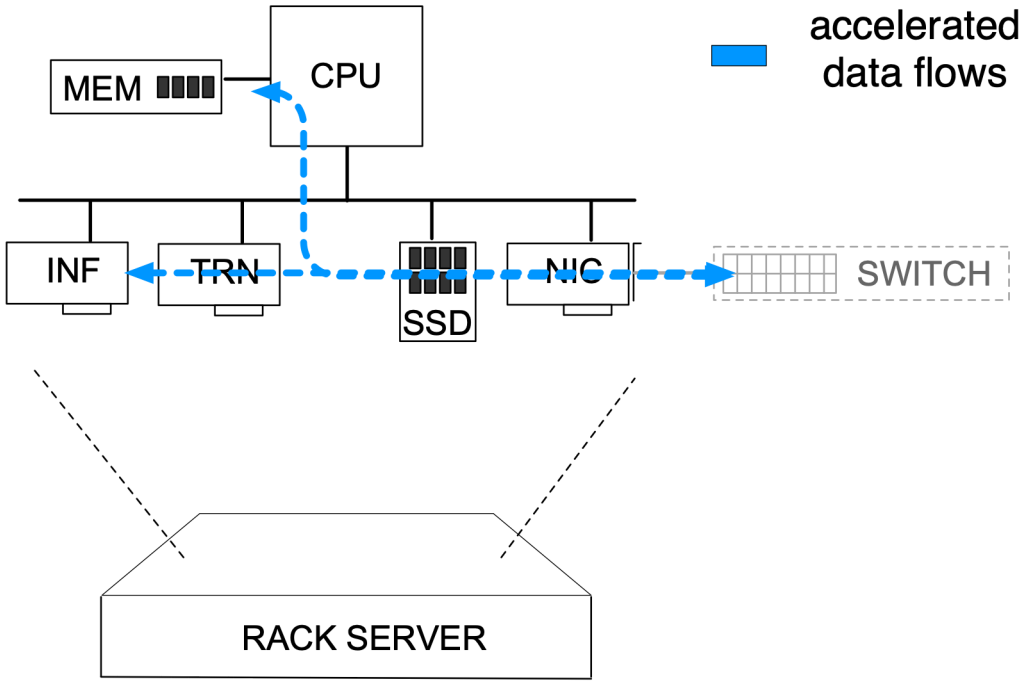
CIDR 2027

MACS: One SSD, Many Places to Compute

Alberto Lerner (Computing Flows GmbH); Philippe Bonnet (University of Copenhagen); Kibin Park (Computing Flows GmbH); Sangjin Lee (Computing Flows GmbH)

https://rdcu.be/6MBtGJLBG9Ma

Programmable Devices on the Data Path



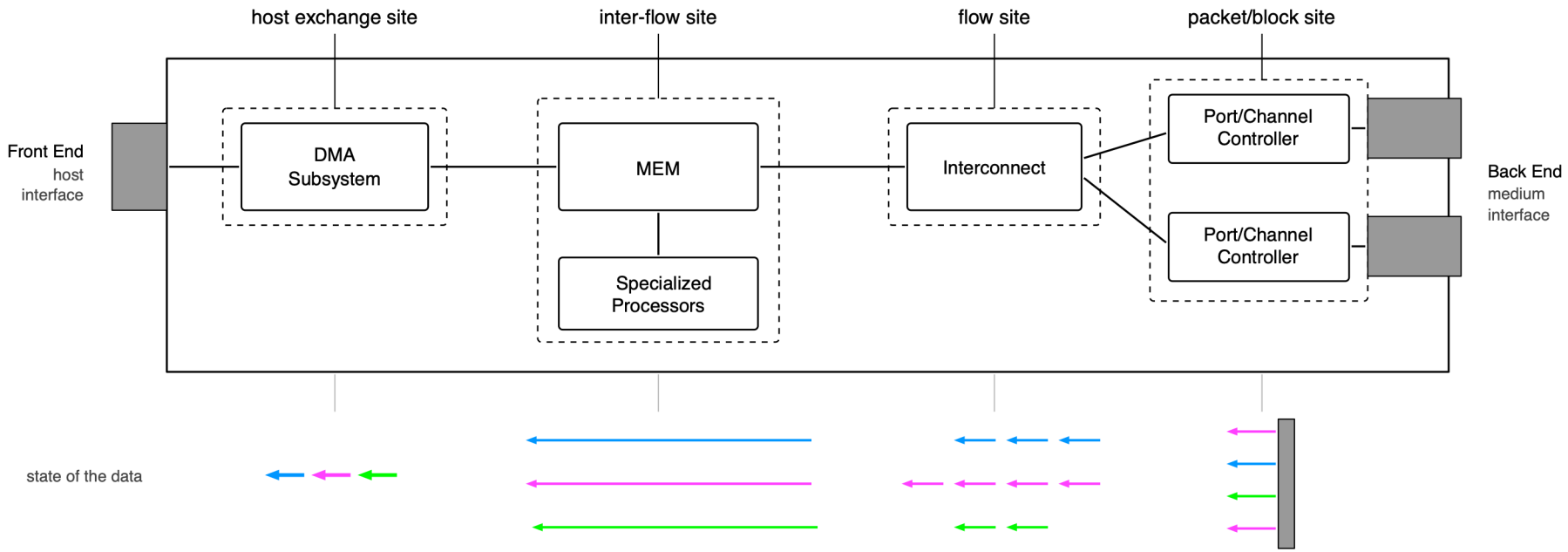
Near-Data Processing in CHORYS

Philippe Bonnet[1*], Alberto Lerner[2], Davide Zoni[5], Dirk Habich[3],
Erwann Kervennic[6], Gianni Antichi[5], Giovanni Grandi[8], Johannes Pietrzyk[3],
Martijn Wokke[7], Noa Omer[7], Nuno Neves[9], Nuno Roma[9], Pedro Tomas[9],
Wolfgang Lehner[3], Zsolt István[4]

Insight #1: Near-Data Processing leads to accelerated data flows along traditional or new paths.

NDP Architecture

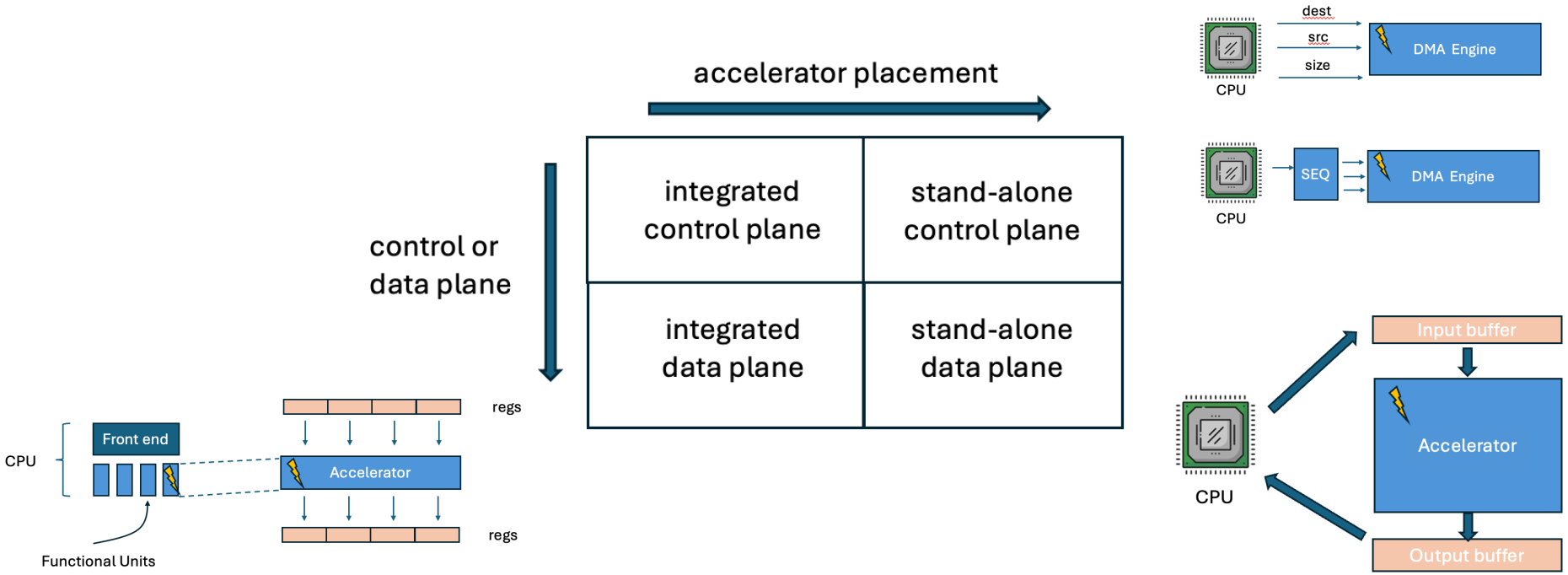
WIP led by A.Lerner



Insight #2: 4 compute sites are programmable along the data path.

Cores – Accelerators Integration

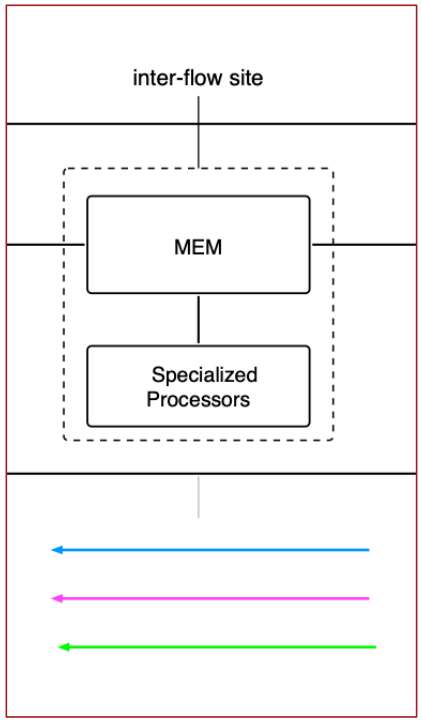
D4.1 by A.Lerner



Insight #3: Each compute site mixes multiple cores and accelerators.

Runtime

WIP at UCPH



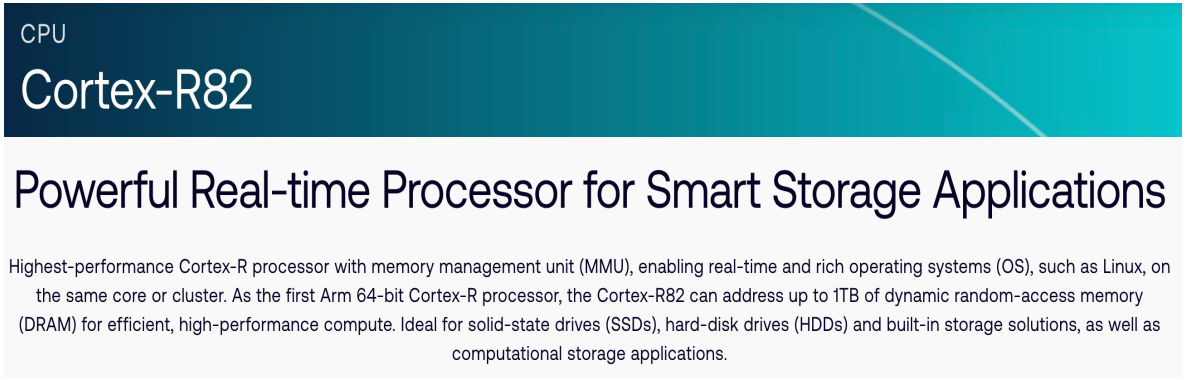
Consider a programmable SSD.

The inter-flow site is the site where:

- state is managed (e.g., L2P table)
- Multiple flows are processed.

=> Need for a runtime that efficiently supports stateful, parallel programs on multiple cores (with associated accelerators)

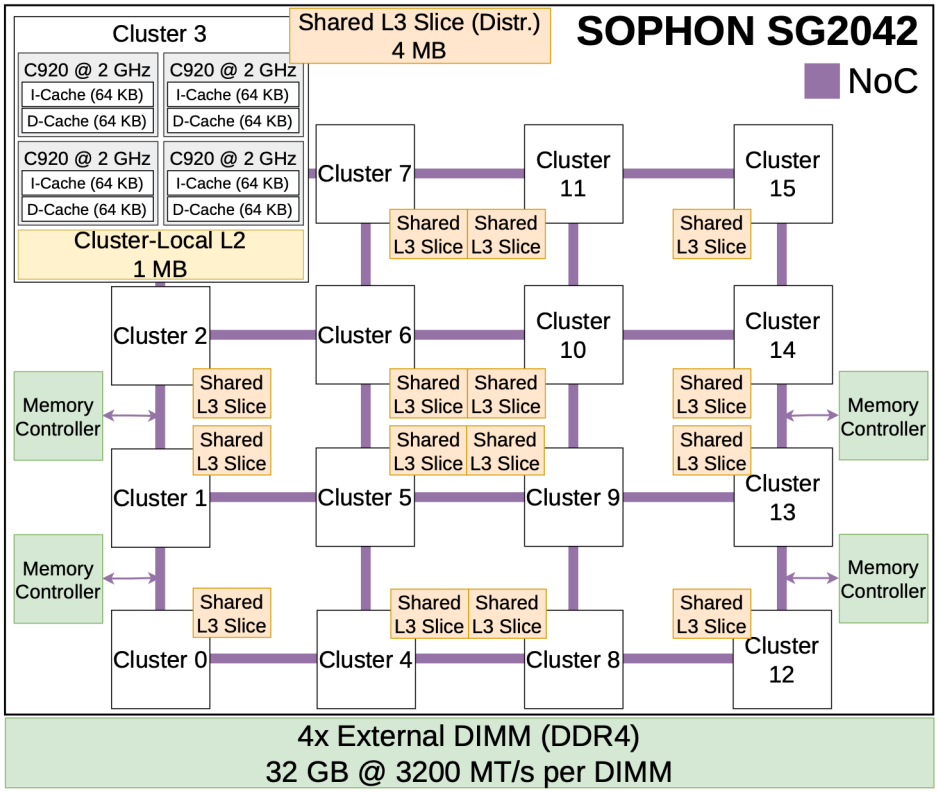
Runtime

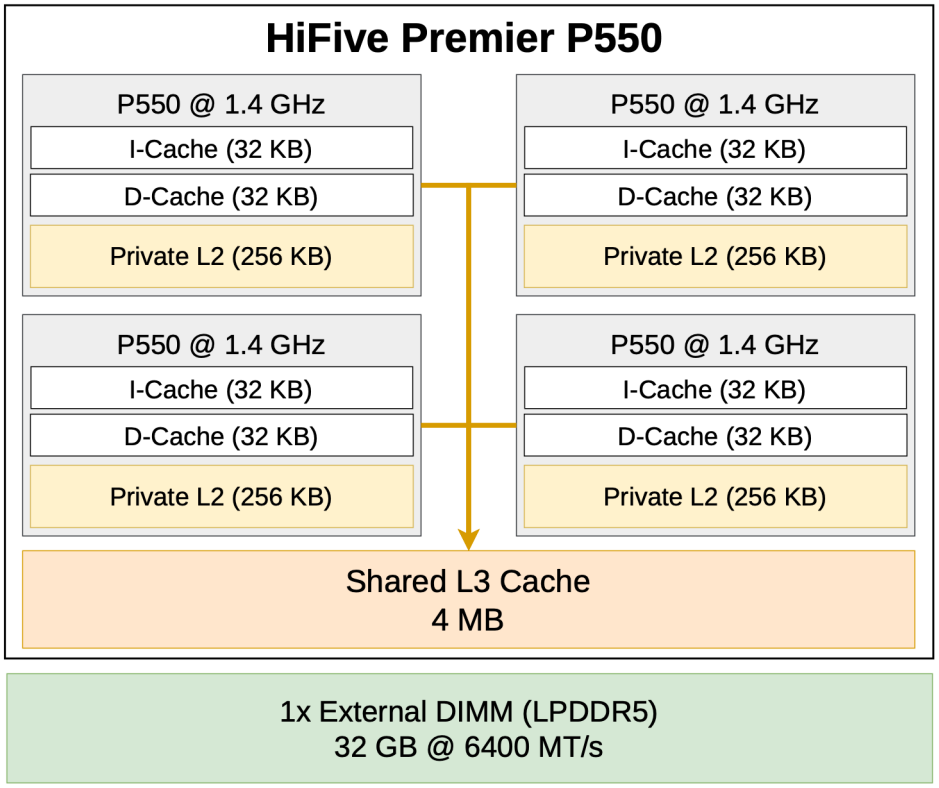


Is Linux a good candidate for this runtime?

Existing work on multicore FTLs all rely on a version of embedded linux.

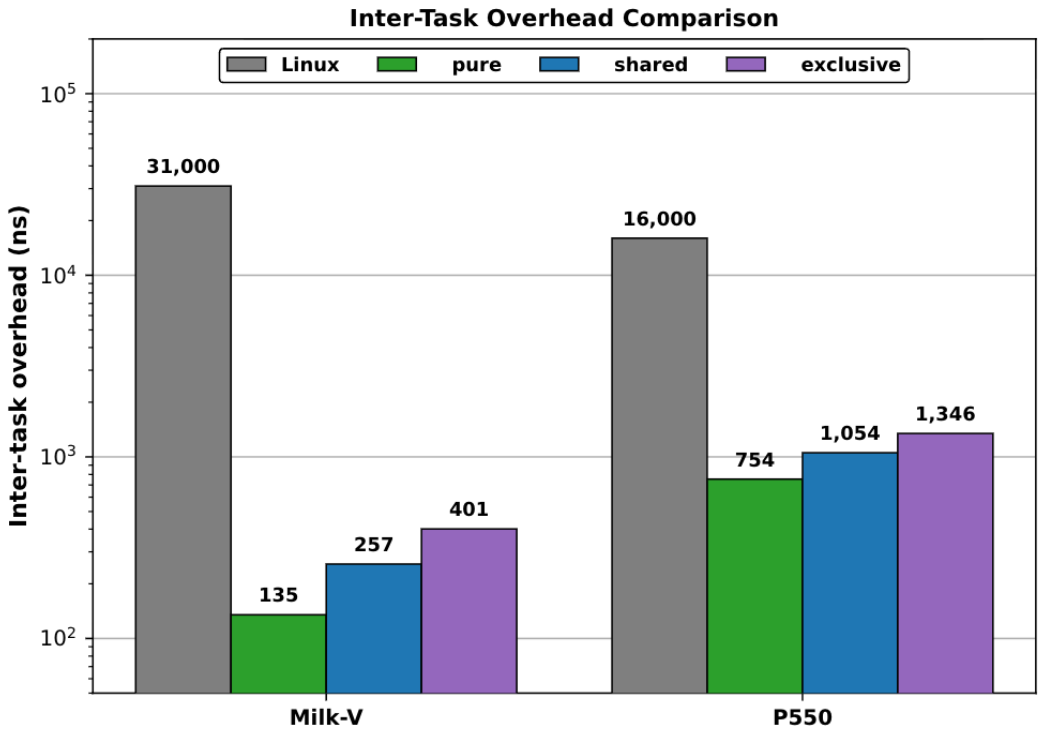
Experimental Platforms (RISC-V)





Inter-task overhead

WIP at UCPH

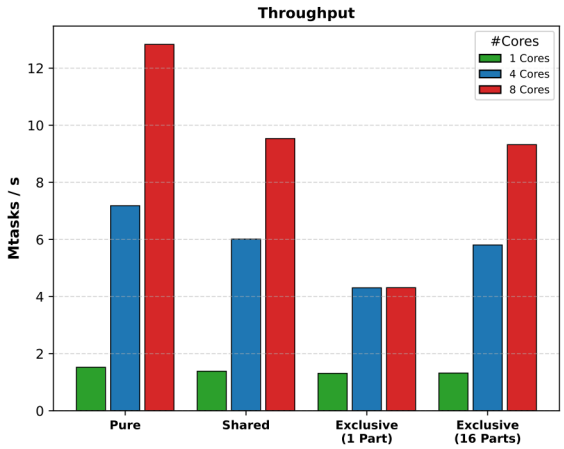


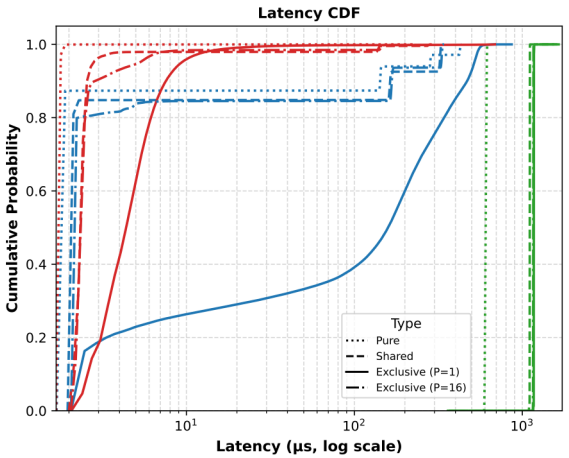
Insight #4: Linux is not a candidate to support any run-time for the inter-flow site.

CosT-RT

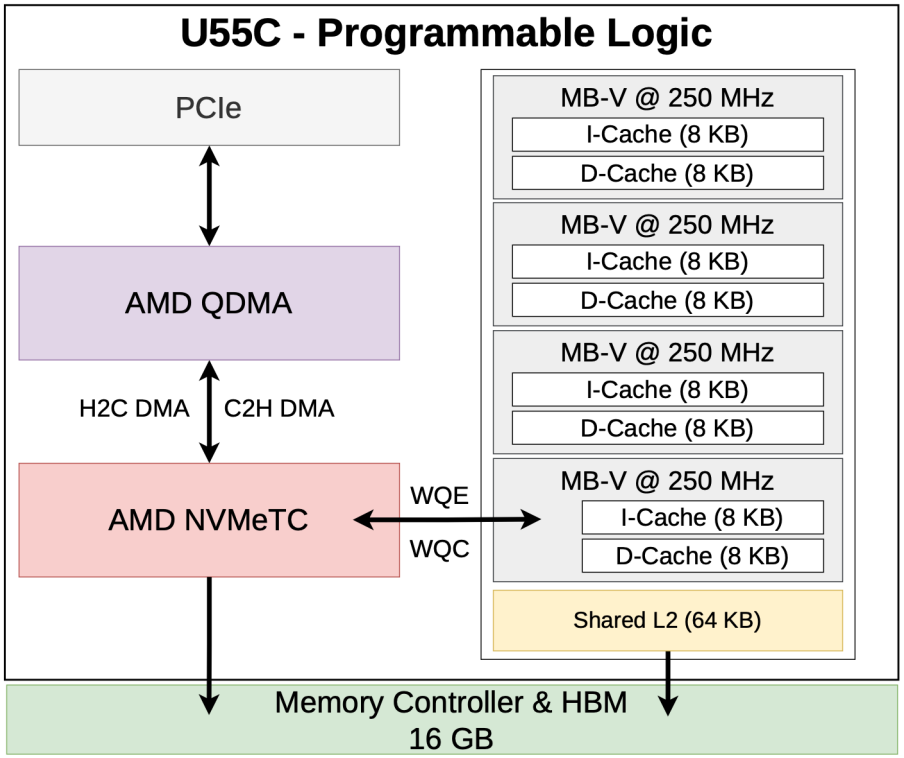
WIP at UCPH

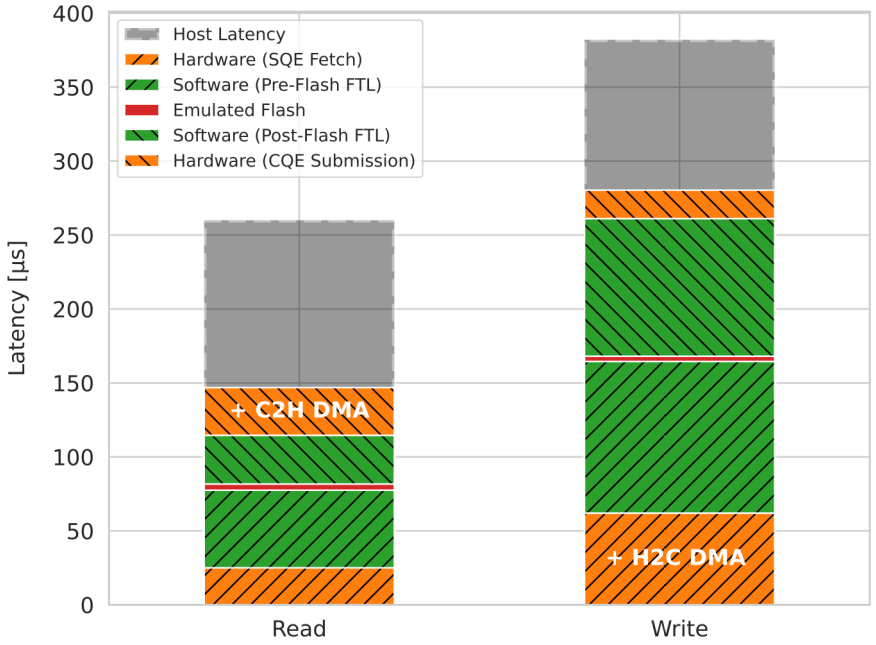
- Programs as
  - Relocatable object files
  - Graph of tasks
    - Core affinity
    - Declared access to task memory: shared, exclusive, any
- Run-time
  - A task executes to completion on a core
  - Shared and exclusive accesses to main-memory are guaranteed
  - Task delegation to avoid lock spinning on multiple cores





First programmed SSD (no flash yet)





GranPipe

Proposal: GranPipe

Dresden Database Research Group

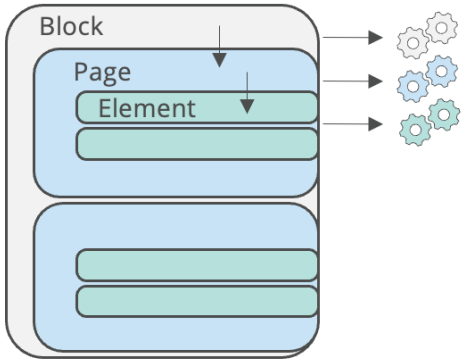
General Idea

- Structured, **composable** pipeline framework
- Separation of data access + traversal from processing
- **Hierarchical data views**: blocks, chunks, pages, elements

Design Principles

- Extensible: new granularities can be added
- No data allocation or scheduling → integration with heterogeneous/NDP systems

GranPipe Hierarchical Views



Insight #5: Composable pipelines as flow abstraction.

Conclusion

- Near-Data Processing is worth exploring
  - End-to-end cross-layer optimizations
  - Departure from traditional memory abstraction
- Early CHORYS insights:
  - New forms of data flows across architecture
  - 4 Compute sites along the data flows within an NDP device (packet/block, flow, inter-flow, and host exchange)
  - Efficient run-time for executing SSD programs
    - Memory is still the scarce resource
  - Composable pipeline as flow abstraction for high-level libraries and frameworks.

