MACH: Breaking the CPU Speed Barrier with In-Flight Data Processing

Alberto Lerner – eXascale Infolab
University of Fribourg – Switzerland

HPTS'22
Asilomar, California – USA

XI Lab

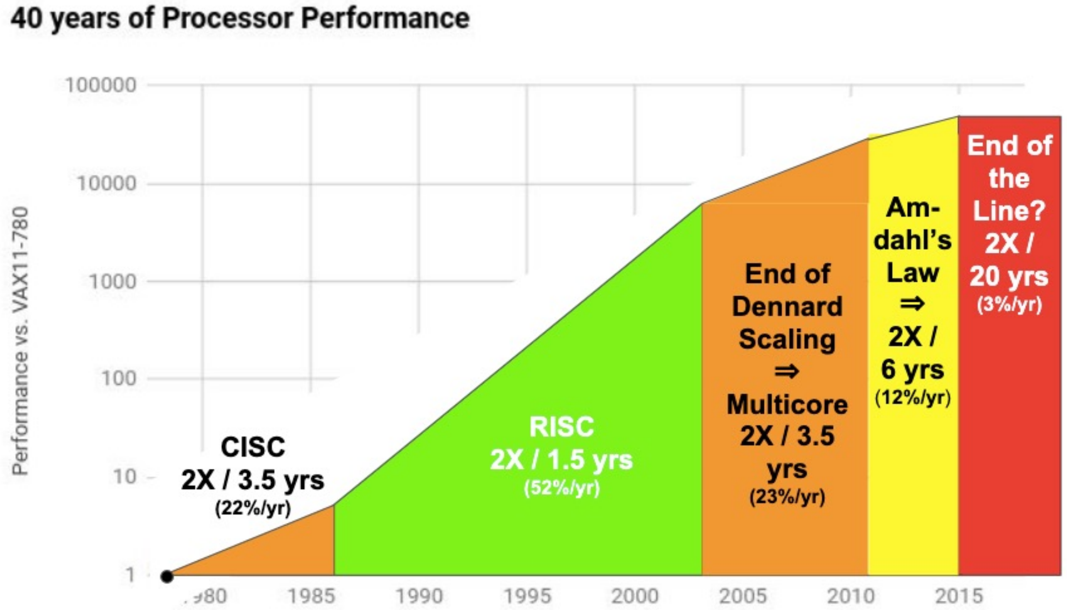
- Data Infrastructures for social / scientific / AI applications

https://exascale.info/

erc European Research Council FNS SNF HORIZON 2020 HASLER STIFTUNG SAP VERISIGN Google armasuisse amazon

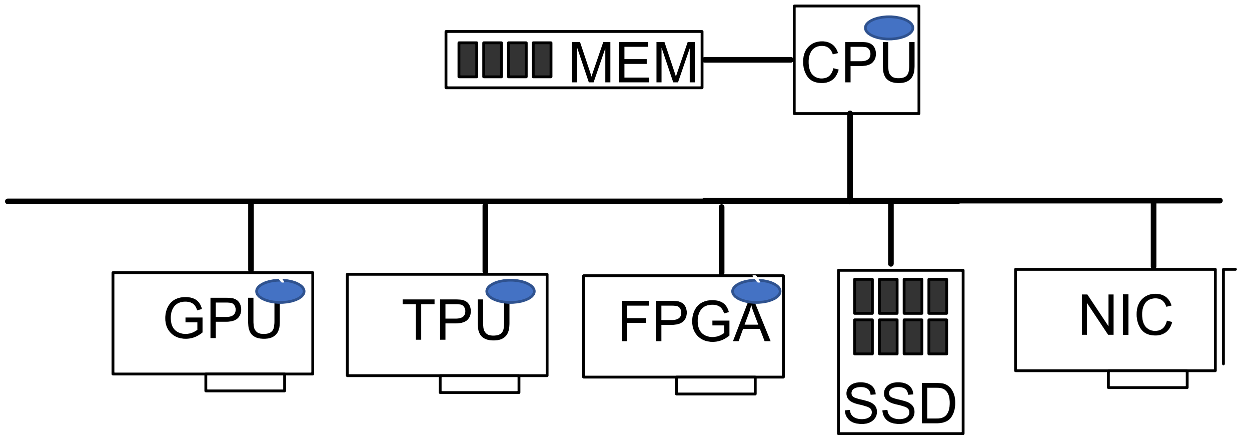
Motivation

- End of growth of single program speed (Patterson and Hennessy Turing Award lecture @ ISCA'18)
- Specialization is the answer!



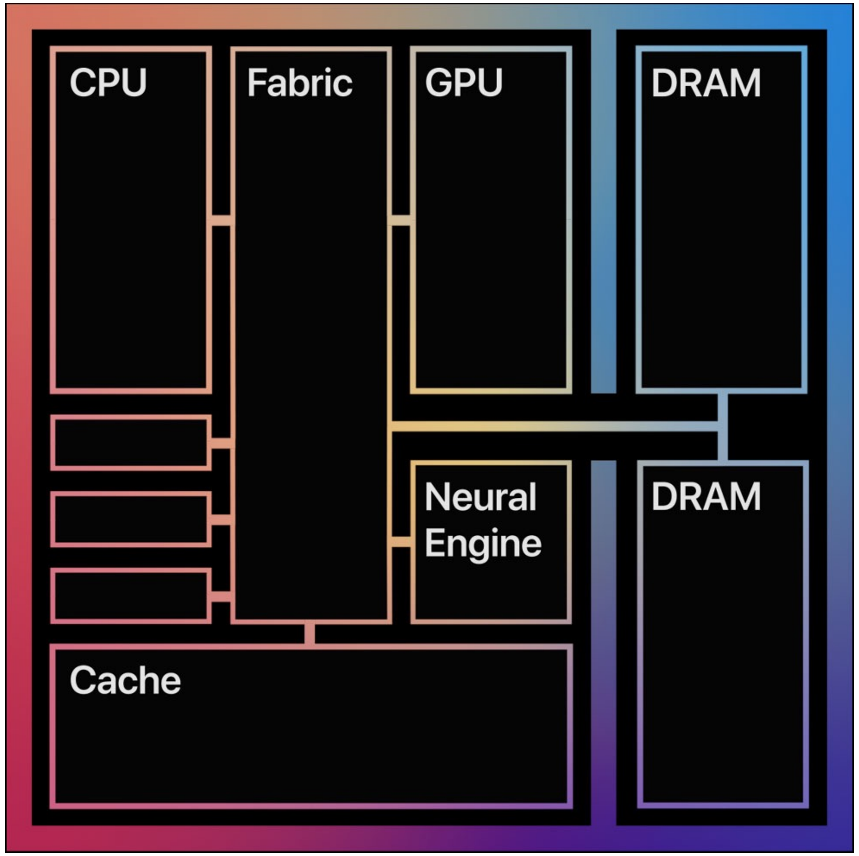
Specialization I

- Different computing units offer different functionalities



Specialization I

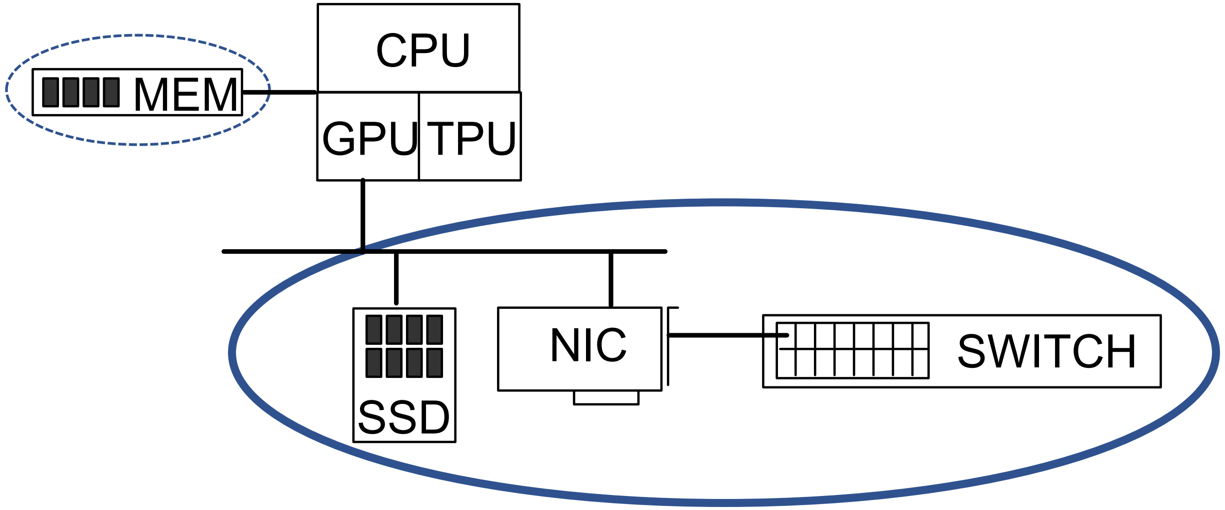
- Different computing units offer different functionalities
- A recent example: the M1 chip from Apple



Apple

Specialization II

- Different computing units offer different functionalities
- A recent example: the M1 chip from Apple
- Push functionality to units that were "passive" so far
  - Excellent work being done in Processing-In-Memory (PIM)
  - But today, we focus on I/Os



No I/O should go untapped!

Goals for Today

- Introduce (or refresh) the potential benefits of heterogenous HW
  - Emphasis on Query Execution but not only
- Introduce alternative models for using the technology in products
  - MACH
- Gauge interest in making some of the effort community based

How can we “tap” into an I/O?

- For NICs and SSDs, look into application code immediately before or immediately after a file or network descriptor for potential offloads
- For acceleration opportunities exist, partially or completely restructuring the device around an application domain (examples upcoming)

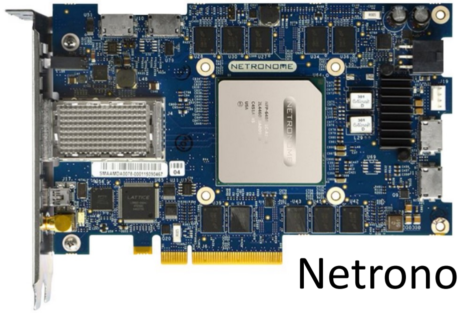
- For switches, consider why data is being transferred to a remote server: input to a computation?
  - That computation might be performed early by the switch
- The switch can also route packets looking at its contents instead of the designated destination address

Programming application logic into the I/O devices is possible!

Device Programmability w/o Domain Expertise

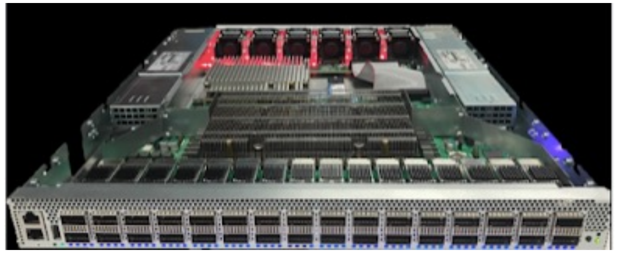
Samsung

- Historically closed but Computational Device Standard imminent



Netronome

- Choice of computing models



Intel

- Unique computing model

100% Software Programmable (but some are ”bump in the wire” model)

eBPF

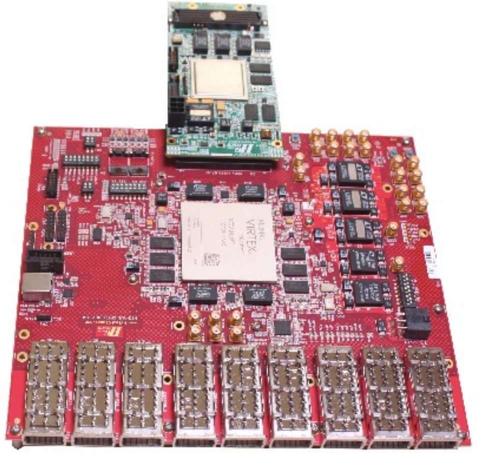
C & P4

P4

Device Programmability with Domain Expertise

CRZ - Korea

Xilinx



HTG

- 4th Generation OpenSSD
- 4 ARM cores for FW programming
- Works with vanilla NVMe driver

- Excellent 3rd party, open-source tooling (Corundum) with driver provided

- Control plane could be 100% software programming

100% Access to Control and Data paths and Firmware

No Fabrication or PCB design required. FPGA allows changing hardware via programming.

Database Offloading/Acceleration Examples

SSD

X-SSD – low latency logging & replication [SIGMOD’22]

Checkpoint Derivation

Caribou - near-data processing

GraphSSD – semantics aware storage

NIC

D-RDMA – lowering CPU usage [CIDR’22]

DB Annihilator [VLDB’22]

nanoPU – new sort record

LaKe – serving indices

SWITCH

NetAccel – 2x - … performance [CIDR’19]

Graph Mining – 32x performance

Transaction Triaging – 2x RDMA speed [VLDB’21]

Harmonia – Txn routing

P4DB – Txn Execution

Why should we care?

- Networking roadmap
  - 100Gb -> 400Gb -> 800Gb -> 1.6Tb
- PCIe roadmap
  - 3 (1GB/s lane) -> 4 (2GB/s) -> 5 (4GB/s) -> 6 (8GB/s)
- CXL
  - Potential to integrate heterogenous devices through Coherent Memory

available

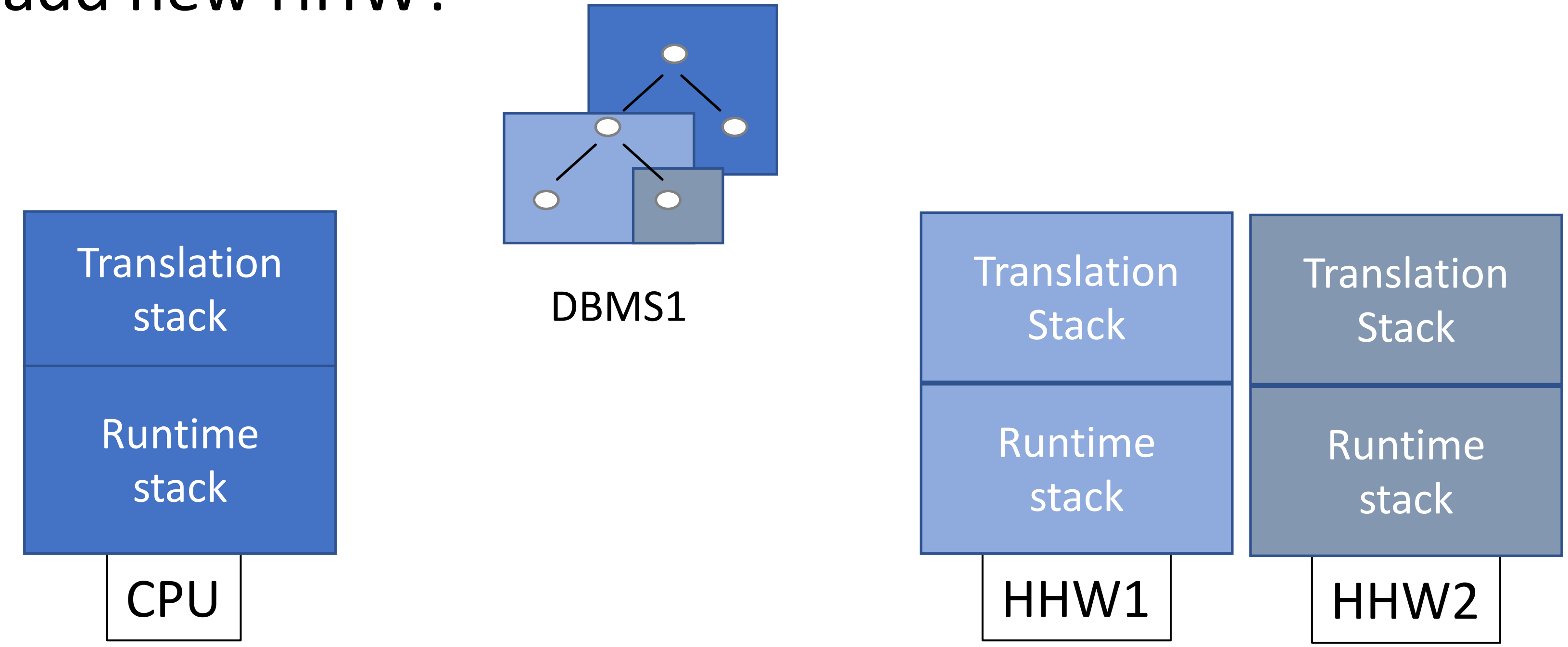
ratified

being discussed

If we peg our computations to these features, we may restore some performance growth.

Alternative 1 – Fixed-Functions Scenarios

- “Code-once” functionality
- Every database vendor fends for itself
- What happens if we add new HHW?



DBMS1

Translation stack

Runtime stack

CPU

Translation Stack

Runtime stack

HHW1

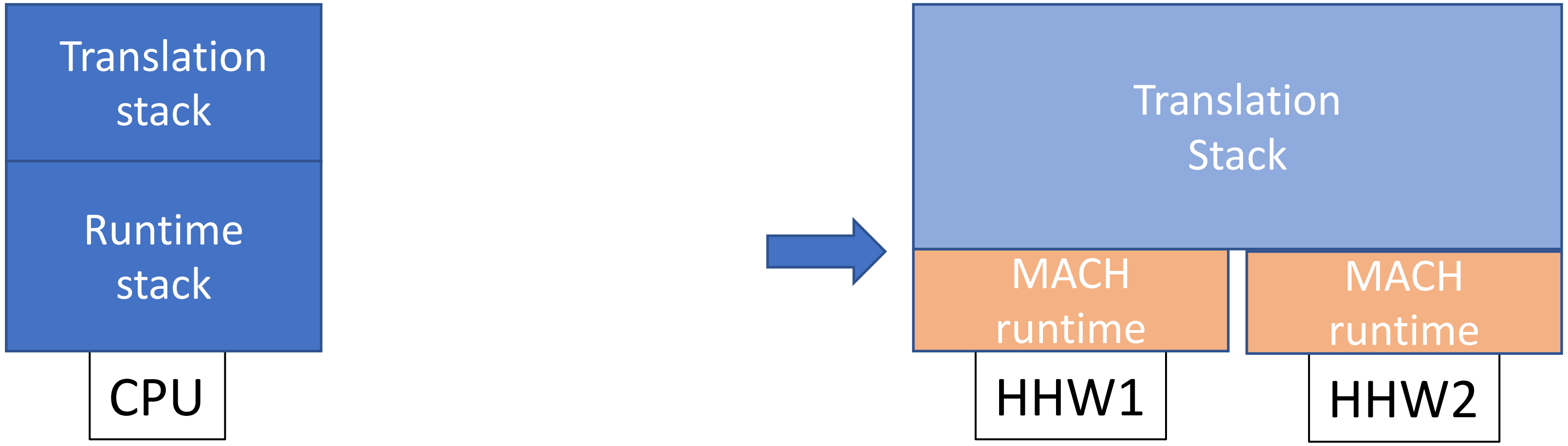
Translation Stack

Runtime stack

HHW2

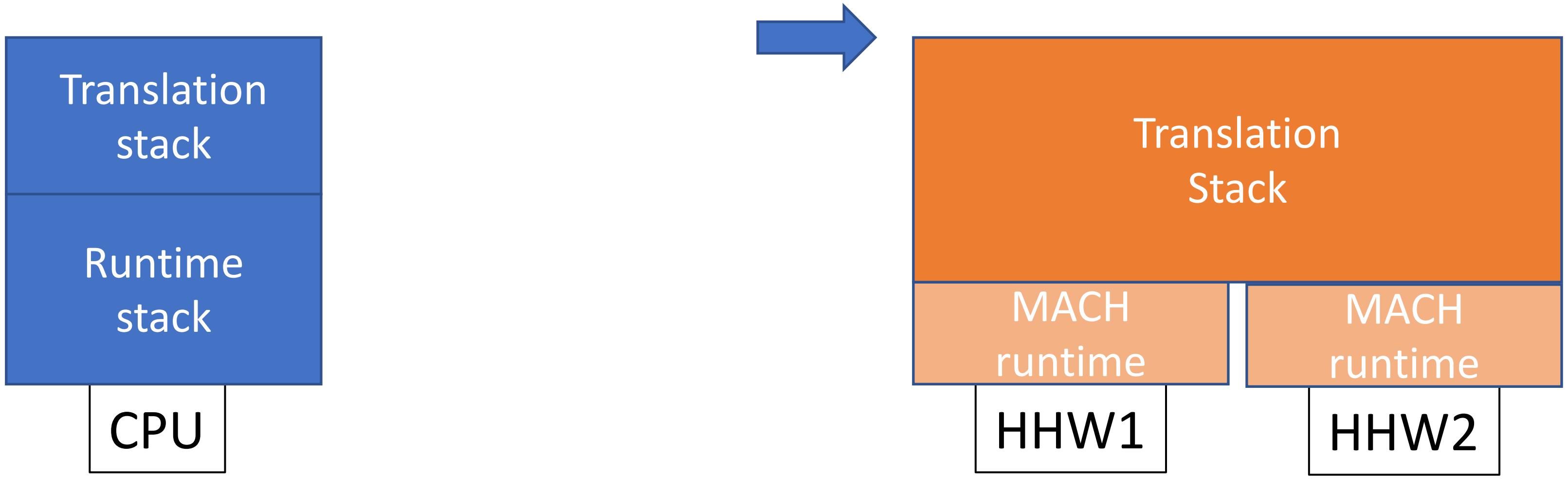
Alternative 2 – MACH Phase I

- Propose a runtime
  - Could it be common across different device types (but with “capabilities”)?
  - Data manipulation, control (FSMs?), and security
  - Low-level (not exactly SSA but at that level)
- Could we ask the hardware vendors to conform?



Alternative 3 – Mach Phase II

- Does it make sense to start the translation from a physical plan?
  - Potentially more opportunities for optimization
  - But no standard yet



Conclusion

- We are seeing a historically low-entry barrier for programmable hardware
- A database query execution and storage engines can expand beyond the PCIe and network boundary
- The cost to adopt the technology may be a function of the community interest

eXascale Infolab References

- [X-SSD] Sangjin Lee, Alberto Lerner, André Ryser, Kibin Park, Chanyoung Jeon, Jinsub Park, Yong Ho Song, and Philippe Cudré-Mauroux. "**X-SSD: A Storage System with Native Support for Database Logging and Replication**." *SIGMOD'22*.
- [D-RDMA] André Ryser, Alberto Lerner, Alex Forencich, and Philippe Cudré-Mauroux. "**D-RDMA: Bringing Zero-Copy RDMA to Database Systems**." *CIDR 2022*.
- [DBMS Annihilator] Alberto Lerner, Matthias Jasny, Theo Jepsen, Carsten Binnig, and Philippe Cudré-Mauroux. "**DBMS Annihilator: A High-Performance Database Workload Generator in Action**." In *Proceedings of the VLDB Endowment*, 15:3682–85, 2022.
- [NetAccel] Alberto Lerner, Rana Hussein, and Philippe Cudré-Mauroux. "**The Case For Network Accelerated Query Processing**." *CIDR 2019*.
- [Graph Mining] Rana Hussein, Alberto Lerner, André Ryser, Lucas Buergi, Albert Blarer, Philippe Cudré-Mauroux, "Graph Patter Mining." In Submission.
- [Transaction Triaging] Theo Jepsen, Alberto Lerner, Fernando Pedone, Robert Soulé, and Philippe Cudré-Mauroux. "**In-Network Support for Transaction Triaging.**" In *Proceedings of the VLDB Endowment*, 14:1626–39, 2021.

Additional References

- [Caribou] Zsolt Istvan, David Sidler, and Gustavo Alonso. “**Caribou: Intelligent distributed storage.**” In *Proceedings of the VLDB Endowment*, 10:1202-1212, 2017.
- [GraphSSD] Kiran Kumar Matam, Gunjae Koo, Haipeng Zha, Hung-Wei Tseng, and Murali Annavaram, “**GraphSSD: graph semantics aware SSD**.” *ISCA’19*.
- [nanoPU] Stephen Ibanez, Alex Mallery, Serhat Arslan, Theo Jepsen, Muhammad Shahbaz, Changhoon Kim, and Nick McKeown “**The nanoPU: A nanosecond network stack for datacenters**.” *OSDI’21*.
- [LaKe] Yuta Tokusashi, Hiroki Matsutani, and Noa Zilberman, “**LaKe: the power of in-network computing.**” *ReConFig’18*.
- [Harmonia] “**Harmonia: Near-linear scalability for replicated storage with in-network conflict detection**”. In *Proceedings of the VLDB Endowment*, *13:376-389*, *2019*.
- [P4DB] Matthias Jasny, Lasse Thostrup, Tobias Ziegler, and Carsten Binnig. “**P4DB-The Case for In-Network OLTP**.” *SIGMOD’22*.