A Short Study of Recent Smart Storage Solutions for OLAP: Lessons and Opportunities

Faeze Faghih, TU Darmstadt – Systems Group

Zsolt Istvan, TU Darmstadt – Systems Group

Florin Dinu, Huawei Munich Research Center

ADMS Workshop - September 5th 2022

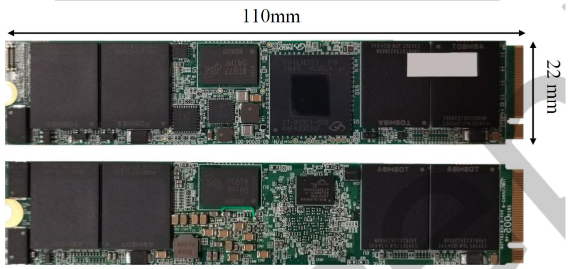
Outline

- What problem does Smart Storage trying to solve?
- Principles for designing next Smart Storage solutions
- Various Aspects of the proposed solutions
  - Location of the processing elements
  - Deployment options
  - Evaluated Applications
- Details about some proposed solutions

Bottleneck of Data Movement

- Increasing usage of data-intensive applications → data movement bottleneck

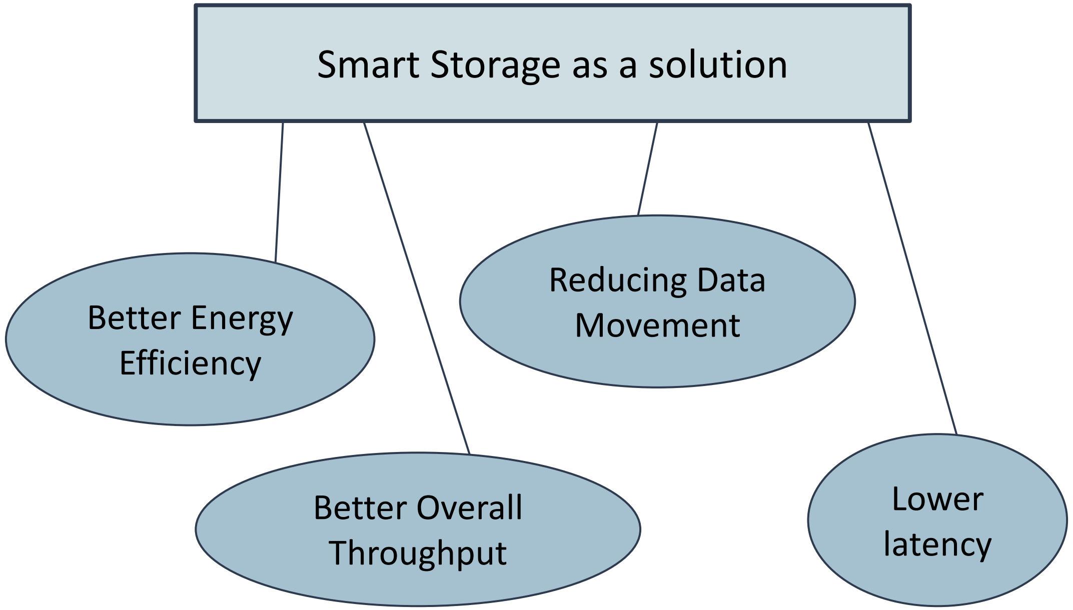
NGD systems Newport



Samsung SmartSSD

ScaleFlux

ScaleFlux CSD 3000 SERIES
COMPUTATIONAL STORAGE DRIVE
ScaleFlux
NVMe STORAGE
ScaleFlux
COMPUTATIONAL STORAGE DRIVE



Smart Storage as a solution

Better Energy Efficiency

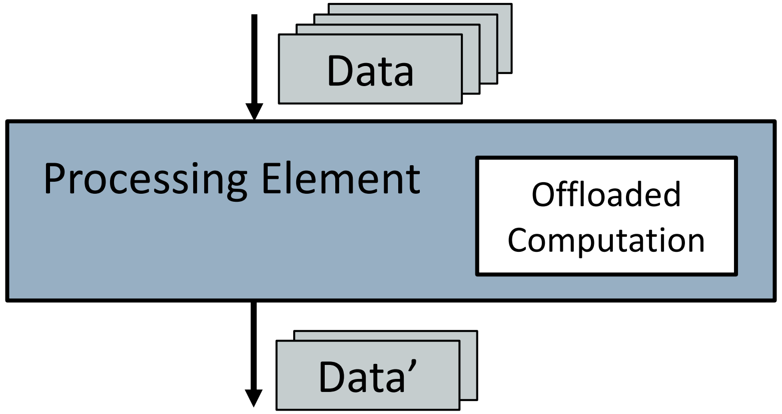
Reducing Data Movement

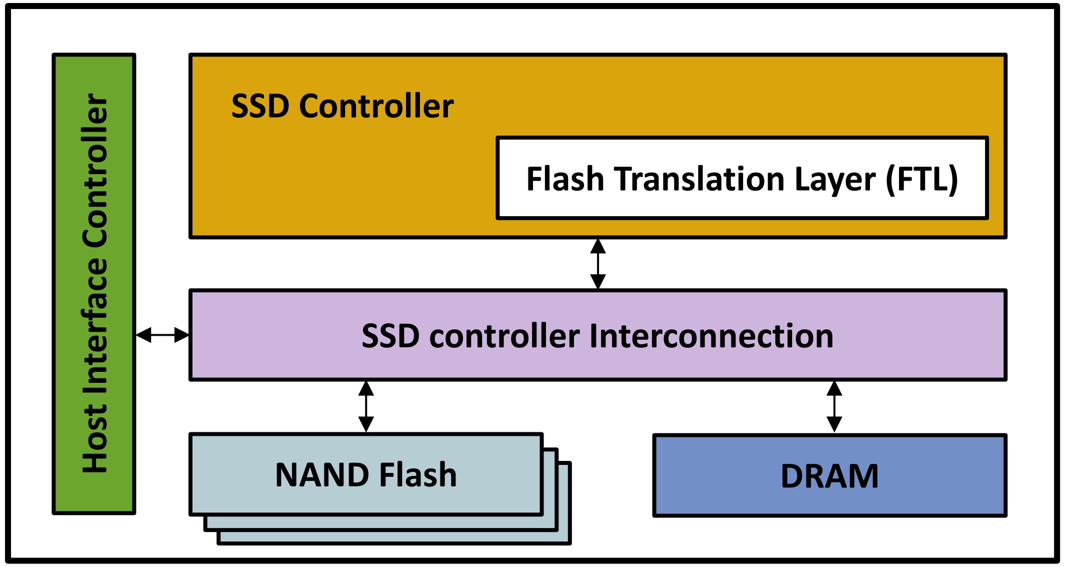
Better Overall Throughput

Lower latency

Processing Near or Inside a Storage

- Inserting processing elements to run the offloaded computation




Host Interface Controller
SSD Controller
Flash Translation Layer (FTL)
SSD controller Interconnection
NAND Flash
DRAM

Guiding Principles for Designing Future Smart Storage

Balance the compute cost of offloading against the data selectivity

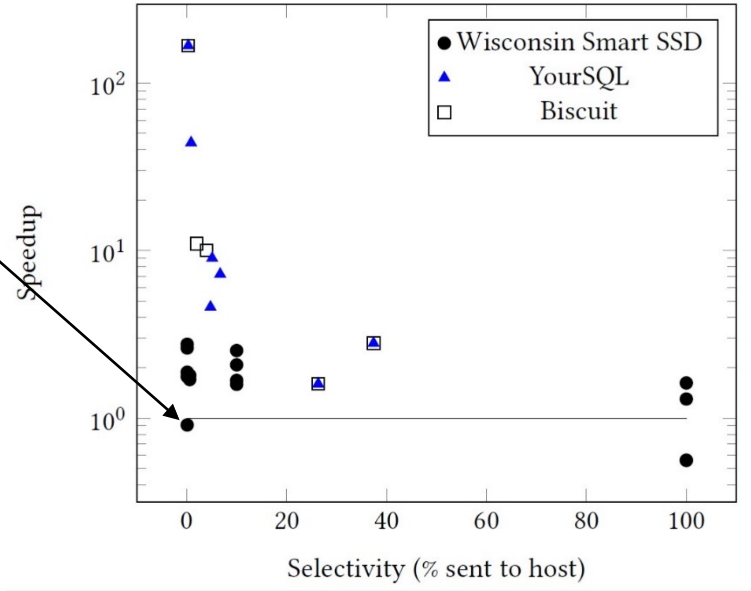
Match storage data layout with expected layout of the application

Embrace heterogeneity

Balance the Compute Cost of Offloading Against the Data Selectivity

- Must consider compute capacity of the processing element

0.1% selectivity -> (relative speedup of 0.91x)



High selectivity in itself is not a guarantee for better performance

Match the Data Layout with the Expected Layout of the Application

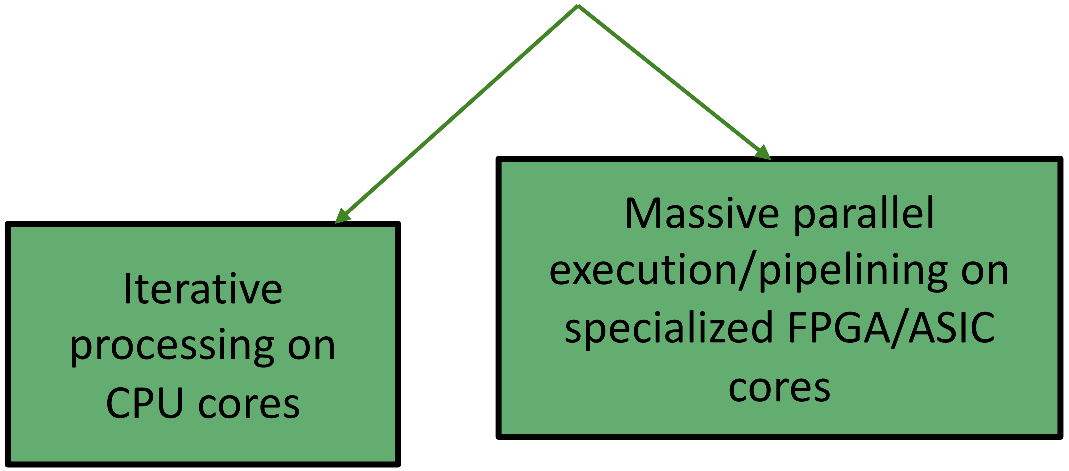
- Performance benefits could be significantly reduced due to
  - Additional data transformations, or
  - Loading data more than necessary

Query (running in Wisconsin Smart SSD)	**NSM (row-based)**	**PAX (column-based)**
Select, selectivity= 0.1%	1.77	2.63
Select-aggregation, selectivity= 0.1%	1.88	2.76

Match the chosen in-storage data layout with that of the processing element and application

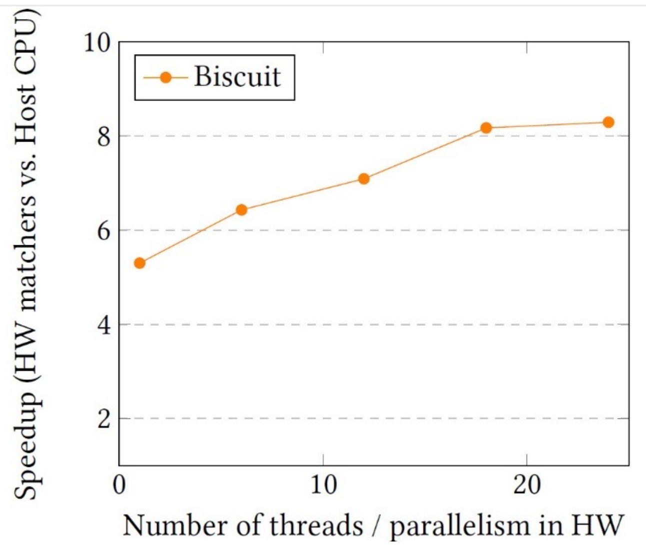
Embrace Heterogeneity

- A heterogeneous design allows for “best of both worlds”



Iterative processing on CPU cores

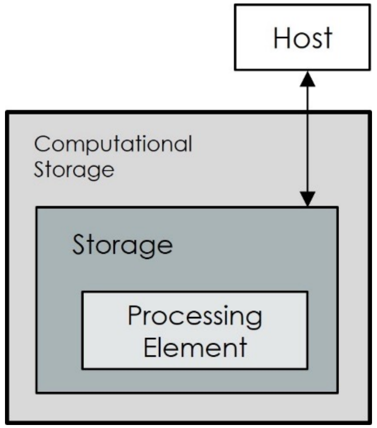
Massive parallel execution/pipelining on specialized FPGA/ASIC cores



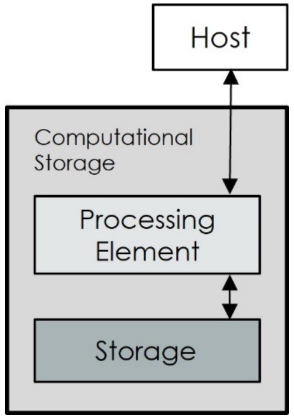
Leveraging heterogeneous processing elements benefits efficiency

Location of The Processing Element

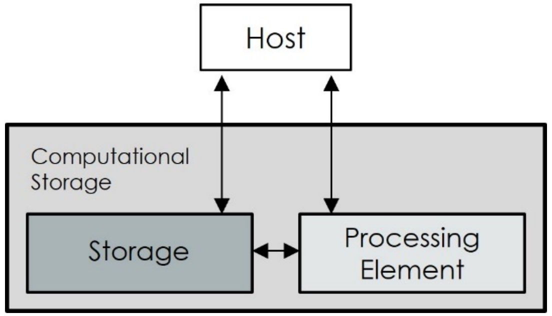
- Three possible locations of processing elements:



Processing happening in CPU shared with SSD controller (co-processor)



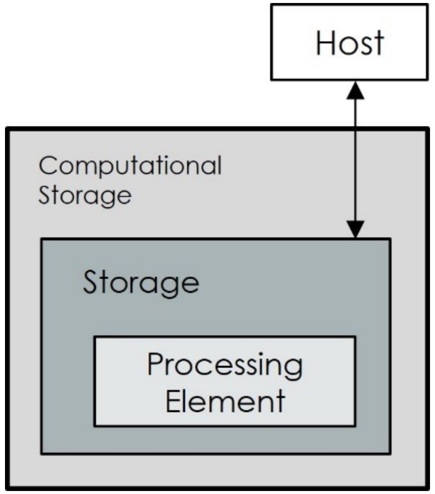
Dedicated processing element in the data path

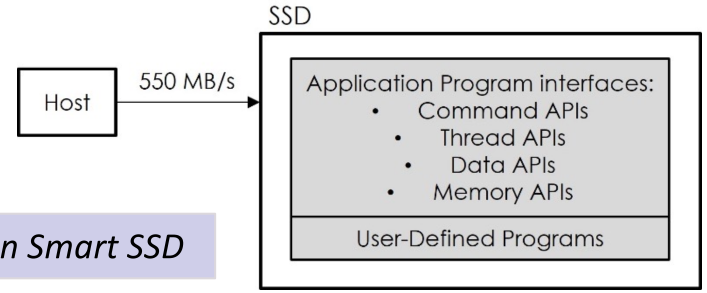


Dedicated processing element sits on the side of the storage

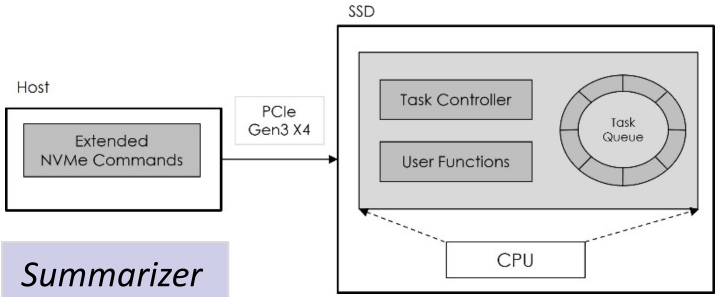
Location of The Processing Element

- Three possible locations of processing elements: 1, **Shared with SSD Controller**





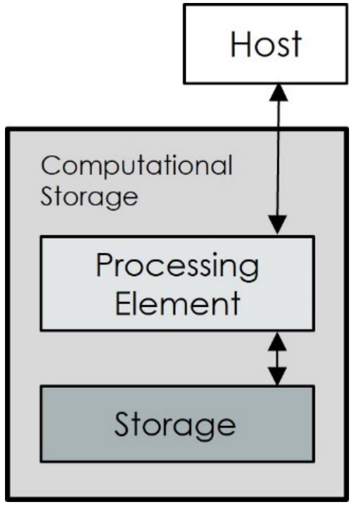
Wisconsin Smart SSD



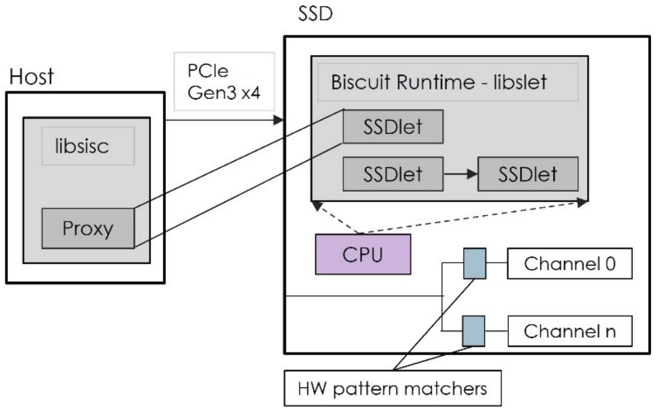
Summarizer

Location of The Processing Element

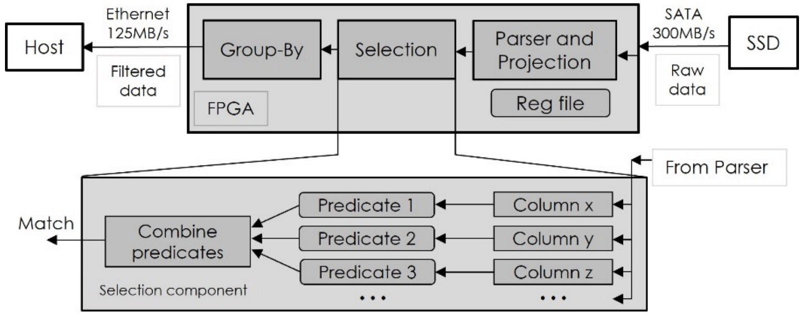
- Three possible locations of processing elements: 2, **Dedicated Processing Element**



Biscuit

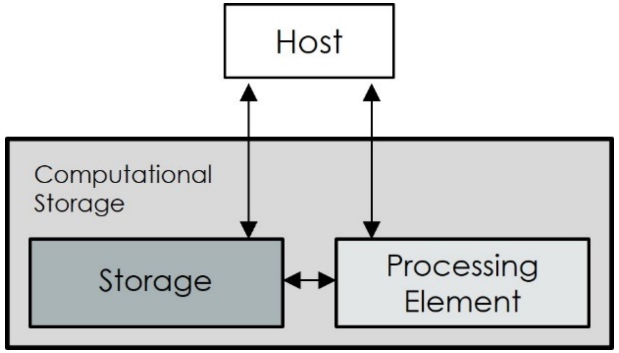


Ibex

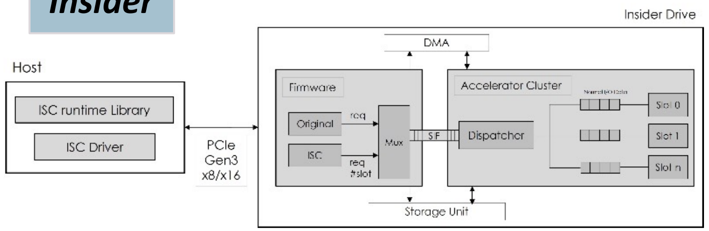


Location of The Processing Element

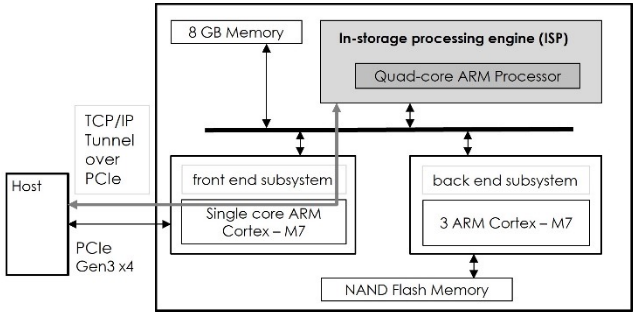
- Three possible locations of processing elements: 3, **Sits on the Side of the Storage**



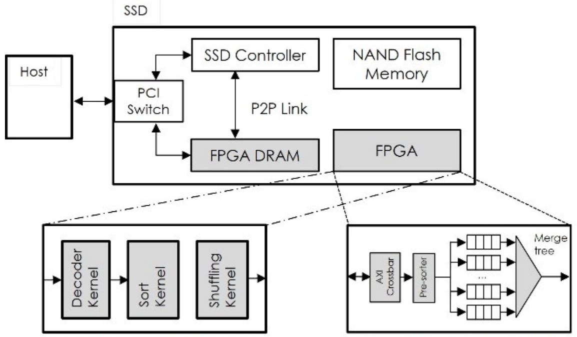
Insider



NGD system newport



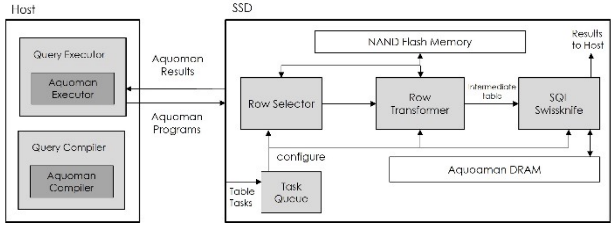
FANS and NASCENT



NASCENT Architecture includes a shuffler kernel to shuffle other columns of data tables after sorting the key column

FANS Architecture implements the merge tree for sorting phase and merging phase with different tree configurations.

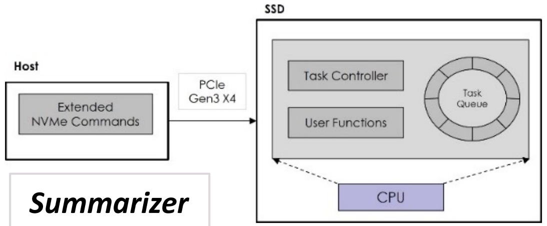
AQUOMAN



Variety of Deployment Options for Processing Element in the Storage Device

Various choice of processing elements for running offloaded computation

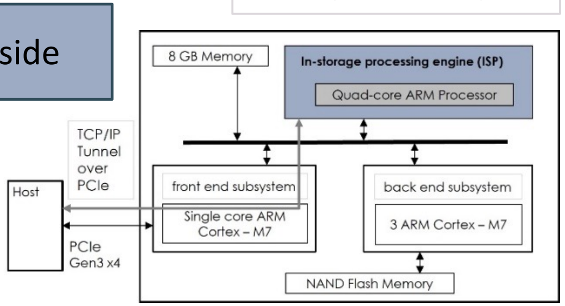
embedded CPU of the disk



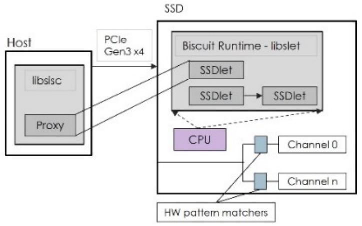
Summarizer

dedicated specific CPU inside

NGD system Newport

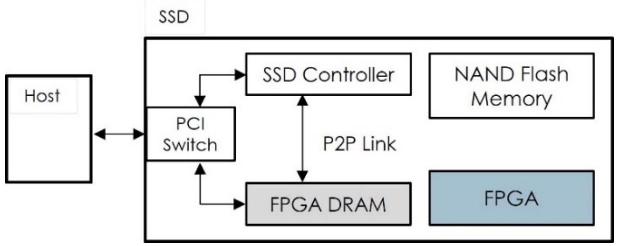


combination of an CPU and a FPGA/ASIC



Biscuit

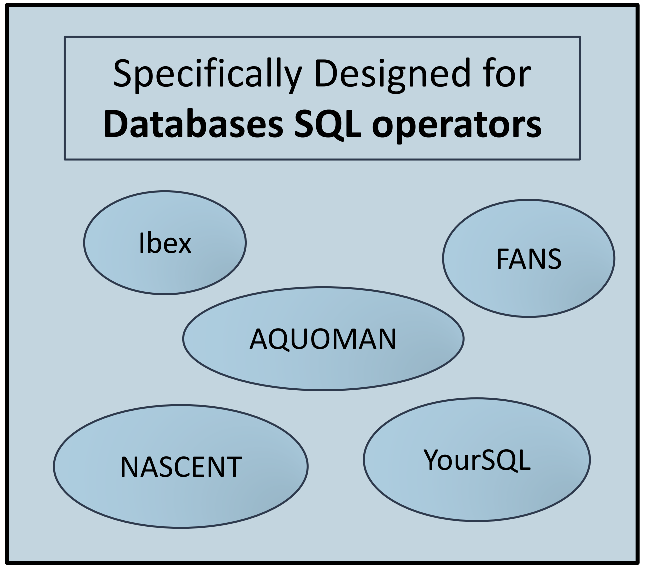
Field Programmable Gate Arrays (**FPGAs**) specifically designed for this purpose



Samsung SmartSSD

Design Goals and Evaluated Applications

- Most of the related works use SQL operators for evaluation with TPCH benchmark



Specifically Designed for **Databases SQL operators**

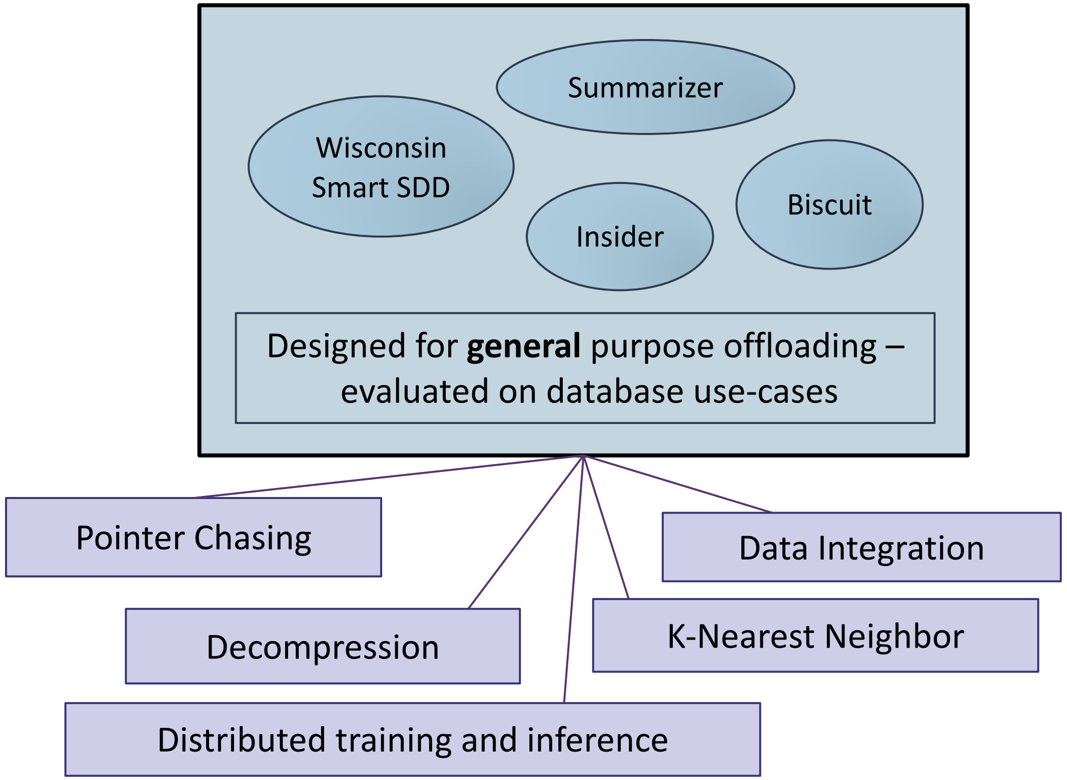
Ibex

FANS

AQUOMAN

NASCENT

YourSQL



Summarizer

Wisconsin Smart SDD

Insider

Biscuit

Designed for **general** purpose offloading – evaluated on database use-cases

Pointer Chasing

Data Integration

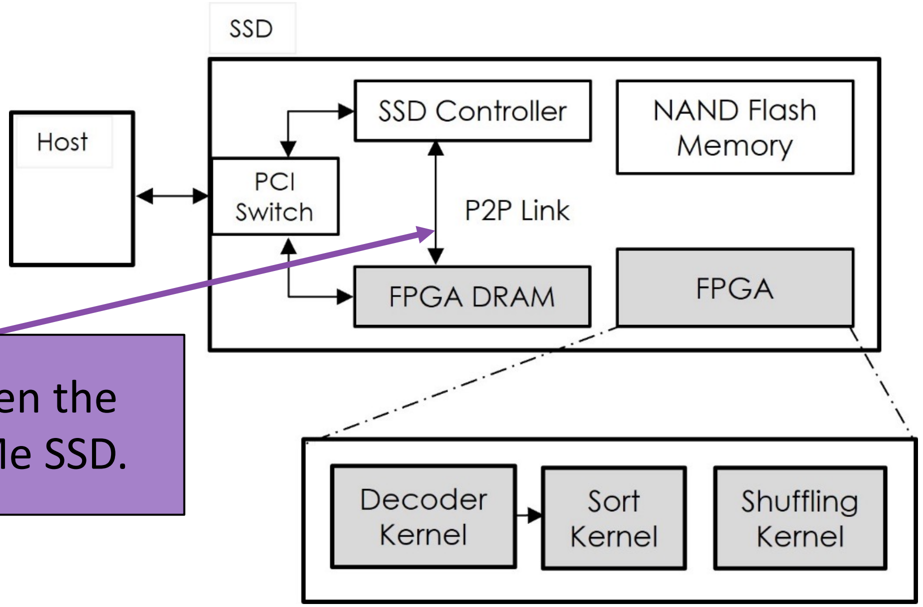
Decompression

K-Nearest Neighbor

Distributed training and inference

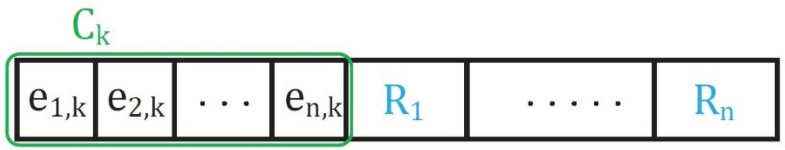
NASCENT (2021), Samsung SmartSSD as Platform

- Implements **Bitonic Sort**
- The host can:
  - Initiate FPGA DRAM read/write requests,
  - Send computation requests to the FPGA



(P2P) connections between the FPGA DRAM and the NVMe SSD.

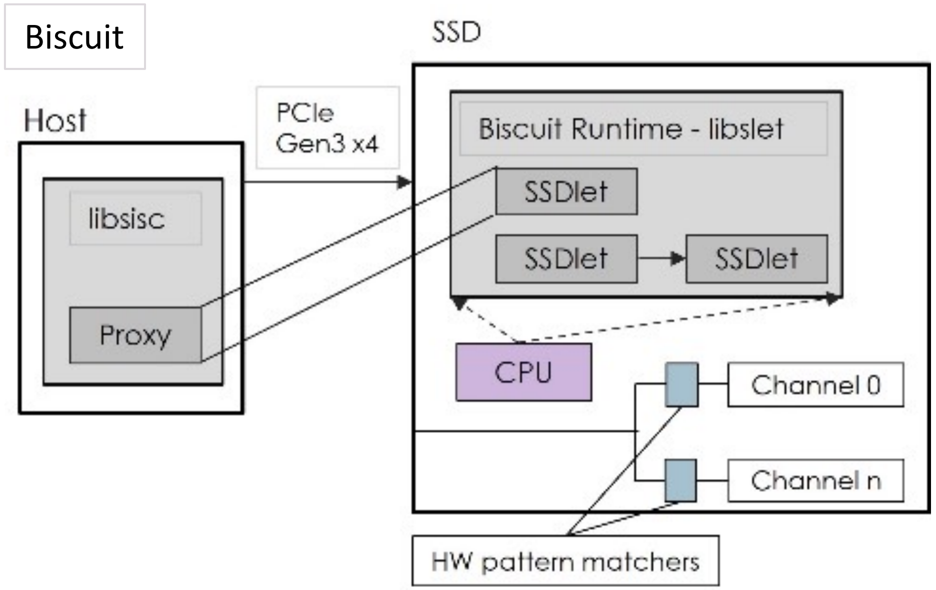
Arrangement of table in storage:

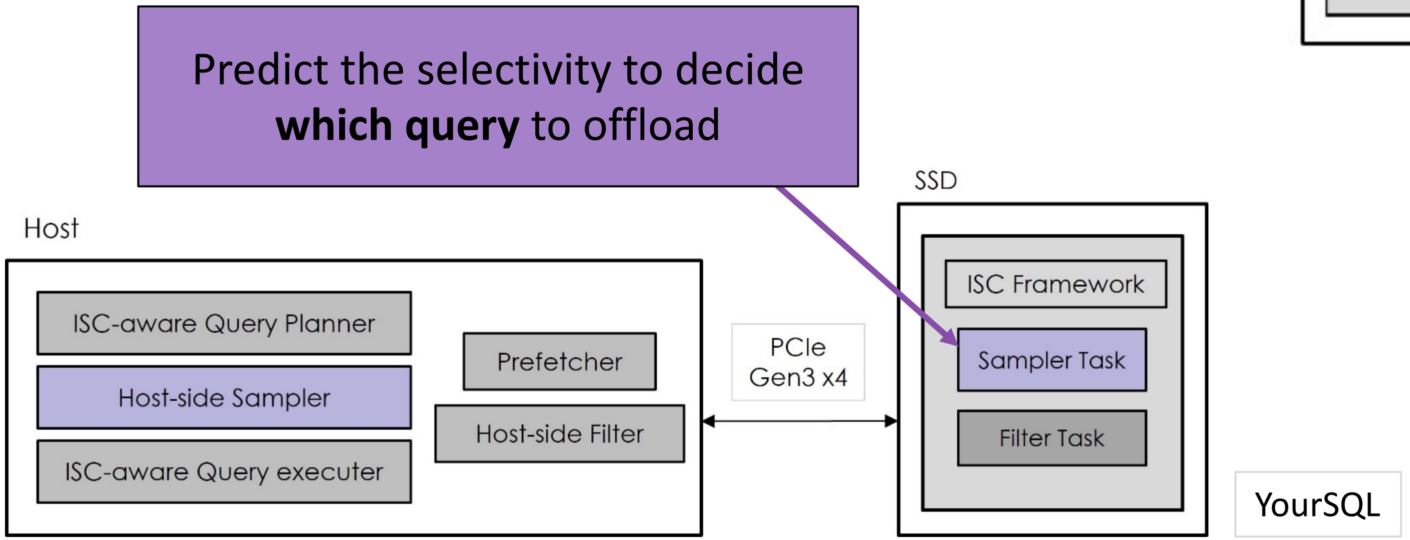


one kernel benefit from row-store and one kernel benefit from accessing key column consecutively

Biscuit, YourSQL (2016)

- Biscuit: a general framework
- As processing elements:
  - CPU and
  - Hardware pattern matcher in each channel of the SSD
- YourSQL: database system integrated with Biscuit




Predict the selectivity to decide **which query** to offload

Closing Thoughts

- Various aspects of the designed Smart Storage solutions
  - Location of processing elements
  - Deployment options
  - Evaluated applications
- Three identified principles:
  - Reduced data movement do not “automatically” guarantee performance improvements
  - Match the storage data layout with that of the application
  - Leveraging heterogeneous processing elements benefits efficiency

References:

- https://scaleflux.com
- https://semiconductor.samsung.com/resources/brochure/Samsung%20SmartSSD%20Computational%20Storage%20Drive.pdf
- Studied Papers:
  - Do, Jaeyoung, et al. "Query processing on smart ssds: Opportunities and challenges." *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 2013.
  - Woods, Louis, Zsolt István, and Gustavo Alonso. "Ibex: An intelligent storage engine with support for advanced sql offloading." *Proceedings of the VLDB Endowment* 7.11 (2014): 963-974.
  - Gu, Boncheol, et al. "Biscuit: A framework for near-data processing of big data workloads." *ACM SIGARCH Computer Architecture News* 44.3 (2016): 153-165.
  - Jo, Insoon, et al. "YourSQL: a high-performance database system leveraging in-storage computing." *Proceedings of the VLDB Endowment* 9.12 (2016): 924-935.
  - Koo, Gunjae, et al. "Summarizer: trading communication with computing near storage." *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. 2017.
  - Ruan, Zhenyuan, Tong He, and Jason Cong. "{INSIDER}: Designing {In-Storage} Computing System for Emerging {High-Performance} Drive." *2019 USENIX Annual Technical Conference (USENIX ATC 19)*. 2019.
  - Xu, Shuotao, et al. "Aquoman: An analytic-query offloading machine." *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2020.
  - Qiao, Weikang, et al. "FANS: FPGA-Accelerated Near-Storage Sorting." *2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2021.
  - Salamat, Sahand, et al. "NASCENT: Near-storage acceleration of database sort on SmartSSD." *The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 2021.
  - HeydariGorji, Ali, et al. "Leveraging Computational Storage for Power-Efficient Distributed Data Analytics." *ACM Transactions on Embedded Computing Systems (TECS)*.