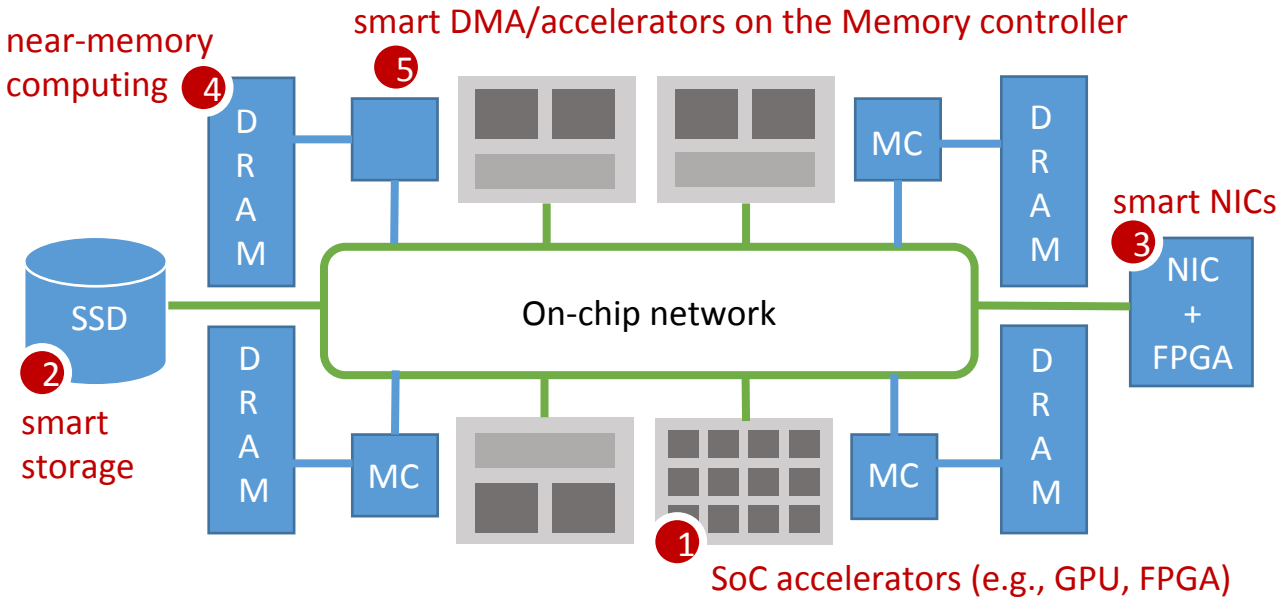
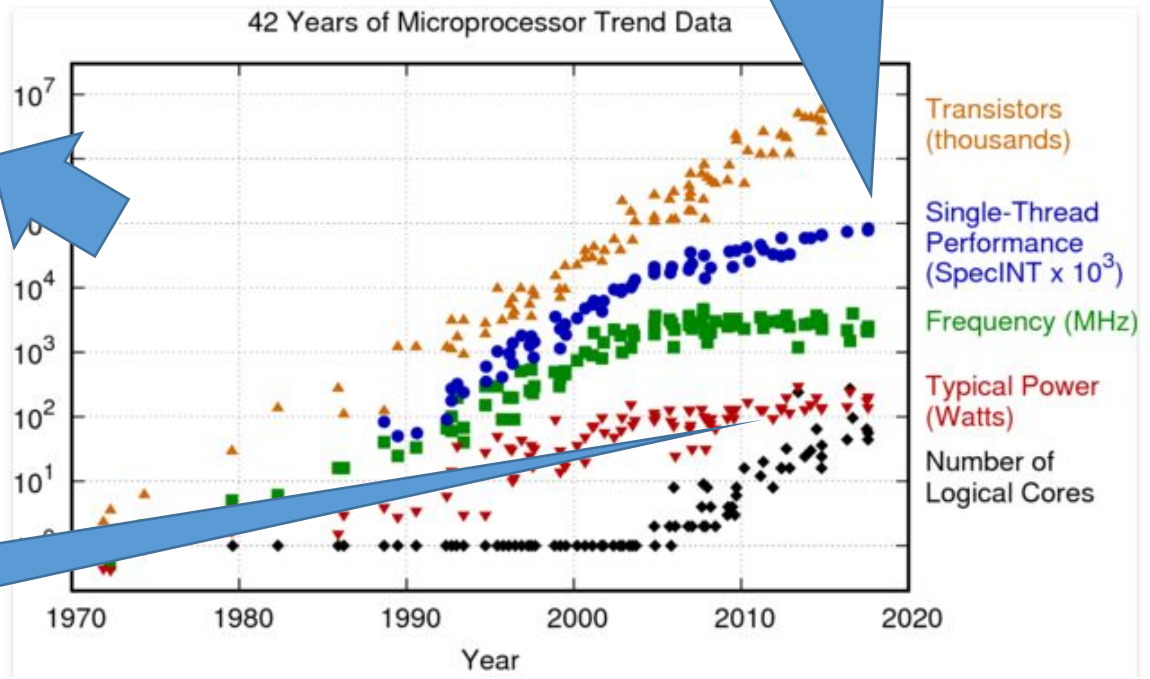
Active Heterogeneous Hardware and its Impact on *System Design*

Jana Giceva
Department of Informatics, TUM

Hardware trends



Only 3% annual increase in last years



Dennard's scaling and power limitation leaves us with one option – specialization

src: https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/

The new Golden Age for Computer Architecture

- Patterson and Hennessy – Turing Award Lecture in 2018
- The next focus should be in:
  - Domain Specific Architectures (DSAs)
  - Big expectations for performance efficiency by linking DSLs to DSAs

Allow the programmer to express the semantics of a program in a high level language and then let the system and compiler do optimizations for the underlying architecture.

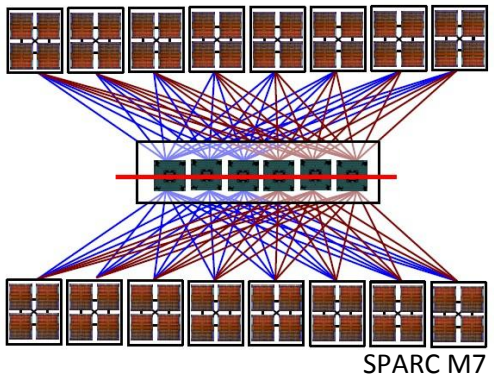
- This is the same philosophy that databases use for decades.

- Time to resurrect the idea for a database machine?

[DIRECT -- DeWitt'78, Gamma Database Machine – DeWitt et al.'90]
[RAPID @ SIGMOD'18, DPU @ MICRO'17, Q100 @ MICRO'15, etc.]

Driving trends for systems research

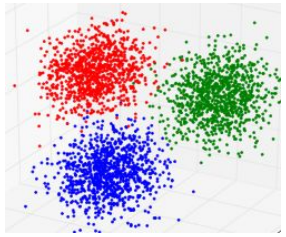
Modern hardware



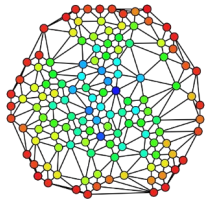
- Increasing size and complexity
- Heterogeneous resources
- Diversity among machines

Application and workload

HTAP and BI



ML



Graph processing

- Data intensive
- Efficient resource usage
- Predictable performance

Opportunity 1: Database technology for the masses

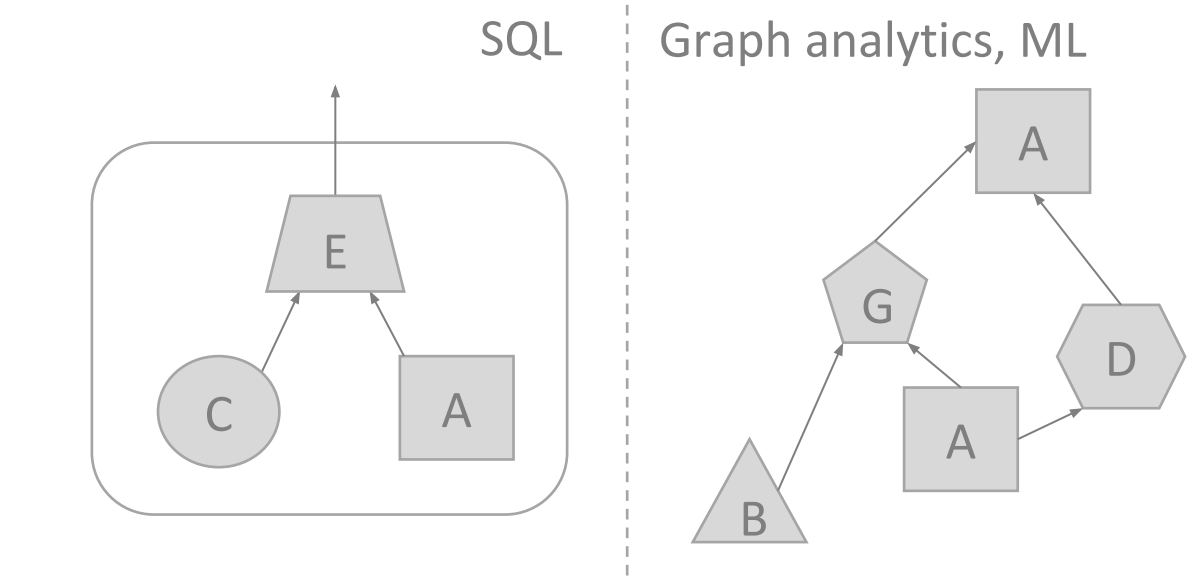
Driven by hardware developments:

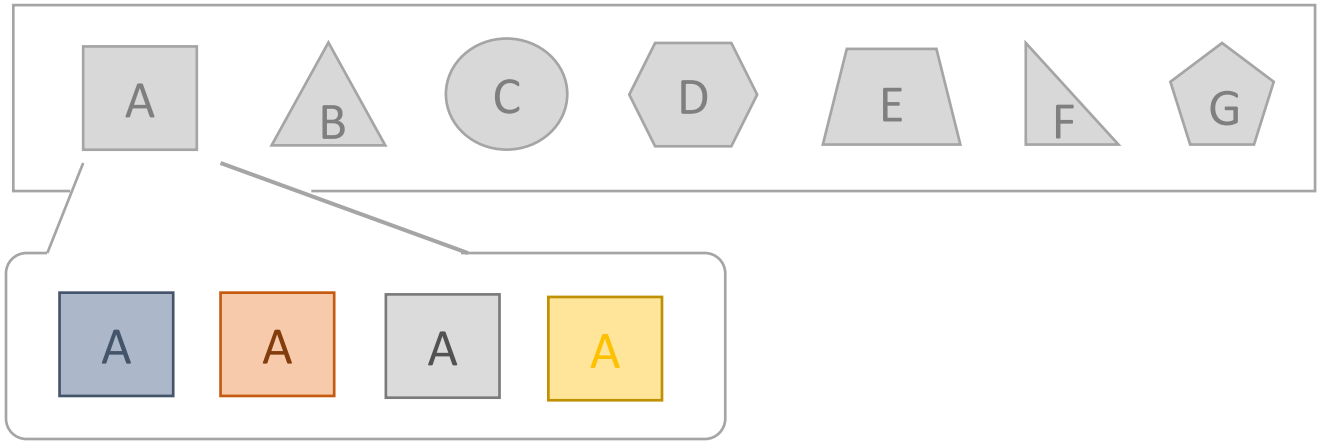
- DSLs for particular application domains
  - e.g., XLA, GreenMarl, LINQ, etc.
- Compilation and optimisation techniques
  - e.g., Bohrium, DLVM, Dimwitted, Weld, Voodoo, etc.
- Cross compilation to run on various platforms (CPU, GPU, FPGA)
  - e.g., TVM, GraphGen, OpenCL, etc.

- Great deal of DB technology that is being mirrored or could be reused.

- Questions: should we use this opportunity to rethink databases and join the effort of extending their support for modern workloads?

An idea: lose SQL operators and make sub-operators first class citizens





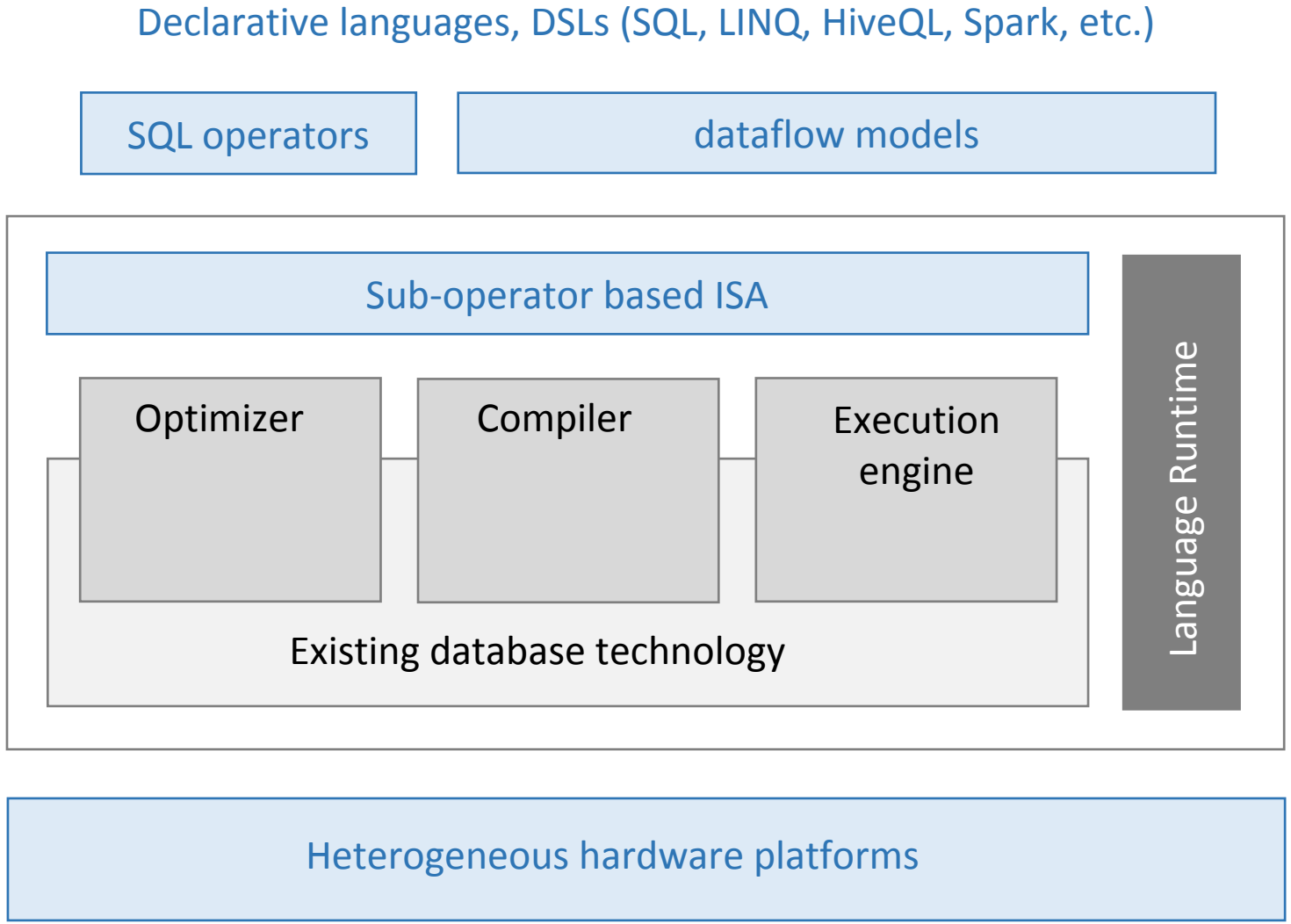
HW platform implementations of sub-operator A

Flexible for constructing various dataflows, both SQL and more complex analytics.

Sub-operators are logical functions that perform basic data transformations and management tasks

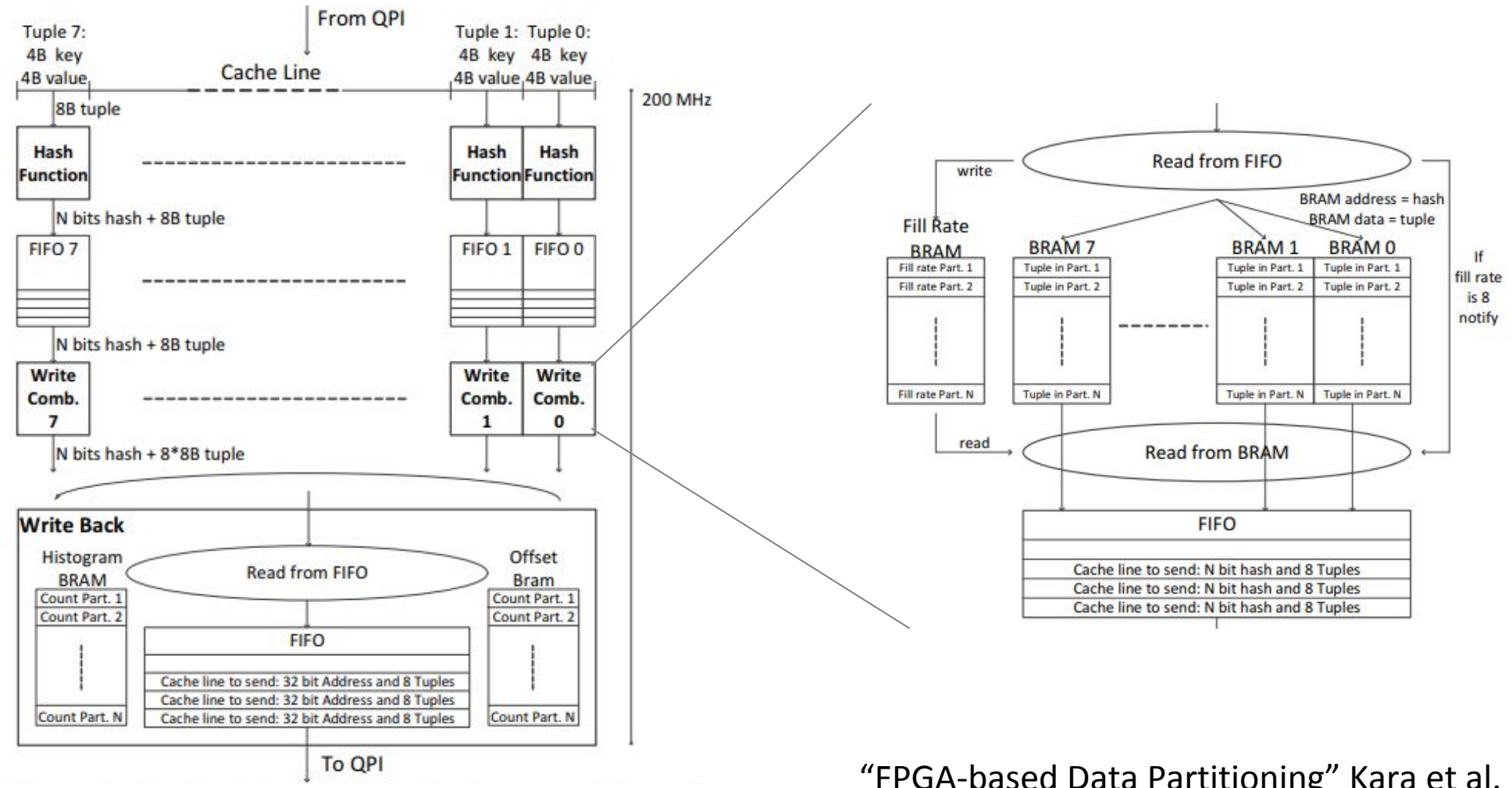
Granularity chosen such that we not only benefit from more efficient compilation to CPU/GPU but also to offload computation to where data sits and moves.

Sub-operator based system architecture



Example sub-operator: data partitioning (FPGA-based)

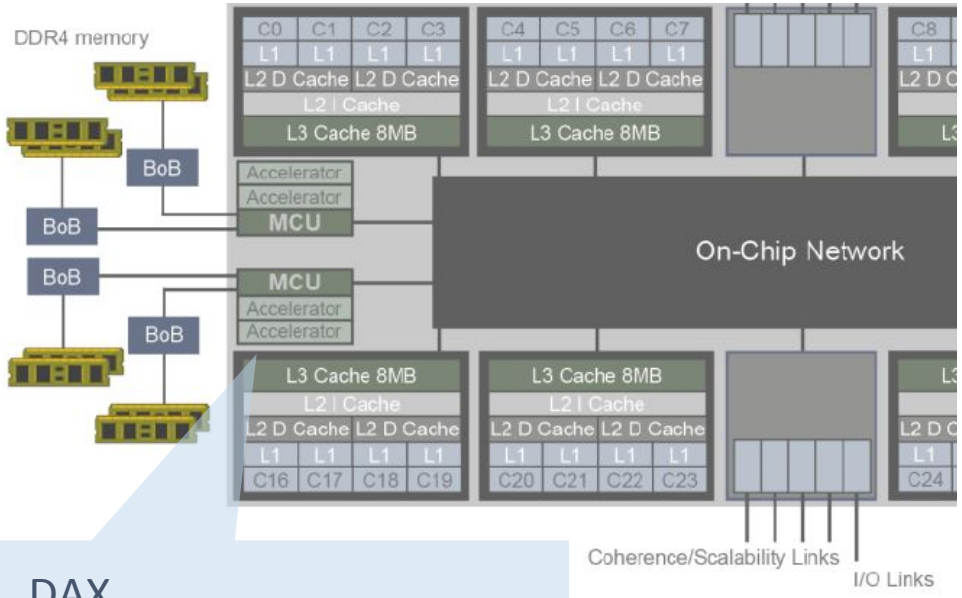
Hybrid (FPGA/CPU) data processing, e.g., FPGA-based data partitioning



"FPGA-based Data Partitioning" Kara et al. [SIGMOD'17]

Industry Example #1 (Oracle)

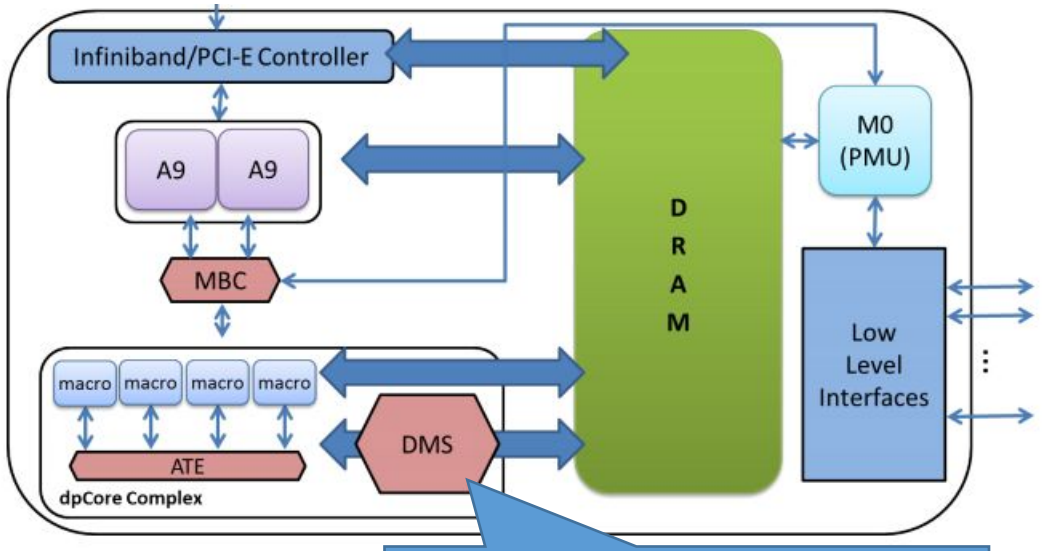
Oracle's SQL in Silicon – SPARC M7



DAX
- In-line compression, decompression
- Bloom-filter
- Predicate evaluation
- Filtering by bit-vector
- Encryption

src: White Paper, August 2016

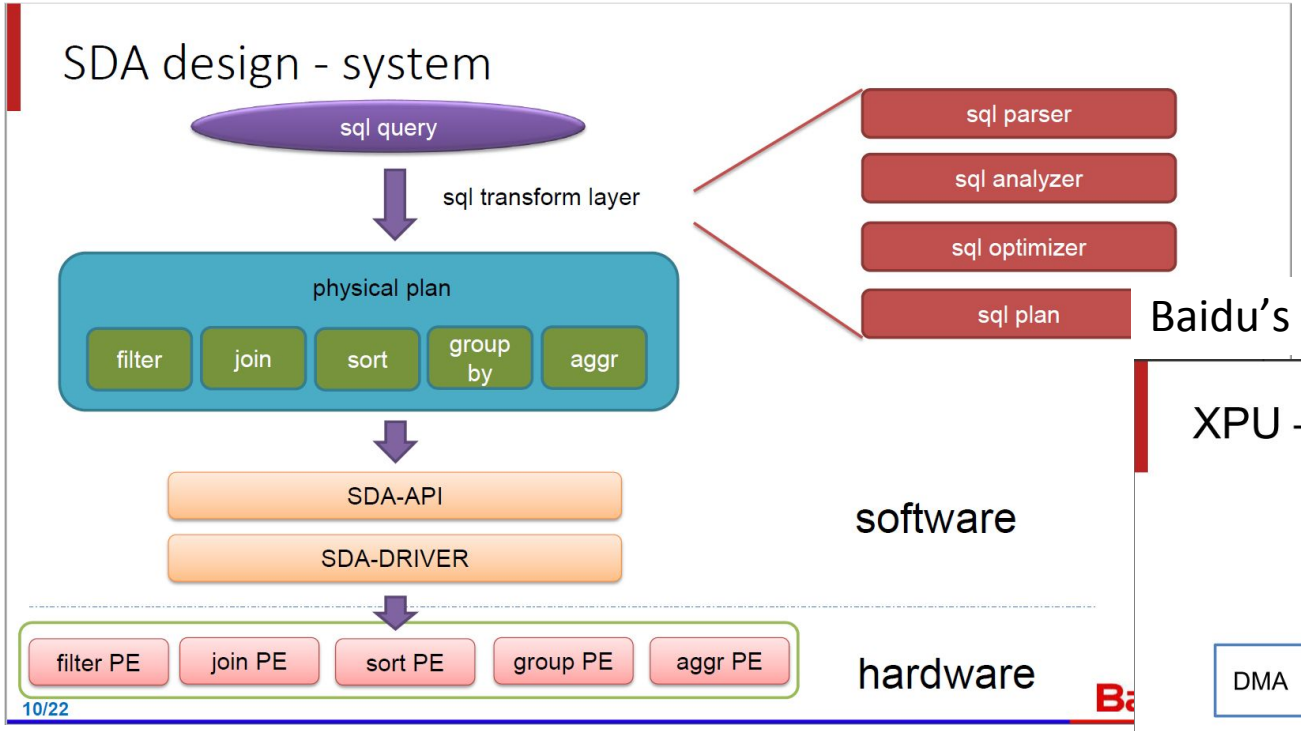
Oracle Lab's DPU (MICRO'17)



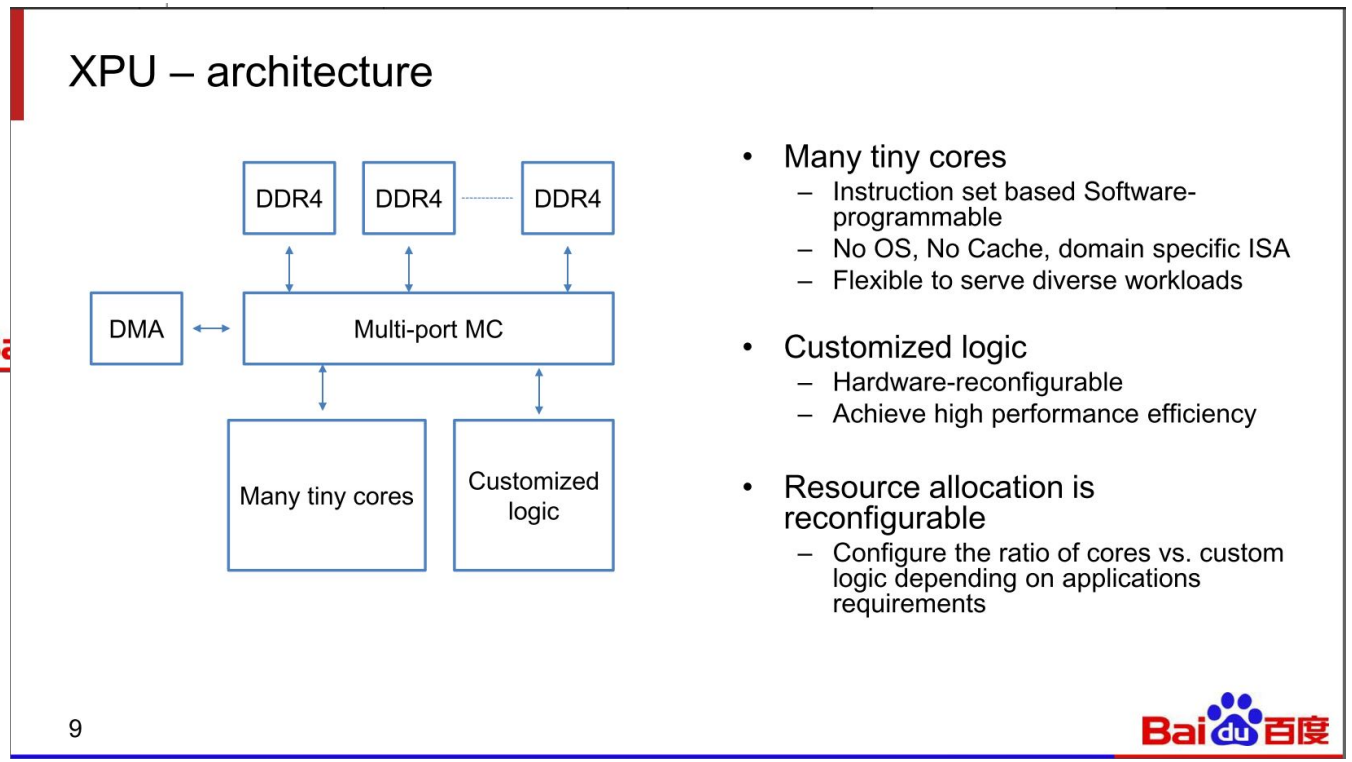
Data Movement System (DMS)
- Filter and projection
- hash/range partioning,
- scatter/gather

Industry Example #2 (Baidu)

Baidu's SQL in the Cloud (Hot Chips'16)



Baidu's XPU (Hot Chips'17)



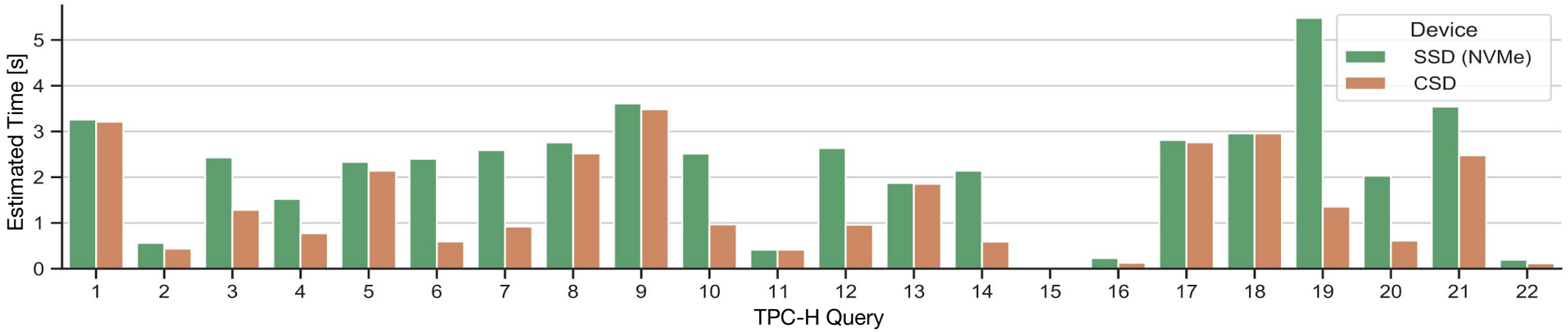
Computational storage

Example on-going project done by Maximilian Bandle pushing computation to data storage

Device: SSD + dedicated ARM A53 SoC (+ FPGA) for computational tasks

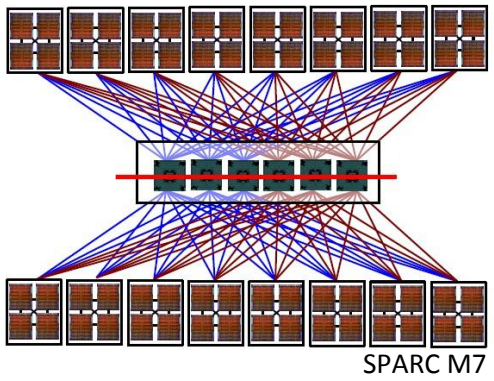
Investigating:

- What pays off to be offloaded?
  - Datasets, workloads, operators
- Building an analytical model of device's resource capacities
  - Compute power, Internal/external bandwidth ratio, memory transfer costs, accelerators, energy efficiency



Driving trends for systems research

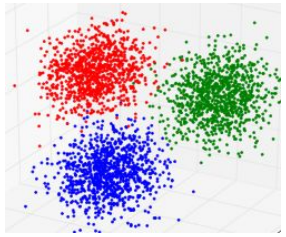
Modern hardware



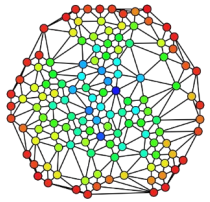
- Increasing size and complexity
- Heterogeneous resources
- Diversity among machines

Application and workload

HTAP and BI



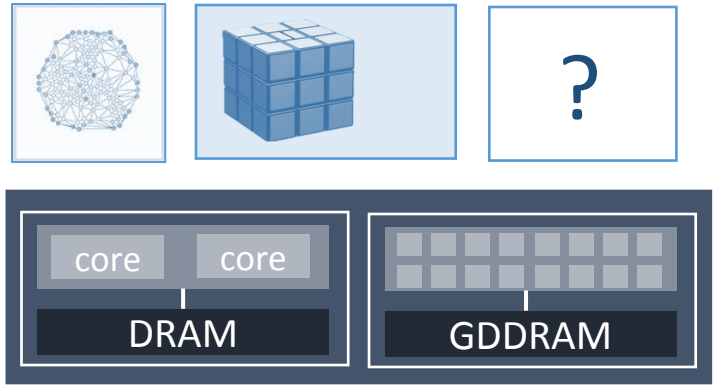
ML



Graph processing

- Data intensive
- Efficient resource usage
- Predictable performance

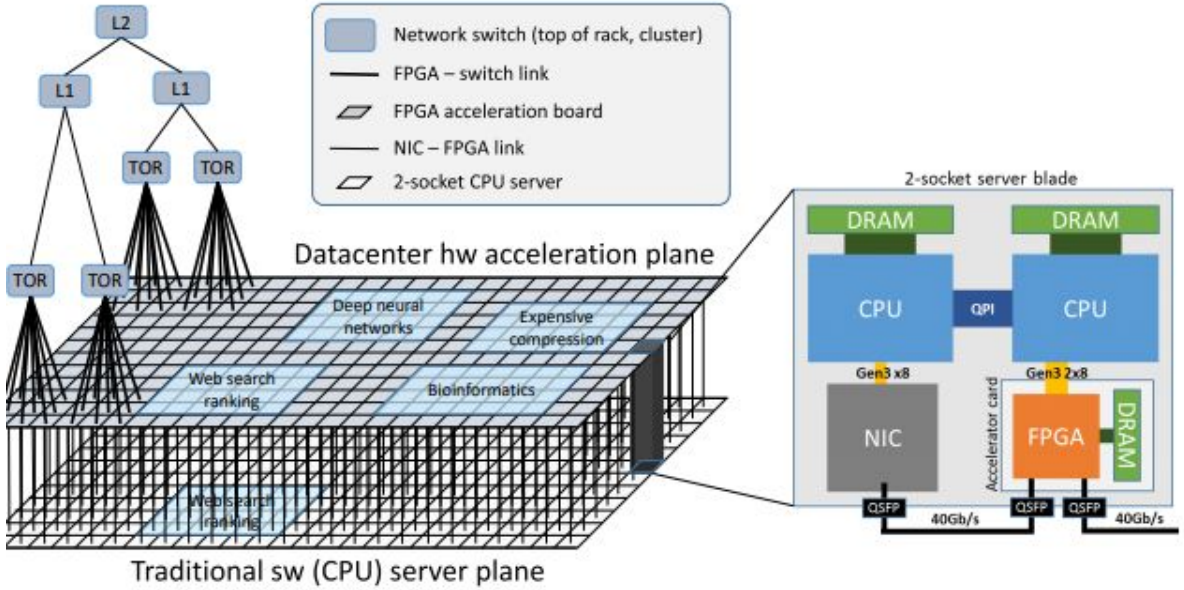
Deployment



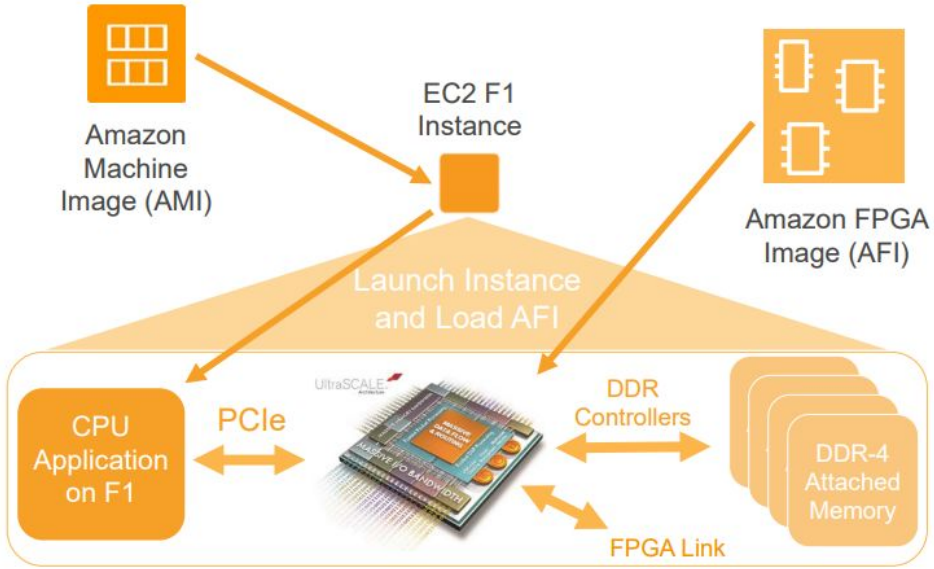
- Server consolidation
- Virtualization
- Multi-tenancy

Accelerators are deployed in the cloud

Microsoft's Azure configurable cloud (MICRO'16)



Amazon's FPGA-acceleration using F1 (HotChips'17)

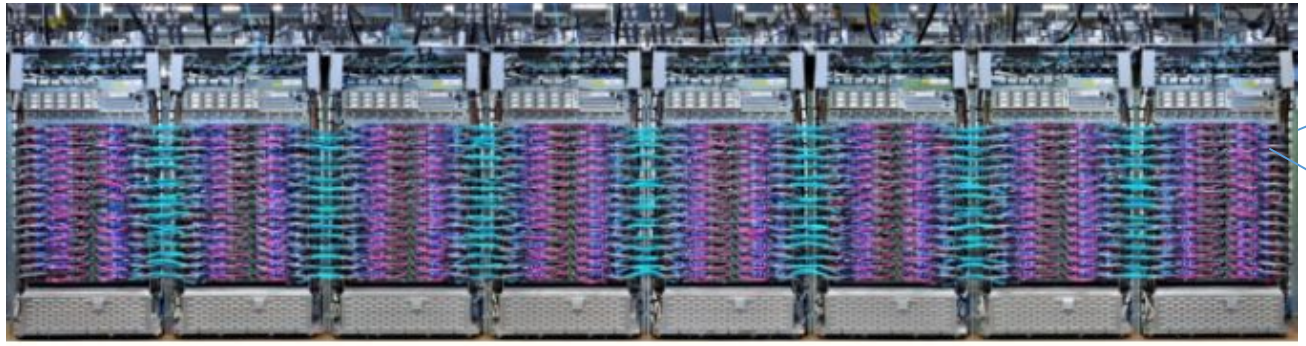


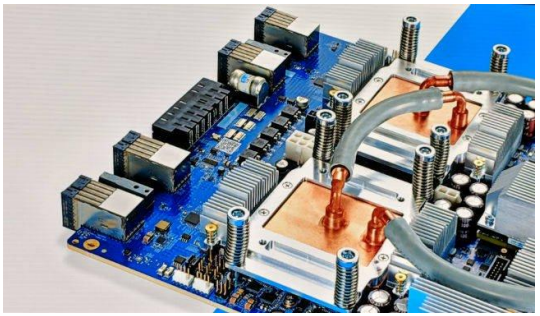
An F1 instance can have any number of AFIs

An AFI can be loaded into the FPGA in less than 1 second

amazon web services

Google's TPU 3.0 pods (Google I/O 2018)





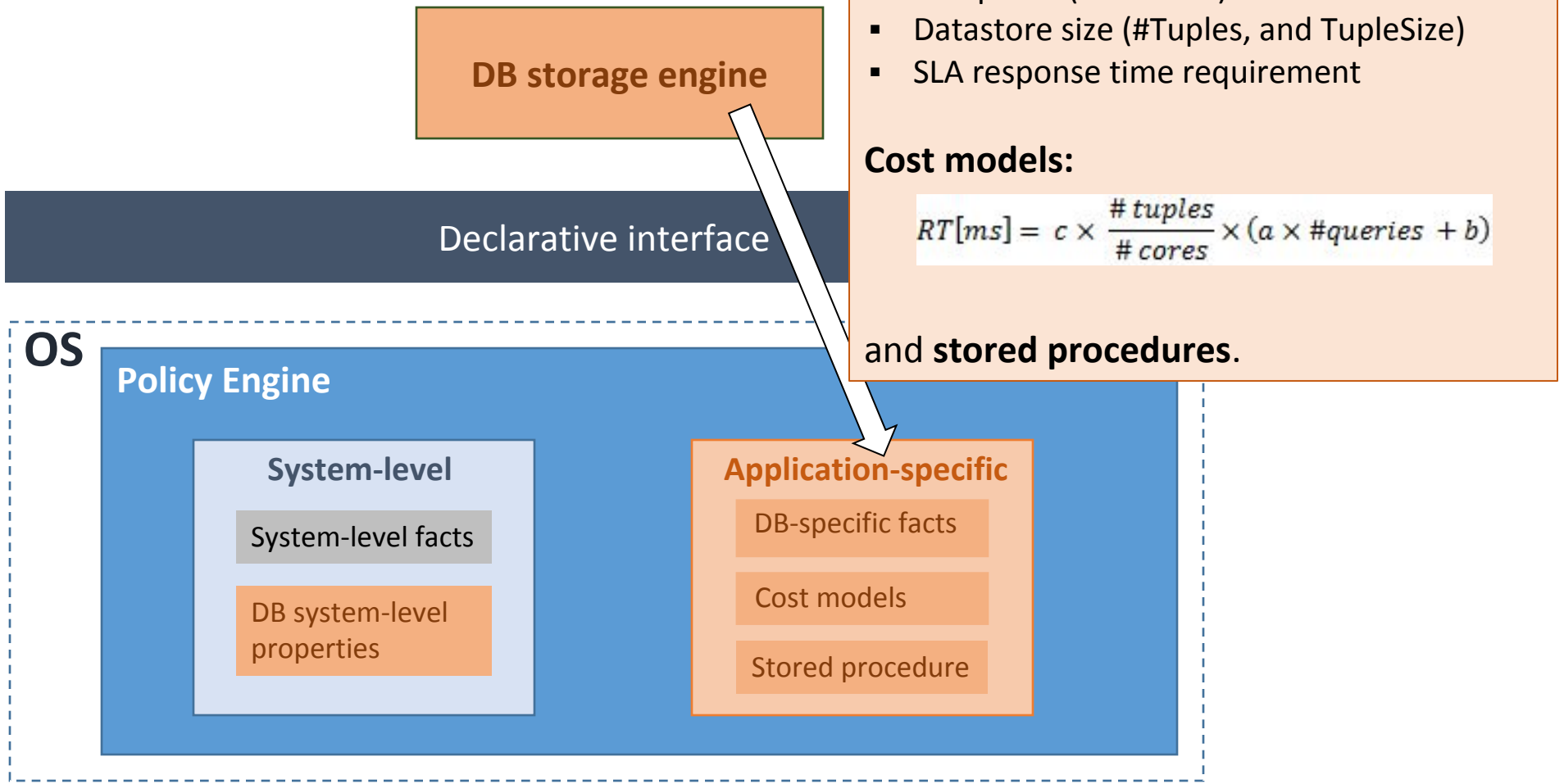
… and they are disaggregated resources

- Almost all are accessible over a high-bandwidth network
- So the CPU and the traditional OS no longer have the high-level overview

Which opens many interesting systems (research) questions, which we could maybe discuss in (after) this retreat?

- How does this impact the load on the network?
- What's the (new) role of the switch?
- Can we explore declarative interfaces, pushing down application-level semantics (cost models) to the ***brain*** (control plane) overlooking the networked resources?

OS Policy Engine



Push **database specific facts:**

- #Requests (in a batch)
- Datastore size (#Tuples, and TupleSize)
- SLA response time requirement

Cost models:

$$RT[ms] = c \times \frac{\#\,tuples}{\#\,cores} \times (a \times \#queries + b)$$

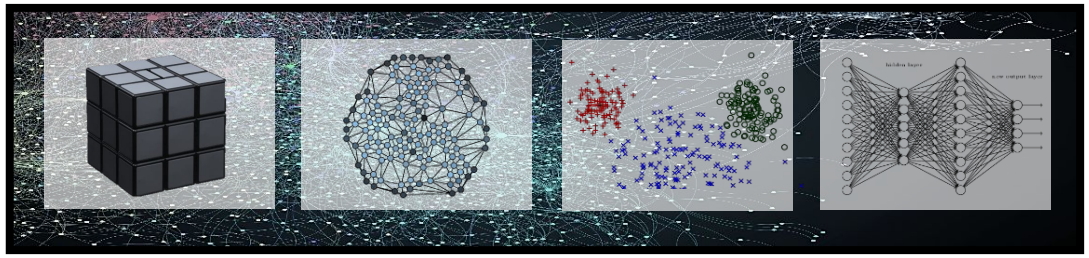
and **stored procedures**.

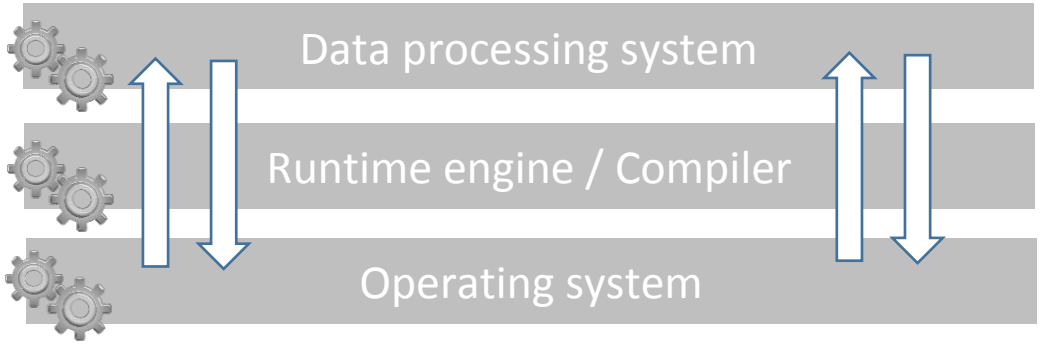
Many implications for systems design

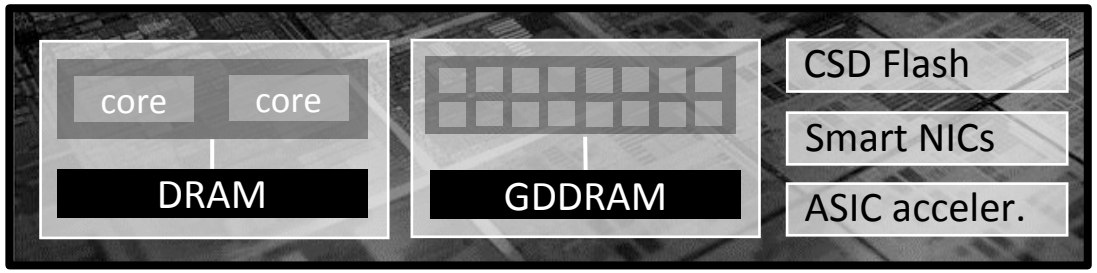
- **Which function should be offloaded and to which device?**
  - Recall that *active components* can be anywhere
  - The granularity of the function-offload depends on the device's capabilities and where the data is at that time
- **How to program the active hardware? What is the interface? DSAs?**
- **How to best leverage compilers?**
  - We are investigating the potential of the MLIR proposal
- **How to decide which computation to offload?**
  - Generate suitable cost-models, runtime resource monitoring
  - Revisit the wisdom of database optimizers
- **Who manages the active components? Should they be context-switched?**
  - How do we virtualize them? Where is the control plane?
  - Their own drivers, or managed by the OS, application, runtime, switch, network?
- **What is the failure domain?**

Requirement for a holistic solution

Modern workloads: data analytics







Modern machines

- Current trends require a holistic approach and cross-layer optimization.
- Needs joint work by:
  - ***Systems***: OS, runtime, networking, virtualization, security.
  - ***Data management:*** data-flows, cost-based optimizers, workload requirements.
  - ***Compilers and PL***: knowledge-transfer through layered IRs between DSLs and DSAs.
  - ***Computer architects*** to influence the future DSAs, architectures and devices.

Computational storage

Example on-going project done by Maximilian Bandle pushing computation to data storage

Device: SSD + dedicated ARM A53 SoC (+ FPGA) for computational tasks

Current status:

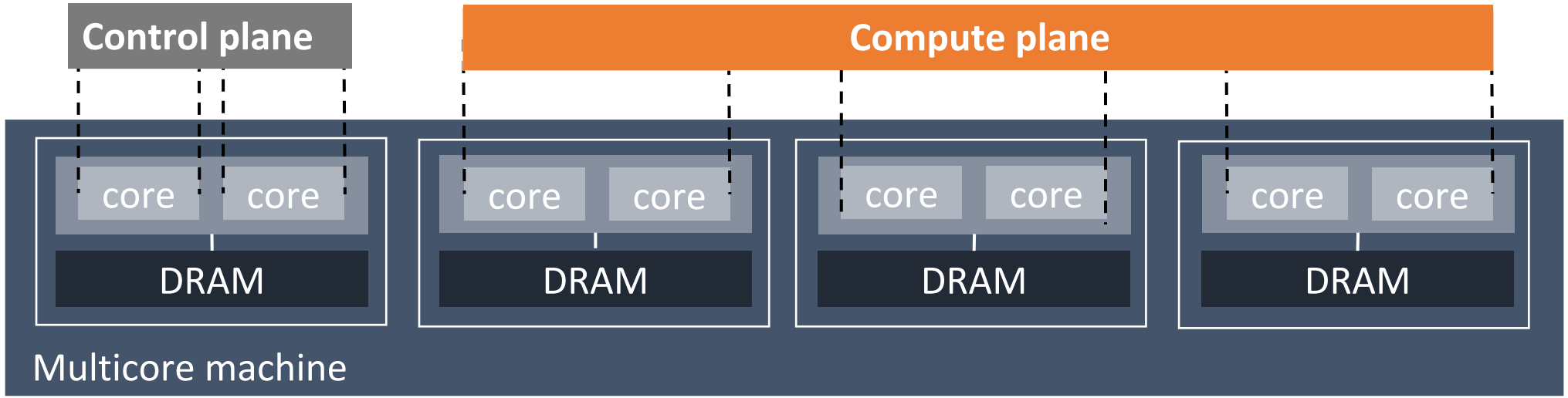
- Umbra database runs on ARM
- Existing model suggests non-trivial savings even for simple push-downs

Next steps:

- End-to-end query execution between the host and the CSD
- Integrating the cost-modeling for when to offload to CSD in the optimizer.

Stay tuned!

OS support for data processing

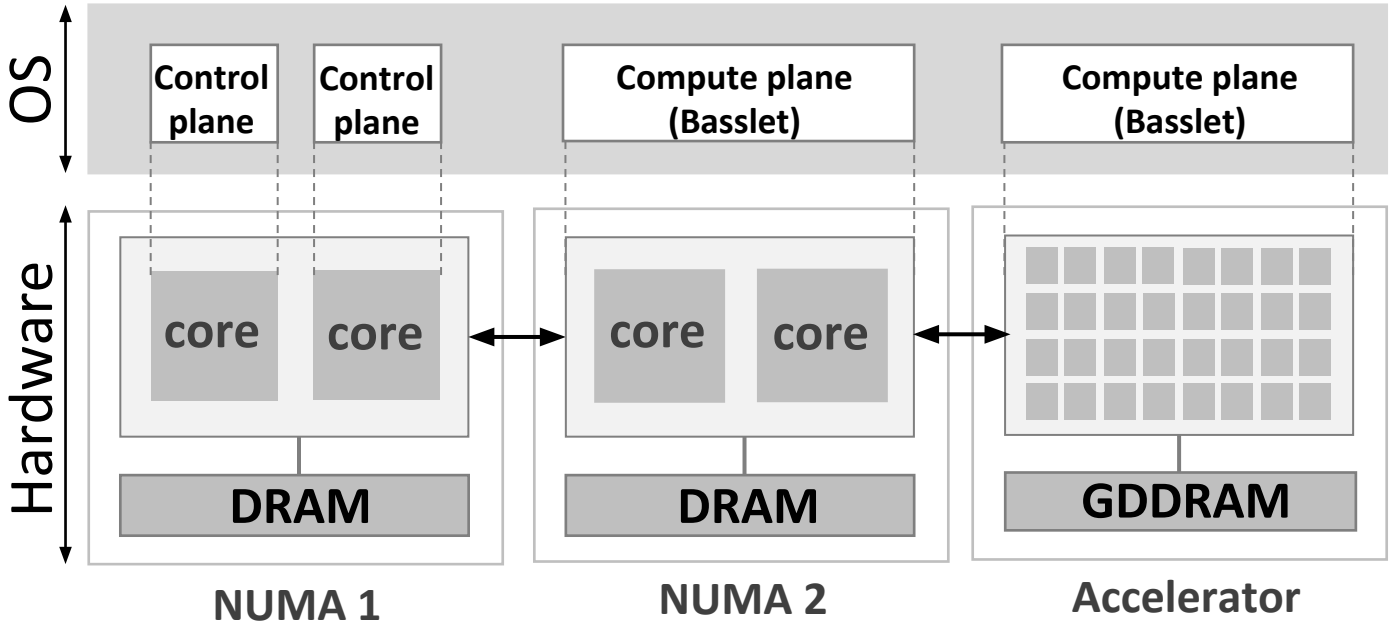


1. Leverage the multi-kernel model (*e.g.*, Barrelfish [SOSP'09]).
2. Split the machine's resources into a *control* and a *compute* plane.
3. *Specialize* the compute plane kernels for parallel data-processing.

In collaboration with Gerd Zellweger (now at VMWare Research)

Customizing the OS for accelerators

- Optimize the OS kernel for heterogeneous architectures



- API for DAG-like jobs that can easily offload tasks to the accelerator transparently
- Link the mechanism to the OS policy engine and the optimizer.

Ongoing work by Daniel Grumberg at Imperial College London

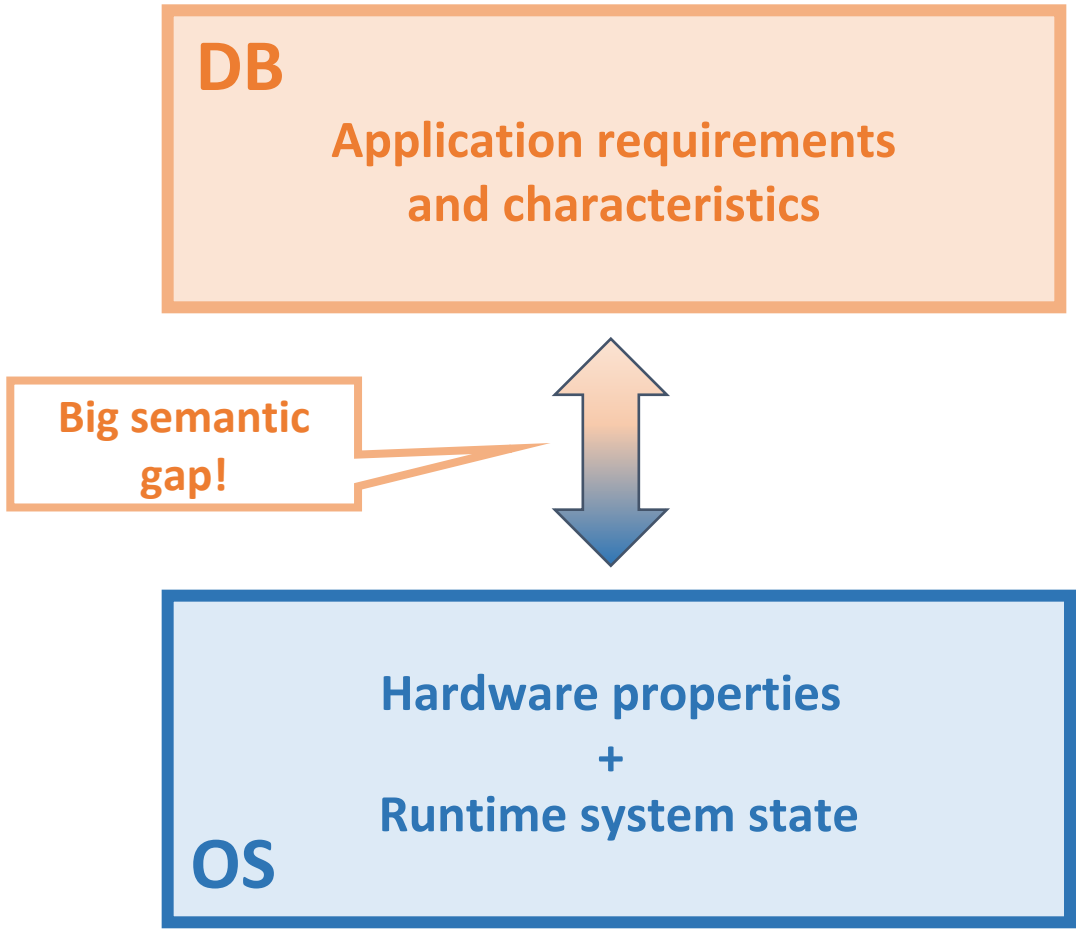
Systems support for heterogeneous hardware

Cloud providers: Microsoft, Amazon, Google, etc.
Platform providers: Intel, Nvidia+ARM, Mellanox, Xilinx

From the research side (just a small subset):

- The Multikernel: A new OS architecture for scalable multicore systems [OSDI'08]
- Helios: Heterogeneous Multiprocessing with Satellite Kernels [SOSP'09]
- IX: Dataplane Operating System [OSDI'14]
- Arrakis: OS is the control plane [OSDI'14]
- M3: A HW/OS co-design to tame heterogeneous manycores [ASPLOS'16]
- Popcorn – OS support for heterogeneous ISA [EuroSys'15, ASPLOS'17]
- LITE – OS support for RDMA in the data centers [SOSP'17]
- Solros – a data-centric OS for heterogeneous computing [EuroSys'18]
- LegoOS: A disseminated, distributed OS for Hardware Resource Disaggregation [OSDI'18]
- SemperOS: A Distributed Capability System [ATC'19]
- Automatic Virtualization of Accelerators [HotOS'19]
- When should the network be the computer? [HotOS'19]
- LeapIO: Efficient and Portable Virtual NVMe Storage on ARM SoCs [ASPLOS'20]
- Do OS abstractions make sense on FPGAs? [OSDI'20]

DB/OS knowledge gap

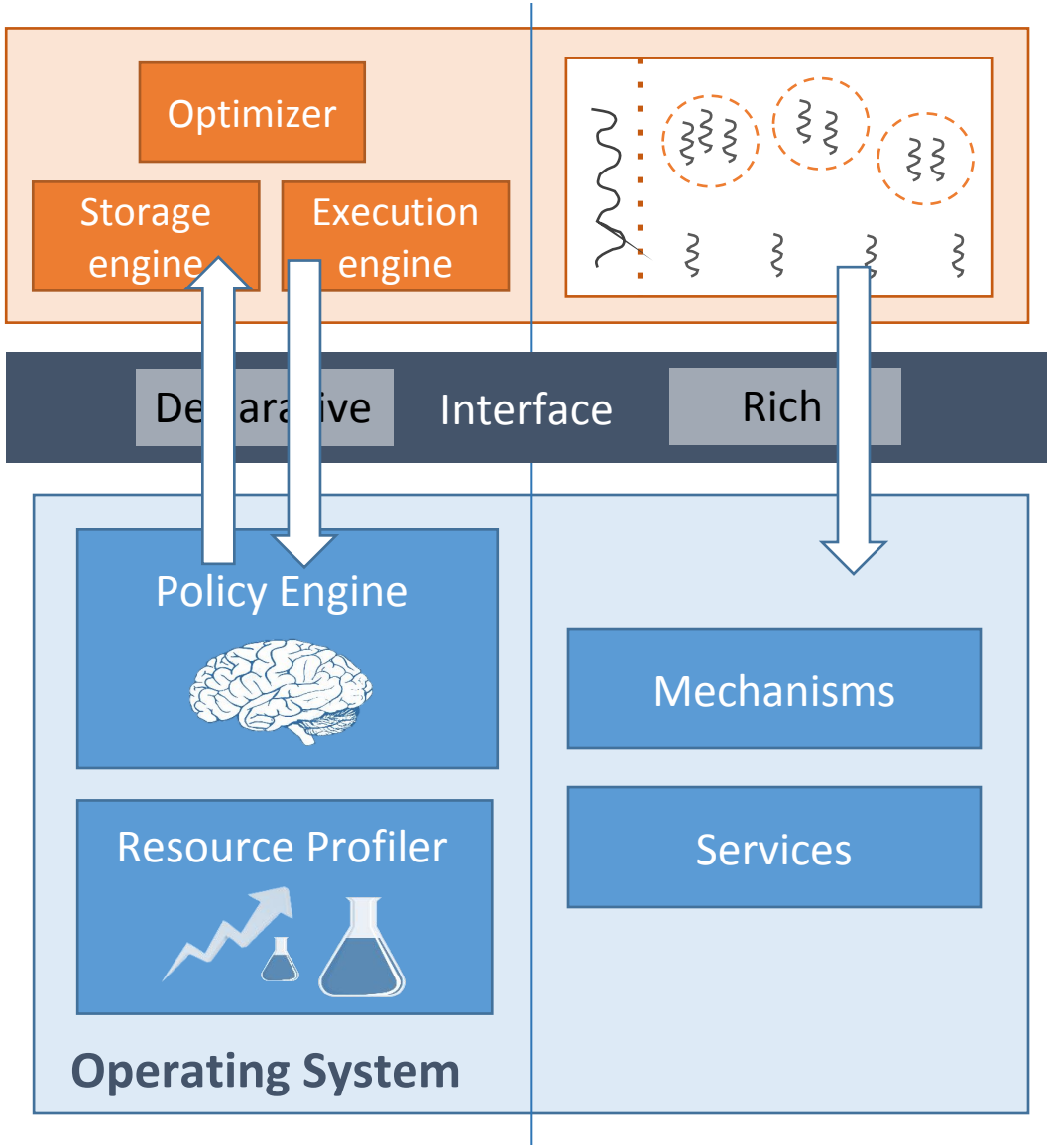


DB/OS co-design

Address the knowledge gap

- Who knows what? Where should knowledge reside?
- How can the OS help with HW complexity & diversity?
- What DB knowledge can improve OS policies?

[CIDR'13, VLDB'14]

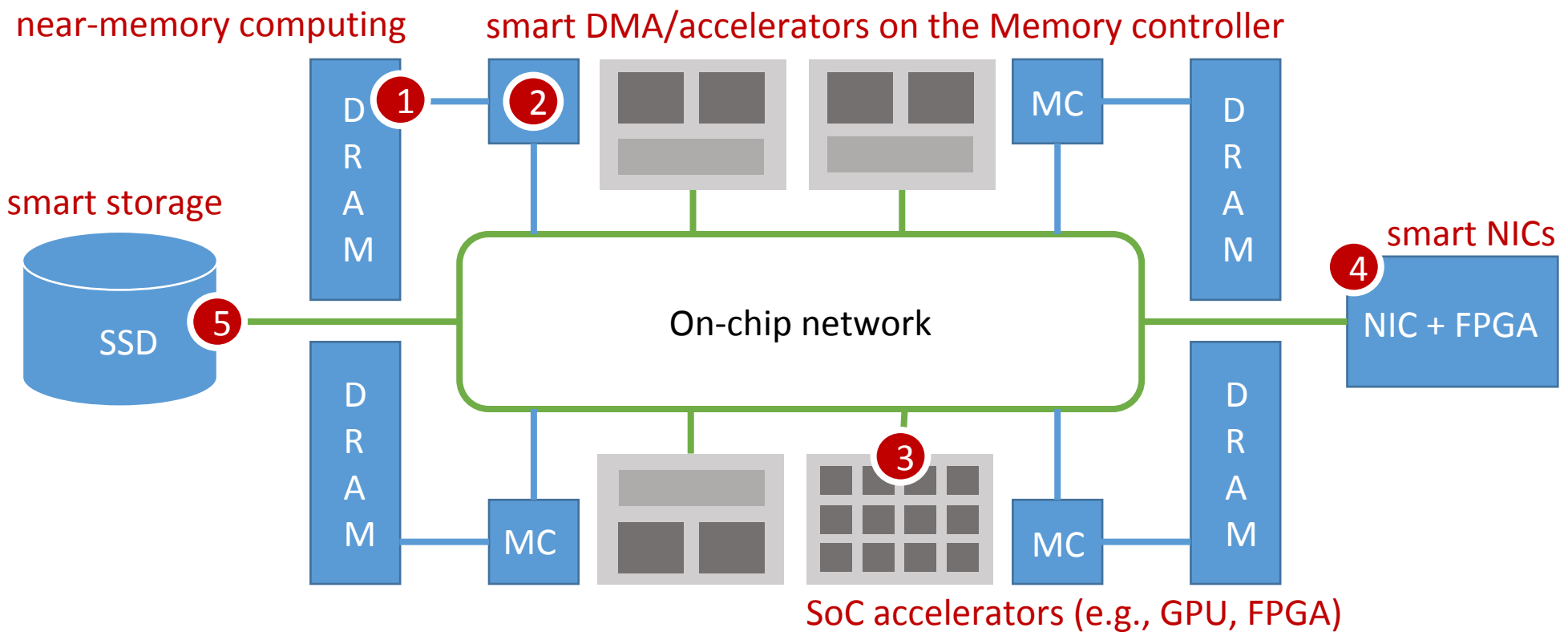


Customize the OS kernel

- Where does the OS gets in the way? What's redundant?
- What mechanisms are needed by modern workloads?
- What OS design can enable kernel customizations?

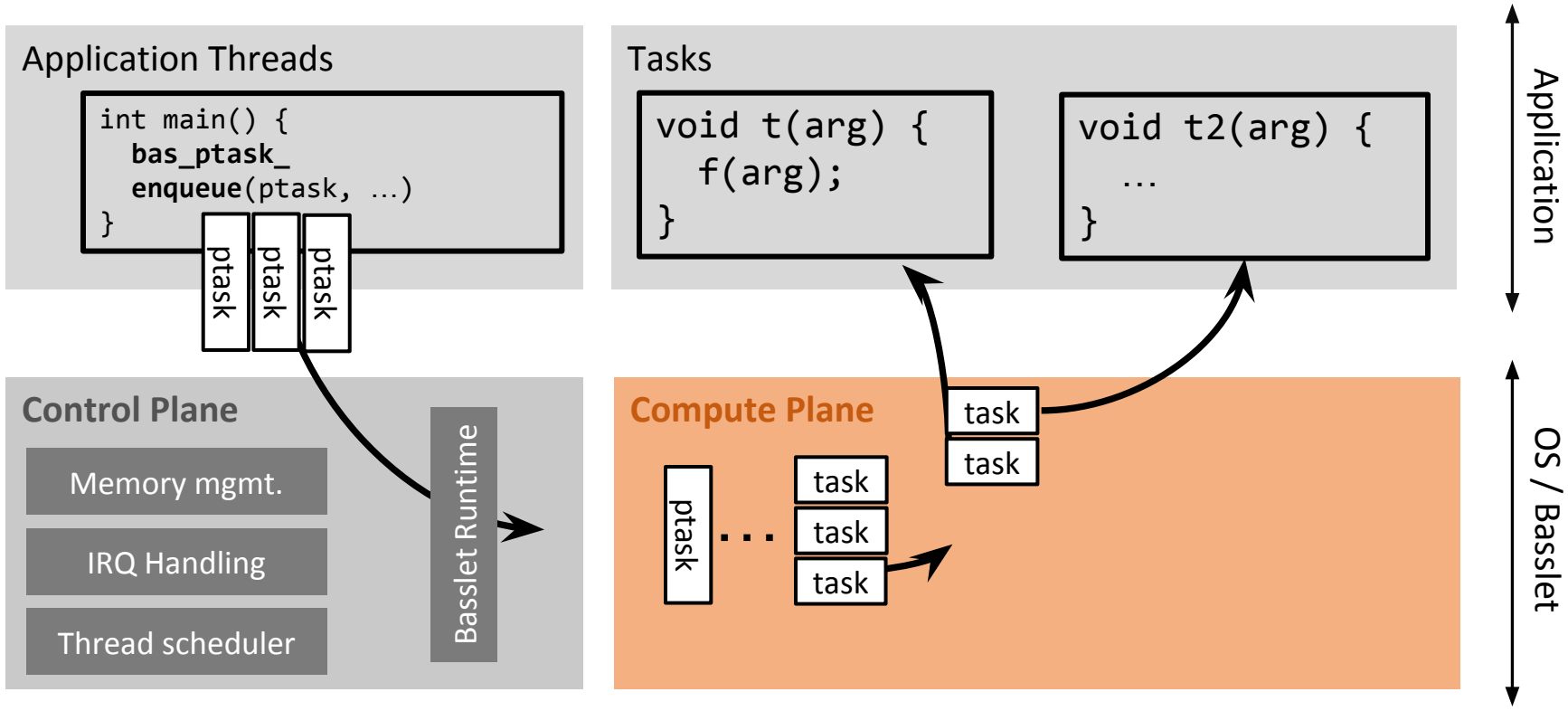
[DaMoN'16]

Where can we find the accelerators?

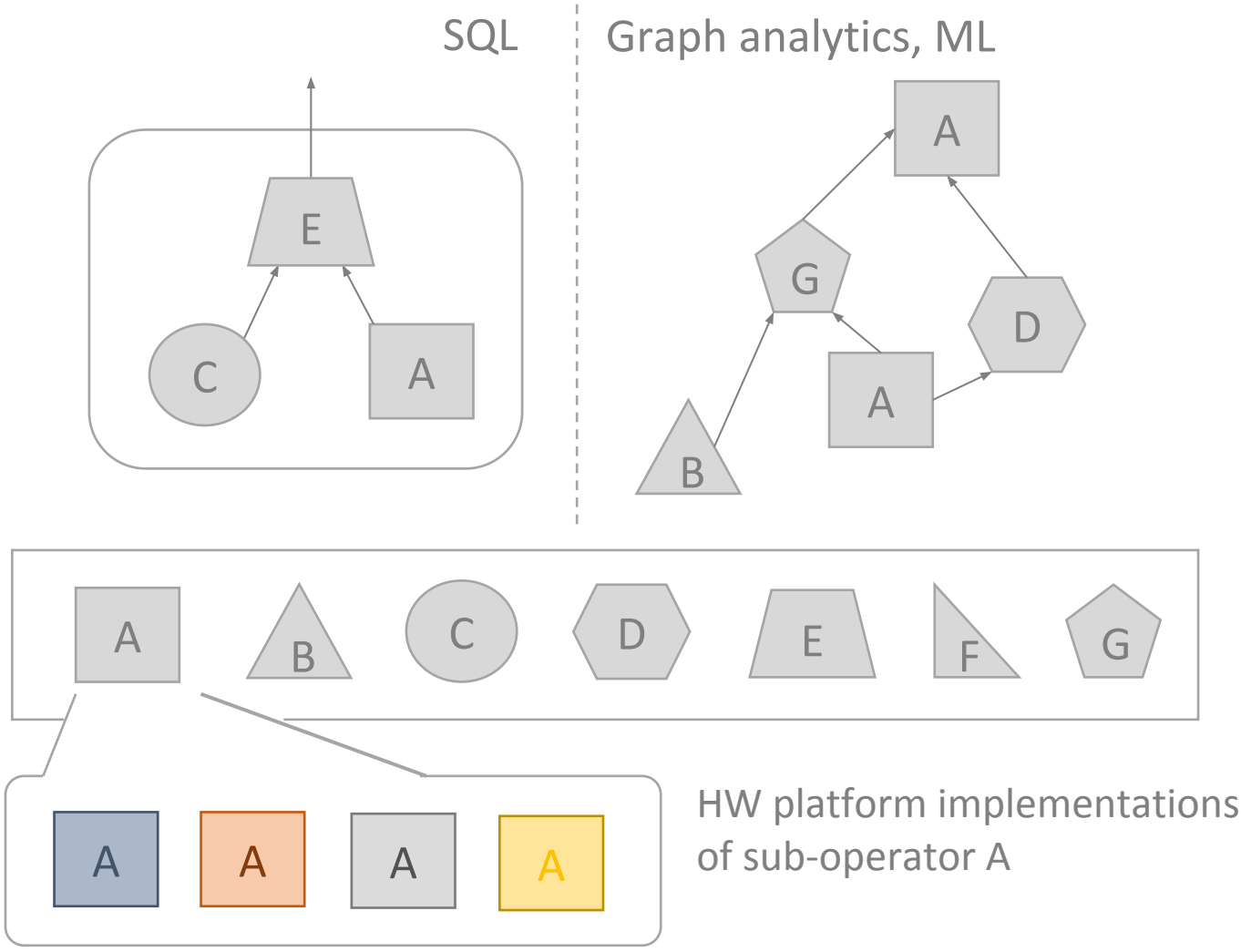


Intel HARP, Xilinx, IBM
DPU, XPU, DAX
NPU, Configurable Cloud

Compute plane – how does it work?



An alternative: sub-operators as first class citizens



- Flexible for constructing complex dataflows
- Operator primitives for hardware acceleration on- and off-chip
  - e.g., SPARC M7 data accelerators (DAX)
- Hybrid data processing (e.g. FGPA/CPU)
  - FPGA-based data partitioning [Kara et al., SIGMOD'16]
  - Stochastic gradient decent (SGD)