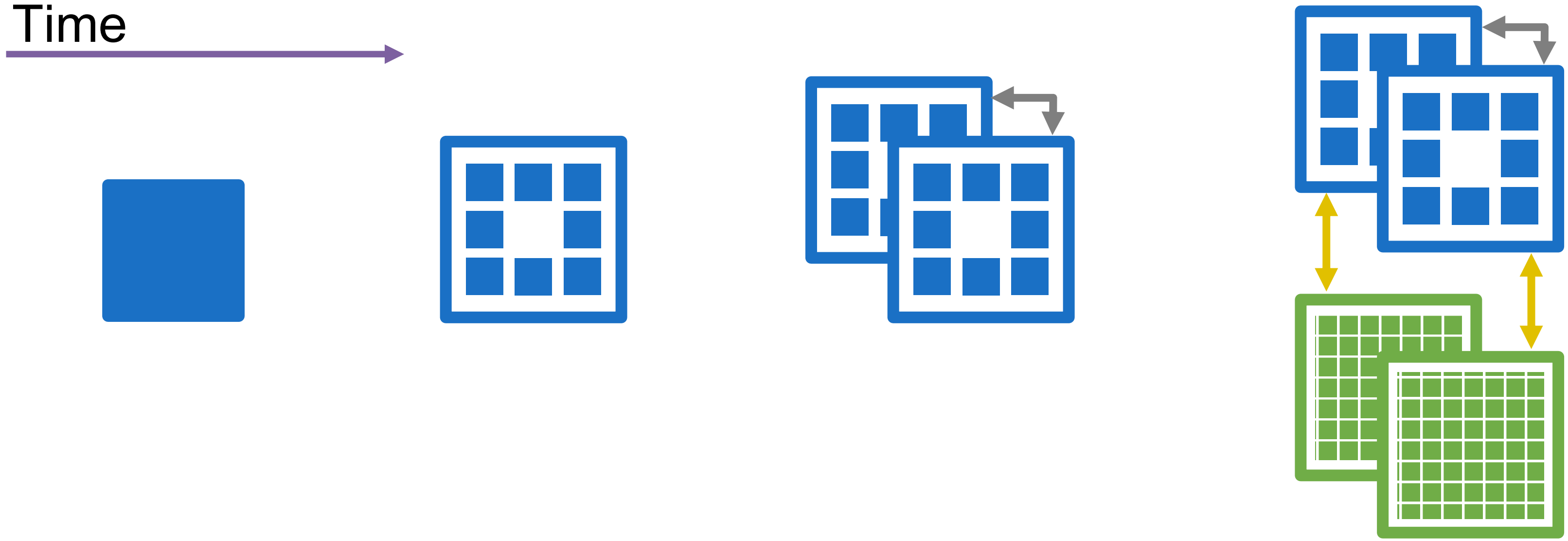
Unlocking GPU potential with JIT

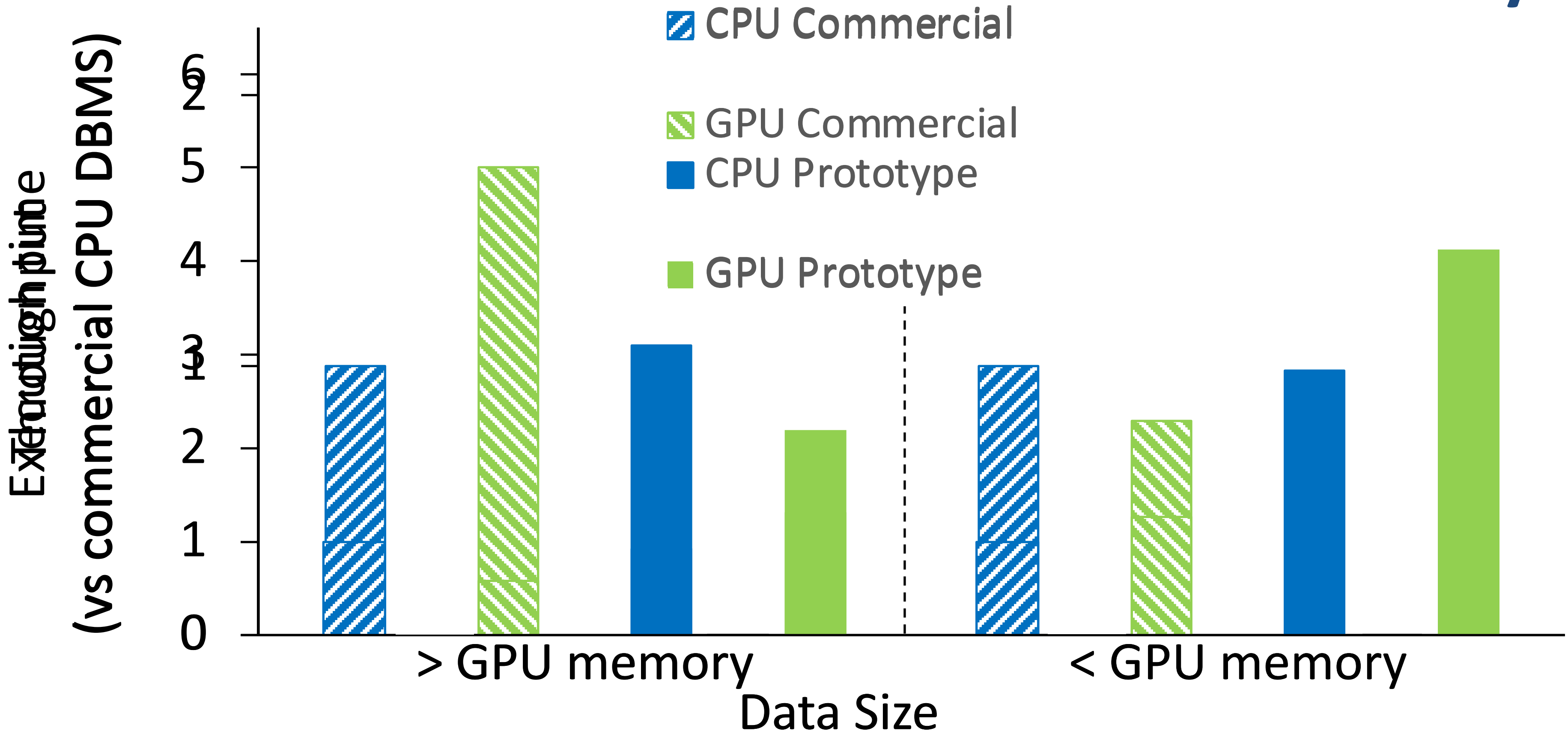
Anastasia Ailamaki
with Periklis Chrysogelos and Panos Sioulas

One hardware does not fit all



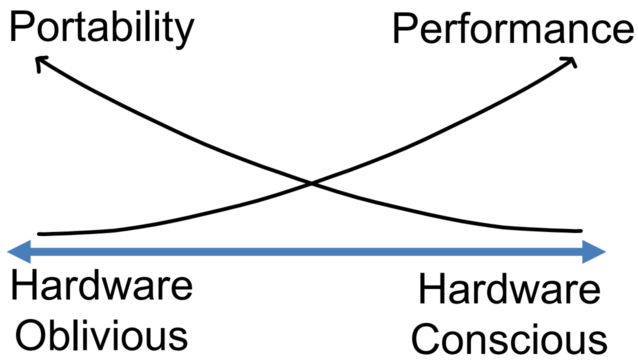
Rethink query engines for accelerator-level parallelism

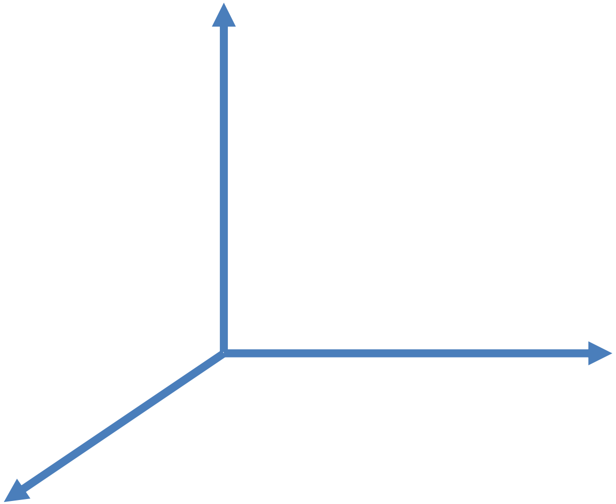
One hardware fits all: The end of an efficient story



20%-80% throughput loss due to lack of portability

Designing query engines for heterogeneous HW



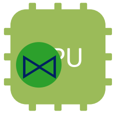


Decomposition of design space to find sweet spot

OLAP in heterogeneous servers: design space

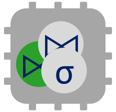
[CIDR2019]

intra-operator

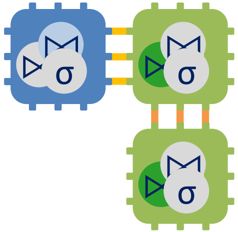


- Performance depends on μ-arch
- Tune operators to memory hierarchy specifics

intra-device



- Portability impacted by specialization
- Inject target-specific info using codegen



inter-device

- Limited device inter-operability
- Encapsulate heterogeneity and balance load

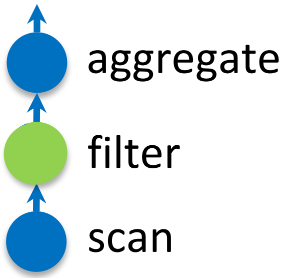
Selective obliviousness

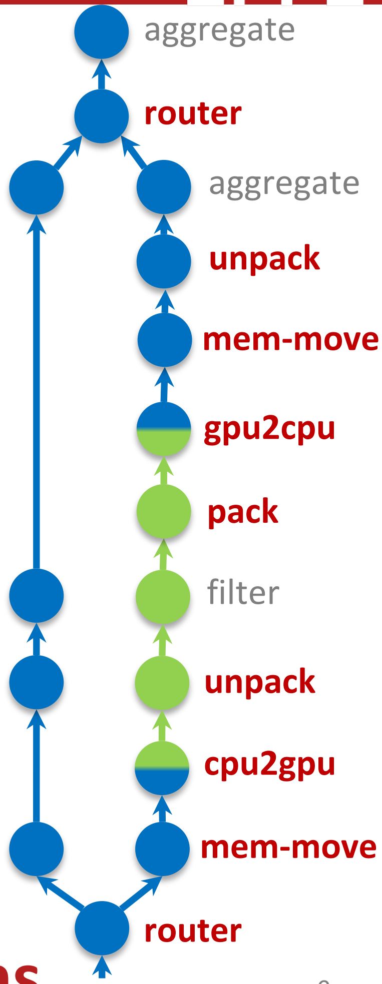
Inter-device: HetExchange

[VLDB2019]

- Decouple data- from control-flow
- Operators encapsulate trait conversions

Flow	Scope	Trait
Control	Delegation	Heterogeneous Parallelism
	Routing	Homogeneous Parallelism
Data	Transfer	Data Locality
	Granularity	Execution Granularity

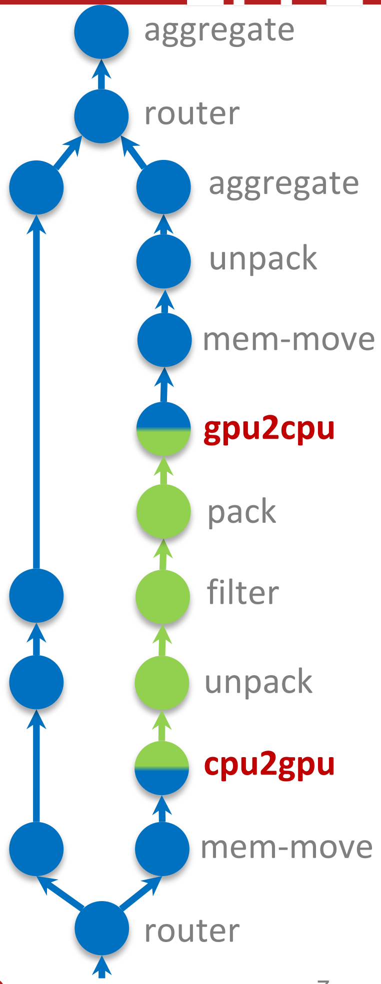




Optimizer can produce cross-device plans

Device Boundary Crossings

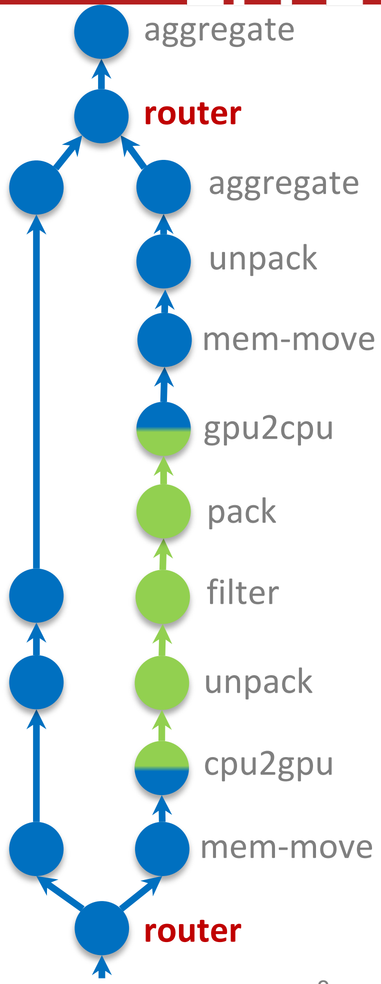
- Cross-device pipelined execution
- Hand-over execution to next device
- Launch kernels/threads, synchronize, backpressure
- Only operators aware of device heterogeneity



Encapsulate heterogeneous parallelism

Concurrent Execution

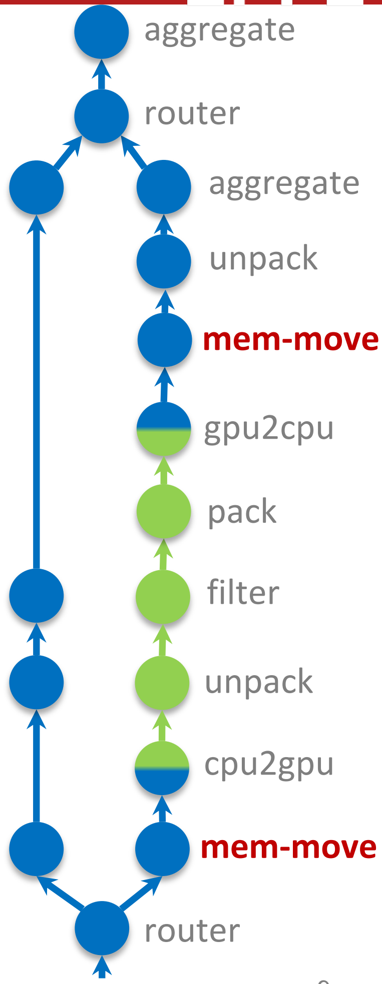
- Horizontal & Vertical parallelism
- Instantiate pipelines multiple times
- Routing policies: load-balance, partition, locality



Encapsulate homogeneous parallelism

Data Transfers

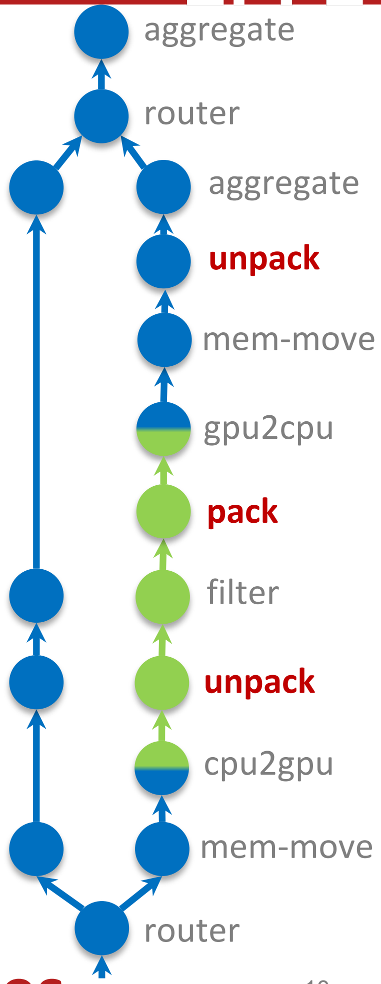
- Handle memory transfers/prefetching
- Hide memory topology
- Overlap transfers with execution



Hide memory heterogeneity

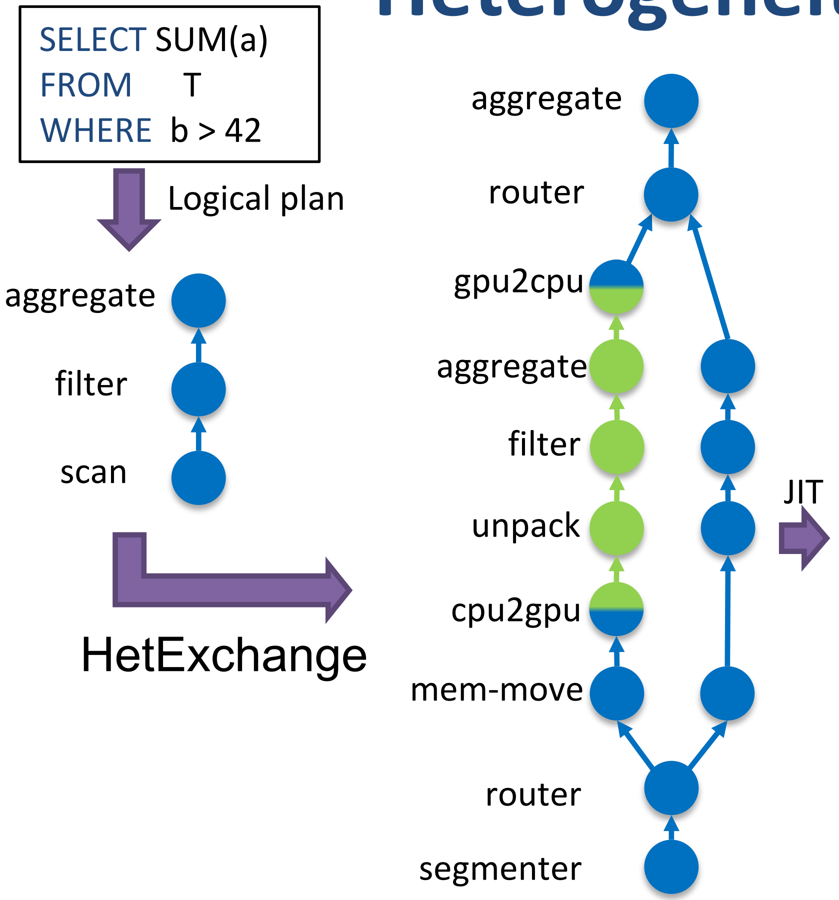
Execution Granularity

- Processing: in-registers => tuple-at-a-time
- Memory transfers: packets => block-at-a-time
- Transition between execution granularities
- Create homogeneous (reg. policy) packets



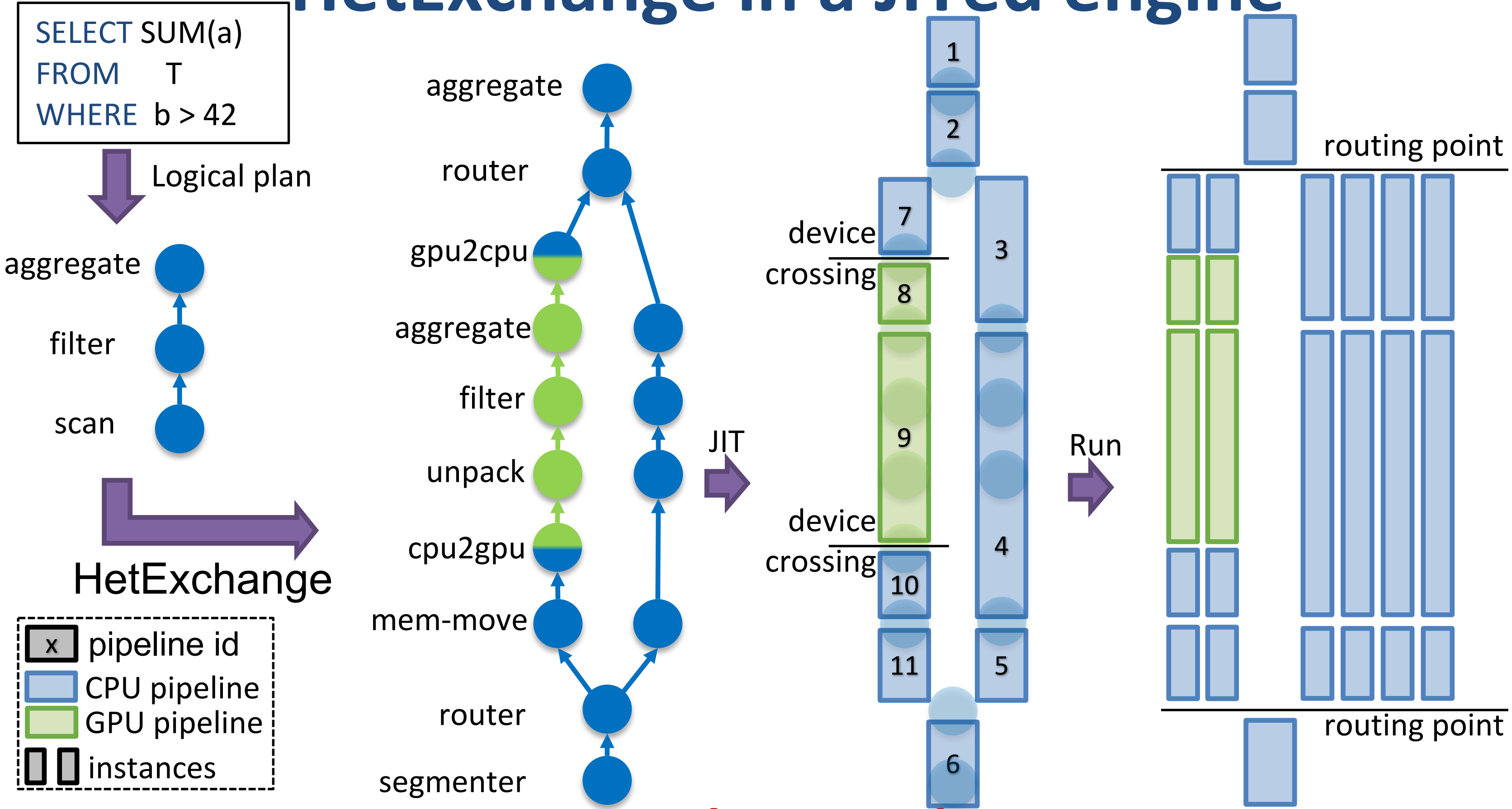
Transition between execution granularities

Heterogeneity-aware plans



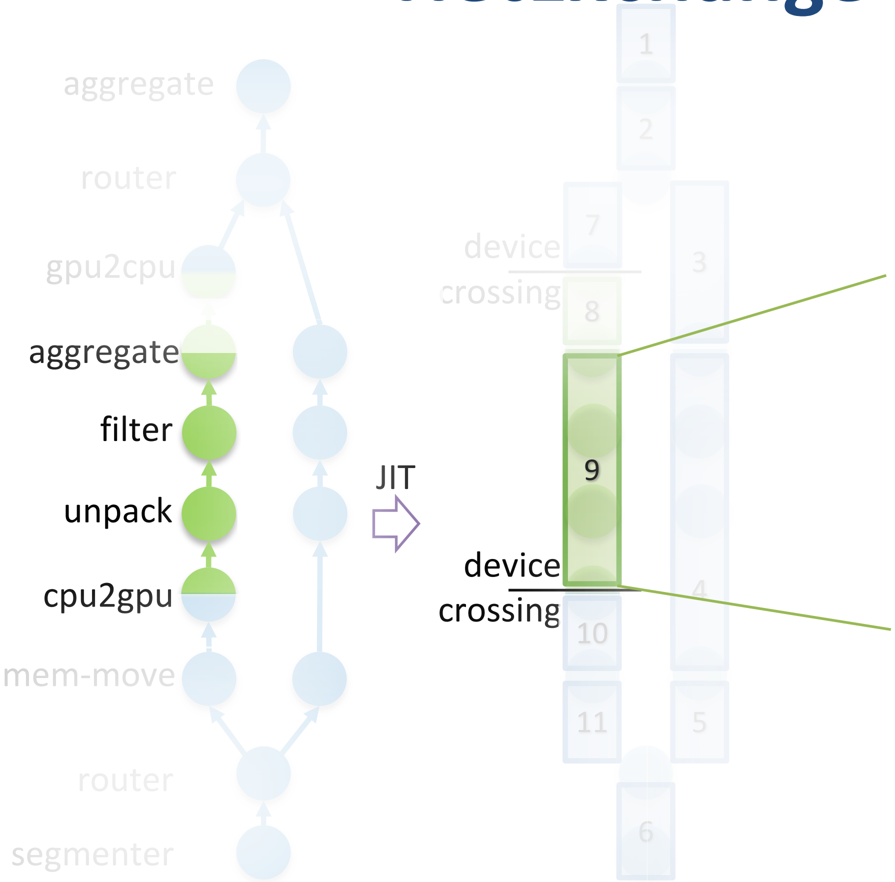
Efficiency
&
Operator portability

HetExchange in a JITed engine



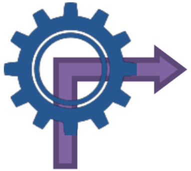
Generate pipelines and instantiate

HetExchange in a JITed engine



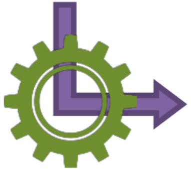
Device providers

CPU Provider



```
def unpack_filter_reduce(data_block, N,
    state)
  local_acc ← 0
  for i = threadIdInWorker to N − 1 with
      step #threadsInWorker
    t ← data_block[i]
    if t.a > 42
      local_acc ← local_acc + t.b
  nh_acc ← neighborhood_reduce(local_acc
  if thread neighborhood leader
    atomic_add(state.acc, nh_acc)
```

GPU Provider



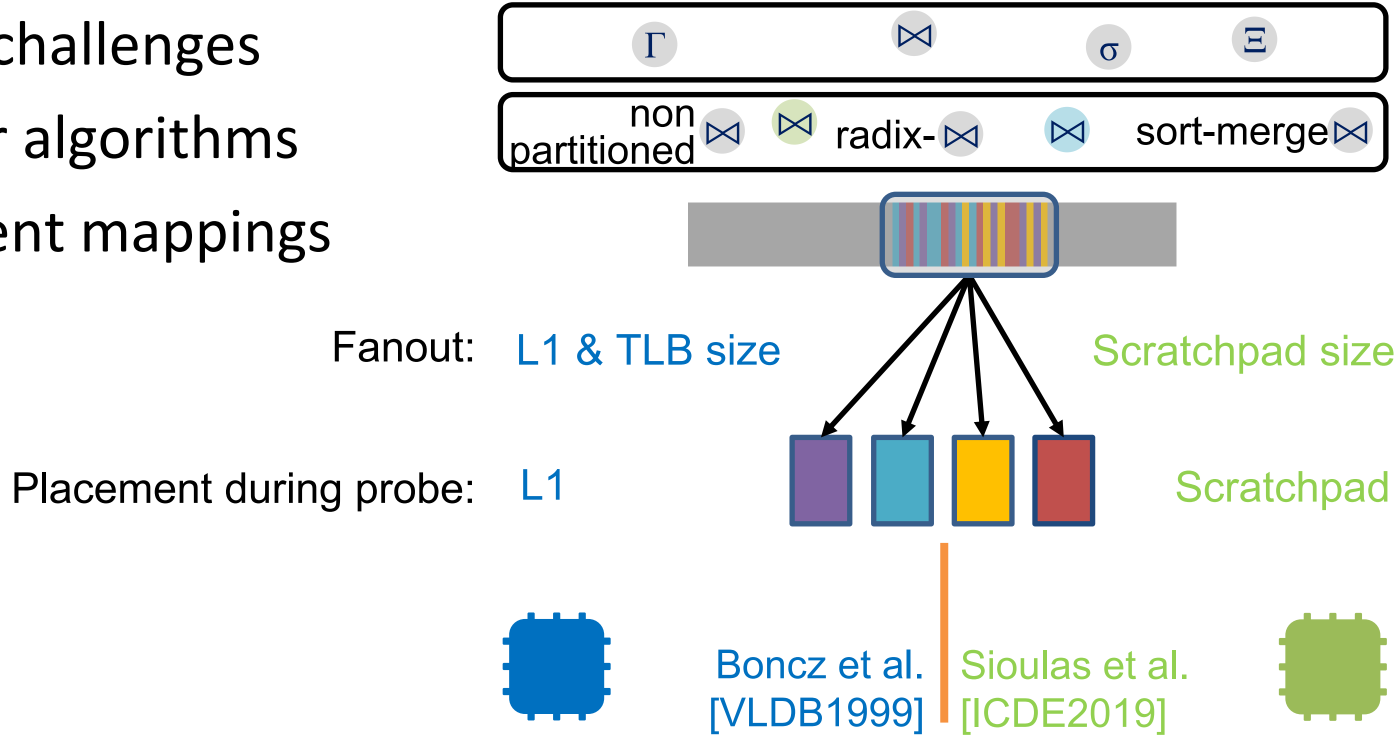
```
function ufr_cpu(data_block, N, state)
  local_acc ← 0
  for i = 0 to N − 1
    t ← data_block[i]
    if t.a > 42
      local_acc ← local_acc + t.b
  nh_acc ← local_acc
  state.acc ← state.acc + nh_acc
```

```
gpu_kernel ufr_gpu(data_block, N, state)
  local_acc ← 0
  for i = threadIdInGrid to N − 1 with
      step gridSize
    t ← data_block[i]
    if t.a > 42
      local_acc ← local_acc + t.b
  nh_acc ← threadblock_reduce(local_acc)
  if threadId = 0
    atomicAdd(state.acc, nh_acc)
```

Inject target-specific info using the JIT infrastructure

Device-optimized operators

- Same challenges
- Similar algorithms
- Different mappings



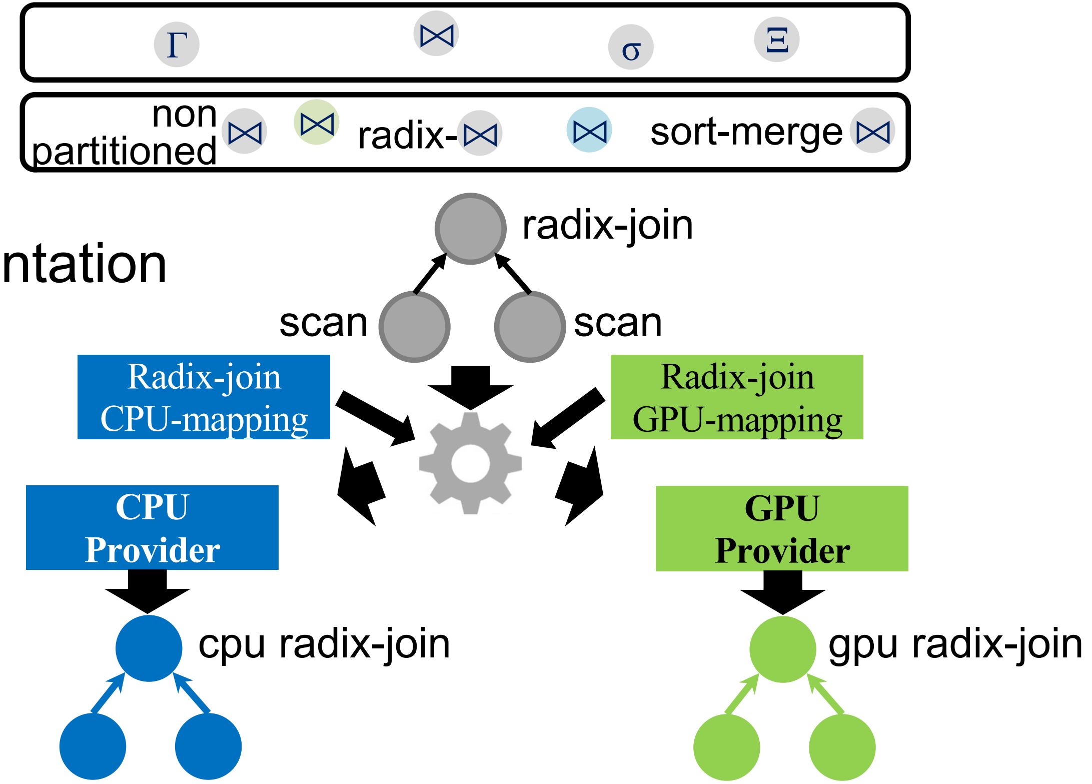
Reuse algorithms, specialize mappings to hardware

Hardware-dependent JIT code

Hardware-aware algorithm

Device-independent implementation

Device-optimized implementation



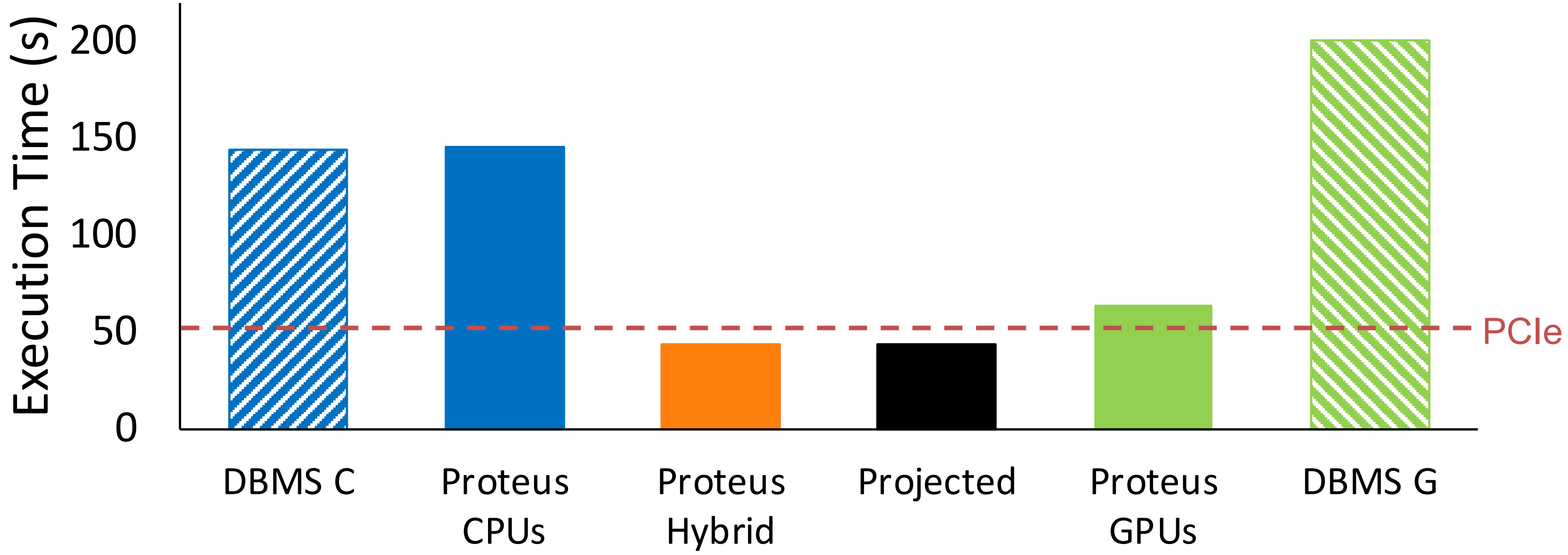
Lower generic description to device-specific code

Experimental Setup

- 2x Intel Xeon E5-2650L v3 12-core @ 1.80GHz, 256GB RAM
- 2x NVIDIA GeForce GTX1080, 8GB, PCIe3 x16 per GPU
- DBMS C/G: state-of-the-art commercial DBMS
  - DBMS C: CPU-based, vector-at-a-time, SIMD, based on MonetDB/X100
  - DBMS G: GPU-based, JIT engine

Performance on CPU-resident data

SSB SF1000, 600GB CSV
working set: 92-138GB / query

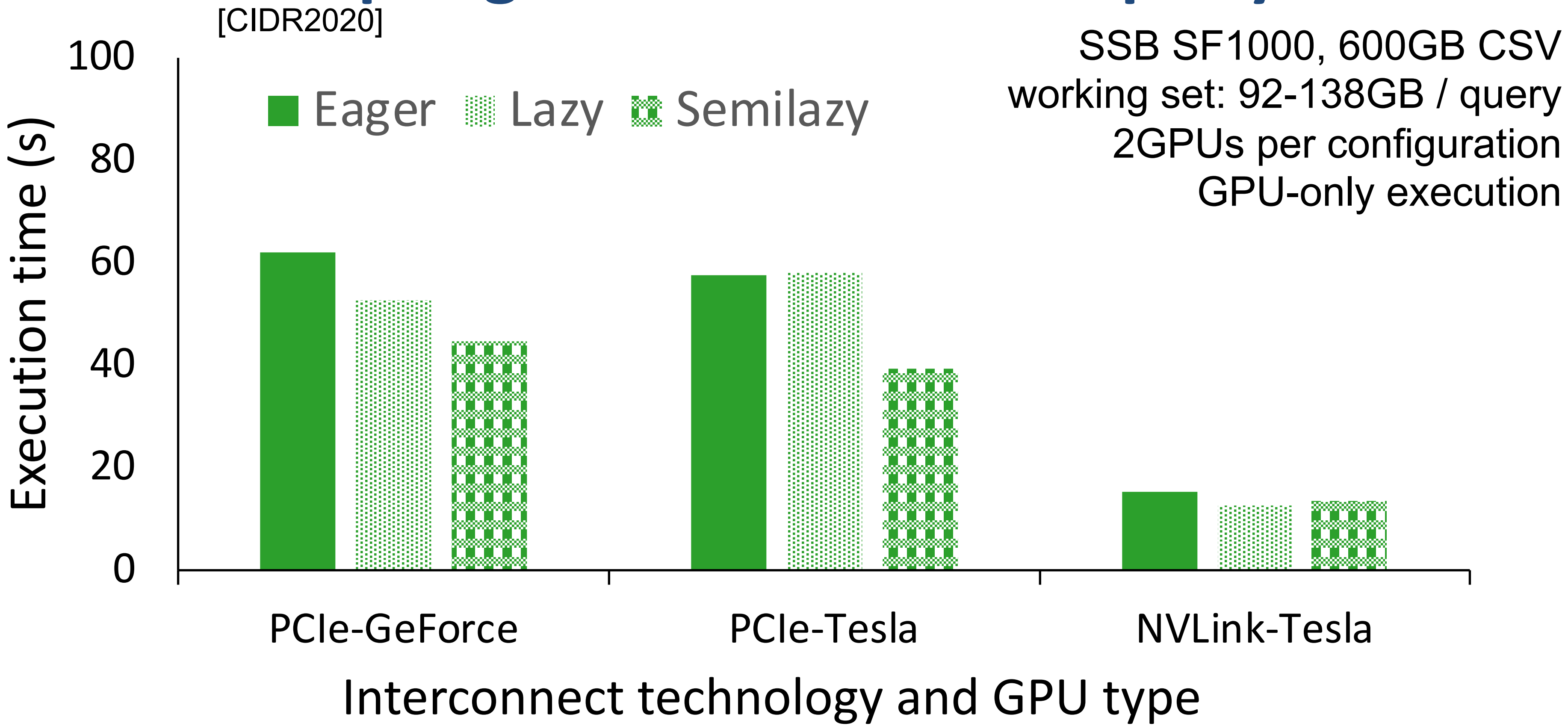


Hybrid throughput = 88.5% (CPU-only + GPU-only), on average

A glimpse into the future

- Effect of interconnects and GPU compute power
- Access to high-throughput network

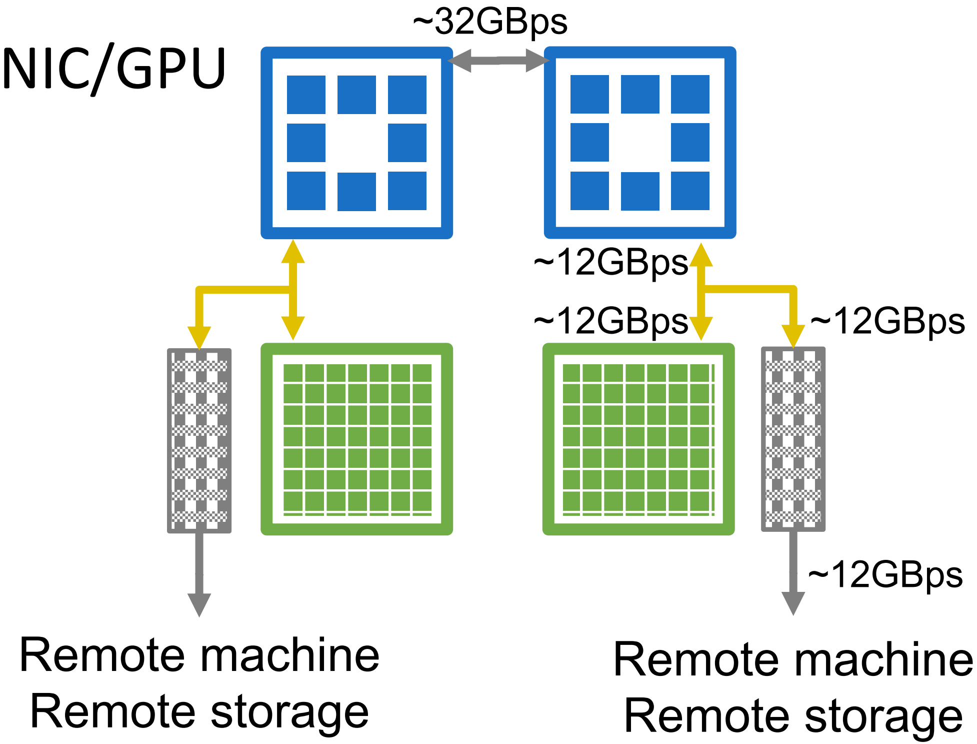
Adapting access method to query



Up to 45% speed-up by tuning access method to hardware

Towards placing the CPU on the side

- Shared & limited PCIe buses to NIC/GPU
- Similar intra/inter-server BW
- Direct NIC-GPU access (RDMA)



Avoid CPU bottleneck => Device-centric OLAP engines

JIT unleashes ALP

- Run on all available devices
- Relational operators oblivious to heterogeneity
- Fast: Inject target-specific information through codegen
- Result: 5x-10x versus CPU-/GPU-specialized systems

Thank you!