Database and
Software Engineering Group
University of Magdeburg

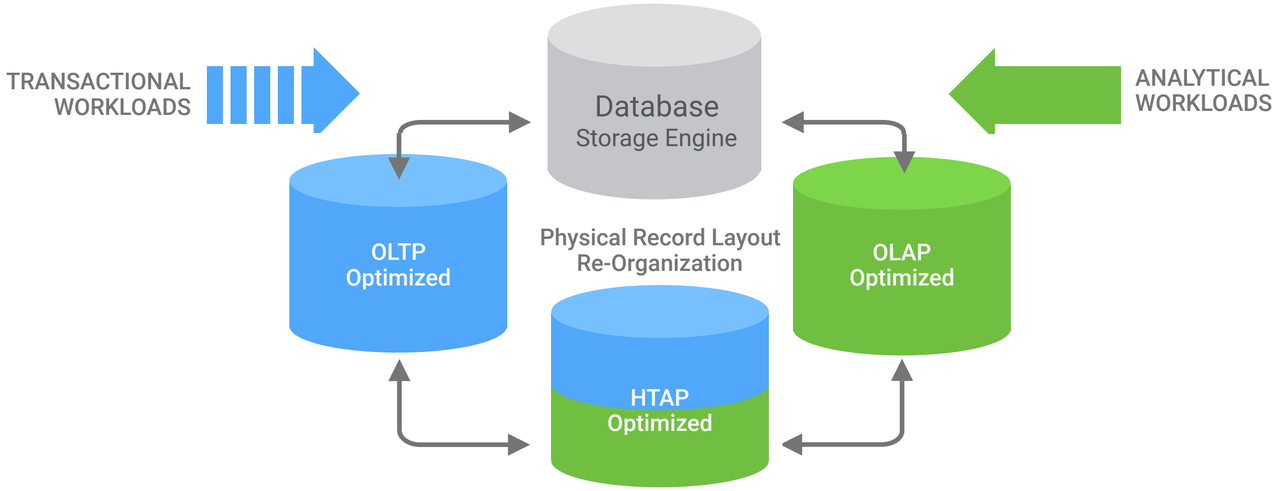
Are Databases Fit for Hybrid Workloads on GPUs?

A Storage Engine's Perspective

Marcus Pinnecke, David Broneske, Gabriel Campero Durand, Gunter Saake

HardBD 2017, San Diego, April 22, 2017

Hybrid Transaction and Analytic Processing (HTAP)



HTAP database systems run both **OLTP & OLAP**

- HyPer, Peloton, HANA, …

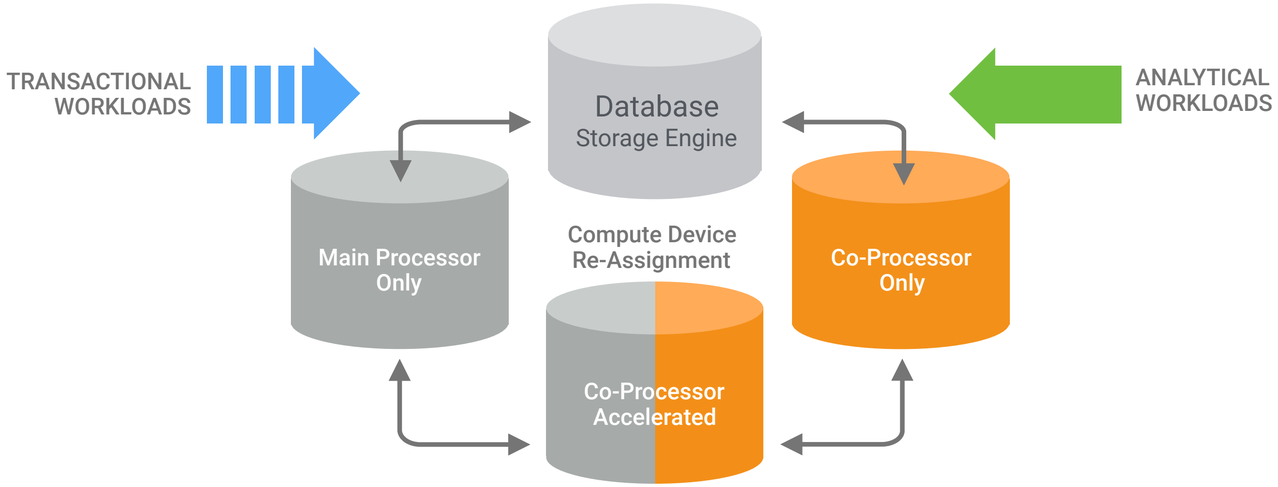
benefit is larger business value, through:

- less latency for analysis
- less synchronization effort

related challenges

- different data access pattern
- adapt record layout (NSM, DSM,…)
- interference between query types
- contradicting optimization goals
- different types of parallelism
- hot and cold data

Database Systems on Heterogenous Platforms



heterogenous systems use **co-processors**

- **host** (CPU), and **device** (e.g., GPU)
- CoGaDB, GPUTx, Ocelot, …

benefit is exploiting compute capacities

- overcome limitations of power wall
- special jobs for specialized processors

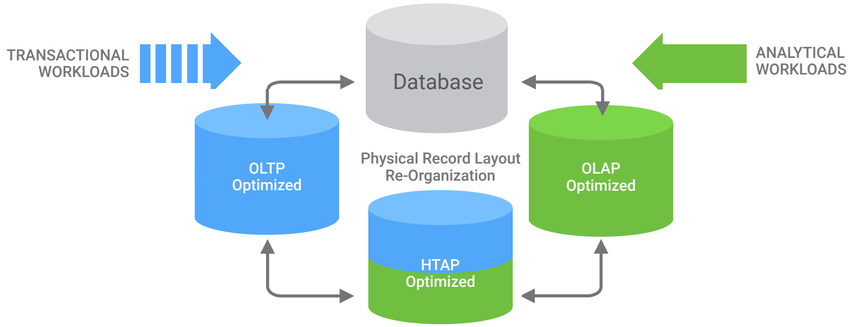
related challenges

- data transfer costs for I/O
- different programming models
- device limitations (e.g., memory capacity)
- data and operator placement

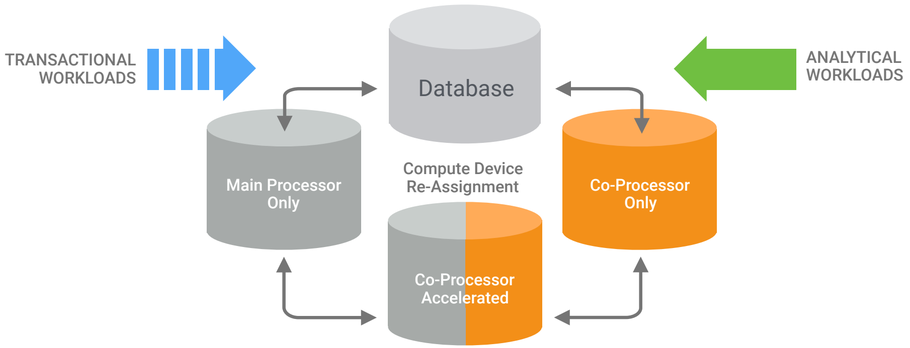
Motivation

Hybridization of HTAP and Heterogenous Computing

First: Is there performance potential?



HTAP Database Systems



Heterogenous Database Systems

TPC-C Benchmark Dataset

measured effort

"OLTP" query materialization

```
select *
from customers
where 150 customers
```

"HTAP" query aggregation of some

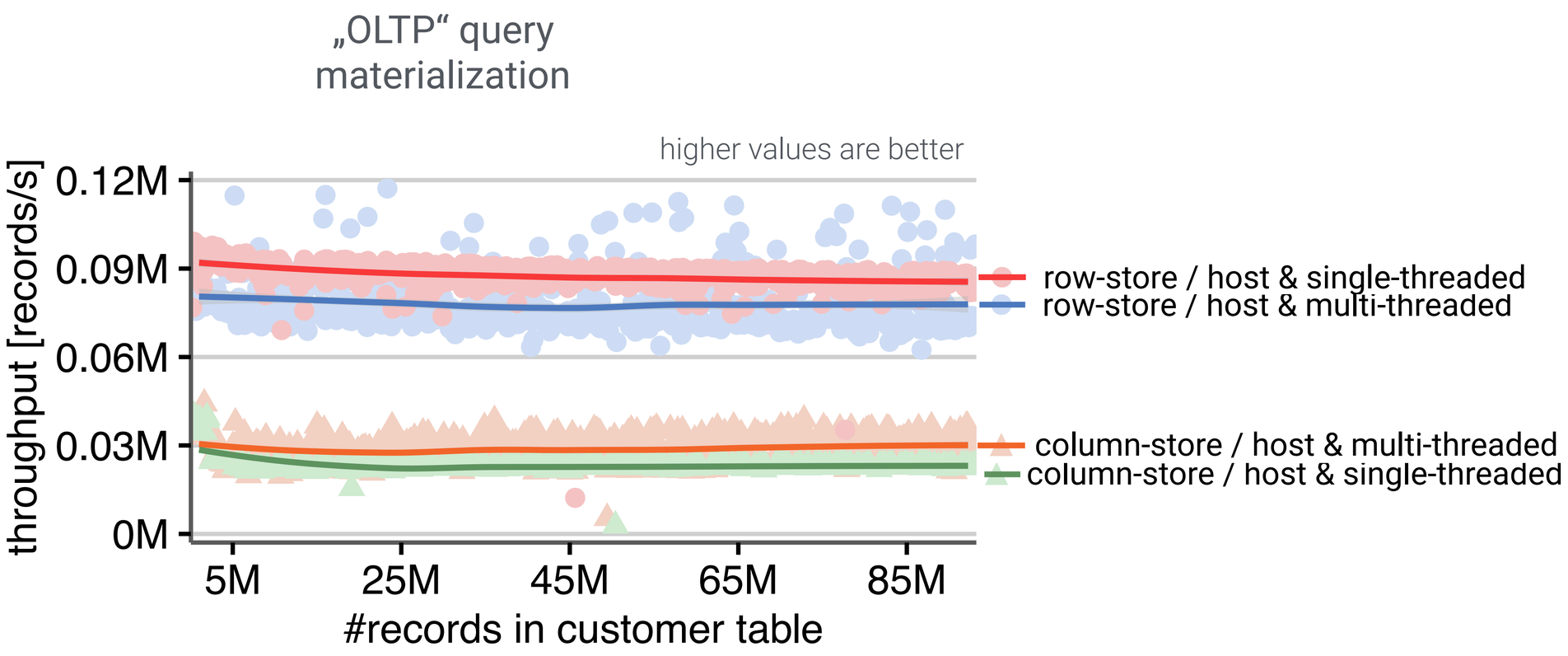
```
select sum(c_bought_item.price)
from customers ⋈ … ⋈ item
where 150 items
```

"OLAP" query aggregation of all

```
select sum(price)
from item
where true
```

Hybridization of HTAP and Heterogenous Computing

First: Is there performance potential?

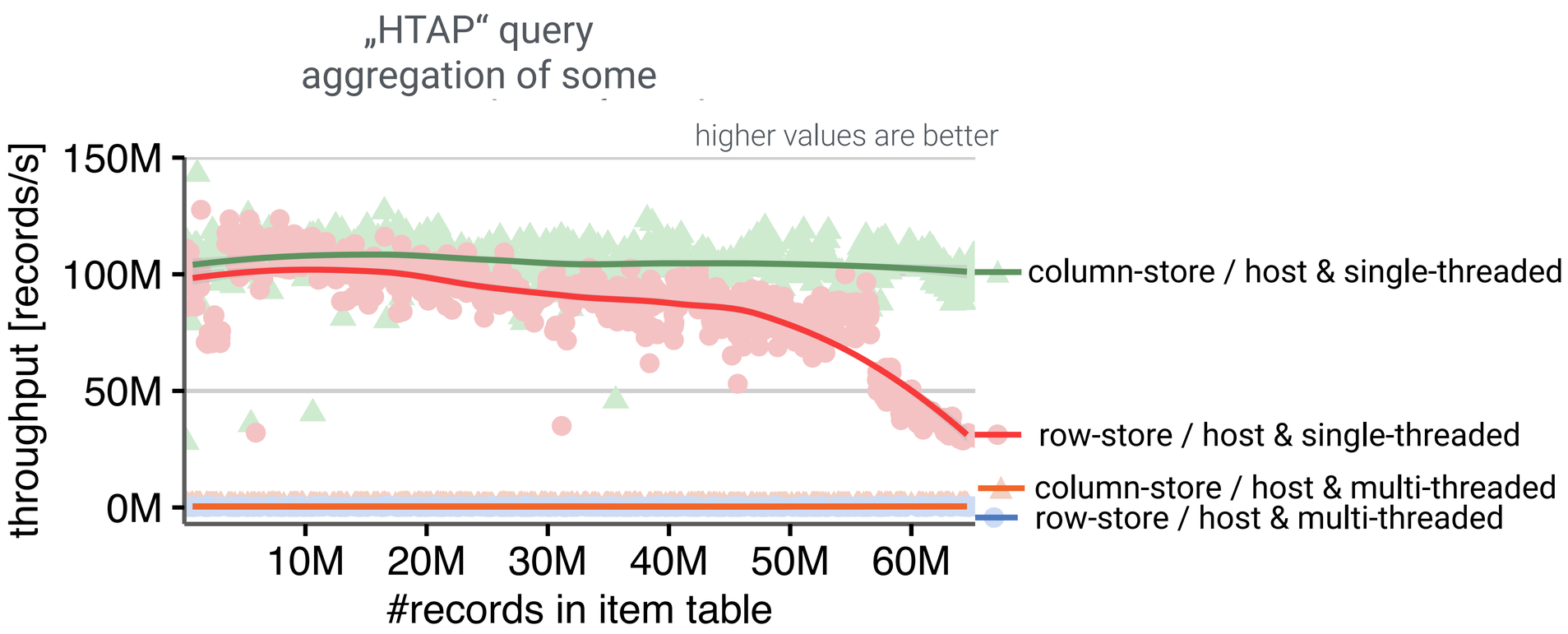


Setup

TPC-C benchmark customer record 96B (21 fields) / item record 20B + 8B (4 fields + price field), **system configuration** operator-at-a-time processing w/ late materialization, host: max. 8 threads blockwise partitioning, device: optimized parallel reduction kernel (>= 1024 blocks w/ 512 threads), final reduction on 1 block w/ 1024 threads, effort for join processing not incl.

Hybridization of HTAP and Heterogenous Computing

First: Is there performance potential?

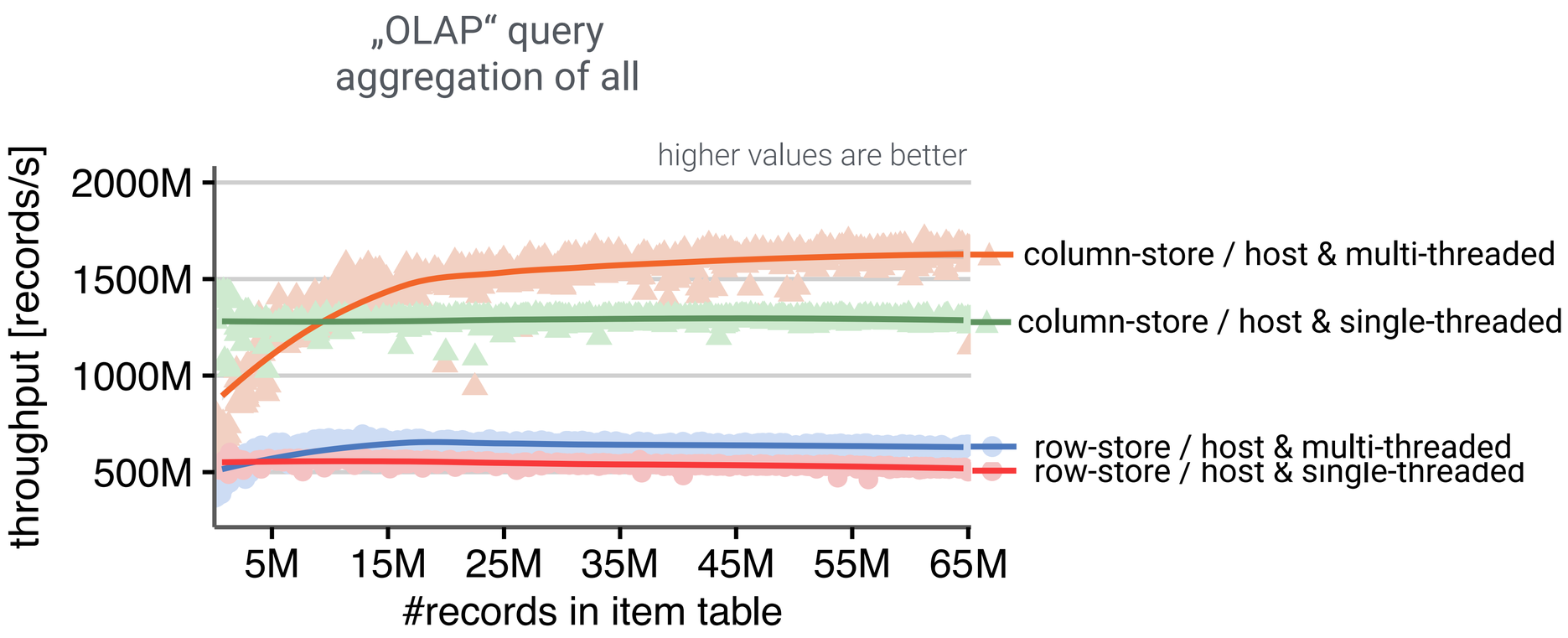


Setup

TPC-C benchmark customer record 96B (21 fields) / item record 20B + 8B (4 fields + price field), **system configuration** operator-at-a-time processing w/ late materialization, host: max. 8 threads blockwise partitioning, device: optimized parallel reduction kernel (>= 1024 blocks w/ 512 threads), final reduction on 1 block w/ 1024 threads, effort for join processing not incl.

Hybridization of HTAP and Heterogenous Computing

First: Is there performance potential?

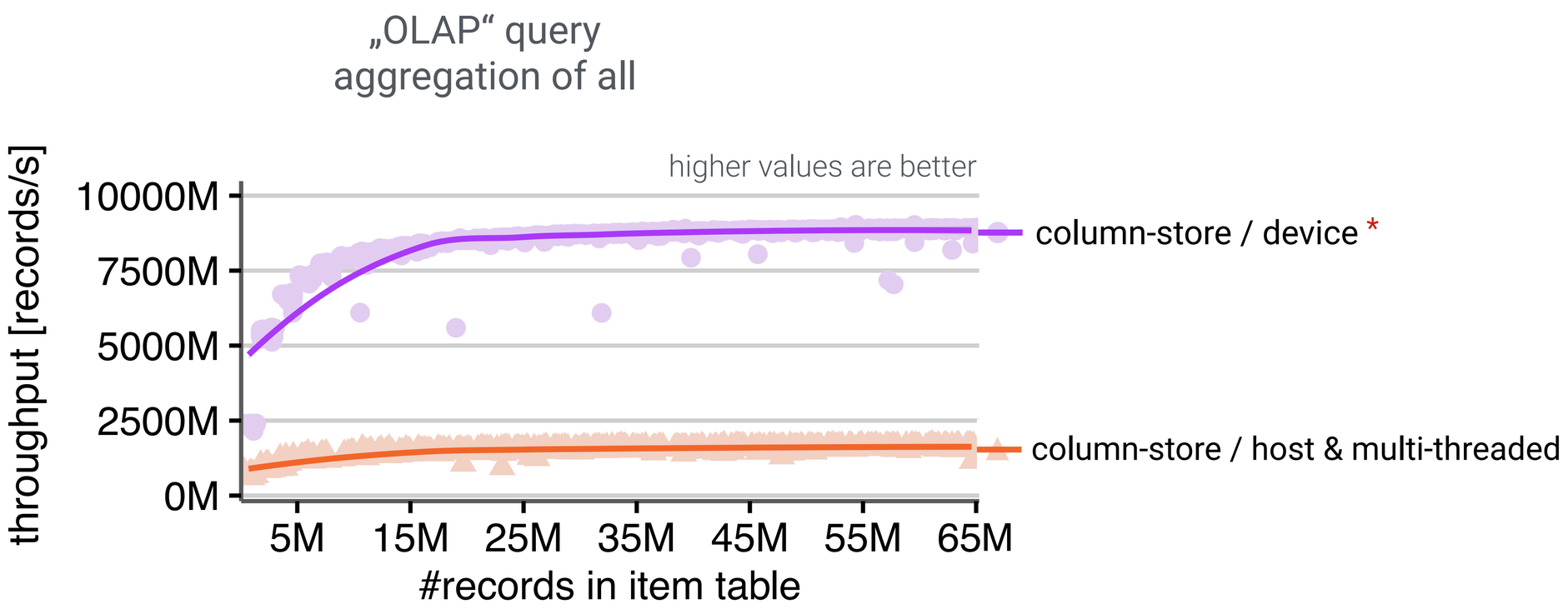


Setup

TPC-C benchmark customer record 96B (21 fields) / item record 20B + 8B (4 fields + price field), **system configuration** operator-at-a-time processing w/ late materialization, host: max. 8 threads blockwise partitioning, device: optimized parallel reduction kernel (>= 1024 blocks w/ 512 threads), final reduction on 1 block w/ 1024 threads, effort for join processing not incl.

Hybridization of HTAP and Heterogenous Computing

First: Is there performance potential?



*Device transfer costs excluded

Setup

TPC-C benchmark customer record 96B (21 fields) / item record 20B + 8B (4 fields + price field), **system configuration** operator-at-a-time processing w/ late materialization, host: max. 8 threads blockwise partitioning, device: optimized parallel reduction kernel (>= 1024 blocks w/ 512 threads), final reduction on 1 block w/ 1024 threads, effort for join processing not incl.

Hybridization of HTAP and Heterogenous Computing

First: Is there performance potential?

„OLTP“ query materialization	„HTAP“ query aggregation of some	„OLAP“ query aggregation of all
150 customers	150 items	all items
row-store / host & single-threaded	column-store / host & single-threaded	column-store / device column-store / host & multi-threaded

Storage Engine

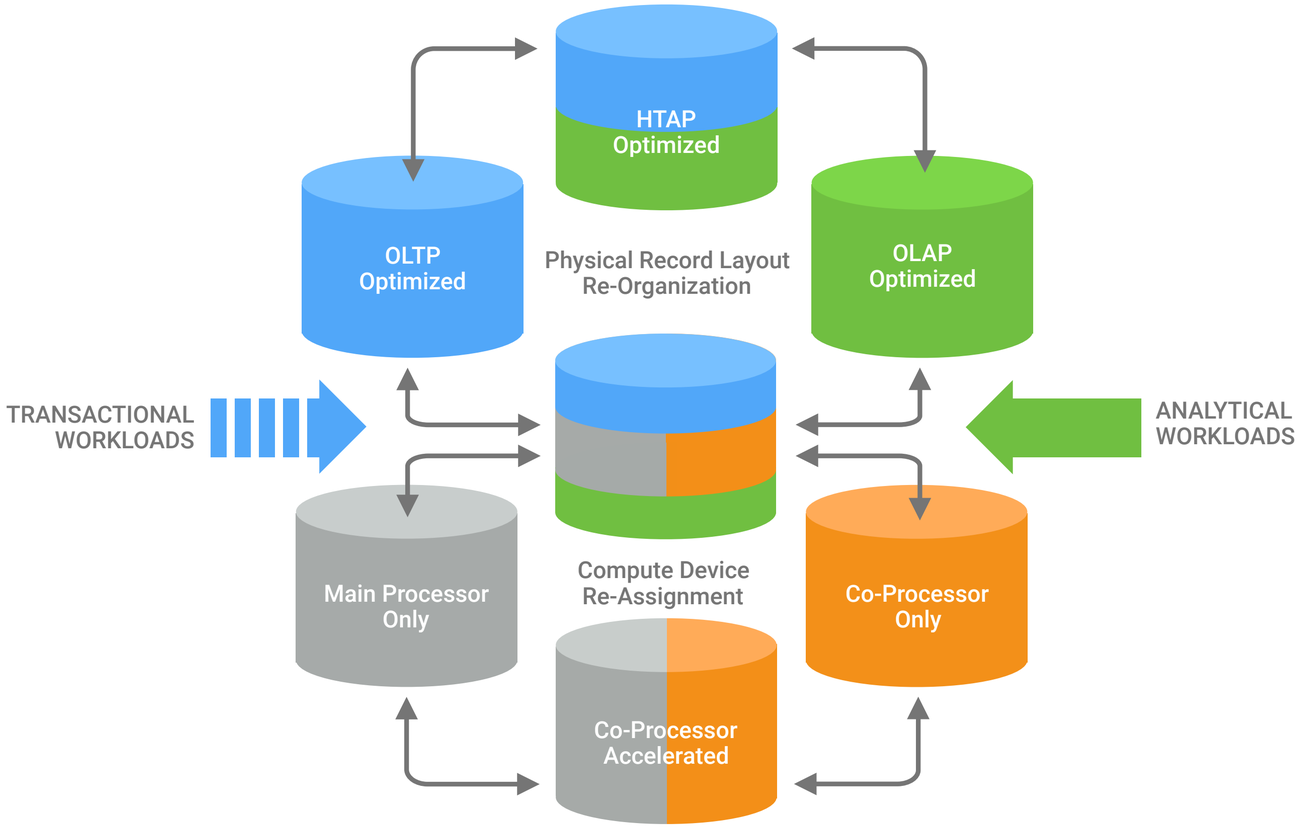
Row-Store ←*change*→ Column-Store — Column-Store

Host ←*switch*→ Device

transition
no transition

Query Engine

Inter-query parallelism — Inter-query parallelism ←*switch*→ Intra-query parallelism



To take advantage, we need to double hybridize to have the best of both worlds.

Contribution

Survey and Classification of SOTA Engines

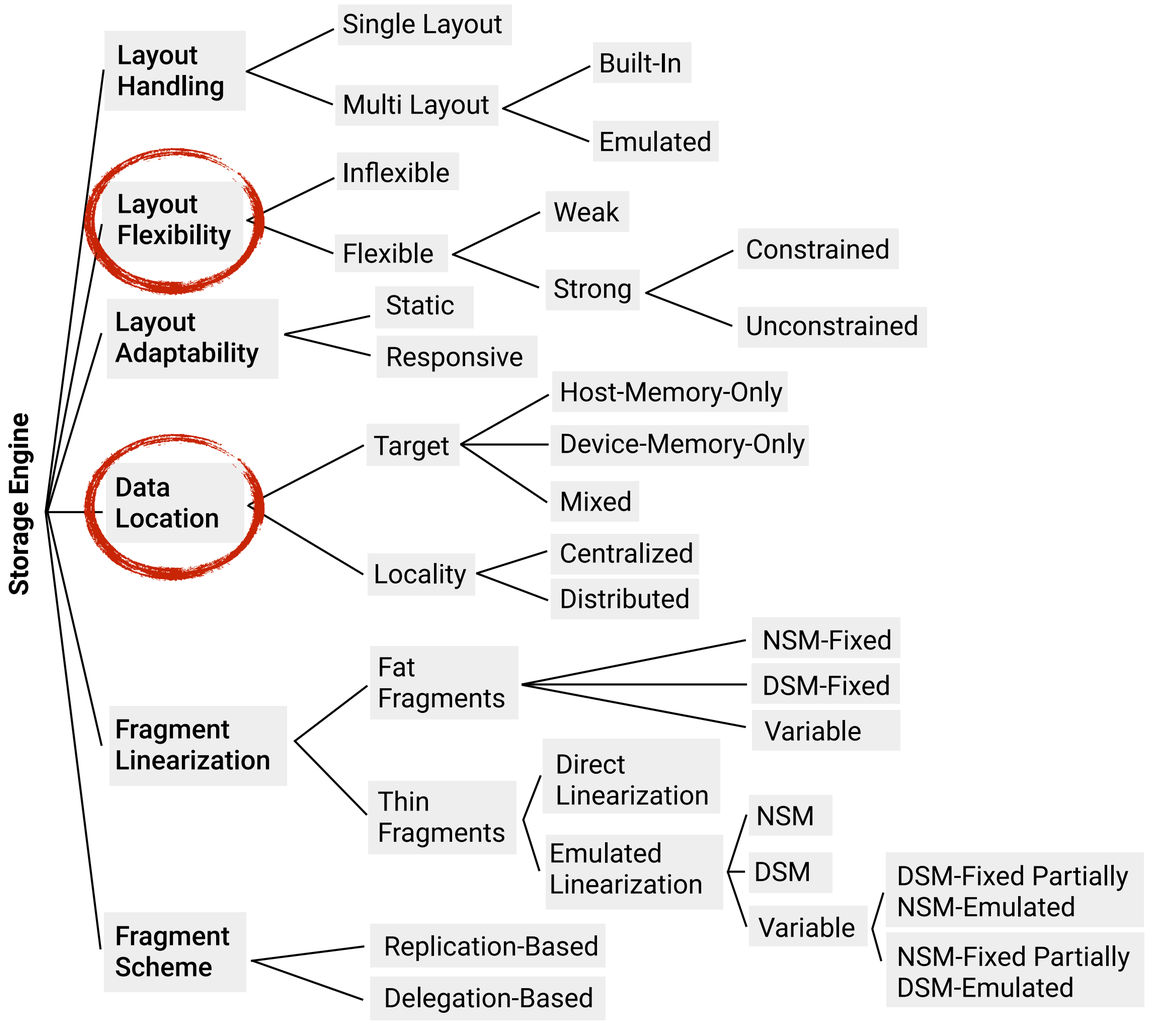
Taxonomy

We defined a **fine-grained set of concepts** and **relations between them** in order to classify systems

- Unified terms to compare different systems
- Overview on design space
- Possibility to assess adaptability w.r.t. transitions

Core concepts

- **Layout.** Division of a relation in terms of fragments.
- **Fragment**. „Gapless“ region of data.
- **Tuplet**. Portion of tuple that fell into a fragment.



Survey and Classification of SOTA Engines

Classification

HTAP systems

- OLAP or OLTP
- OLAP and OLTP
- co-processor support
- multiple layout support
- advanced flexibility for layout
- adaptive layouts during runtime
- mixed NSM/DSM support

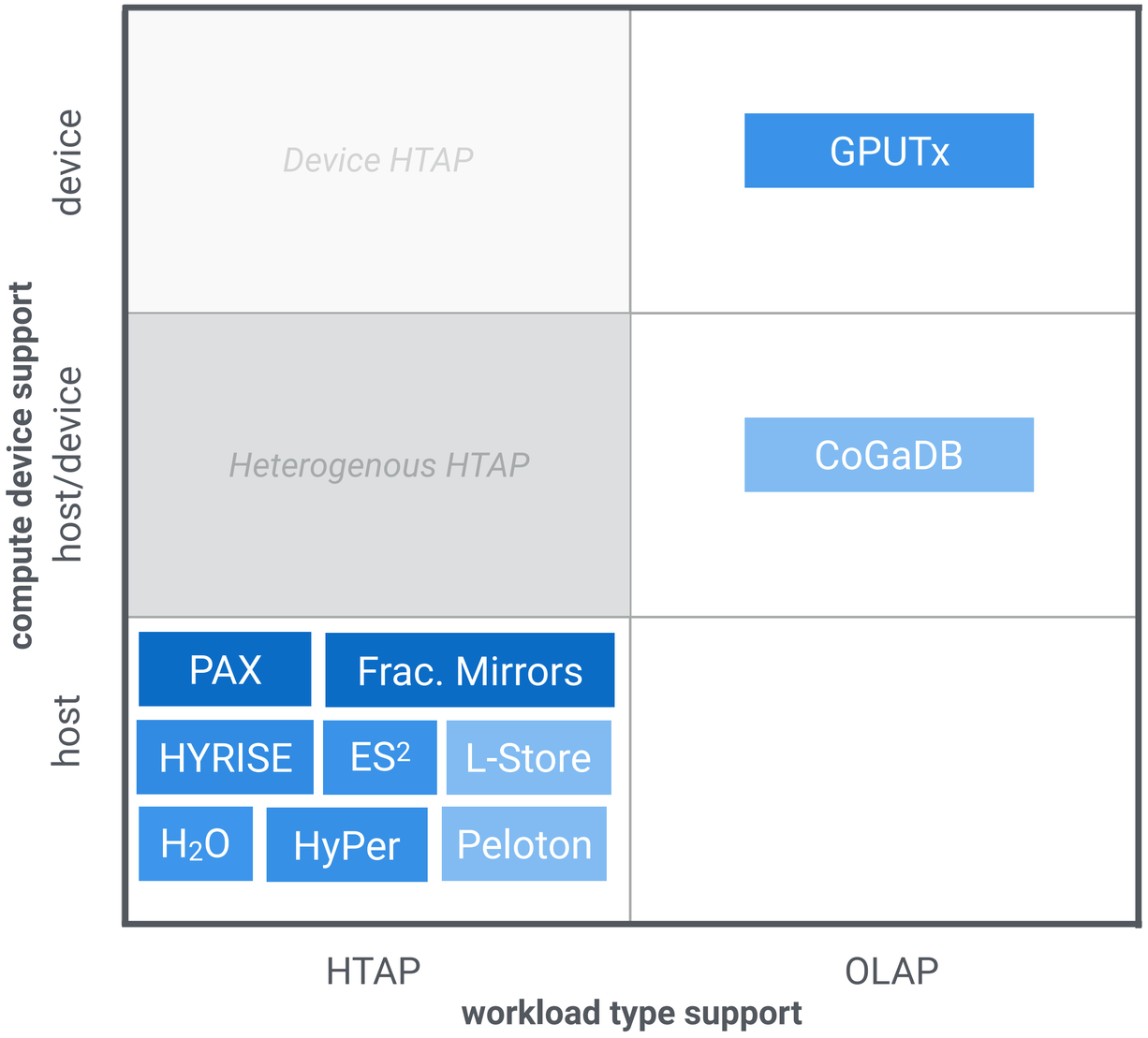
Co-processor systems

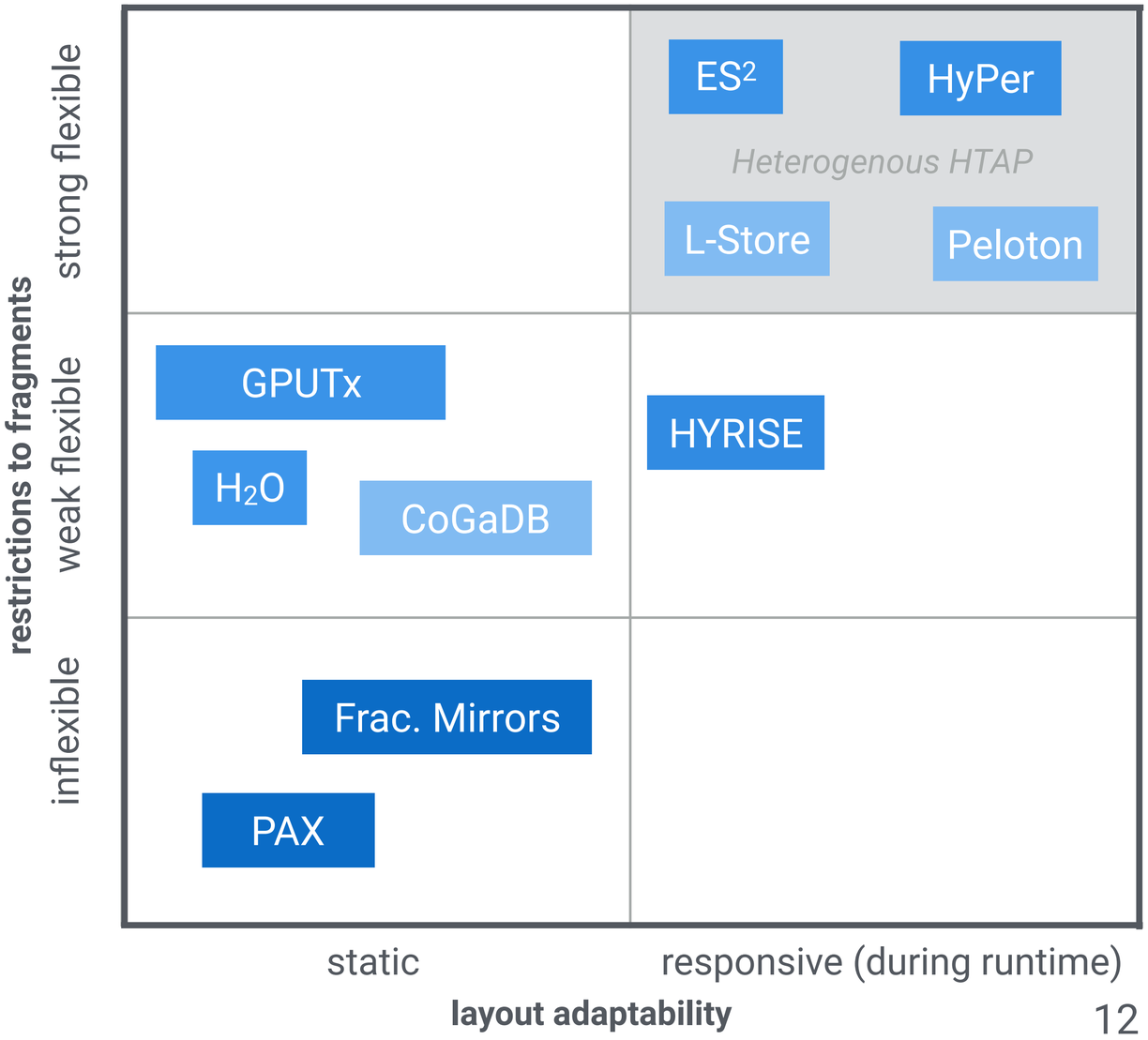
- OLAP or OLTP
- OLAP and OLTP
- co-processor support
- multiple layout-support
- advanced flexibility for layout
- adaptive layouts during runtime
- mixed NSM/DSM support

Survey and Classification of SOTA Engines

View on Results

2002 year 2016





Future Work

MondrianDB

Self-managing multi-model HTAP database system on CPU/GPUs

Codename Vector Pipes
Query Engine

- Micro-batch query execution
- UDFs as first-class citizen
- Multi query execution

Codename Grid Store
Storage Engine

- Adaptive HTAP storage
- Heterogenous platforms
- Advanced-Tile based

Codename Alfred
Optimizer

- Self-managing component
- Learning on event streams

X100 PIPES MapReduce ...

Peloton L-Store CoGaDB ...

Grid Store: A Storage Engine for Heterogenous HTAP

Wrap Up and Outlook. Feedback is welcome

Challenges

HTAP

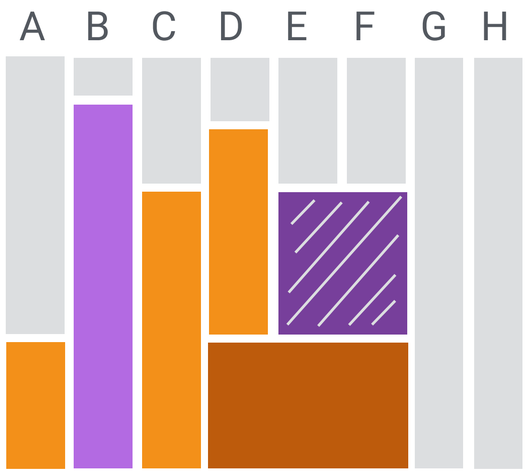
1. different data access pattern
2. adapt record layout (NSM, DSM,…)
   - interference between query types
   - contradicting optimization goals
   - different types of parallelism
3. hot and cold data parts

Heter.

4. data transfer costs for I/O
5. different programming models
6. device limitations (e.g., memory capacity)
7. data and operator placement

Grid Store: A Storage Engine for Heterogenous HTAP

Wrap Up and Outlook. Feedback is welcome



Example table layout in Grid Store
(*unconstrained* strong flexible responsive layout w/ data placement)

- cold data (compressed)
- hot data, stored on **device** (DSM)
- hot data, stored on **device** (NSM)
- hot data, stored on **host** (DSM)
- hot data, stored on **host** (NSM)
- reasonability currently under study

Built tables from *arbitrary tiles* (**grids**) which…

- … may live on host or device (or both) 4 7
- … are self-contained (NSM/DSM/Indexed/…) 1 3 5
- … mutable w.r.t. to access pattern and shape 2 7

Table layouts **respond online to forecasted workload changes**

- shrink or enlarge grids, compress/uncompress 3 6
- move grids (temporary, permanent) to platform 4

Backed by **flexible query engine** (Vector Pipes) and **adaptive self-leaning optimizer on real-time analytics of system event stream** (Alfred)

Grid Store: A Storage Engine for Heterogenous HTAP

Wrap Up and Outlook. Feedback is welcome

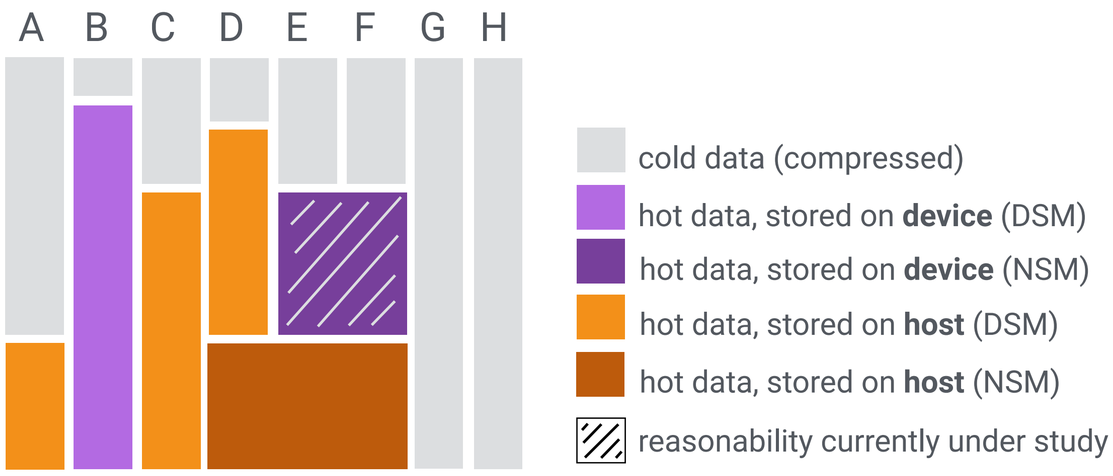
Challenges

HTAP

1. different data access pattern
2. adapt record layout (NSM, DSM,…)
   - interference between query types
   - contradicting optimization goals
   - different types of parallelism
3. hot and cold data parts

Heter.

4. data transfer costs for I/O
5. different programming models
6. device limitations (e.g., memory capacity)
7. data and operator placement



Example table layout in Grid Store
(*unconstrained* strong flexible responsive layout w/ data placement)

Built tables by *arbitrary tiles* (**grids**) which…

- … may live on host or device (or both) 4 7
- … are self-contained (NSM/DSM/Indexed/…) 1 3 5
- … mutable w.r.t. to access pattern and shape 2 7

Table layouts **respond online to forecasted workload changes**

- shrink or enlarge grids, compress/uncompress 3 6
- move grids (temporary, permanent) to platform 4

Backed by **flexible query engine** (Vector Pipes) and **adaptive self-leaning optimizer on real-time analytics of system event stream** (Alfred)