

The background is a deep blue gradient. On the left side, there are several lines of binary code (0s and 1s) in a light blue, slightly glowing font, angled upwards from left to right. On the right side, there are several bright, white, curved light streaks that sweep across the frame, creating a sense of motion and energy. A bright, circular light flare is visible in the center-right area, where the streaks seem to converge or originate from.

SQL+GPU+SSD= ∞

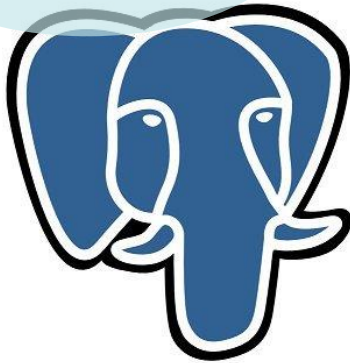
Wasserschwein@Shinagawa

Self Introduction



- Name: Wasserschwein@Shinagawa
- PostgreSQL: 9Years (2006~)
- Works: Security, FDW, etc...
- Hobby: Mixture of heterogeneous technology with PostgreSQL

Very functional
& well-used
database



PostgreSQL

PG-Strom:

What I'm making

***GP*GPU**



Very powerful
computing
capability

What's PG-Strom – Brief overview

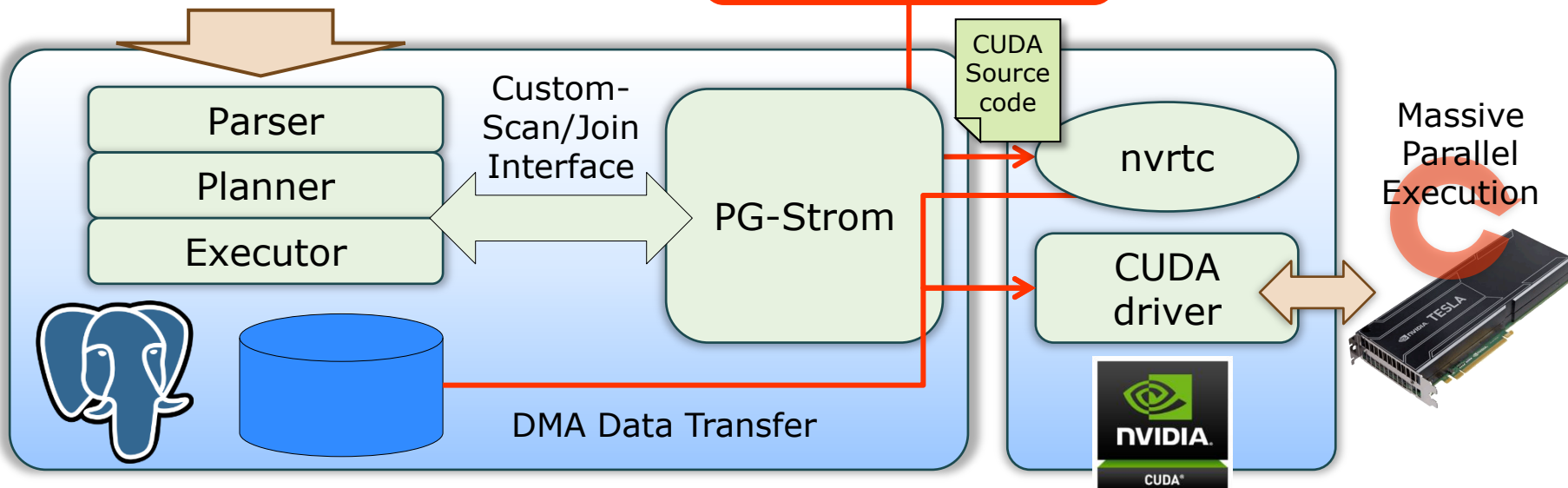
Core ideas

- ① GPU native code generation on the fly
- ② Asynchronous massive parallel execution

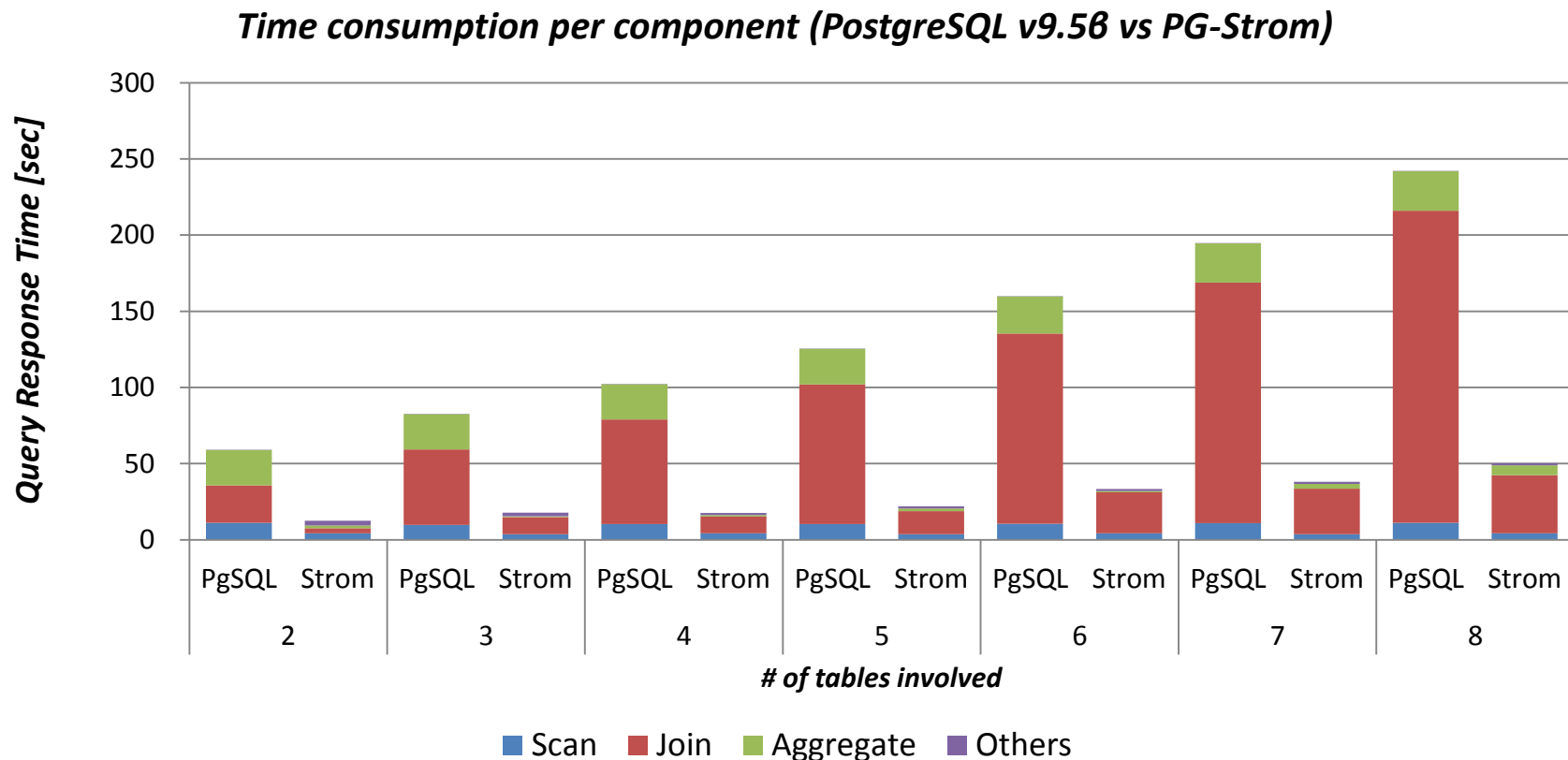
Advantages

- Transparent acceleration with 100% query compatibility
- Commodity H/W and less system integration cost

```
Query: SELECT * FROM l_tbl JOIN r_tbl on l_tbl.lid = r_tbl.rid;
```



Supported Workload – Scan, Join, Aggregation



SELECT cat, AVG(x) FROM t0 NATURAL JOIN t1 [, ...] GROUP BY cat;

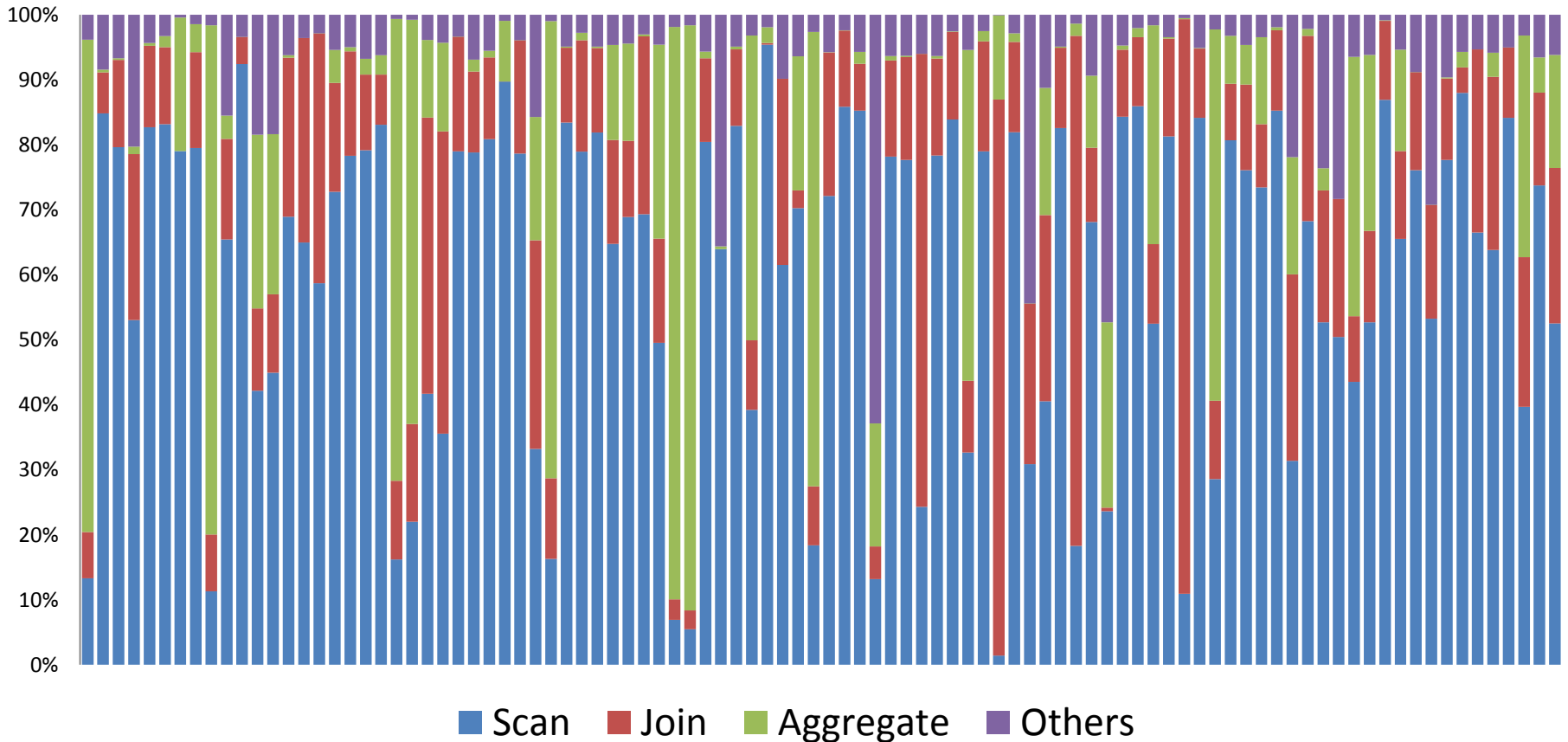
- t0: 100M rows, t1~t10: 100K rows for each, all the data was preloaded.

Environment:

- PostgreSQL v9.5beta1 + PG-Strom (22-Oct), CUDA 7.0 + RHEL6.6 (x86_64)
- CPU: Xeon E5-2670v3, RAM: 384GB, GPU: NVIDIA TESLA K20c (2496cores)

Next target is I/O acceleration – from TPC/DS results

Time consumption per workloads (PostgreSQL v9.5beta+PG-Strom)



So, How to accelerate I/O stuff by GPU?

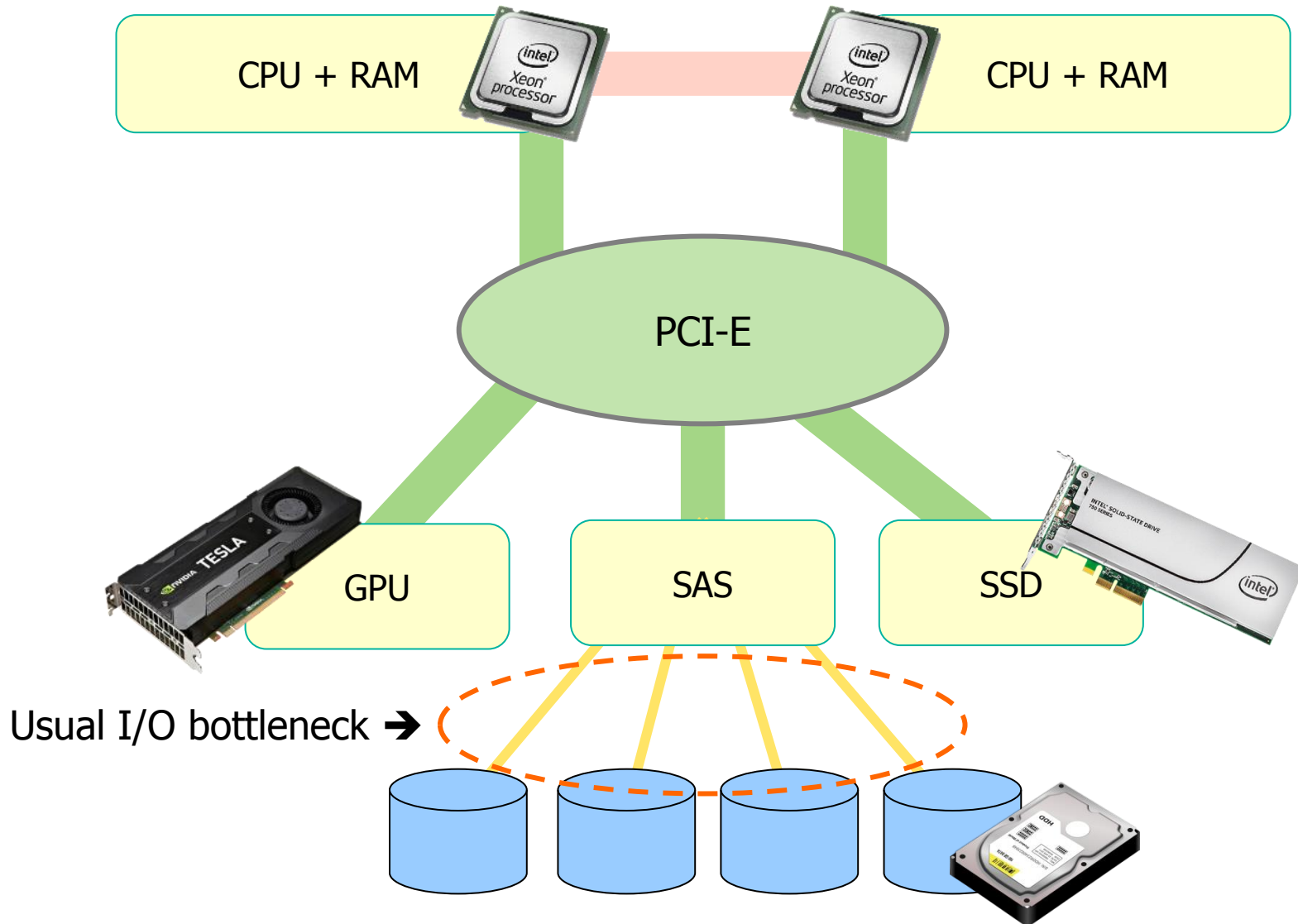
NOTICE

The story I like to introduce next is...

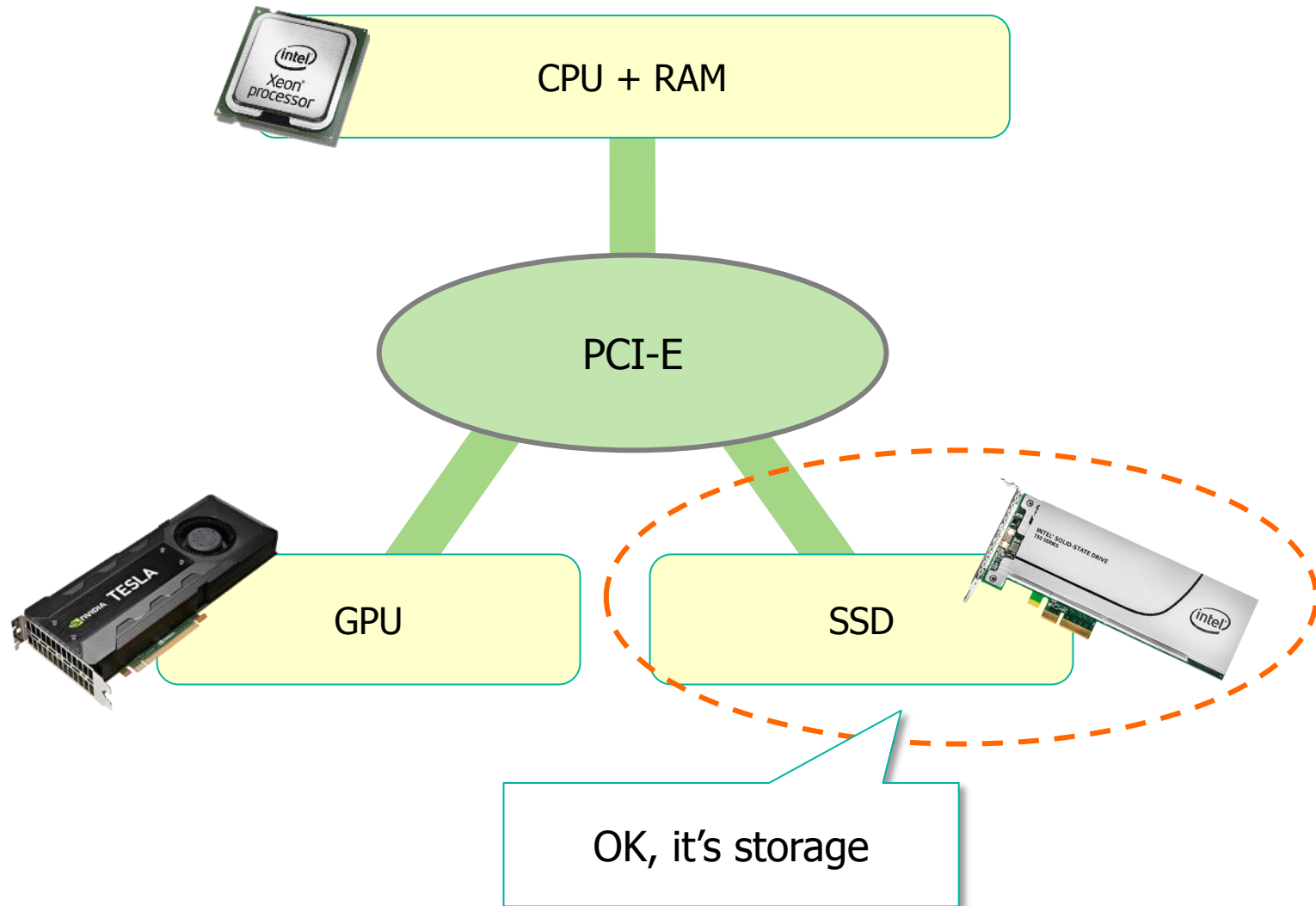
Just my Idea at this moment

.....So, I'll pay my efforts to implement

A rough x86_64 hardware architecture



Simplified diagram for introduction



NVM EXPRESS SSD

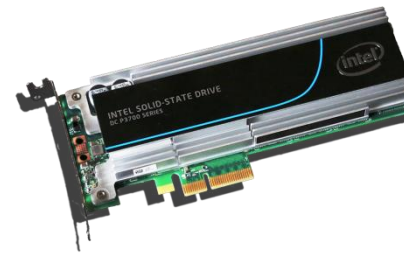
PCI-E direct SSD device – low latency and higher bandwidth



HGST
Ultrastar SN100



Intel
SSD DC P3700



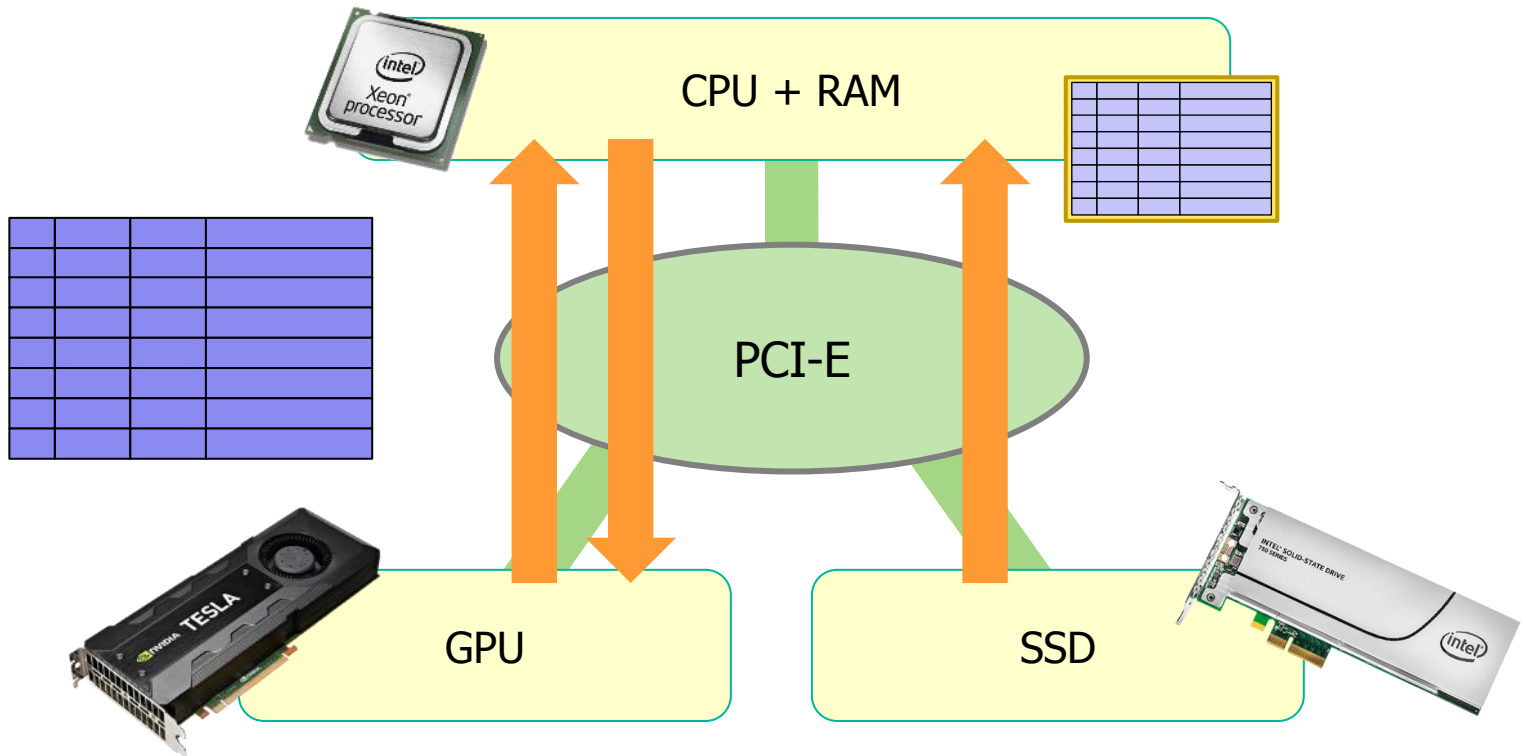
Intel SSD 750



Samsung
SSD 950 PRO



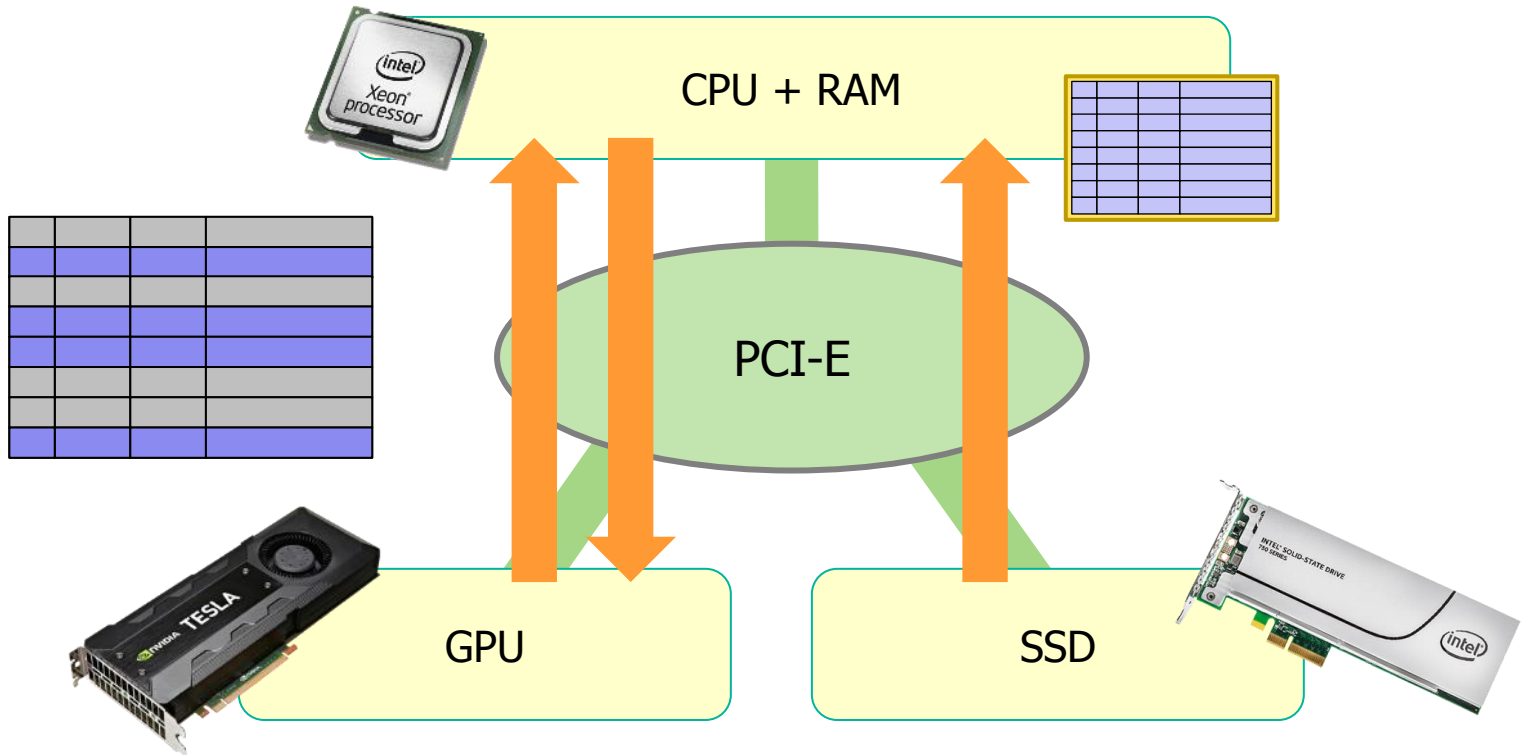
Data Flow in analytic queries



① Data load from storage to CPU/RAM

Table			

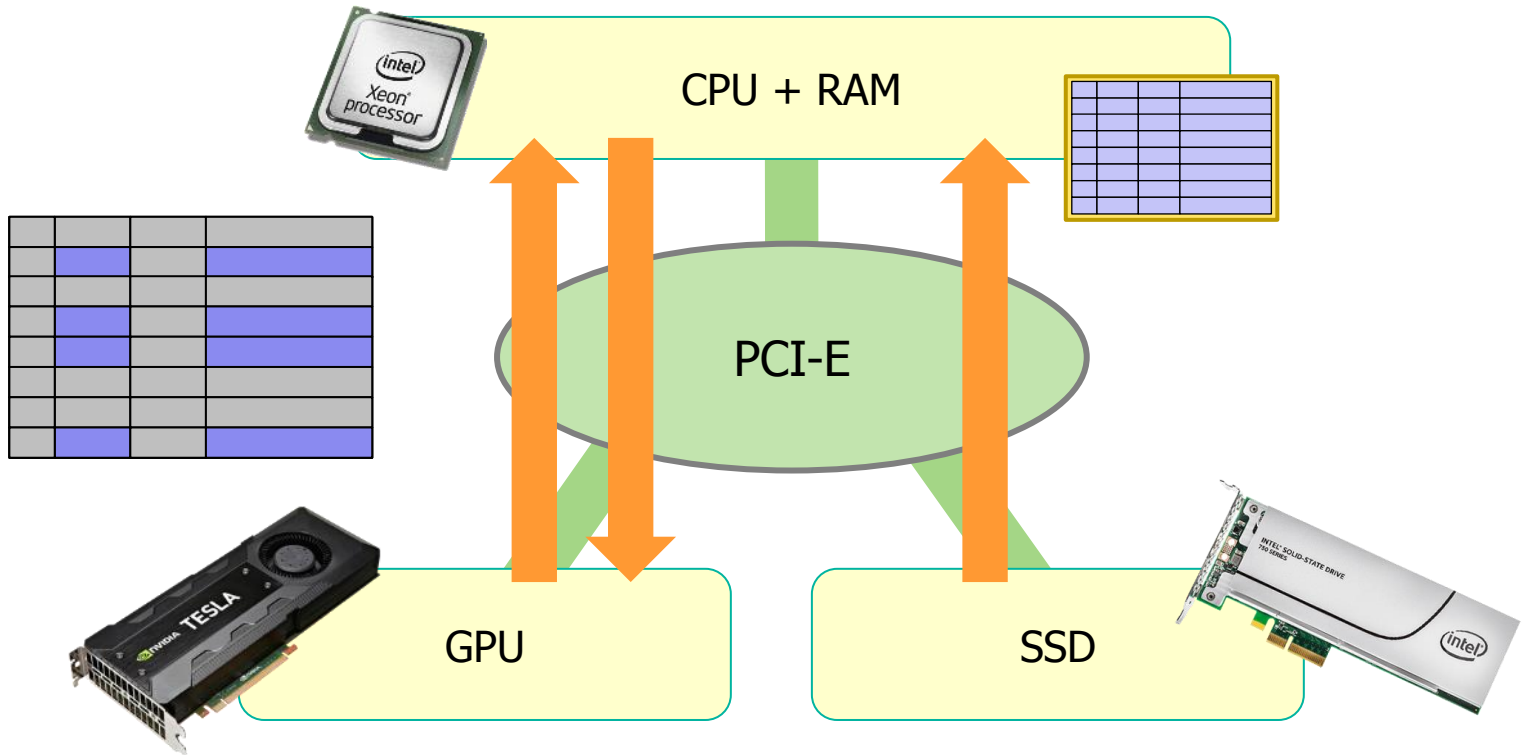
Data Flow in analytic queries



- ① Data load from storage to CPU/RAM
- ② Remove invisible rows (Select)

Table			

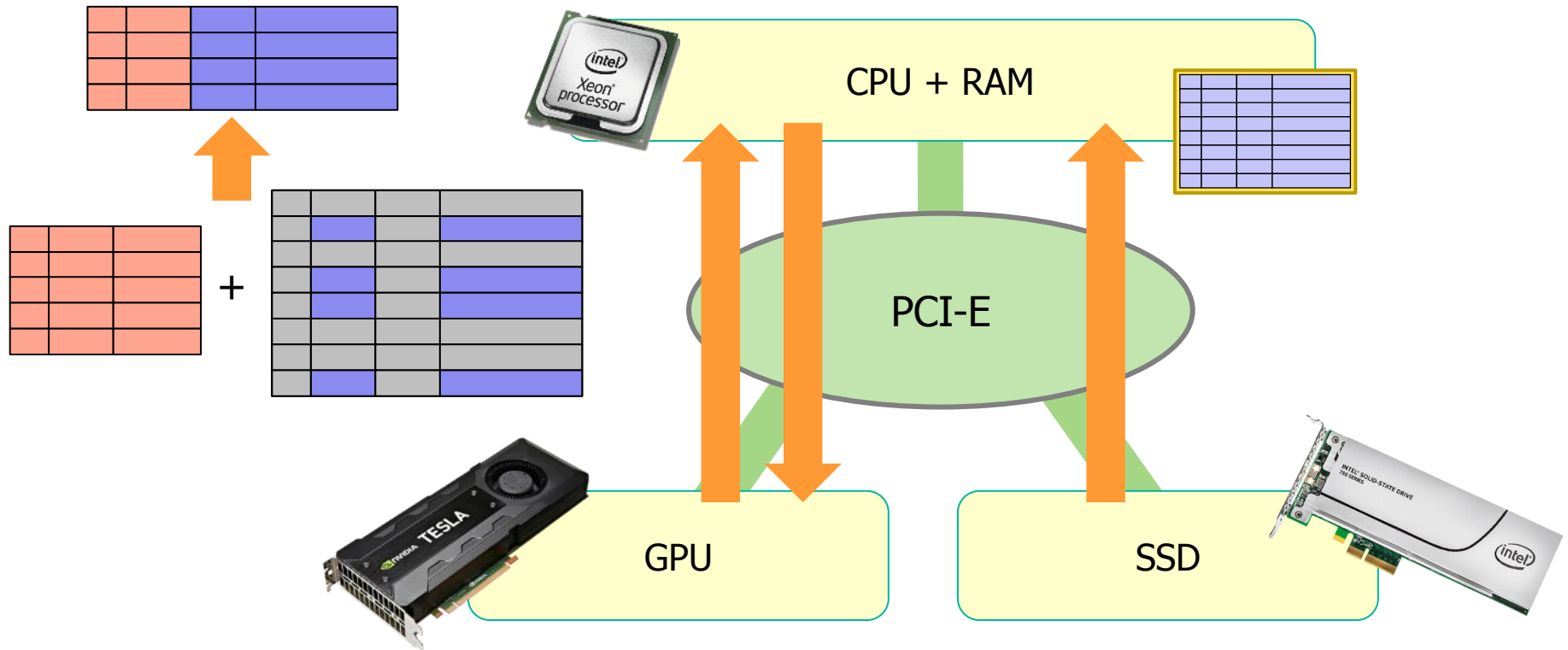
Data Flow in analytic queries



- ① Data load from storage to CPU/RAM
- ② Remove invisible rows (Select)
- ③ Remove unreferenced columns (Projection)

Table			

Data Flow in analytic queries

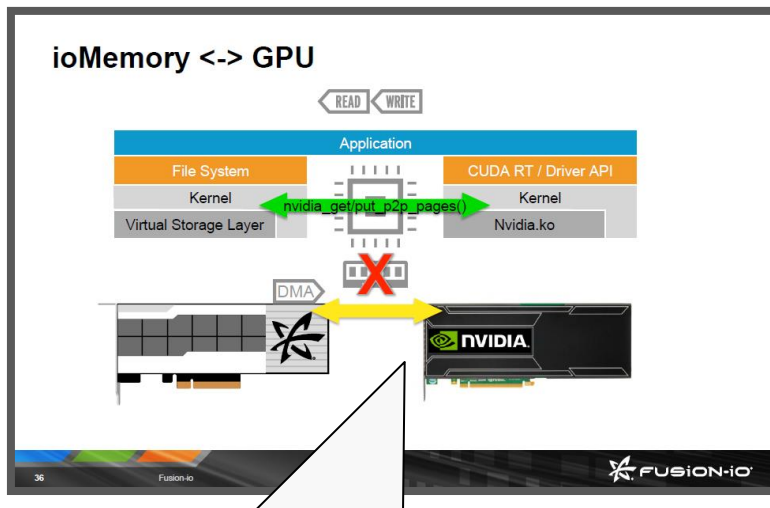
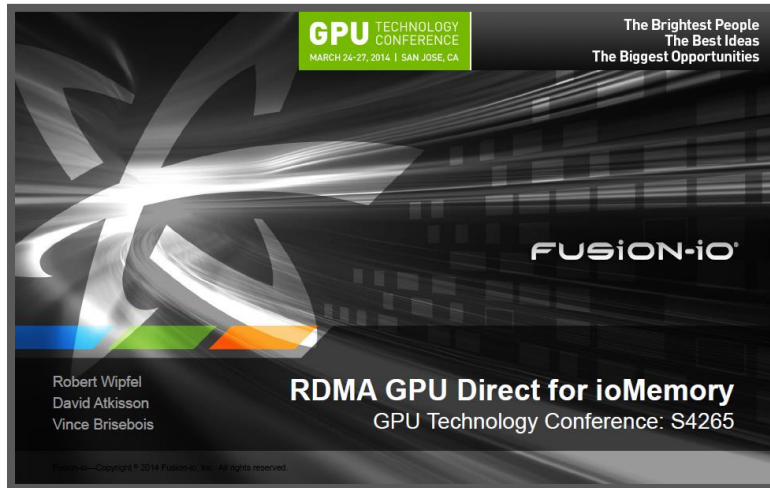


- ① Data load from storage to CPU/RAM
- ② Remove invisible rows (Select)
- ③ Remove unreferenced columns (Projection)
- ④ Join with other tables (Join)

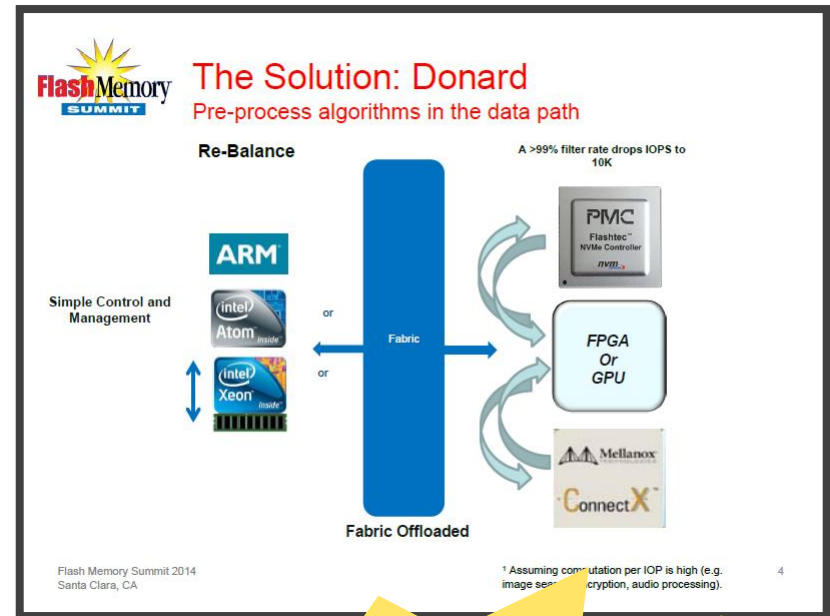
↑ The job of CPU

Table			

SSD-to-GPU Direct

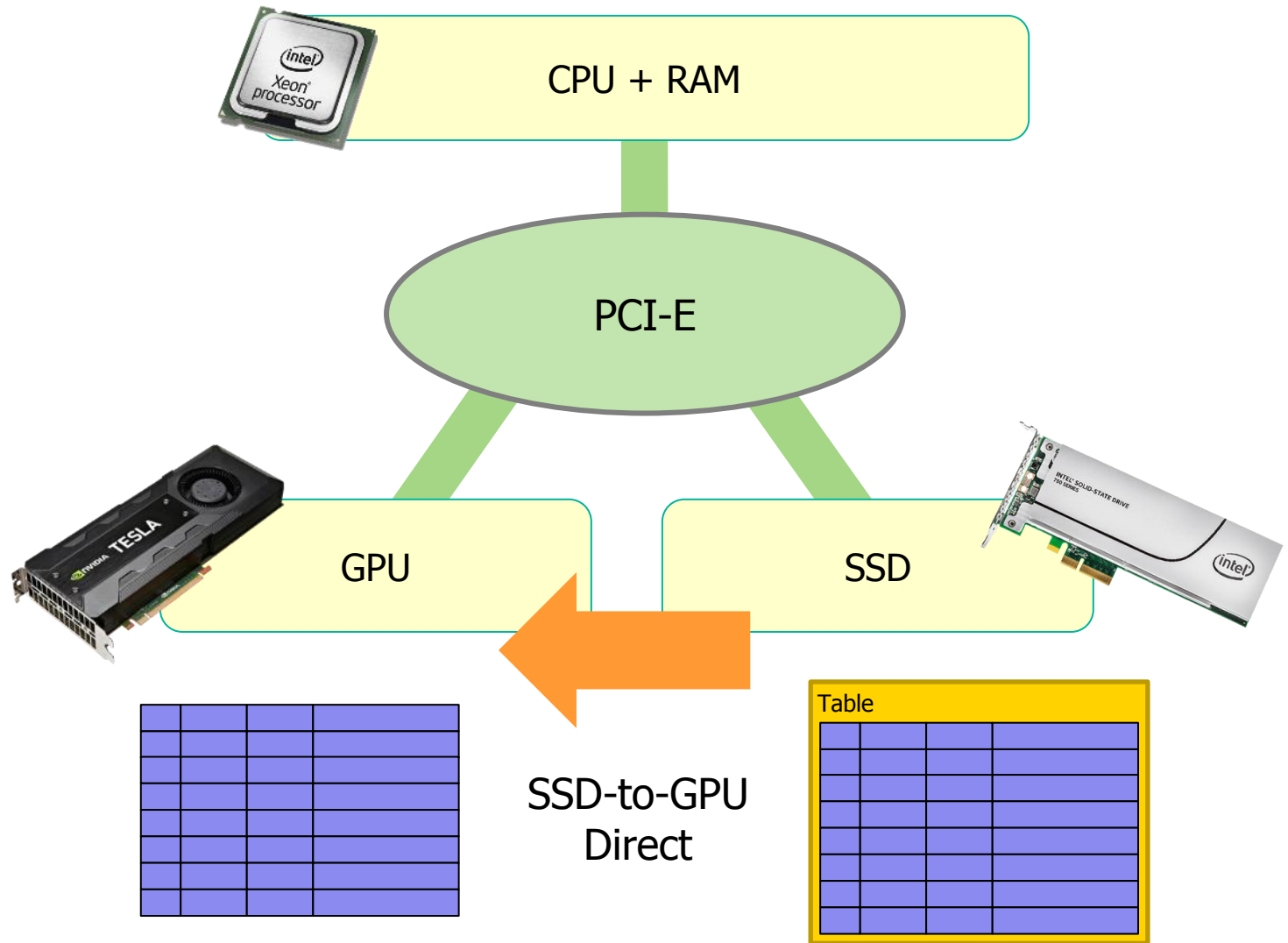


Data transfer between SSD and GPU, bypass CPU/RAM

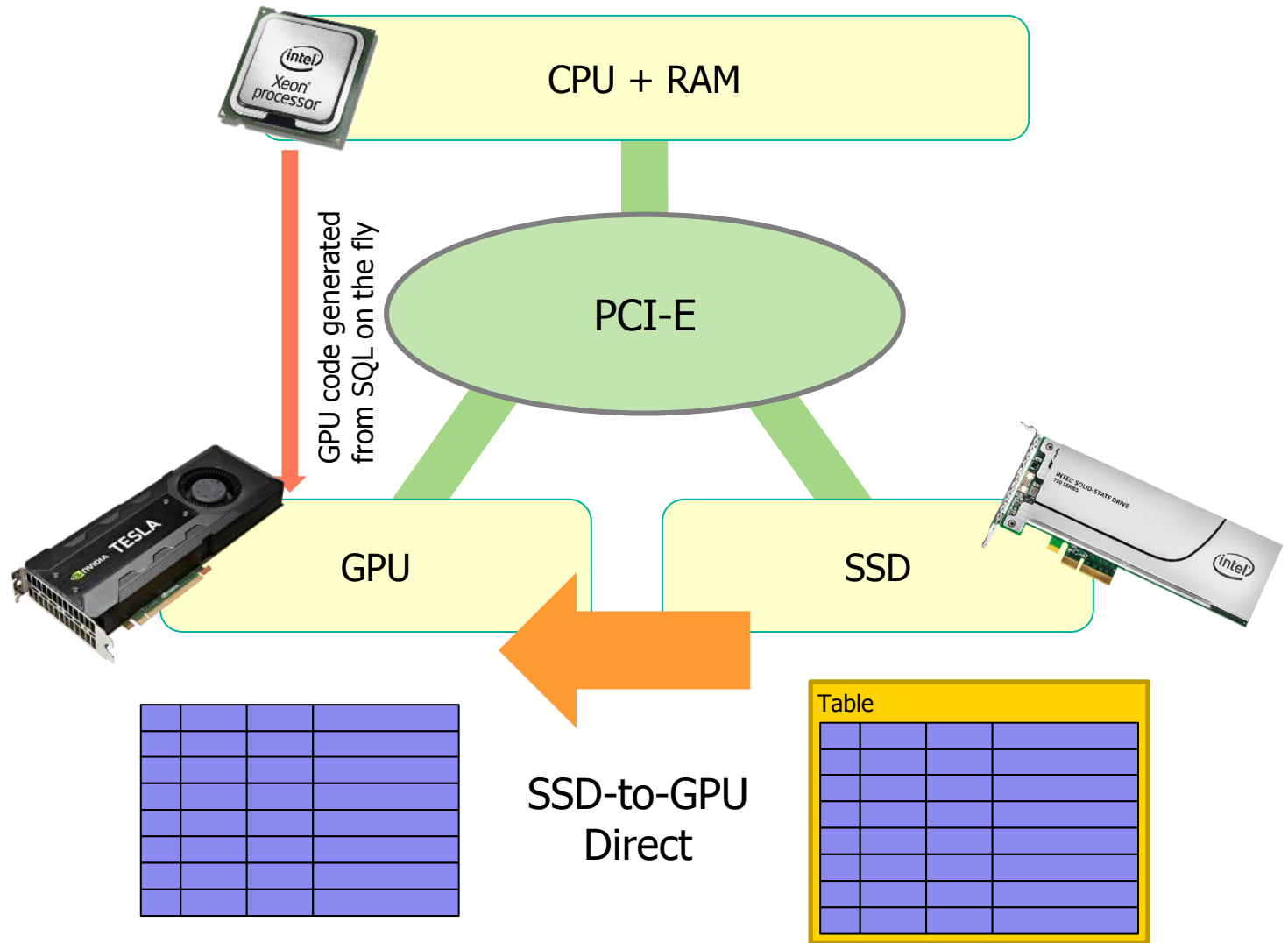


Also available on NVMe,
not only Fusion-IO

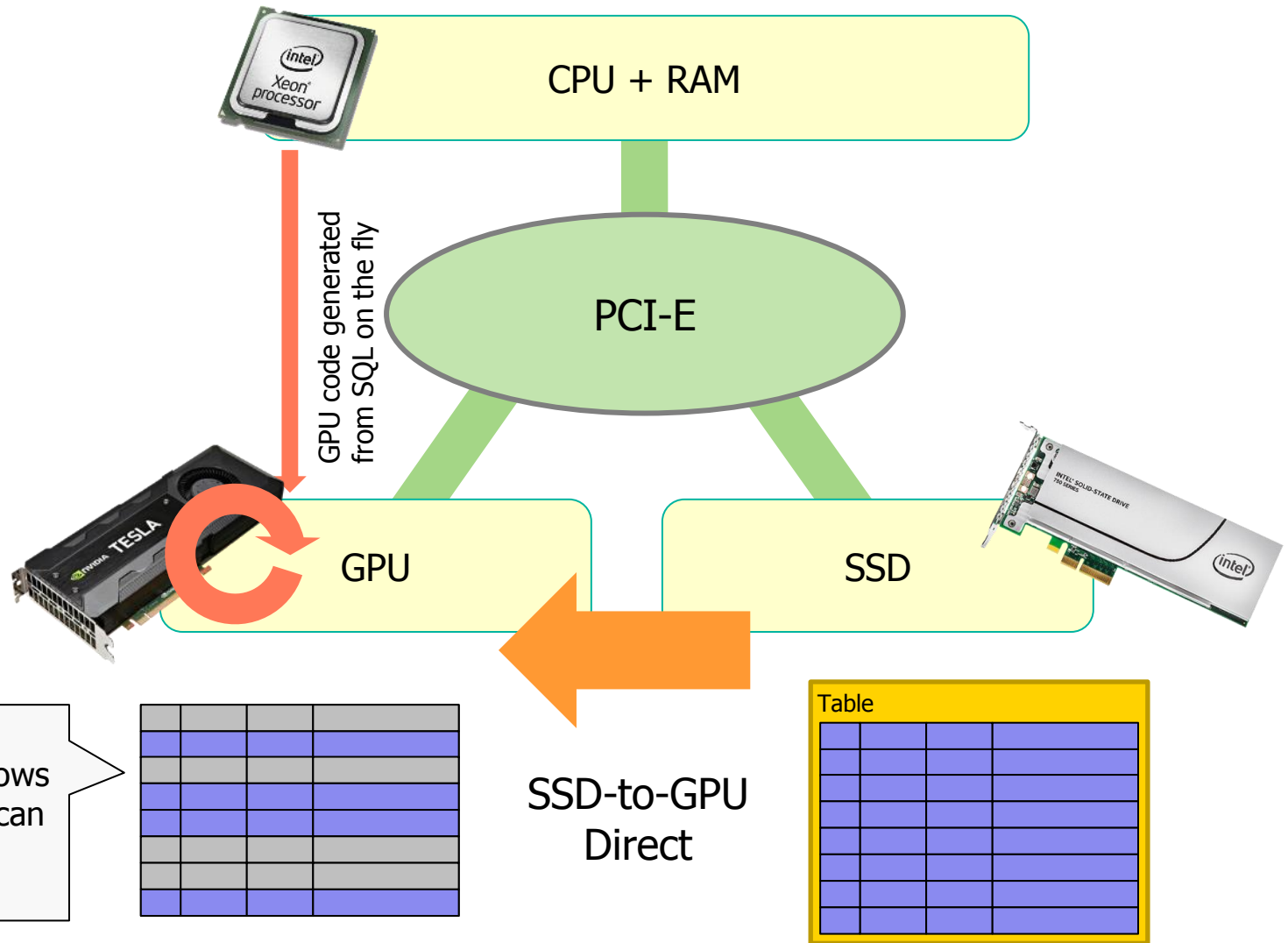
Data Flow in analytic queries (1/3) – Basic



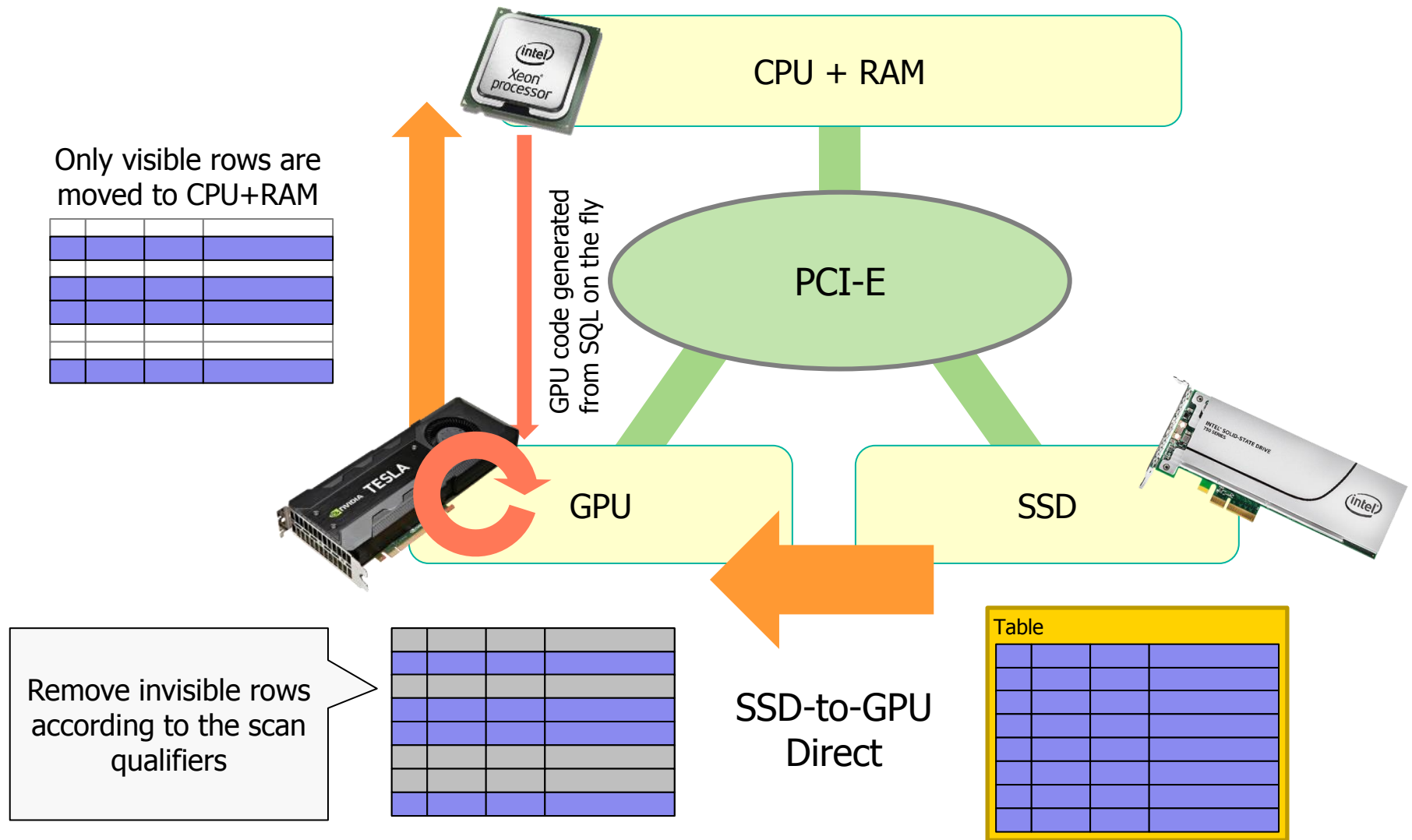
Data Flow in analytic queries (1/3) – Basic



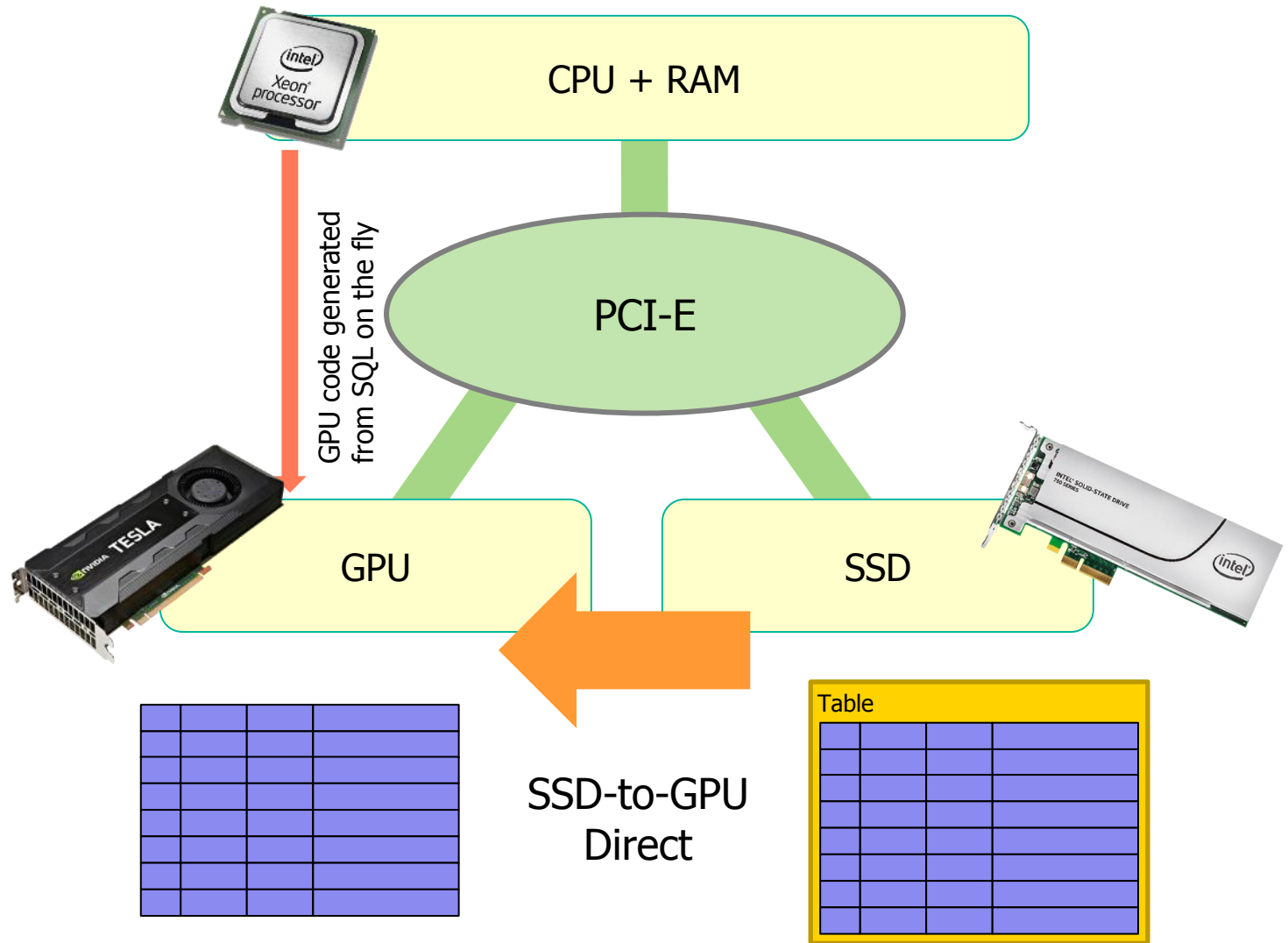
Data Flow in analytic queries (1/3) – Basic



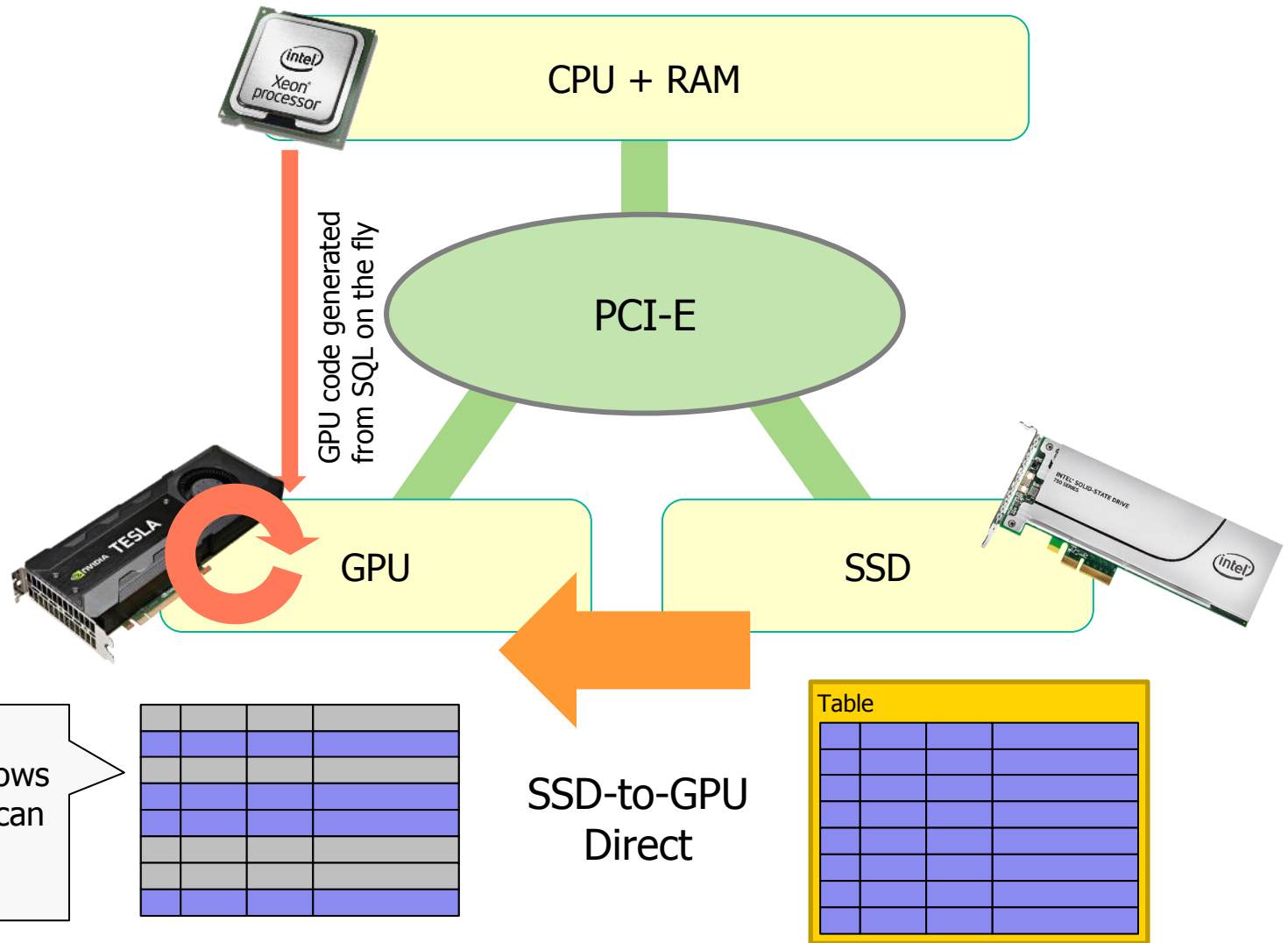
Data Flow in analytic queries (1/3) – Basic



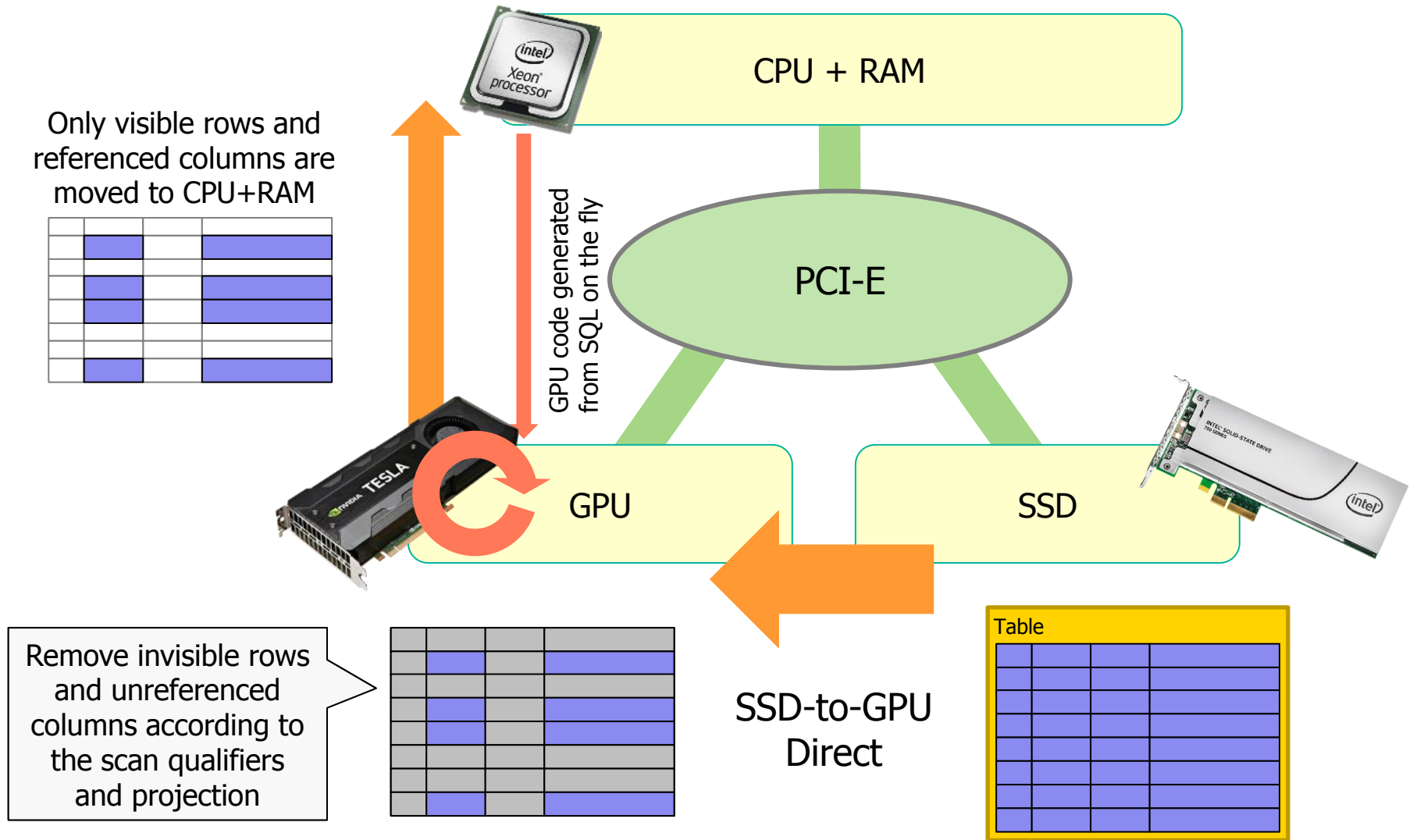
Data Flow in analytic queries (2/3) – Advanced



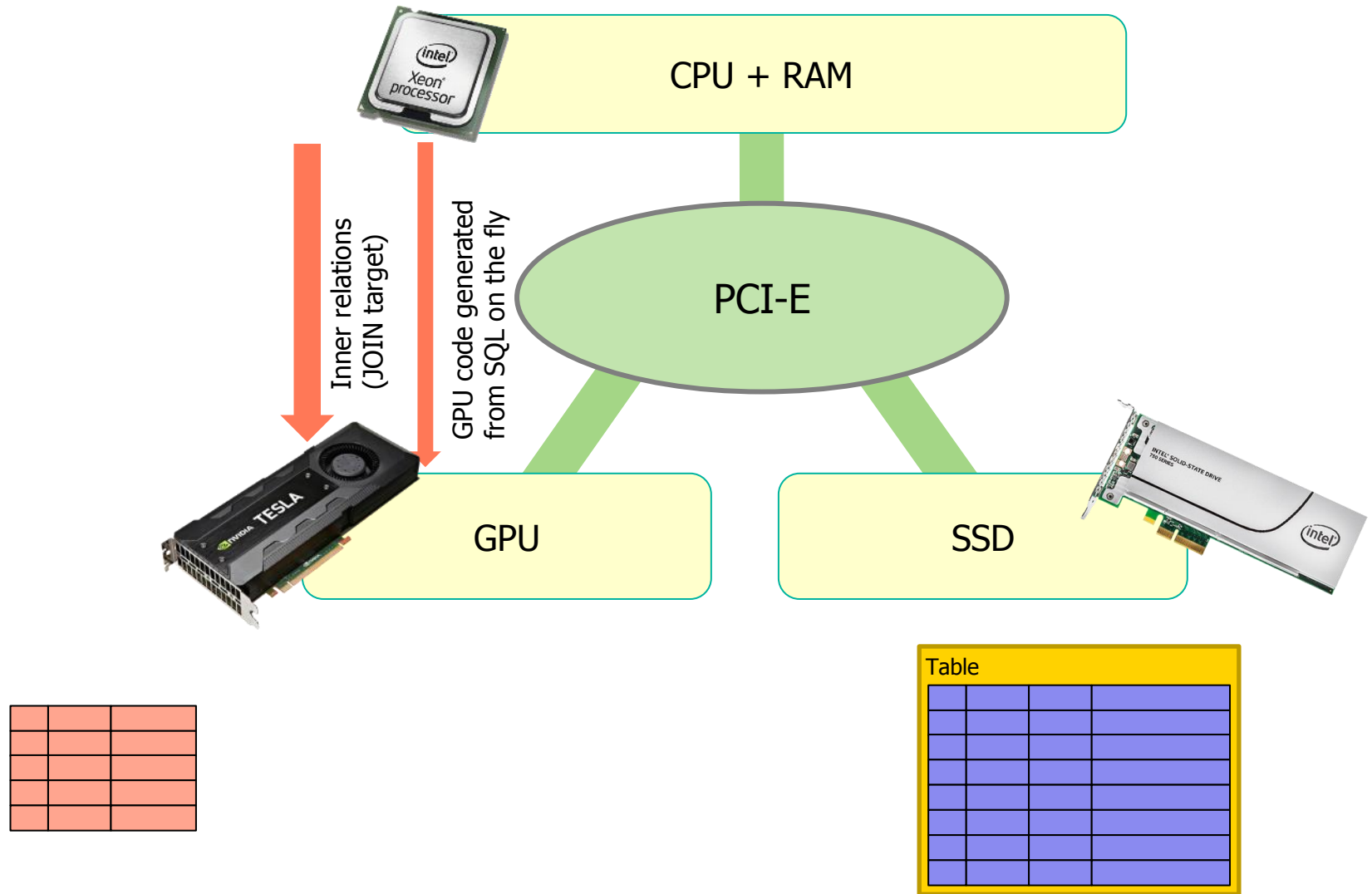
Data Flow in analytic queries (2/3) – Advanced



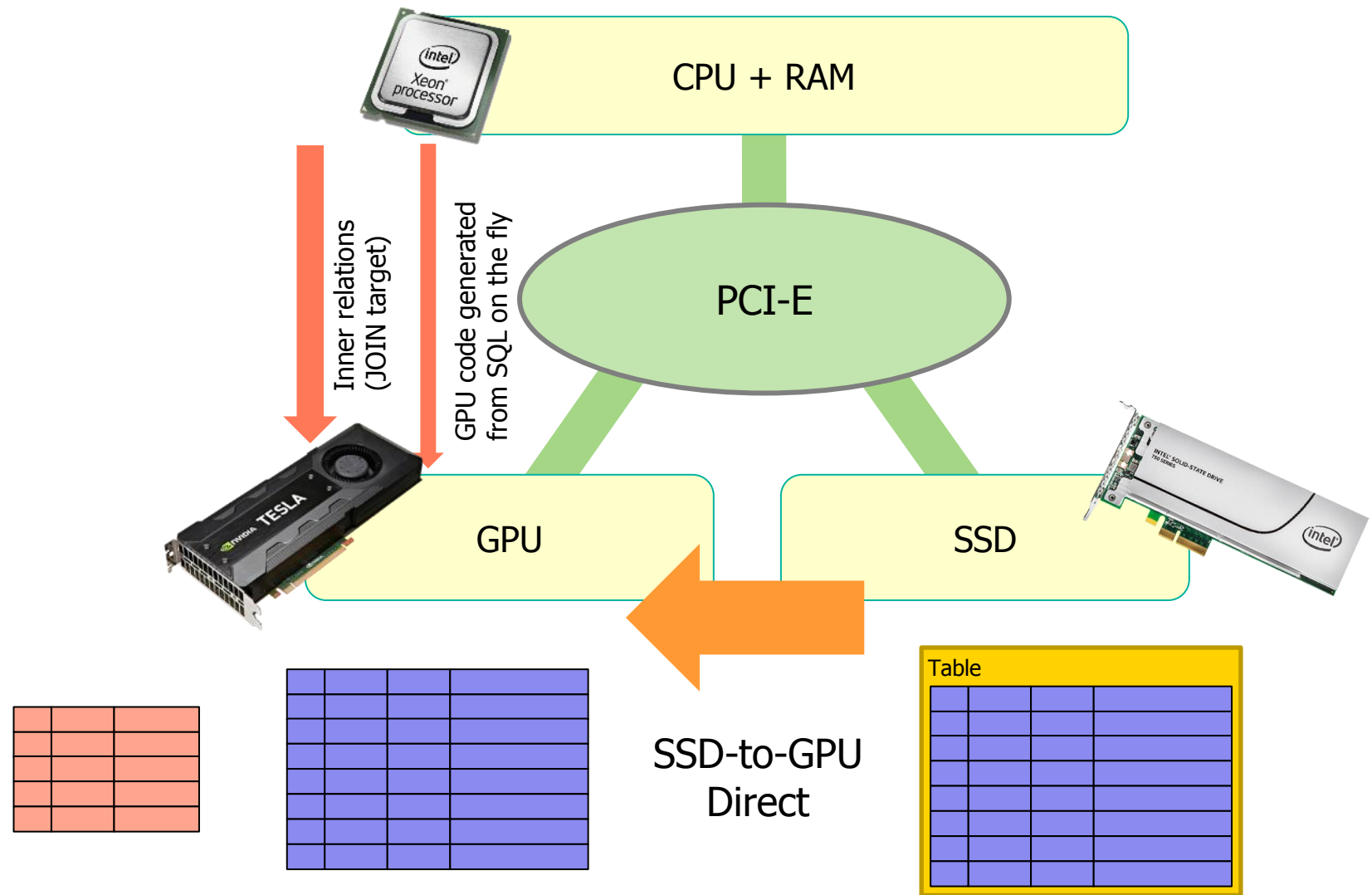
Data Flow in analytic queries (2/3) – Advanced



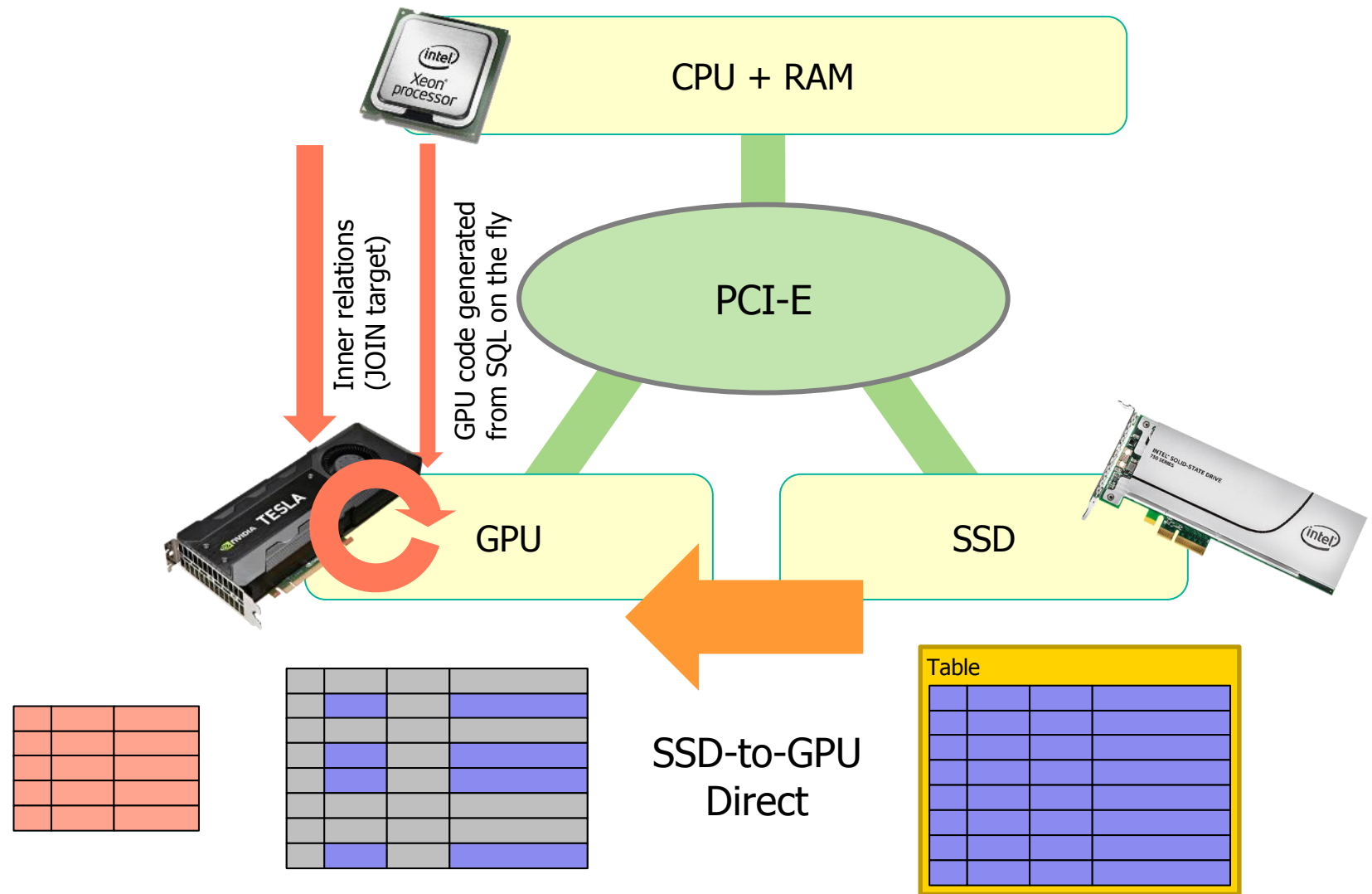
Data Flow in analytic queries (3/3) – Ultimate



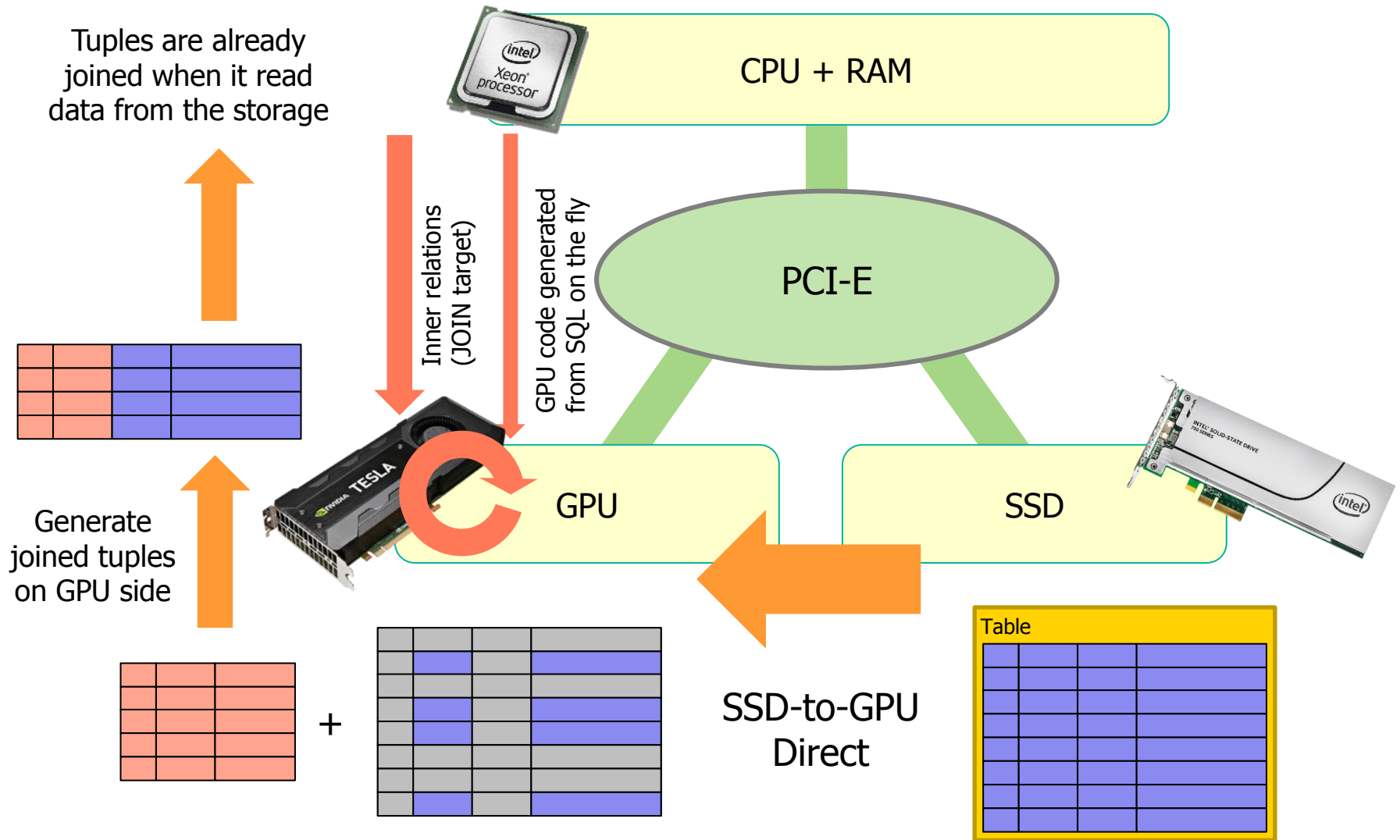
Data Flow in analytic queries (3/3) – Ultimate



Data Flow in analytic queries (3/3) – Ultimate



Data Flow in analytic queries (3/3) – Ultimate



Primitive Technologies

- NVIDIA GPUDirect enhancement on NVMe device driver
 - Interaction between NVMe and NVIDIA drivers are needed
- Usage statistics of shared_buffers per relations
 - To avoid SSD→GPU direct on relations that is already preloaded
- Add new access mode to shared_buffers
 - Nobody can make the buffer dirty under the SSD→GPU Direct transfer

We are welcome all the developer
who join to PG-Strom project

Coming Soon?

