Microsoft

Tensor Query Processing: How do we ride the AI investment wave for database analytics?

Carlo Curino

Microsoft Azure Data
Gray Systems Lab

Microsoft

Tensor Query Processing: ~~How do we ride the AI investment wave for database analytics?~~ Ok, Skynet is coming after the human race... but can we run queries on a T850?

Carlo Curino

Microsoft Azure Data
Gray Systems Lab

Image courtesy of pngimg.com

Tensor Query Processing: ~~How do we ride the AI investment wave for database analytics?~~

Ok, Skynet is coming after the human race… but can we run queries on a ~~T850~~ A100?

Carlo Curino

Microsoft Azure Data
Gray Systems Lab

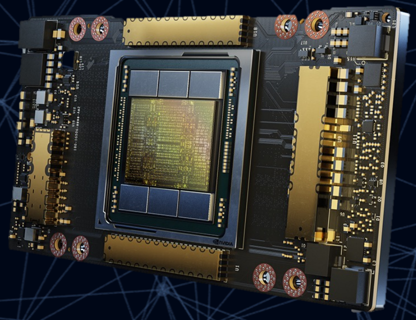
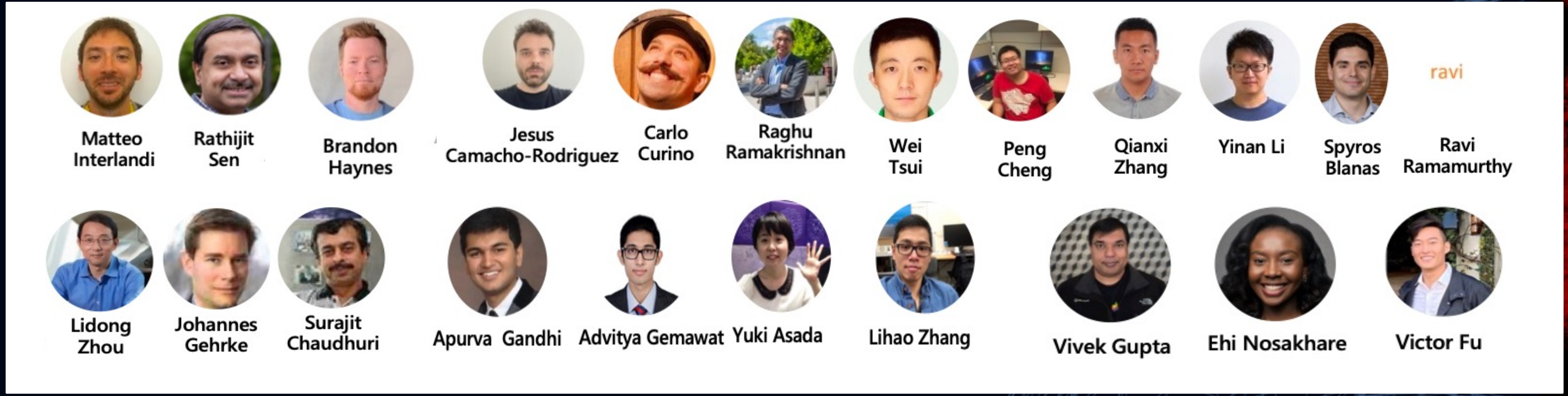


Image courtesy of pngimg.com

The Resistance (delaying Skynet keeping its HW busy with SQL)

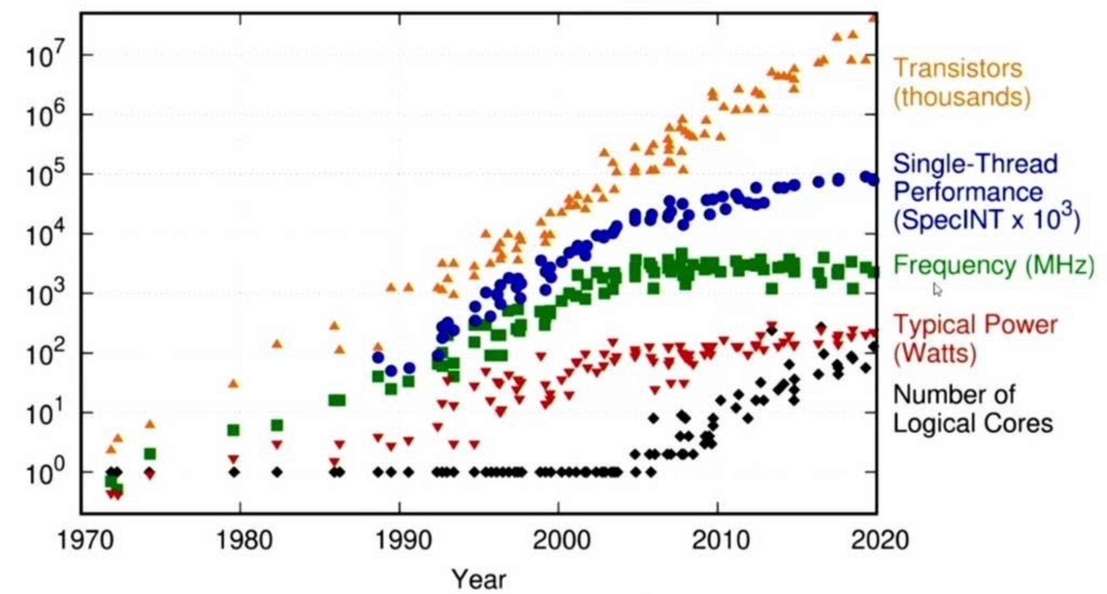


1. Wei Cui, Qianxi Zhang, Spyros Blanas, Jesús Camacho-Rodríguez, Brandon Haynes, Yinan Li, Ravi Ramamurthy, Peng Cheng, Rathijit Sen, and Matteo Interlandi. 2023. Query Processing on Gaming Consoles. In Proceedings of the 19th International Workshop on Data Management on New Hardware (DaMoN '23). Association for Computing Machinery, New York, NY, USA, 86–88. https://doi.org/10.1145/3592980.3595313
2. Matthias Boehm, Matteo Interlandi, and Chris Jermaine. 2023. Optimizing Tensor Computations: From Applications to Compilation and Runtime Techniques. In Companion of the 2023 International Conference on Management of Data (SIGMOD '23). Association for Computing Machinery, New York, NY, USA, 53–59. https://doi.org/10.1145/3555041.3589407
3. Dong He, Supun C Nakandala, Dalitso Banda, Rathijit Sen, Karla Saur, Kwanghyun Park, Carlo Curino, Jesús Camacho-Rodríguez, Konstantinos Karanasos, and Matteo Interlandi. 2022. Query processing on tensor computation runtimes. Proc. VLDB Endow. 15, 11 (July 2022), 2811–2825. https://doi.org/10.14778/3551793.3551833
4. Apurva Gandh, Yuki Asada, Victor Fu, Advitya Gemawat, Lihao Zhang, Rathijit Sen, Carlo Curino, Jesús Camacho-Rodríguez, Matteo Interlandi. The Tensor Data Platform: Towards an AI-centric Database System. CIDR 2023
5. Yuki Asada, Victor Fu, Apurva Gandhi, Advitya Gemawat, Lihao Zhang, Dong He, Vivek Gupta, Ehi Nosakhare, Dalitso Banda, Rathijit Sen, and Matteo Interlandi. 2022. Share the tensor tea: how databases can leverage the machine learning ecosystem. Proc. VLDB Endow. 15, 12 (August 2022), 3598–3601. https://doi.org/10.14778/3554821.3554853
6. Dimitrios Koutsoukos, Supun Nakandala, Konstantinos Karanasos, Karla Saur, Gustavo Alonso, and Matteo Interlandi. 2021. Tensors: an abstraction for general data processing. Proc. VLDB Endow. 14, 10 (June 2021), 1797–1804. https://doi.org/10.14778/3467861.3467869
7. Gyeong-In Yu, Saeed Amizadeh, Sehoon Kim, Artidoro Pagnoni, Ce Zhang, Byung-Gon Chun, Markus Weimer, and Matteo Interlandi. 2021. WindTunnel: towards differentiable ML pipelines beyond a single model. Proc. VLDB Endow. 15, 1 (September 2021), 11–20. https://doi.org/10.14778/3485450.3485452
8. Supun Nakandala, Karla Saur, Gyeong-In Yu, Konstantinos Karanasos, Carlo Curino, Markus Weimer, and Matteo Interlandi. 2021. A Tensor Compiler for Unified Machine Learning Prediction Serving. In OSDI 2020.

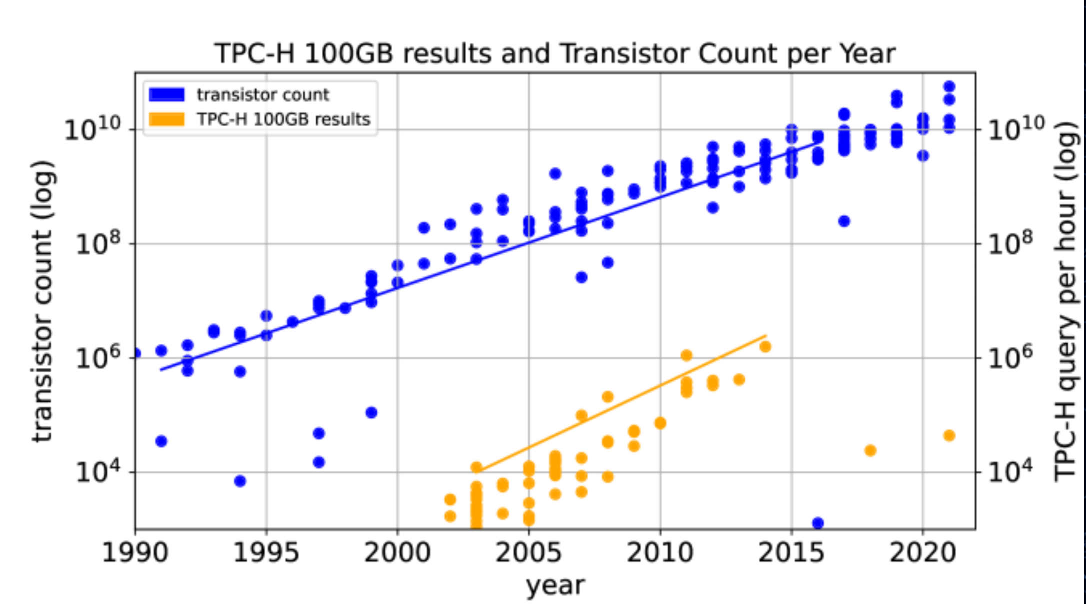
Resistance | Terminator Wiki | Fandom

Backdrop: DB perf gains are hard to come by!

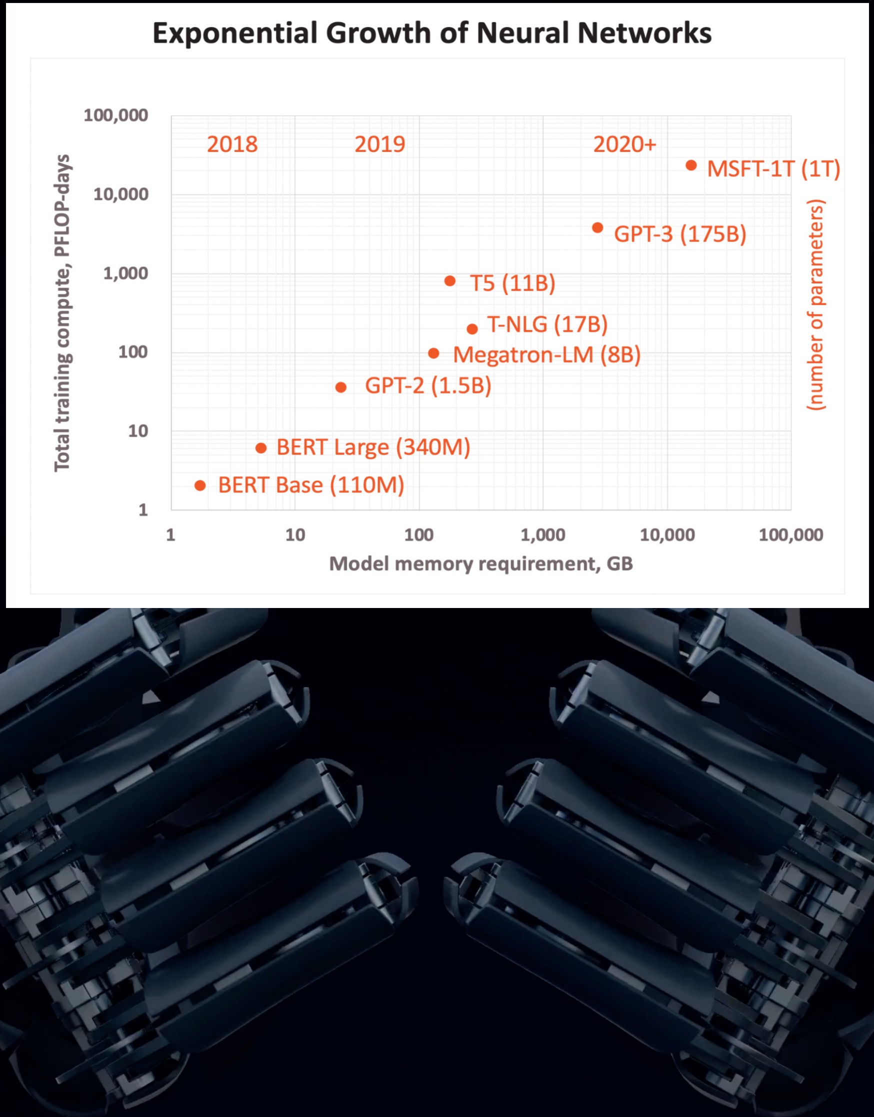
Slowing HW-driven perf improvements



Decades of optimization saturated SW gains



Backdrop: AI interest (and HW) is exploding!



https://www.statista.com/statistics/1003890/worldwide-artificial-intelligence-hardware-market-revenues/

Big $$$ spent on Special HW for NN

VCs are pouring $2B/quarter

Market expected to exceed $200B/year by 2025.

Some examples of AI HW

Type	Tech (nm)	Architecture	# Trans.	Power	Cache	Mem / Storage	Mem BW
NN-Chips	7	Cerebras WSE-2	2.6 T	20 KW	40 GB	4 TB - 2.4 PB	20 PB/sec
GPU	7	NVIDIA A100	54 B	400 W	40 MB	40/80 GB	1.6 TB/sec (HBM)

* Data courtesy of Rathijit Sen

Tensor Runtime

Tensor as de-facto API

Very large/active communities

What about HW investments for Database?

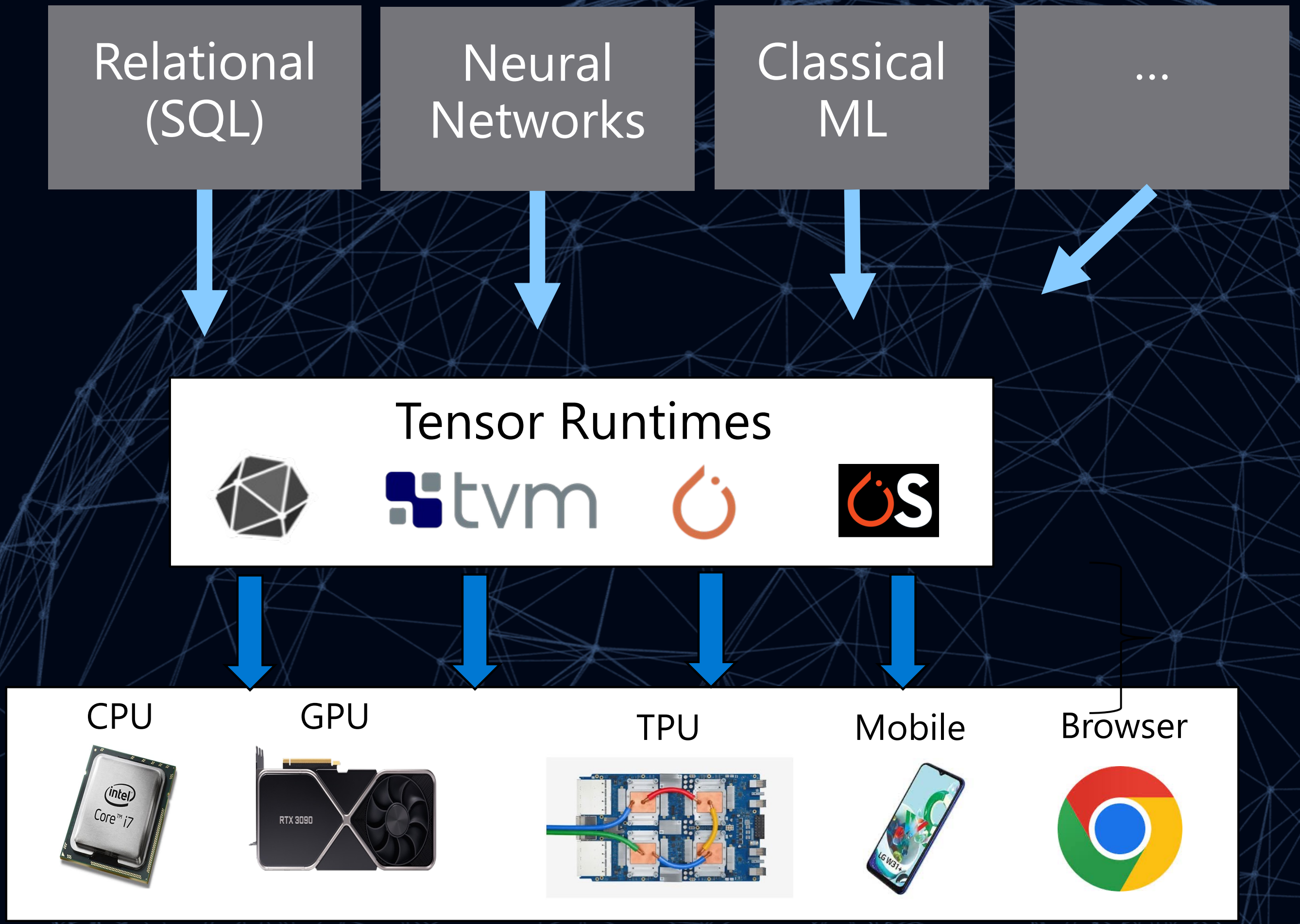
- ❌ Existent but modest w.r.t AI
- ❌ Porting to each new HW is a costly N x M problem

Our Goal: Save Humanity from Skynet!

How? By keeping the AI HW busy with database analytics queries

Main Idea: Tensor Query Processing

Compile SQL, Classical ML, etc. to the popular tensor abstraction!



Pros and Cons of ”Tensor Query Processing”

Pros

- ✓ Leverage the massive investments in special HW
- ✓ Scalable Approach (tensor runtimes are getting ported to each new HW)

Cons

- ? Is this even possible?
- ? What about performance? (as compared with state-of-the-art)
- ? How expensive is it going to be? (engineering wise)

System Design

SQL Query

```sql
SELECT
  MAX(p_supplycost)
  AS price,
  s_name AS supp
FROM supplier
JOIN partsupp
ON
  ps_suppkey=s_suppkey
GROUP BY
  supplier.s_name
ORDER BY
  price DESC;
```

Parsing Layer

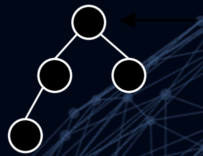
System Design

SQL Query

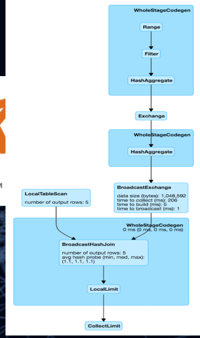
```
SELECT
  MAX(p_supplycost)
  AS price,
  s_name AS supp
FROM supplier
JOIN partsupp
ON
  ps_suppkey=s_suppkey
GROUP BY
  supplier.s_name
ORDER BY
  price DESC;
```

Parsing Layer

IR Graph

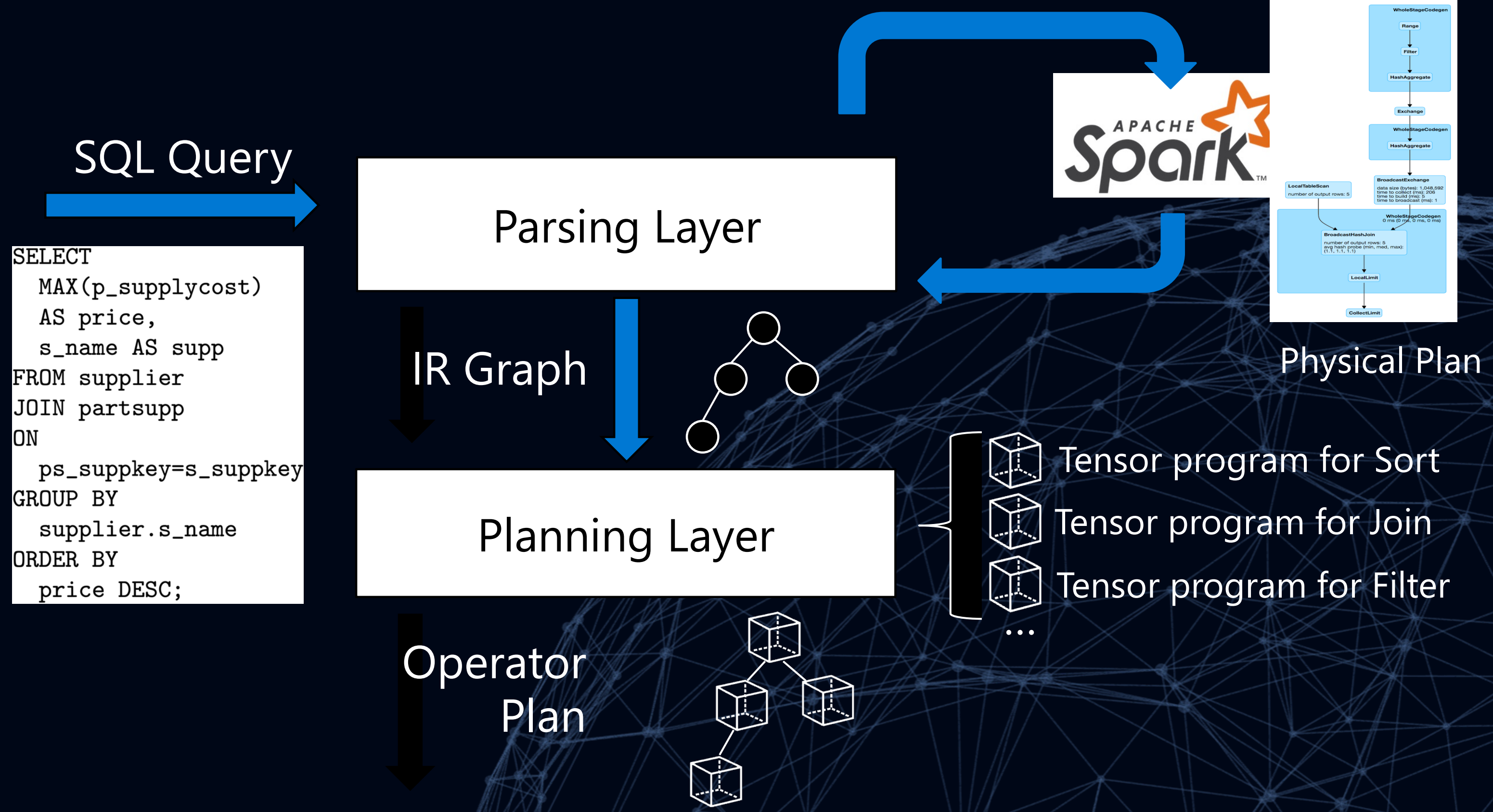


Physical Sort Operator

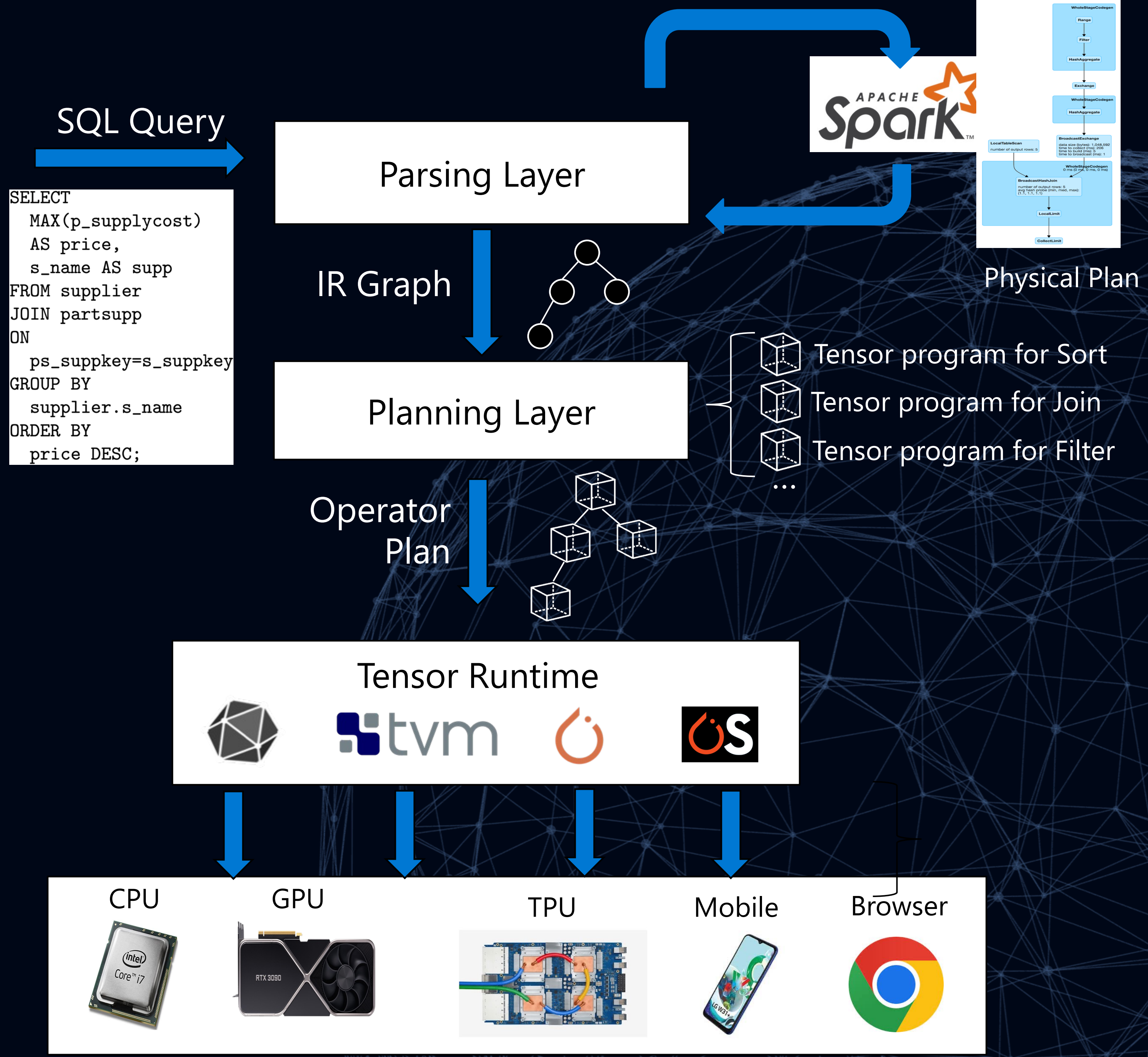


Physical Plan

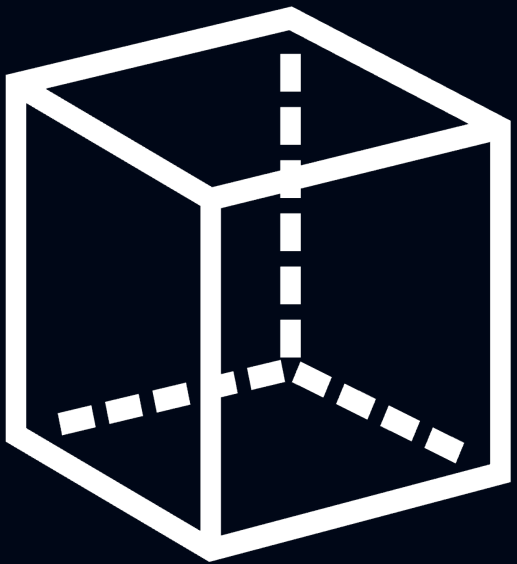
System Design



System Design



Example: Tensor Program for Filter



```
WHERE L_QUANTITY < 24
```

Opt 1:

```
mask = torch.lt(l_quantity, 24)
output = torch.masked_select(l_quantity, mask)
```

Opt 2:

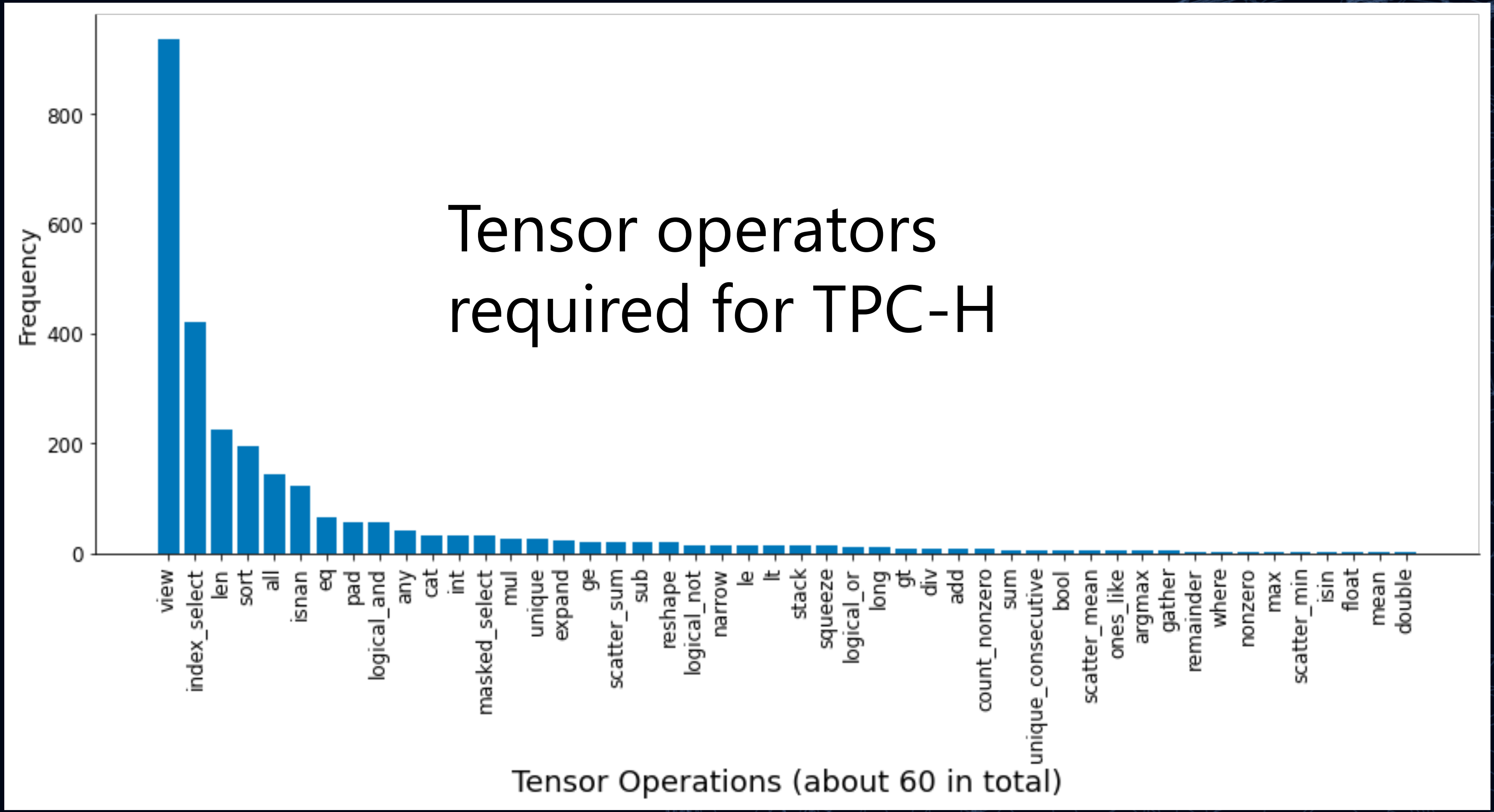
```
mask = torch.lt(l_quantity, 24)
idx = torch.nonzero(mask)
output = torch.index_select(l_quantity, dim=0, idx )
```

Numeric as
N x 1 tensors

Dates as
N x 1 numeric

Strings as UTF-8
N x max_length

Implementing SQL operators using tensor ops



Pros and Cons of "Tensor Query Processing"

Pros

- ✓ Leverage the massive investments in special HW
- ✓ Scalable Approach (tensor runtimes are getting ported to each new HW)

Cons

- ? Is this even possible?
- ? What about performance? (as compared with state-of-the-art)
- ? How expensive is it going to be? (engineering wise)

Pros and Cons of ”Tensor Query Processing”

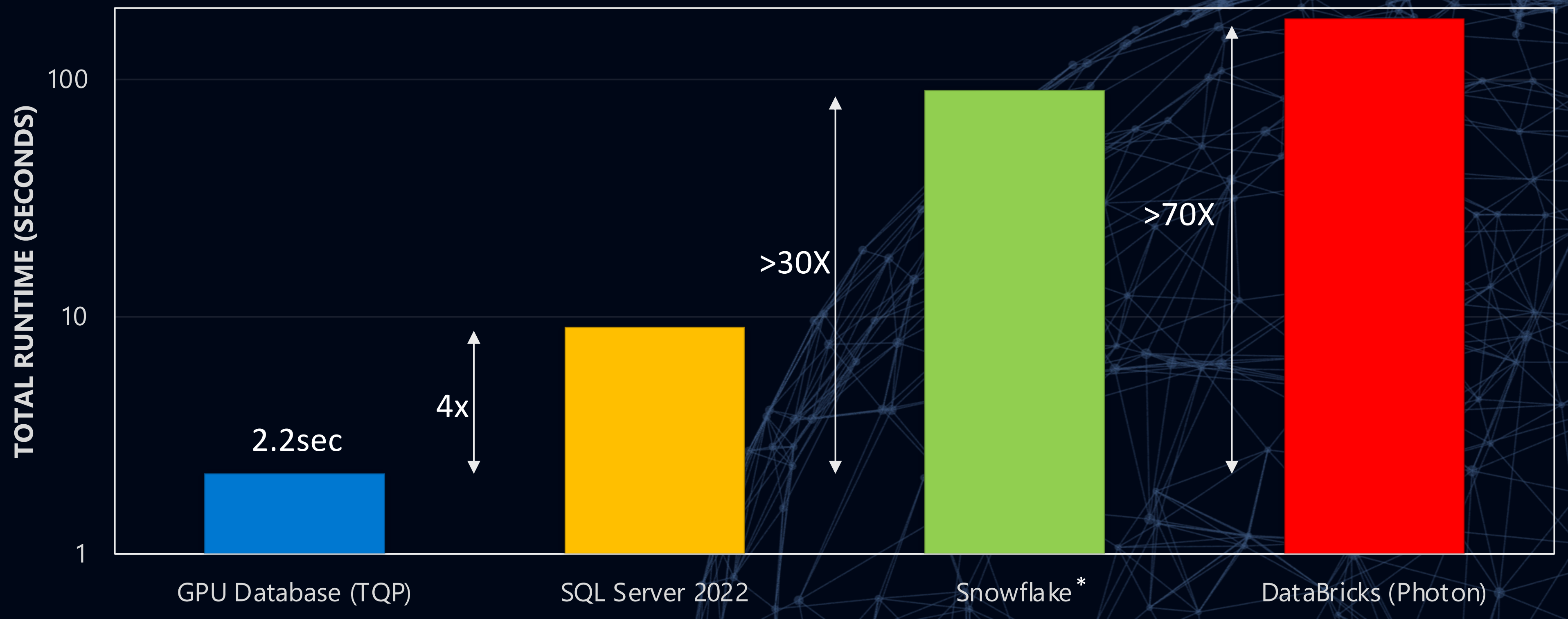
Pros

- ✓ Leverage the massive investments in special HW
- ✓ Scalable Approach (tensor runtimes are getting ported to each new HW)
- ✓ Is this even possible? → YES we can easily cover TPC-H

Cons

- ? What about performance? (as compared with state-of-the-art)
- ? How expensive is it going to be? (engineering wise)

TPCH SF 50

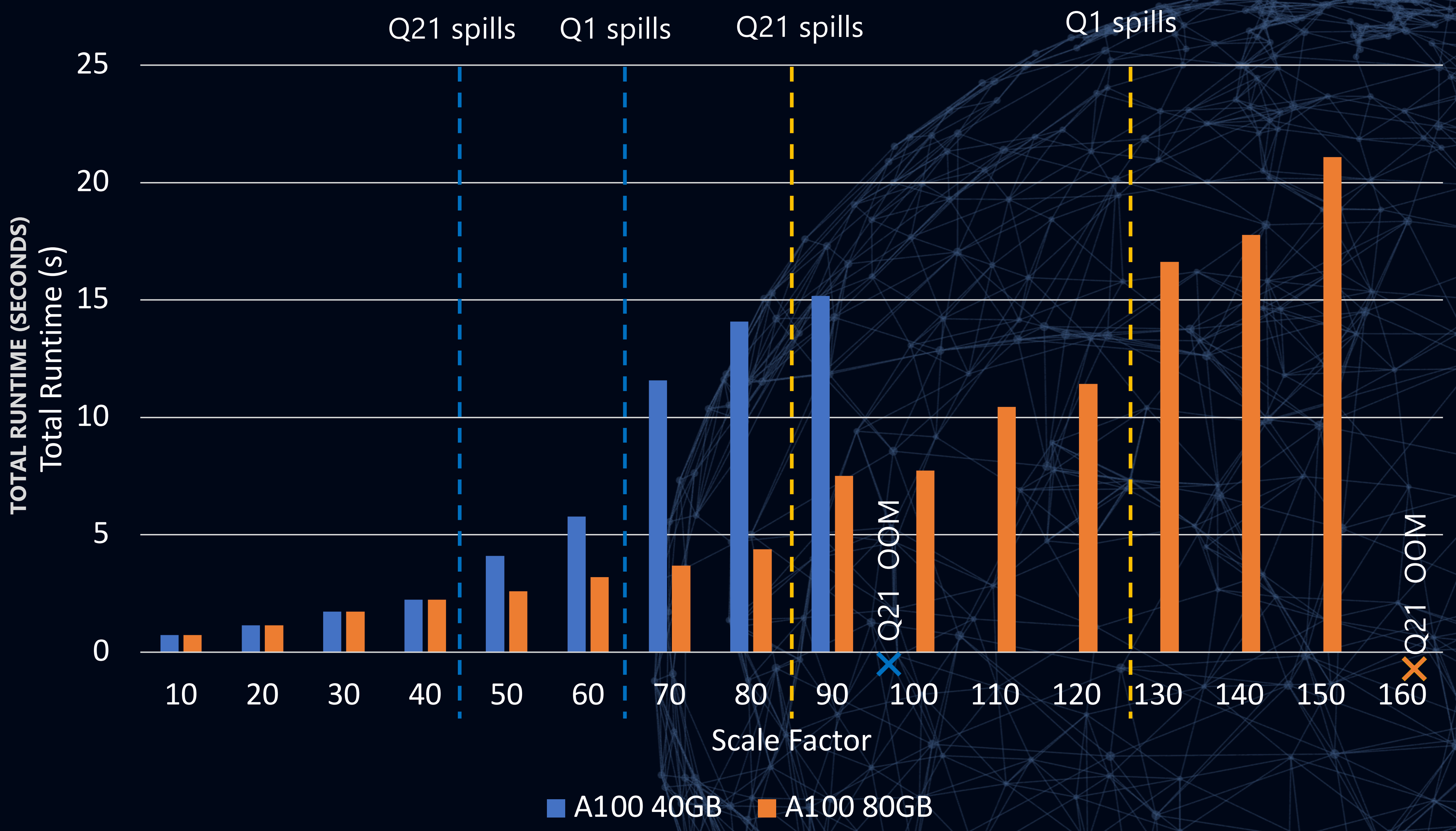


TPCH Scale Factor 50

*indirect estimate

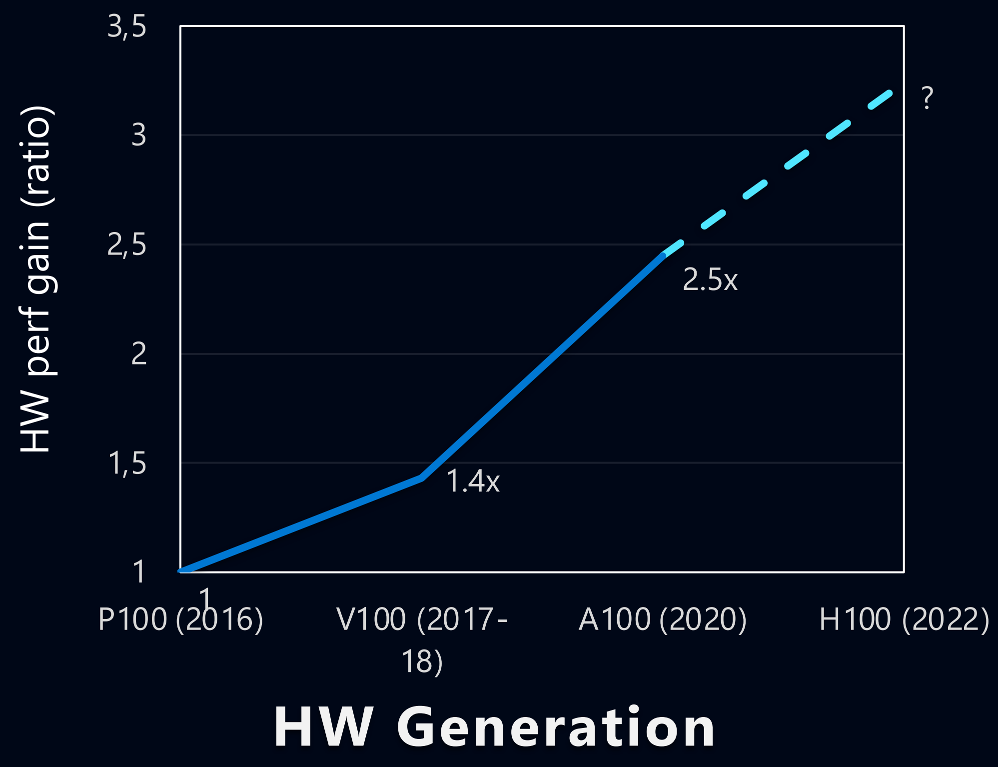
SQL Server and DataBricks: Standard D64s v5 (64 vcpus, 256 GiB memory)
TQP: Standard NC24ads A100 v4 (24 vcpus, 220 GiB memory)
about 50% more expensive than the CPU HW

Buffer pools and TQP Scalability

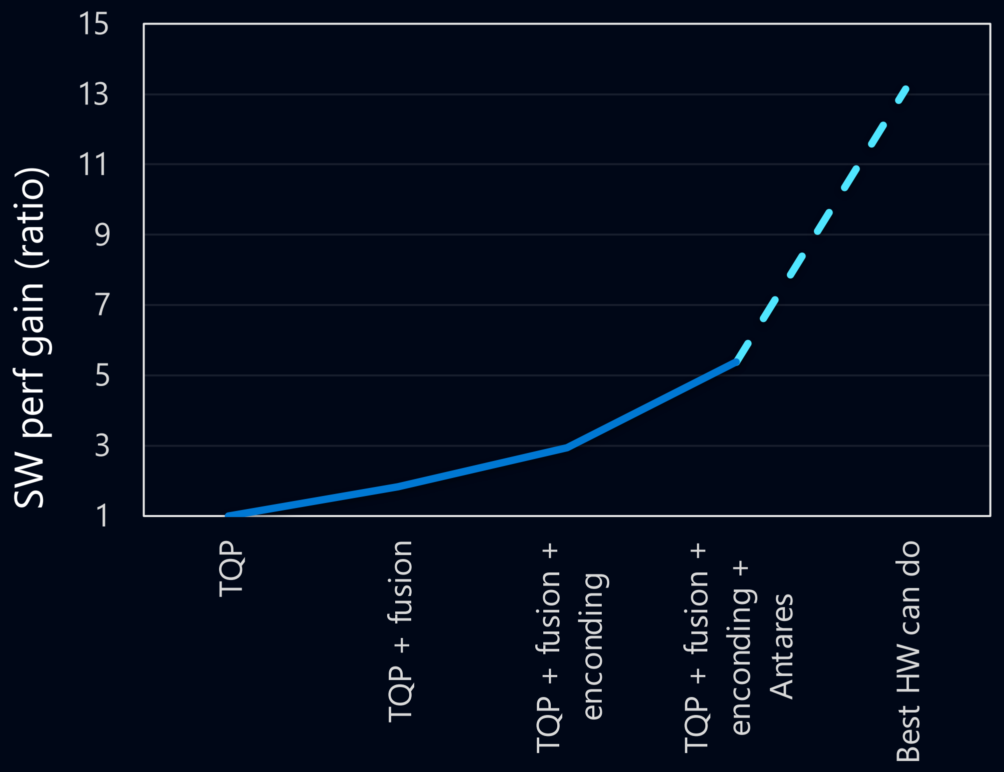


Is this a one-time gain?

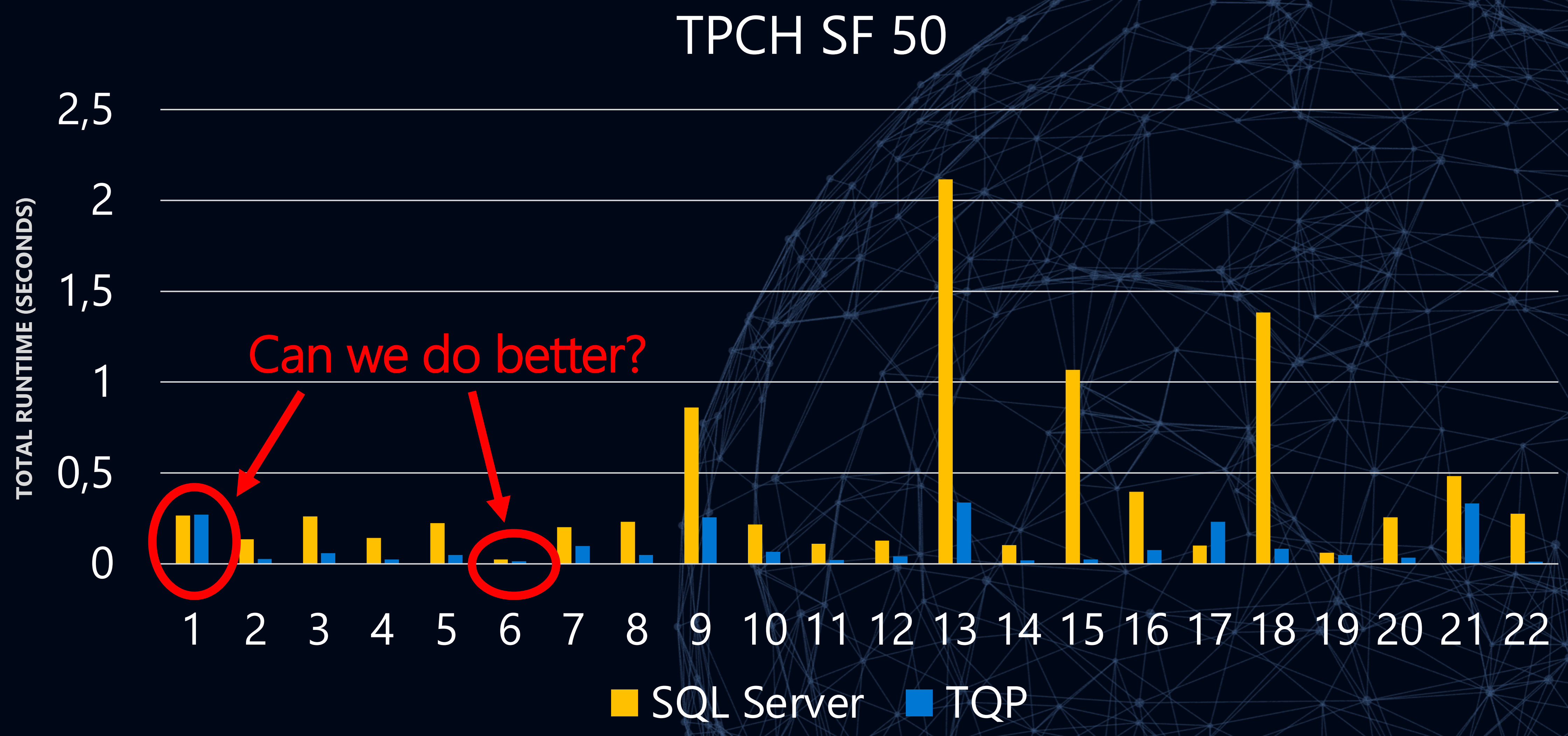
More perf coming from HW improvements



Lots of headroom via SW optimization



TPCH SF 50 drilldown



Missing Optimizations

```
WHERE L_QUANTITY < 24
```

Opt 1:

```
mask = torch.lt(l_quantity, 24)
output = torch.masked_select(l_quantity, mask)
```

Opt 2:

```
mask = torch.lt(l_quantity, 24)
idx = torch.nonzero(mask)
output = torch.index_select(l_quantity, dim=0, idx )
```

Numeric as
N x 1 tensors

Dates as
N x 1 numeric

Strings as UTF-8
N x max_length

Ongoing:
HW-customized operators
Operator Fusion
Representation / compression
Co-execution of CPU/GPU
IO-bottlenecks / Distributed exe

Future:
Tensor-aware Optimizer

TQP: Computing on RLE compressed data

QP directly on data encoded for compression
- Raw datasets can be larger than GPU memory
- Lower data transfer overhead
- Faster query processing

Encodings: dictionary, **RLE**, bit-packing, …

TQP/Vanilla

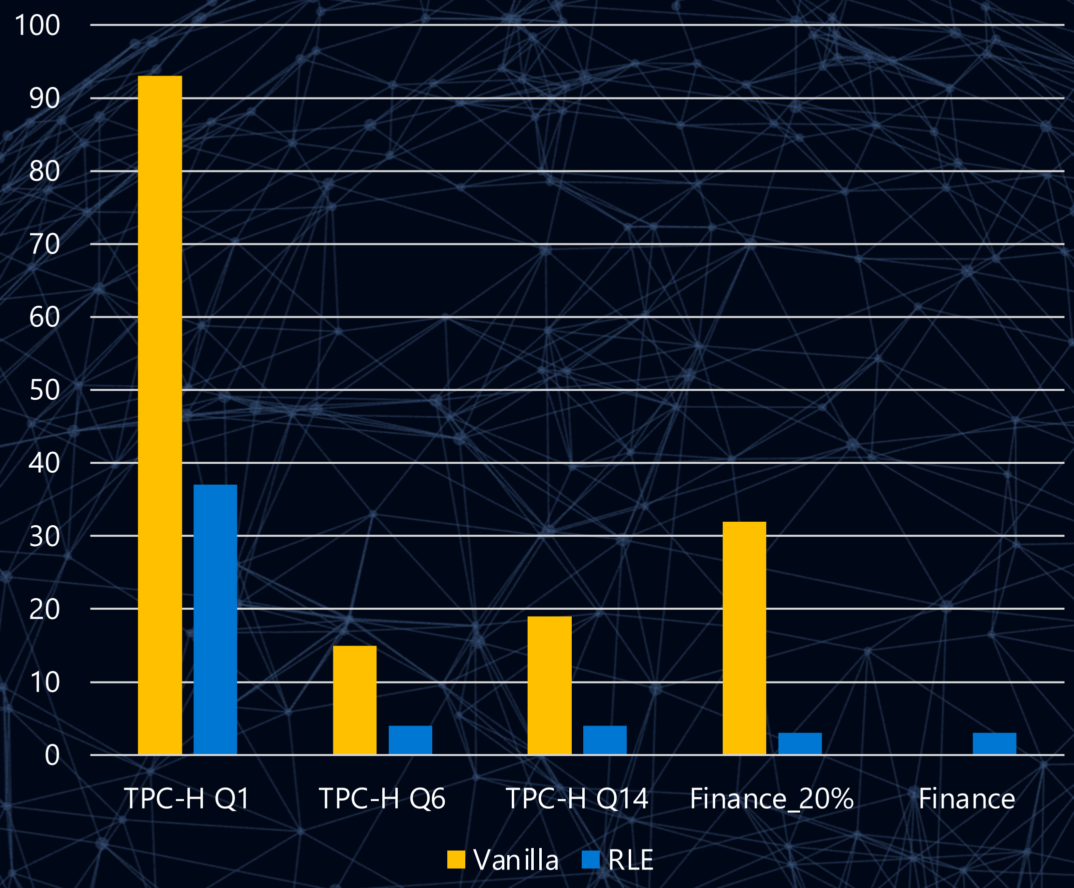
```
Rome________,
Rome________,
Rome________,
Rome________,
Tokyo_______,
Tokyo_______,
Tokyo_______,
Los Angeles,
Los Angeles,
Los Angeles,
Los Angeles,
Rome________
```

Dict

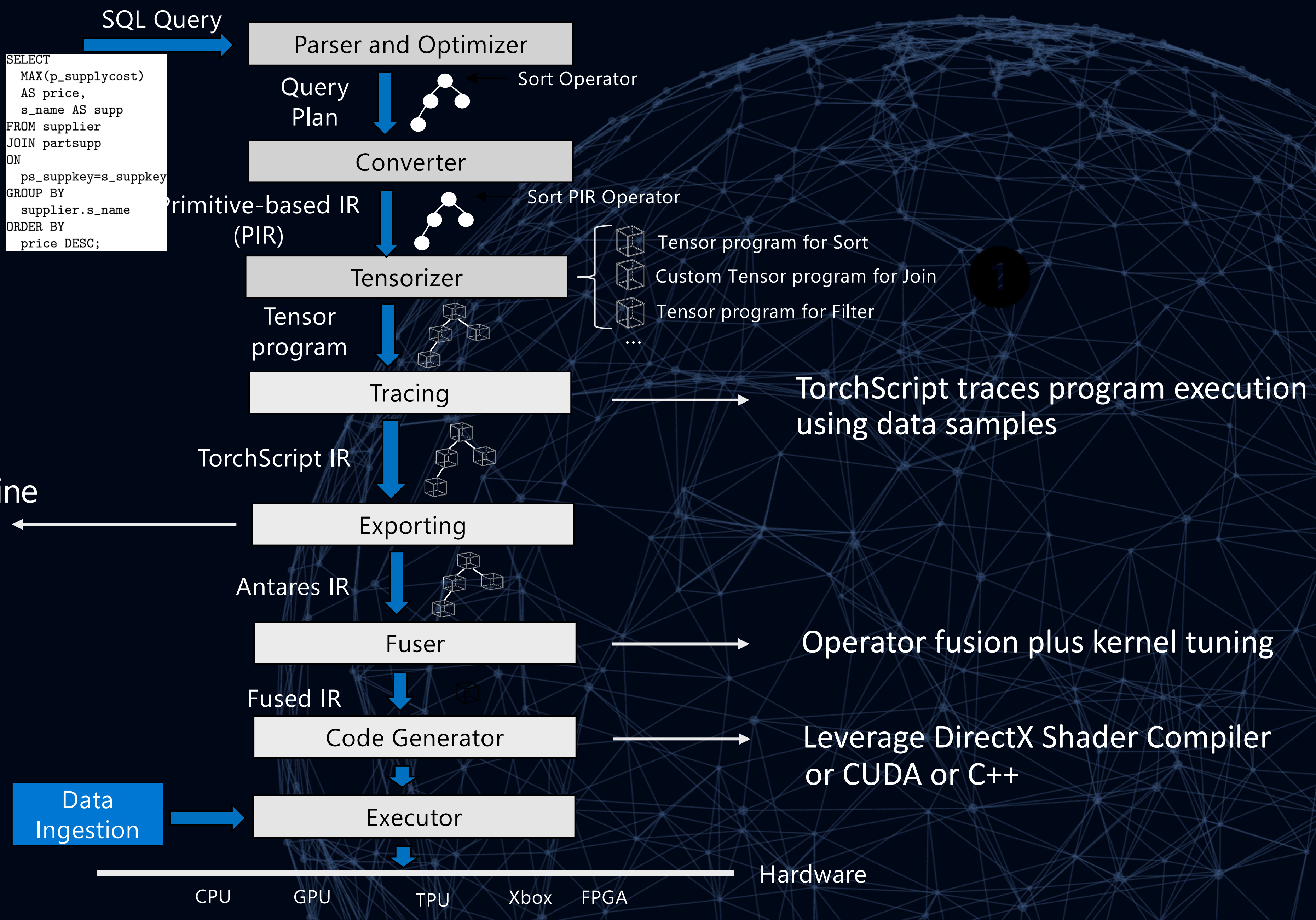
```
Rome:00,
Tokyo:01,
LosAngeles:10
```

RLE

```
00,0
01,4
10,7
00,11
```



Optimizing TQP: Fusion + Portable custom operators

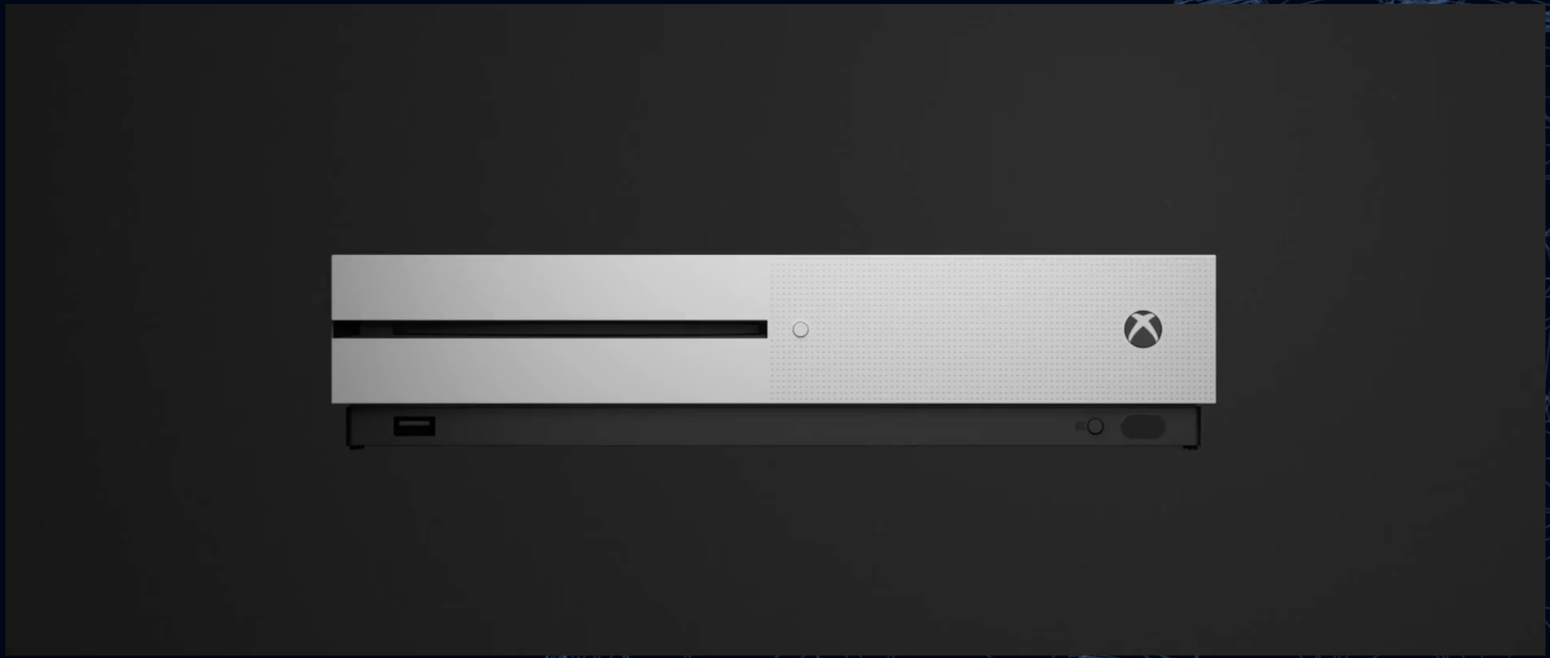


Antares is a Cross-compiling Engine

XBOX Intel GRAPHICS

https://github.com/microsoft/antares

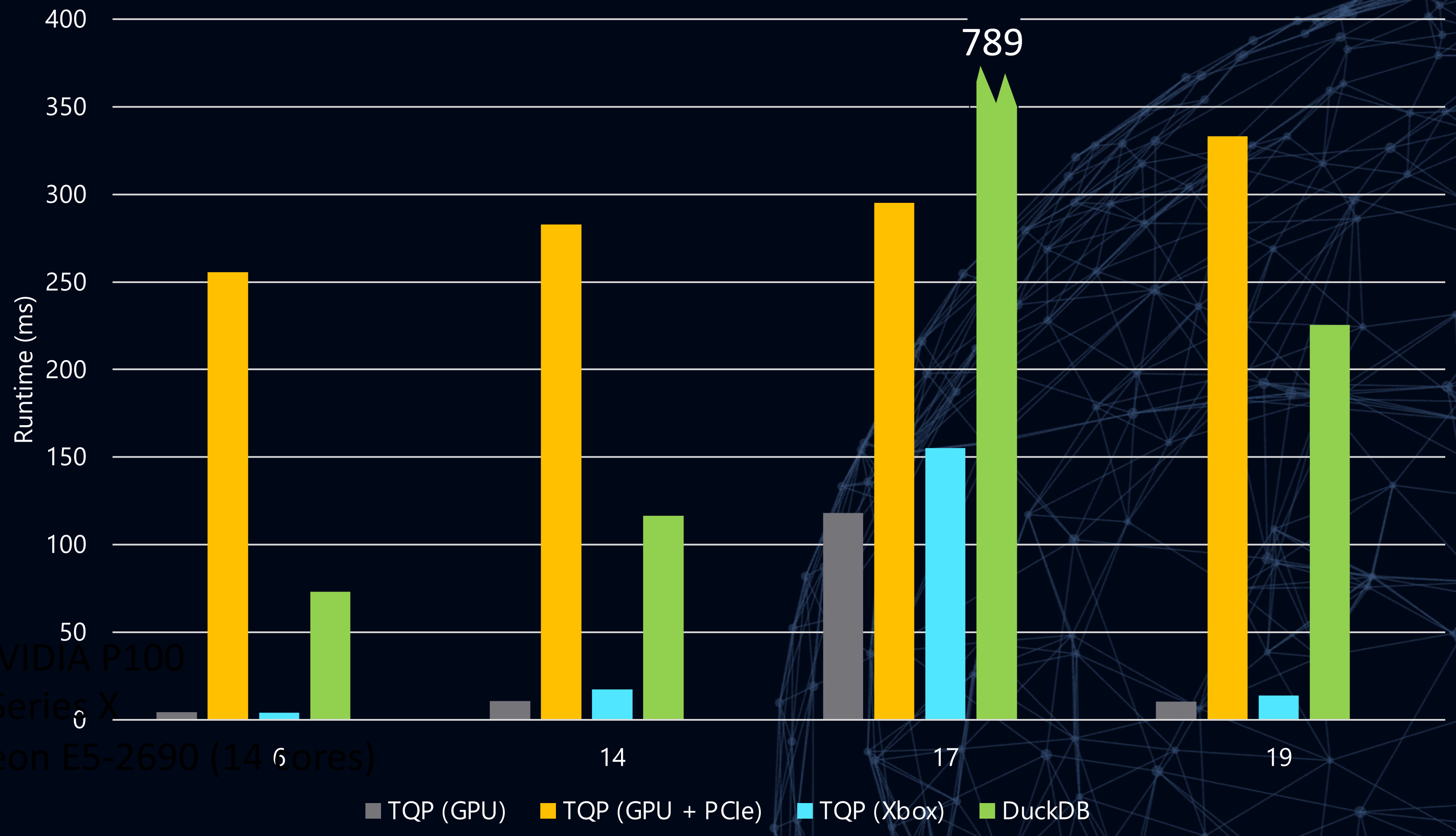
xCloud



Interesting HW configuration: APU design where CPU and GPU share HBM (no PCI-e)

Predictable usage pattern: gamers mostly play during the evenings (AKA dark time)

TQP on Xbox (SF 10, P100)



	Xeon E5-2690	P100	Xbox Series X
Memory Bandwidth (GB/s)	154	732	560
Unidirectional PCIv3 (GB/s)	-	16	-
Theoretical TFLOps	1.4	9.5	12.0

Pros and Cons of ”Tensor Query Processing”

Pros

- ✓ Leverage the massive investments in special HW
- ✓ Scalable Approach (tensor runtimes are getting ported to each new HW)
- ✓ Is this even possible?

Cons

- ? What about performance? (as compared with state-of-the-art)
- ? How expensive is it going to be? (engineering wise)

Pros and Cons of ”Tensor Query Processing”

Pros

- ✓ Leverage the massive investments in special HW
- ✓ Scalable Approach (tensor runtimes are getting ported to each new HW)
- ✓ Is this even possible?
- ✓ What about performance? (as compared with state-of-the-art)

Cons

- ❓ How expensive is it going to be? (engineering wise)

Pros and Cons of ”Tensor Query Processing”

Pros

✓ Leverage the massive investments in special HW

✓ Scalable Approach (tensor runtimes are getting ported to each new HW)

✓ Is this even possible?

✓ What about performance? (as compared with state-of-the-art)

Cons

❓ How expensive is it going to be? (engineering wise)

Pros and Cons of ”Tensor Query Processing”

Pros

- ✓ Leverage the massive investments in special HW
- ✓ Scalable Approach (tensor runtimes are getting ported to each new HW)
- ✓ Is this even possible?
- ✓ What about performance? (as compared with state-of-the-art)
- ✓ How expensive is it going to be? (engineering wise)

Less than 20k LoC

tpch_5 - Databricks

https://adb-727522010725101.1.azuredatabricks.net/?o=727522010725101#notebook/3328526773681...

Microsoft Azure | databricks | Search CTRL + P | tfPaperWsP1 | apurvagandhi@microsoft.com

tpch_5 Python

File Edit View Run Help Give feedba | Run all | Unknown | Schedule | Share

Compile the TPC-H 5 query

Cmd 8

```python
query = """select
                N_NAME,
                sum(L_EXTENDEDPRICE * (1 - L_DISCOUNT)) as revenue
            from
                customer,
                orders,
                lineitem,
                supplier,
                nation,
                region
            where
                C_CUSTKEY = o_custkey
                and L_ORDERKEY = O_ORDERKEY
                and l_suppkey = S_SUPPKEY
                and c_nationkey = s_nationkey
                and s_nationkey = n_nationkey
                and n_regionkey = r_regionkey
                and r_name = 'ASIA'
                and O_ORDERDATE >= date '1994-01-01'
                and O_ORDERDATE < date '1994-01-01' + interval '1' year
            group by
                N_NAME
            order by
                revenue desc"""
```

Cmd 9

```python
spark_query = sql_context.sql(query)
```

Cmd 10

Run!

demo-comparison.ipynb M | TensorBoard | demo-image-search.ipynb M

notebooks > sql > demo-comparison.ipynb > TQP > TQP Execution Profiling demo with Tensorboard > Set use_tensorboard=True a

+ Code + Markdown | Run All | Clear All Outputs | Restart | Variables | Outline | Python 3.8.10

TQP

Use Case: TPC-H 5 Query

```python
query = """select
                N_NAME,
                sum(L_EXTENDEDPRICE * (1 - L_DISCOUNT)) as revenue
            from
                customer,
                orders,
                lineitem,
                supplier,
                nation,
                region
            where
                C_CUSTKEY = o_custkey
                and L_ORDERKEY = O_ORDERKEY
                and l_suppkey = S_SUPPKEY
                and c_nationkey = s_nationkey
                and s_nationkey = n_nationkey
                and n_regionkey = r_regionkey
                and r_name = 'ASIA'
                and O_ORDERDATE >= date '1994-01-01'
                and O_ORDERDATE < date '1994-01-01' + interval '1' year
            group by
                N_NAME
            order by
                revenue desc"""
```

Python

Register the table (Dataframe whether in Spark on Pandas format)

```python
# Register lineitem as table in TQP.
```

Container container (elegant_bardeen) @ ... | mainter/tb_fix* | Cell 13 of 19

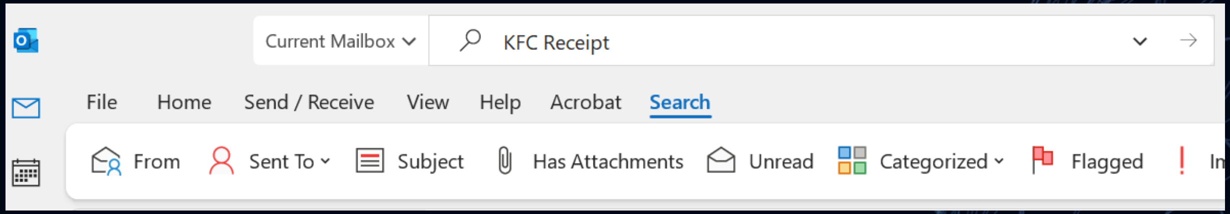
Future directions

1. Continue Perf Work (especially around I/O and multi-GPU)
2. Broader applicability (e.g., Classical ML with Hummingbird project*, pagerank)
3. Multimodal SQL (+ Differentiable SQL) for unstructured data input→ single (optimizable) tensor program!

* GitHub - microsoft/hummingbird: Hummingbird compiles trained ML models into tensor computation

Teaser: Multi-modal query support

Broader implications of having a DBMS co-existing with an ML runtime




```sql
SELECT
    input AS images,
    image_text_similarity_model("KFC Receipt", input) AS score
FROM attachments
ORDER BY score DESC
LIMIT 1
```

tpch_5

Compile the TPC-H 5 query

Cmd 8

```python
query = """select
                N_NAME,
                sum(L_EXTENDEDPRICE * (1 - L_DISCOUNT)) as revenue
            from
                customer,
                orders,
                lineitem,
                supplier,
                nation,
                region
            where
                C_CUSTKEY = o_custkey
                and L_ORDERKEY = O_ORDERKEY
                and l_suppkey = S_SUPPKEY
                and c_nationkey = s_nationkey
                and s_nationkey = n_nationkey
                and n_regionkey = r_regionkey
                and r_name = 'ASIA'
                and O_ORDERDATE >= date '1994-01-01'
                and O_ORDERDATE < date '1994-01-01' + interval '1' year
            group by
                N_NAME
            order by
                revenue desc"""
```

Cmd 9

```python
spark_query = sql_context.sql(query)
```

Cmd 10

Run!

TQP

Use Case: TPC-H 5 Query

```python
query = """select
                N_NAME,
                sum(L_EXTENDEDPRICE * (1 - L_DISCOUNT)) as revenue
            from
                customer,
                orders,
                lineitem,
                supplier,
                nation,
                region
            where
                C_CUSTKEY = o_custkey
                and L_ORDERKEY = O_ORDERKEY
                and l_suppkey = S_SUPPKEY
                and c_nationkey = s_nationkey
                and s_nationkey = n_nationkey
                and n_regionkey = r_regionkey
                and r_name = 'ASIA'
                and O_ORDERDATE >= date '1994-01-01'
                and O_ORDERDATE < date '1994-01-01' + interval '1' year
            group by
                N_NAME
            order by
                revenue desc"""
```

Register the table (Dataframe whether in Spark on Pandas format)

```python
# Register lineitem as table in TQP.
```

Conclusion

Save humanity from Skynet!

How? Keep it busy running SQL (compiled to Tensors)

- Free-ride on AI investments
- Great perf/cost trade-offs
- Fun future directions

Microsoft Azure Data
Gray Systems Lab

Join us… let's invent
the future together!

https://aka.ms/gsl

Gray Systems Lab (GSL)

GSL is an applied and embedded research group, comprised of Data-Scientists, Engineers and Researchers.