USENIX ATC'20
2020 USENIX Annual Technical Conference
JULY 15–17, 2020

FineStream: Fine-Grained Window-Based Stream Processing on CPU-GPU Integrated Architectures

Feng Zhang, Lin Yang, Shuhao Zhang, Bingsheng He, Wei Lu, Xiaoyong Du

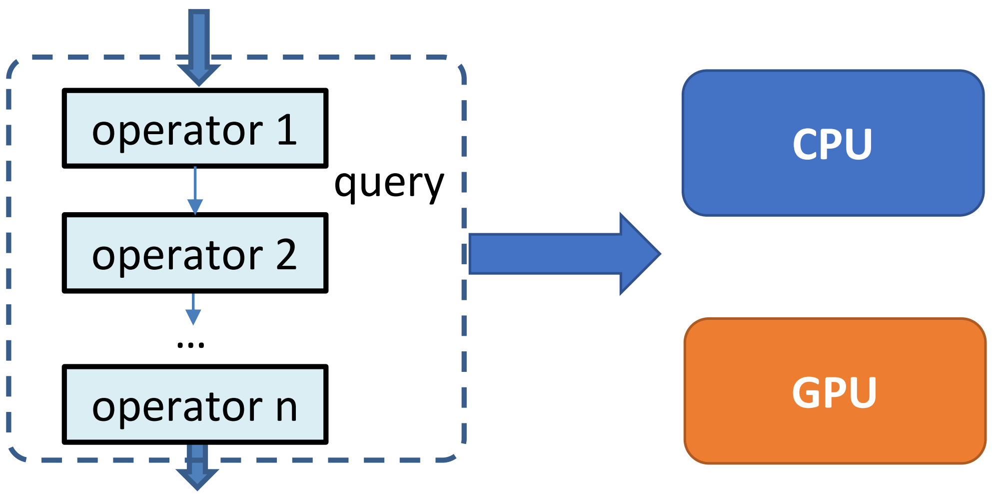
Renmin University of China
Technische Universität Berlin
National University of Singapore

Outline

1. **Background**
2. Motivation
3. Challenges
4. FineStream
5. Evaluation
6. Conclusion

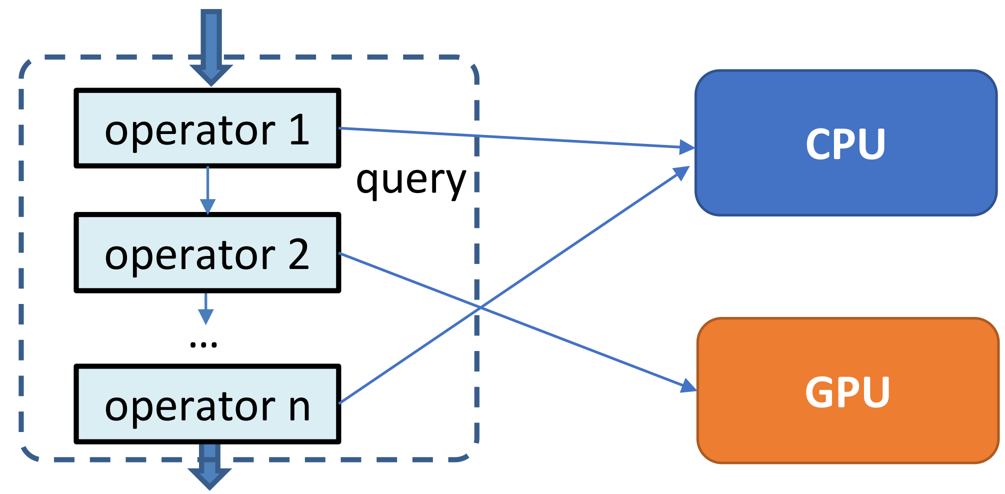
1. Background

- Bulk-synchronous parallel model
  - query granularity



[SIGMOD'16] **Saber**: Window-based hybrid stream processing for heterogeneous architectures

- Continuous operator model
  - operator granularity

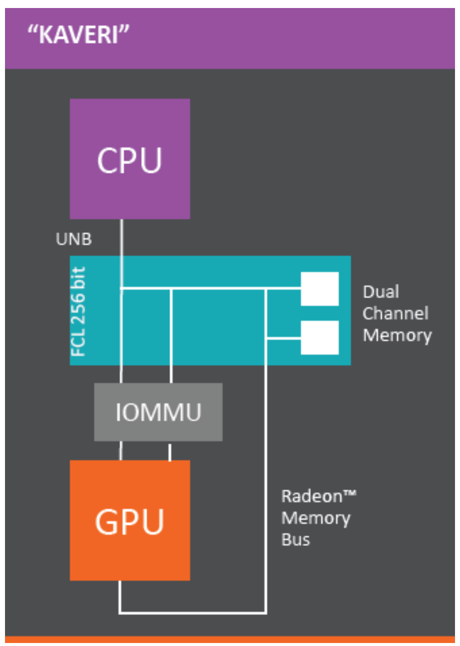


This paper

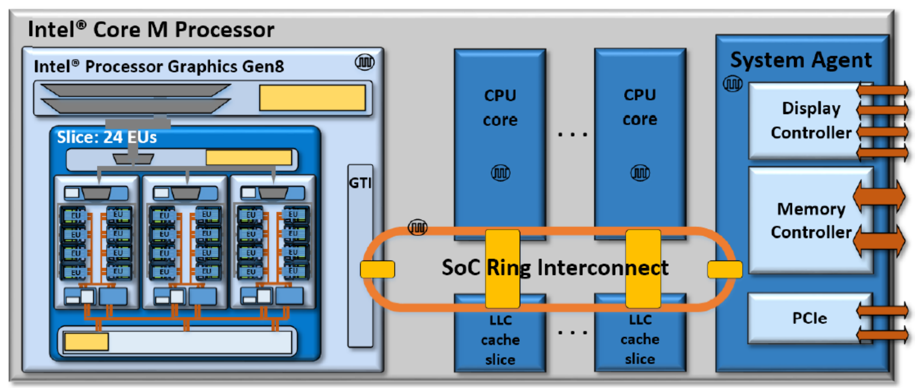
CPU and GPU can concurrently execute in both cases — only the granularity is different.

2. Integrated Architectures

- 2011, Jan
- AMD APU



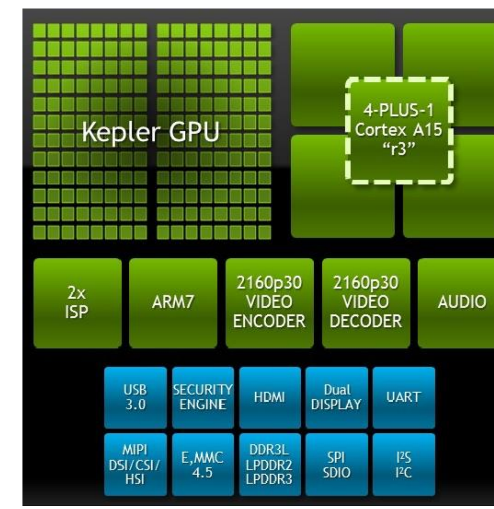
- 2012, Jan
- **Intel Ivy Bridge**



Benefits

- No PCI-e transfer overhead
- Shared global memory
- High energy efficiency

- 2014, Apr
- **Nvidia Tegra**



1. Background

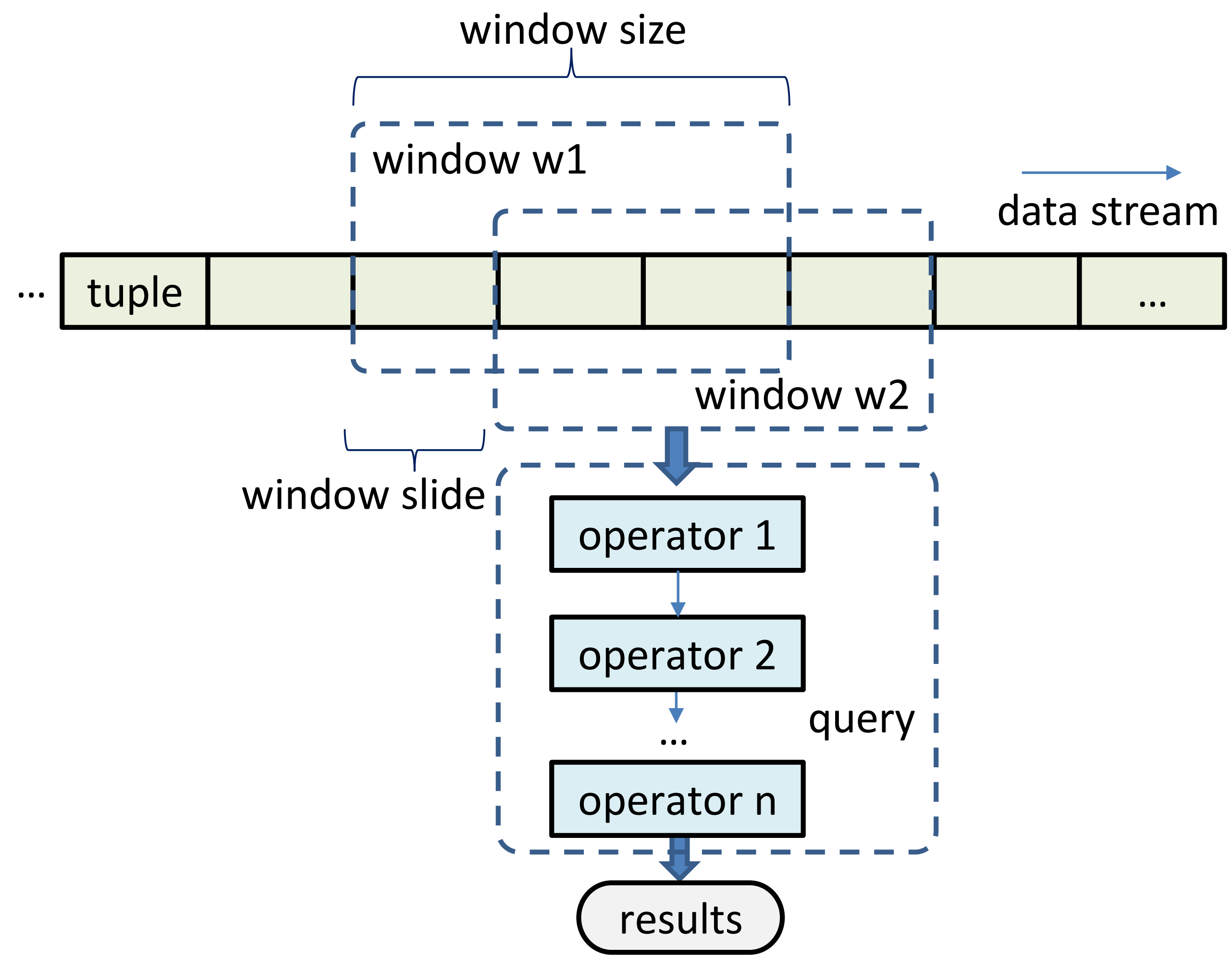
- Integrated architectures vs. discrete architectures

	Integrated architectures		Discrete architectures	
Architecture	A10-7850K	Ryzen5 2400G	GTX 1080Ti	V100
#cores	512+4	704+4	**3584**	**5120**
TFLOPS	0.9	1.7	**11.3**	**14.1**
bandwidth (GB/s)	25.6	38.4	484.4	900
price ($)	**209**	**169**	1100	8999
TDP (W)	**95**	**65**	250	300

3. Stream Processing with SQL

- Data stream
- Window
- Operator
- Query

* Batch

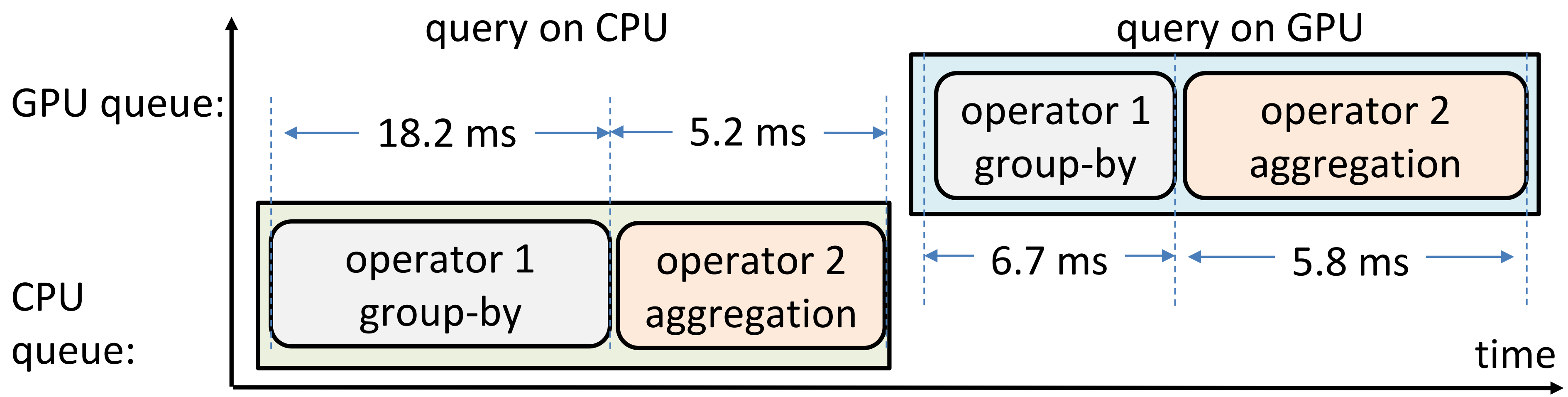


Outline

1. Background
2. **Motivation**
3. Challenges
4. FineStream
5. Evaluation
6. Conclusion

2. Motivation

- Varying Operator-Device Preference



2. Motivation

- Performance (tuples/s) of operators on the CPU and the GPU of the integrated architecture.

Operator	CPU only	GPU only	Device choice
Projection	14.2	**14.3**	GPU
Selection	13.1	**14.1**	GPU
Aggregation	**14.7**	13.5	CPU
Group-by	8.1	**12.4**	GPU
Join	**0.7**	0.1	CPU

2. Motivation

- Fine-Grained Stream Processing
  - A fine-grained stream processing method that can consider both **integrated architecture characteristics** and **operator features** shall have better performance.
    - memory bandwidth limit
    - operators - preferred devices

Key Idea!

☺ CPU and GPU have good performance

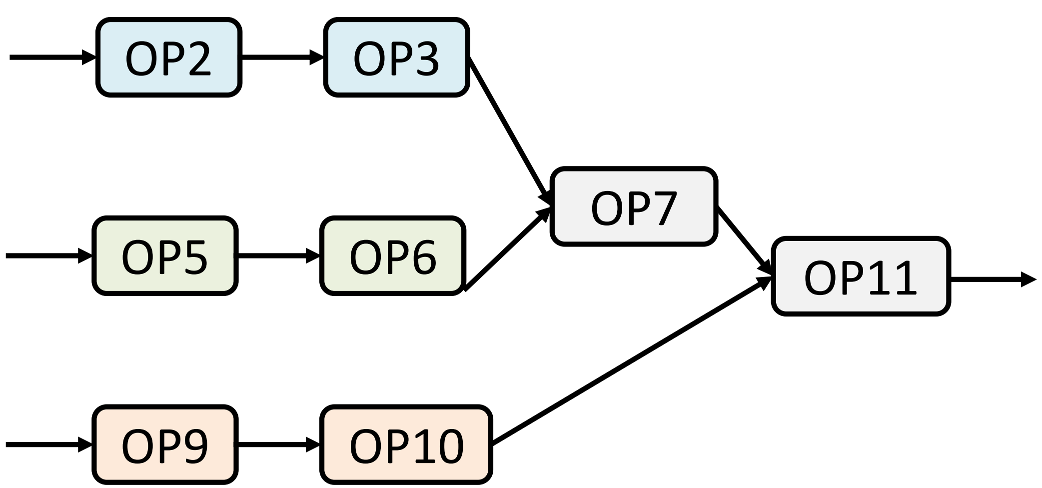
☺ consider the interplay of operator features and architecture difference.

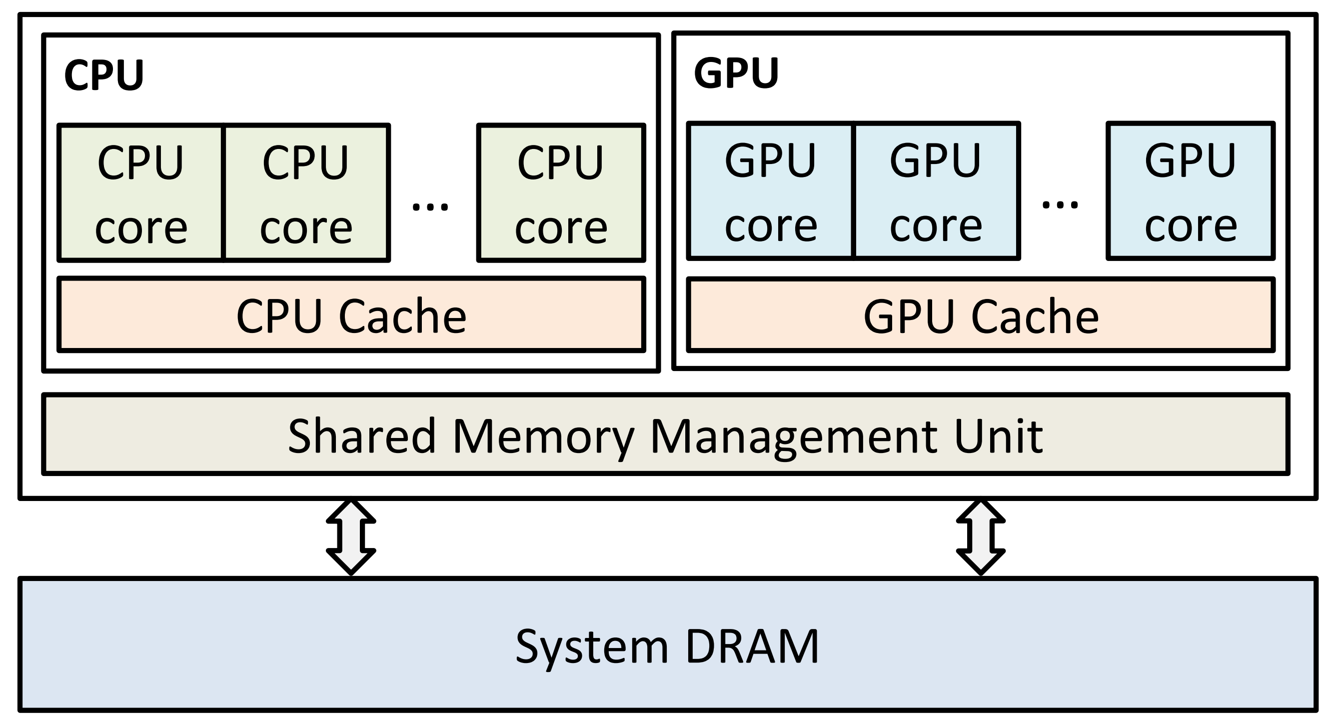
Outline

1. Background
2. Motivation
3. **Challenges**
4. FineStream
5. Evaluation
6. Conclusion

3. Challenges

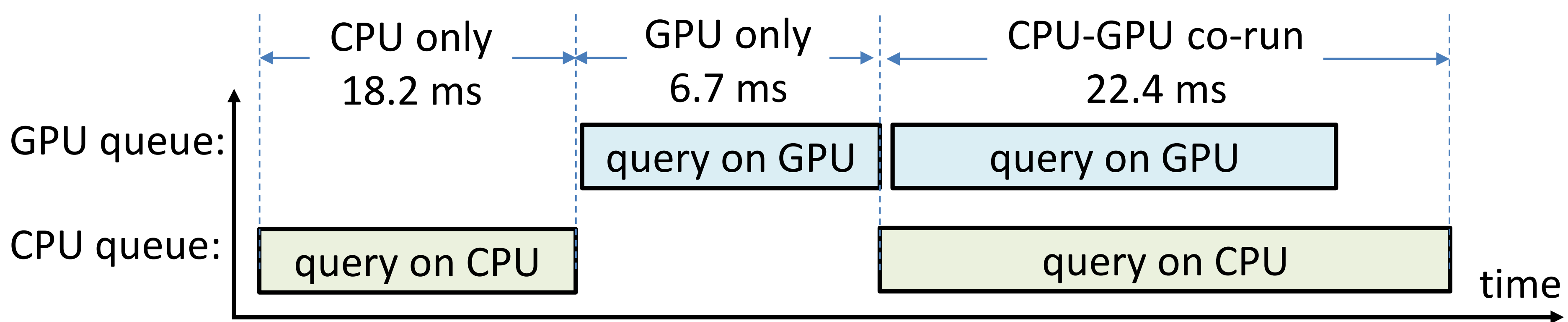
- Challenge 1: Application topology combined with architectural characteristics





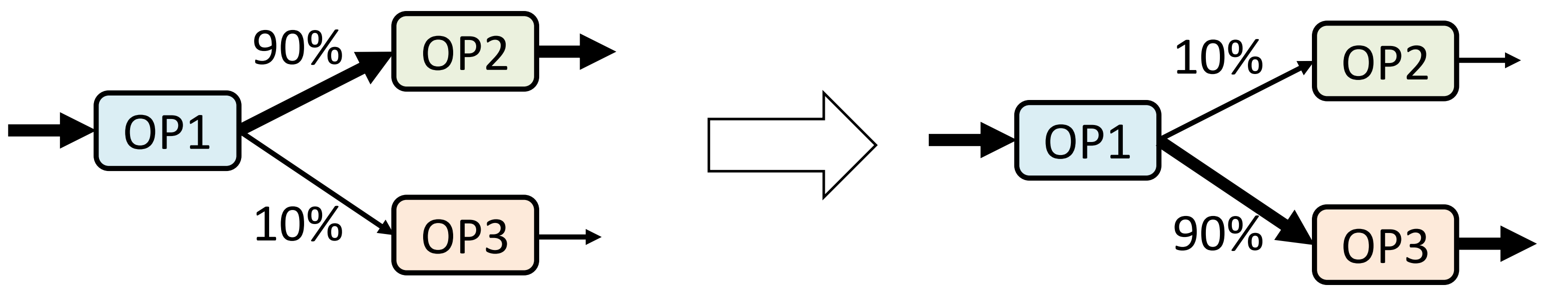
3. Challenges

- Challenge 2: SQL query plan optimization with shared main memory



3. Challenges

- Challenge 3: Adjustment for dynamic workload

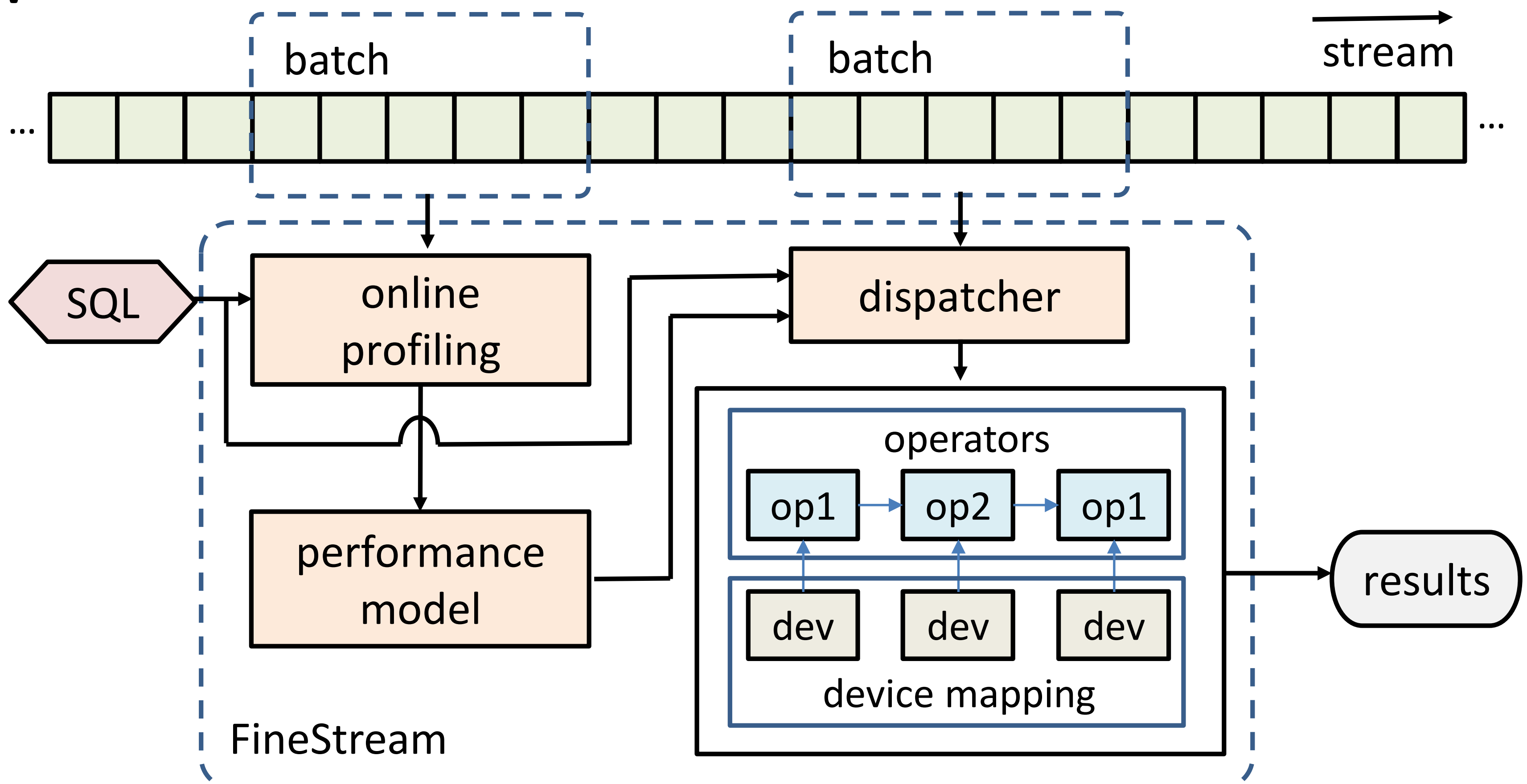


Outline

1. Background
2. Motivation
3. Challenges
4. **FineStream**
5. Evaluation
6. Conclusion

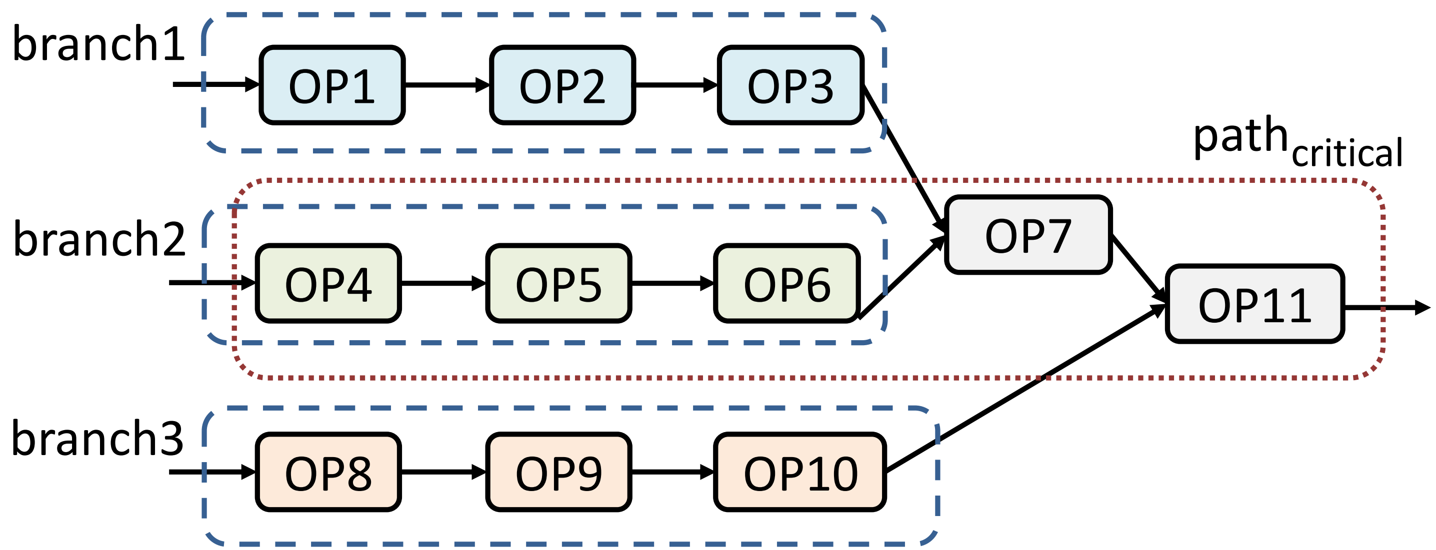
4. FineStream

- Overview



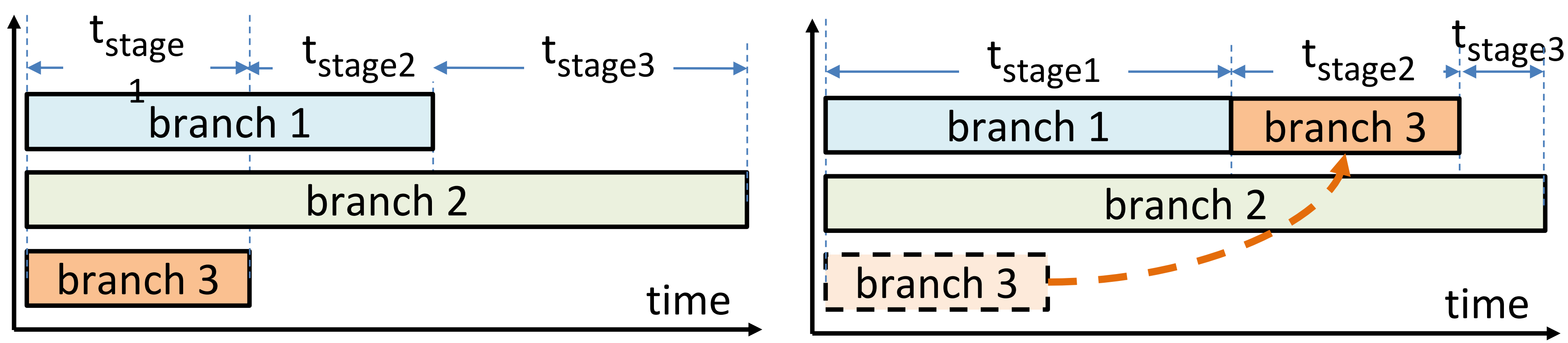
4. FineStream

- Topology



4. FineStream

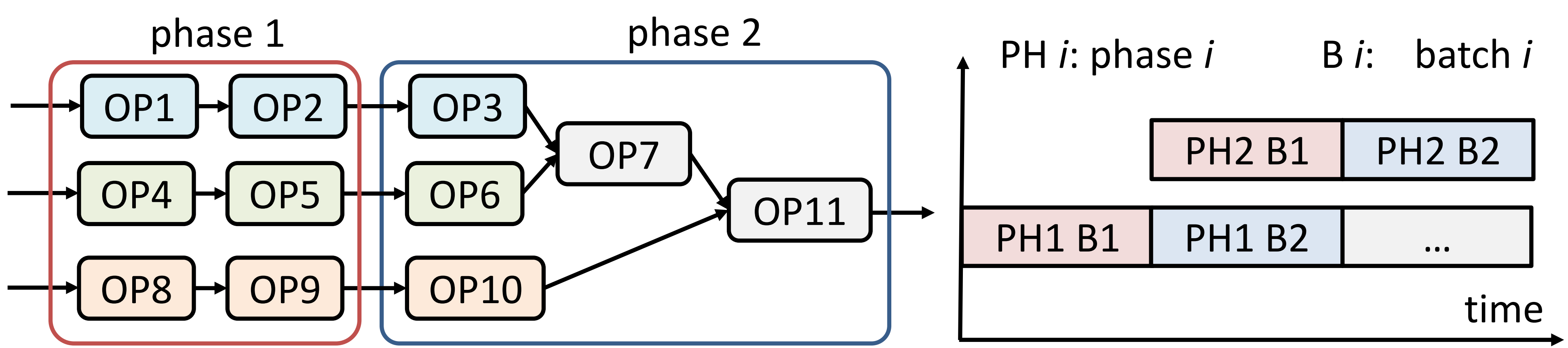
- Optimization 1: Branch Co-Running



(a) Branch parallelism. (b) Branch scheduling optimization.

4. FineStream

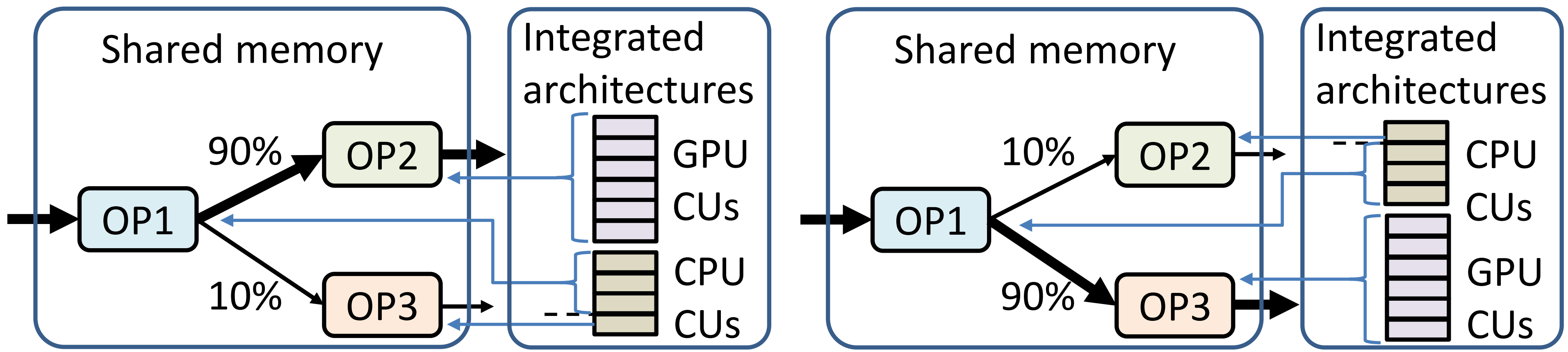
- Optimization 2: Batch Pipeline



(a) Phase partitioning. (b) Batch pipeline.

4. FineStream

- Optimization 3: Handling Dynamic Workload
  - Light-Weight Resource Reallocation

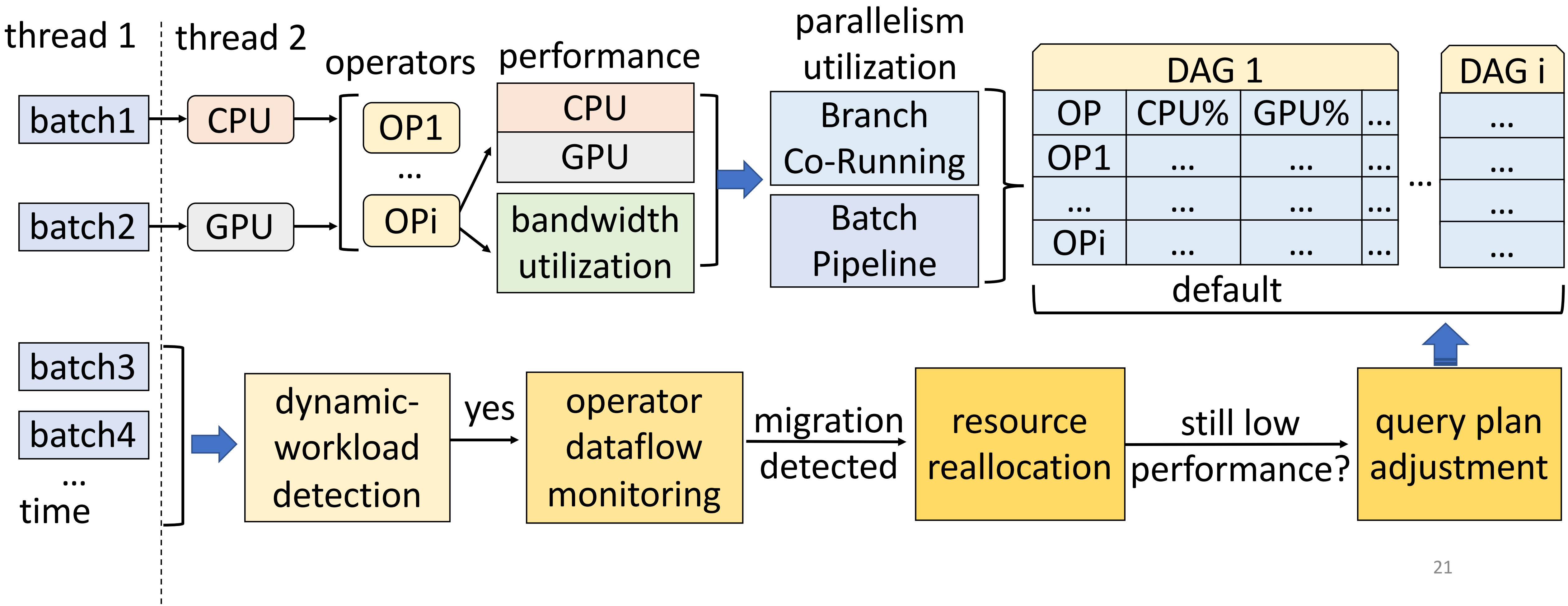


(a) 90% workload goes to OP2. (b) 90% workload goes to OP3.

  - Query Plan Adjustment

4. FineStream

- Execution flow



Outline

1. Background
2. Motivation
3. Challenges
4. FineStream
5. **Evaluation**
6. Conclusion

5. Evaluation

- Platforms
  - AMD A10- 7850K
  - Ryzen 5 2400G
- Datasets
  - Google compute cluster monitoring
  - Anomaly detection in smart grids
  - Linear road benchmark
  - Synthetically generated dataset
- Benchmarks
  - Nine queries

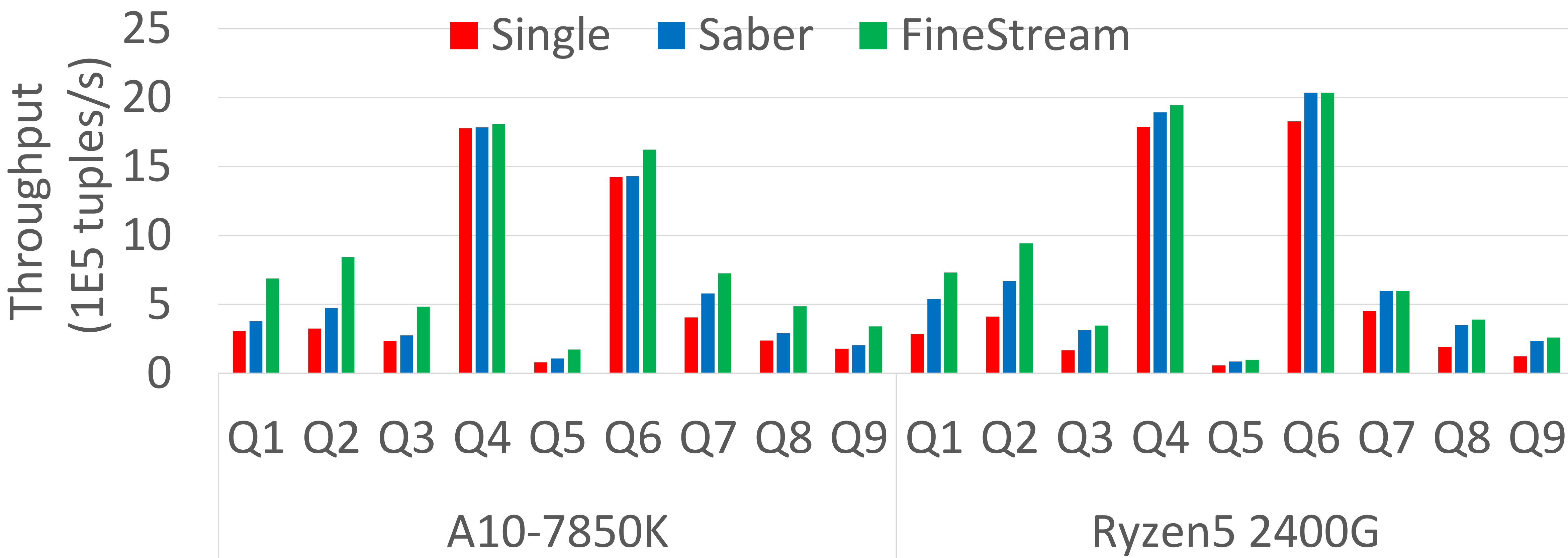
Example - Q1

(Google compute cluster monitoring)

select timestamp, category, **sum**(cpu)
as totalCPU
from TaskEvents [**range** 256 **slide** 1]
group by category

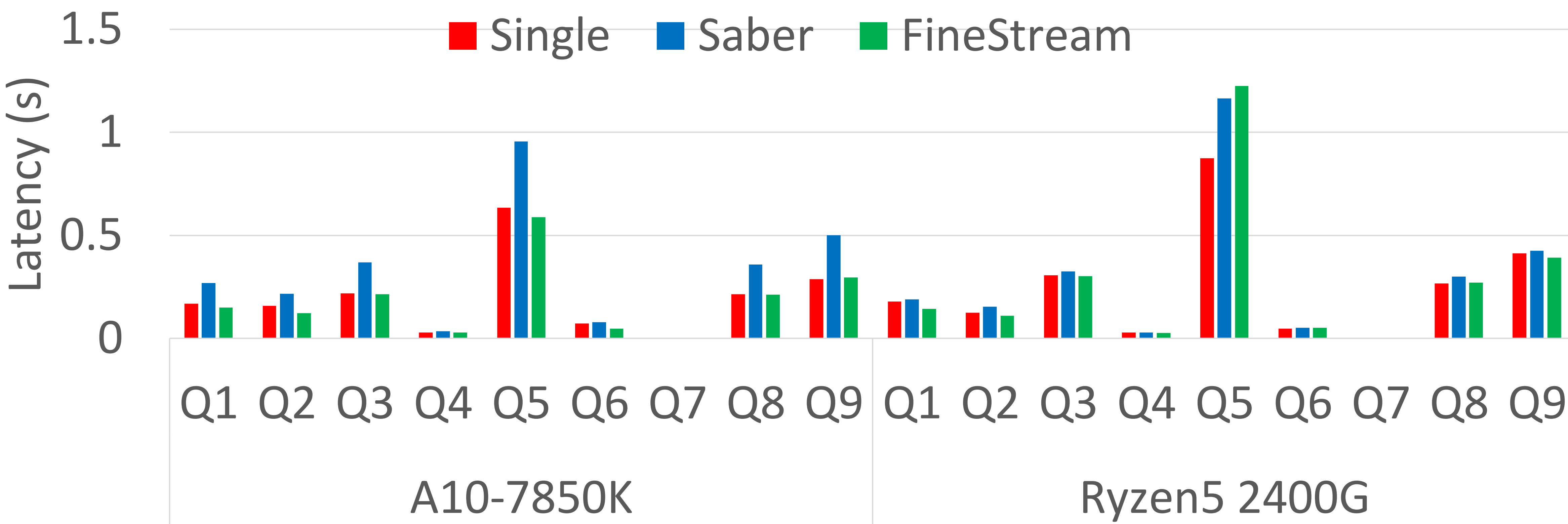
5. Evaluation

- Throughput: FineStream achieves the best performance in most cases.



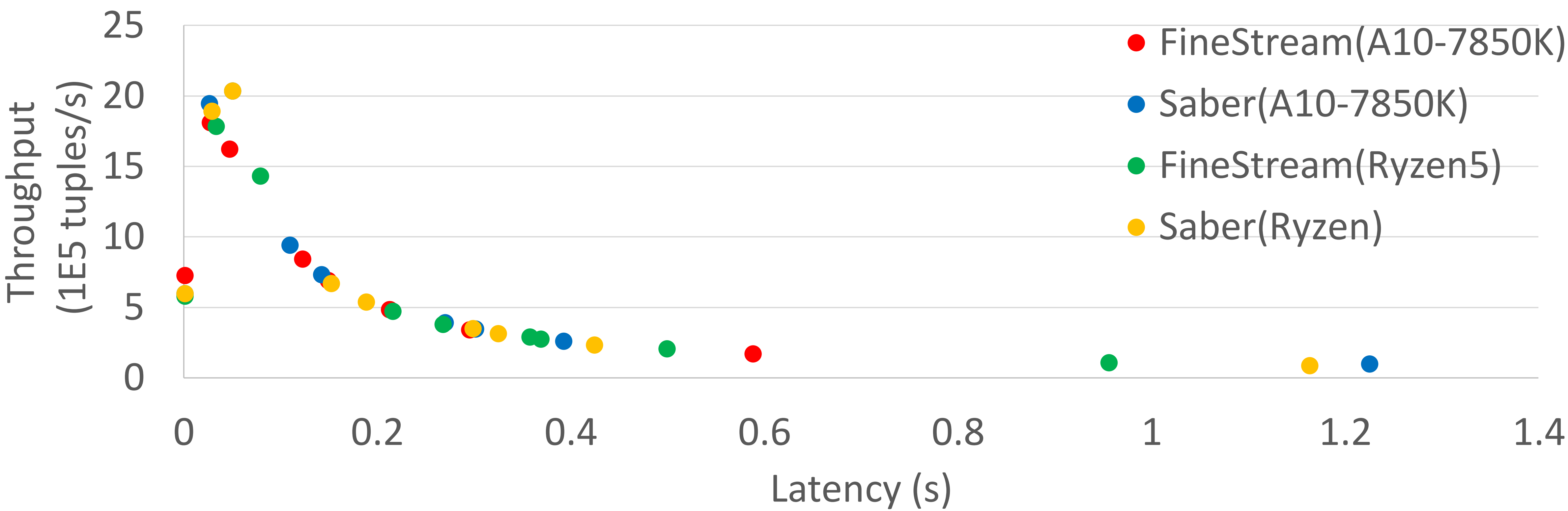
5. Evaluation

- Latency: Low latency in most cases.



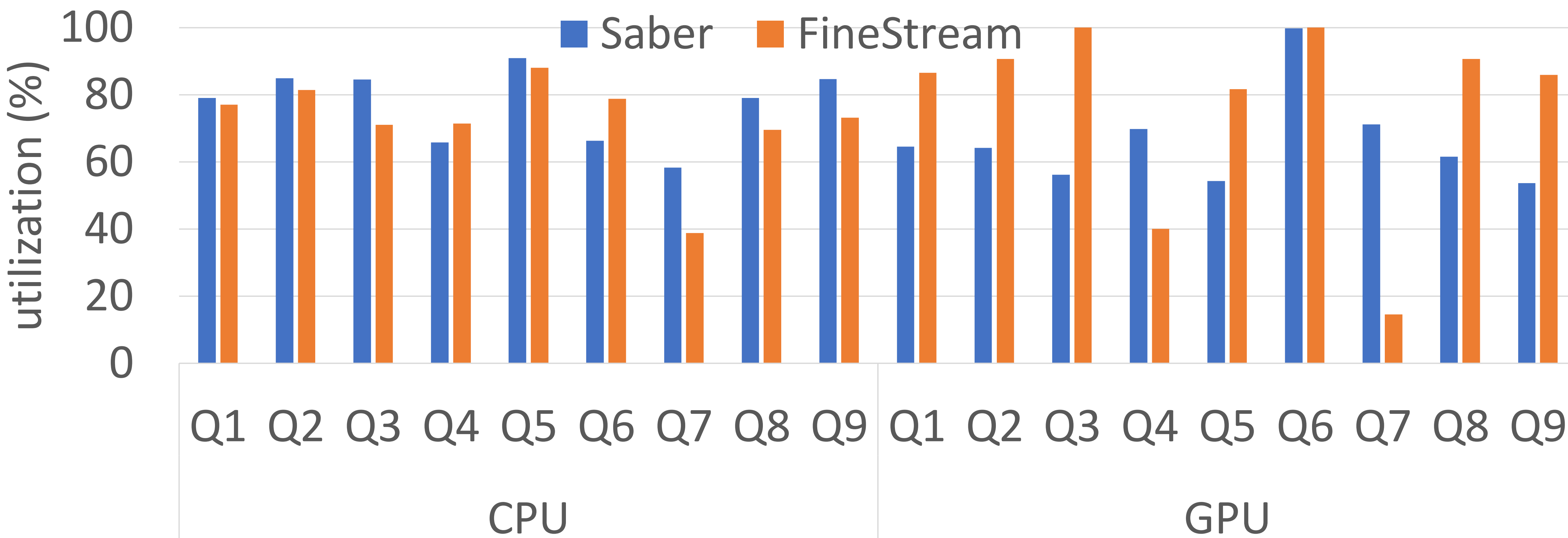
5. Evaluation

- Throughput vs. latency
  - Queries with high throughput usually have low latency, and vice versa.



5. Evaluation

- Utilization
  - FineStream utilizes the GPU device better on the integrated architecture.

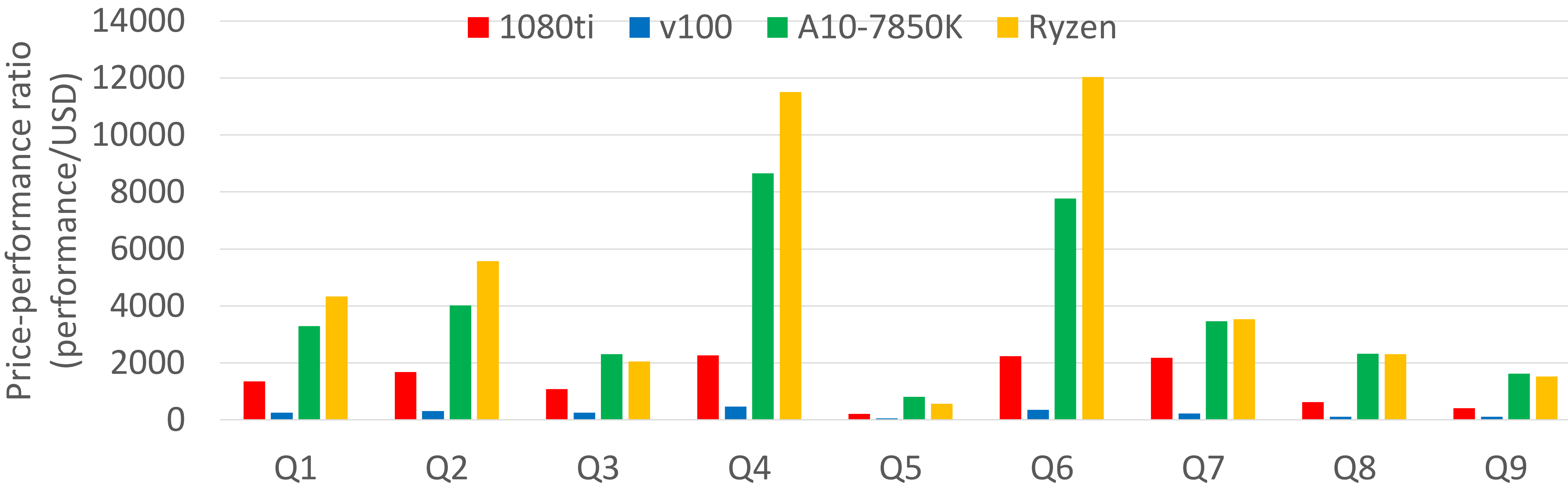


5. Evaluation

- Comparison with Discrete Architectures
  - Throughput: The discrete GPUs exhibit 1.8x to 5.7x higher throughput than the integrated architectures, due to the more computational power of discrete GPUs.
  - Latency:
    - Discrete GPUs: Ttotal = TPCIe_transmit + Tcompute
    - Integrated GPUs: Ttotal = Tcompute

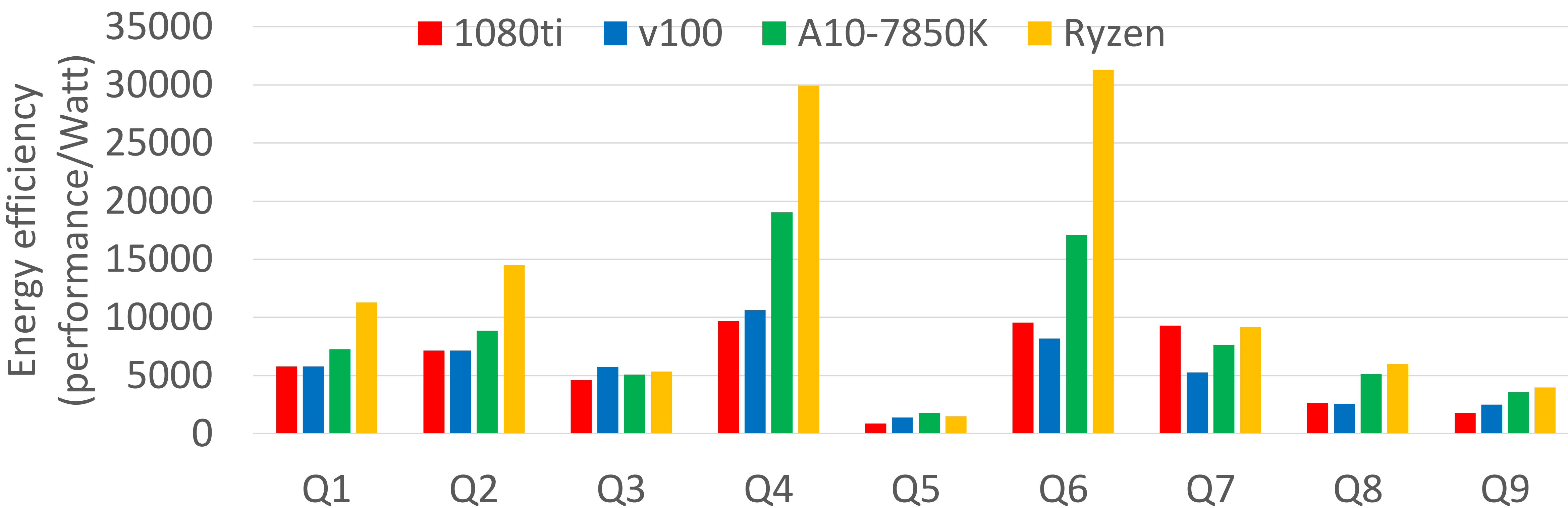
5. Evaluation

- Comparison with Discrete Architectures
  - High Price-Throughput Ratio



5. Evaluation

- Comparison with Discrete Architectures
  - High Energy Efficiency



Outline

1. Background
2. Motivation
3. Challenges
4. FineStream
5. Evaluation
6. **Conclusion**

6. Conclusion

- The first fine-grained window-based relational stream processing.
- Lightweight query plan adaptations handling dynamic workloads.
- FineStream evaluation on a set of stream queries.

Thank you!

Feng Zhang, Lin Yang, Shuhao Zhang, Bingsheng He, Wei Lu, Xiaoyong Du
Renmin University of China, Technische Universität Berlin, National University of Singapore

fengzhang@ruc.edu.cn, yanglin2330@ruc.edu.cn, shuhao.zhang@tu-berlin.de,
hebs@comp.nus.edu.sg, lu-wei@ruc.edu.cn, duyong@ruc.edu.cn