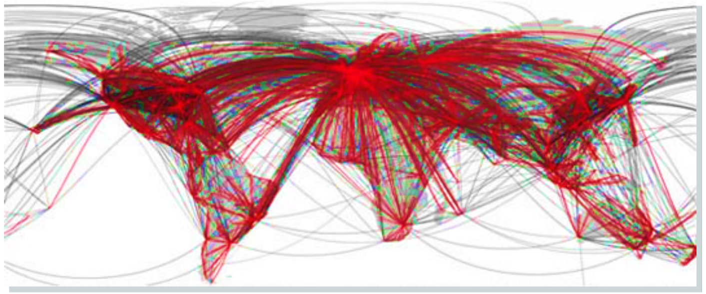
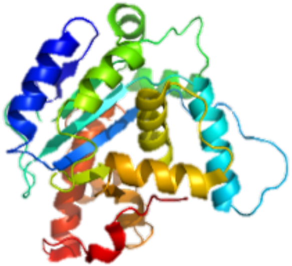
Fast and Concurrent RDF Queries using *RDMA*-*assisted* GPU Graph Exploration

Siyuan Wang, Chang Lou, **Rong Chen**, Haibo Chen

Institute of Parallel and Distributed Systems (IPADS)
Shanghai Jiao Tong University
http://ipads.se.sjtu.edu.cn

Graphs are Everywhere





Online ***graph query*** plays a vital role for searching, mining and reasoning linked data

photo: whatsnextblog.com

Jena

TAO

Unicorn

RDF and SPARQL

Resource Description Framework (**RDF**)

- Representing linked data on the Web
- Public knowledge bases: DBpedia, Wikidata, PubChemRDF
- Google's knowledge graph

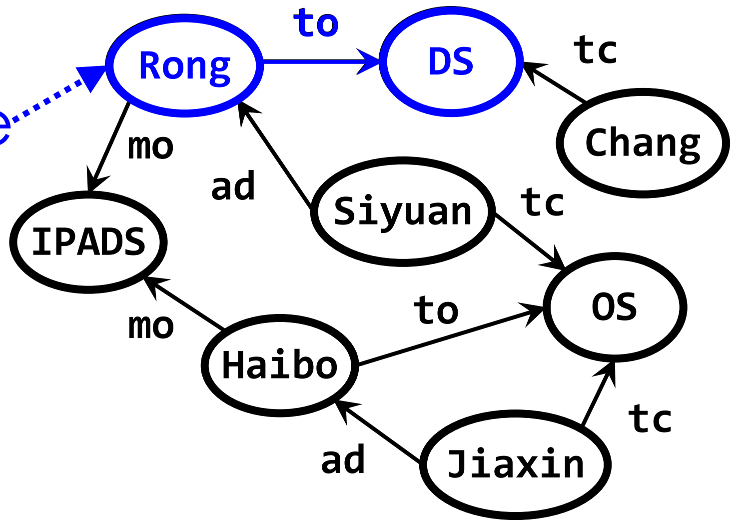
RDF and SPARQL

mo: MemberOf
ad: ADvisor
to: TeacherOf
tc: TakeCourse

RDF is a graph composed by a set of ⟨**S**ubject, **P**redicate, **O**bject⟩ triples

Rong	to	DS
Rong	mo	IPADS
Siyuan	ad	Rong
Siyuan	tc	OS
Haibo	to	OS
Haibo	mo	IPADS
Jiaxin	ad	Haibo
. . .		

triple



RDF and SPARQL

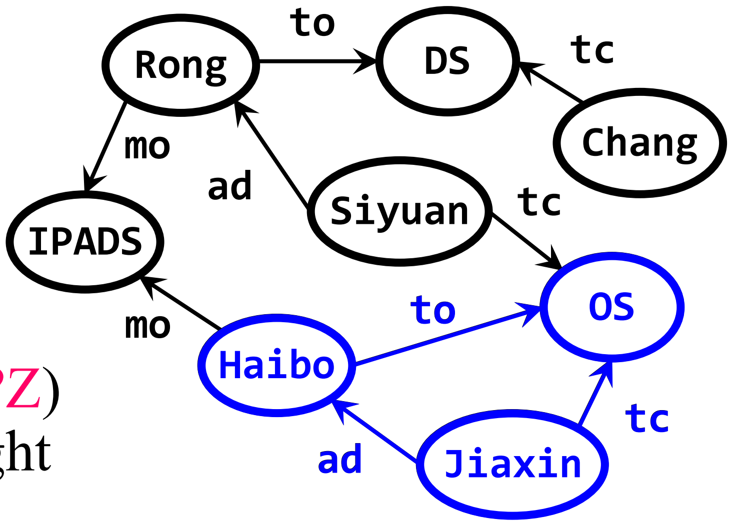
SPARQL is standard query language for RDF

Triple Pattern

```
      SELECT ?X ?Y ?Z WHERE {
TP1     ?X teacherof ?Y .
TP2     ?Z takecourse ?Y .
TP3     ?Z adivsor ?X .
      }
```

Variable

Professor (?X) advises (**ad**) student (?Z) who also takes (**tc**) a course (?Y) taught by (**tc**) the professor

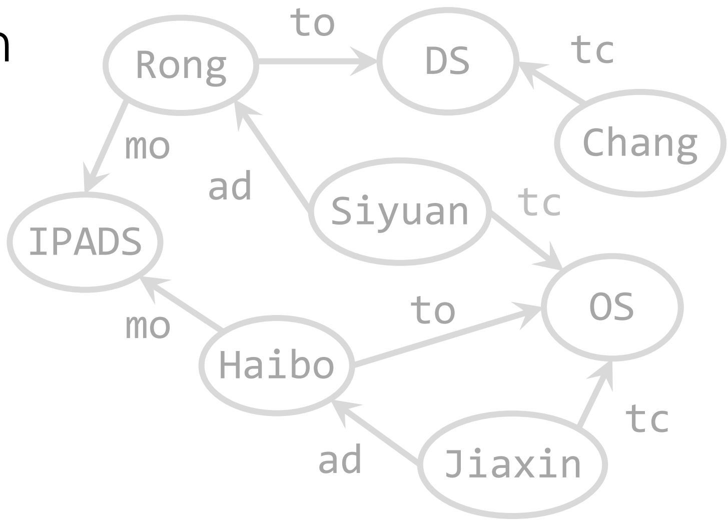


Queries are Heterogeneous

```
SELECT ?X ?Y ?Z WHERE {
 ?X teacherof ?Y .
 ?Z takecourse ?Y .
 ?Z adivsor ?X .       }
```

Heavy Query (Q_H)

- Start from a set of vertices
- Explore a large part of graph



Light Query (Q_L)

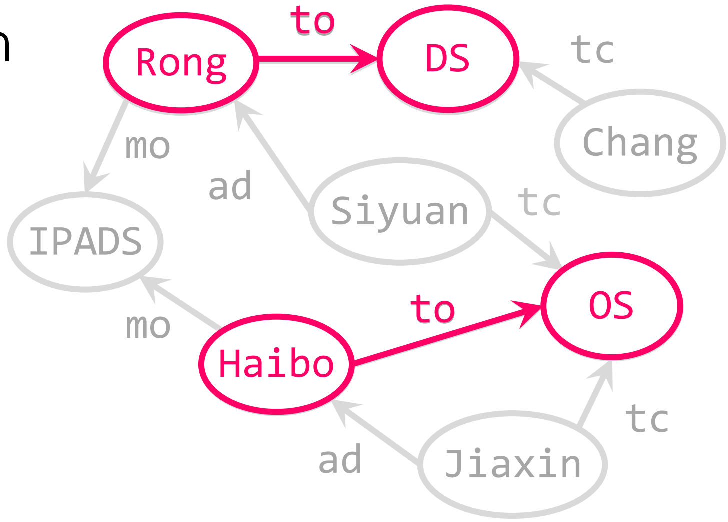
Queries are Heterogeneous

Heavy Query (Q_H)

- Start from a set of vertices
- Explore a large part of graph

Light Query (Q_L)

```
     SELECT ?X ?Y ?Z WHERE {
TP1   ?X teacherof ?Y .
      ?Z takecourse ?Y .
      ?Z adivsor ?X .      }
```

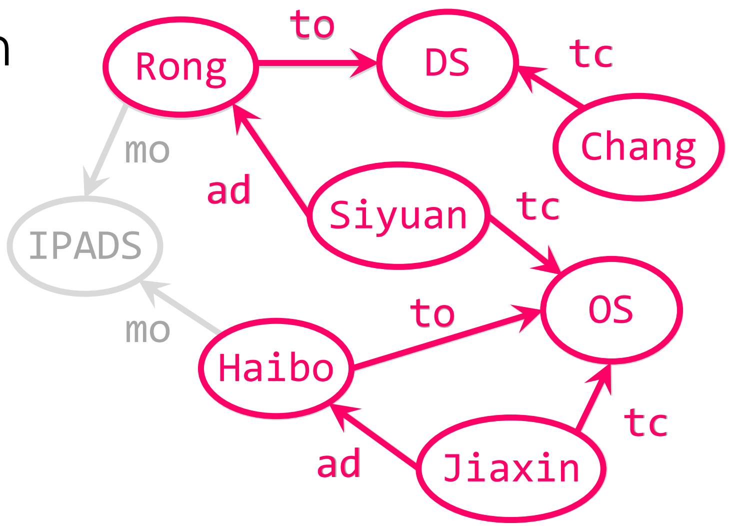


Queries are Heterogeneous

```
SELECT ?X ?Y ?Z WHERE {
TP1  ?X teacherof ?Y .
TP2  ?Z takecourse ?Y .
TP3  ?Z adivsor ?X .      }
```

Heavy Query (Q_H)

- Start from a set of vertices
- Explore a large part of graph



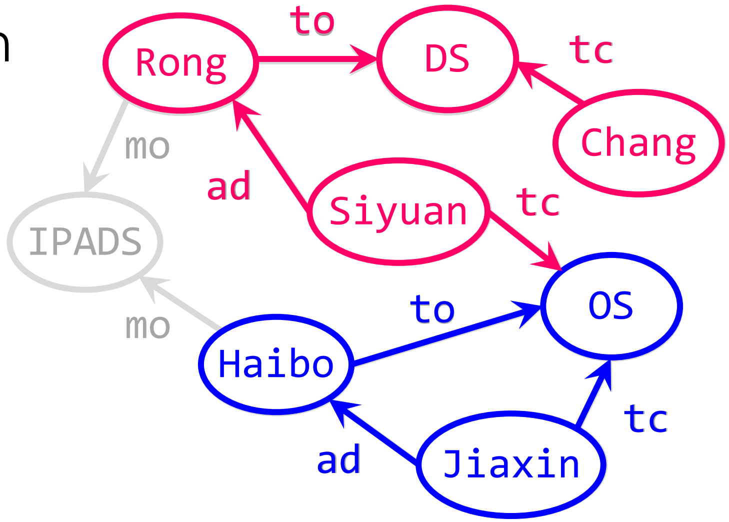
Light Query (Q_L)

Queries are Heterogeneous

```
SELECT ?X ?Y ?Z WHERE {
TP1  ?X teacherof ?Y .
TP2  ?Z takecourse ?Y .
TP3  ?Z adivsor ?X .      }
```

Heavy Query (Q_H)

- Start from a set of vertices
- Explore a large part of graph



Light Query (Q_L)

Queries are Heterogeneous

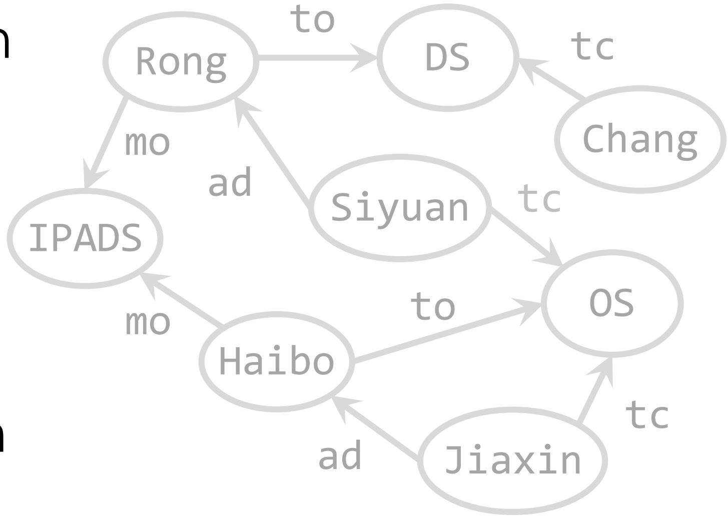
```
SELECT ?X WHERE {
 ?X advisor Rong .
 ?X takecourse OS . }
```

Heavy Query (Q_H)

- Start from a set of vertices
- Explore a large part of graph

Light Query (Q_L)

- Start from a given vertex
- Explore a small part of graph

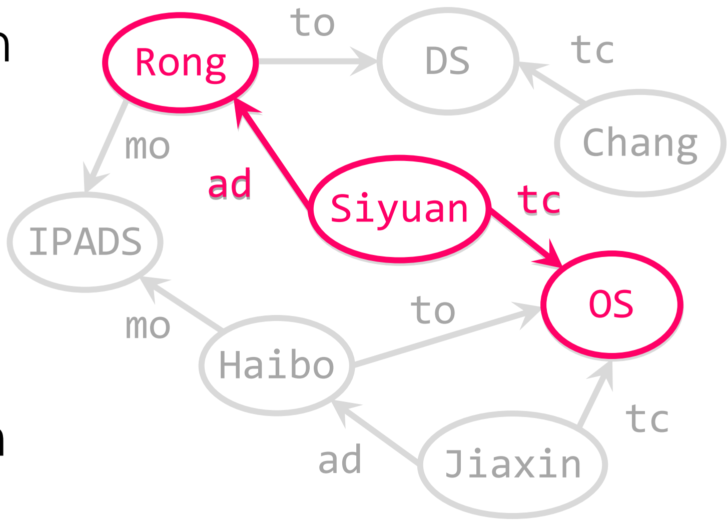


Queries are Heterogeneous

```
SELECT ?X WHERE {
 ?X advisor Rong .
 ?X takecourse OS . }
```

Heavy Query (Q_H)

- Start from a set of vertices
- Explore a large part of graph



Light Query (Q_L)

- Start from a given vertex
- Explore a small part of graph

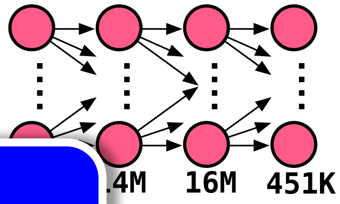
Queries are Heterogeneous

Heavy Query (Q_H)

- Start from a set of vertices
- Expl

Q7*

390 ms



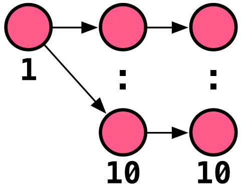
Incompetent to handle heavy queries efficiently

Light Query (Q_L)

- Start from a given vertex
- Explore a small part of graph

Q5*

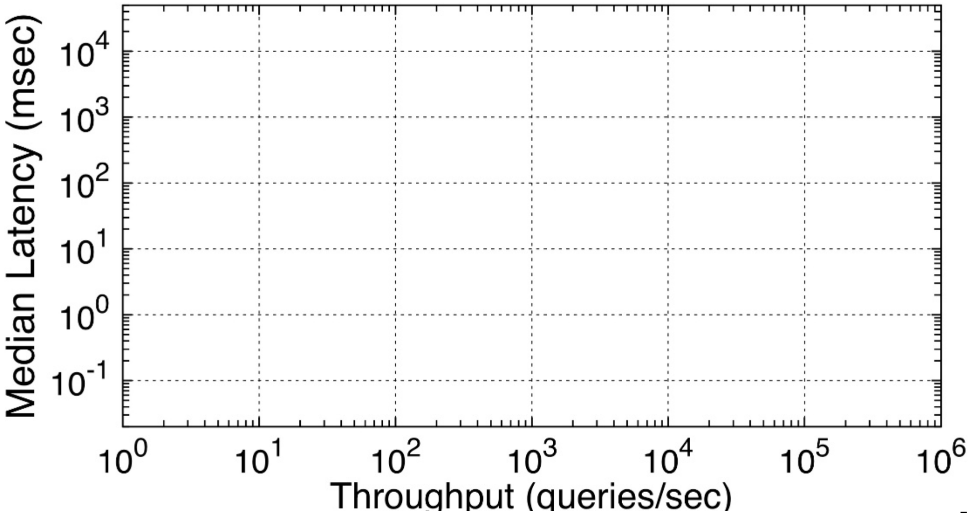
0.13 ms

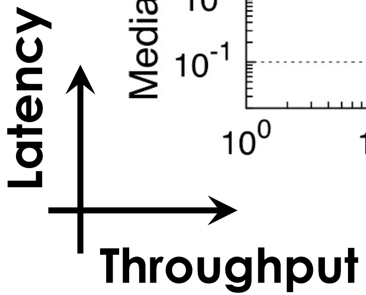


* Wukong on a 10-server cluster for LUBM-10240 dataset

Concurrent Workload

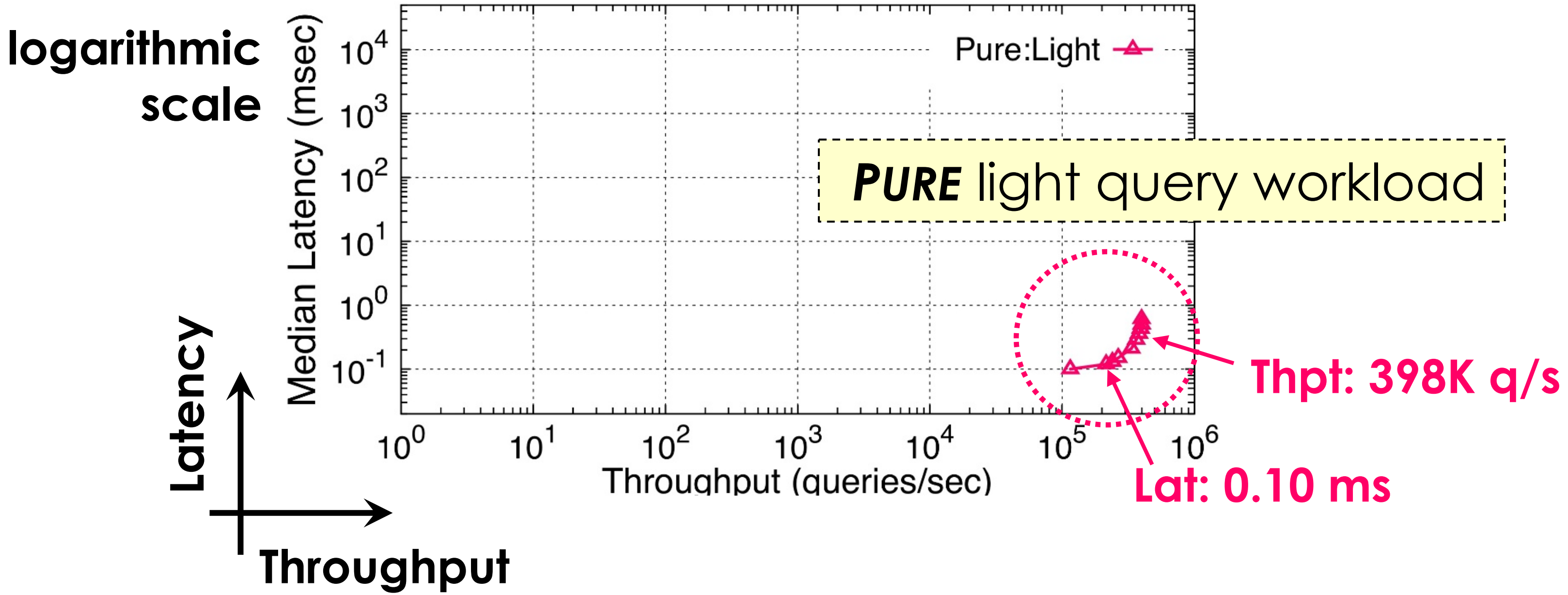
logarithmic scale





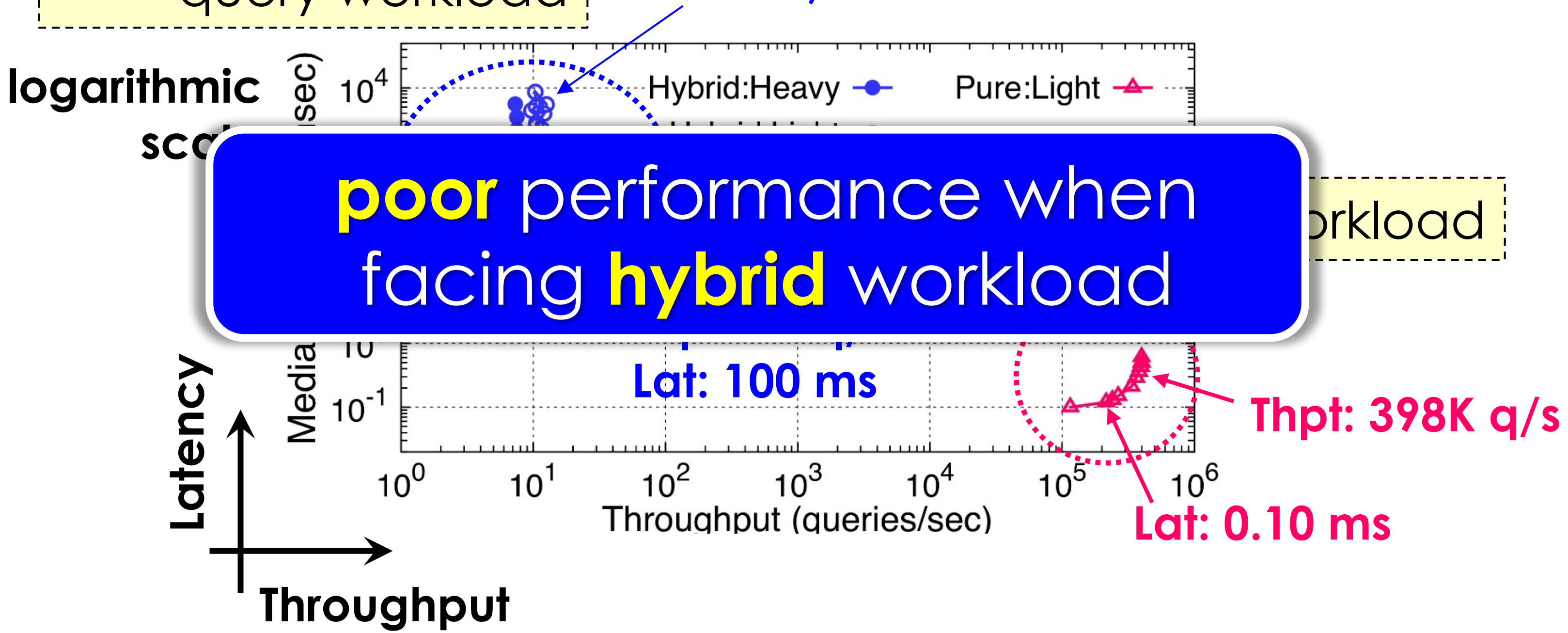
better

Concurrent Workload



Concurrent Workload

HYBRID light & heavy query workload

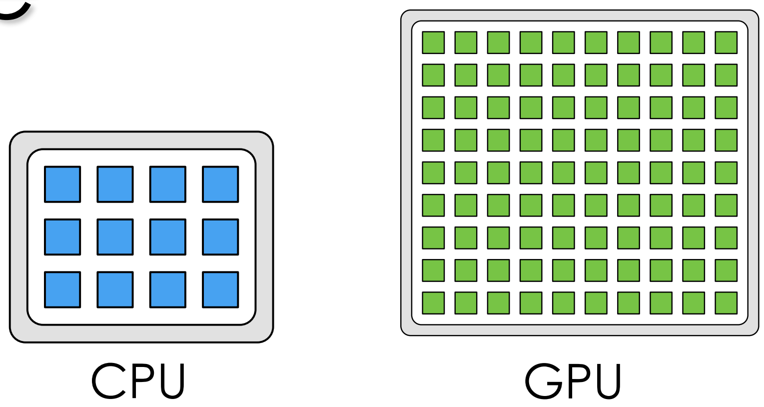


poor performance when facing hybrid workload

Advanced Hardware

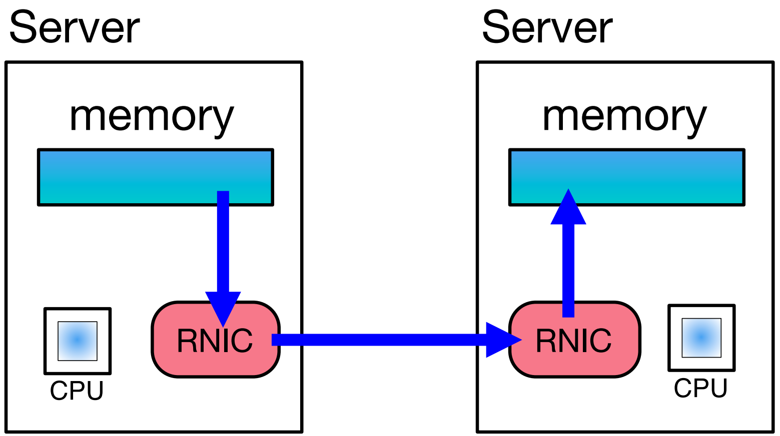
Heterogeneity

- **GPU** has many cores and high memory bandwidth



Fast Communications

- **RDMA** enables direct data transfer between machines

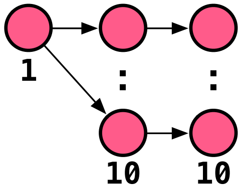


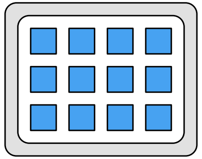
General Idea

Heterogeneous Workload

Heterogeneous Hardware

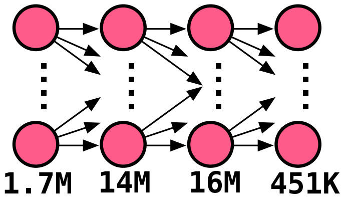
Light Query

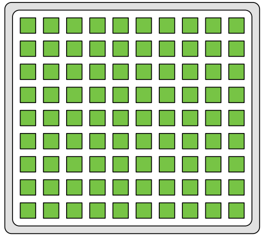




CPU

Heavy Query





GPU

System Overview

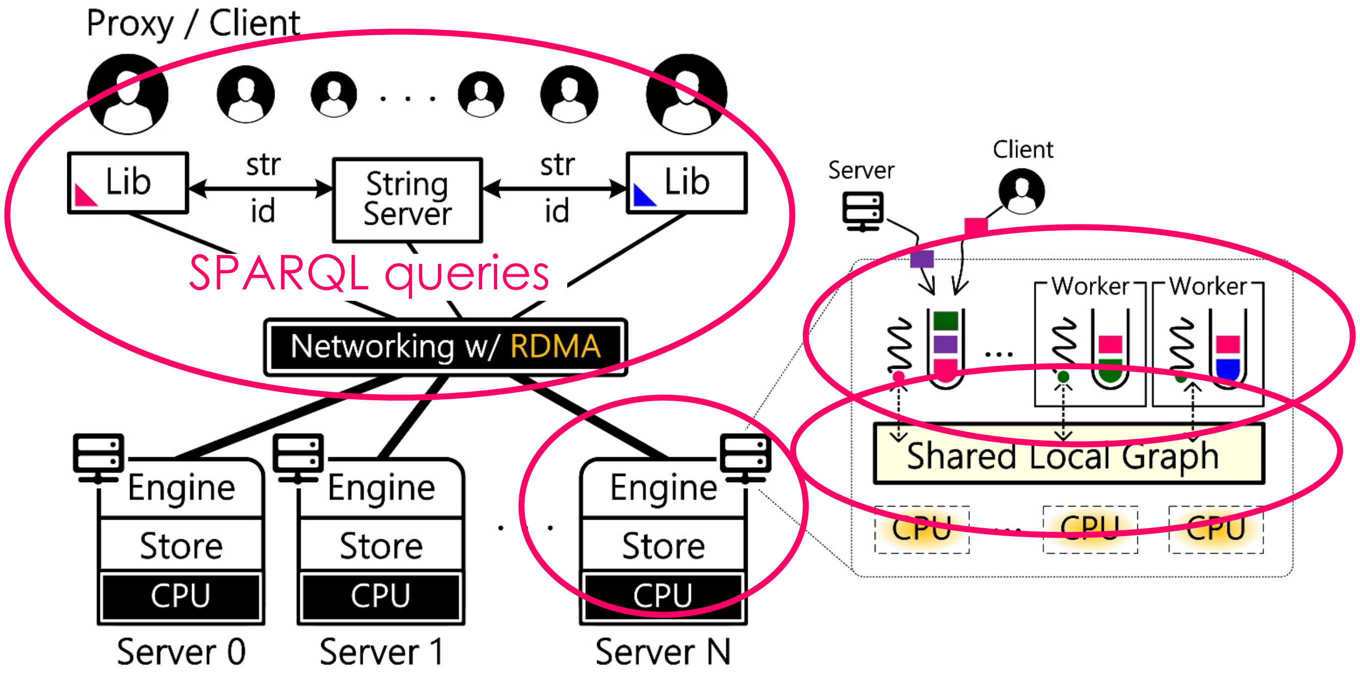
Wukong+G : a distributed graph query system that can leverage CPU/GPU to handle hybrid workload

1. GPU-enable graph exploration
2. GPU-friendly RDF store
3. Heterogeneous RDMA Communication

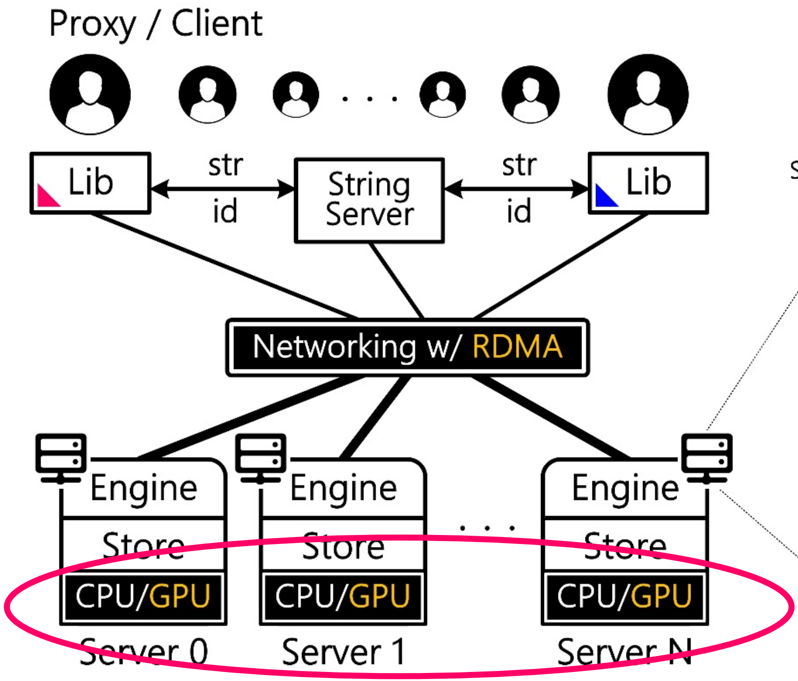
Performance improvement

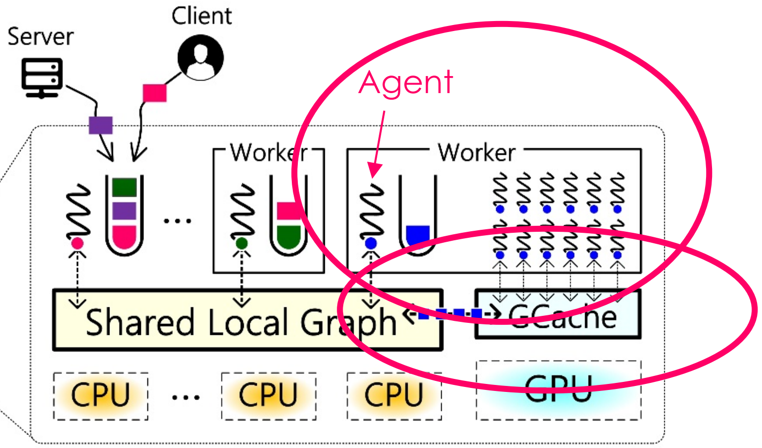
- Latency: **2.3X** - **9.0X** speedup for **heavy** query
- Throughput: **345K** queries/sec in **hybrid workloads**

Wukong Architecture



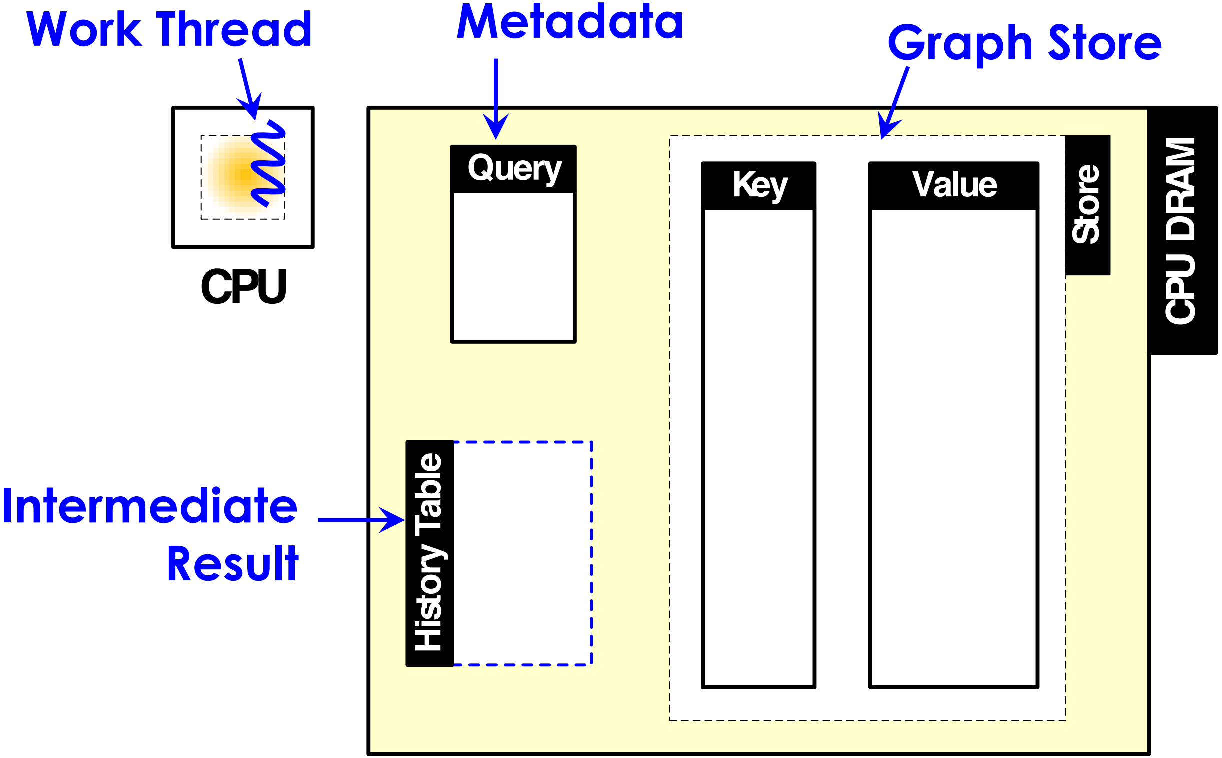
Wukong+G Architecture





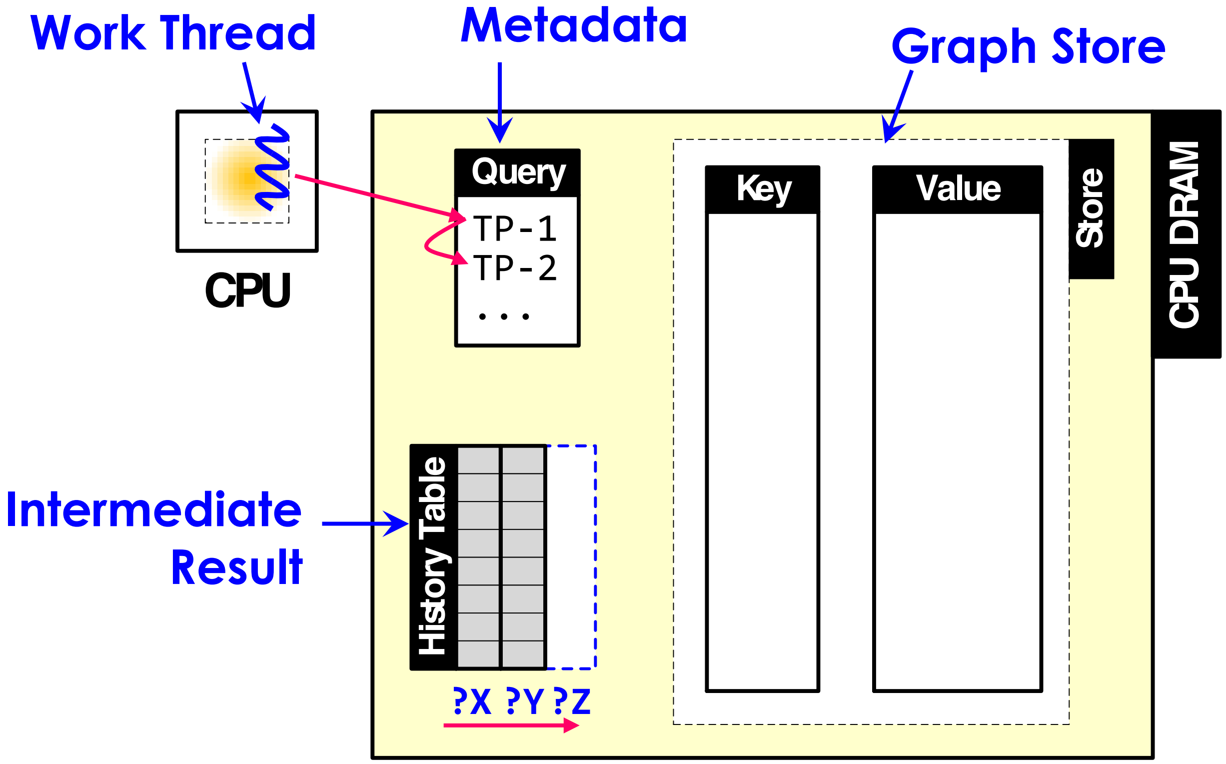
Query Execution on CPU

```
SELECT ?X ?Y ?Z WHERE {
 ?X teacherof ?Y .
 ?Z takecourse ?Y .
 ?Z adivsor ?X .       }
```



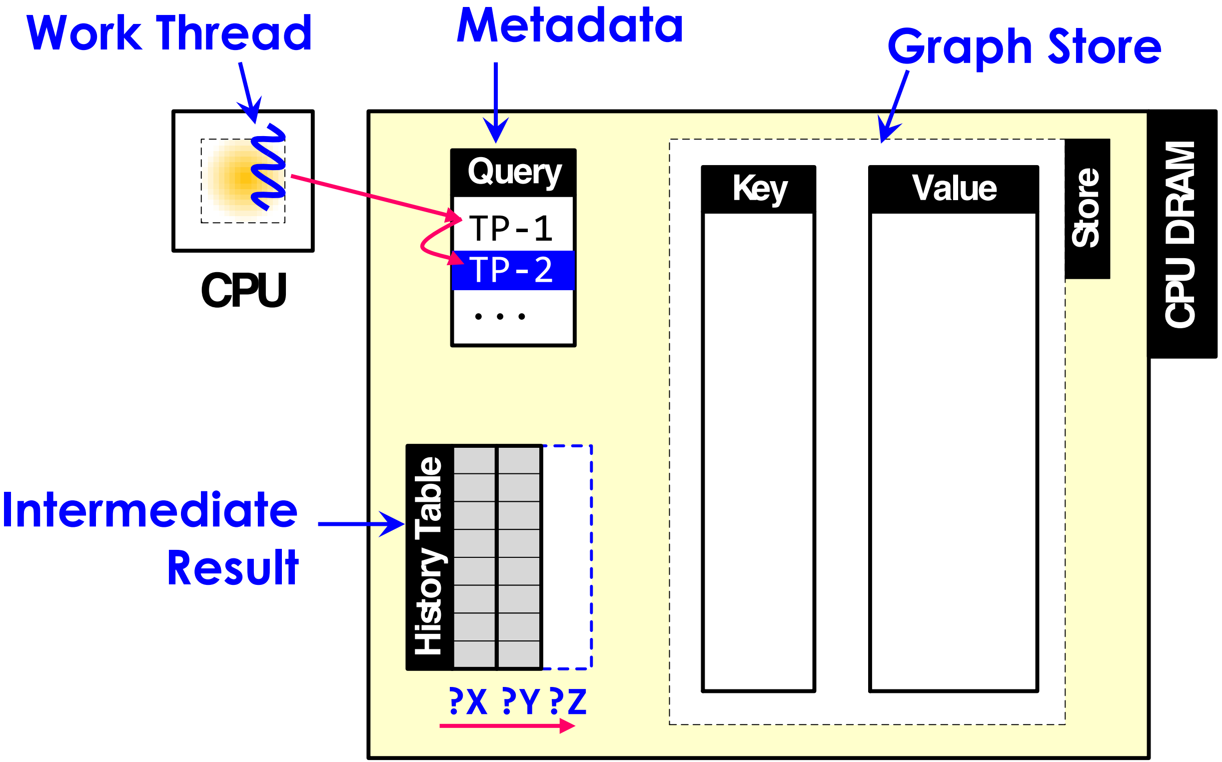
Query Execution on CPU

```
SELECT ?X ?Y ?Z WHERE {
 ?X teacherof ?Y .
 ?Z takecourse ?Y .
 ?Z adivsor ?X .       }
```



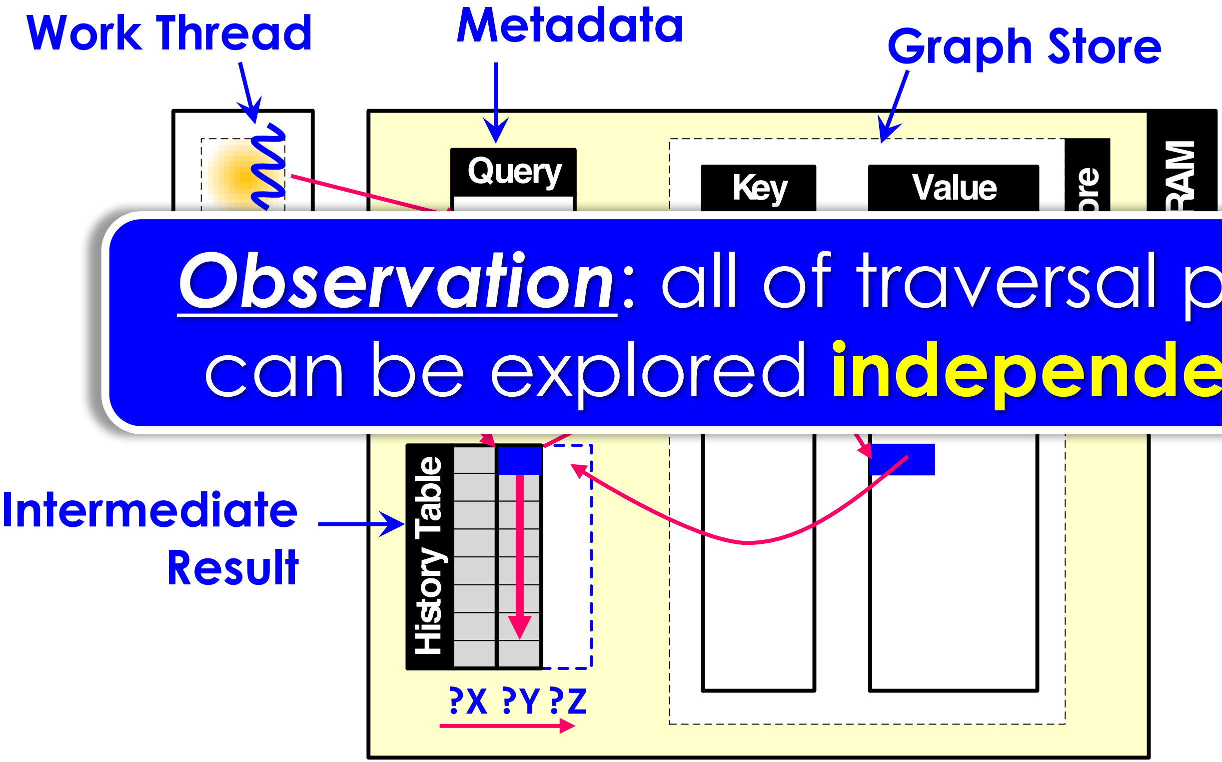
Query Execution on CPU

```
     SELECT ?X ?Y ?Z WHERE {
TP1   ?X teacherof ?Y .
TP2   ?Z takecourse ?Y .
TP3   ?Z adivsor ?X .        }
```



Query Execution on CPU

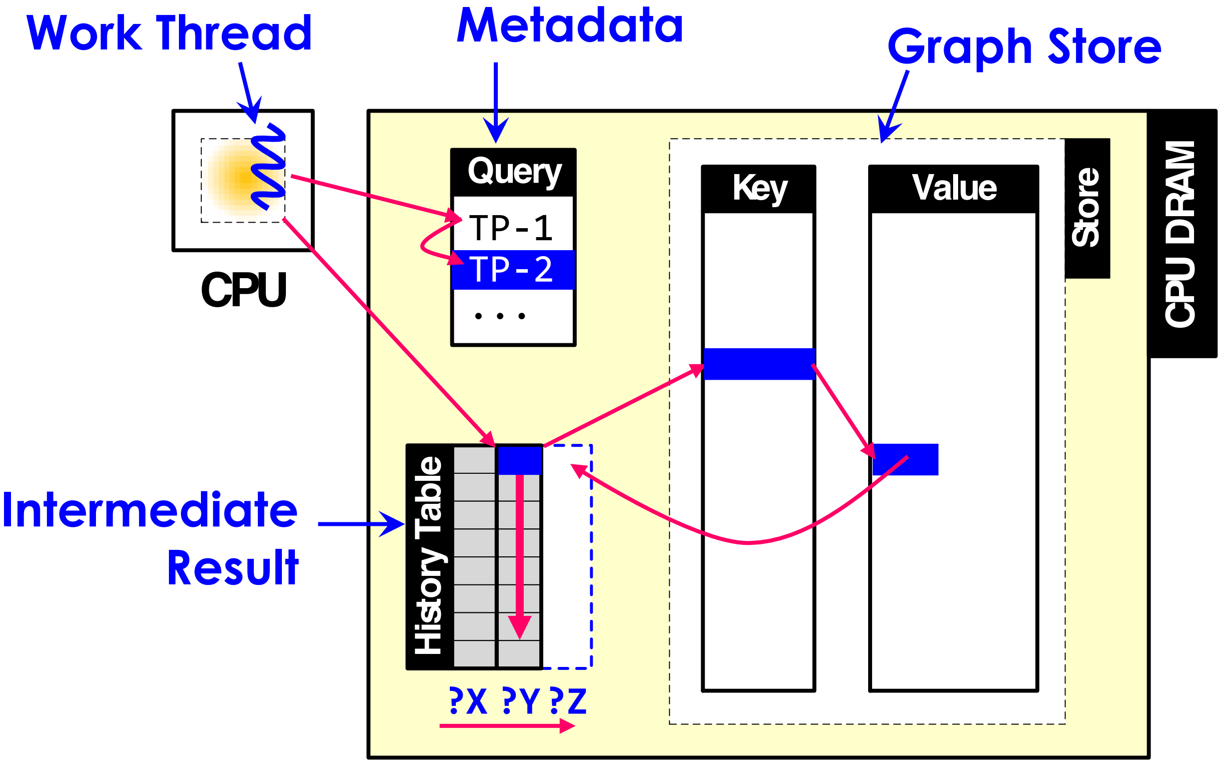
```
     SELECT ?X ?Y ?Z WHERE {
TP1   ?X teacherof ?Y .
TP2   ?Z takecourse ?Y .
TP3   ?Z adivsor ?X .        }
```



Observation: all of traversal paths can be explored **independently**

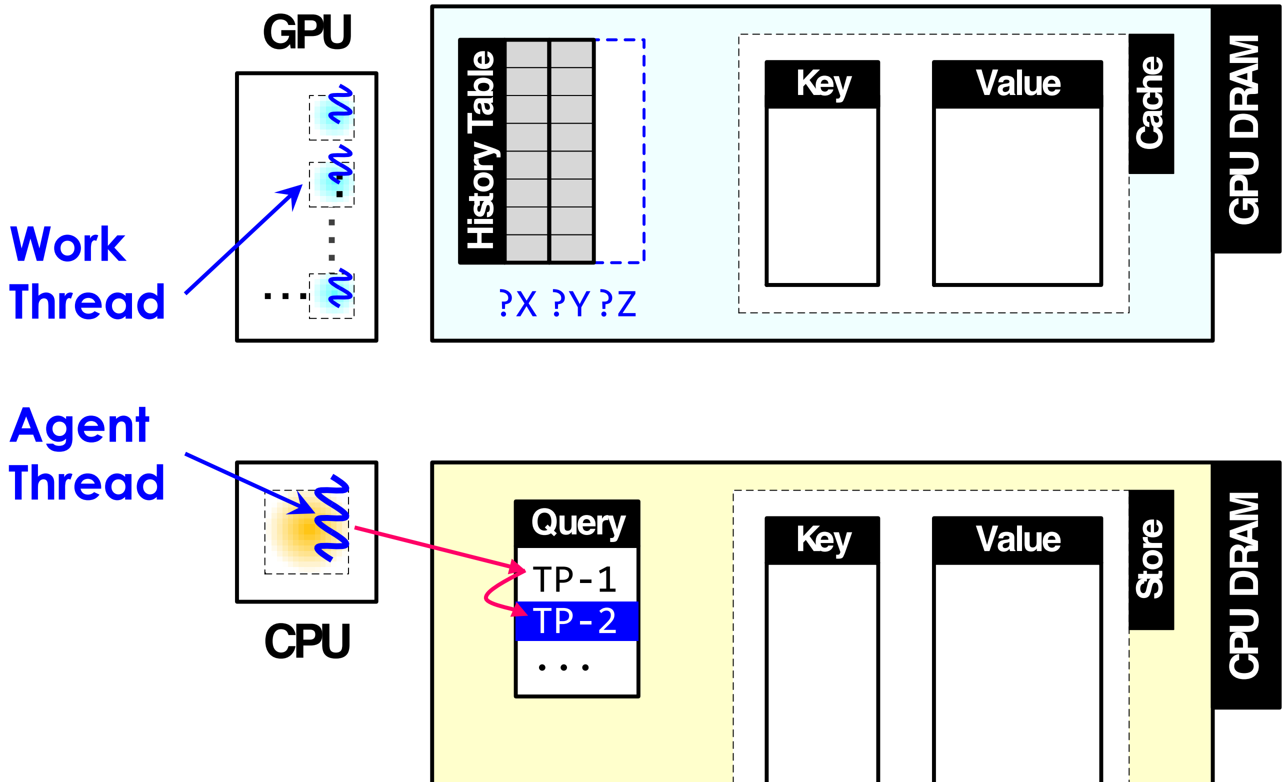
Query Execution on CPU

```
     SELECT ?X ?Y ?Z WHERE {
TP1   ?X teacherof ?Y .
TP2   ?Z takecourse ?Y .
TP3   ?Z adivsor ?X .          }
```



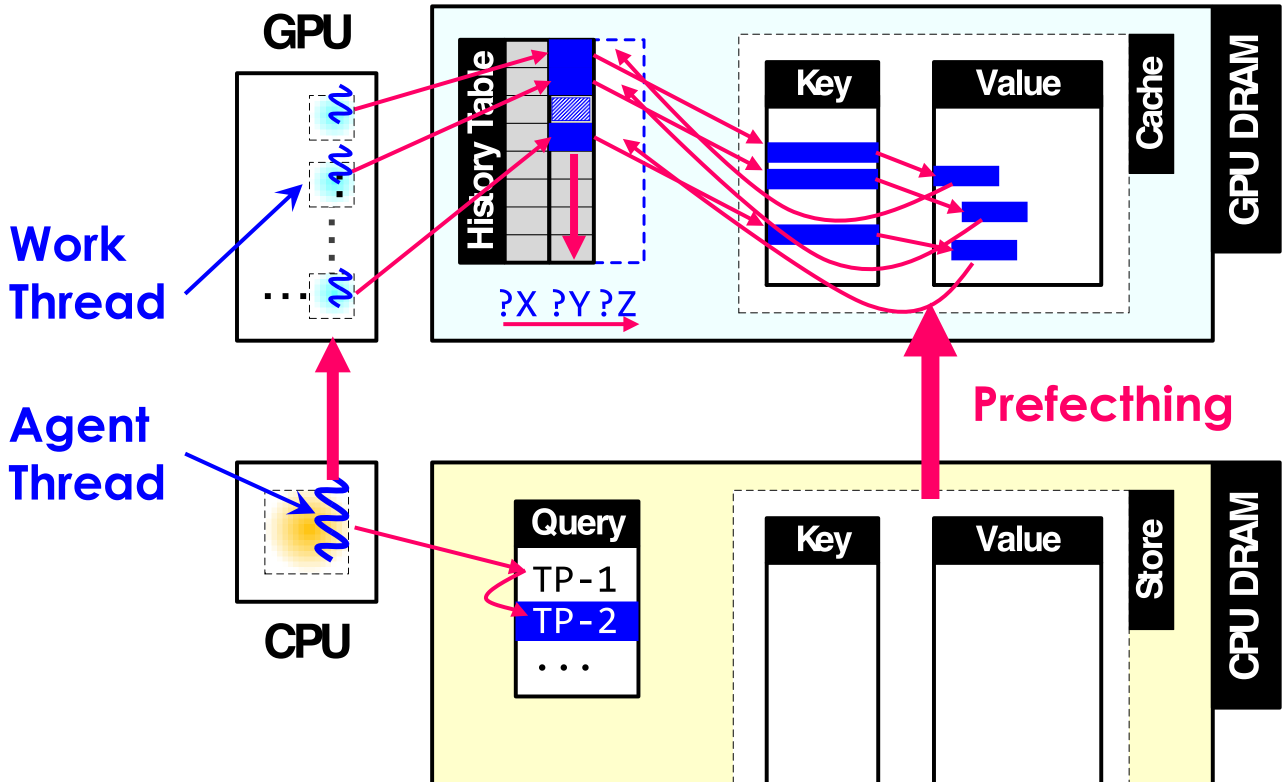
Query Execution on GPU

```
TP1    SELECT ?X ?Y ?Z WHERE {
         ?X teacherof ?Y .
TP2      ?Z takecourse ?Y .
TP3      ?Z adivsor ?X .        }
```



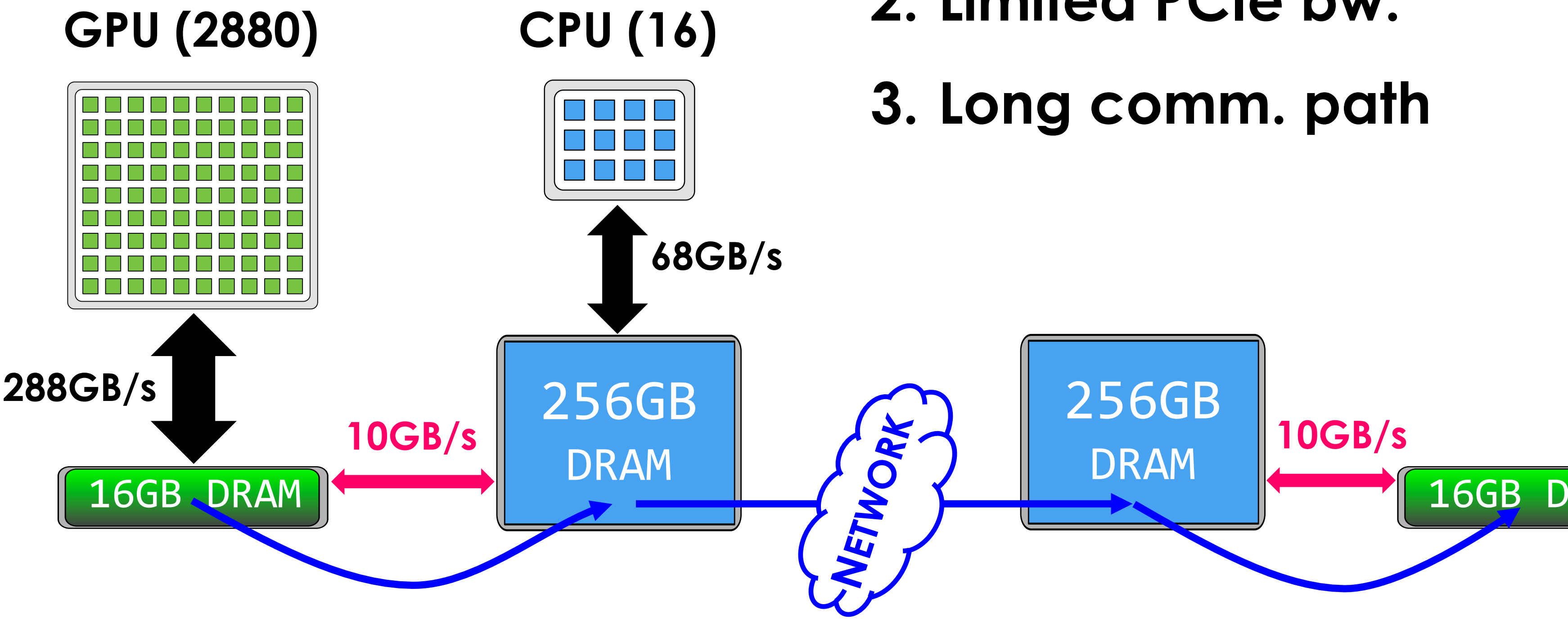
Query Execution on GPU

```
SELECT ?X ?Y ?Z WHERE {
TP1  ?X teacherof ?Y .
TP2  ?Z takecourse ?Y .
TP3  ?Z adivsor ?X .      }
```



Challenges

1. Small GPU memory
2. Limited PCIe bw.
3. Long comm. path



Smart data prefetching

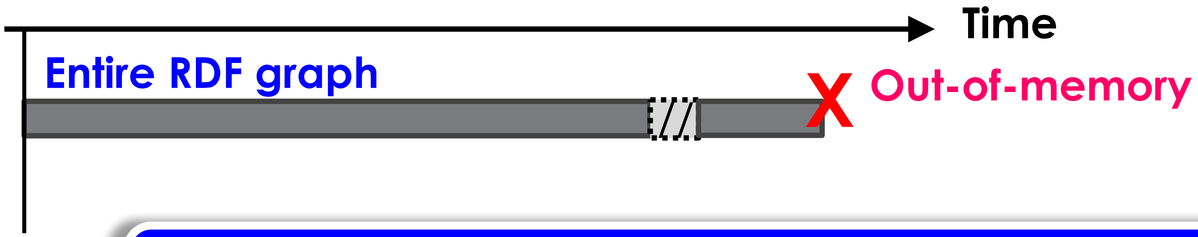
GPU-friendly key/value store

Heterogeneous RDMA comm.

Evaluation

Smart Data Prefetching

```
TP1
TP2
TP3
SELECT ?X ?Y ?Z WHERE {
  ?X teacherof ?Y .
  ?Z takecourse ?Y .
  ?Z adivsor ?X .      }
```



OB1: a query only touches **a part** of RDF data

560

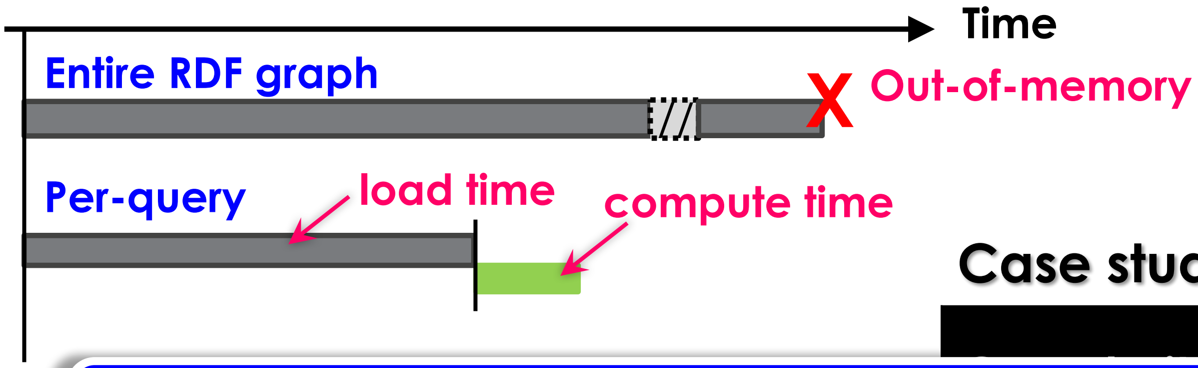
Granularity	Footprint	Data Transfer
Entire graph	16.3GB	Failed

OB2: the **predicate** of TP is commonly **known**

Smart Data Prefetching

TP1
TP2
TP3

```
SELECT ?X ?Y ?Z WHERE {
  ?X teacherof ?Y .
  ?Z takecourse ?Y .
  ?Z adivsor ?X .        }
```



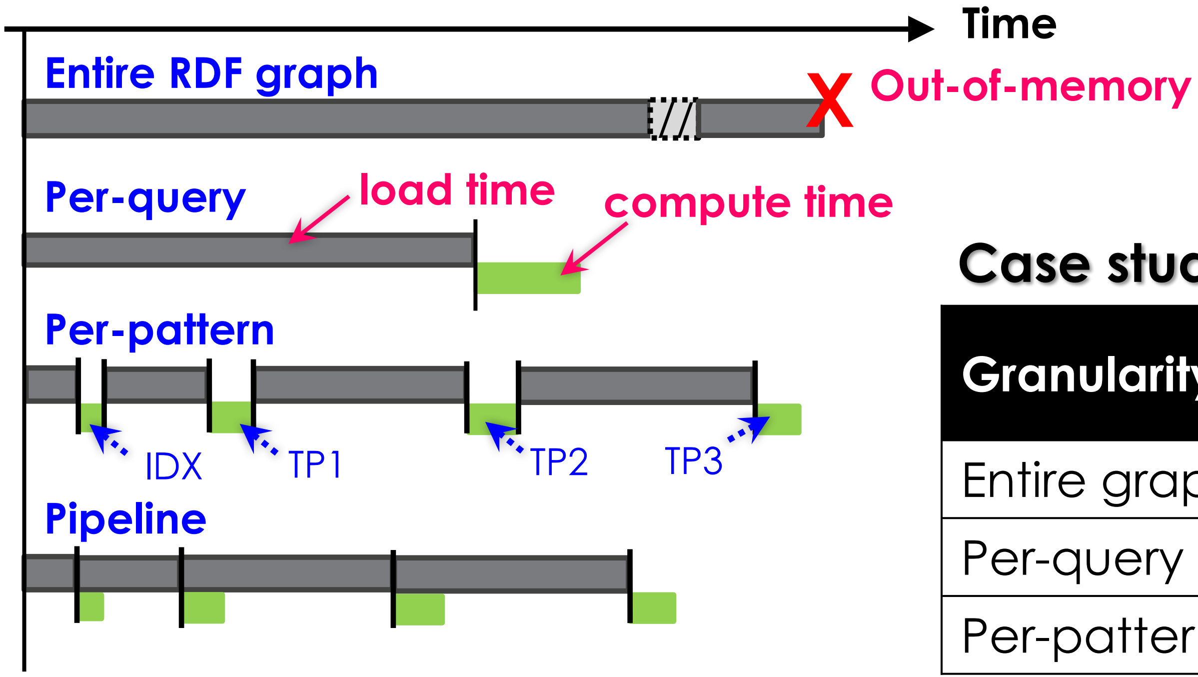
Case study: Q7 on LUBM-2560

	Memory	Data ...fer
		...ed
Per-query	**5.6GB**	5.6GB

OB3: **TPs** of a query will be executed **in sequence**

Smart Data Prefetching

```
TP1  SELECT ?X ?Y ?Z WHERE {
TP2    ?X teacherof ?Y .
TP3    ?Z takecourse ?Y .
       ?Z adivsor ?X .          }
```

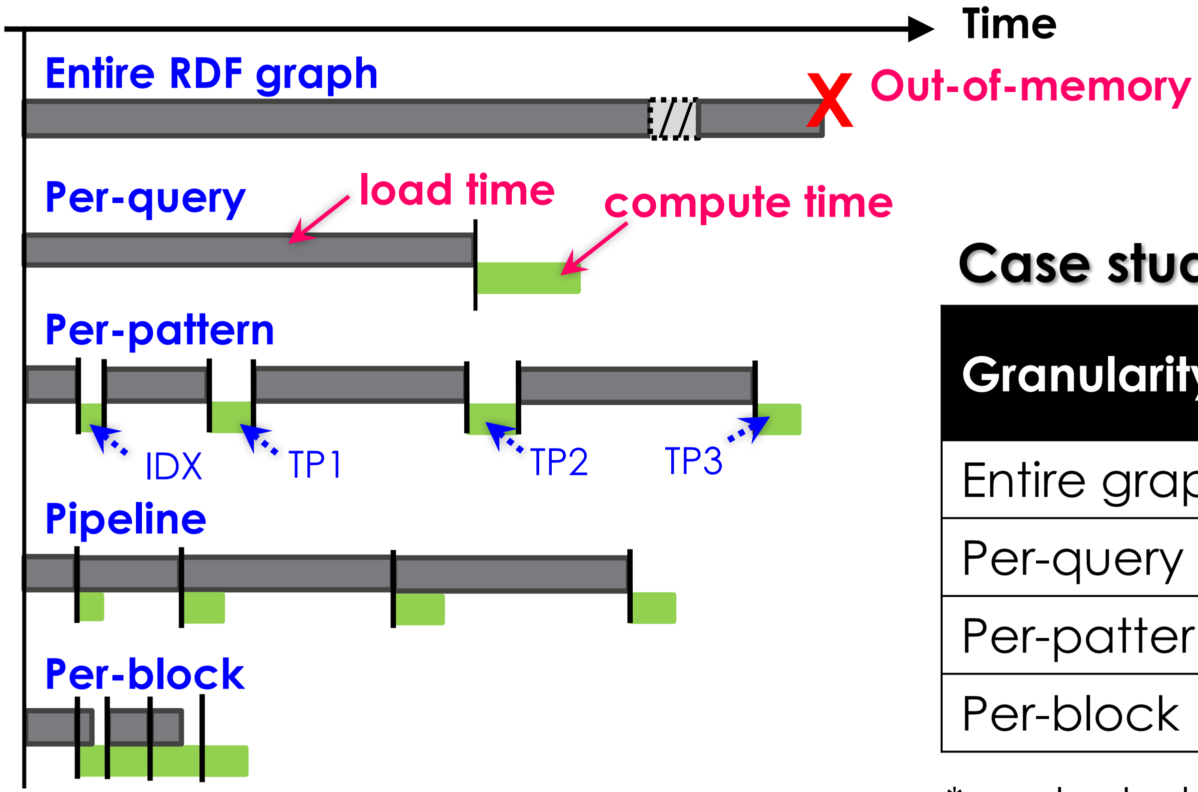


Case study: Q7 on LUBM-2560

Granularity	Memory Footprint	Data Transfer
Entire graph	16.3GB	Failed
Per-query	5.6GB	5.6GB
Per-pattern	**2.9GB**	5.6GB

Smart Data Prefetching

```
SELECT ?X ?Y ?Z WHERE {
TP1  ?X teacherof ?Y .
TP2  ?Z takecourse ?Y .
TP3  ?Z adivsor ?X .      }
```



Case study: Q7 on LUBM-2560

Granularity	Memory Footprint	Data Transfer
Entire graph	16.3GB	Failed
Per-query	5.6GB	5.6GB
Per-pattern	2.9GB	5.6GB
Per-block	2.9GB	**0.7GB***

* evaluated on 6GB GPU memory

GPU-enable query processing

GPU-friendly key/value store

Heterogeneous RDMA comm.

Evaluation

Original RDF Store (Wukong)

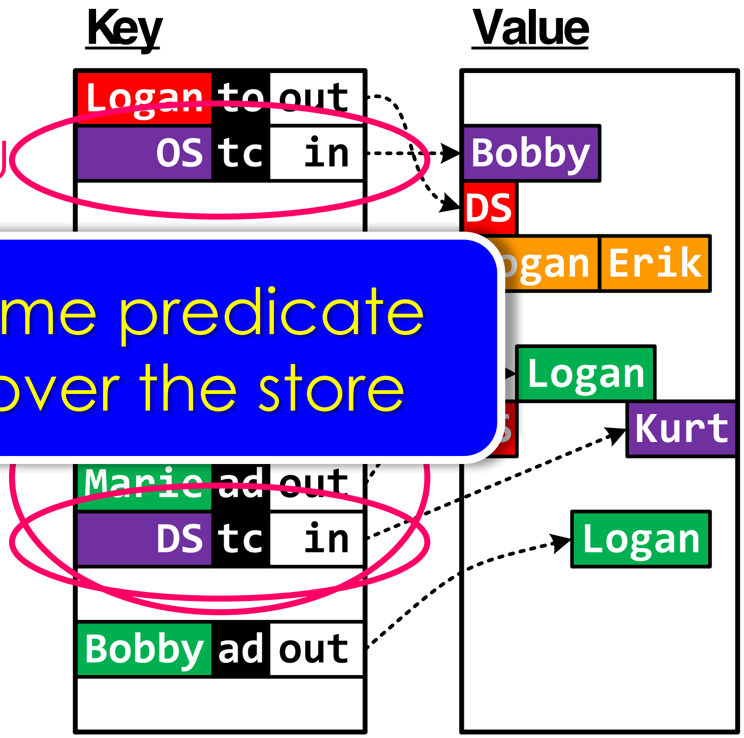
- **Predicate-based decomposition**

1. Efficient query processing on CPU

Hash

Triples with the same predicate are sprinkled all over the store

2. Provide possibility to prefetching triples with a certain predicate

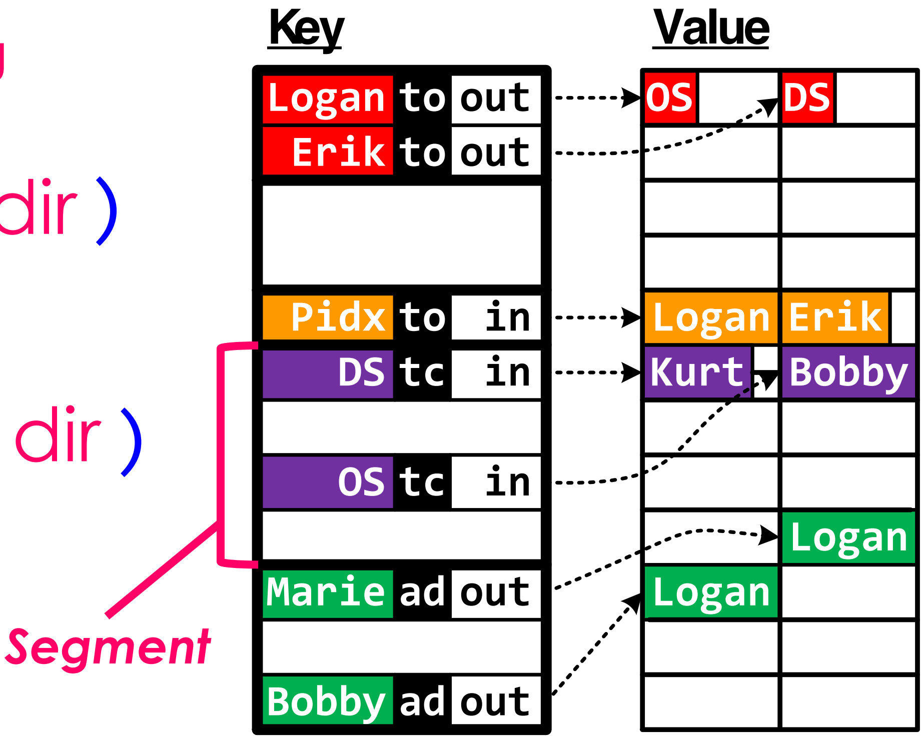


GPU-friendly RDF Store

- **Predicate-based grouping**

$$\mathtt{SEG\_OFFSET}(\mathit{pred}, \mathit{dir}) + \mathit{Hash}(\mathit{vtx}) \ \% \ \mathtt{SEG\_SZ}(\mathit{pred}, \mathit{dir})$$

RDF Store (CPU)



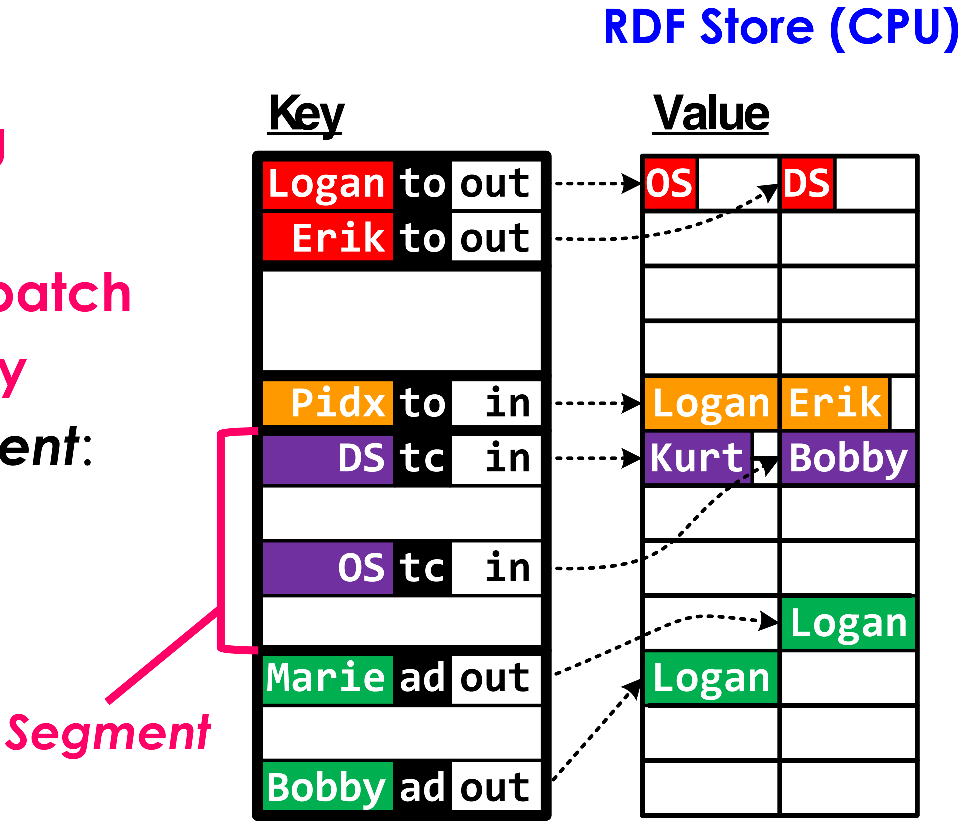
GPU-friendly RDF Store

- **Predicate-based grouping**

1. ***Segment***: prefetching in **batch**
2. ***Hashing***: lookup **efficiency**
3. ***Occupancy rate of segment***:

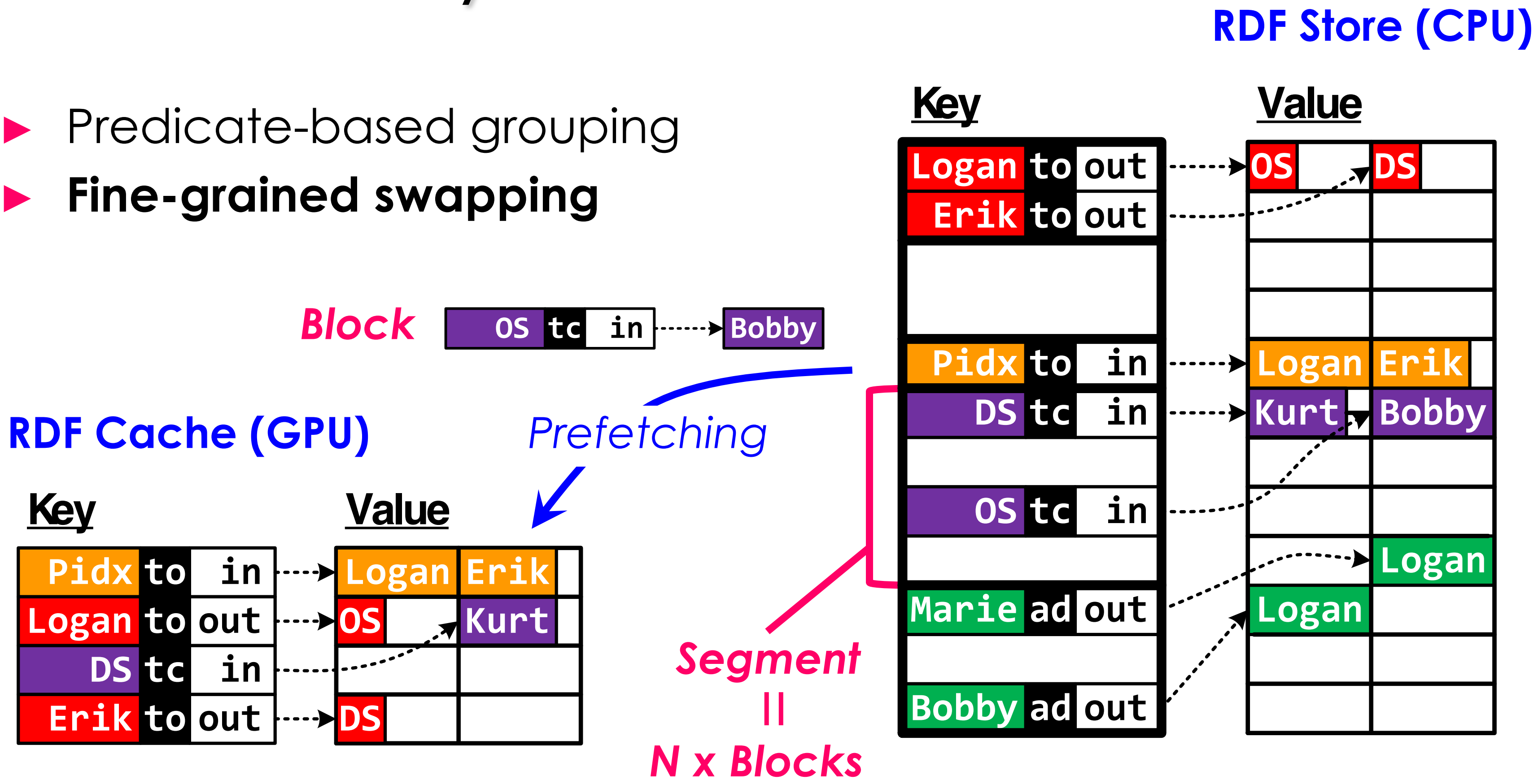
Prefetching Cost
vs.
Lookup Overhead

Tradeoff



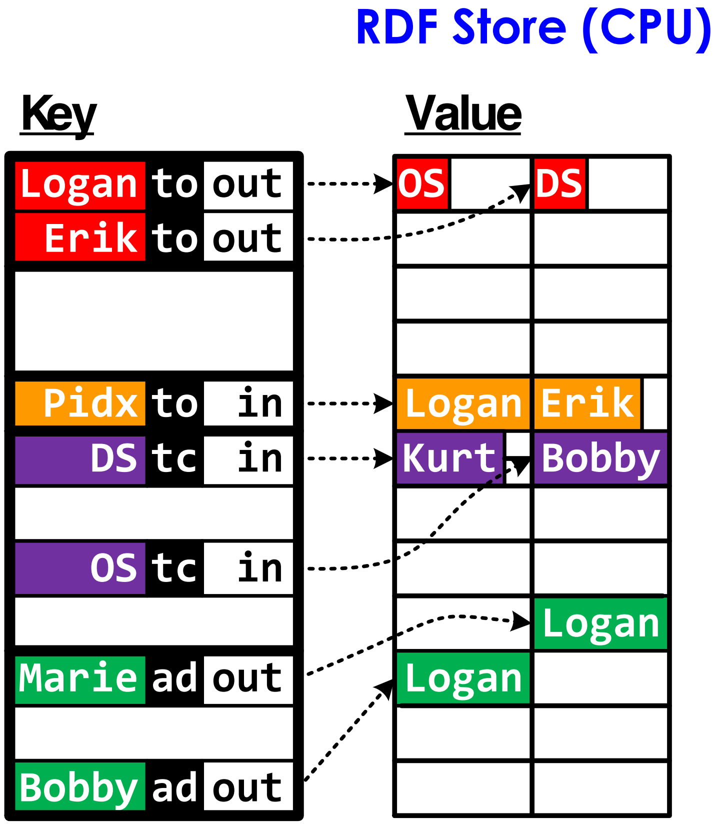
GPU-friendly RDF Store

- Predicate-based grouping
- **Fine-grained swapping**



GPU-friendly RDF Store

- Predicate-based grouping
- Fine-grained swapping
- **Pairwise caching**
- **Look-ahead replacement** (see paper)



Agenda

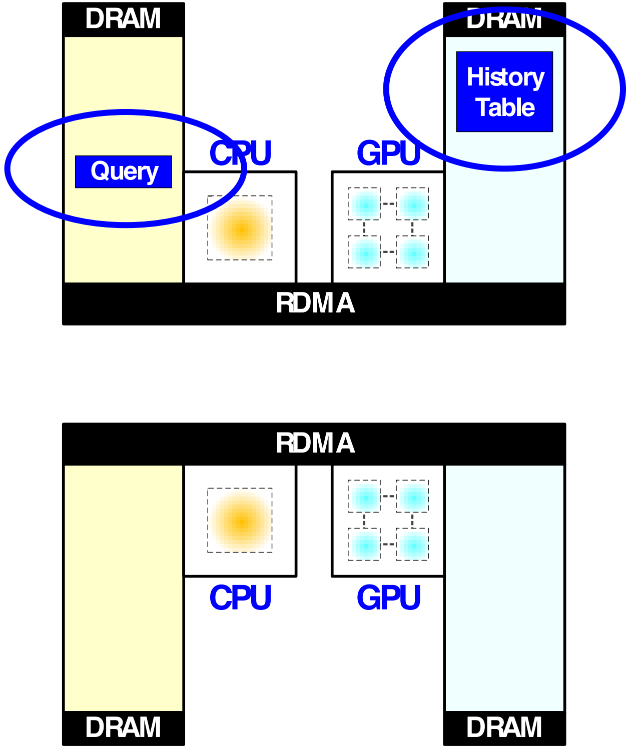
GPU-enable query processing

GPU-friendly key/value store

Heterogeneous RDMA comm.

Evaluation

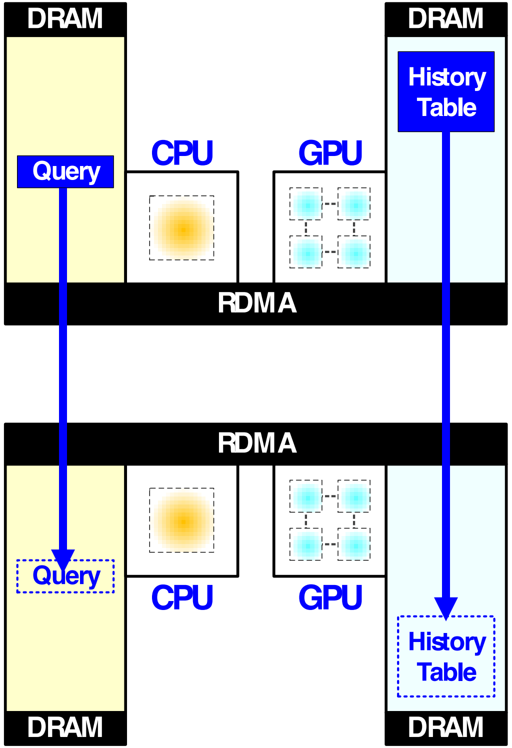
Heterogeneous RDMA Communication



Metadata: *Query*

Data: *History Table*

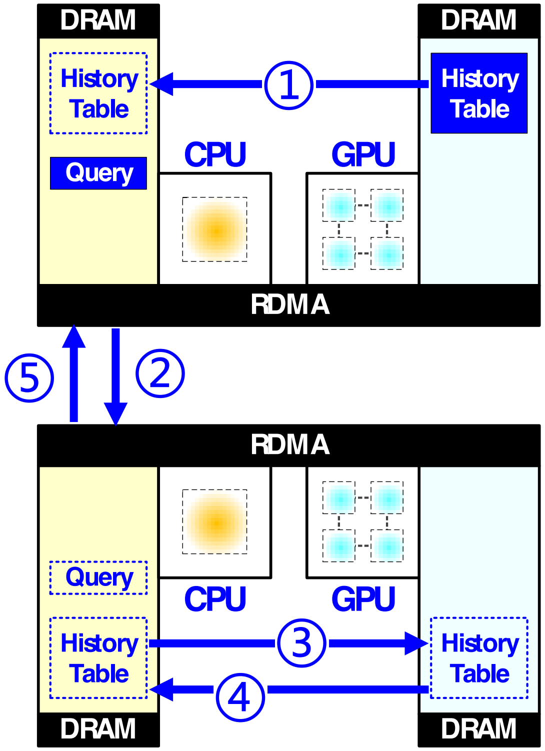
Heterogeneous RDMA Communication



Metadata: *Query*

Data: *History Table*

Heterogeneous RDMA Communication



(Native) **RDMA**

Metadata: *Query*

② CPU → CPU (RDMA)

Data: *History Table*

① GPU → CPU (PCIe)

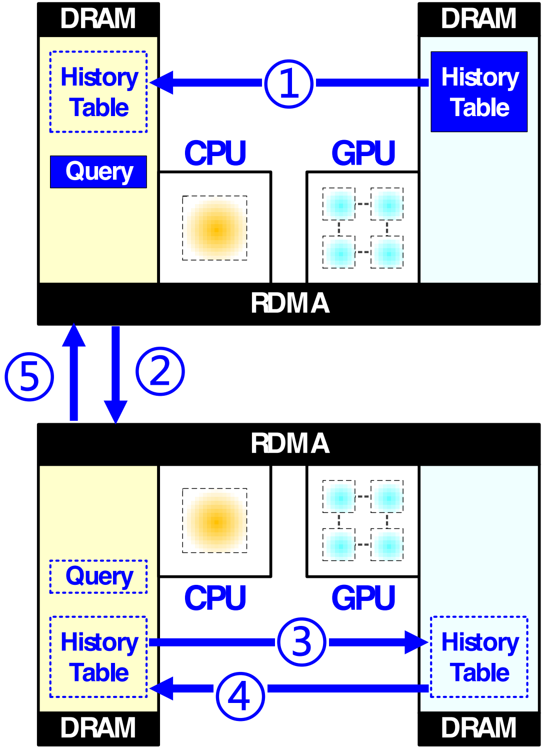
② CPU → CPU (RDMA)

③ CPU → GPU (PCIe)

④ GPU → CPU (PCIe)

⑤ CPU → CPU (RDMA)

Heterogeneous RDMA Communication



RDMA with **GPUDirect**

Metadata: *Query*

② CPU → CPU (RDMA)

Data: *History Table*

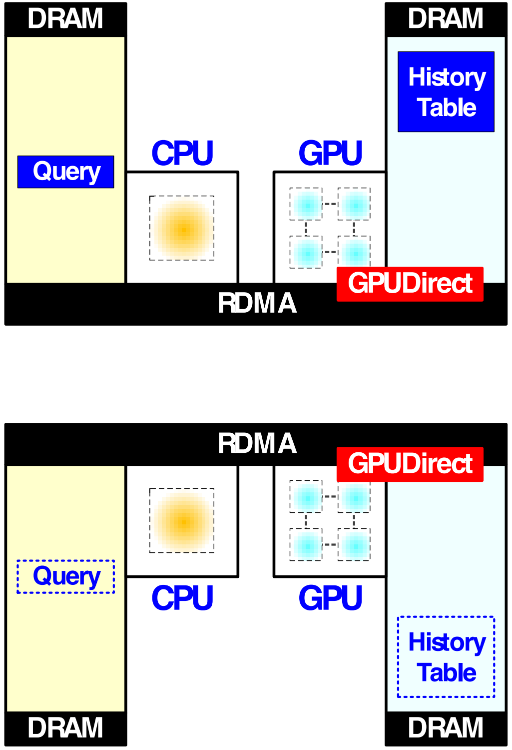
① GPU → CPU (PCIe)

② CPU → CPU (RDMA)

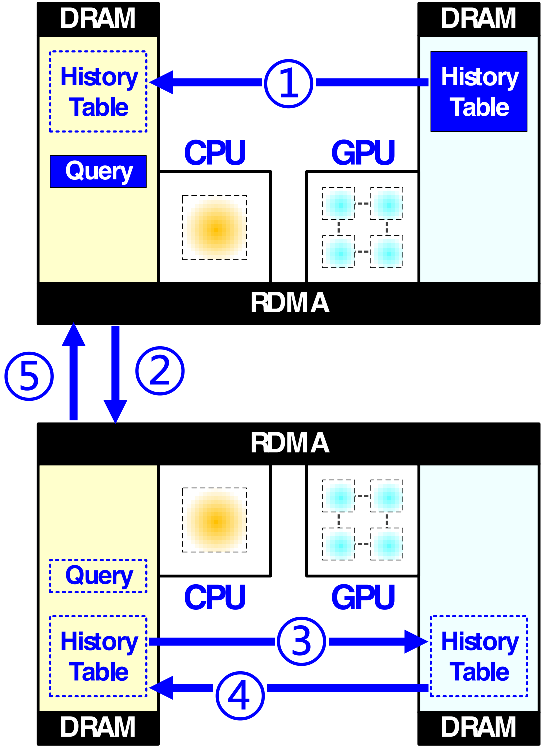
③ CPU → GPU (PCIe)

④ GPU → CPU (PCIe)

⑤ CPU → CPU (RDMA)



Heterogeneous RDMA Communication



RDMA with **GPUDirect**

Metadata: *Query*

① CPU → CPU (RDMA)

Data: *History Table*

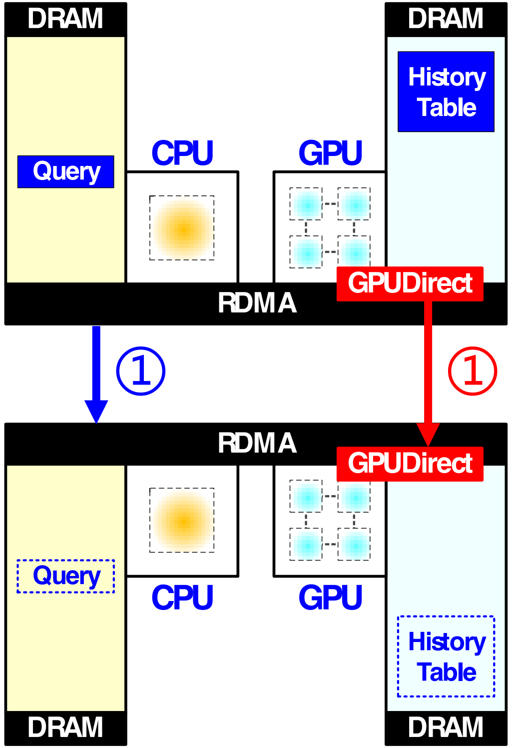
① GPU → CPU (PCIe)

① GPU → CPU (RDMA+G)

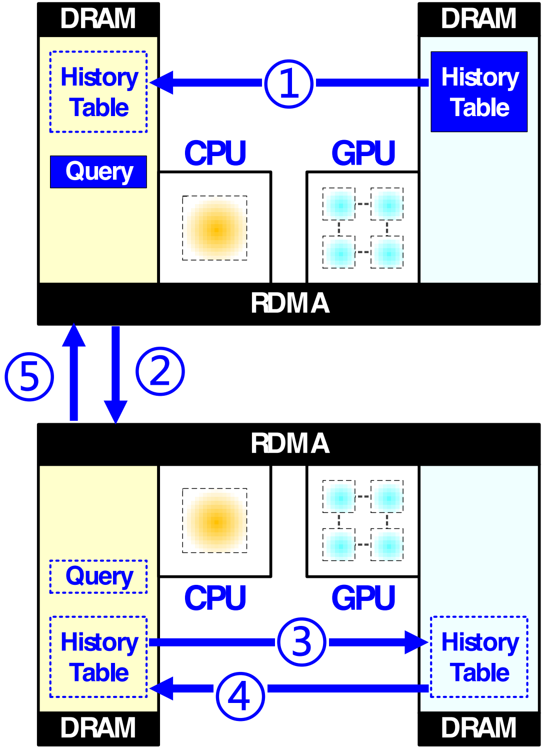
③ CPU → GPU (PCIe)

④ GPU → CPU (PCIe)

⑤ CPU → CPU (RDMA)



Heterogeneous RDMA Communication



RDMA with **GPUDirect**

Metadata: *Query*

① CPU → CPU (RDMA)

Data: *History Table*

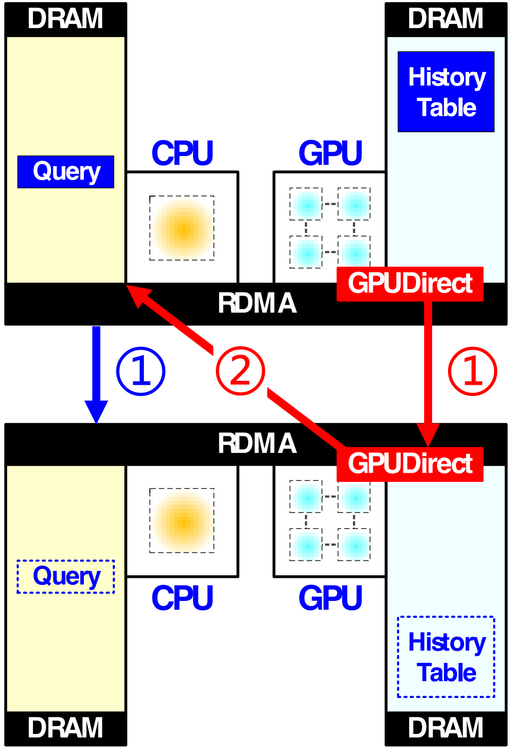
① GPU → CPU (PCIe)

① GPU → CPU (RDMA+G)

③ CPU → GPU (PCIe)

④ GPU → CPU (PCIe)

② GPU → CPU (RDMA+G)



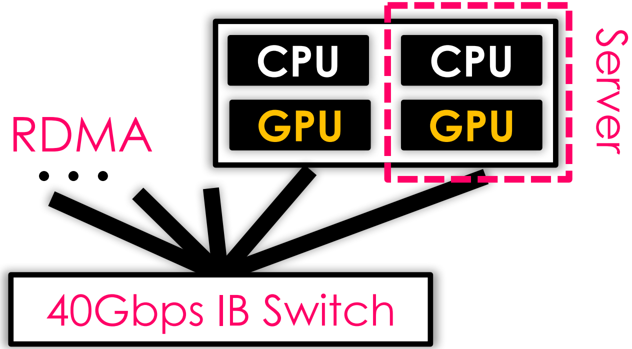
GPU-enable query processing

GPU-friendly key/value store

Heterogeneous RDMA comm.

Evaluation

Evaluation



Baseline: state-of-the-art systems

- Wukong, TriAD (distributed triple store)

Platforms: **10** servers on a rack-scale **5**-node cluster

- RDMA: Mellanox 56Gbps IB NIC, 40Gbps IB Switch
- Two servers run on a single machine
- Each server: 12-core Intel Xeon, 128GB DRAM, NVIDIA Tesla K40m (2880 cores, 12GB DRAM)

Benchmarks

- Synthetic: LUBM
- Real-life: DBPSB, YAGO2

Dataset	#T	#S	#O	#P	Size†
LUBM-2560	352 M	55 M	41 M	17	58GB
LUBM-10240	1,410 M	222 M	165 M	17	230GB
DBPSB	15 M	0.3 M	5.2 M	14,128	2.8GB
YAGO2	190 M	10.5 M	54.0 M	99	13GB

Single Query Latency (msec)

Heavy queries (Q1-Q3, Q7)

- Start from a set of vertices
- Touch a large part of graph
- Speedup: **2.3X~9X** vs. Wukong

Light queries (Q4-Q6)

- Start from a given vertex
- Touch a small part of graph
- Negligible slowdown

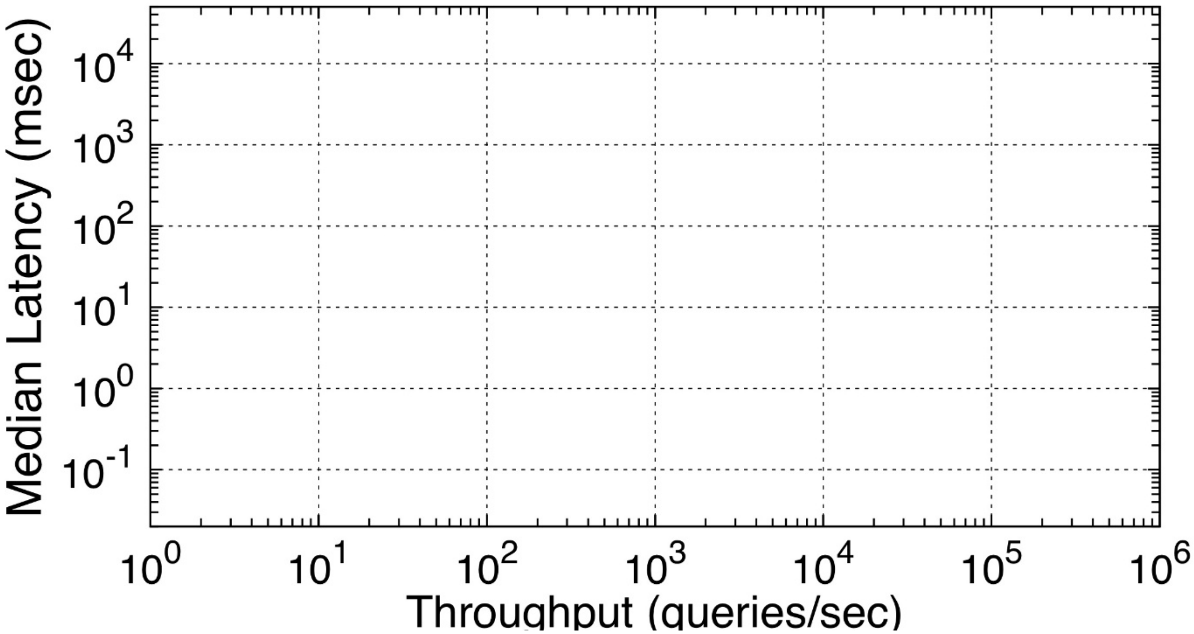
LUBM-2560		Wukong	Wukong+G
H	**Q1** (3.6GB)	992	**165**
	Q2 (2.4GB)	138	**31**
	Q3 (3.6GB)	340	**63**
	Q7 (5.6GB)	828	**100**
	Geo. M	443	**75**
L	**Q4**	**0.13**	0.16
	Q5	**0.09**	0.11
	Q6	**0.49**	0.51
	Geo. M	**0.18**	0.21

single server

LUBM-10240		Wukong	Wukong+G
H	**Q1** (14.25GB)	480	**211**
	Q2 (9.74GB)	66	**12**
	Q3 (14.25GB)	171	**19**
	Q7 (22.58GB)	390	**100**
	Geo. M	215	**47**
L	**Q4**	**0.44**	0.46
	Q5	**0.13**	0.17
	Q6	**0.70**	0.71
	Geo. M	**0.34**	0.38

10-server cluster

Performance of Hybrid Workloads



WKD (default)

Heavy/Light: ALL of CPUs **(10)**

WKI (Isolation)

Heavy: HALF of CPUs **(5)**

Light: HALF of CPUs **(5)**

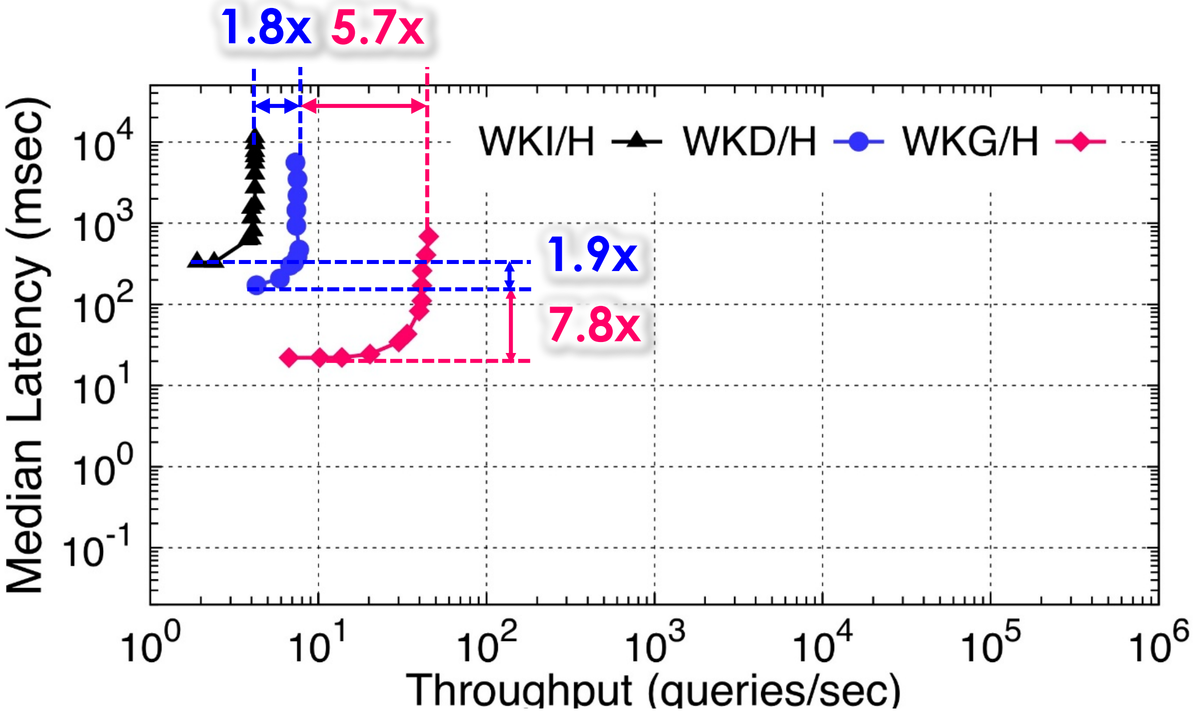
WKG (+G)

Heavy: CPU **(1)** + GPUs **(1)**

Light: **REST of** CPU **(9)**

Workload: 6 classes of light queries + 4 classes of heavy queries

Performance of Hybrid Workloads



WKD (default)

Heavy/Light: ALL of CPUs **(10)**

WKI (Isolation)

Heavy: HALF of CPUs **(5)**

Light: HALF of CPUs **(5)**

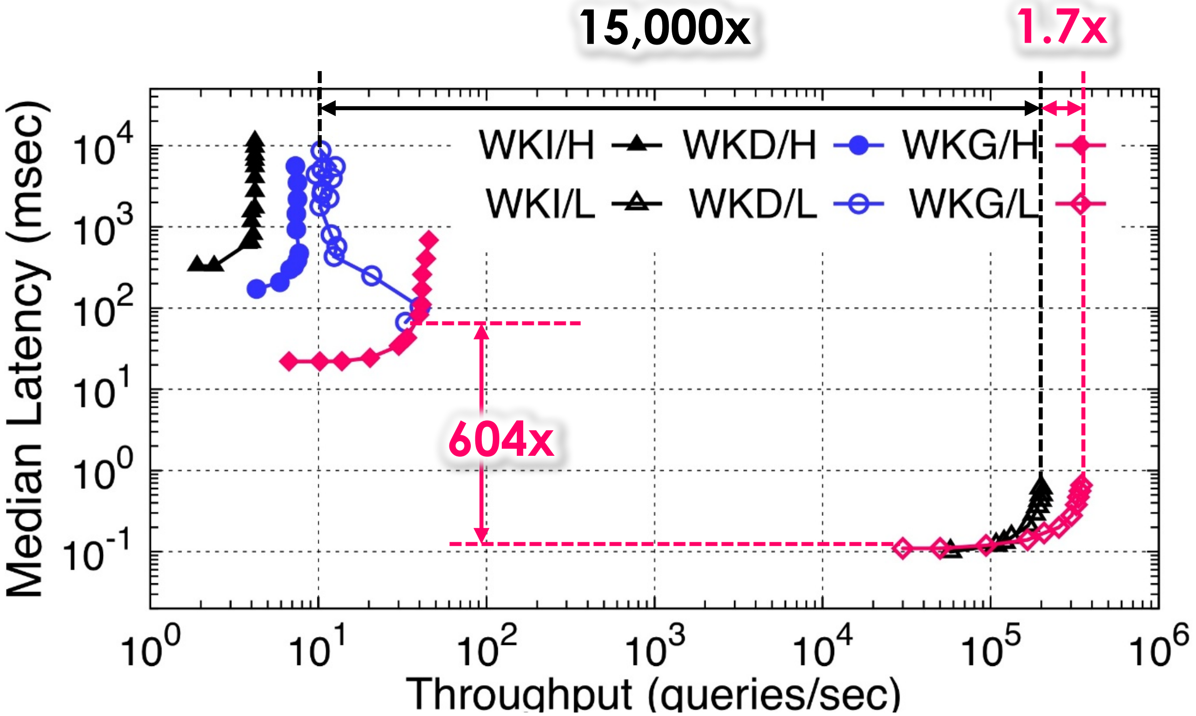
WKG (+G)

Heavy: CPU **(1)** + GPUs **(1)**

Light: **REST of** CPU **(9)**

Workload: 6 classes of light queries + 4 classes of heavy queries

Performance of Hybrid Workloads



WKD (default)

Heavy/Light: ALL of CPUs **(10)**

WKI (Isolation)

Heavy: HALF of CPUs **(5)**

Light: HALF of CPUs **(5)**

WKG (+G)

Heavy: CPU **(1)** + GPUs **(1)**

Light: **REST of** CPU **(9)**

Workload: 6 classes of light queries + 4 classes of heavy queries

Conclusion

Hardware heterogeneity opens opportunities for hybrid workloads on graph data

Wukong+G : a distributed RDF query system supports heterogeneous CPU/GPU processing for hybrid queries on graph data

Outperform prior state-of-the-art systems by more than one order of magnitude when facing hybrid workloads

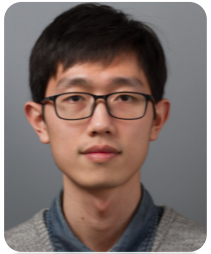
Website: http://ipads.se.sjtu.edu.cn/projects/wukong

GitHub: https://github.com/SJTU-IPADS/wukong

Thanks

Wukong+G

GitHub: https://github.com/SJTU-IPADS/wukong

Institute of Parallel and Distributed Systems
Shanghai Jiao Tong University

Questions

Distinguish Heavy & Light Queries

Query plan optimizer

- **Query plan**: the order of triple patterns
- Using a **cost-model** to estimate the execution time of different plans for a given query
- For SPARQL query, cost model is roughly based on **#paths** may be explored

Wukong+G uses a user-defined threshold for #paths to distinguish heavy and light queries

GPU Memory Size Limitation

Too large predicate segment

1. Load one part of segment to GPU memory
2. Do traversal work

Repeat 1 and 2

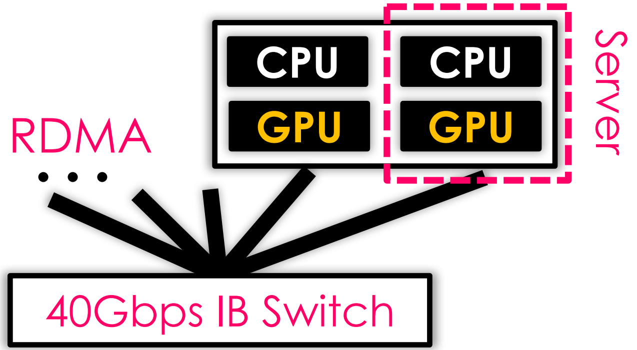
Too large intermediate results

1. Load one part of history table to GPU memory
2. Do traversal work

Repeat 1 and 2

Multi-GPUs Support

- Run a separate server for each GPU card and several co-located CPU cores (usually a socket)
- All servers comply with the same communication mechanism via GPUDirect RDMA operations



Graph Analytics vs. Graph Query

	Graph Analytics	Graph Query
Graph Model	Property Graph	Semantic (RDF) Graph
Working Set	A whole Graph	A small frac. of Graph
Processing	Batched & Iterative	Concurrent
Metrics	Latency	Latency & Throughput

Factor Analysis of Improvement

- Single Server w/ 3GB GPU memory
- LUBM-2560

LUBM-2560	Per-query	Per-parttern	Per-block	Pipeline
Q1 (3.6GB)	x	743	313	295
Q2 (2.4GB)	284	283	32	31
Q3 (3.6GB)	x	309	62	63
Q7 (5.6GB)	x	893	622	610

RDF Cache of GPU

- LUBM-2560

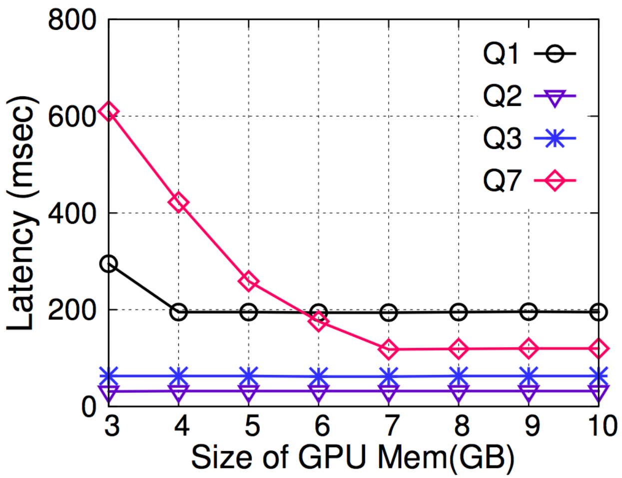


Fig. 11: The latency with the increase of GPU memory.

- LUBM-2560
- 10GB GPU Memory

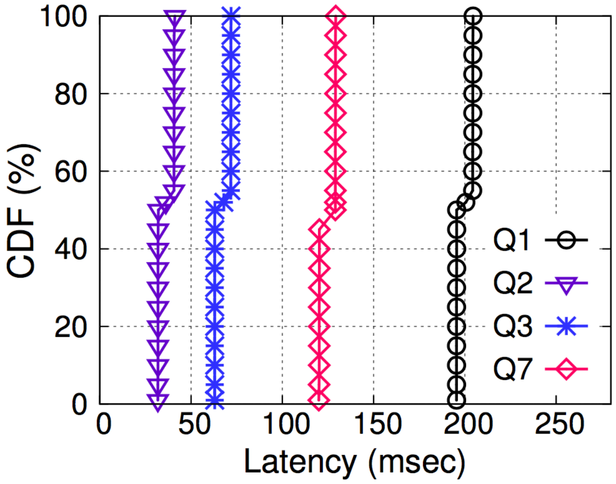


Fig. 12: The CDF of latency for mixed heavy workload.