Wire Speed Name Lookup: A GPU-based Approach

Yi Wang, Yuan Zu, Ting Zhang, Kunyang Peng, Qunfeng Dong, **Bin Liu**, Wei Meng, Huichen Dai, Xin Tian, Hao Wu, Di Yang

Tsinghua University

——Outline

1. Introduction
2. Name Lookup: Algorithm and Data Structure
3. Implementation
4. Experimental Results
5. Conclusion

清華大學 Tsinghua University

——Background & Movivation

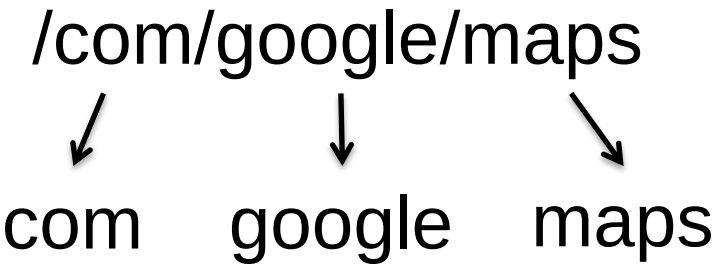
- **Name Lookup is widely used in a broad range of technological fields**
  - search engine
  - information retrieval
  - text processing
  - intrusion detection/prevention
  - …
- **But we have met a new research issue in the CCN scenario**

——Background & Motivation

- **Content-Centric Networking (CCN)**
  - CCN uses **names** to identify every piece of contents instead of IP addresses for hardware devices attached to IP network.
  - A forwarding table consists of **name prefixes**.
  - A core challenge and enabling technique in implementing CCN is exactly to perform **name lookup** for packet forwarding at wire speed.

——Background & Motivation

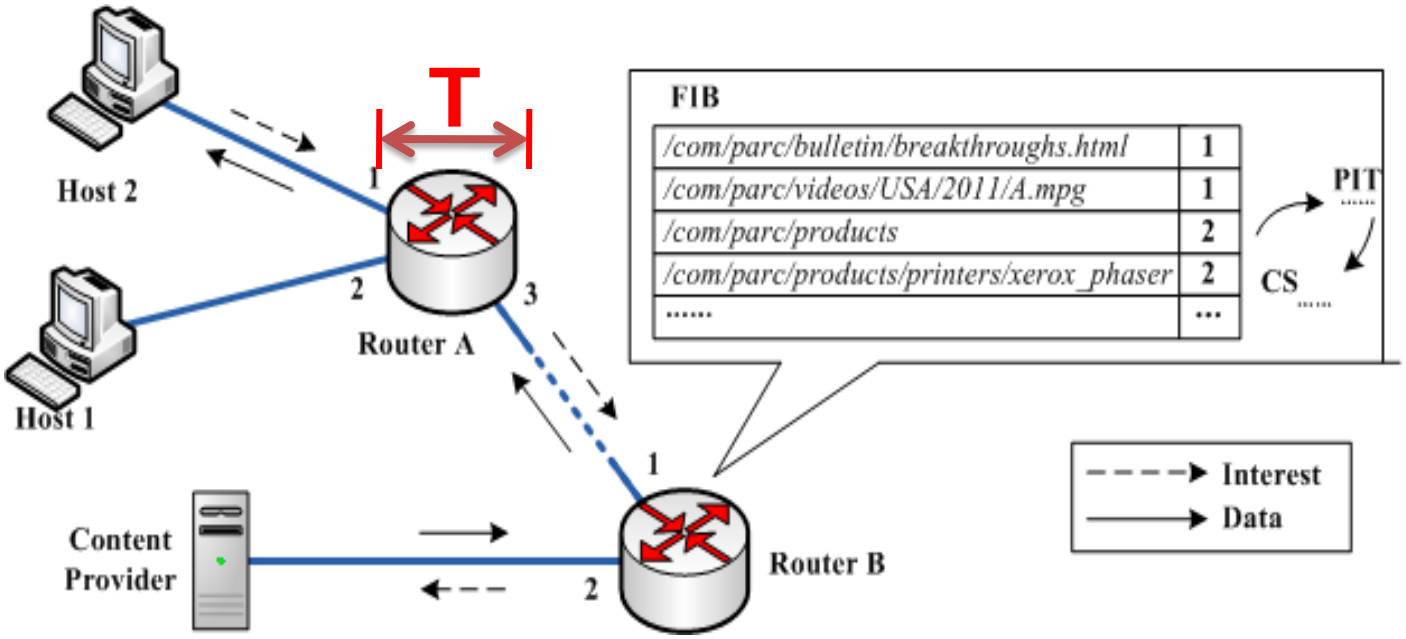
- Naming in CCN
  - A CCN name is hierarchically structured and composed of explicitly delimited components



——Background & challenges

- **Two High-level Requirements for CCN Name**

Lookup:



1) Longest name Prefix Matching(LPM)

2) Strict latency requirement (<100us)

——name lookup challenges

- **The detailed challenges of name lookup**
  - Complex name structure

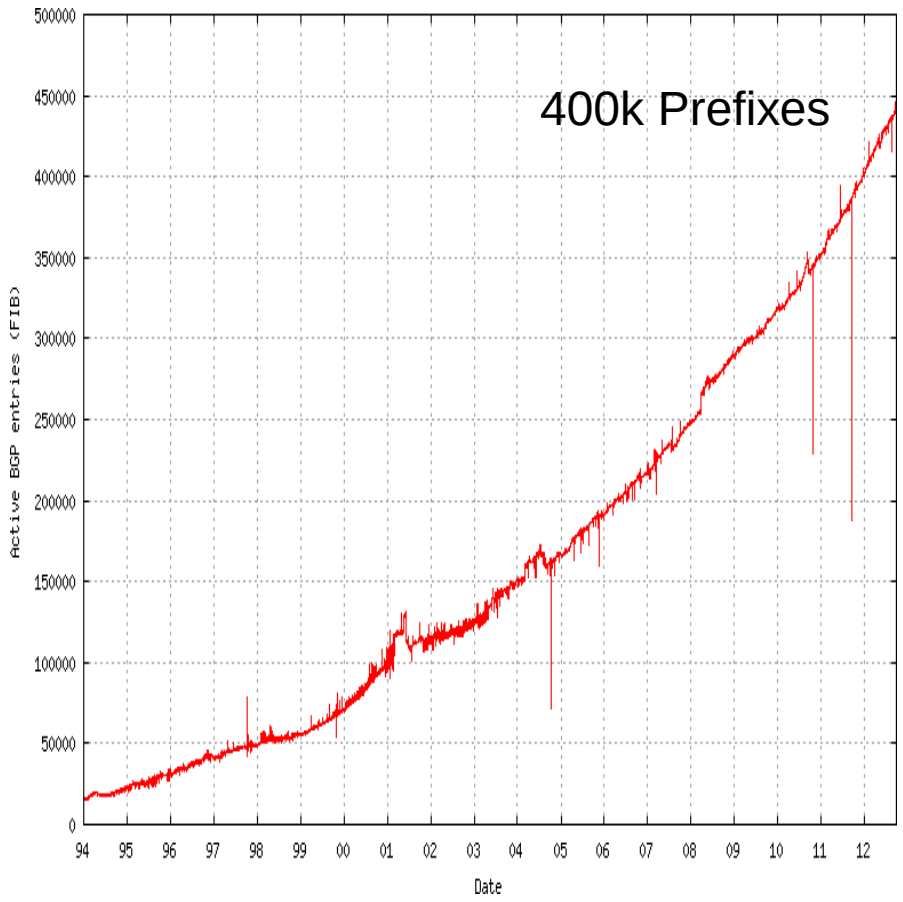
    1) consists of digits and characters;

    2) variable length name;

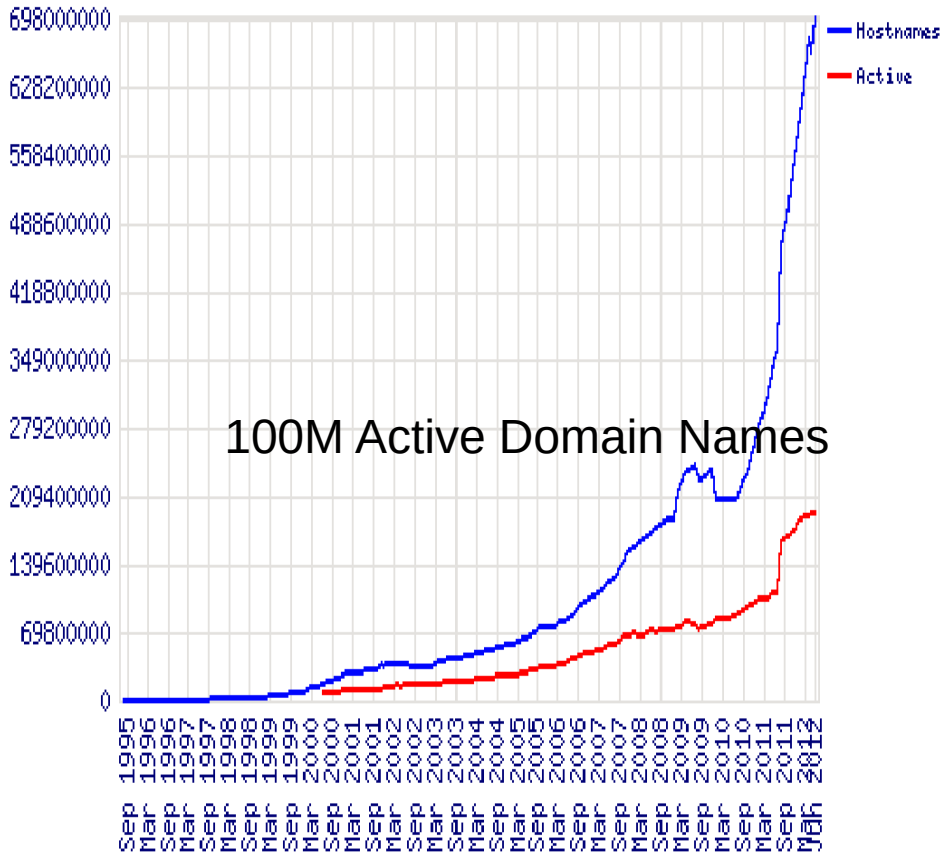
    3) without an externally imposed upper bound.
  - The large-scale name table

Tsinghua University

——name lookup challenges



Backbone IP Table Size



Number of Active Web-site Worldwide

Name tables could be 2~3 orders of magnitude larger than IP lookup table

——name lookup challenges

- **The detailed challenges of name lookup**
  - Complex name structure

    1) consists of digits and characters;

    2) variable length name;

    3) without an externally imposed upper bound.
  - The large-scale name table (**2~3 orders larger**)
  - Frequently update
  - Wire Speed (100Gbps Ethernet, OC-3072)

清華大學 Tsinghua University

——Contributions

1. We present the first design, implementation and evaluation of a GPU-based name lookup engine, which obtains **63.52M** searches per second, enabling line rate of **127** Gbps.
2. A new technique called *multiple aligned transition arrays (MATA)* is used to greatly compress storage space.
3. Stream-based pipeline approach ensures actual per-packet latency (*less than 100us*) while keeping high lookup throughput.

Tsinghua University

——Outline

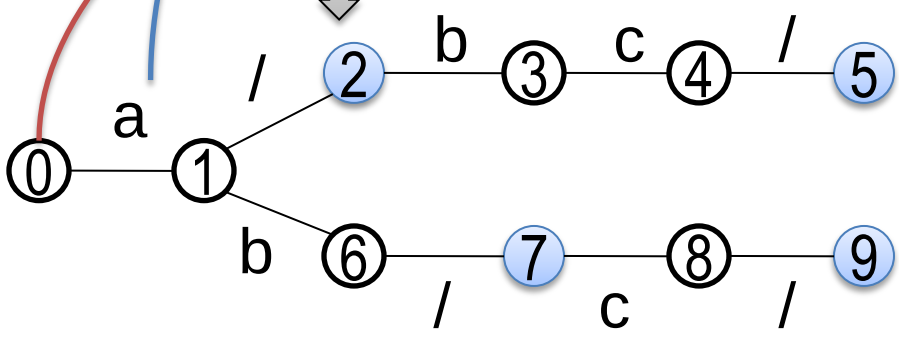
1. Background and Challenges
2. Name Lookup: Algorithm and Data Structure
3. Implementation
4. Experimental Results
5. Conclusion

—Algorithm & Data Structure

- Character Trie → Two-Dimensional State Transition Table (STT)

Name Table

/a/bc/
/ab/c
/a/
/ab/



Character Trie

	…	/	…	a	b	c	…
0				1			
1		2			6		
2					3		
3						4	
4		5					
5							
6		7					
7						8	
8		9					
9							
…							

- Two-Dimensional State Transition Table (STT)
  - Advantage
    - Easy to build
    - Fast speed: One State Transition needs one memory access only
  - Disadvantage
    - Too much memory required to be implemented

清華大學 Tsinghua University

—Algorithm & Data Structure

- Aligned Transition Array (ATA) to compress STT

STT

	…	/	…	a	b	c	…
0				1			
1		2			6		
2					3		
3						4	
4		5					
5							
…							

1000

998 ~~998~~

Aligned Transition Array

Address	Transition
…	…
1045	/,1001
1046	/,1003
…	…
1096	b,999
1097	a,998
1098	/,997
1099	b,1002
1100	/,1004
1101	c,1051
1102	c,1053
…	…

998+b=1096

~~999+b=1097~~

1000+a=1097

offset + character's ASCII code

- ATA
  - Advantage
    - Keep fast speed: one state transition needs one memory access
    - Low memory space
  - Disadvantage
    - Building speed is too slow for large-scale name table
    - Cannot support incremental updates

Tsinghua University

—Algorithm & Data Structure

Multiple Stride Character Trie

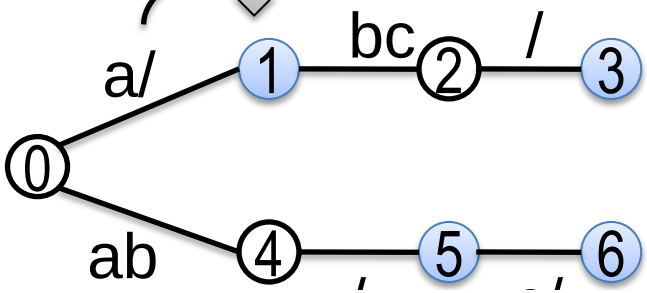
ATA

Name Table

/a/bc/
/ab/c
/a/
/ab/

"a/" = 24879

Address	Transition
0	
1	a/,…
…	
24879	ab,…
…	



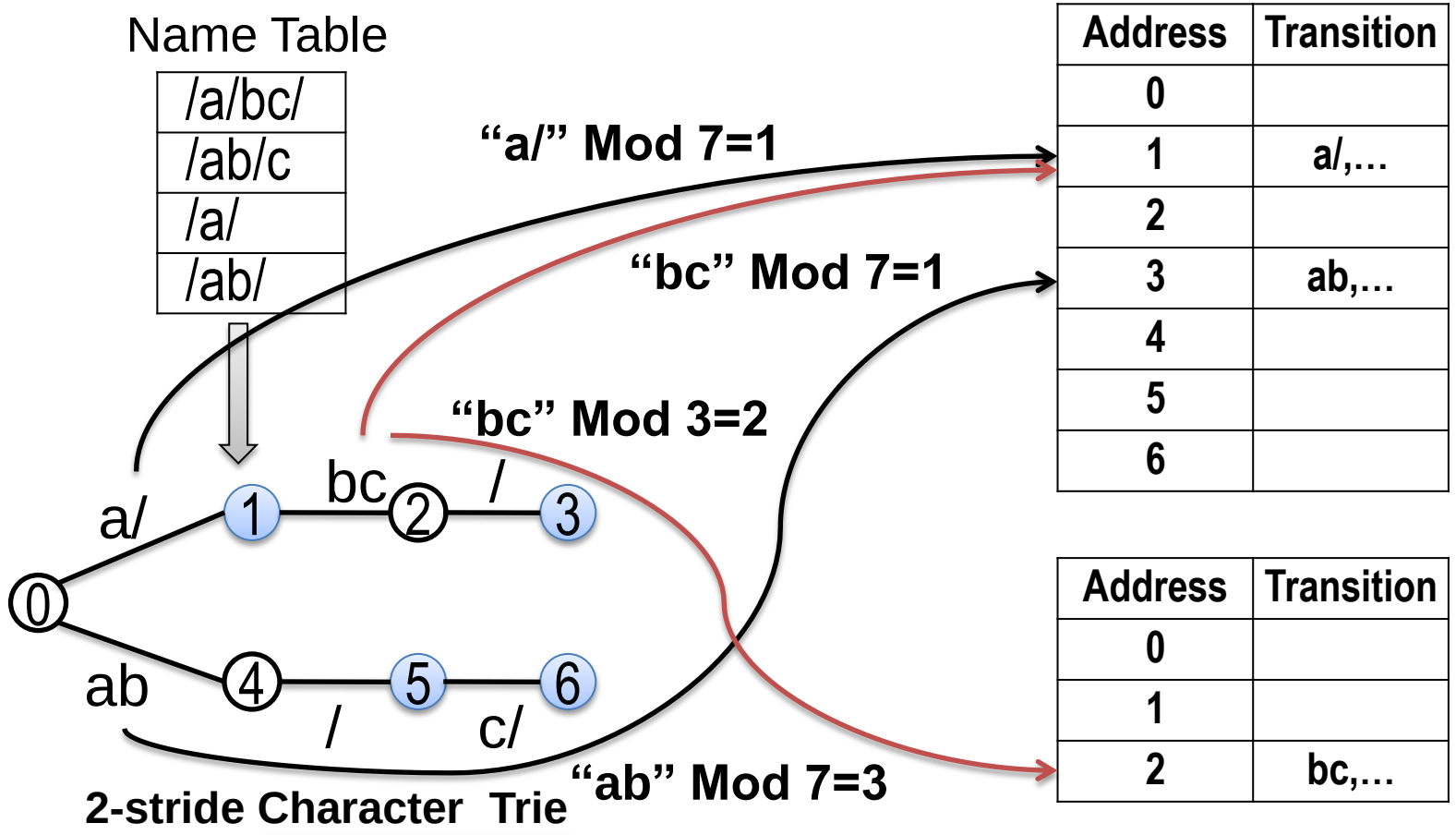
d-stride character trie, every state may have 2^{8d} *transitions at most.*

ATA cannot support multiple Stride Character Trie

Tsinghua University

—Algorithm & Data Structure

Multiple Stride Character Trie ⟶ Multi-ATA



- MATA
  - Advantage
    - **Improve lookup throughput**: one state transition consumes multiple characters, and each state transition requires only one memory access
    - Further compress memory space
    - Small ATAs in MATA are easier to build and manage
    - Support fast incremental update

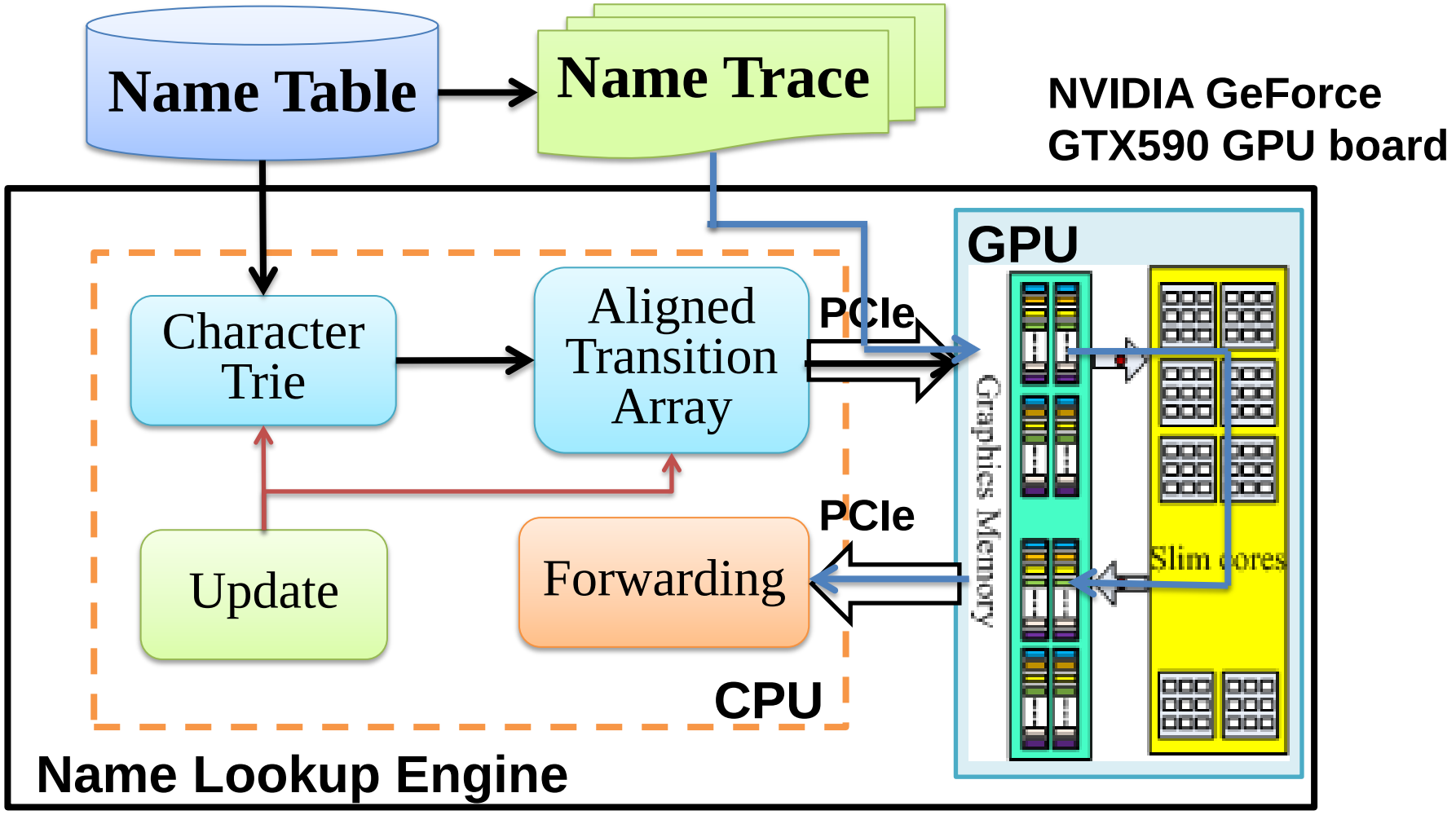
清華大學 Tsinghua University

——Outline

1. Introduction
2. Name Lookup: Algorithm and Data Structure
3. Implementation
4. Experimental Results
5. Conclusion

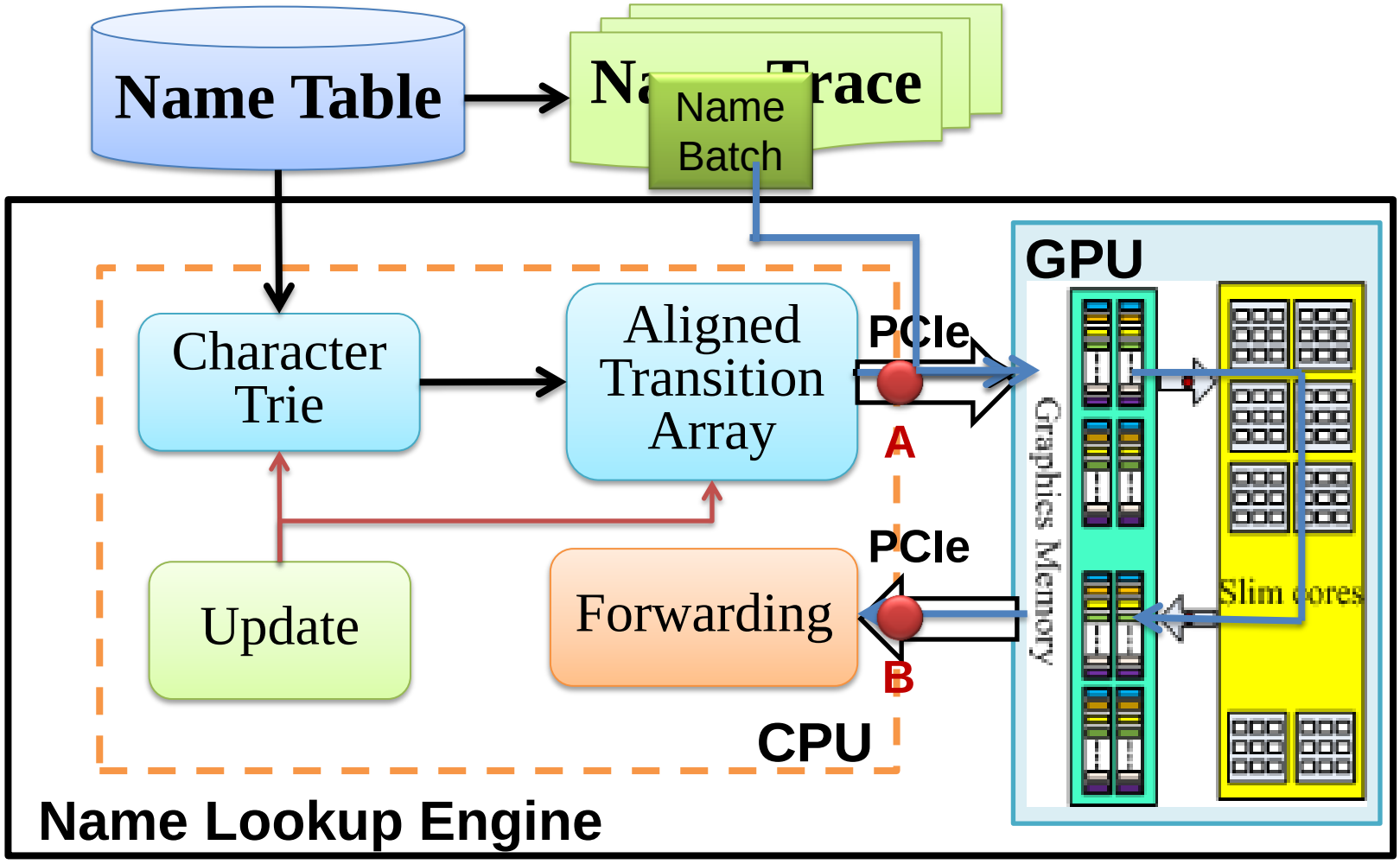
Tsinghua University

—Name Lookup Engine Framework



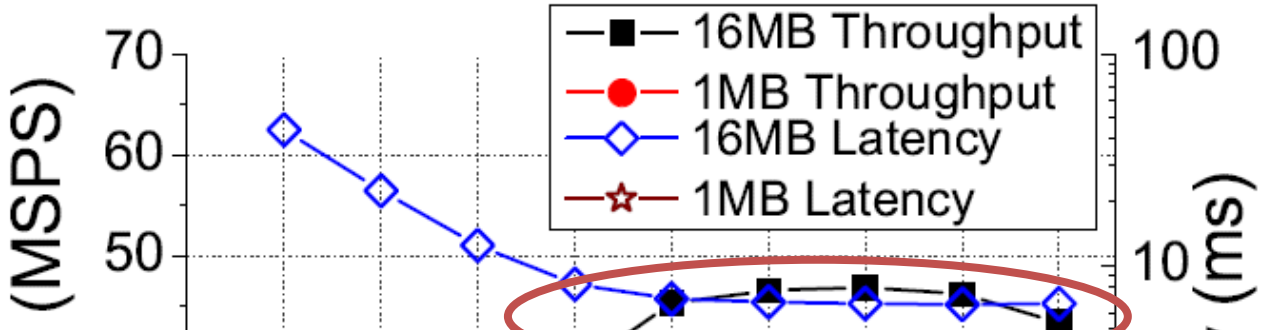
Tsinghua University

—Latency Optimization

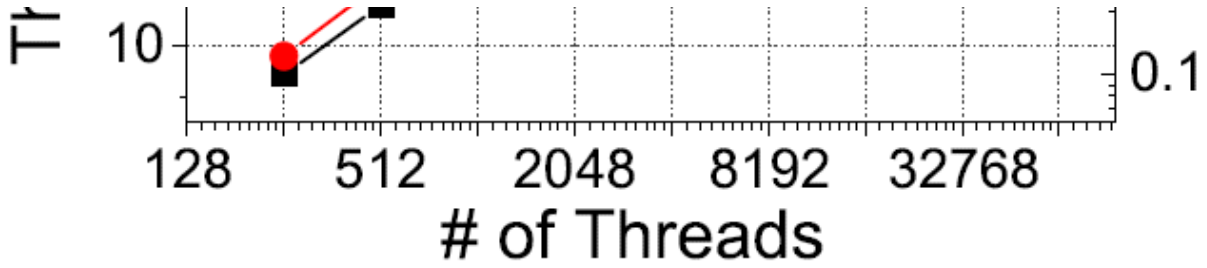


—Latency Optimization

- Batch Size: 16MB vs. 1MB

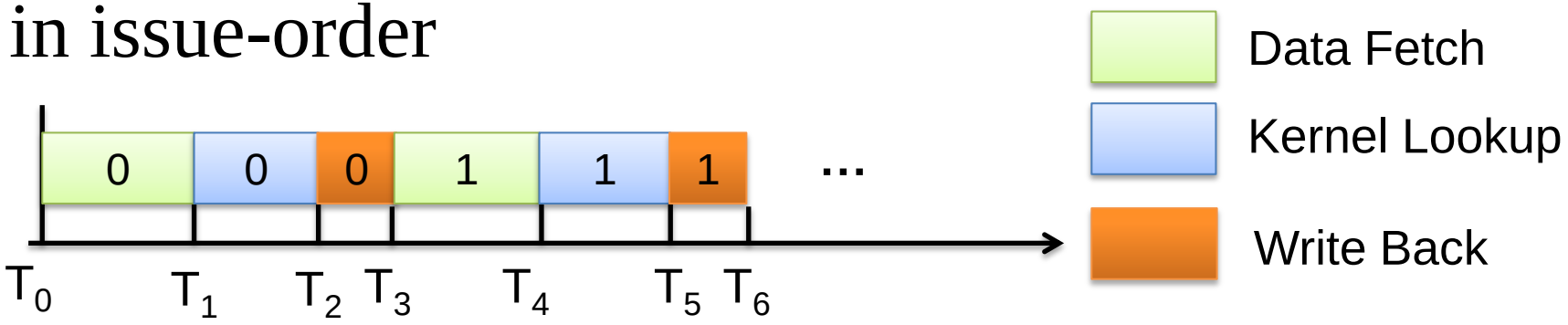


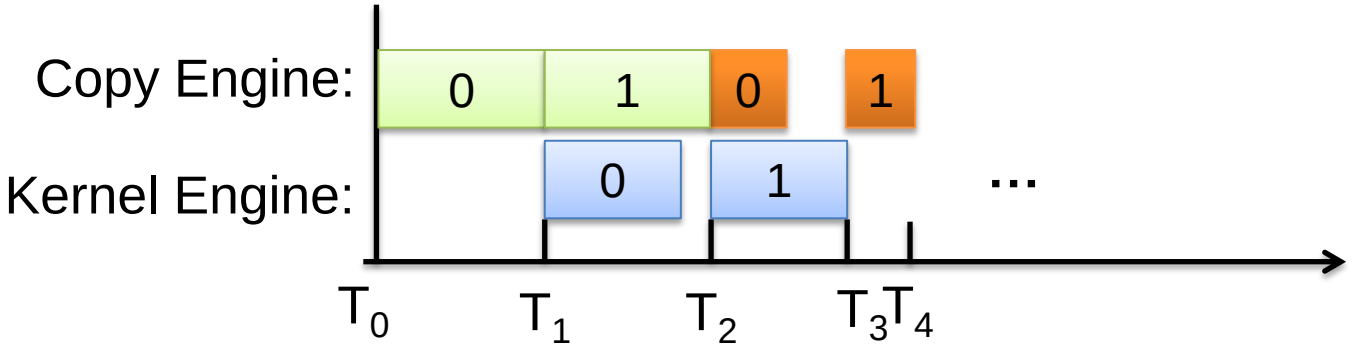
How can we reduce name lookup latency while keeping high throughput?



—Latency Optimization

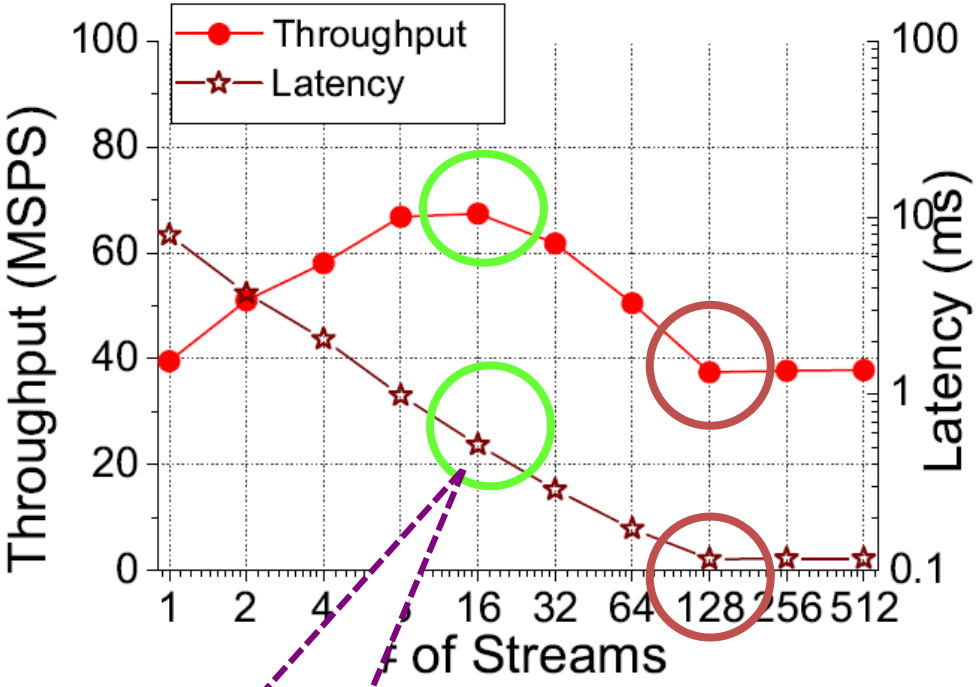
- CUDA Stream:
  - a stream is a sequence of operations that execute in issue-order





—Latency Optimization

- CUDA Streams effectively reduce latency



We still need to reduce the delay

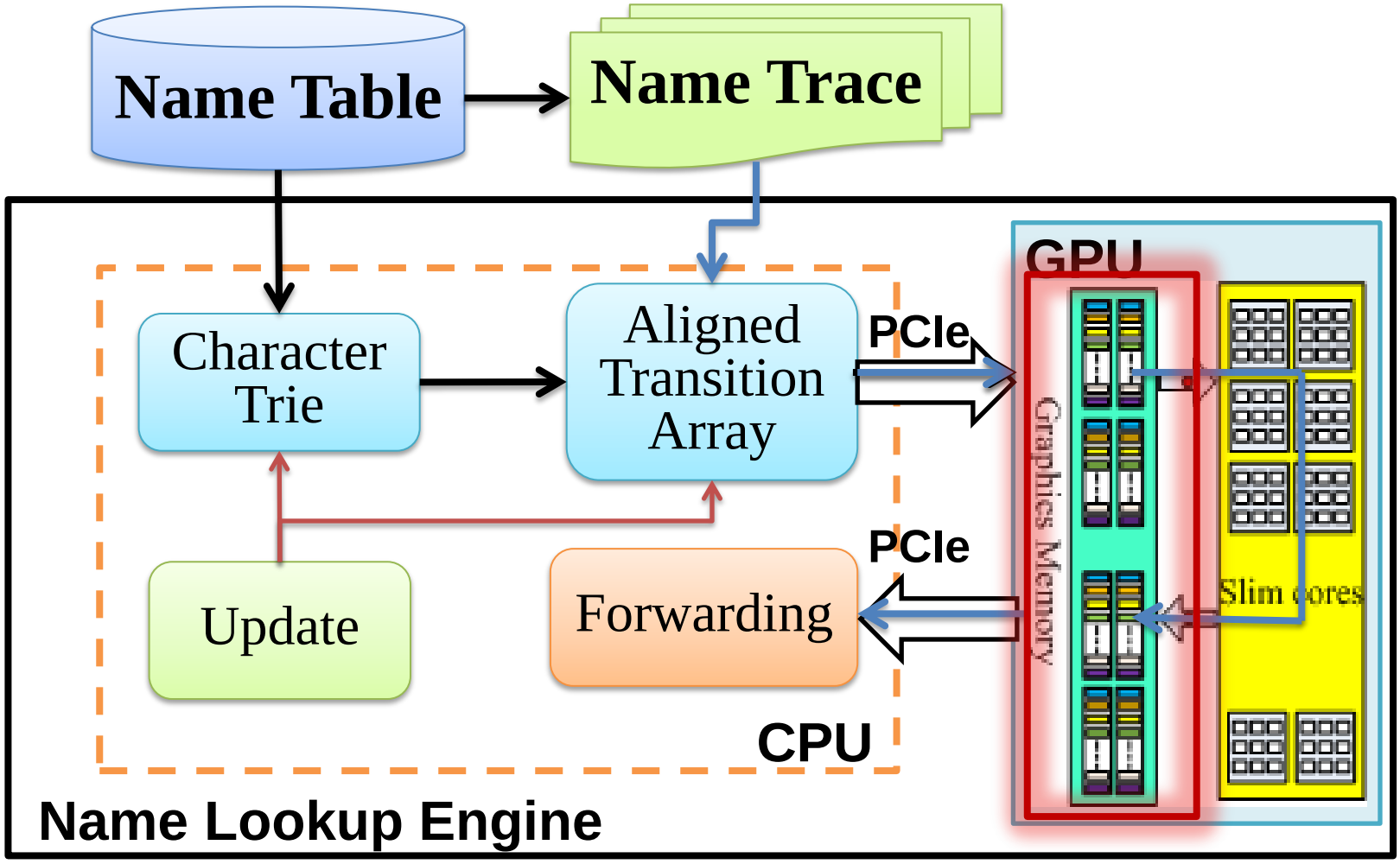
$$T = 2t_{start} * N + \frac{M_{batch} + M_{result}}{S_{PCIe}} + \frac{M_{batch}}{N * S_{kernel}}$$

$$f'(N) = 0$$

$$N = \sqrt{\frac{M_{batch}}{2t_{start} * S_{kernel}}}$$

Tsinghua University

—GPU Memory Access Optimization



Tsinghua University

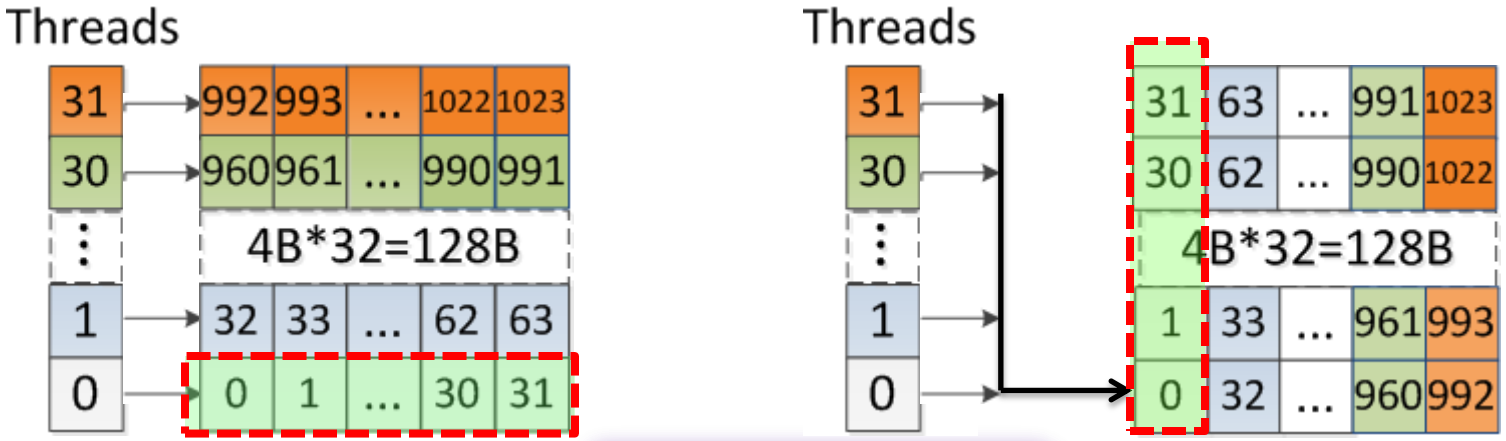
—GPU Memory Access Optimization

- GPU: Single-Instruction Multiple-Data (SIMD)
  - 32 threads are organized as a *Warp;*
  - 32 threads in a Warp synchronously run in SIMD manner;
- GPU Memory:
  - Partition into 128-byte blocks;
  - Every memory access fetches a 128-byte block;

Tsinghua University

—GPU Memory Access Optimization

- Each 128-bytes block stores a name
  - Problem: when the 32 threads simultaneously read the first piece of data from each of the names they are processing, resulting in 32 separate memory accesses.
- **Interweaved Layout**
  - **A name is divided into 32 pieces;**
  - **32 pieces from 32 names are stored in one 128-byte block**



Tsinghua University

——Outline

1. Introduction
2. Name Lookup: Algorithm and Data Structure
3. Implementation
4. Experimental Results
5. Conclusion

——Experimental Results

- Platform: A commodity PC

Item	Specification
Motherboard	ASUS Z8PE-D12X (INTEL S5520)
CPU	Intel Xeon E5645×2 (6 cores, 2.4GHz)
RAM	DDR3 ECC 48GB (1333MHz)
GPU	NVIDIA GTX590 (2×512 cores, 2×1536MB)

- Name Table
  - Download from DMOZ website: 3M
  - Crawl from Internet: 10M
- Name Trace
  - Average workload: random name prefix + suffix
  - Heavy workload: the longest 10% name prefix + suffix

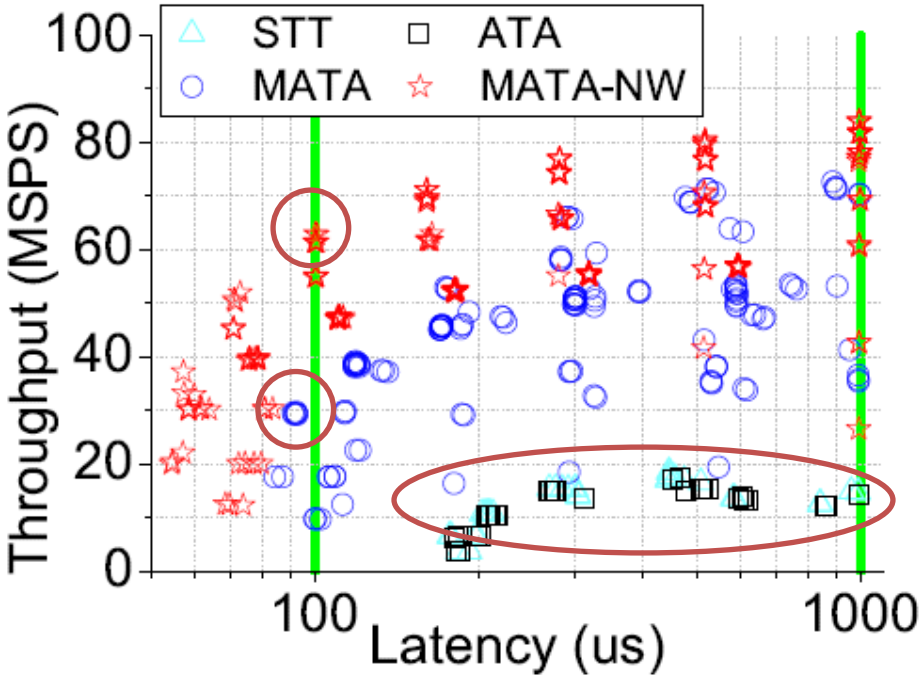
——Experimental Results

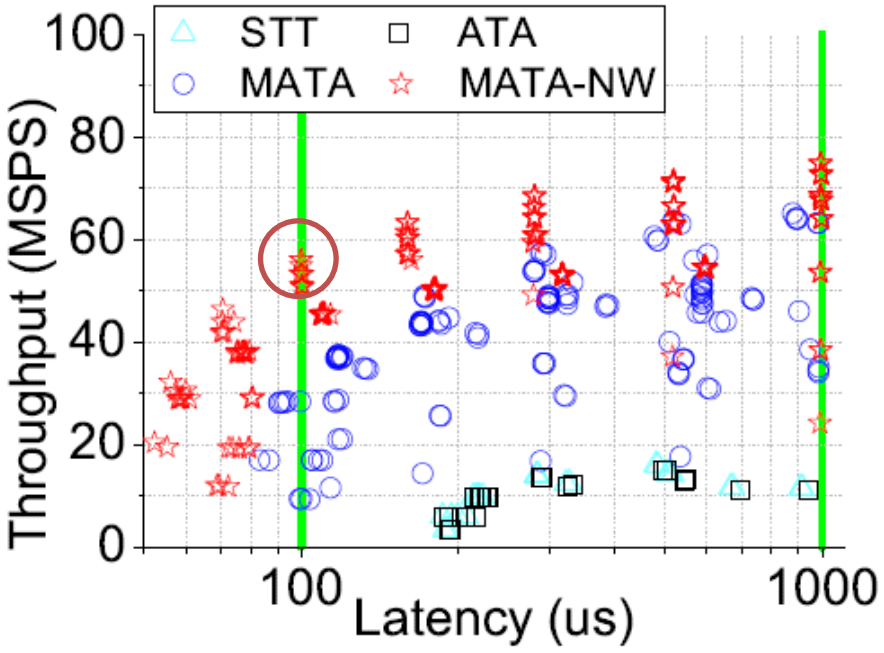
- Memory Space
  - 3M name table
    - ATA vs STT: 101×
    - MATA vs STT: **130** ×
  - 10M name table
    - ATA vs STT: 102×
    - MATA vs STT: **142**×

Tsinghua University

——Experimental Results

- Lookup Speed (Million Searches per Second, MSPS)
  - 100K, Average Workload
  - 100K, Heavy Workload

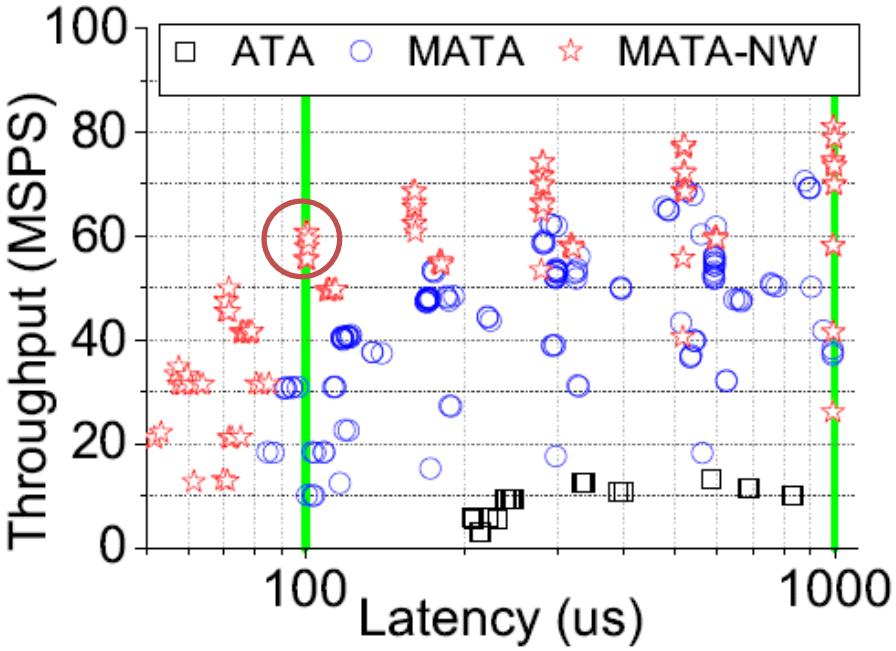


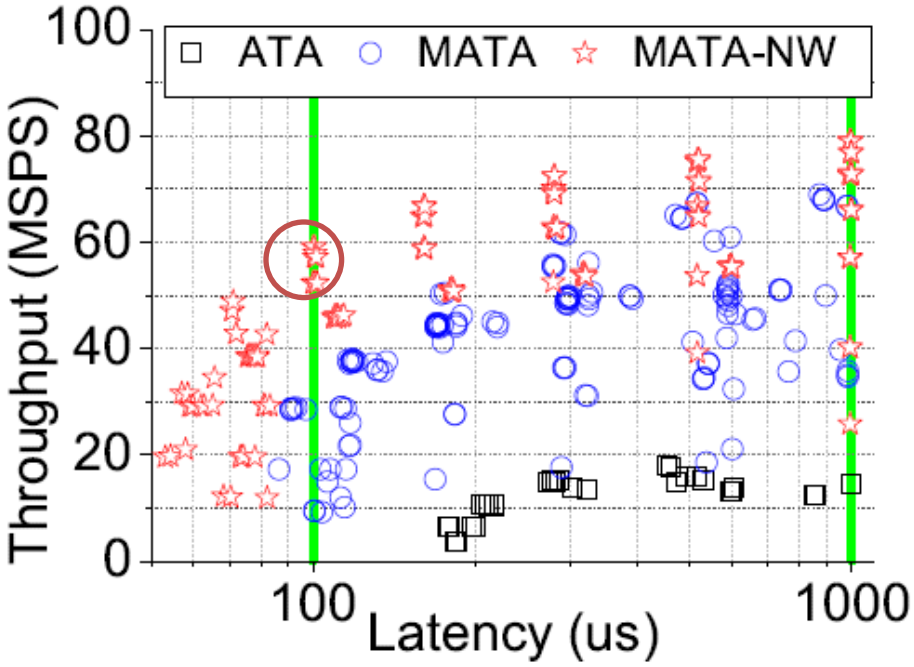


Tsinghua University

——Experimental Results

- Lookup Speed (Million Searches per Second, MSPS)
  - 10M, Average Workload
  - 10M, Heavy Workload

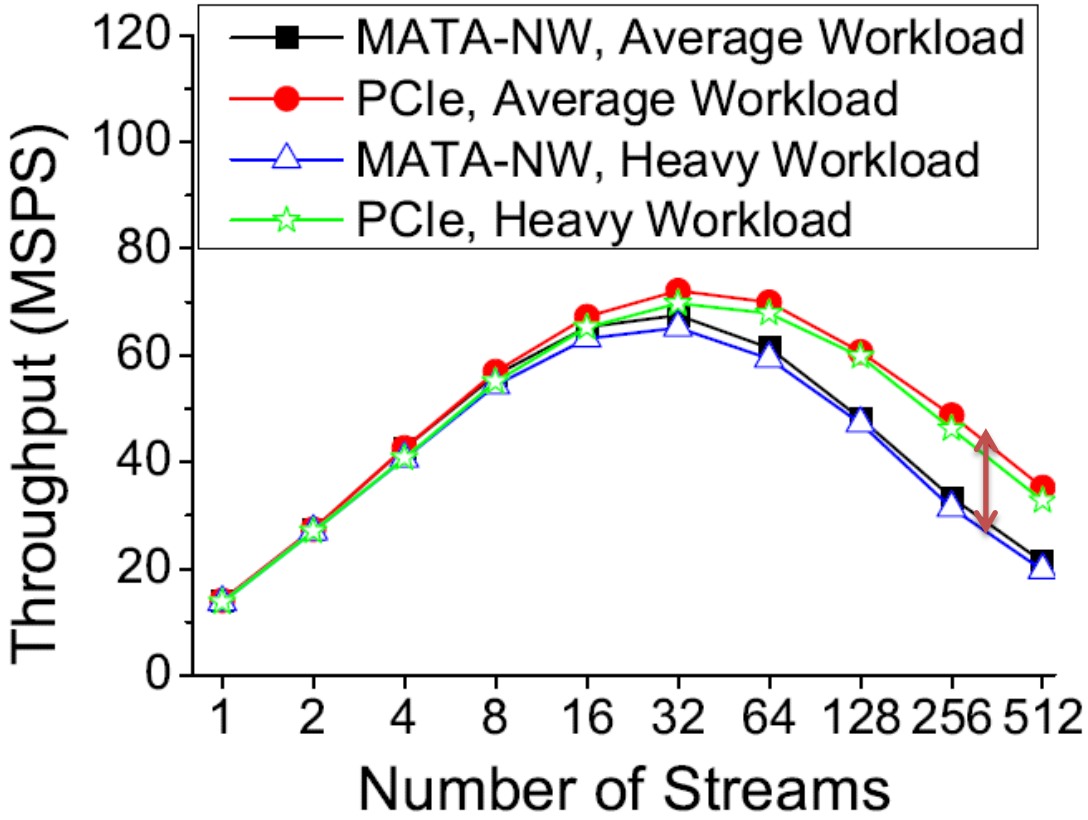




清華大學 Tsinghua University

——Experimental Results

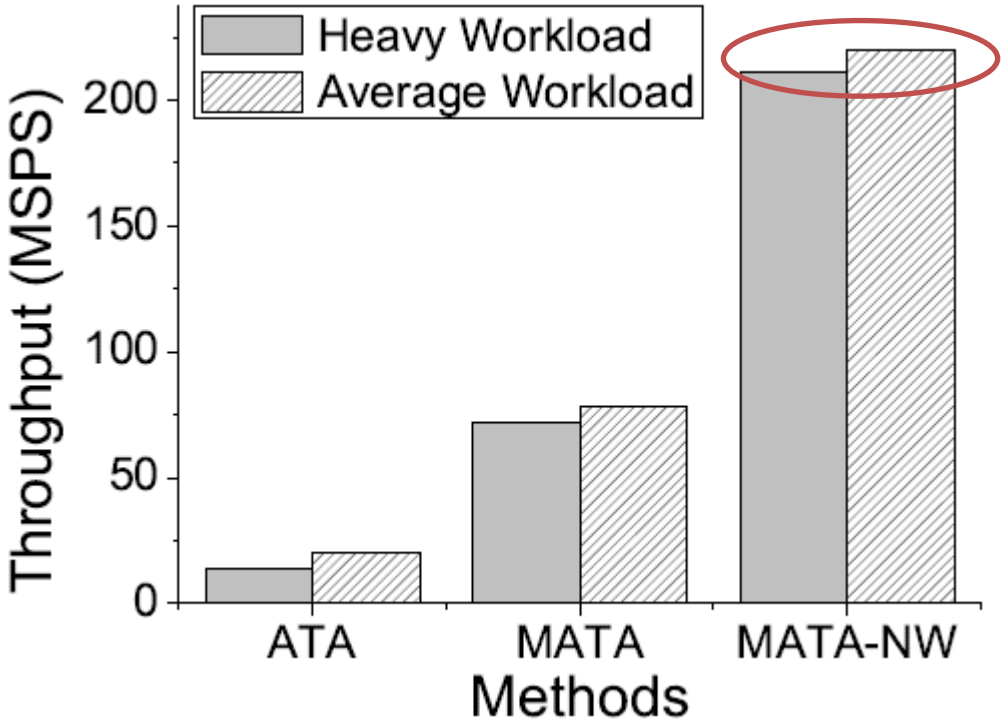
- Which is the bottleneck of name lookup engine?
  - PCIe bus or GPU kernel?



Tsinghua University

——Experimental Results

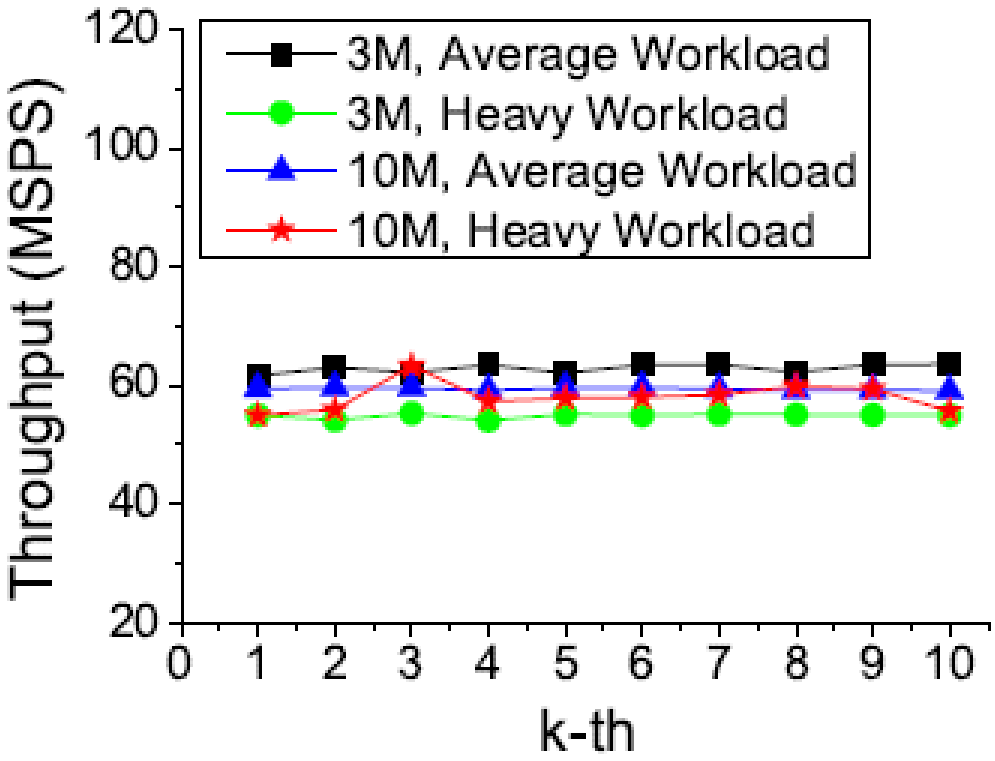
- Which is the bottleneck of name lookup engine?
  - PCIe bus or GPU kernel?



清華大學 Tsinghua University

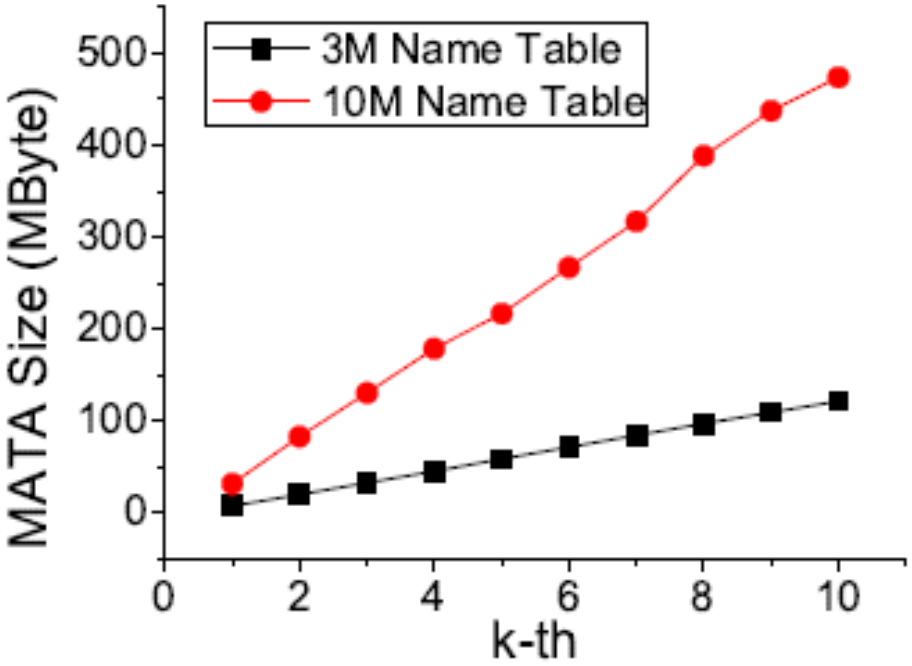
——Experimental Results

- Scalability
  - Lookup speed
  - Memory
  - Latency



——Experimental Results

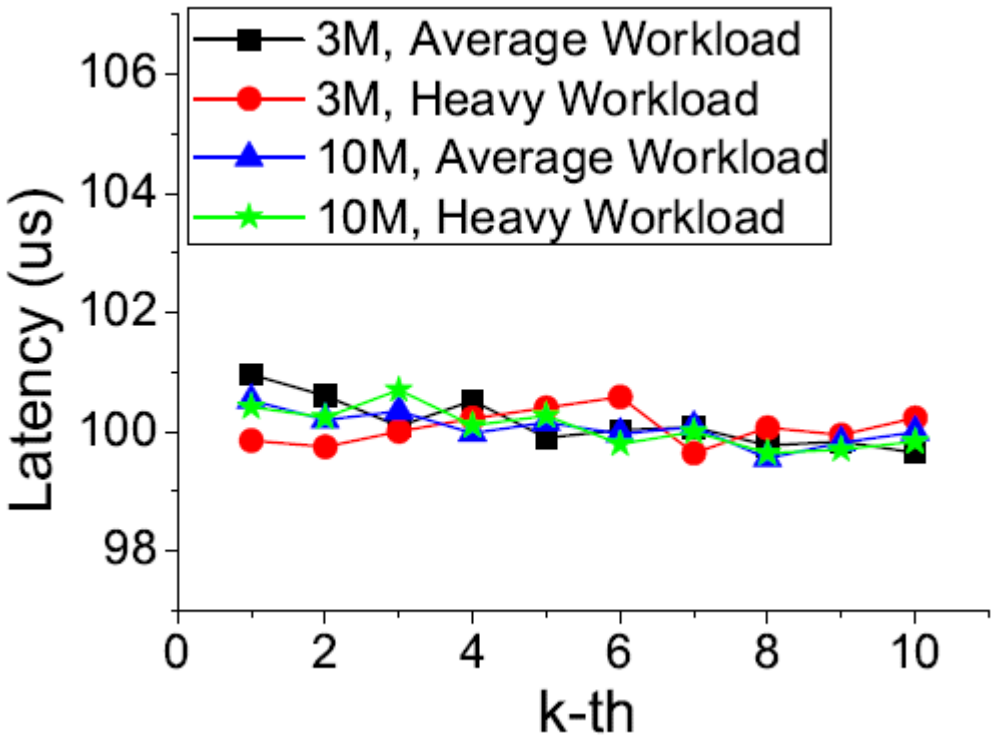
- Scalability
  - Lookup speed
  - Memory
  - Latency



清華大學 Tsinghua University

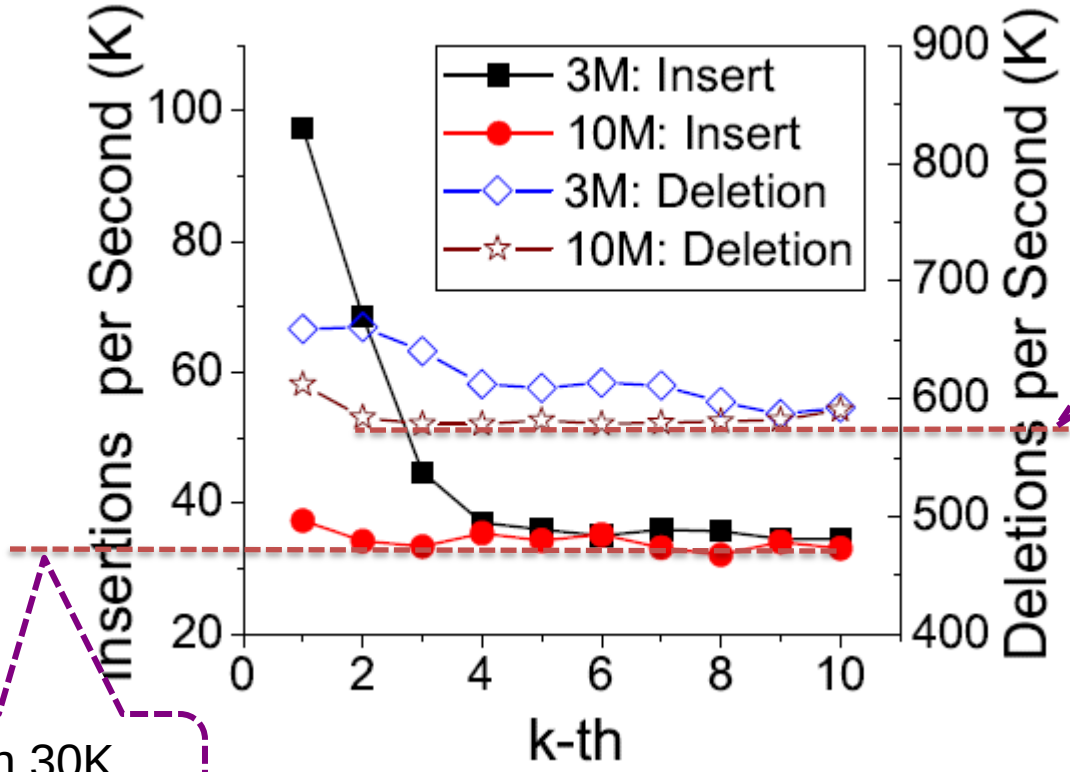
——Experimental Results

- Scalability
  - Lookup speed
  - Memory
  - Latency



——Experimental Results

- Update



Nearly 600K deletion per second

More than 30K insertion per second

清華大學 Tsinghua University

——Outline

1. Introduction
2. Name Lookup: Algorithm and Data Structure
3. Implementation
4. Experimental Results
5. Conclusion

清華大學 Tsinghua University

——Conclusion

1. ***MATA*** is proposed to compress memory space and improve name lookup speed
2. Implement a wire speed name lookup engine based on a commodity PC installed with a GTX590 GPU board
3. Extensive experiments demonstrate:
   - Name lookup speed: 63.52 MSPS,>100 Gbps wire-speed
   - Latency: <100us
   - Memory: compress >100×
   - Good Scalability

Thanks

Q & A