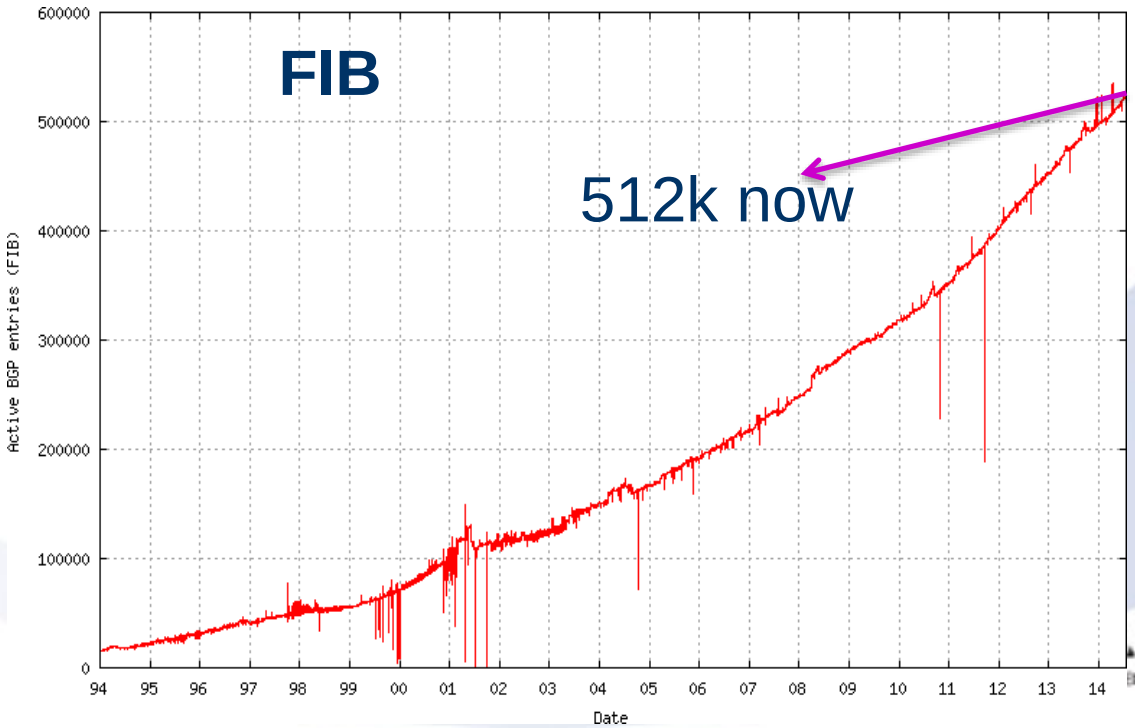
中科院计算所
INSTITUTE OF COMPUTING TECHNOLOGY,CAS

Guarantee IP Lookup Performance with FIB Explosion

Tong Yang(ICT), Gaogang Xie(ICT), Yanbiao Li(HNU), Qiaobin Fu(ICT)
Alex X. Liu(MSU), Qi Li(ICT), Laurent Mathy(ULG)

Performance Issue in IP Lookup

- **FIB increasing: 15% per year; FIB size: 512,000**
- **512k bug** : In 2014.8, Cisco says that web browsing speeds could slow over the next week as old hardware is upgraded to handle the 512K FIB.



Motivation

- On-chip vs. Off-chip memory. 10 times faster, but limited in size.
- With FIBs increasing, for almost all packets

Constant yet small footprint for FIB: On-chip Memory

+

Constant yet fast lookup speed: Low Time Complexity

Ideal IP Lookup Algorithm

State-of-the-art

- Achieving constant IP lookup time
  - TCAM-based
  - Trie pipeline using FPGA
  - full-expansion
  - DIR-24-8
- Achieving small memory
  - Based on Bloom Filter
  - Level compression, path compression
  - LC-trie

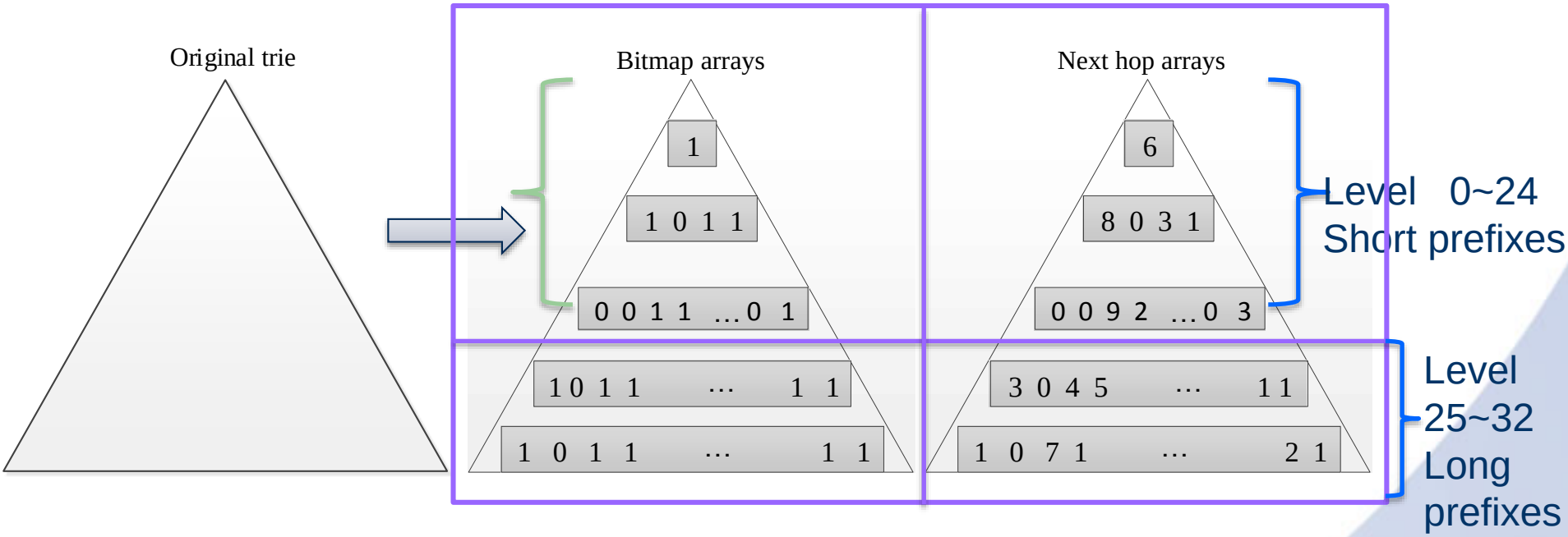
How to satisfy both constant lookup time and small on-chip memory usage?

SAIL Framework

- Observation: almost all packets hit 0~24 prefixes
- Two Splitting
  - Splitting lookup process
  - Splitting prefix length

	Finding prefix length	Finding next hop
Prefix length 0~24	On-chip	Off-chip
Prefix length 25~32	Off-chip	Off-chip

Splitting



Bit Maps 0-24

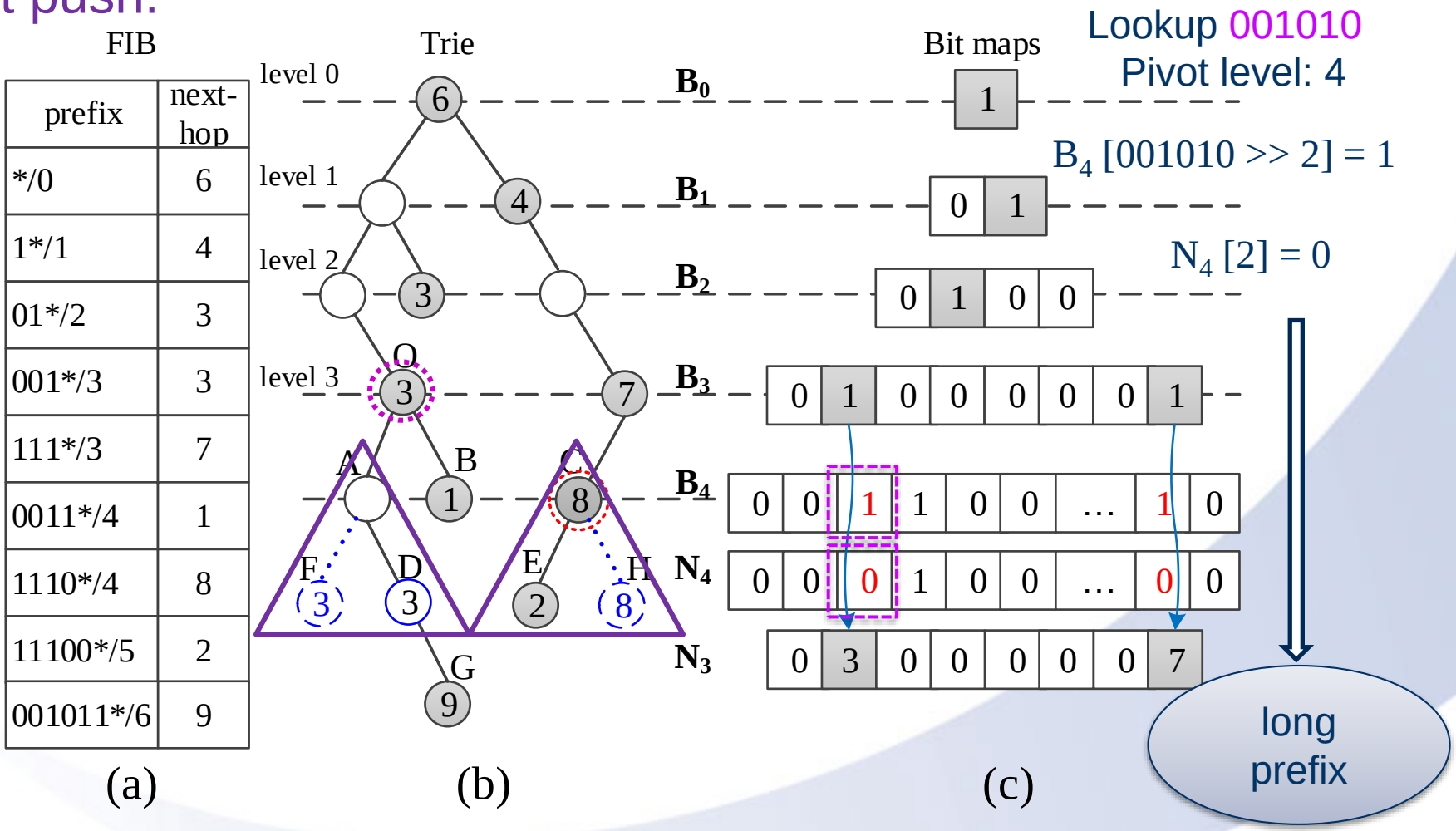
On-Chip

$$\sum_{i=0}^{24} 2^i = 4MB$$

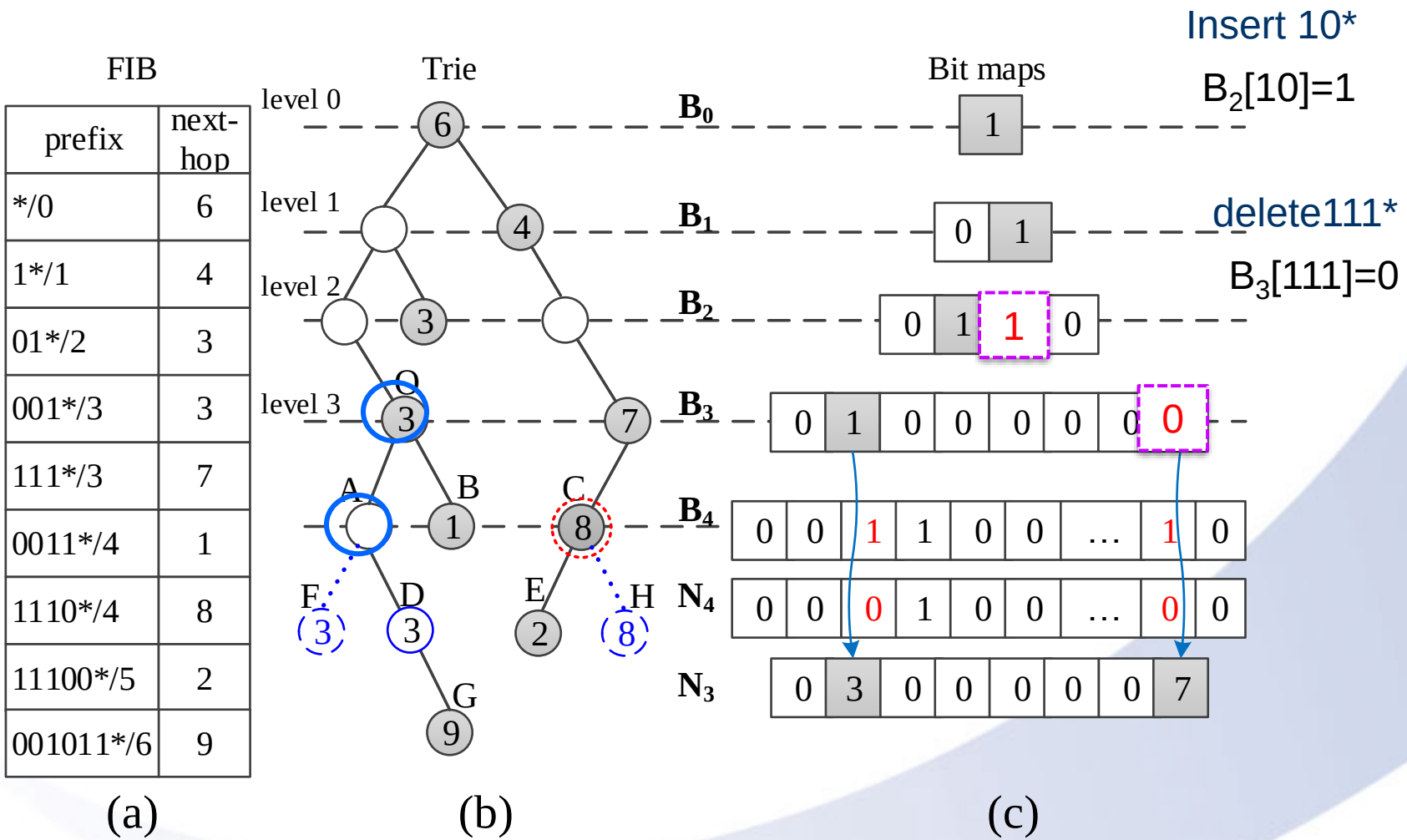
How to avoid searching both short and long prefixes?

Pivot Pushing & Lookup

Pivot push:



Update of SAIL_B



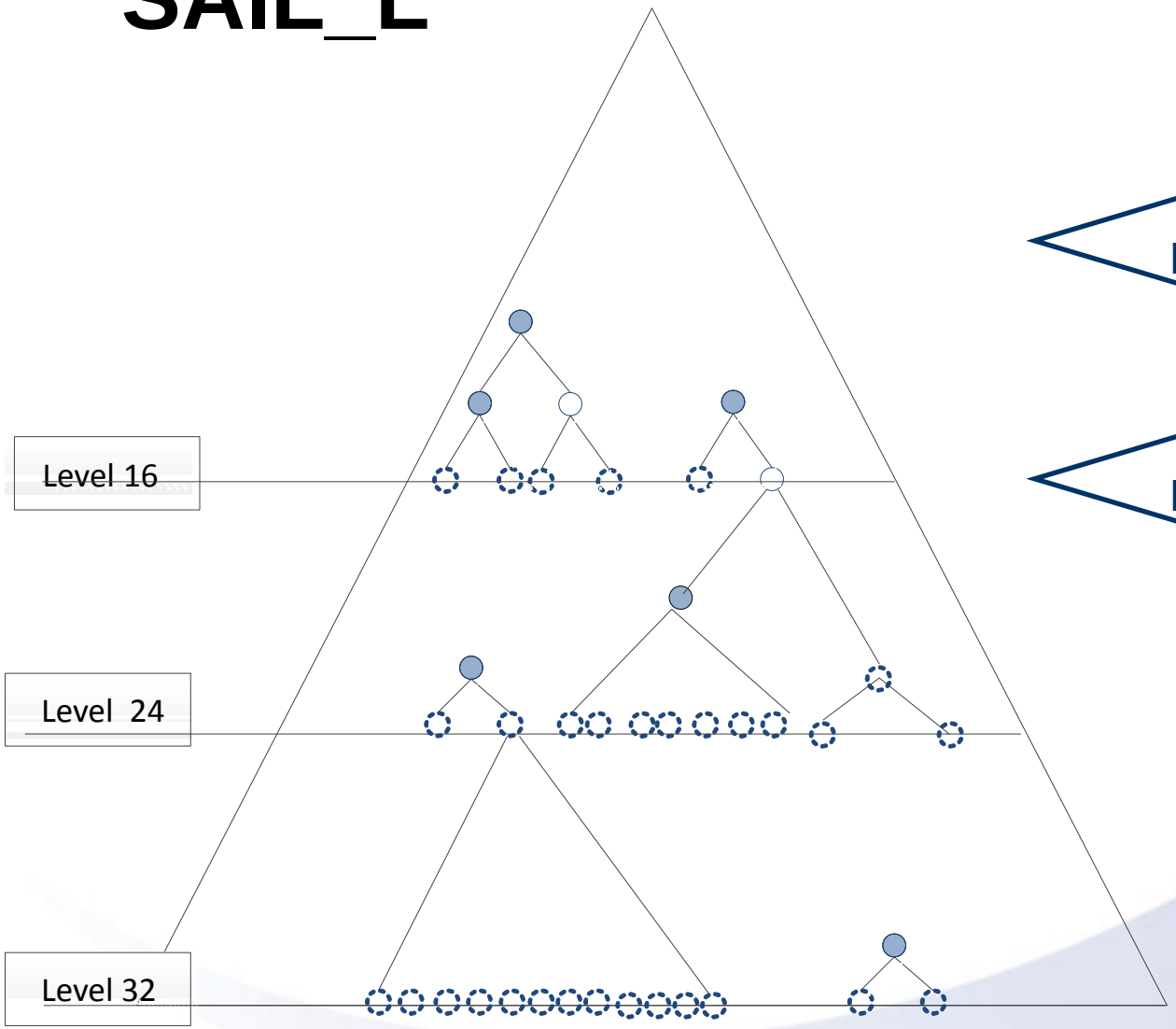
prefix	next-hop
*/0	6
1*/1	4
01*/2	3
001*/3	3
111*/3	7
0011*/4	1
1110*/4	8
11100*/5	2
001011*/6	9

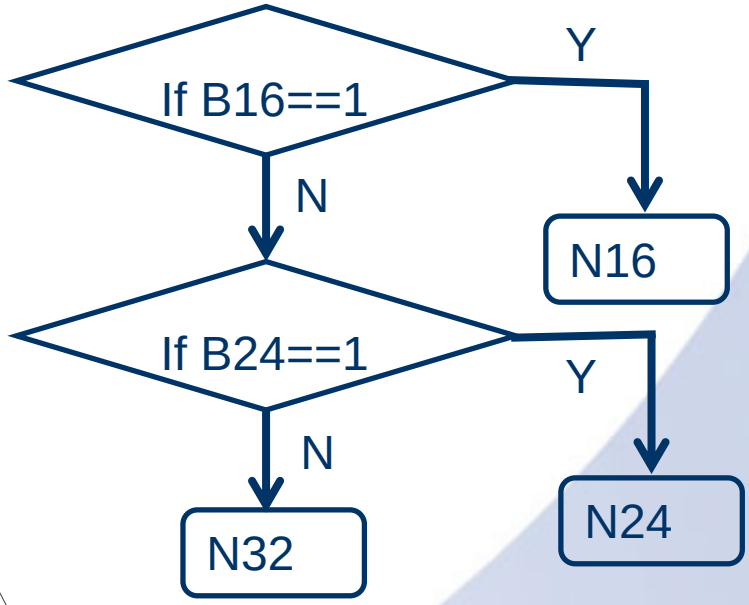
changing 001*, or inserting 0010*
only need to update off-chip tables

Optimization

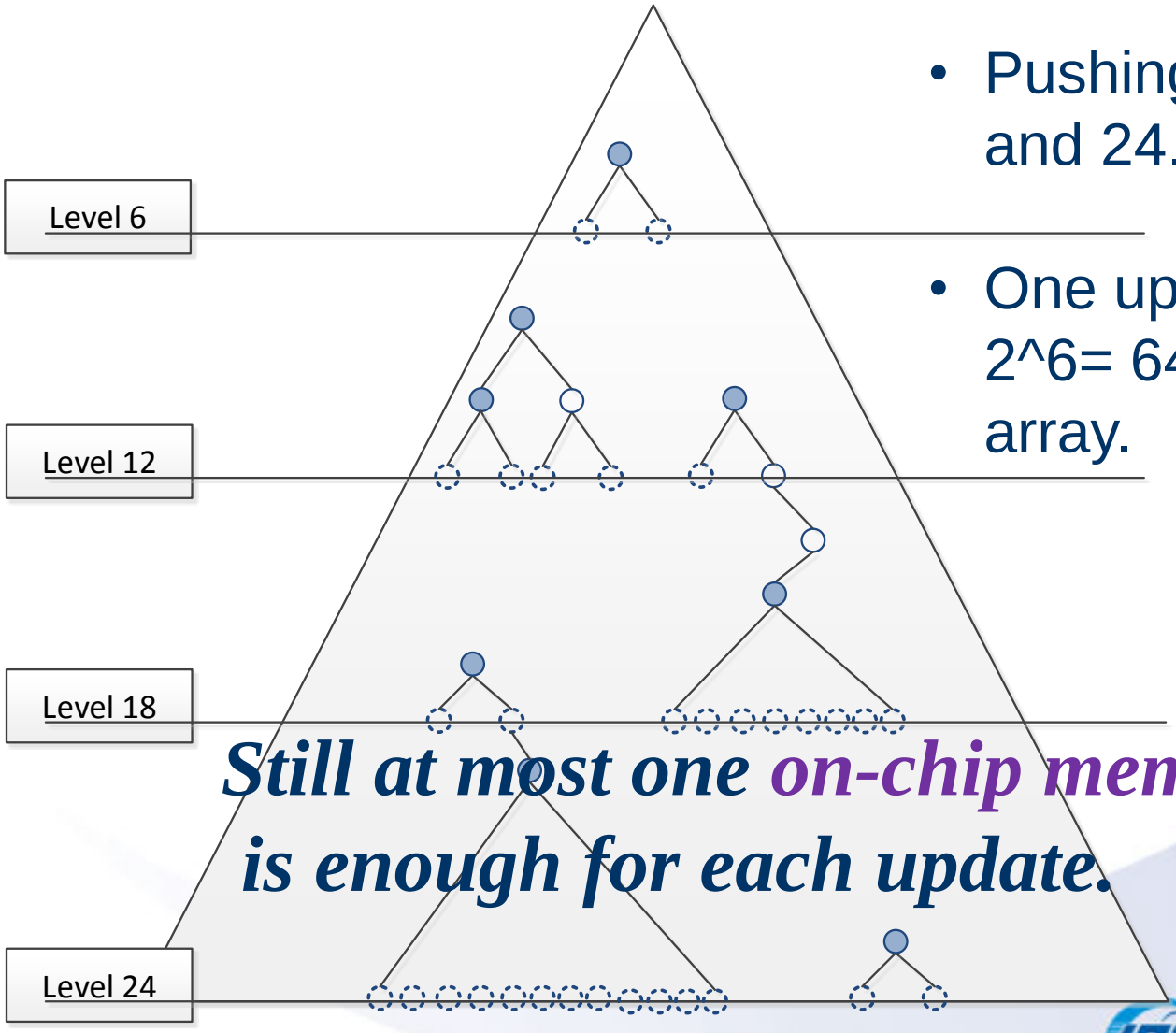
- **SAIL_B**
  - **Lookup: 25 on-chip memory accesses in worst case**
  - **Update: 1 on-chip memory access**
- **Lookup Oriented Optimization (SAIL_L)**
  - **Lookup: 2 on-chip memory accesses in worst case**
  - **Update: unbounded, low average update complexity**
- **Update Oriented Optimization (SAIL_U)**
  - **Lookup: 4 on-chip memory accesses in worst case**
  - **Update: 1 on-chip memory access**
- **Extension: SAIL for Multiple FIBs (SAIL_M)**

SAIL_L





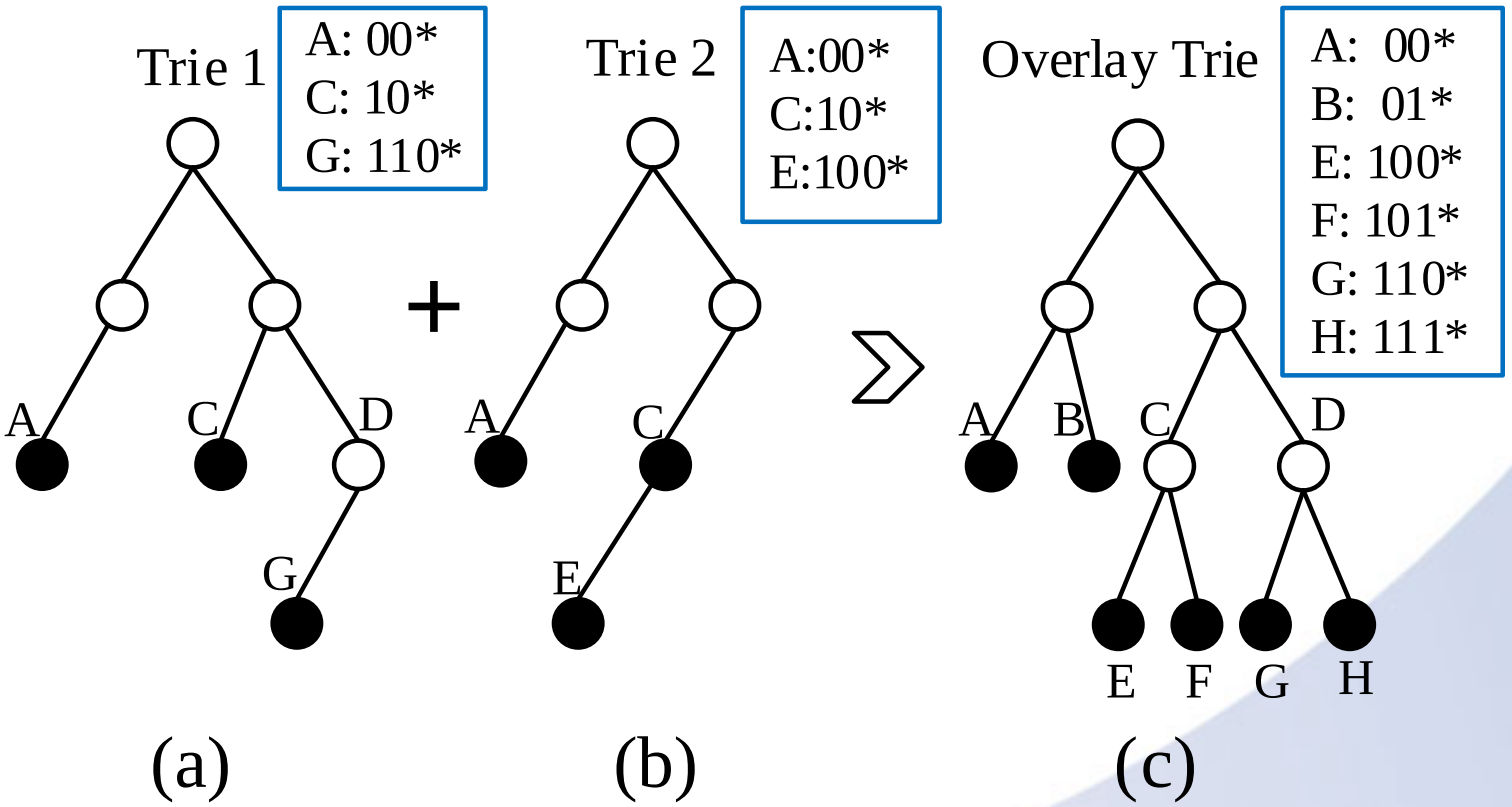
SAIL_U



- Pushing to levels 6, 12, 18, and 24.
- One update at most affects 2^6= 64 bits in the bitmap array.

Still at most one on-chip memory access is enough for each update.

SAIL_M



SAILs in worst case

	On-Chip Memory	Lookup (on-chip)	Update (on-chip)
SAIL_B	= 4MB	25	1
SAIL_L	≤ 2.13MB	2	Unbounded
SAIL_U	≤ 2.03MB	4	1
SAIL_M	≤ 2.13MB	2	Unbounded

Worst case: 2 off-chip memory accesses for lookup

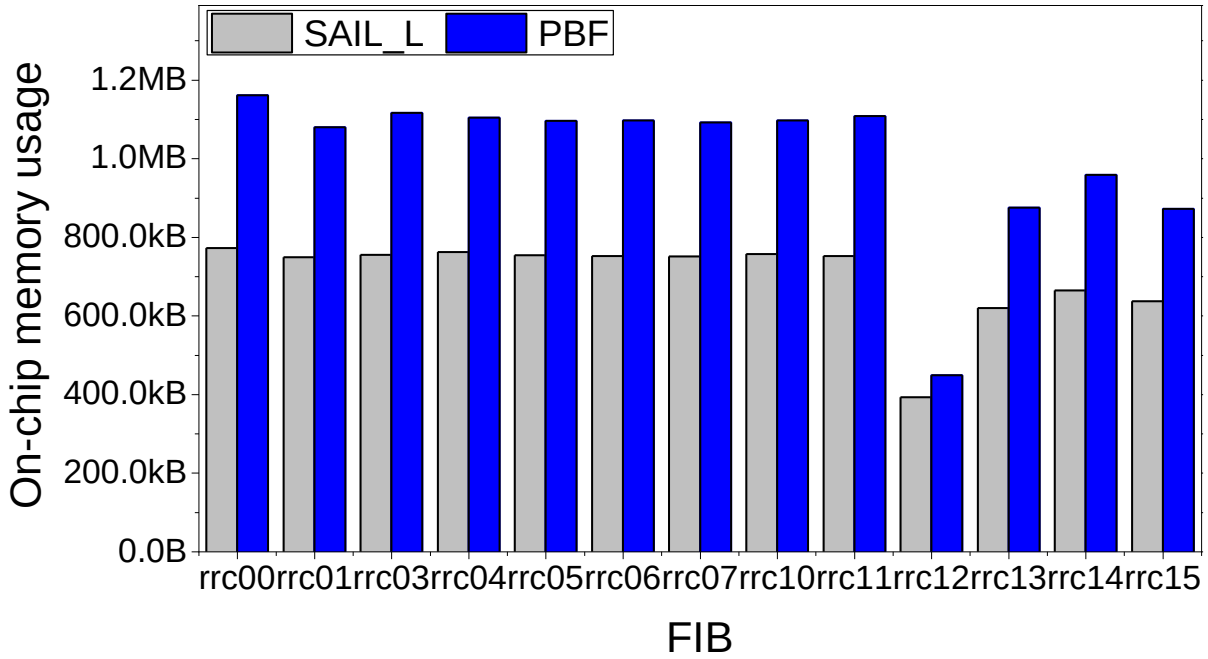
Implementations

- **FPGA:** Xilinx ISE 13.2 IDE; Xilinx Virtex 7 device; On-chip memory is 8.26MB
  - **SAIL_B, SAIL_U, and SAIL_L**
- **Intel CPU:** Core(TM) i7-3520M 2.9 GHz; 64KB L1, 512KB L2, 4MB L3; DRAM 8GB
  - **SAIL_L and SAIL_M**
- **GPU:** NVIDIA GPU (Tesla C2075, 1147 MHz, 5376 MB device memory, 448 CUDA cores), Intel CPU (Xeon E5-2630, 2.30 GHz, 6 Cores).
  - **SAIL_L**
- **Many-core**: TLR4-03680, 36 cores, each 256K L2 cache.
  - **SAIL_L**

Evaluation

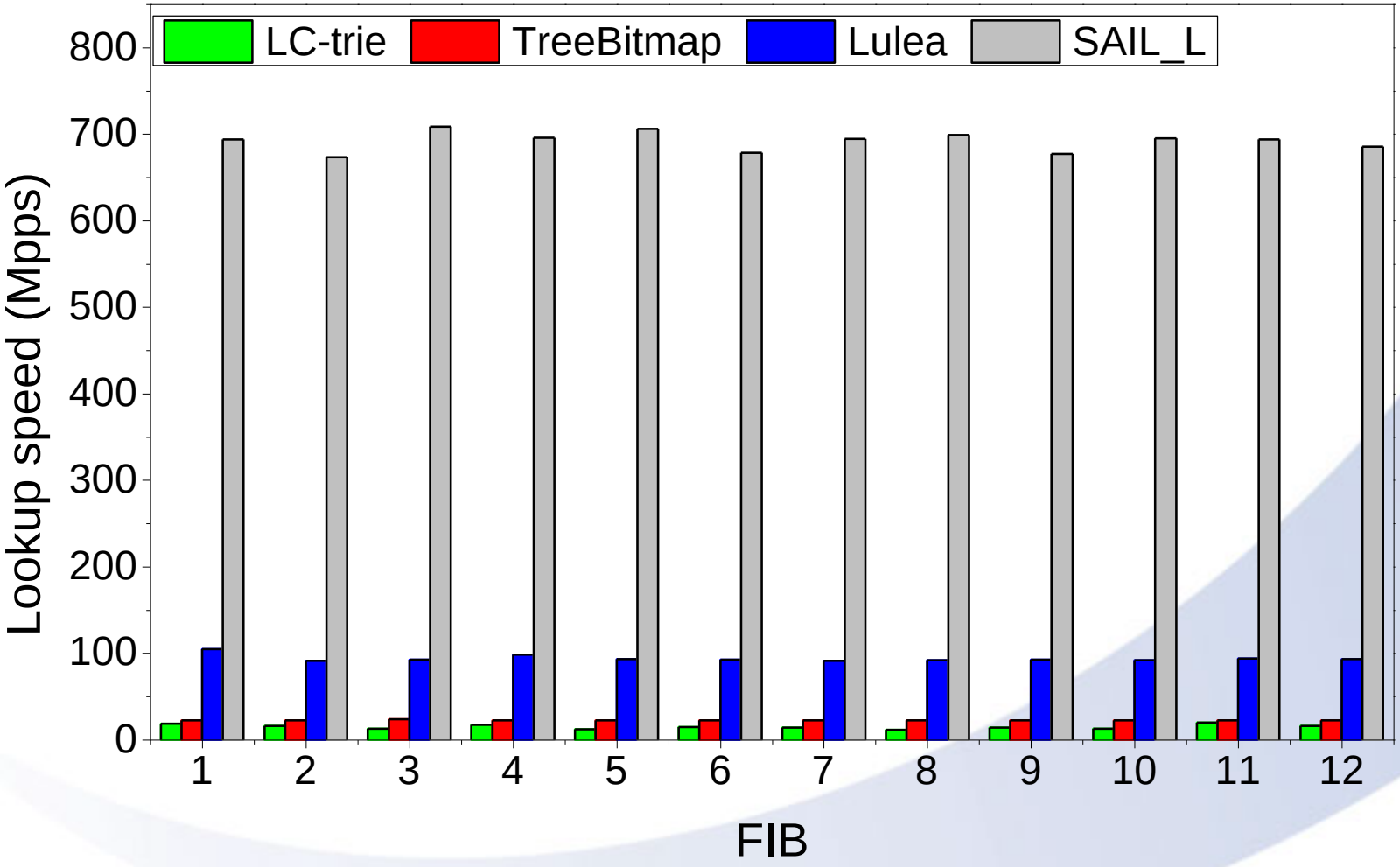
- **FIBs**
  - **Real FIB from a tier-1 router in China**
  - **18 real FIBs from www.ripe.net**
- **Traces**
  - **Real packet traces from the same tier-1 router**
  - **Generating random packet traces**
  - **Generating packer traces according to FIBs**
- **Comparing with**
  - **PBF** [sigcomm 03]
  - **LC-trie** [applied in Linux Kernel]
  - **Tree Bitmap**
  - **Lulea** [sigcomm 97 best paper]

FPGA Simulation

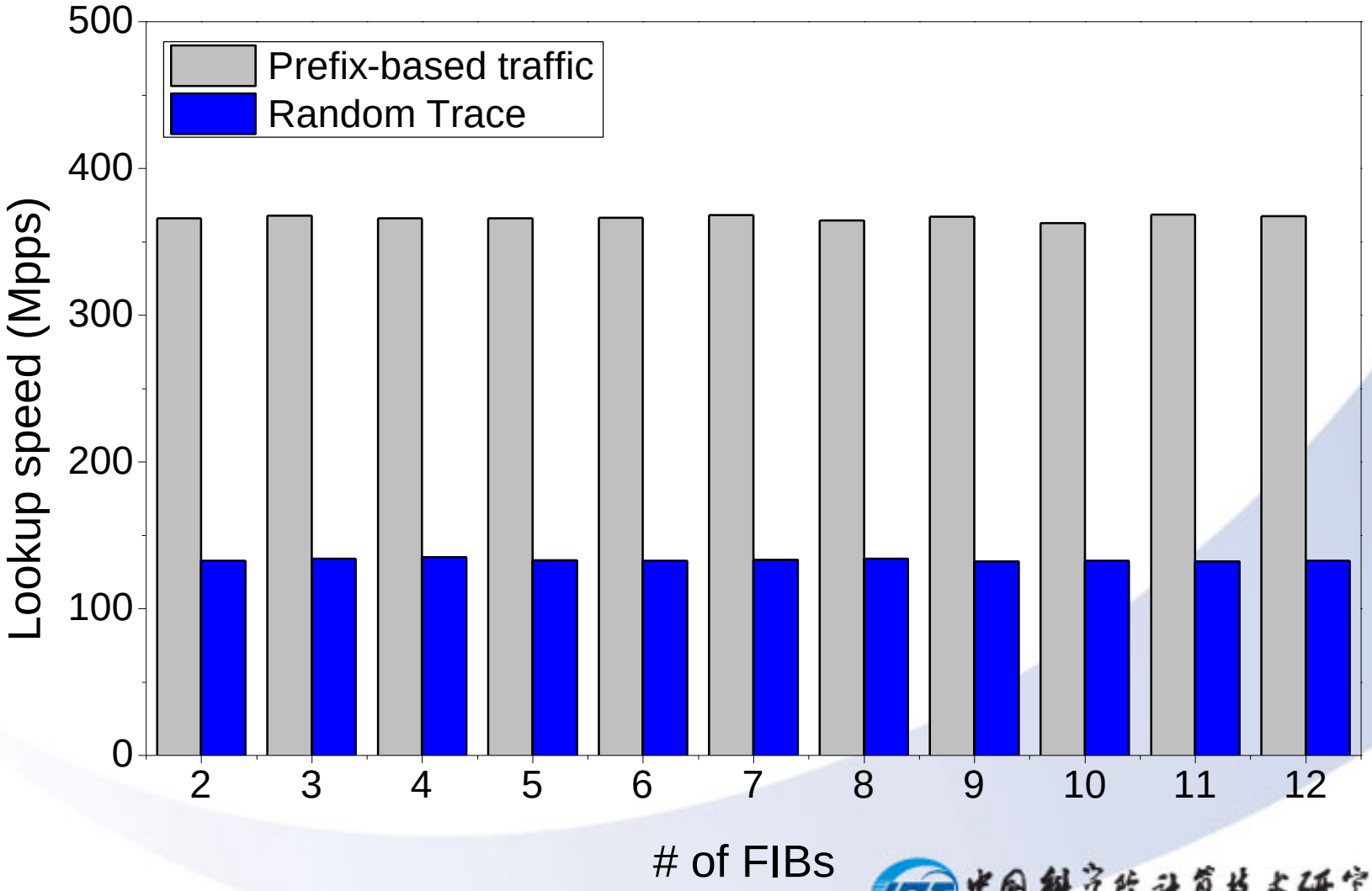


SAIL Algorithms	Lookup Speed	Throughput
SAIL_B	351Mpps	112Gbps
SAIL_U	405Mpps	130Gbps
SAIL_L	479Mpps	153Gbps

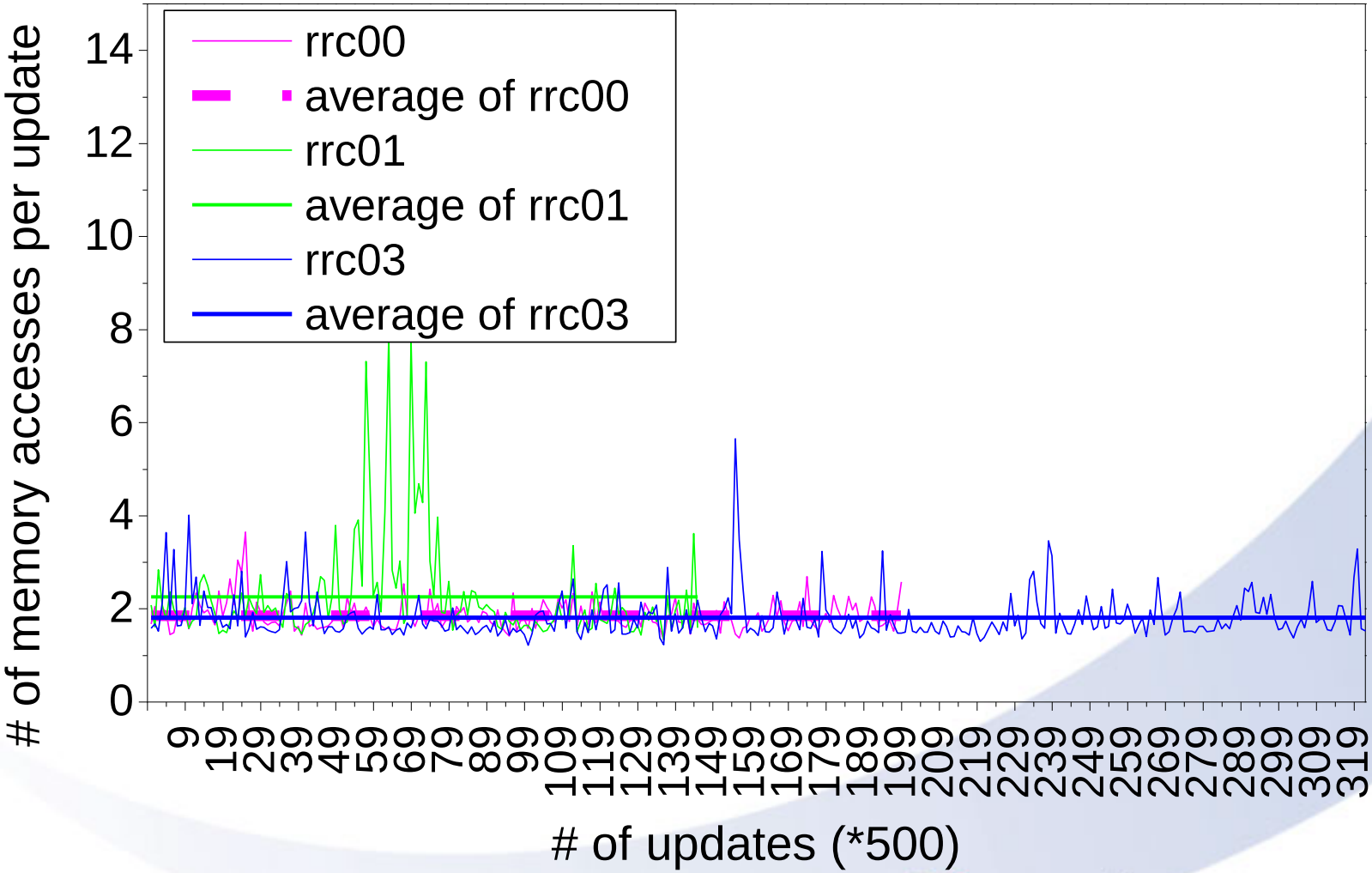
Intel CPU: real FIB and traces



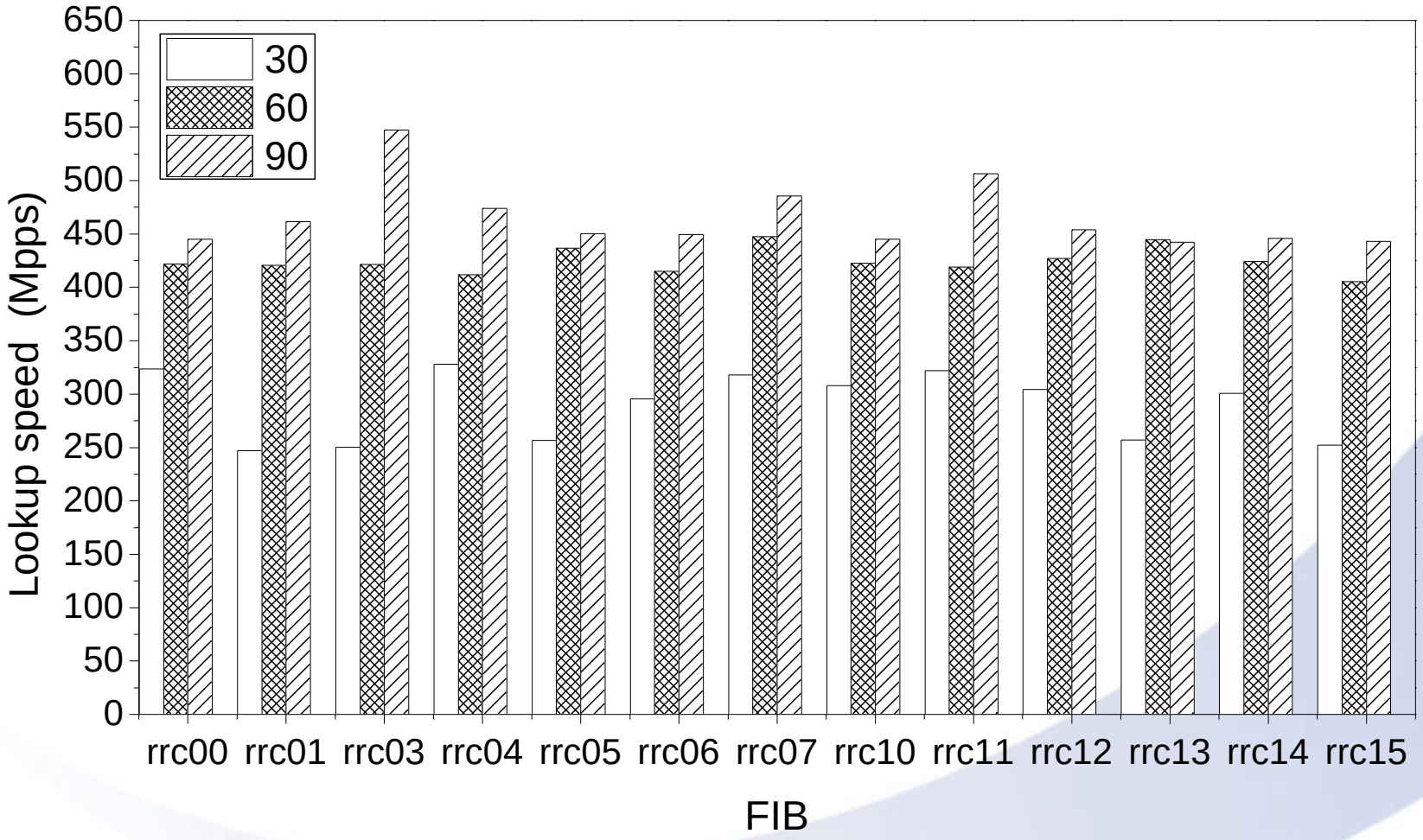
Intel CPU: 12 FIBs using prefix-based and random traces



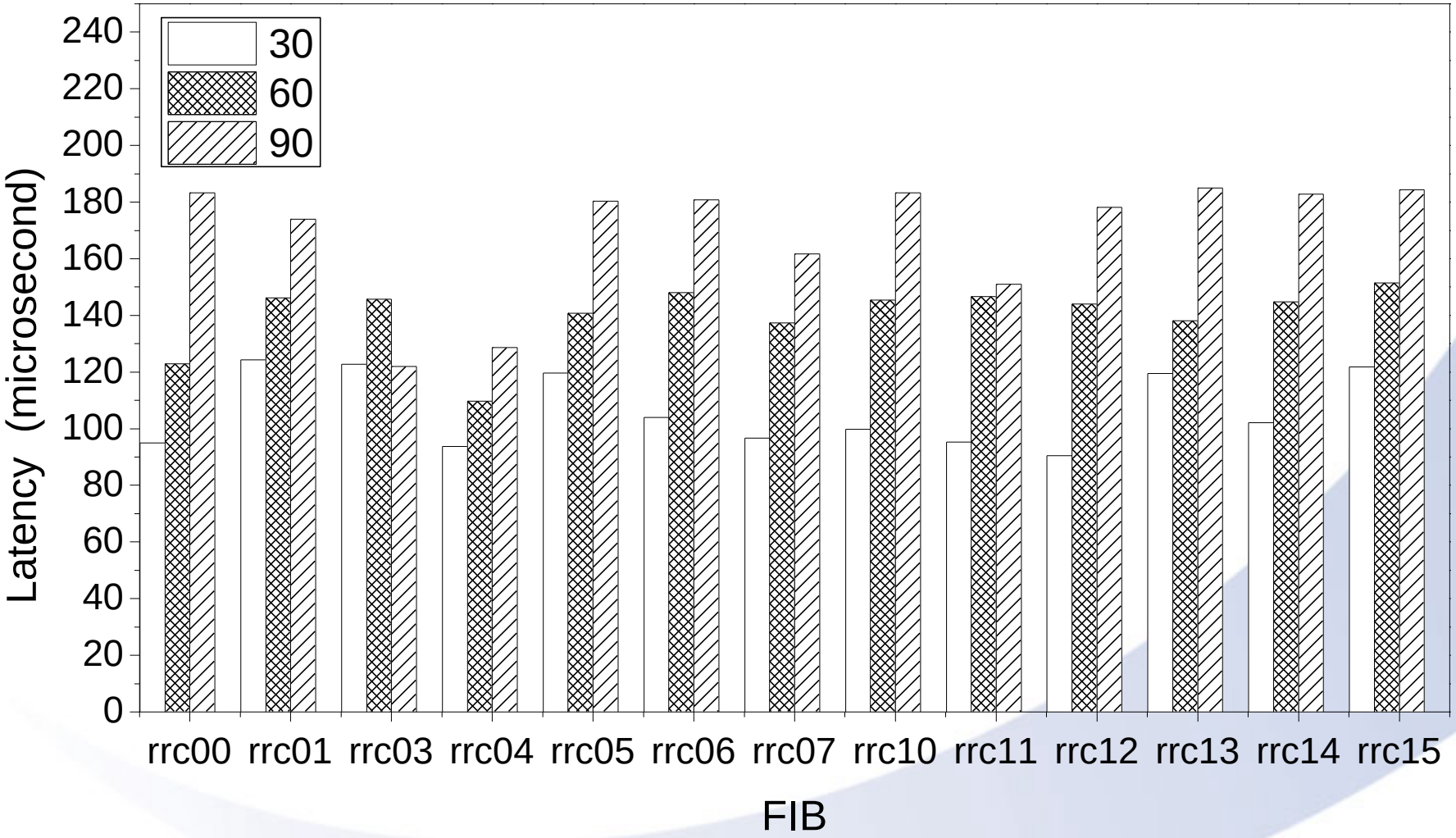
Intel CPU: Update



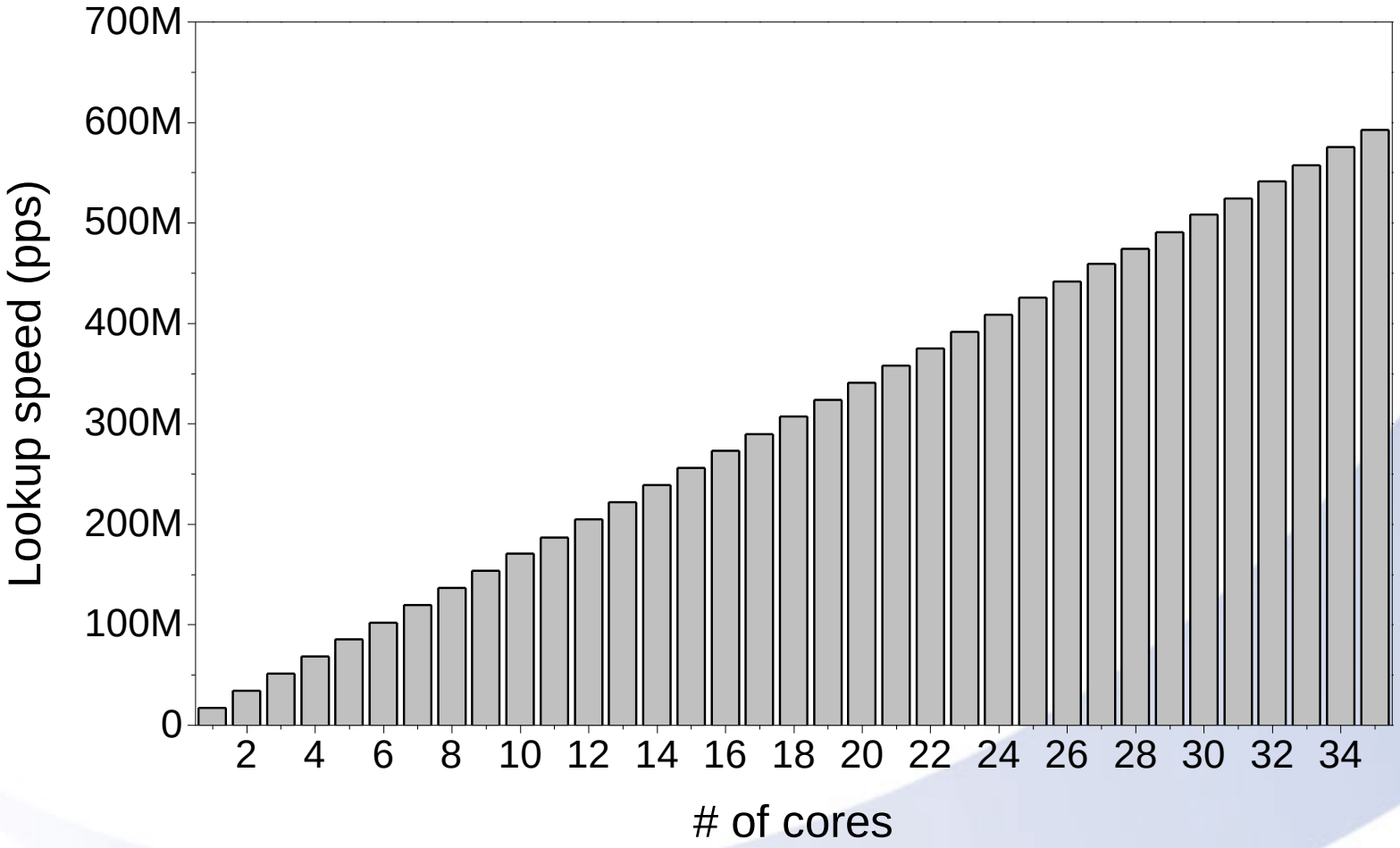
GPU: Lookup speed VS. batch size



GPU: Lookup latency VS. batch size



Tilera GX-36: Lookup VS. # of cores



Conclusion

- **Two-dimensional Splitting Framework: SAIL**
- **Three optimization algorithms**
  - **SAIL_U, SAIL_L, SAIL_M**
  - Up to 2.13MB on-chip memory usage
  - 2 off-chip memory accesses
- **Suitable for different platforms**
  - **FPGA, CPU, GPU, Many-core**
  - Up to 673.22~708.71 Mpps
- **Future work: SAIL to IPv6 lookup**

Source codes of SAIL, LC-trie, Tree Bitmap, and Lulea

http://fi.ict.ac.cn/firg.php?n=PublicationsAmpTalks.OpenSource

Thanks

http://fi.ict.ac.cn

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES