High-Performance Packet Classification on GPU

Shijie Zhou, Shreyas G. Singapura, and
Viktor K. Prasanna

Ming Hsieh Department of Electrical Engineering
University of Southern California

Outline

- Introduction
- Background
- Contributions
- Algorithm
- Evaluation
- Conclusion and Future Work

Outline

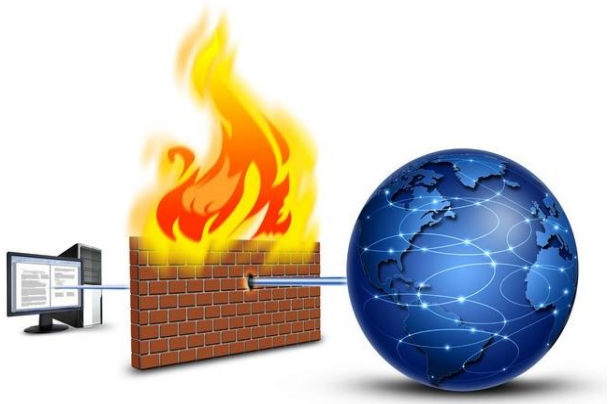
- Introduction
- Background
- Contributions
- Algorithm
- Evaluation
- Conclusion and Future Work

Introduction (1)

- **Internet:** Global system of interconnected computer networks
- Exponentially increasing network traffic
- Future Internet
  - More network traffic
  - Large amounts of data
  - Changing more frequently

Introduction (2)

- **Multi-field Packet Classification Applications**
  - Routing
  - Access control in firewalls
  - Provision of differentiated qualities of service
  - OpenFlow flow table lookup



Outline

- Introduction
- Background
- Contributions
- Algorithm
- Evaluation
- Conclusion and Future Work

Multi-field Packet Classification (1)

- 5-Fields
  - Source IP address
  - Destination IP address
  - Source port number
  - Destination port number
  - Protocol

	Src IP	Des IP	Src Port	Des Port	Protocol
Bits	32	32	16	16	8

Multi-field Packet Classification (2)

- **Rule-set**
  - A certain number of rules
  - Matching criteria for each field
  - Wildcard bit in the rule:*1*, *0* or * (do not care)
  - Priority

Multiple matches →choose the highest priority rule

→ take the action

Multi-field Packet Classification (3)

ID	Src IP	Des IP	Src Port	Des Port	Protocol	Priority	ACTION
1	175.77.88.155/31	119.106.158.230/32	0-65535	6888-6888	0x06	1	Act 0
2	175.77.88.6/20	36.174.239.222/32	0-65535	1604-1704	0x06	2	Act 1
3	12.2.0.0/16	192.1.1.0/24	20-30	1024-1024	0x11	3	Act 2

Related Work

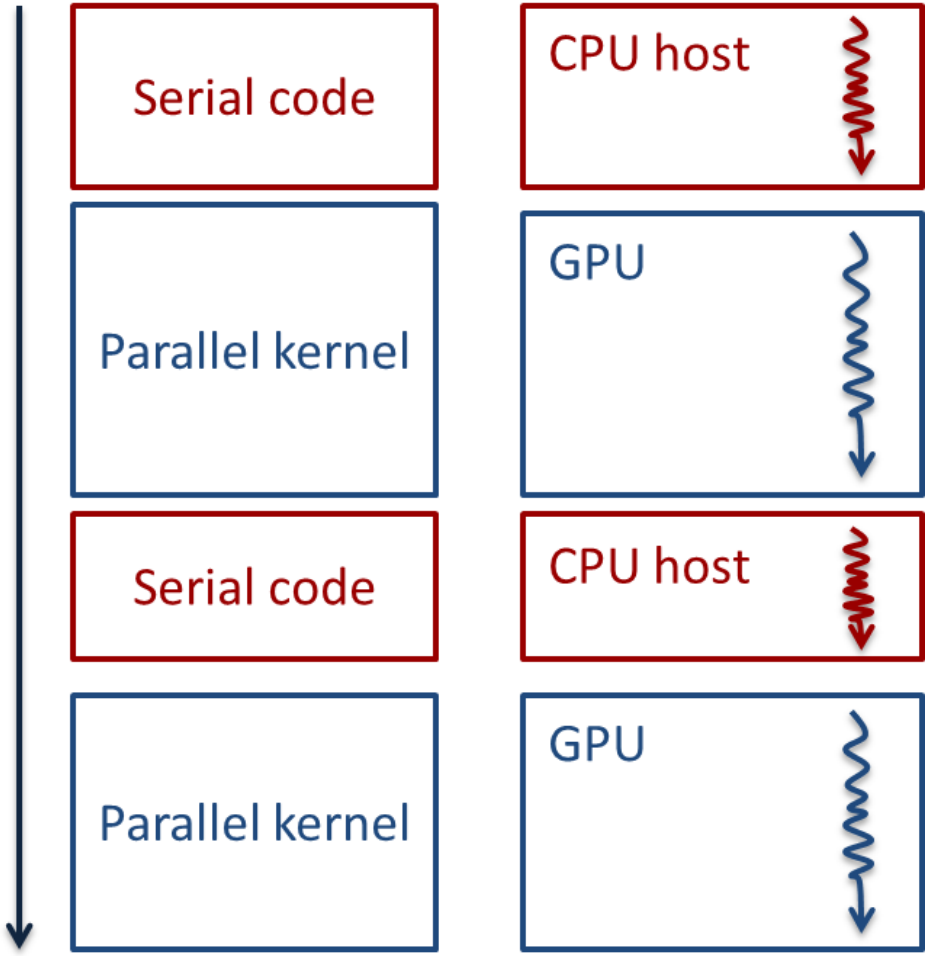
- Packet classification on GPU
  - Relatively less explored

- Previous GPU implementations
  - Unique Rules: small[1]
  - Throughput or Latency not discussed[2]
  - ~11 and 5 MPPS for 500 and 2000 rules[3]

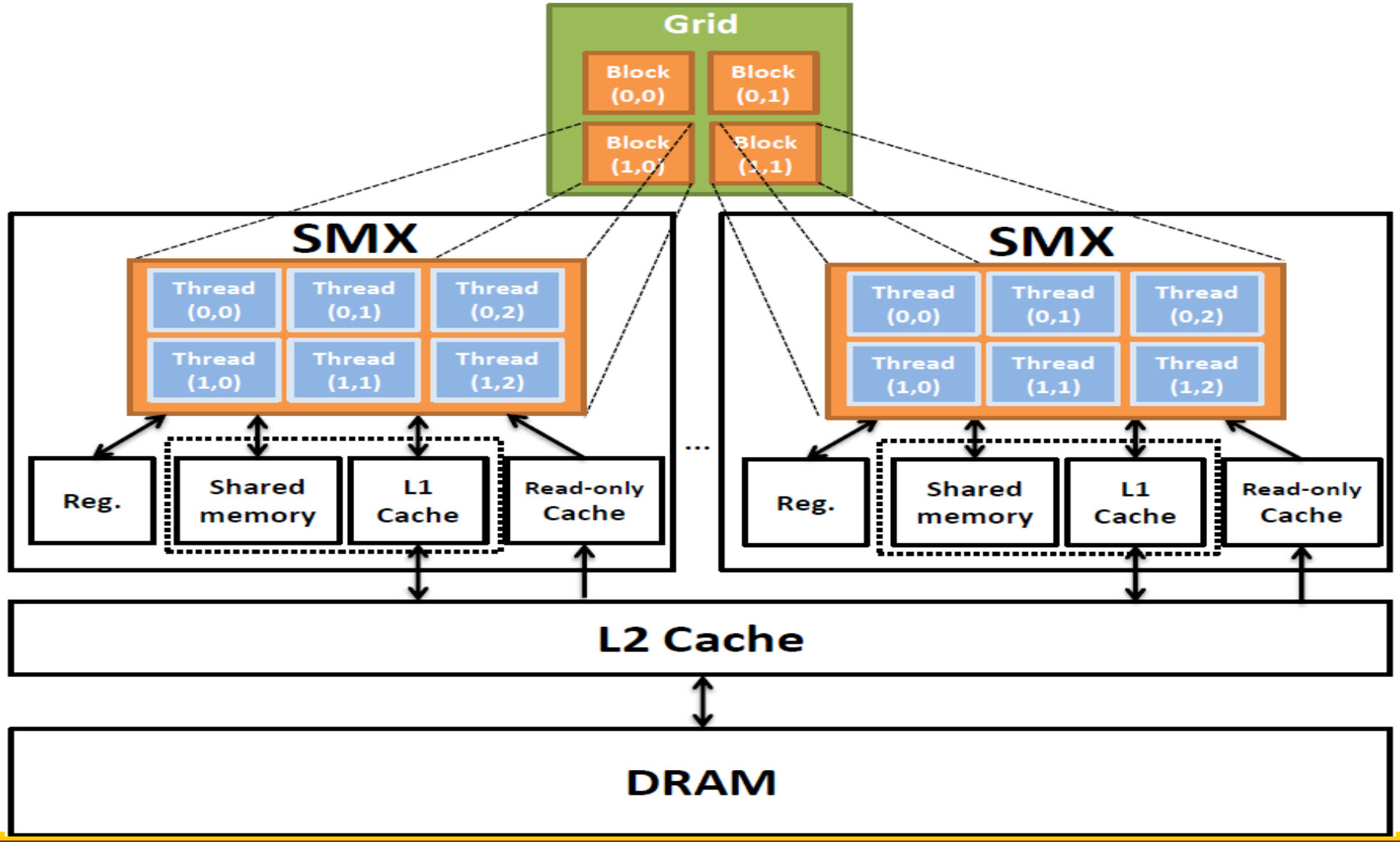
[1] A. Nottingham and B. Irwin, “Parallel packet classification using GPU co-processors,” in SAICSIT Conf.ACM., pp. 231-241, 2010.
[2] C. Hung, Y. Lin, K. Li, H. Wang and S. Guo, “Efficient GPGPUbased parallel packet classification,” in Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1367-1374, 2011.
[3] K. Kang and Y. S. Deng, “Scalable packet classification via GPU metaprograming,” in Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1-4, 2011.

CUDA Programming Model

- CUDA program
  - Host + Kernel
- Host function runs on CPU
- Kernel function runs on GPU



GPU Architecture



Outline

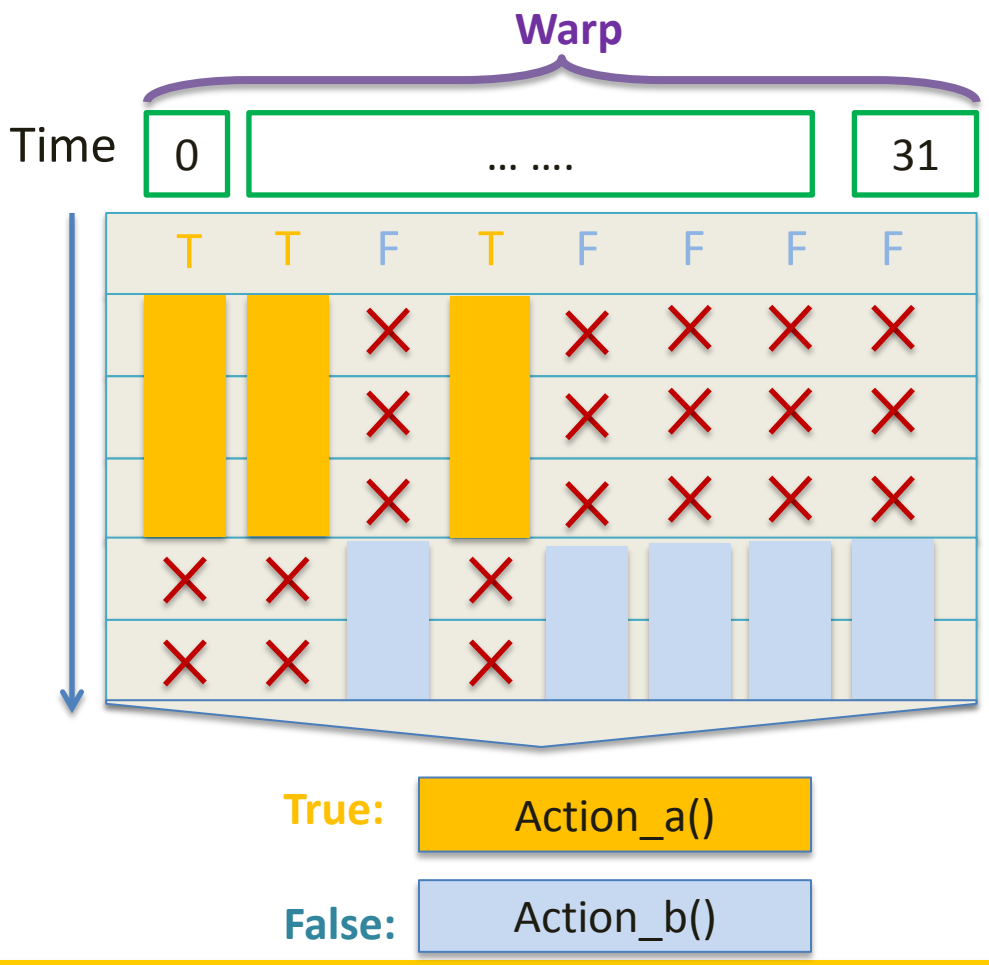
- Introduction
- Background
- Contributions
- Algorithm
- Evaluation
- Conclusion and Future Work

Contributions

- Range-tree search and bit vector (BV) based packet classifier on GPU
- Efficient range-tree search on GPU
- Optimize data layout to minimize shared memory bank conflict
- Throughput of 85 MPPS for 512-rule rule-set

Challenges

- Divergence Overhead
- Limited on-chip memory: data layout
- Classic Tree-Search: pointers to connect nodes

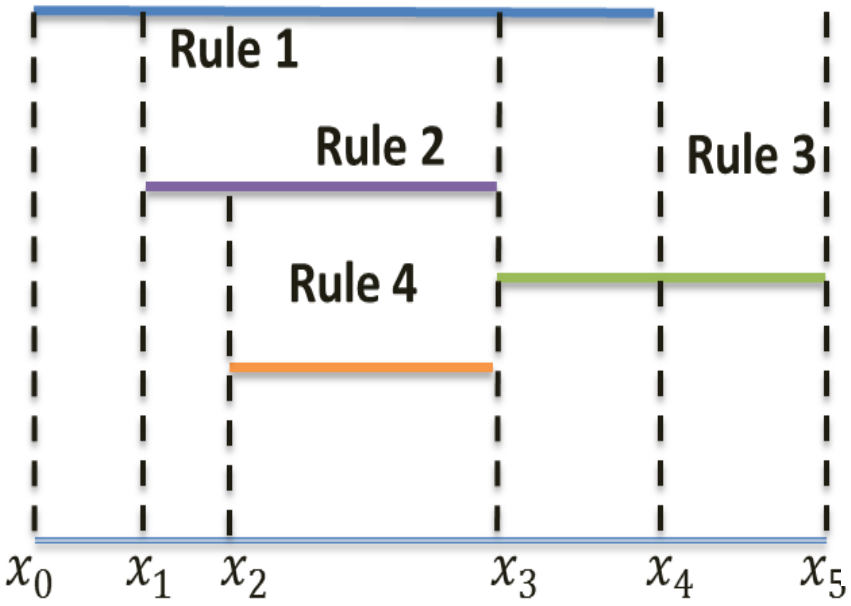


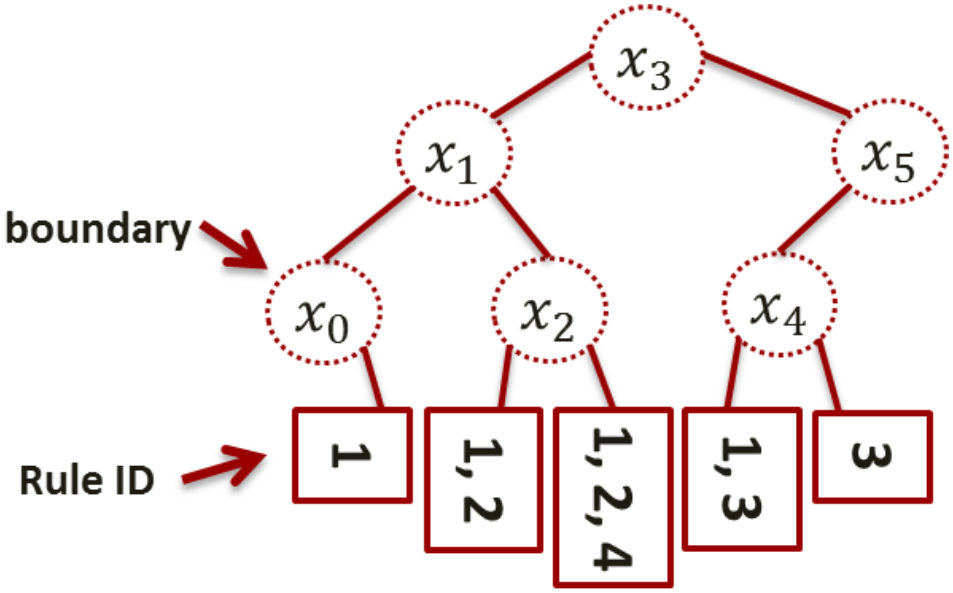
Outline

- Introduction
- Background
- Contributions
- Algorithm
- Evaluation
- Conclusion and Future Work

Algorithm (1)

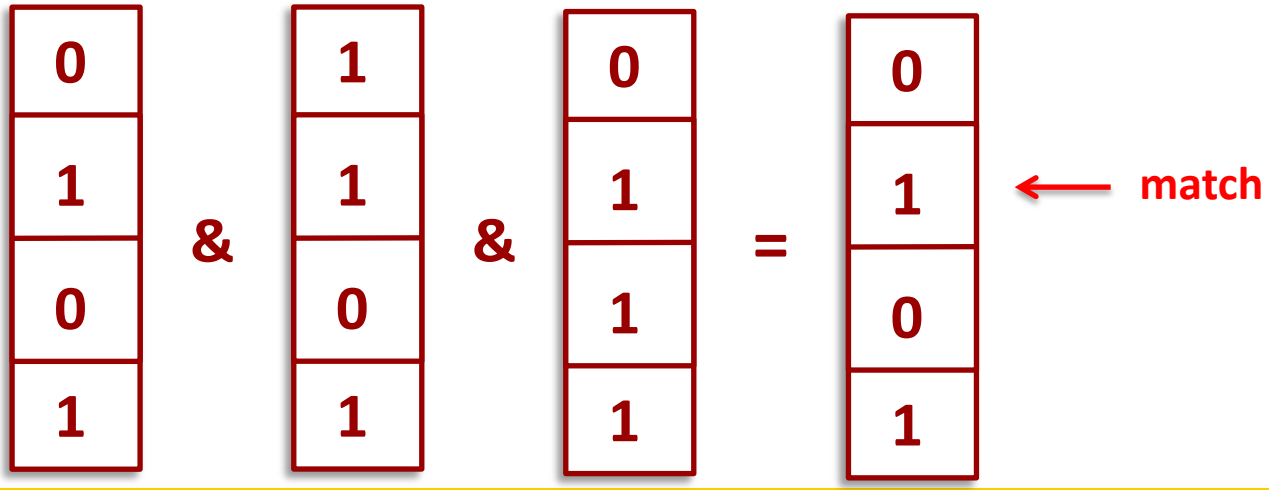
- Decomposition-based approach
  - Range-tree Search





Algorithm (2)

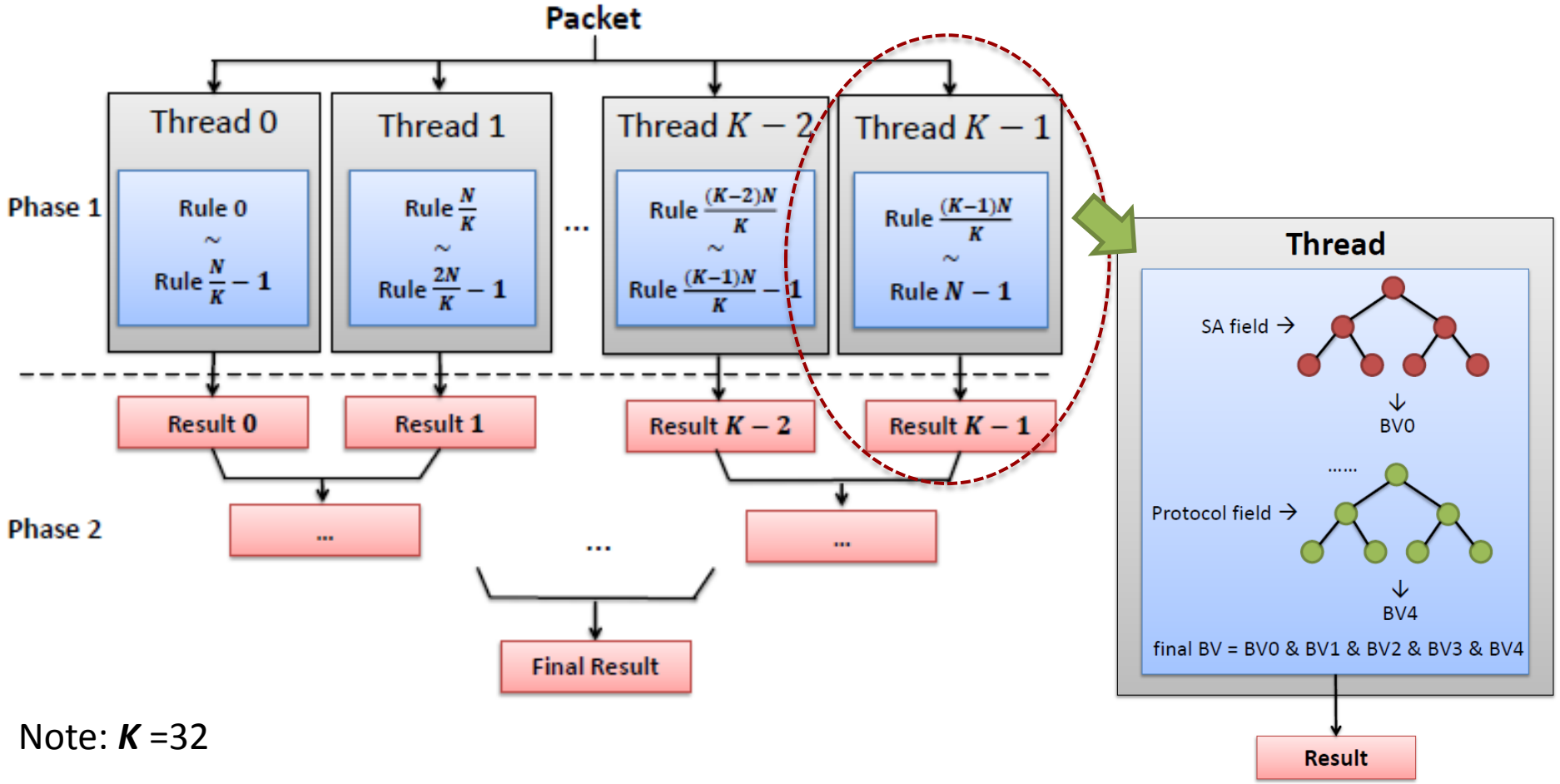
- **Bit Vector Representation**
  - *i*th bit → *i*th original rule
    - 1 → match
    - 0 → not match
  - Merge by *Bit AND* operation



Algorithm (3)

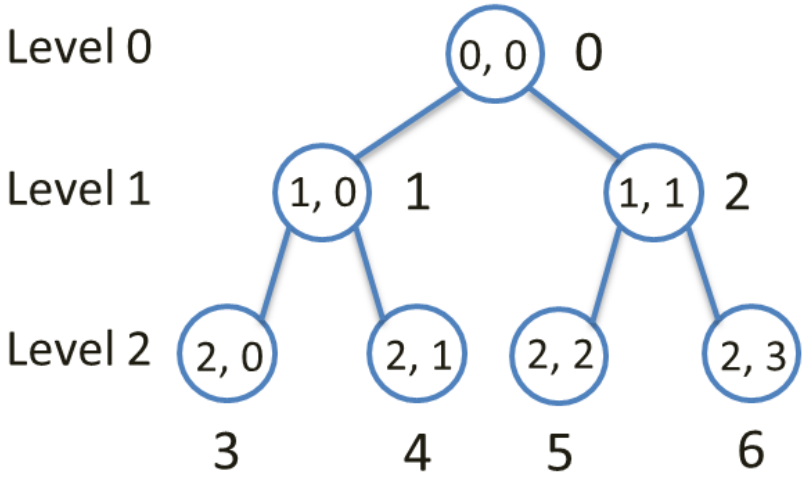
- 32 threads (a warp) per packet
- Pre-processing (In CPU):
  - Partition rule-set into 32 subsets
  - Construct range-trees & BVs for each subset
- Classification (In GPU):
  - Phase 1: obtain an intermediate result (using the range-trees and BVs)
  - Phase 2: intermediate results → final result

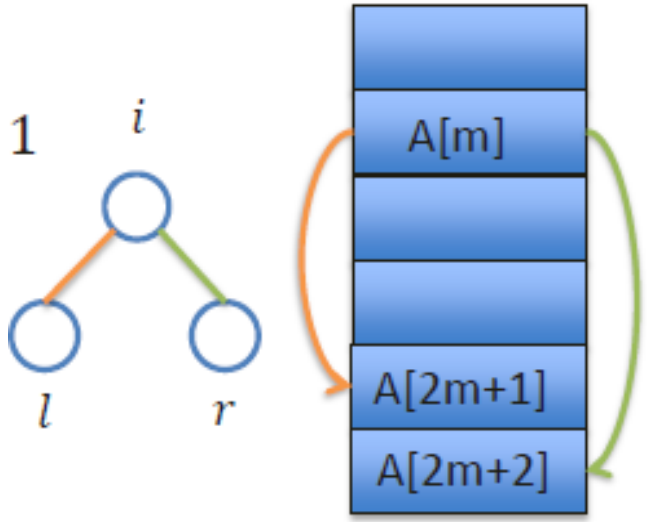
Architecture

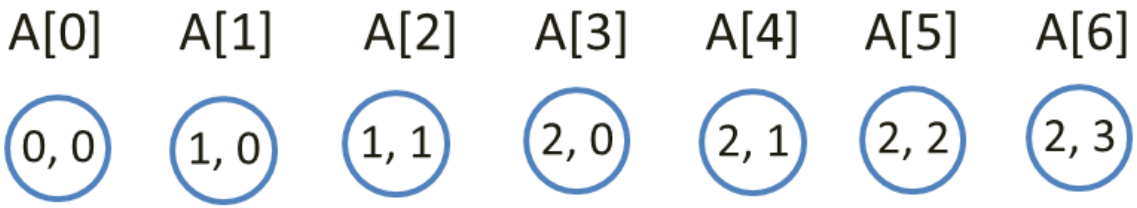


Note: ***K*** =32

Optimizations (1)

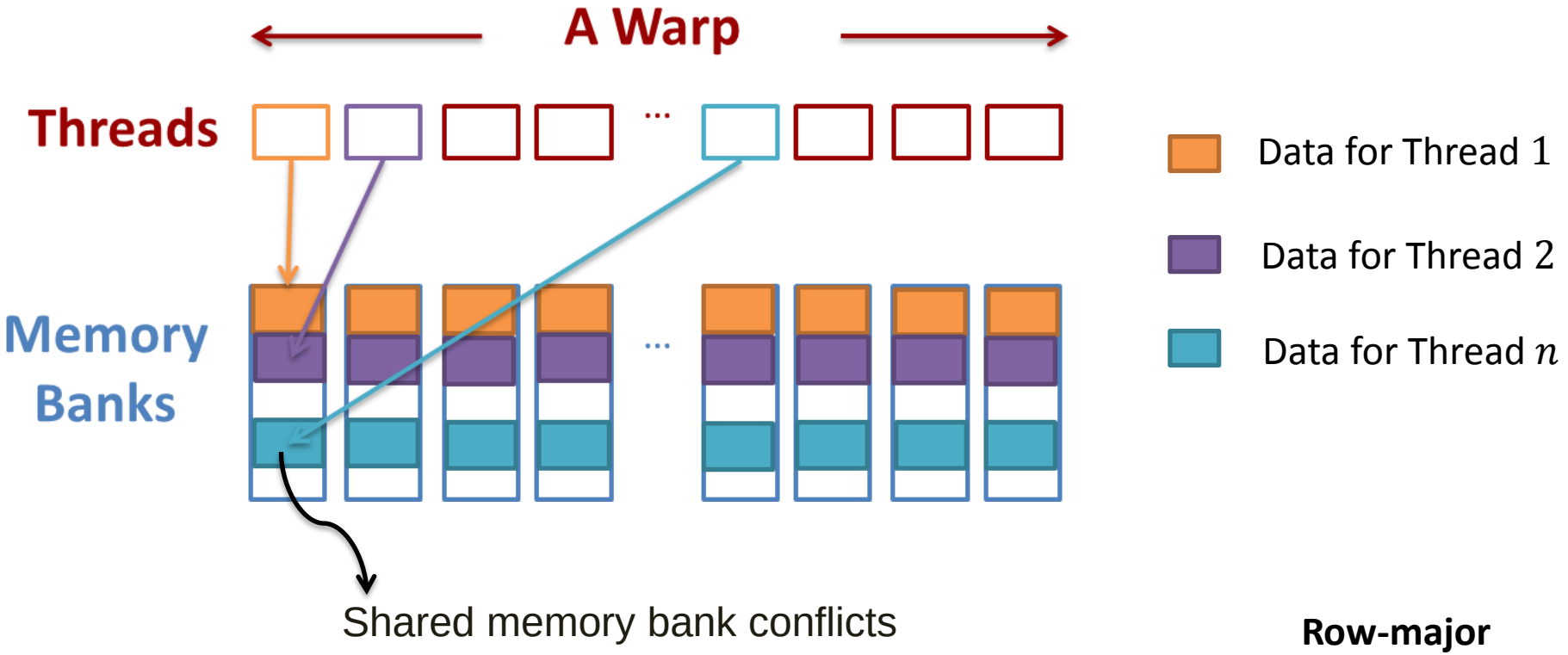






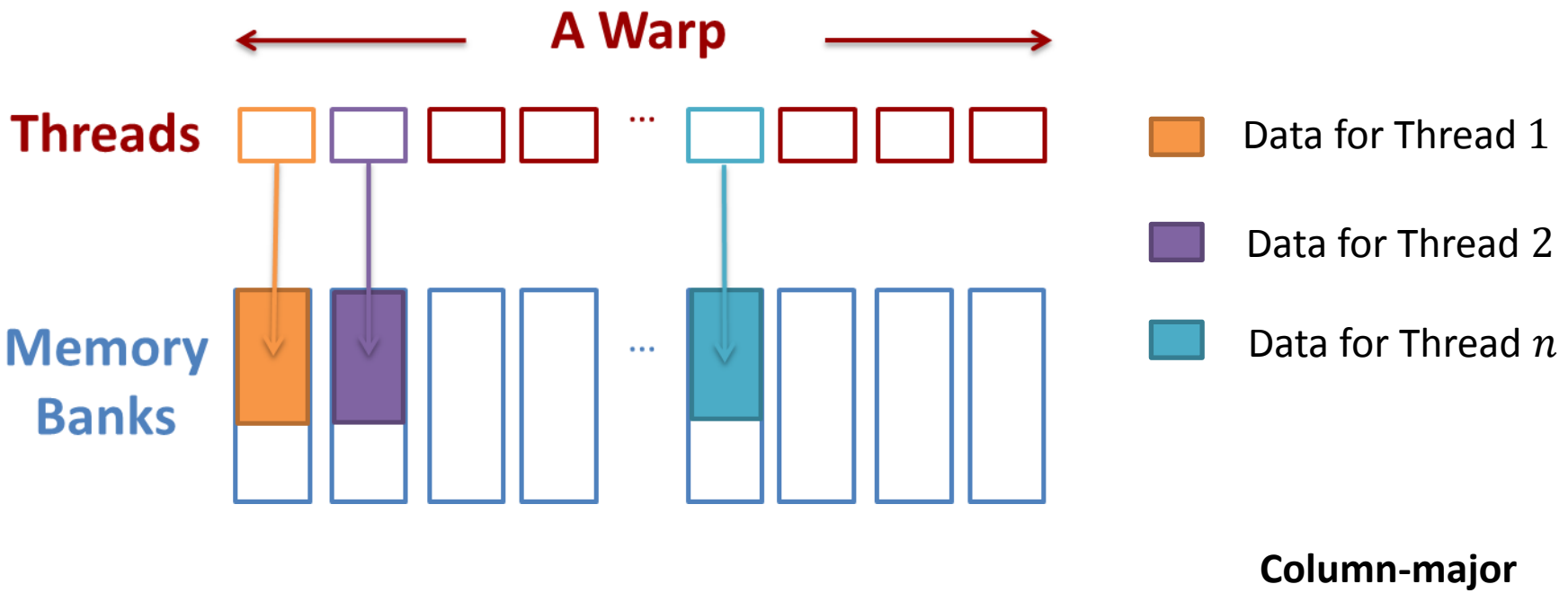
Optimizations (2)

- Store range-trees in shared memory



Optimizations (3)

- Minimize shared memory bank conflicts



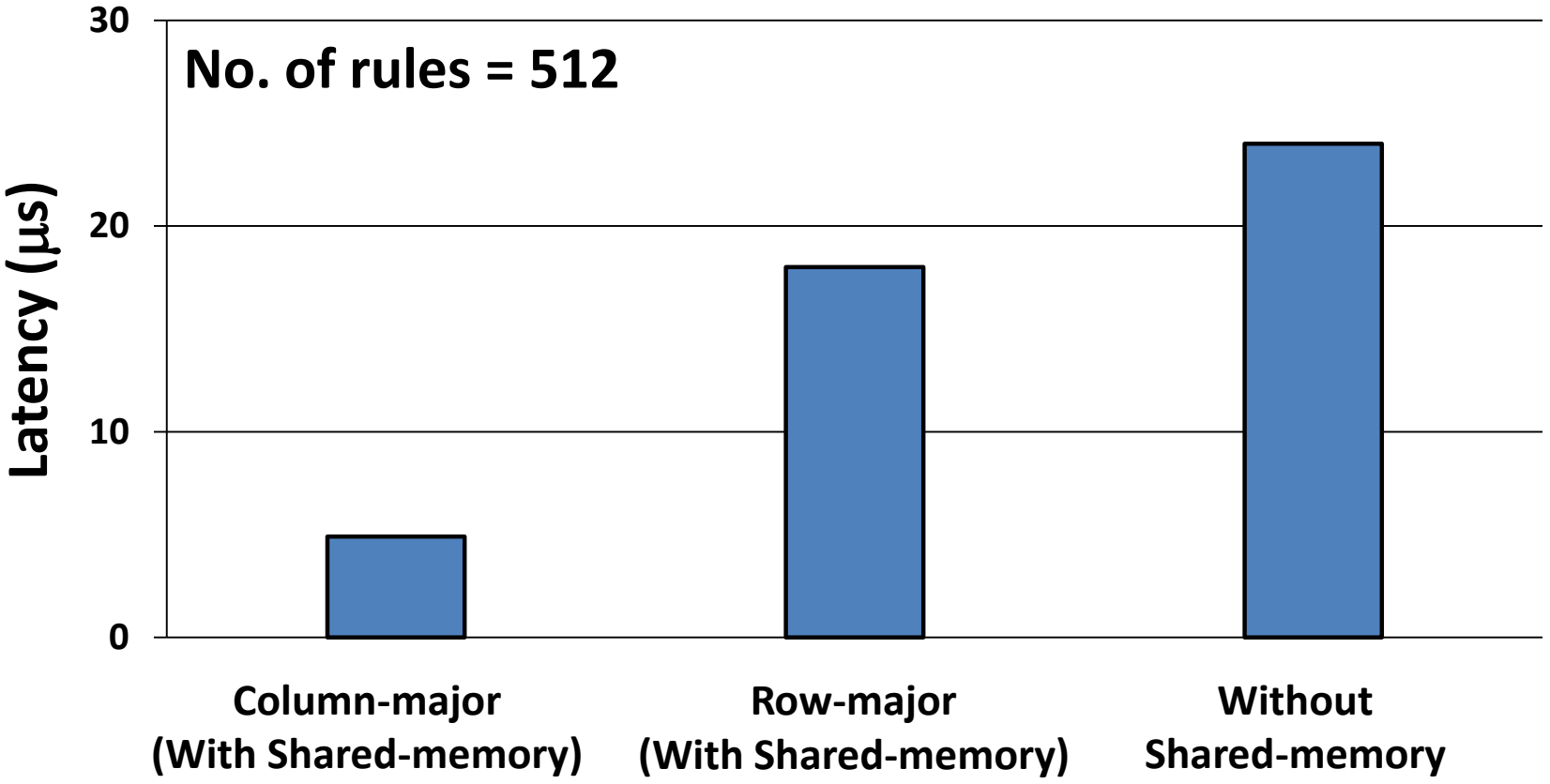
Outline

- Introduction
- Background
- Contributions
- Algorithm
- Evaluation
- Conclusion and Future Work

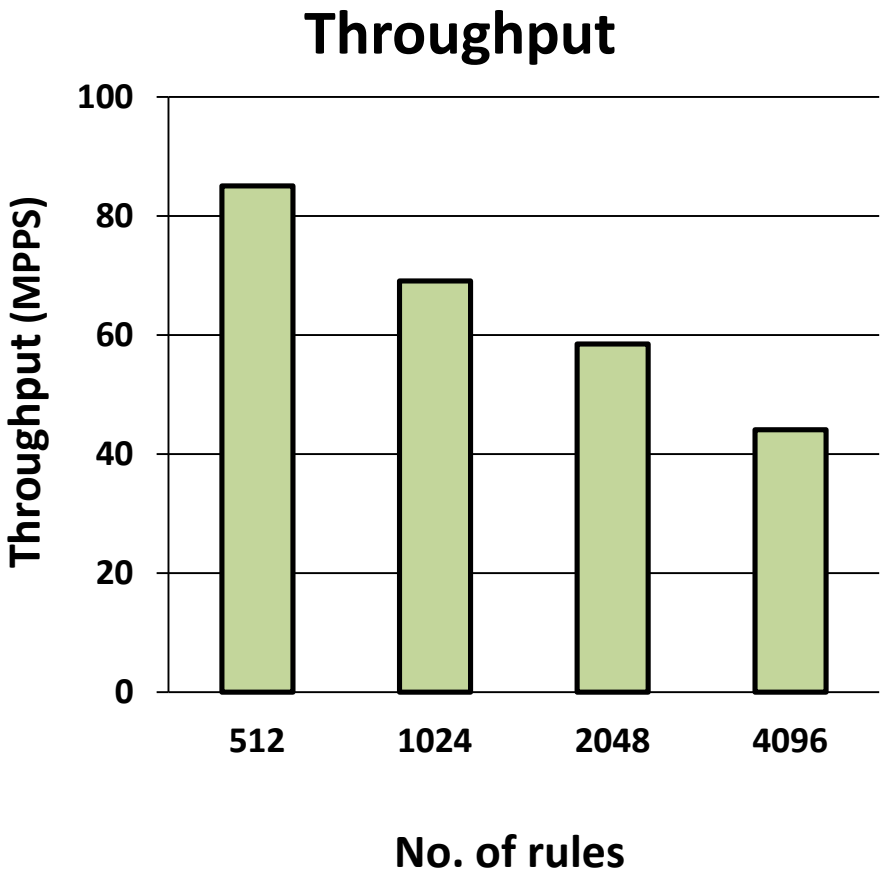
Platform

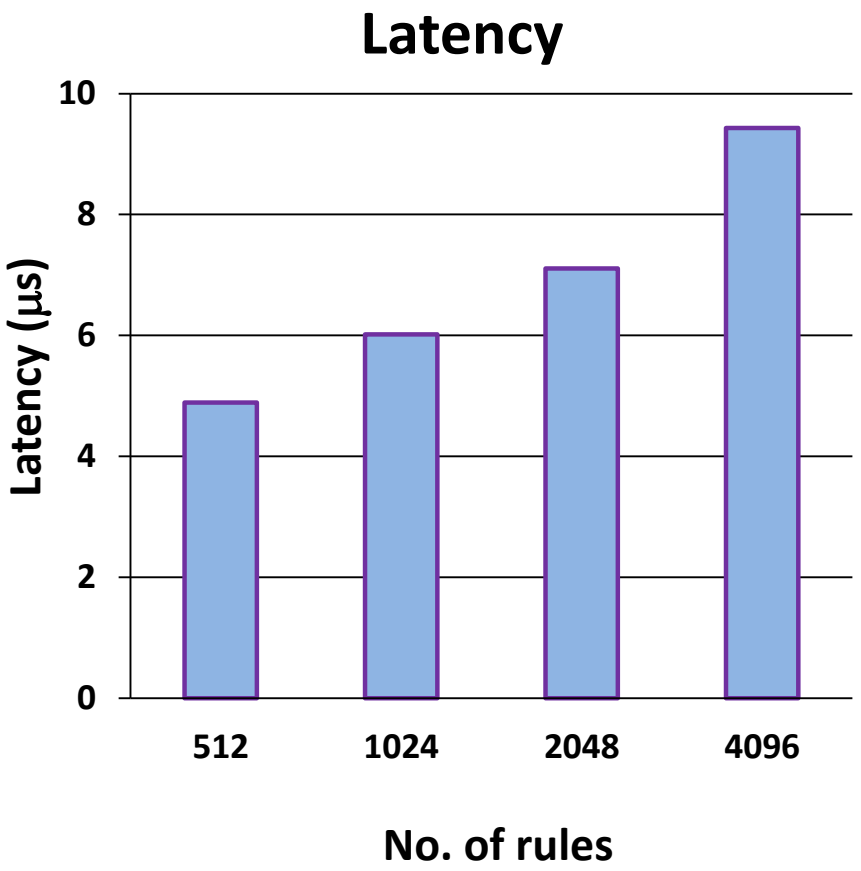
- CPU (Intel E5-2665)
  - Cores: 16
  - Frequency: 2.4 GHz
- GPU (NVIDIA K20 Kepler)
  - Streaming Multi-Processor (SMX): 13
  - CUDA cores: 2496
  - Frequency: 705.5 MHz

Performance (1)



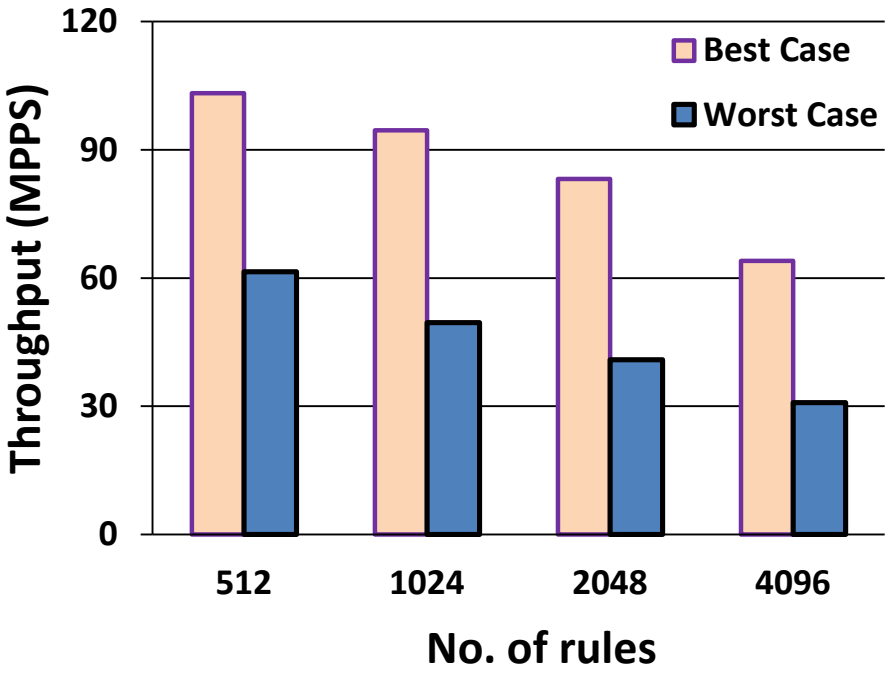
Performance (2)

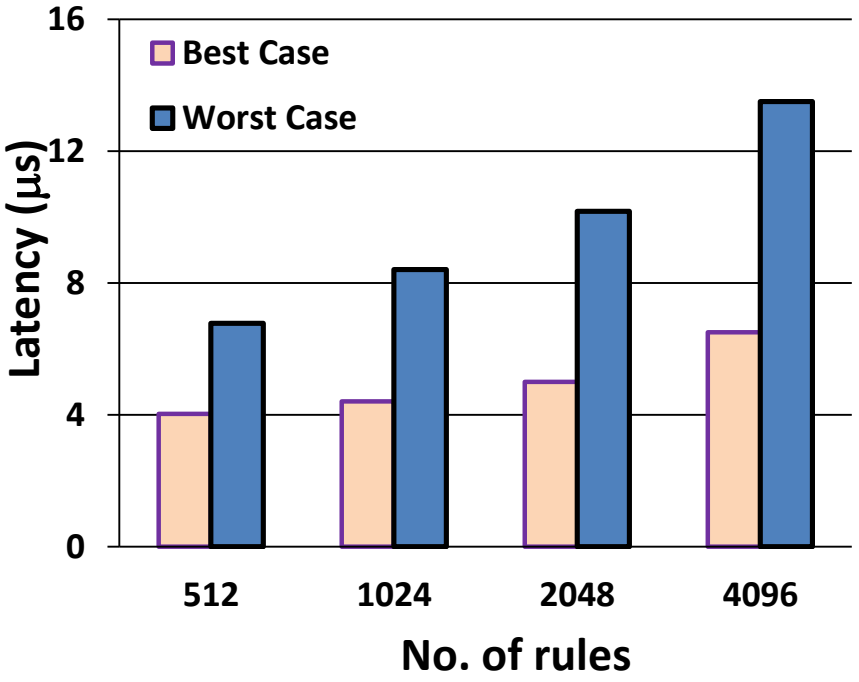




Performance (3)

- Best Case: smallest possible range-trees
- Worst Case: largest possible range-trees





Outline

- Introduction
- Background
- Summary of Contributions
- Algorithm
- Evaluation
- Conclusion and Future Work

Conclusions

- Range-trees + BV packet packet classifier on GPU: 85 MPPS for 512-rule rule-set

- Performance: throughput and latency
  - number of rules (512 ~ 4096)
  - data layout in shared memory

- Compared to state-of-the-art multi-core implementation: 2x improvement in throughput

Future Work

- Hash-based packet classification algorithms

- Other networking applications using GPUs
  - Traffic classification
  - OpenFlow flow table lookup

Thank you!

- Group Webpage:
  - http://ganges.usc.edu/wiki/Parallel_Computing
- Email IDs:
  - shijiezh@usc.edu, singapur@usc.edu,
  - prasanna@usc.edu