High-Performance Traffic Classification on GPU

Shijie Zhou, Nittoor Rao Prashant and Viktor K. Prasanna
Ming Hsieh Department of Electrical Engineering

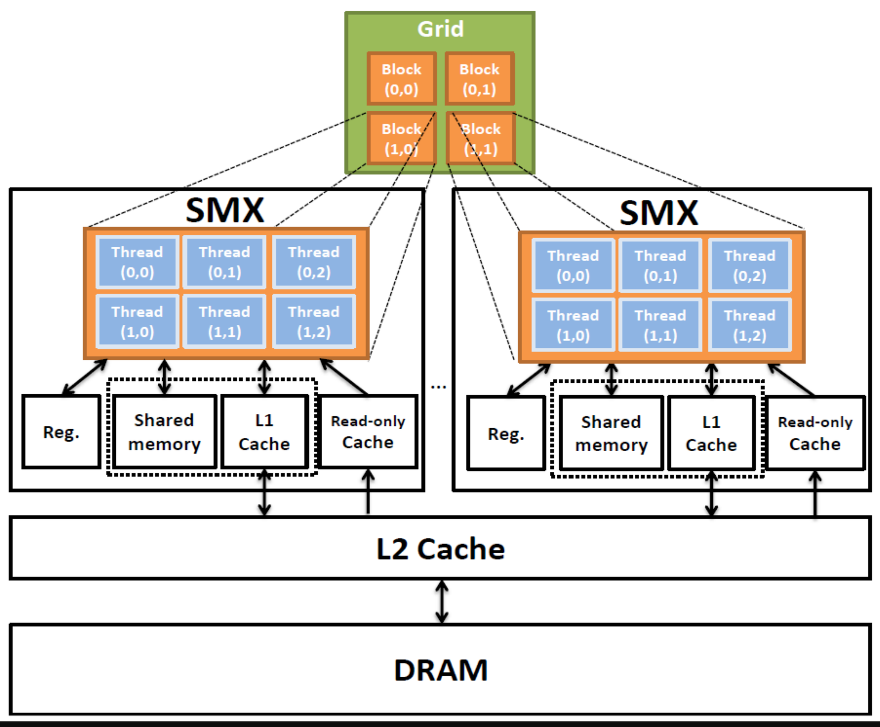
Introduction & Background

Traffic classification

- Network management
  - ❑ Flow prioritization
  - ❑ Traffic shaping
  - ❑ Traffic policing
- Approaches
  - ❑ Transport layer port number-based
    - Port numbers can be dynamically allocated
  - ❑ Deep packet inspection-based
    - Cannot handle encrypted traffic
  - ❑ Heuristic-based
    - Low accuracy
  - ❑ Statistics-based
    - Machine learning
    - High accuracy

CUDA programming model

- Host function + Kernel function
- SIMT execution model
- Various memory types on GPU



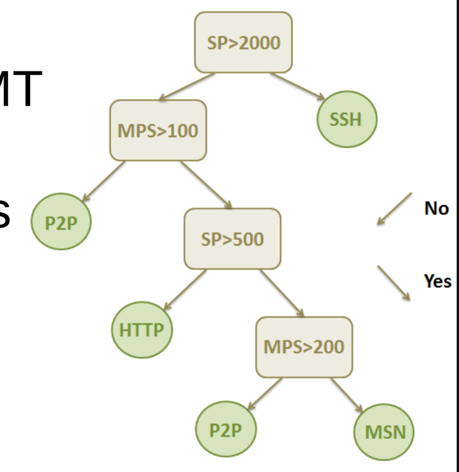
C4.5 Decision tree

- Statistics-based approach
- Built Based on information entropy
- Examine features of packet header

TABLE I: Candidate Features

Source port number	Destination port number
Average packet size	Variance of packet size
Maximum packet size	Minimum packet size
Maximum inter-arrival time	Minimum inter-arrival time
# of Bytes in forward direction	# of Bytes in back direction

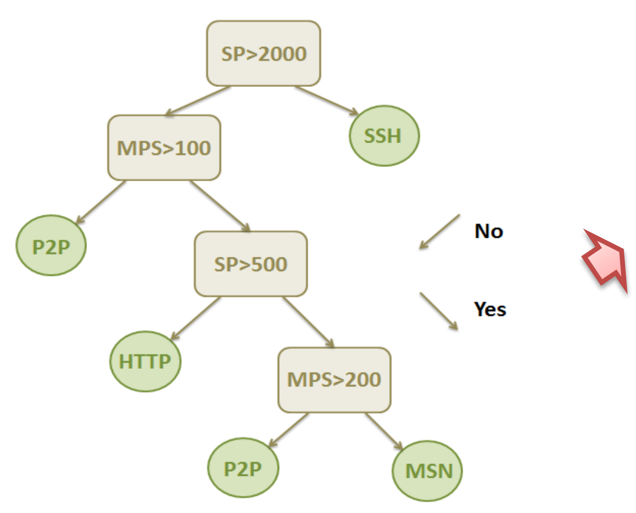
- Highly unbalanced
  - ❑ Unappropriate for SIMT execution model
  - ❑ Performance depends on traffic pattern



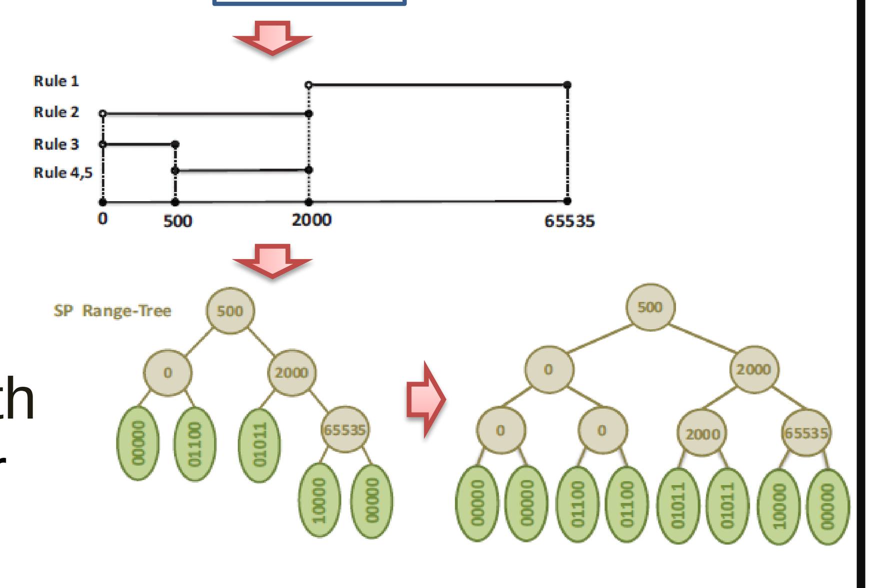
Algorithms

Convert C4.5 decision tree to rule set table

- Each leaf node → one rule



Rule ID	SP	MPS	App. Class
1	(2000, 65535]	(0, 65535]	SSH
2	(0, 2000]	(0, 100]	P2P
3	(0, 500]	(100, 65535]	HTTP
4	(500, 2000]	(0, 200]	P2P
5	(500, 2000]	(200, 65535]	MSN



Build a perfect range-tree for each column (feature)

- All leaves have the same depth
- Each leaf contains a bit-vector
  - ❑ One bit for one rule
  - ❑ '1' in the i^{th} position: the i^{th} rule contains this range

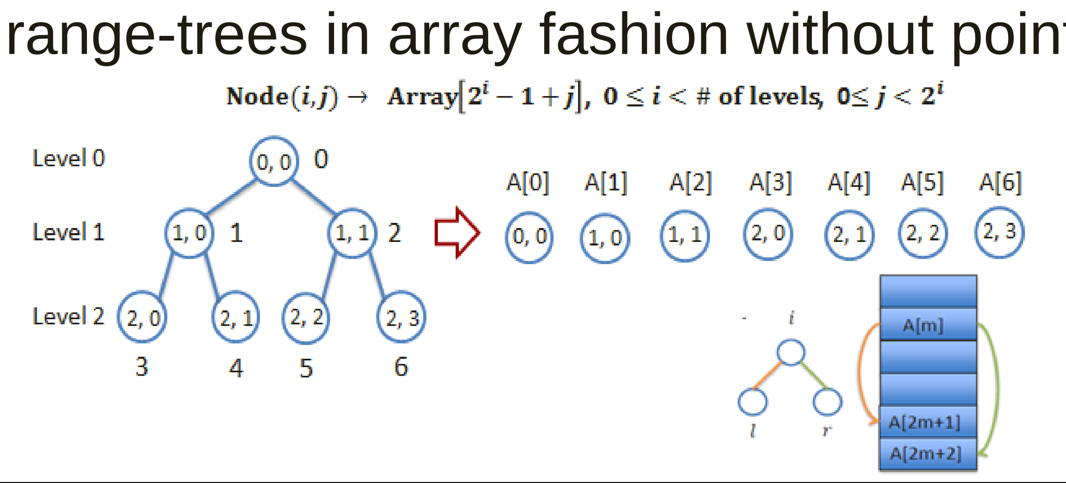
Decomposition-based classification

- Search
  - ❑ Range-tree search for each field
  - ❑ Each field obtains a bit-vector
- Merge
  - ❑ Bitwise-AND all bit-vectors
  - ❑ Only one '1' in the result bit vector

Optimizations

- Use shared-memory to store range-trees
- Store range-trees in array fashion without pointers

$$\text{Node}(i,j) \rightarrow \text{Array}[2^i - 1 + j],\ 0 \le i < \#\text{ of levels},\ 0 \le j < 2^i$$



Performance Evaluation

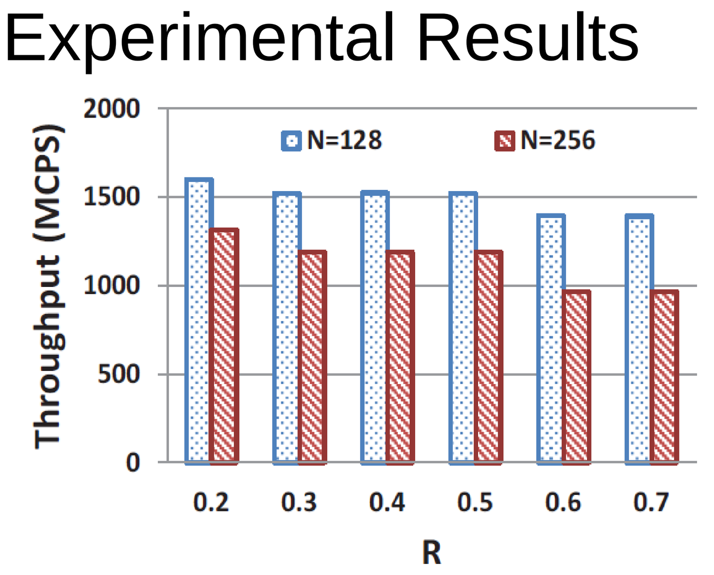
Platform

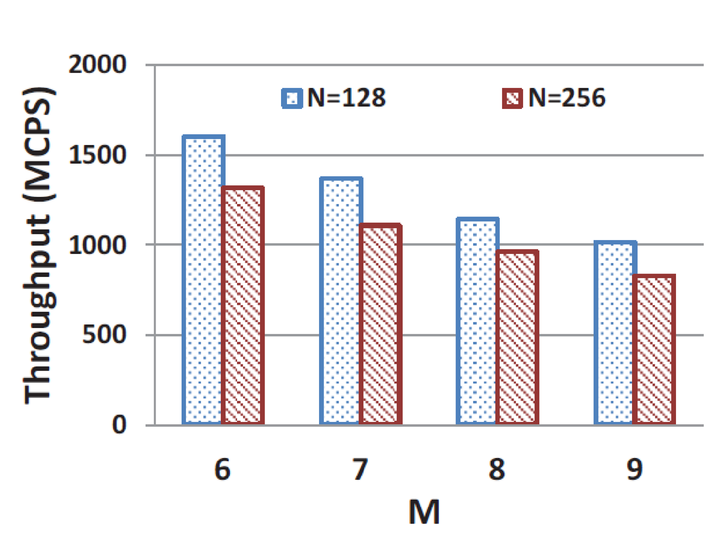
- Dual 8-core Intel E5-2665; 2.4 GHz
- NVIDIA K20 Kepler
  - ❑ 2496 CUDA cores
  - ❑ 705.5 MHz

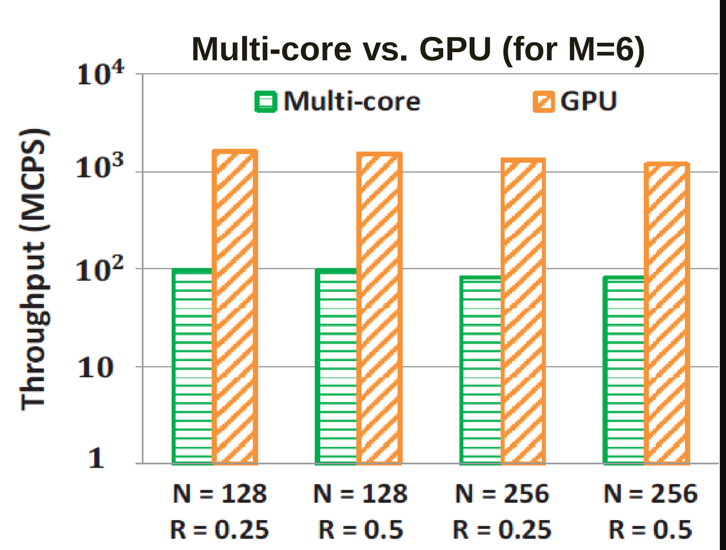
Notations

- N: the number of rules
- M: the number of features
- R: the percentage of the unique values in each feature

Experimental Results







Conclusion

- Convert unbalanced C4.5 decision tree into perfect binary range-trees
  - ❑ Decomposition-based classification
  - ❑ 1.88x higher throughput compared with C4.5 decision tree
- Store range-trees using compact arrays without explicit pointers in shared memory
- Achieve a high throughput of 1.6 billion classifications per second (GCPS)
  - ❑ 128 leaf nodes; 6 features
  - ❑ 16x improvement compared with state-of-the-art multi-core implementation

Supported by the National Science Foundation under Grant No. 1018801. The target platform from USC's Center for High-Performance Computing is gratefully acknowledged.

***Email:** {shijiezh, nittoor, prasanna}@usc.edu*
***Web:** ganges.usc.edu*

Ming Hsieh Institute
Ming Hsieh Department of Electrical Engineering