



北京大學
PEKING UNIVERSITY

HybridCuts: A Scheme Combining Decomposition and Cutting for Packet Classification

Wenjun Li
Xianfeng Li

School of Electronic and Computer Engineering (ECE)
Peking University

IEEE Hot Interconnects 21
San Jose, CA, August 21-22, 2013



Outline

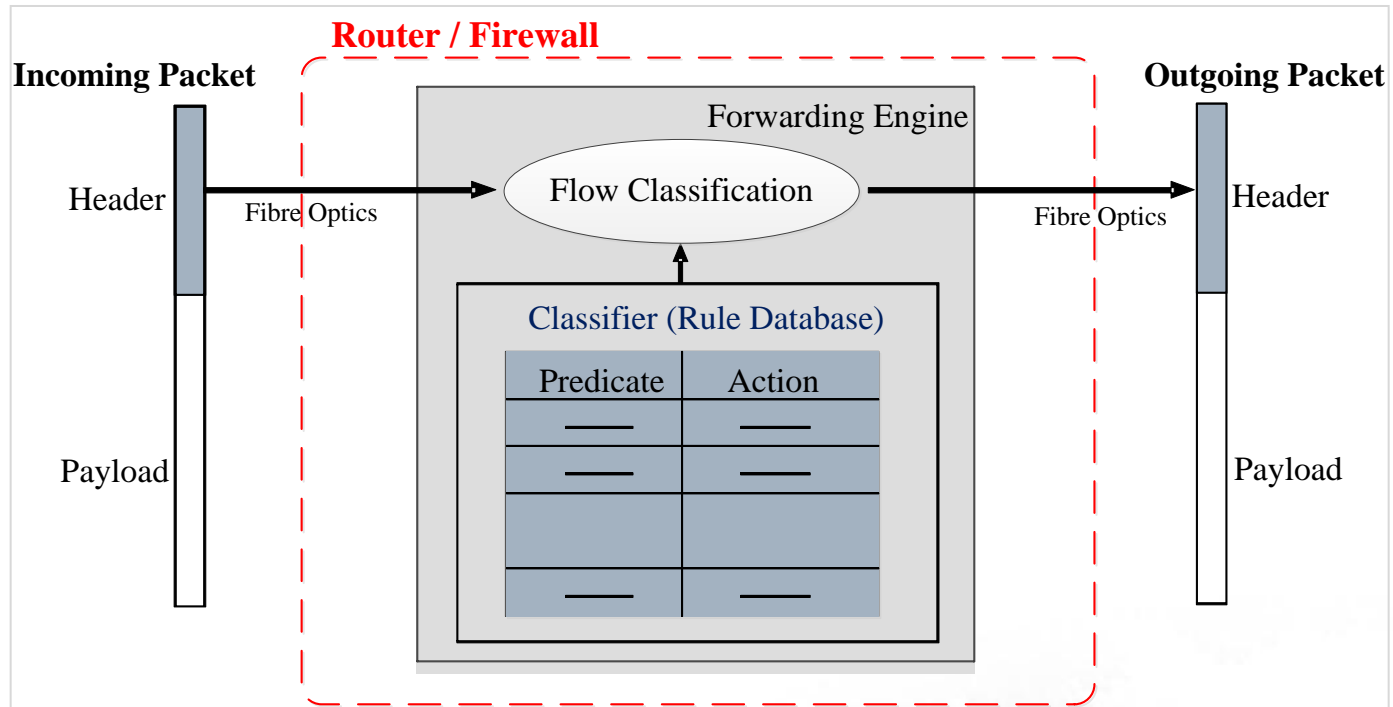
- **Background**
- **HybridCuts**
- **Evaluation**
- **Conclusion**



PART I: Background

Packet Classification

□ Key for policy enforcement in packet forwarding



#	SA	DA	SP	DP	Prot	Action
r_1	1.2.3.0/20	192.168.0.1	[1,65534]	[1,65534]	TCP	accept
r_2	1.2.3.11/24	1.2.3.11/24	80	[1,65534]	UDP	accept
r_3	*	*	*	*	*	discard



Why Yet Another Paper?

A well established problem

without

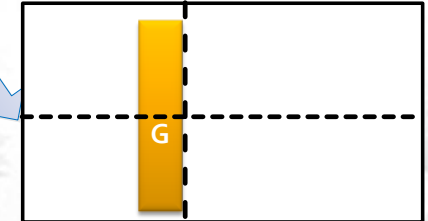
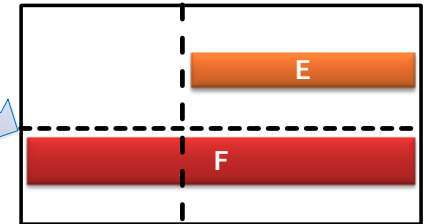
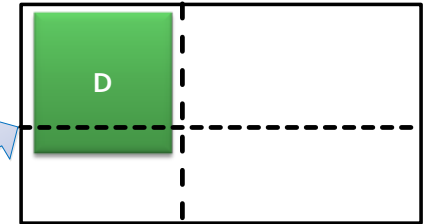
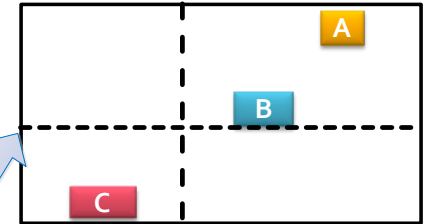
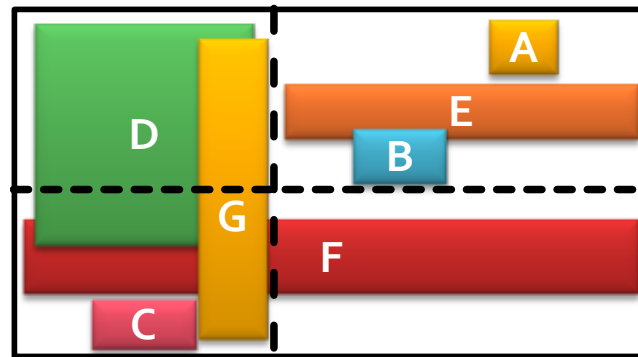
Well established solutions

- ❑ **Algorithmic:** Desired but speed/memory inefficient
- ❑ **Architectural:** Fast but expensive, power hungry, poor scalability and suffer from range expansion

Recent Efforts on Algo. Solutions

EffiCuts [SIGCOMM'10]

- ❑ Reduction by Separation
- ❑ Equal-dense cutting, etc



Pros

- ❑ Reduction on memory consumption

Cons

- ❑ Increase on #memory accesses



Better Solutions?

Rule separation: the right direction

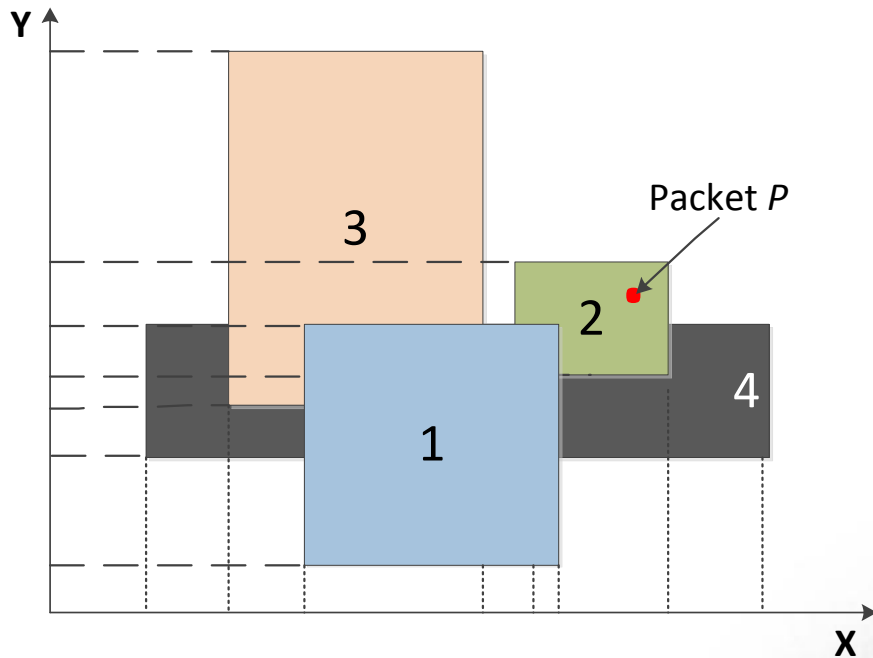
**Better separations
(with less rule groups)**



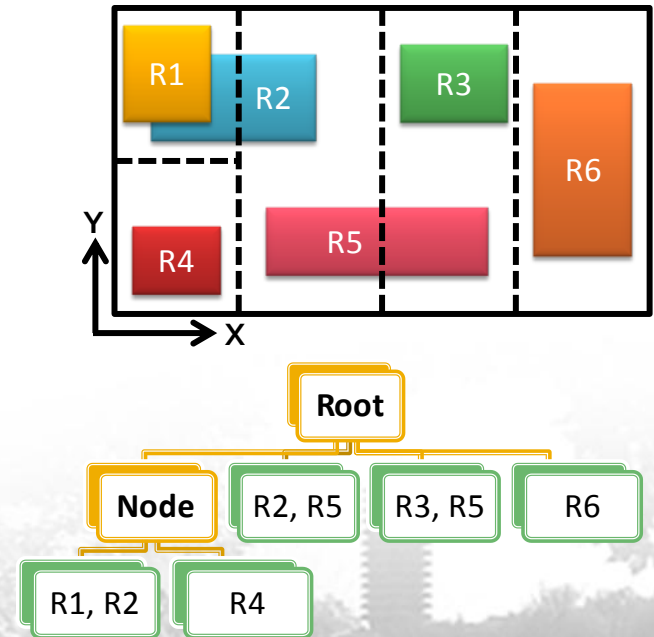
**Better cuttings
(by exploiting characteristics)**

A Little Review...

□ Decomposition



□ Cutting

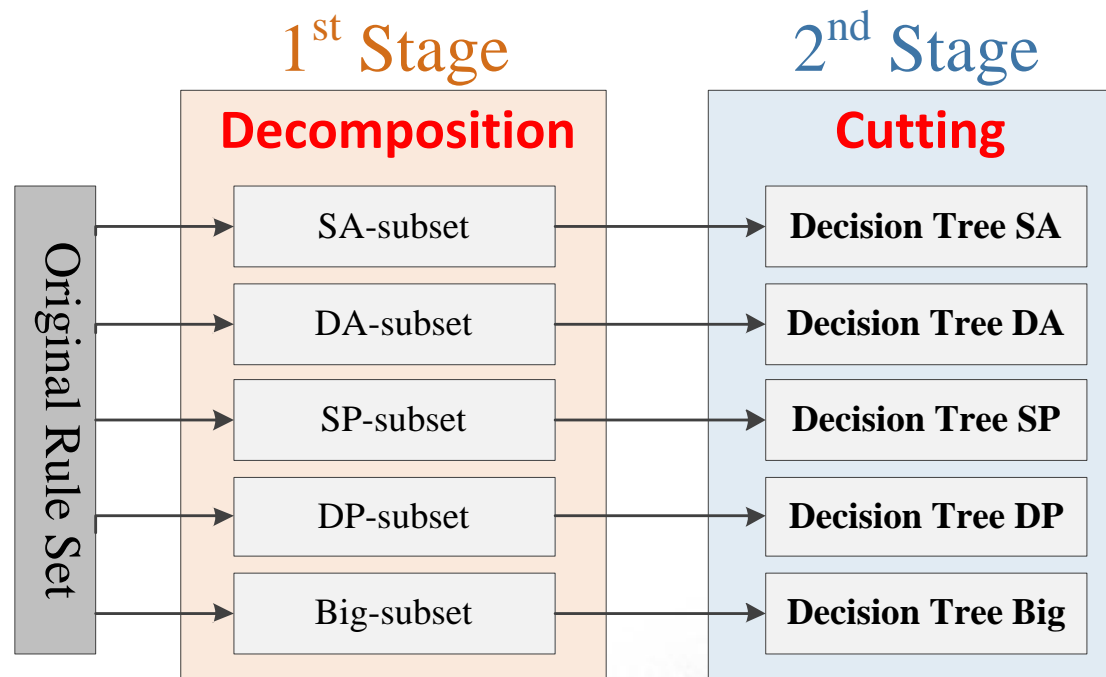




PART II: HybridCuts

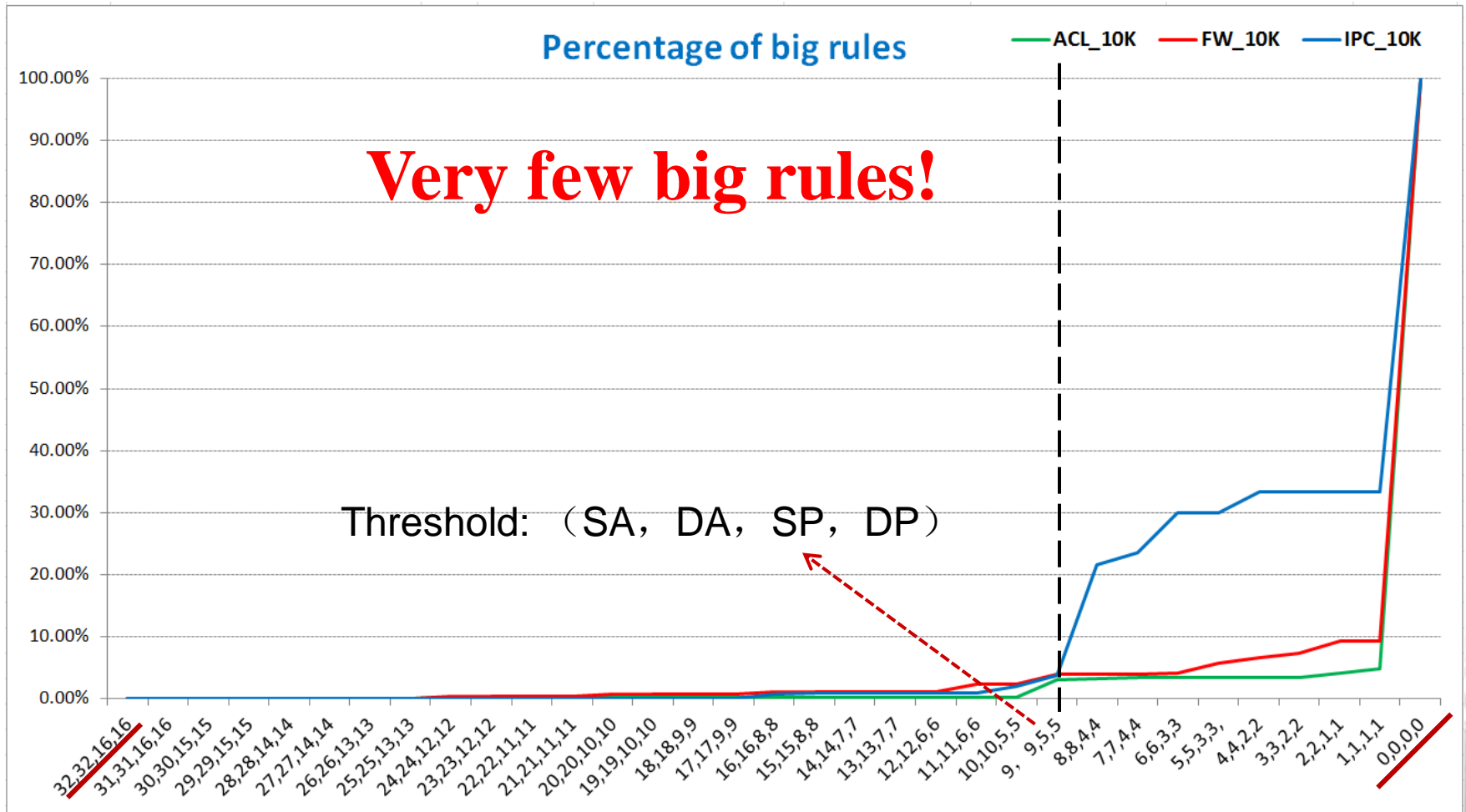
HybridCuts

A two-stage scheme

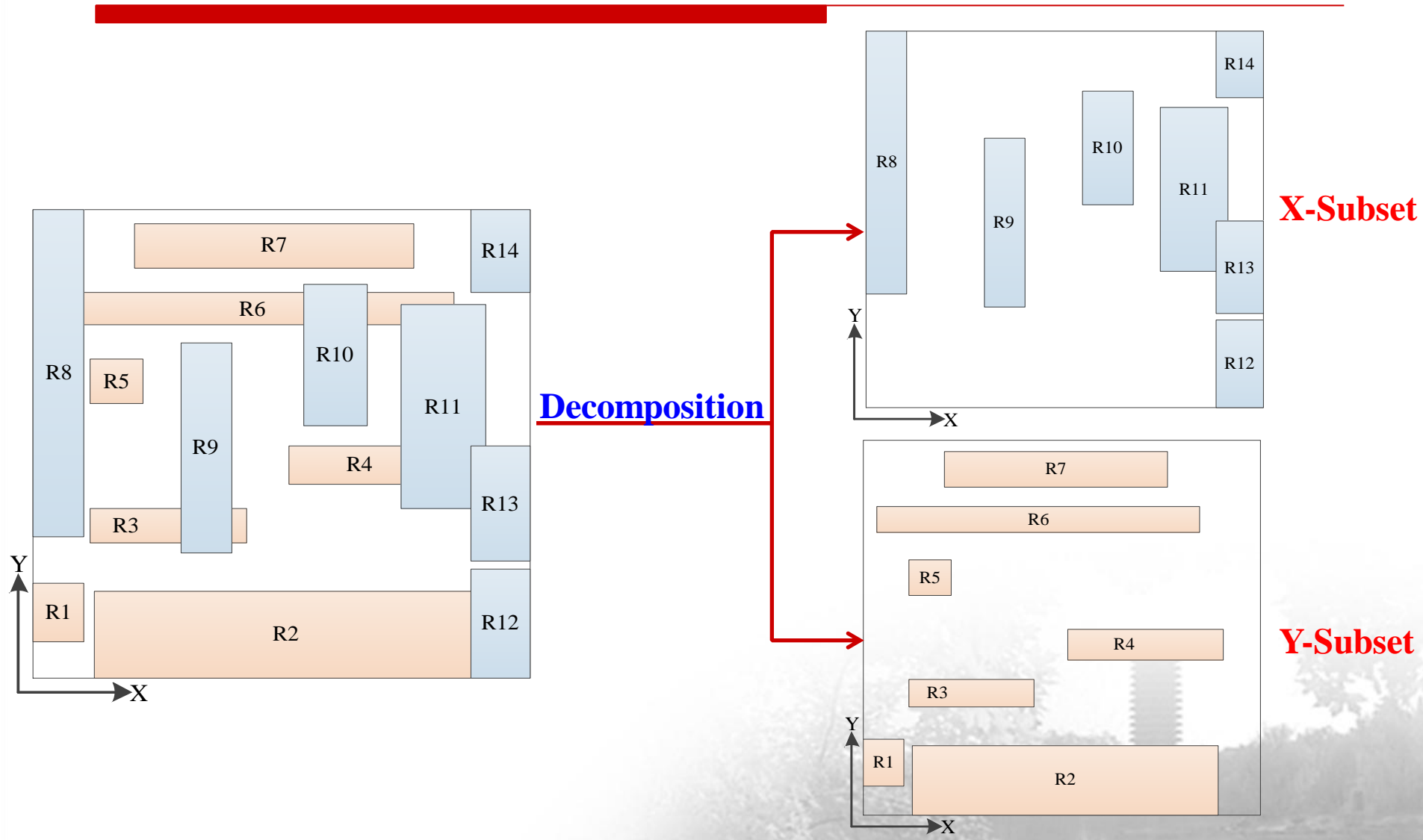


Preprocessing && Constructing search structure

Observations (1)

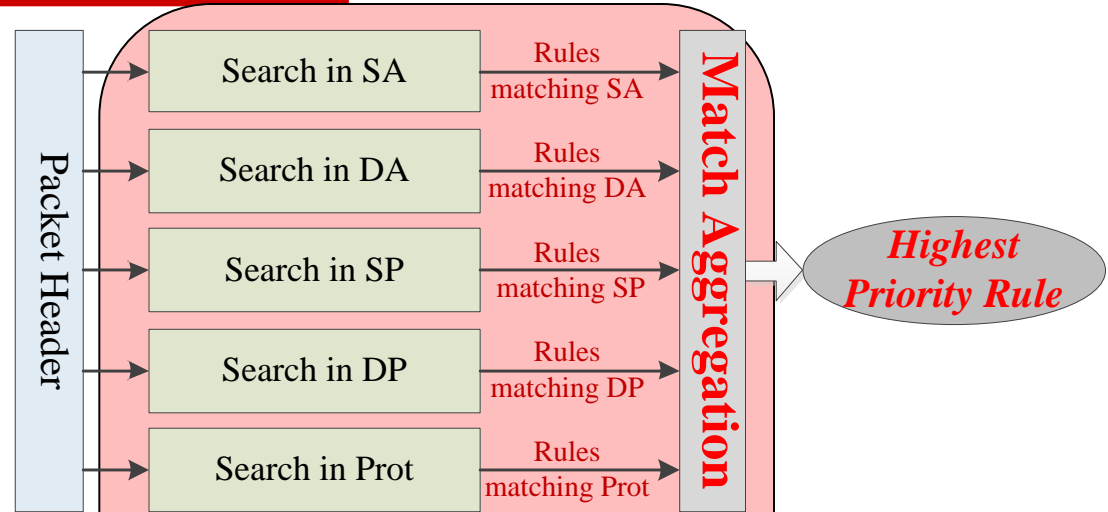


Observation (2)

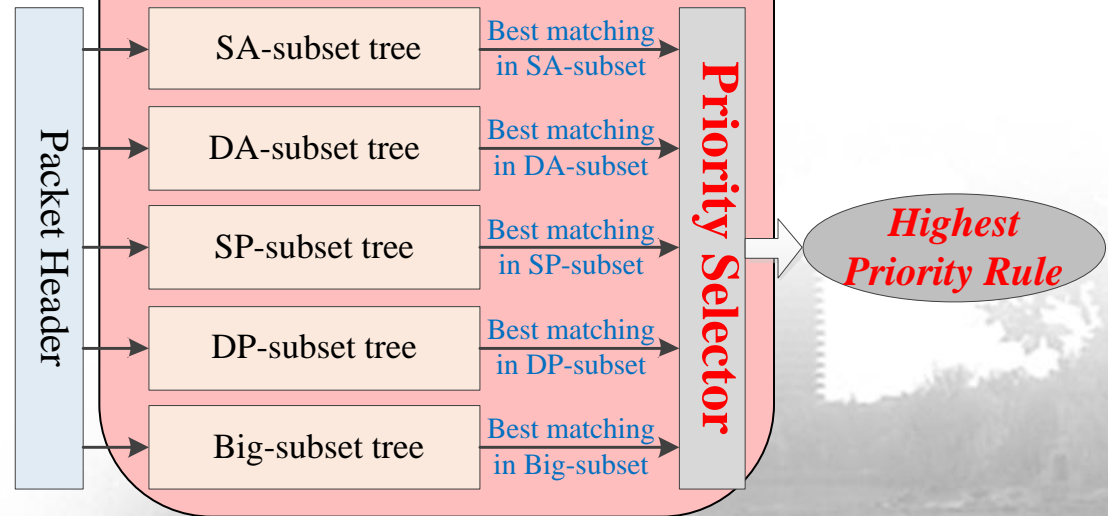


Decomposition

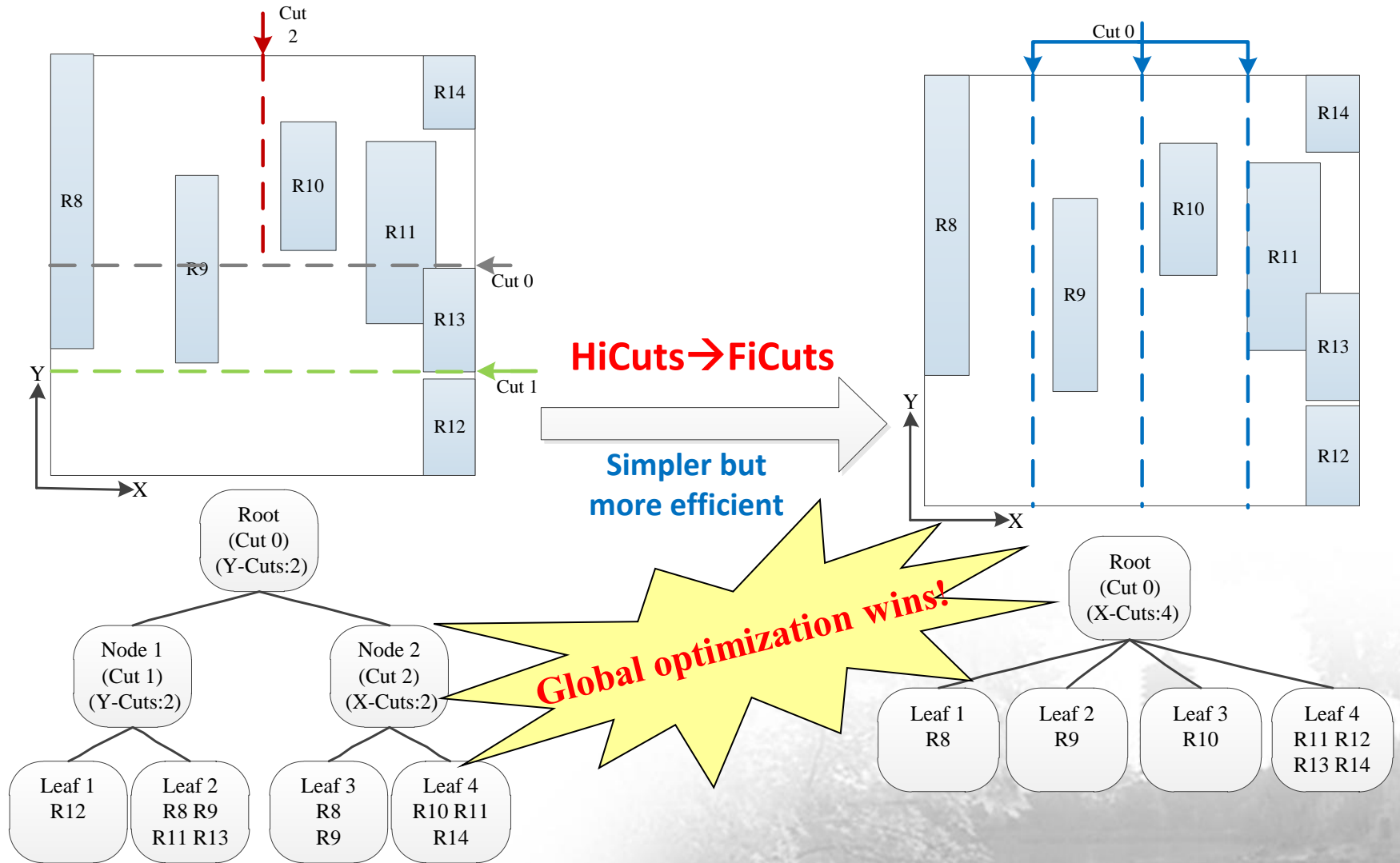
Traditional Decomposition



Improved Decomposition



FiCuts: Fixed intelligent Cuttings



A hybrid FiCuts + HyperCuts

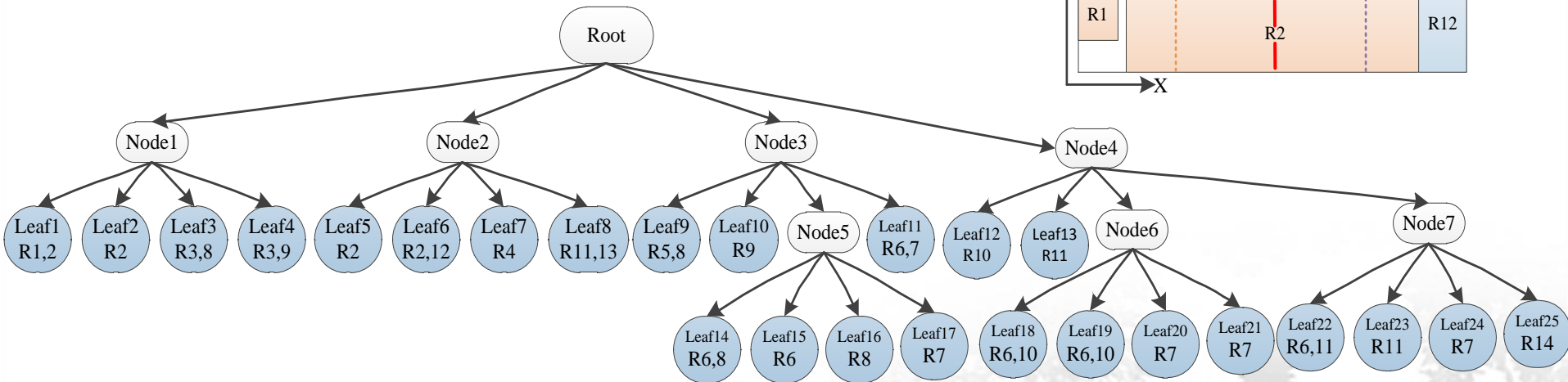
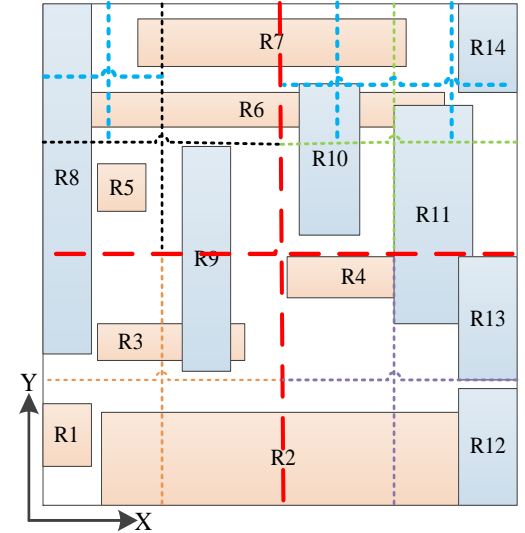


□ When to switch to HyperCuts?

- Subspace becomes small, and rule replication becomes intense
- A threshold to trigger the FiCut=>HyperCuts switching

Effectiveness (1)

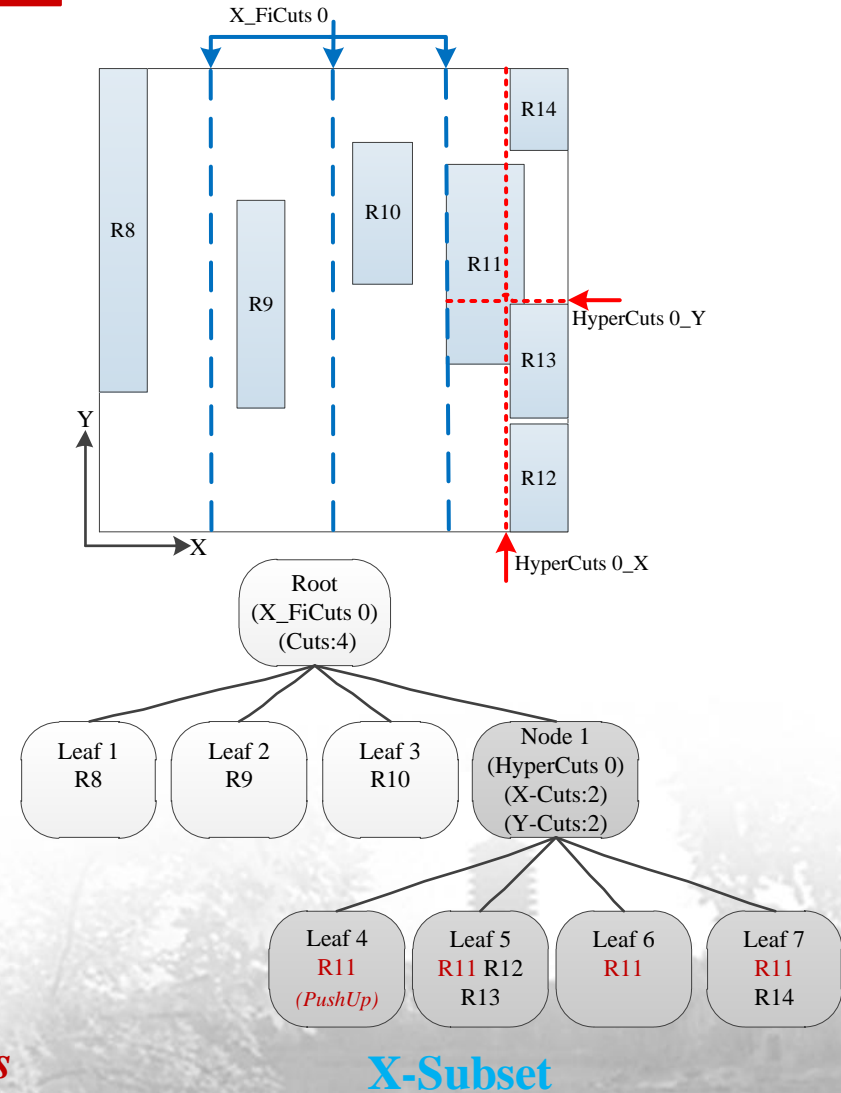
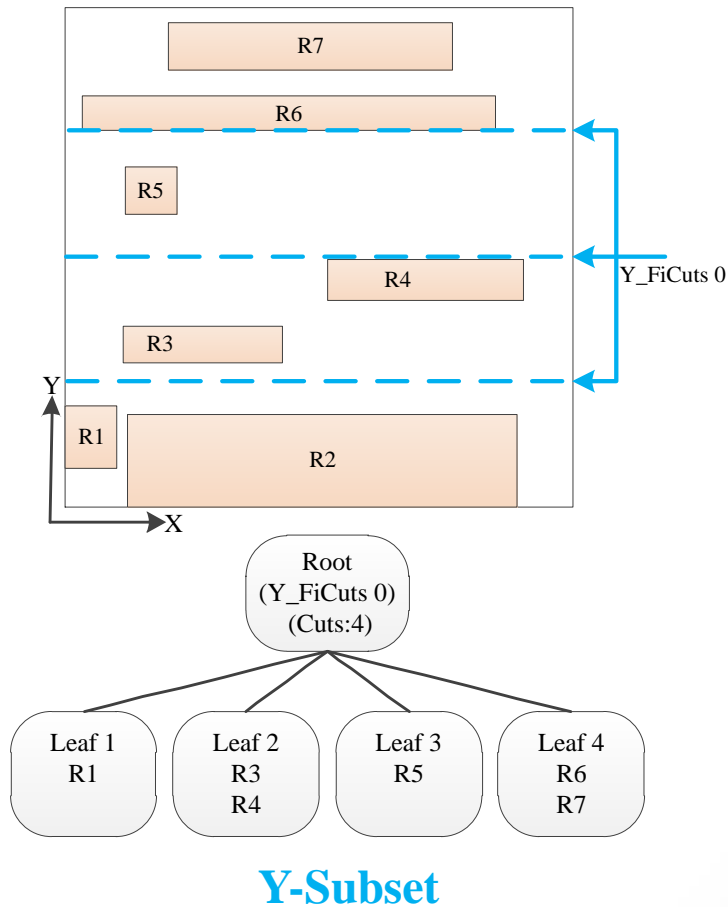
■ 14 → 36 rules



HyperCuts

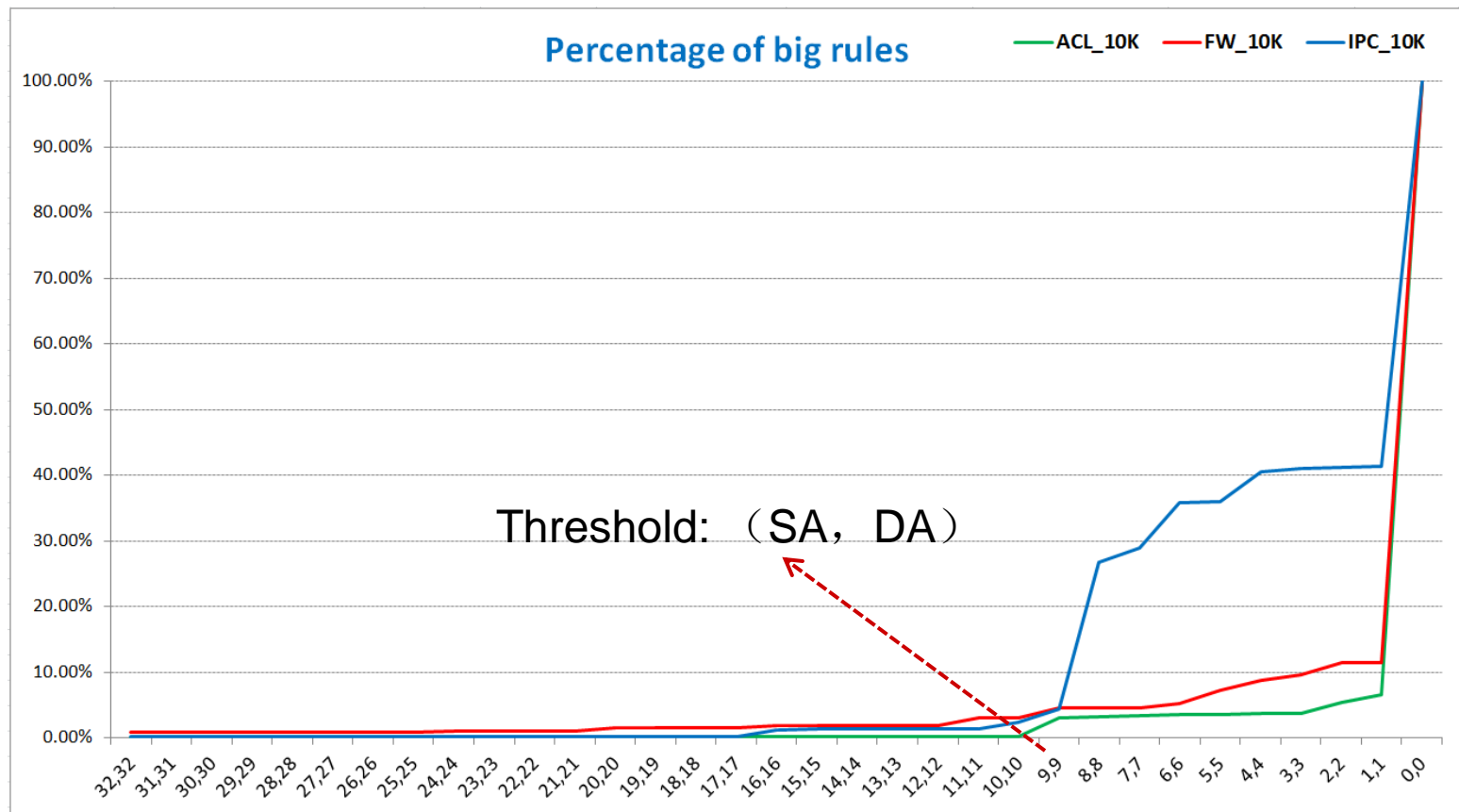
Effectiveness (2)

■ 14 \rightarrow 14 rules



Optimization

■ Can be smaller? 5→3 subsets





PART III: Evaluation



Experimental Setup

□ Tested with

- A publicly available rule set from Washington University
 - Used the ACL & FW & IPC 1K, 10K
- ClassBench
 - Generate ACL & FW & IPC 100K

□ Compared with

- HyperCuts & EffiCuts

□ Primary metrics

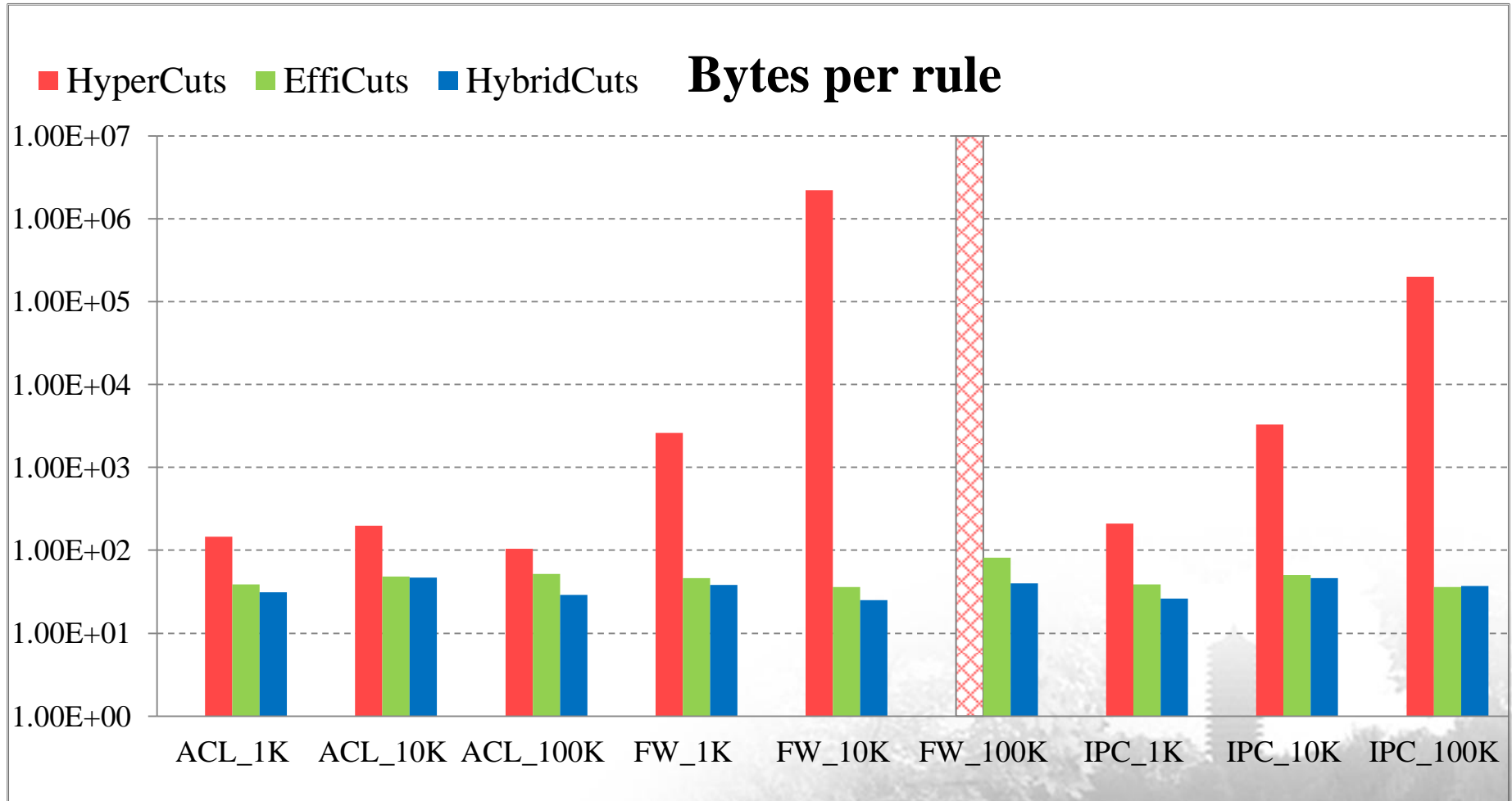
- Memory consumption (Bytes/rule)
- Number of memory accesses

□ Open Source for HybridCuts

- <https://github.com/lwj4333765/HybridCuts>

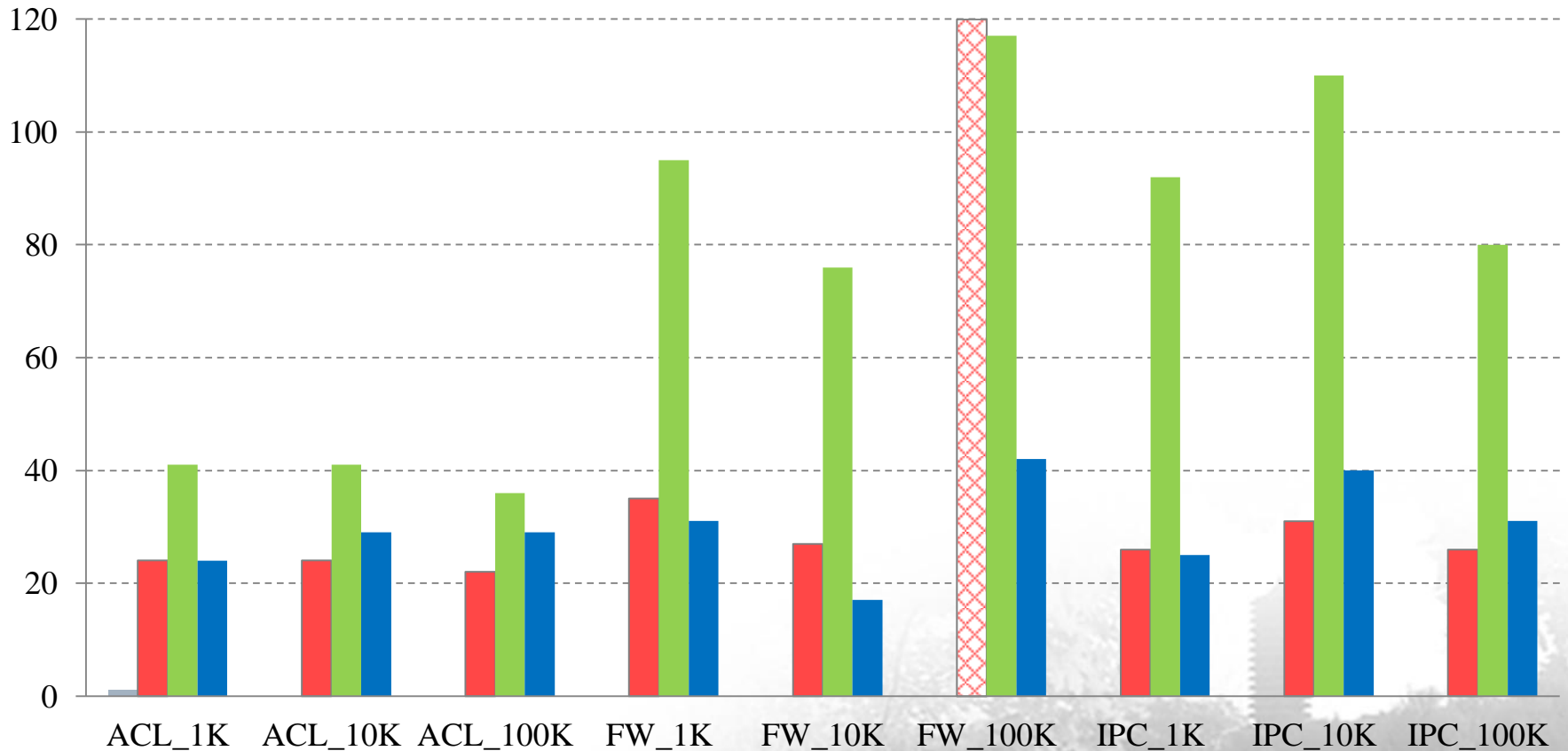


Memory Consumption



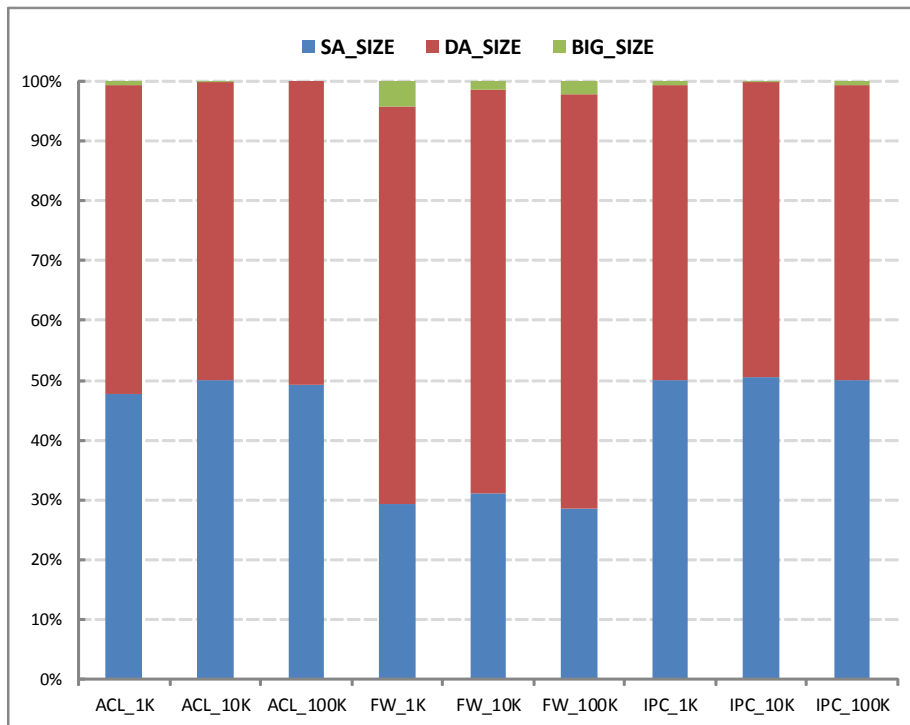
Memory Accesses

■ HyperCuts ■ EffiCuts ■ HybridCuts **Overall Memory Accesses**

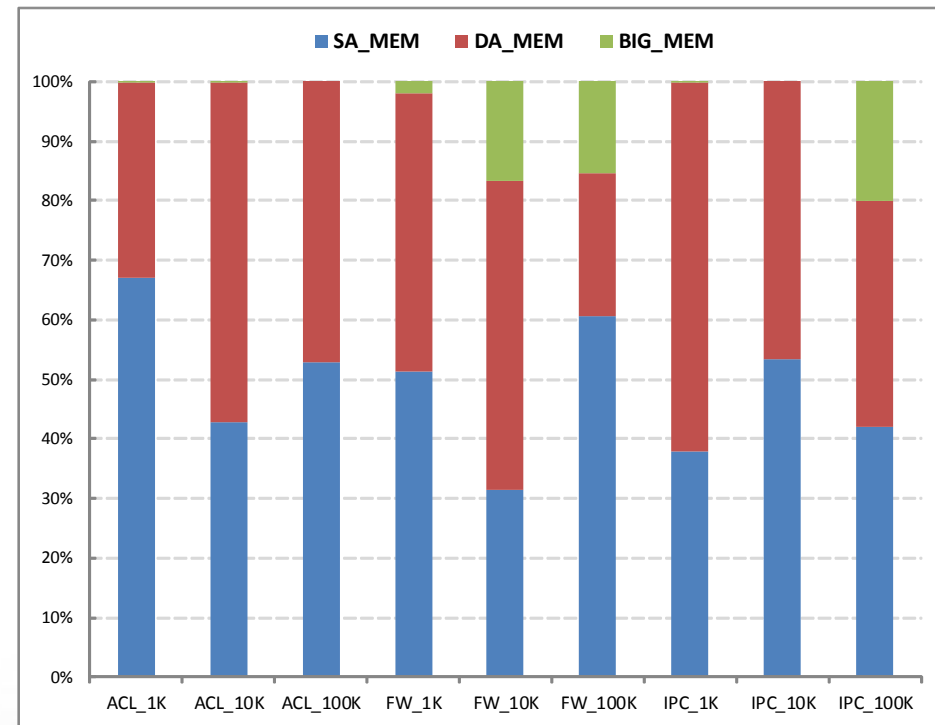


More Insights

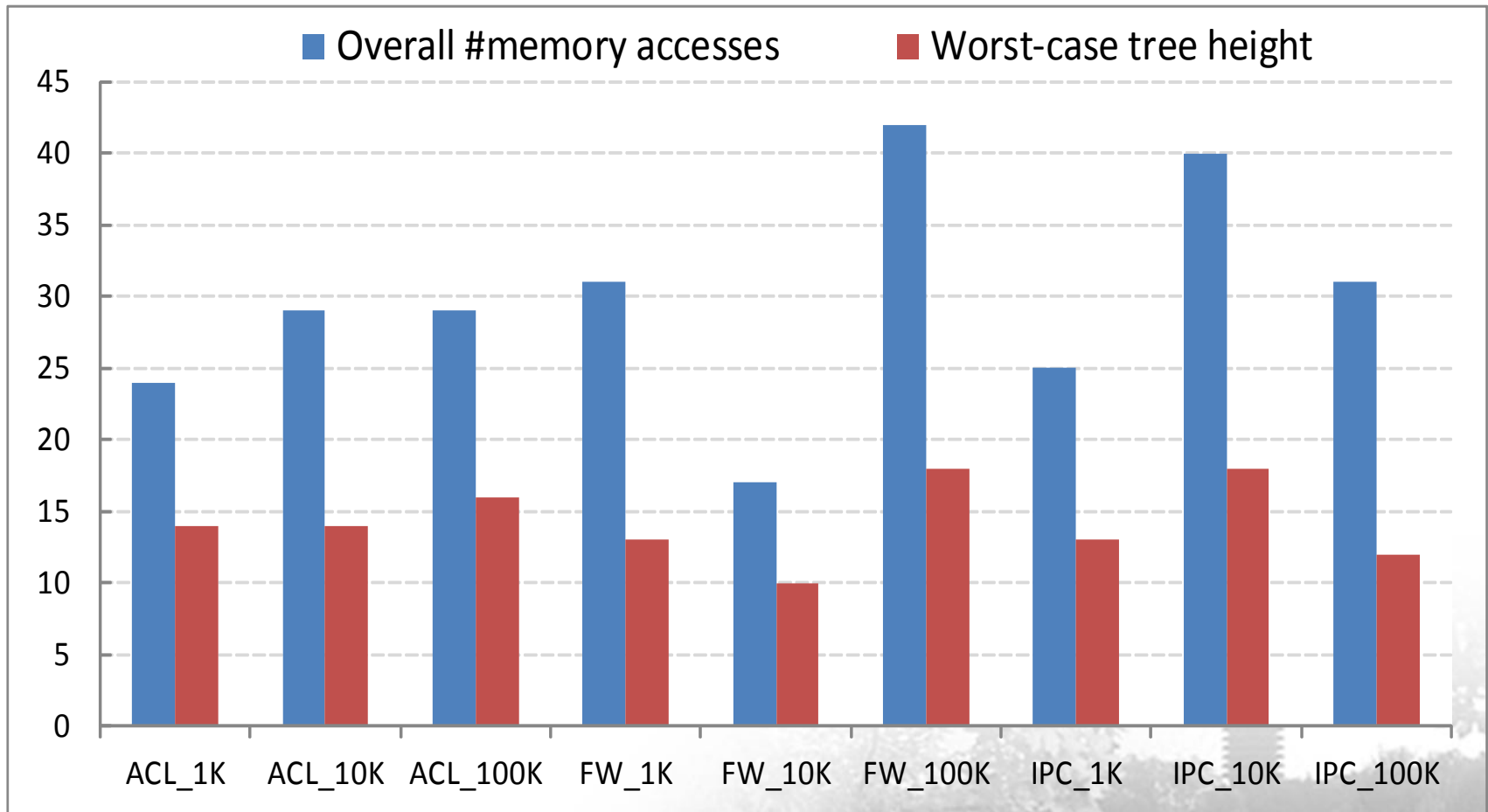
The sizes of subsets



The sizes of trees



Potential Gain with Parallelization





PART V: Conclusion

Conclusion

□ HybridCuts:

decomposition + cutting

- New observations
- A new rule set decomposition
- A hybrid One- + Multi- dimensional cutting

□ Future Works

- OpenFlow
- Software-hardware combined, e.g., FPGA
- Combine with TCAM



北京大學
PEKING UNIVERSITY

Thank you!

E-mail: liwenjun@sz.pku.edu.cn

Web: <http://liwenjun.weebly.com>