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

Diagram:

- **Incoming Packet**
  - Header
  - Payload

- **Router / Firewall**
  - Forwarding Engine
  - Flow Classification
  - Classifier (Rule Database)

- **Outgoing Packet**
  - Header
  - Payload

### Table

<table>
<thead>
<tr>
<th>#</th>
<th>SA</th>
<th>DA</th>
<th>SP</th>
<th>DP</th>
<th>Prot</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>1.2.3.0/20</td>
<td>192.168.0.1</td>
<td>[1,65534]</td>
<td>[1,65534]</td>
<td>TCP</td>
<td>accept</td>
</tr>
<tr>
<td>r2</td>
<td>1.2.3.11/24</td>
<td>1.2.3.11/24</td>
<td>80</td>
<td>[1,65534]</td>
<td>UDP</td>
<td>accept</td>
</tr>
<tr>
<td>r3</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>discard</td>
</tr>
</tbody>
</table>
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**

Diagram showing decomposition and cutting with packets and nodes.
PART II: HybridCuts
HybridCuts

A two-stage scheme

1\textsuperscript{st} Stage
Decomposition

- SA-subset
- DA-subset
- SP-subset
- DP-subset
- Big-subset

2\textsuperscript{nd} Stage
Cutting

- Decision Tree SA
- Decision Tree DA
- Decision Tree SP
- Decision Tree DP
- Decision Tree Big

Preprocessing & Constructing search structure
Observations (1)

Very few big rules!

Threshold: (SA, DA, SP, DP)
Observation (2)

Decomposition

X-Subset

Y-Subset
Decomposition

Traditional Decomposition

Packet Header
- Search in SA
- Search in DA
- Search in SP
- Search in DP
- Search in Prot

Match Aggregation
- Rules matching SA
- Rules matching DA
- Rules matching SP
- Rules matching DP
- Rules matching Prot

Priority Selector
- Best matching in SA-subset
- Best matching in DA-subset
- Best matching in SP-subset
- Best matching in DP-subset
- Best matching in Big-subset

Highest Priority Rule

Improved Decomposition

Packet Header
- SA-subset tree
- DA-subset tree
- SP-subset tree
- DP-subset tree
- Big-subset tree

Match Aggregation
- Rules matching SA
- Rules matching DA
- Rules matching SP
- Rules matching DP
- Rules matching Prot

Priority Selector
- Best matching in SA-subset
- Best matching in DA-subset
- Best matching in SP-subset
- Best matching in DP-subset
- Best matching in Big-subset

Highest Priority Rule
FiCuts: Fixed intelligent Cuttings

HiCuts $\rightarrow$ FiCuts

Simpler but more efficient

Global optimization wins!
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 → 14 rules

Y-Subset

HybridCuts

X-Subset
Optimization

- Can be smaller? 5 → 3 subsets

Threshold: (SA, DA)
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](https://github.com/lwj4333765/HybridCuts)
Memory Consumption

Bytes per rule

- HyperCuts
- EffiCuts
- HybridCuts

ACL_1K  ACL_10K  ACL_100K  FW_1K  FW_10K  FW_100K  IPC_1K  IPC_10K  IPC_100K
Memory Accesses

Overall Memory Accesses

HyperCuts  EffiCuts  HybridCuts

ACL_1K  ACL_10K  ACL_100K  FW_1K  FW_10K  FW_100K  IPC_1K  IPC_10K  IPC_100K
More Insights

The sizes of subsets

The sizes of trees
Potential Gain with Parallelization

Overall #memory accesses
Worst-case tree height

ACL_1K ACL_10K ACL_100K FW_1K FW_10K FW_100K IPC_1K IPC_10K IPC_100K
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
Thank you!

E-mail: liwenjun@sz.pku.edu.cn
Web: http://liwenjun.weebly.com