vCRIB: Virtual Cloud Rule Information Base

Masoud Moshref, Minlan Yu, Abhishek Sharma, Ramesh Govindan
HotCloud 2012
Datacenters use rules to implement **management policies**

- Access control
- Rate limiting
- Traffic measurement
- Traffic engineering
Datacenters use **rules** to implement management policies.

An **action** on a hypercube of **flow space**

Examples:
- Deny
- Normal
- Enqueue

Flow fields examples:
- Src IP / Dst IP
- Protocol
- Src Port / Dst Port
Datacenters use rules to implement management policies.

An action on a hypercube of flow space.

R1:
- SrcIP: 12.0.0.0/8
- DstIP: 10.0.0.0/16

R1=Accept
Current Practice

Rules are saved on **predefined fixed** machines
Current Practice

1. Machines have limited resources
2. Datacenters have different resource constraints
3. Multiple policies may compete for resources

Introduction | Motivation | Architecture | Evaluation | Conclusion
vCRIB Goal: Flexible Rule Placement

Find the best feasible rule placement based on resource constraints
Future Datacenters will have many fine-grained rules

**Regulating VM pair communication**
- Access control (CloudPolice)
- Bandwidth allocation (Seawall)

**Per flow decision**
- Flow measurement for traffic engineering (MicroTE, Hedera)

**VLAN per server**
- Traffic management (NetLord, Spain)
Where to place rules? Hypervisors vs. Switches

**Hypervisor**
- Performance: Software, Slow
- Flexibility: Complex rules
- Entry point: Close to VMs
- Resources: Limited CPU budget

**Switch**
- Performance: Hardware, Fast
- Flexibility: OpenFlow rules
- Entry point: External traffic, Aggregate traffic
- Resources: # TCAM entries
Storing rules at hypervisor incurs CPU processing overhead
Move the rule to ToR switch and forward traffic
Can we reduce Open vSwitch CPU usage?

The set of ignore bits in the mask
R1=Accept, DstIP: 10.0.0.0/16, SrcIP: 12.0.0.0/8
111111111111111111111111111111
111111111111111111111111111111

# wildcard patterns
# rules

CPU usage

Rules

Wildcard Pattern

# wildcard patterns
# rules

The set of ignore bits in the mask
R1=Accept, DstIP: 10.0.0.0/16, SrcIP: 12.0.0.0/8
111111111111111111111111111111, 111111111111111111111111111111

Introduction  Motivation  Architecture  Evaluation  Conclusion
Can we reduce Open vSwitch CPU usage?

- **Motivation**
  - Handle same number of new flows with lower CPU budget
Rule Location Trade-off (Resource vs. Bandwidth Usage)

If rule memory is limited in one switch
Can tradeoff bandwidth within the switch fabric, in addition to trading-off bandwidth between hypervisors and switches.
Our Approach: vCRIB, a Virtual Cloud Rule Information Base

Proactive rule placement abstraction layer

Allow operators to define fine-grained rules without worrying about placement

vCRIB

R1
R3
R2
R4

Rules
Our Approach: vCRIB, a Virtual Cloud Rule Information Base

**Flexible rule placement at hypervisors and switches**

**Optimize performance given resource constraints**

Rules

vCRIB

- R1
- R2
- R3
- R4

Network State

Agg1

ToR1

Agg2

ToR2

ToR3
Challenges: Overlapping Rules

- R0
- R1
- R2
- R3
- R4
Challenges: Overlapping Rules
Challenges: Overlapping Rules

Partitions rules to reduce overlapping rules dependency
Challenges: Overlapping Rules

Partitions rules to reduce overlapping rules dependency

Splitting rules covering multiple partitions causes inflation
vCRIB: Partitioning

Recursively cut partitions to create a BSP tree

Select a cut that
• balances two partitions
• creates fewest number of new rules

Smaller partitions
• are more flexible to place
• match fewer communicating VMs

Stop whenever a resource at a node is exhausted
Challenges: Placement Complexity

Constraints
• Functionality
• Machine resources

Goal
• Minimize traffic overhead
• Minimize delay
• Minimize cost of bandwidth usage vs. saved CPU

• Different partition sizes
• Different machine capacities
• Different traffic overhead for each partition location

Generalized Assignment Problem
vCRIB: Placement (Branch and Bound)

Select the largest unassigned partition

Place it on a switch/hypervisor

- Capable of handling its rules
  - Functionality
  - Resources
- Make minimum traffic overhead
vCRIB Architecture

Traffic and Topology information

Rule Placement

Rules

Partitioning

Placement

Partitioning

ToR1
Evaluation: Goal

Can partitioning algorithm achieve small partitions?

Can placement algorithm leverage resource availability to decrease traffic overhead?

Configuration
- 100 VMs per machine
- 10K flows (10KB) per machine
- ClassBench rules
- 1K rule capacity per switches
Evaluation: Partitioning

Change rule capacity to show the effect of different CPU budgets

Maximum size of partitions goes down as resources increase
Evaluation: Placement

For each machine select VM addresses from a contiguous IP range.

Traffic decreases as resources increase.

Aggregated addresses make lower traffic overhead.
Evaluation: Placement

No traffic decrease

Replication
vCRIB provides an abstraction layer for placement of rules in datacenters

Places the rules on both hypervisors and switches to achieve the best performance given the resource constraints
Future Work

- Exploit performance model of hypervisors & switches
- Online Algorithm adjusting to traffic changes
- Replication in the partitioning and placement algorithm
vCRIB: Virtual Cloud Rule Information Base

Masoud Moshref, Minlan Yu, Abhishek Sharma, Ramesh Govindan
HotCloud 2012