Control-Plane Scalability

COS 597E: Software Defined Networking

Jennifer Rexford
Princeton University

MW 11:00am-12:20pm

Controller Scalability Challenges

- Number of switches
  - Number of control sessions
  - Frequency of events
  - Size of topology state
- Performance metrics
  - Throughput: flow rules installed per second
  - Latency: delay to handle a packet-in

2

NOX Controller

- NOX controller
  - Single-threaded, unoptimized C++ controller
  - 30K flow initiations per second
  - Sub-10ms flow install time
- NOX-MT
  - Multi-threaded, better I/O handling, optimized malloc() implementation, …
  - On an 8-core machine, 1.6M request/sec and average response time of 2 msec

3

Distributed Controllers

- Better scalability
  - Smaller topology
  - Fewer events
  - Fewer flow installations
- Better performance
  - Closer to the switches
  - Lower control-plane latency
- Better reliability
  - Failover to a backup controller

4

Controller Configurations

- Hierarchical
  - Global controller, with multiple local controllers
- Peers
  - Each handling different portions of the topology, flow space, slice, or applications
- Replicas
  - Master, with multiple slaves
  - Multiple active replicas

5

Working with Multiple Controllers

- Local scope
  - MAC learning
  - Elephant flow detection
  - Aggregate/threshold traffic statistics
- Network-wide scope
  - Computing shortest paths
  - Selecting links to shut down to save energy
  - User mobility and virtual machine migration

6

Discussion Questions

- Does the programmer know there is a multi-threaded/distributed controller?
  - Automated partitioning? Language constructs?
- Does the application have to behave exactly the same as on a single controller?
  - Approximate shortest paths?
- Do the techniques for scalability interact with the techniques for fault tolerance?
  - Shared reliable distributed data store?
- Any useful changes to OpenFlow protocol?

7

Going Through Some Examples

- MAC learning
- Histogram by source IP address
- Stateful firewall
- Shortest-path routing
- MAC learning at edge, shortest-path in core
- Traffic engineering
- Consistent updates
- Hedera (elephant flow routing)

8