CSC 2229 – Software-Defined Networking

Handout # 5: Scaling Controllers in SDN - Kandoo

Professor Yashar Ganjali
Department of Computer Science
University of Toronto

yganjali@cs.toronto.edu
http://www.cs.toronto.edu/~yganjali

Joint work with Soheil Hassas Yeganeh
EVENTS

- Rare
  - Link state changes

- Frequent and Exhaustive
  - Network-wide stat collection
  - Packet-ins (if flow-entries are not installed proactively)
SCALABILITY ISSUES

Frequent events stress the control plane.

- Stress the control channels.
- Stress controller’s resources.
EXISTING SOLUTIONS

Distributed Controllers:

• Consider this as an intrinsic limitation.
• HyperFlow, Onix, Devolved Controllers, ...

Data Plane Extensions:

• Delegate more responsibilities to the data plane.
• DIFANE, DevoFlow, ...

CSC2229 -- Software-Defined Networking
University of Toronto -- Fall 2015
EXISTING SOLUTIONS

Still, high control channel consumption.

Comes at the cost of visibility.

Need to modify the data plane.
How to handle frequent events close to the metal without modifying OpenFlow?
THE IDEA

OFFLOADING

LOCAL CONTROL APPS

TO

LOCAL RESOURCES.

Applications that do not need the network-wide state.

Resources close to switches.
Local Apps

- An assumption in distributed controllers:
  - All control apps require the network-wide state.

- But, there are many apps that are local in scope:
  - Applications that require only local switch state.
LOCAL APPS

• Local applications:
  • Learning Switch
  • Local Policy Enforcer
  • Link Discovery

• Local components in control applications:
  • Elephant Flow Detection in an Elephant Flow Rerouting application.

Local apps have implicit parallelism.
We can **offload** local apps to computing resources **next to switches**.
**KANDOO**

- **Two layers of controllers:**
  - A logically centralized **Root Controller**.
  - **Local Controllers**.

  The root controller runs non-local apps.

  Light weight and easy to implement.

  Local controllers shield the root controller.

  Local controllers run local apps.

The diagram shows:

- **Root Controller**
- **Local Controller**
- **Switch**

- **Rare Events**
- **Frequent Events**
AN EXAMPLE: ELEPHANT FLOW REROUTEING

Diagram:
- Switch
- OF Controller
  - App_reroute
  - App_detect
- Stat Request
- Stat Reply
AN EXAMPLE: ELEPHANT FLOW REROUTING

Application-specific events. Scales linearly with the number of switches.

Kandoo's event channels.
SIMPLE, YET FLEXIBLE, ARCHITECTURE

Scale at the edge.

One local controller per switch!
EVALUATION SUMMARY

• Implemented Kandoo:
  • Handles 1.3 Mp/s on a single core of Xeon E7-4807.

• Elephant Flow Rerouting:
  • In an emulated environment.
  • More than 5x less channel consumption.
  • Significantly better scalability in regards to the network size.
FINAL COMMENTS

- Controller scalability is a concern in SDN
  - Control channels
  - Controller resources

- Distributed controllers alleviate the problem to some extent
  - Many complications

- Developers are not shielded
  - As originally promised

- Troubleshooting, debugging still complex