

Project Title: SDNC+ Learning

Student's Name: Monika Patel

Course : TELE6400 (SDN), Semester: 2nd

Professor's Name: Prof. Dr. Bhumip KHASNABISH

Presentation Date, Location & Time: 8th December 2019

Outline

- Abstract and Summary
- Types of Load Balancing Techniques
- Main Focus of the Project
- Usefulness of the Current Work
- Simulation/Emulation Details
- Setup & steps
- Results
- Q&A and Discussion
- References

Abstract and Summary

Load balancing is a technique to distribute the Load among available resources and it is supervised learning. Load balancing helps to optimize resources, maximize throughput, minimize response time and avoid overload of any single resource.

Flow Clustering is the task of dividing the population or data points into a number of groups.

Clustering is unsupervised learning because it automatically divides the data into clusters, or groups.

Types of Load Balancing Techniques

- Random IP selection: Backend server IP selection performed on random basis
- Round robin IP selection: Backend server IP selection performed upon one by one chronological order
- Weighted load IP selection: Backend server IP selection performed upon load condition on server.

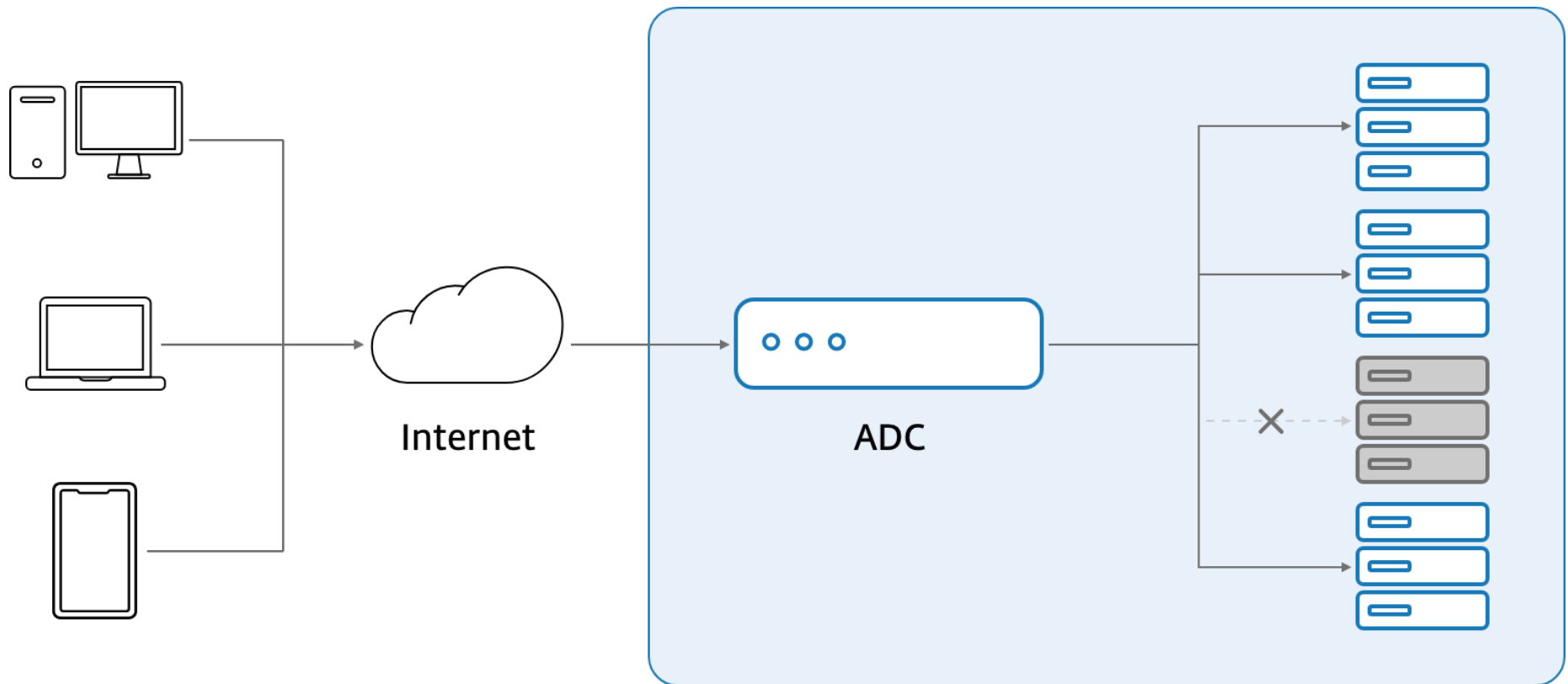
Main Focus of the Project

- A Software-Defined Network-based load balancing scheme is proposed for distributing client's requests among a group of available servers at which the next request is forwarded to the least-loaded server.
- The performance of the proposed scheme is evaluated with the Round-robin and we also analyze the effects on the overall network performance.
- The performance are tested in Mininet emulation environment and I have used pox controller.

Usefulness of the current work

- Increased Scalability. If you have a website, you must be uploading engaging content to attract readers. ...
- Redundancy. Advertisement. ...
- Reduced Downtime, Increased Performance.
- Efficiently Manages Failures. ...
- Increased Flexibility.

Topology Diagram



Setup and steps

- In SDN, we must separate data plane and control plane
- Hence, we can enhance and add scalability to the system
- Here, I have used “ POX Controller” as a control plane element and “Mininet” as a data plane element.
- Download and install Mininet on pc (<http://mininet.org/download/>). This can be performed in various ways,

Setup and steps contd...

- Mininet VM installation
- Native installation from source
- Installation from packages
- Upgrading an existing Mininet Installation

Open the terminal in Ubuntu OS and perform following list of commands

```
$git clone git://github.com/mininet/mininet
```

```
$cd mininet
```

```
$git tag
```

```
$git checkout -b 2.2.1 (latest version)
```

```
$mininet/util/install.sh-a ("a" to install everything using home directory)
```

It will take time and after completion check whether mininet is installed successfully or not by using the command

```
$sudo mn -test pingall
```

Result

```
boot.pyc  datapaths/  __init__.py  license.pyo  proto/  tk.py
boot.pyo  forwarding/  __init__.pyc  log/  py.py  topology/
core.py  help.py  __init__.pyo  messenger/  py.pyc  web/
core.pyc  host_tracker/  lib/  misc/  py.pyo
ubuntu@sdnhubvm:~/pox/pox[09:07] (eel)$ ./pox.py log.level --DEBUG misc.ip_loadb
balancer --ip=10.0.0.254 --servers=10.0.0.1,10.0.0.2,10.0.0.3
bash: ./pox.py: No such file or directory
ubuntu@sdnhubvm:~/pox/pox[09:09] (eel)$ cd
ubuntu@sdnhubvm:~/pox[09:09] (eel)$ ./pox.py log.level --DEBUG misc.ip_loadbalan
ter --ip=10.0.0.254 --servers=10.0.0.1,10.0.0.2,10.0.0.3
POX 0.5.0 (eel) / Copyright 2011-2014 James McCauley, et al.
DEBUG:core:POX 0.5.0 (eel) going up...
DEBUG:core:Running on CPython (2.7.6/Nov 13 2018 12:45:42)
DEBUG:core:Platform is Linux-3.13.0-24-generic-x86_64-with-Ubuntu-14.04-trusty
INFO:core:POX 0.5.0 (eel) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
INFO:openflow.of_01:[00-00-00-00-00-01 1] connected
INFO:iplb:IP Load Balancer Ready.
INFO:iplb:Load Balancing on [00-00-00-00-00-01 1]
INFO:iplb.00-00-00-00-00-01:Server 10.0.0.1 up
INFO:iplb.00-00-00-00-00-01:Server 10.0.0.2 up
INFO:iplb.00-00-00-00-00-01:Server 10.0.0.3 up
```

Result

[illegible]

Results

```
Terminal
File Edit View Terminal Tabs Help
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.3
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.2
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.4
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.3
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.2
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.4
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.3
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.2
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.4
FO:ip1b.00-00-00-00-00-01:Server 10.0.0.1 up
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.4
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.3
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.2
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.1
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.4
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.3
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.2
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.1
BUG:ip1b.00-00-00-00-00-01:Directing traffic to 10.0.0.4

*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> xterm h1
mininet> xterm h2
mininet> xterm h3
mininet> xterm h4
mininet> xterm h5

Node: h5
<li><a href="/mininet/">mininet</a>
<li><a href="/Music/">Music</a>
<li><a href="/oflops/">oflops</a>
<li><a href="/onos/">onos</a>
<li><a href="/openflow/">openflow</a>
<li><a href="/openvswitch/">openvswitch</a>
<li><a href="/Pictures/">Pictures</a>
<li><a href="/pox/">pox</a>
<li><a href="/Public/">Public</a>
<li><a href="/pyretic/">pyretic</a>
<li><a href="/ryu/">ryu</a>
<li><a href="/sdnhub.png">sdnhub.png</a>
<li><a href="/SDNHub_Opendaylight_Tutorial/">SDNHub_Opendaylight_Tutorial</a>
<li><a href="/Templates/">Templates</a>
<li><a href="/test.py">test.py</a>
<li><a href="/Topology.py">Topology.py</a>
<li><a href="/Topology.py.py">Topology.py.py</a>
<li><a href="/trema/">trema</a>
<li><a href="/Videos/">Videos</a>
</ul>
<hr>
</body>
</html>
root@sdnhubvm:~[08:01]$
```

Results

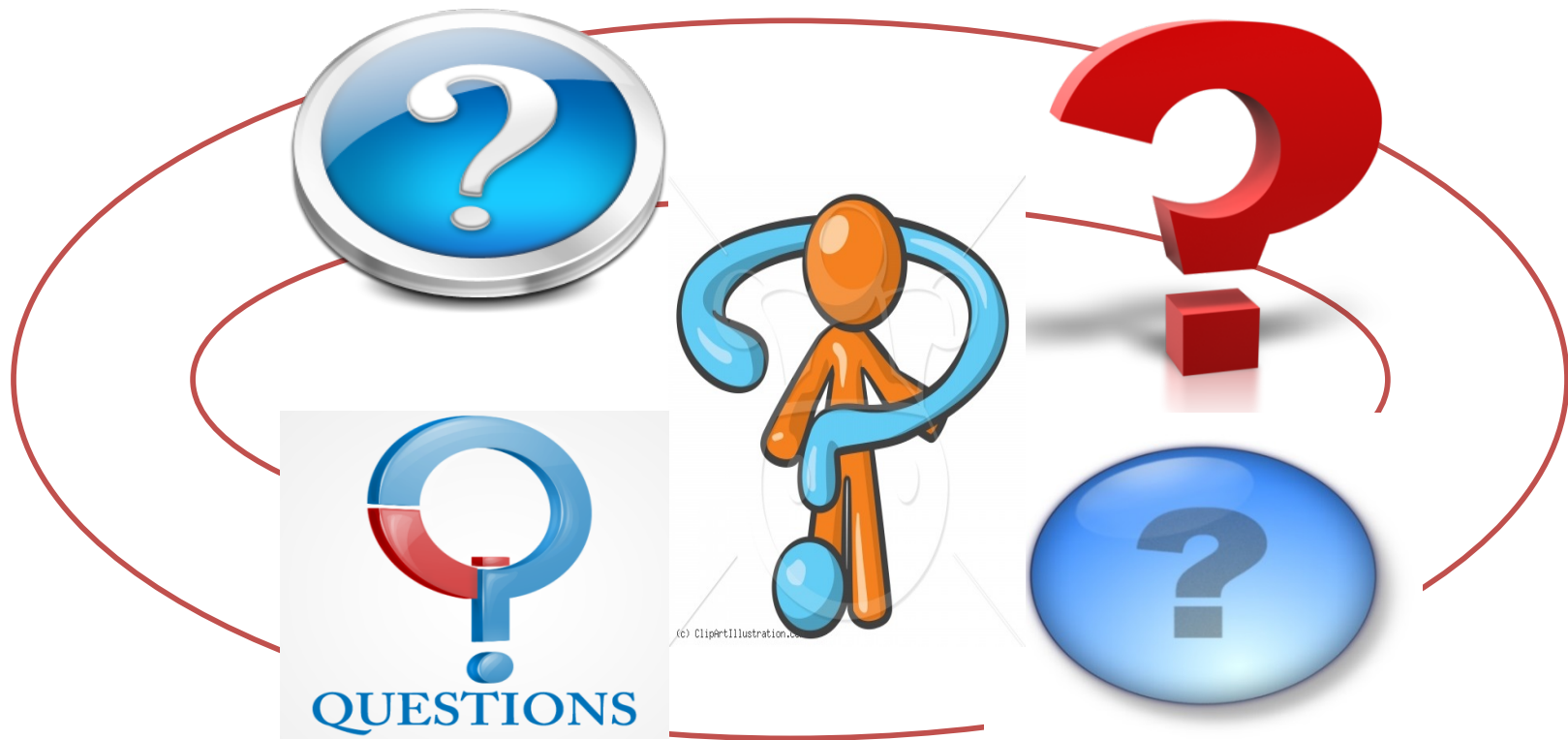
```
Terminal
e Edit View Terminal Tabs Help
UG:iplb.00-00-00-00-00-01:Expired 2 flows
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.3
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.4
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.3
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.4
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.3
UG:iplb.00-00-00-00-00-01:Server 10.0.0.2 up
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.2
UG:iplb.00-00-00-00-00-01:Expired 2 flows
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.4
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.3
UG:iplb.00-00-00-00-00-01:Expired 2 flows
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.2
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.4
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.3
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.2
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.4
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.3
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.2
UG:iplb.00-00-00-00-00-01:Directing traffic to 10.0.0.4

*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1) (h5, s1) (h6, s1)
*** Configuring hosts
h1 h2 h3 h4 h5 h6
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> xterm h1
mininet> xterm h2
mininet> xterm h3
mininet> xterm h4

Node: h5
<li><a href="mininet/">mininet/</a>
<li><a href="Music/">Music/</a>
<li><a href="oflops/">oflops/</a>
<li><a href="onos/">onos/</a>
<li><a href="openflow/">openflow/</a>
<li><a href="openvswitch/">openvswitch/</a>
<li><a href="Pictures/">Pictures/</a>
<li><a href="pox/">pox/</a>
<li><a href="Public/">Public/</a>
<li><a href="pyretic/">pyretic/</a>
<li><a href="ryu/">ryu/</a>
<li><a href="sdnhub.png">sdnhub.png</a>
<li><a href="SDNHub_Openaylight_Tutorial/">SDNHub_Openaylight_Tutorial/</a>
<li><a href="Templates/">Templates/</a>
<li><a href="test.py">test.py</a>
<li><a href="Topology.py">Topology.py</a>
<li><a href="Topology.py.py">Topology.py.py</a>
<li><a href="trema/">trema/</a>
<li><a href="Videos/">Videos/</a>
</ul>
<hr>
</body>
</html>
root@sdnhubvm:~[08:00]$
```

Q&A and Discussion

Thanks for Your KIND Attention



DISCUSSION

References

- <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8260023&tag=1>
- <https://www.youtube.com/watch?v=uyFXzsrBeVQ>
- <https://github.com/T-NOVA/SDN-Control-Plane-Load-Balancer>

Pox Controller

- POX is networking software written in python
- POX currently communicates on OpenFlow version 1.0
- POX rich features:
 1. Pythonic OpenFlow interface
 2. Reusable sample components for path selection, topology discovery
 3. Runs anywhere
 4. Targets Linux, Mac OS and Windows