



Análise de Big Data Streams

Identificando e tratando fluxo Hadoop

Marcelo Vaz Netto
Nilson Moraes Filho



Propósito do Artigo Original

Programming Your Network at Run-time for Big Data Applications

- Discutir a otimização do fluxo de dados na rede através do gerenciamento SDN
- Alteração, em tempo real, da topologia da rede usando Switches Ópticos
- Exemplo: Processos Hadoop
 - *Hadoop é um framework que permite o processamento distribuído de grandes conjuntos de dados através de um cluster de computadores utilizando um modelo simples de programação*

Finalidade do nosso projeto

Análise de Big Data Streams – tratando fluxo Hadoop

- Objetivo: Otimizar fluxo interno de dados em grandes Data Centers
- Criar um ambiente dentro do Mininet para execução dos processos Hadoop.
- Instalar toda a funcionalidade Hadoop no ambiente Mininet
 - Não utilizar diferentes ambientes virtualizados para o Hadoop
- Usando SDN, criar rotas específicas para tráfego de controle e tráfego de dados
 - Tráfego de dados simula a utilização dos switches ópticos
 - Tráfego de controle, por ser de menor volume, simula a utilização de switches tradicionais

Ambiente utilizado nos testes

● Laptop 1

- Windows XP, 4GB RAM, Intel Core Duo @2.26GHz
- VMWare Player
- Imagem Mininet 2.1.0-130919-ubuntu-1304-server-i386 com 2GB de RAM e 8GB de storage alocados
- Hadoop 1.2.1

● Laptop 2

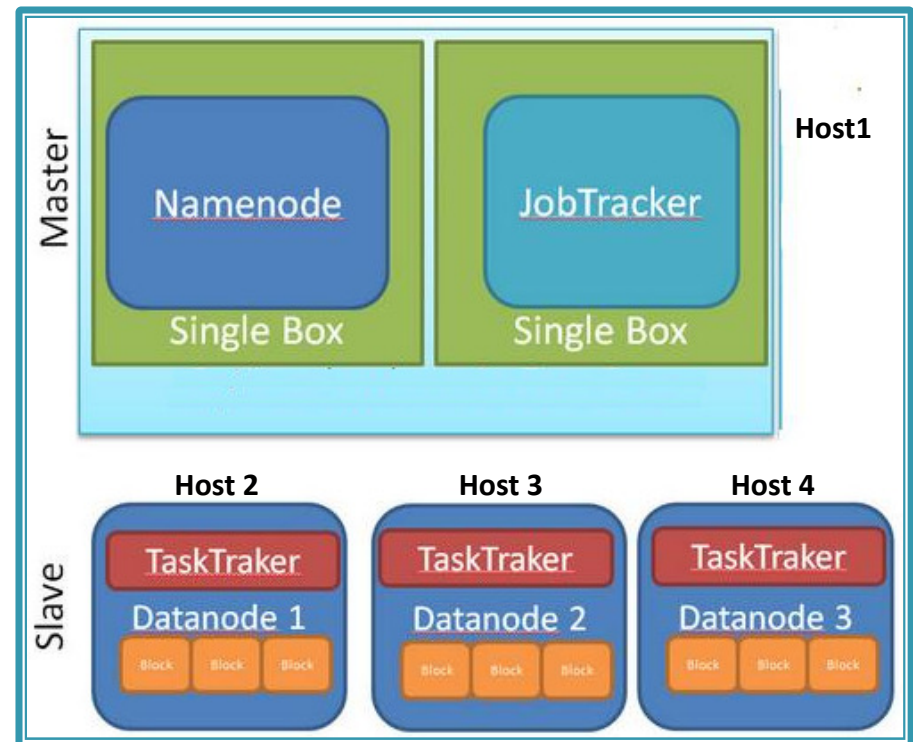
- Ubuntu 14.04, 3GB RAM, Intel processor
- Hadoop 1.2.1

Recordando um pouco sobre Hadoop

Principais processos do Hadoop:

- **NameNode**: é o ponto central do Hadoop File System (HDFS). Mantém a árvore de diretórios de todos os arquivos e gerencia onde o arquivo é armazenado através do cluster. Aplicações clientes trocam mensagens com o NameNode para localizar um arquivo.
- **DataNode**: armazena arquivos no HDFS
- **MapReduce**: é o algoritmo chave que distribui as tarefas através dos clusters
- **Job Tracker** – é um serviço dentro do Hadoop que distribui as tarefas MapReduce para nós específicos no cluster, de preferência onde os dados estão armazenados ou pelo menos no mesmo rack
- **Task Tracker**: é um nó do cluster que aceita tarefas – operações Map, Reduce e Shuffle – de um Job Tracker

Ambiente Mininet



Nosso primeiro desafio ...



● Problema:

- Mininet suporta *network namespaces* (virtualização leve que fornece individualmente aos processos interface de redes independentes e tabelas de roteamento e ARP separadas)
- Entretanto para rodar o Hadoop de forma distribuída em cada host, necessitávamos também da independência de processos (pid) e disco (mount)

● Solução:

- Utilizamos LXC Containers: o objetivo do LXC é o de criar um ambiente o mais próximo possível de uma instalação padrão Linux mas sem a necessidade de um kernel em separado
- LXC utiliza as seguintes funções do kernel para conter os processos: Kernel namespaces (ipc, uts, mount, pid, network and user), Chroots, Kernel capabilities e Control groups (cgroups)
- Outras opções de containers: Dockers, Debian ...

Primeiro Passo

Carregando Topologia Mininet e controlador POX

```
mininet@mininet-vm: ~/mininet/custom
mininet@mininet-vm:~/mininet/custom$ sudo ./mn.py
*** Creating nodes
Unable to contact the remote controller at 127.0.0.1:6633
*** Creating links
*** Starting network
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
*** Starting 8 switches
s1 s2 s3 s4 s5 s6 s7 s8
*** Running CLI
*** Starting CLI:
mininet> nodes
available nodes are:
c1 h1 h2 h3 h4 s1 s2 s3 s4 s5 s6 s7 s8
mininet> dump
<Host h1: h1-eth0:10.0.3.151 pid=1457>
<Host h2: h2-eth0:10.0.3.152 pid=1458>
<Host h3: h3-eth0:10.0.3.153 pid=1459>
<Host h4: h4-eth0:10.0.3.154 pid=1460>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth
pid=1468>
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth
pid=1473>
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth
pid=1473>
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth
pid=1483>
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth
pid=1488>
<OVSSwitch s6: lo:127.0.0.1,s6-eth1:None,s6-eth
pid=1493>
<OVSSwitch s7: lo:127.0.0.1,s7-eth1:None,s7-eth
pid=1498>
<OVSSwitch s8: lo:127.0.0.1,s8-eth1:None,s8-eth
pid=1498>
<RemoteController c1: 127.0.0.1:6633 pid=1505>
mininet>
```

```
mininet@mininet-vm: ~/pox
mininet@mininet-vm:~/pox$ ./pox.py v3_controller
POX 0.1.0 (beta) / Copyright 2011-2013 James McCauley, et al.
INFO:v3_controller:*** Starting... ***
INFO:v3_controller:Waiting for 8 switches
INFO:core:POX 0.1.0 (beta) is up.
INFO:openflow.of_01:[00-00-00-00-00-04 1] connected
INFO:openflow.of_01:[00-00-00-00-00-07 4] connected
INFO:openflow.of_01:[00-00-00-00-00-02 2] connected
INFO:openflow.of_01:[00-00-00-00-00-05 3] connected
INFO:openflow.of_01:[00-00-00-00-00-01 5] connected
INFO:openflow.of_01:[00-00-00-00-00-03 6] connected
INFO:openflow.of_01:[00-00-00-00-00-06 8] connected
INFO:openflow.of_01:[00-00-00-00-00-08 7] connected
INFO:v3_controller: *** Installing static flows... ***
INFO:v3_controller: *** Static flows installed. ***
```

Segundo Passo

Acessando os hosts (Xterm)

```
Node: h1
root@mininet-vm:~/mininet/custom# ifconfig
h1-eth0 Link encap:Ethernet HWaddr 00:00:00:00:00:01
        inet addr:10.0.3.151 Bcast:10.255.255.255 Mask:255.0.0.0
        inet6 addr: fe80::200:ff:fe00:1/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B) TX bytes:558 (558.0 B)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:65536 Metric:1
        RX packets:250 errors:0 dropped:0 overruns:0 frame:0
        TX packets:250 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:60960 (60.9 KB) TX bytes:60960 (60.9 KB)

root@mininet-vm:~/mininet/custom#
```

```
Node: h3
root@mininet-vm:~/mininet/custom# ifconfig
h3-eth0 Link encap:Ethernet HWaddr 00:00:00:00:00:03
        inet addr:10.0.3.153 Bcast:10.255.255.255 Mask:255.0.0.0
        inet6 addr: fe80::200:ff:fe00:3/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B) TX bytes:636 (636.0 B)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:65536 Metric:1
        RX packets:247 errors:0 dropped:0 overruns:0 frame:0
        TX packets:247 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:66984 (66.9 KB) TX bytes:66984 (66.9 KB)

root@mininet-vm:~/mininet/custom#
```

```
Node: h2
root@mininet-vm:~/mininet/custom# ifconfig
h2-eth0 Link encap:Ethernet HWaddr 00:00:00:00:00:02
        inet addr:10.0.3.152 Bcast:10.255.255.255 Mask:255.0.0.0
        inet6 addr: fe80::200:ff:fe00:2/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B) TX bytes:636 (636.0 B)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:65536 Metric:1
        RX packets:287 errors:0 dropped:0 overruns:0 frame:0
        TX packets:287 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:68244 (68.2 KB) TX bytes:68244 (68.2 KB)

root@mininet-vm:~/mininet/custom#
```

```
Node: h4
root@mininet-vm:~/mininet/custom# ifconfig
h4-eth0 Link encap:Ethernet HWaddr 00:00:00:00:00:04
        inet addr:10.0.3.154 Bcast:10.255.255.255 Mask:255.0.0.0
        inet6 addr: fe80::200:ff:fe00:4/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:9 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B) TX bytes:706 (706.0 B)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:65536 Metric:1
        RX packets:225 errors:0 dropped:0 overruns:0 frame:0
        TX packets:225 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:56428 (56.4 KB) TX bytes:56428 (56.4 KB)

root@mininet-vm:~/mininet/custom#
```


Terceiro Passo

Iniciando os containers



```
Node: h1
root@mininet-vm:~/mininet/custom# sudo lxc-start -d -n NameNode
root@mininet-vm:~/mininet/custom# sudo lxc-attach -n NameNode
root@NameNode:/# ifconfig
h1-eth0  Link encap:Ethernet  HWaddr 00:00:00:00:00:01
         inet addr:10.0.3.151  Bcast:10.0.3.255  Mask:255.255.255.0
         inet6 addr: fe80::200:ff:fe00:1/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:36 errors:0 dropped:0 overruns:0 frame:0
         TX packets:28 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:1512 (1.5 KB)  TX bytes:1768 (1.7 KB)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:65536  Metric:1
         RX packets:8 errors:0 dropped:0 overruns:0 frame:0
         TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:752 (752.0 B)  TX bytes:752 (752.0 B)

root@NameNode:/# █
```

Quarto Passo

Iniciando o Hadoop



```
Node: h1
root@NameNode:~# start-all.sh
starting namenode, logging to /usr/local/hadoop/libexec/./logs/hadoop-root-name
node-NameNode.out
10.0.3.152: starting datanode, logging to /usr/local/hadoop/libexec/./logs/hado
op-root-datanode-DataNode1.out
10.0.3.154: starting datanode, logging to /usr/local/hadoop/libexec/./logs/hado
op-root-datanode-DataNode3.out
10.0.3.153: starting datanode, logging to /usr/local/hadoop/libexec/./logs/hado
op-root-datanode-DataNode2.out
10.0.3.151: starting secondarynamenode, logging to /usr/local/hadoop/libexec/./
logs/hadoop-root-secondarynamenode-NameNode.out
starting jobtracker, logging to /usr/local/hadoop/libexec/./logs/hadoop-root-jo
btracker-NameNode.out
10.0.3.153: starting tasktracker, logging to /usr/local/hadoop/libexec/./logs/h
adoop-root-tasktracker-DataNode2.out
10.0.3.152: starting tasktracker, logging to /usr/local/hadoop/libexec/./logs/h
adoop-root-tasktracker-DataNode1.out
10.0.3.154: starting tasktracker, logging to /usr/local/hadoop/libexec/./logs/h
adoop-root-tasktracker-DataNode3.out
root@NameNode:~# █
```

Quinto Passo

Verificando os processos do Hadoop

```
Node: h1
root@NameNode:~# jps
528 JobTracker
693 Jps
310 NameNode
461 SecondaryNameNode
root@NameNode:~#
```

```
Node: h2
root@DataNode1:~# jps
480 Jps
372 TaskTracker
292 DataNode
root@DataNode1:~#
```

```
Node: h3
root@DataNode2:~# jps
371 TaskTracker
291 DataNode
472 Jps
root@DataNode2:~#
```

```
Node: h4
root@DataNode3:~# jps
368 TaskTracker
288 DataNode
475 Jps
root@DataNode3:~#
```

JPS: Java Virtual Machine Process Status Tool – lista as JVM rodando no sistema

Sexto Passo

Executando o Hadoop – Testando com o wordcount

```
Node: h1
root@NameNode:/# hadoop jar /usr/local/hadoop/hadoop*examples*.jar wordcount /data/4300.txt /data/results10
14/07/08 17:31:38 INFO input,FileInputFormat: Total input paths to process : 1
14/07/08 17:31:38 INFO util,NativeCodeLoader: Loaded the native-hadoop library
14/07/08 17:31:38 WARN snappy,LoadSnappy: Snappy native library not loaded
14/07/08 17:31:49 INFO mapred,JobClient: Running job: job_201407081715_0001
14/07/08 17:31:50 INFO mapred,JobClient: map 0% reduce 0%
14/07/08 17:34:25 INFO mapred,JobClient: map 100% reduce 0%
14/07/08 17:34:58 INFO mapred,JobClient: map 100% reduce 33%
14/07/08 17:35:01 INFO mapred,JobClient: map 100% reduce 66%
14/07/08 17:35:09 INFO mapred,JobClient: map 100% reduce 100%
14/07/08 17:35:46 INFO mapred,JobClient: Job complete: job_201407081715_0001
14/07/08 17:35:46 INFO mapred,JobClient: Counters: 29
14/07/08 17:35:46 INFO mapred,JobClient: Map-Reduce Framework
14/07/08 17:35:46 INFO mapred,JobClient: Spilled Records=153927
14/07/08 17:35:46 INFO mapred,JobClient: Map output materialized bytes=740321
14/07/08 17:35:46 INFO mapred,JobClient: Reduce input records=51309
14/07/08 17:35:46 INFO mapred,JobClient: Virtual memory (bytes) snapshot=582385664
14/07/08 17:35:46 INFO mapred,JobClient: Map input records=33056
14/07/08 17:35:46 INFO mapred,JobClient: SPLIT_RAW_BYTES=102
14/07/08 17:35:46 INFO mapred,JobClient: Map output bytes=2601826
14/07/08 17:35:46 INFO mapred,JobClient: Reduce shuffle bytes=740321
14/07/08 17:35:46 INFO mapred,JobClient: Physical memory (bytes) snapshot=186499072
14/07/08 17:35:46 INFO mapred,JobClient: Reduce input groups=50095
14/07/08 17:35:46 INFO mapred,JobClient: Combine output records=51309
14/07/08 17:35:46 INFO mapred,JobClient: Reduce output records=50095
14/07/08 17:35:46 INFO mapred,JobClient: Map output records=267980
14/07/08 17:35:46 INFO mapred,JobClient: Combine input records=267980
14/07/08 17:35:46 INFO mapred,JobClient: CPU time spent (ms)=6710
14/07/08 17:35:46 INFO mapred,JobClient: Total committed heap usage (bytes)=132190208
14/07/08 17:35:46 INFO mapred,JobClient: File Input Format Counters
14/07/08 17:35:46 INFO mapred,JobClient: Bytes Read=1573078
14/07/08 17:35:46 INFO mapred,JobClient: FileSystemCounters
14/07/08 17:35:46 INFO mapred,JobClient: HDFS_BYTES_READ=1573180
14/07/08 17:35:46 INFO mapred,JobClient: FILE_BYTES_WRITTEN=2333043
14/07/08 17:35:46 INFO mapred,JobClient: FILE_BYTES_READ=1480648
14/07/08 17:35:46 INFO mapred,JobClient: HDFS_BYTES_WRITTEN=527555
14/07/08 17:35:46 INFO mapred,JobClient: Job Counters
14/07/08 17:35:46 INFO mapred,JobClient: Launched map tasks=1
14/07/08 17:35:46 INFO mapred,JobClient: Launched reduce tasks=1
14/07/08 17:35:46 INFO mapred,JobClient: SLOTS_MILLIS_REDUCES=43976
14/07/08 17:35:46 INFO mapred,JobClient: Total time spent by all reduces waiting after reserving slots (ms)=0
14/07/08 17:35:46 INFO mapred,JobClient: SLOTS_MILLIS_MAPS=99032
14/07/08 17:35:46 INFO mapred,JobClient: Total time spent by all maps waiting after reserving slots (ms)=0
14/07/08 17:35:46 INFO mapred,JobClient: Data-local map tasks=1
14/07/08 17:35:46 INFO mapred,JobClient: File Output Format Counters
14/07/08 17:35:46 INFO mapred,JobClient: Bytes Written=527555
root@NameNode:/#
```

Aplicação wordcount vem com o Hadoop

Verificando os Resultados

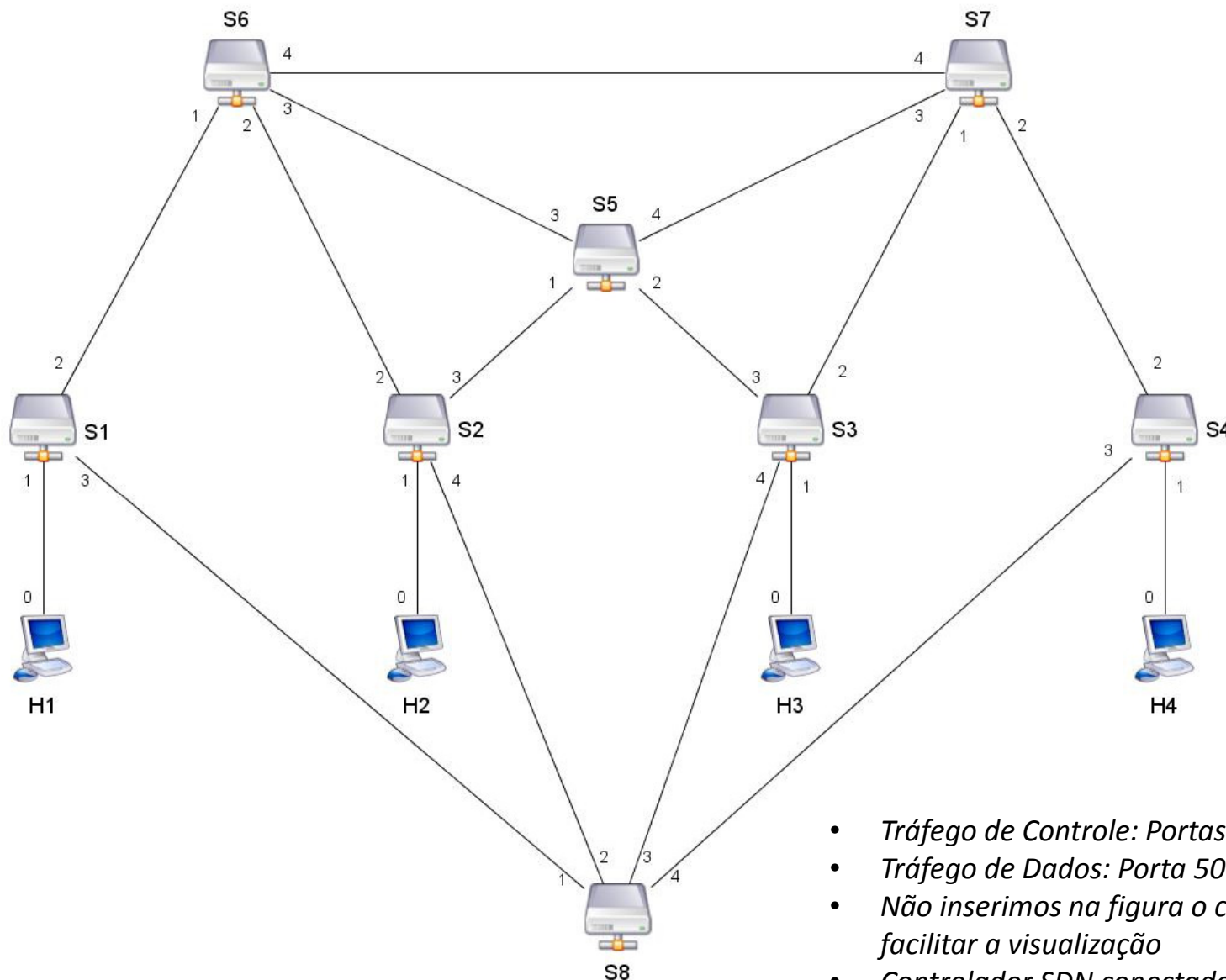
Contando as palavras do livro Ulysses de James Joyce

```
Node: h1
root@NameNode:/tmp# hadoop fs -ls /data
Found 6 items
-rw-r--r--  3 root supergroup  1573078 2014-06-09 13:32 /data/4300.txt
drwxr-xr-x  - root supergroup      0 2014-06-09 13:38 /data/results
drwxr-xr-x  - root supergroup      0 2014-07-02 03:27 /data/results1
drwxr-xr-x  - root supergroup      0 2014-07-08 17:35 /data/results10
drwxr-xr-x  - root supergroup      0 2014-06-20 00:08 /data/results5
drwxr-xr-x  - root supergroup      0 2014-06-20 00:56 /data/results6
root@NameNode:/tmp# hadoop fs -ls /data/results10
Found 3 items
-rw-r--r--  3 root supergroup      0 2014-07-08 17:35 /data/results10/_SUCCESS
drwxr-xr-x  - root supergroup      0 2014-07-08 17:31 /data/results10/_logs
-rw-r--r--  3 root supergroup  527555 2014-07-08 17:35 /data/results10/part-r-00000
root@NameNode:/tmp# hadoop fs -copyToLocal /data/results10/part-r-00000 .
root@NameNode:/tmp# █
```

```
Node: h1
Allsop, 2
Allude 1
Alluding 1
Alluring, 1
Ally 1
Almany 2
Almidano 7
Almighty 4
Almighty's 1
Almonds 1
Almost 4
Alo! 1
Aloft 1
Alone, 2
Alone. 1
Along 3
Aloysius 6
Alpha 1
Alpha, 1
Alphonsus 1
Alphonsus, 1
Alpine 1
Also 20
Also, 1
Although 1
Always 18
Am 8
Amant 1
Ambidexterity 1
Ambrose 1
Amby? 1
Amen! 2
Amen. 8
Amen._ 1
America 3
America, 1
America, 5
American 6
American, 1
Amid 3
Amiens 4
Ammon 1
Amn't 2
Among 1
Amongst 1
Amor 2
Amor, 1
Amsterdam 1
Amsterdam, 1
Amused 1
An 58
Analogous 1
Ananias 1
Anarchist, 1
Anchor, 1
Ancient 1
```

Como foi implementado

Topologia e SDN



Roteamento de Dados

H1 -> H2 : S1--S6--S2
H1 -> H3 : S1--S6--S5--S3
H1 -> H4 : S1--S6--S7--S4
H2 -> H1 : S2--S6--S1
H2 -> H3 : S2--S5--S3
H2 -> H4 : S2--S5--S7--S4
H3 -> H1 : S3--S5--S6--S1
H3 -> H2 : S3--S5--S2
H3 -> H4 : S3--S7--S4
H4 -> H1 : S4--S7--S6--S1
H4 -> H2 : S4--S7--S5--S2
H4 -> H3 : S4--S7--S3

Roteamento de Controle

H1 -> H2 : S1--S8--S2
H1 -> H3 : S1--S8--S3
H1 -> H4 : S1--S8--S4
H2 -> H1 : S2--S8--S1
H3 -> H1 : S3--S8--S1
H4 -> H1 : S4--S8--S1

- *Tráfego de Controle: Portas 10001 e 10002 – Switch S8*
- *Tráfego de Dados: Porta 50010*
- *Não inserimos na figura o controlador SDN para facilitar a visualização*
- *Controlador SDN conectado a todos os Switches*

Regras do Controlador POX (exemplos de regras)

```
#Flow 1.3.3:
switch133 = 1
flow133msg = of.ofp_flow_mod()
flow133msg.cookie = 0
flow133msg.priority = 30
flow133msg.match.in_port = 1
flow133msg.match.nw_proto = 6
flow133msg.match.dl_type = 0x800
flow133msg.match.tp_src = 10002
# ACTIONS-----
flow133out = of.ofp_action_output (port = 3)
flow133msg.actions = [flow133out]

#-----
# SWITCH 2
#-----
#Flow 2.1.1:
switch211 = 2
flow211msg = of.ofp_flow_mod()
flow211msg.cookie = 0
flow211msg.priority = 20
flow211msg.match.dl_dst = EthAddr("00:00:00:00:00:02")
# ACTIONS-----
flow211out = of.ofp_action_output (port = 1)
flow211msg.actions = [flow211out]

#Flow 2.1.2:
switch212 = 2
flow212msg = of.ofp_flow_mod()
flow212msg.cookie = 0
flow212msg.priority = 30
flow212msg.match.in_port = 4
flow212msg.match.dl_type = 0x0806
# ACTIONS-----
flow212out = of.ofp_action_output (port = 1)
flow212msg.actions = [flow212out]
```

Pacote TCP
Porta de Controle

Tráfego para o
Switch S2

0x0806 = ARP

```
#Flow 8.5.1:
switch851 = 8
flow851msg = of.ofp_flow_mod()
flow851msg.cookie = 0
flow851msg.priority = 30
flow851msg.match.dl_type = 0x0806
# ACTIONS-----
flow851out = of.ofp_action_output (port = of.OFPP_ALL)
flow851msg.actions = [flow851out]

def install_flows():
    log.info("    *** Installing static flows... ***")
    # Push flows to switches
    core.openflow.sendToDPID(switch111, flow111msg)
    core.openflow.sendToDPID(switch112, flow112msg)
    core.openflow.sendToDPID(switch121, flow121msg)
    core.openflow.sendToDPID(switch131, flow131msg)
    core.openflow.sendToDPID(switch132, flow132msg)
    core.openflow.sendToDPID(switch133, flow133msg)
```

Switch de Controle S8
recebe pacote ARP
Flood em todas as portas

Verificando os Resultados

Fluxo de Controle (wireshark)

The image shows the Wireshark interface with a network capture. The main pane displays a list of packets. Packet 2116 is highlighted in blue, indicating it is selected. The details pane below shows the structure of this packet: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. A red arrow points from the word 'Controle' in a box to the 'Seq: 2119, Ack: 1851, Len: 0' field in the TCP details pane.

No.	Time	Source	Destination	Protocol	Length	Info
2108	208.373552000	10.0.3.152	10.0.3.151	TCP	337	[TCP Retransmission] 43824 > documentum [PSH, ACK] Seq=545 /
2109	208.375688000	10.0.3.151	10.0.3.152	TCP	173	[TCP Retransmission] documentum > 43824 [PSH, ACK] Seq=1423
2110	208.375754000	10.0.3.152	10.0.3.151	TCP	66	43824 > documentum [ACK] Seq=816 Ack=1530 Win=29696 Len=0 TS
2111	208.737039000	10.0.3.154	10.0.3.151	TCP	337	42258 > documentum [PSH, ACK] Seq=1306 Ack=1530 Win=29696 Le
2112	208.738365000	10.0.3.151	10.0.3.154	TCP	173	documentum > 42258 [PSH, ACK] Seq=1530 Ack=1577 Win=34816 Le
2113	208.738423000	10.0.3.154	10.0.3.151	TCP	66	42258 > documentum [ACK] Seq=1577 Ack=1637 Win=29696 Len=0 T
2114	208.557486000	10.0.3.153	10.0.3.151	TCP	337	44409 > documentum [PSH, ACK] Seq=1848 Ack=1744 Win=29696 Le
2115	208.559332000	10.0.3.151	10.0.3.153	TCP	173	documentum > 44409 [PSH, ACK] Seq=1744 Ack=2119 Win=36864 Le
2116	208.559399000	10.0.3.153	10.0.3.151	TCP	66	44409 > documentum [ACK] Seq=2119 Ack=1851 Win=29696 Len=0 T
2117	208.702541000	10.0.3.153	10.0.3.151	TCP	382	52026 > scp-config [PSH, ACK] Seq=5519 Ack=2359 Win=44544 Le
2118	208.703628000	10.0.3.151	10.0.3.153	TCP	214	scp-config > 52026 [PSH, ACK] Seq=2359 Ack=5835 Win=55808 Le
2119	208.703686000	10.0.3.153	10.0.3.151	TCP	66	52026 > scp-config [ACK] Seq=5835 Ack=2507 Win=45568 Len=0 T
2120	208.557505000	10.0.3.153	10.0.3.151	TCP	337	[TCP Retransmission] 44409 > documentum [PSH, ACK] Seq=1848
2121	208.559309000	10.0.3.151	10.0.3.153	TCP	173	[TCP Retransmission] documentum > 44409 [PSH, ACK] Seq=1744
2122	208.559410000	10.0.3.153	10.0.3.151	TCP	66	44409 > documentum [ACK] Seq=2119 Ack=1851 Win=29696 Len=0 T
2123	208.693207000	10.0.3.152	10.0.3.151	TCP	337	43824 > documentum [PSH, ACK] Seq=816 Ack=1530 Win=29696 Len
2124	208.694509000	10.0.3.151	10.0.3.152	TCP	173	documentum > 43824 [PSH, ACK] Seq=1530 Ack=1087 Win=32256 Le
2125	208.694595000	10.0.3.152	10.0.3.151	TCP	66	43824 > documentum [ACK] Seq=1087 Ack=1637 Win=29696 Len=0 T
2126	208.702560000	10.0.3.153	10.0.3.151	TCP	382	[TCP Retransmission] 52026 > scp-config [PSH, ACK] Seq=5519
2127	208.703611000	10.0.3.151	10.0.3.153	TCP	214	[TCP Retransmission] scp-config > 52026 [PSH, ACK] Seq=2359
2128	208.703695000	10.0.3.153	10.0.3.151	TCP	66	52026 > scp-config [ACK] Seq=5835 Ack=2507 Win=45568 Len=0 T

▶ Frame 2116: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 1
▶ Ethernet II, Src: 00:00:00_00:00:03 (00:00:00:00:00:03), Dst: 00:00:00_00:00:01 (00:00:00:00:00:01)
▶ Internet Protocol Version 4, Src: 10.0.3.153 (10.0.3.153), Dst: 10.0.3.151 (10.0.3.151)
▶ Transmission Control Protocol, Src Port: 44409 (44409), Dst Port: documentum (10002), Seq: 2119, Ack: 1851, Len: 0

```
0000  00 00 00 00 00 01 00 00  00 00 00 03 08 00 45 00  .....E.  
0010  00 34 89 a3 40 00 40 06  95 f1 0a 00 03 99 0a 00  .4..@.@. ....  
0020  03 97 ad 79 27 12 a3 b1  b3 b1 90 2f 42 3d 80 10  ...y'... /B=..  
0030  00 3a 1b 56 00 00 01 01  08 0a 00 02 47 b1 00 02  ..:V.... .G..  
0040  47 b1
```


Verificando os Resultados

Fluxo de Dados (wireshark)

*18 interfaces [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Salvar

No.	Time	Source	Destination	Protocol	Length	Info
11107	419.154713000	10.0.3.152	10.0.3.153	TCP	66	50010 > 39004 [ACK] Seq=1 Ack=76 Win=28960 Len=0 TSval=22247
11108	419.166950000	10.0.3.152	10.0.3.153	TCP	67	50010 > 39004 [PSH, ACK] Seq=1 Ack=76 Win=28960 Len=1 TSval=22247
11109	419.166992000	10.0.3.153	10.0.3.152	TCP	66	39004 > 50010 [ACK] Seq=76 Ack=2 Win=29696 Len=0 TSval=22247
11110	419.167126000	10.0.3.152	10.0.3.153	TCP	67	50010 > 39004 [PSH, ACK] Seq=2 Ack=76 Win=28960 Len=1 TSval=22247
11111	419.167143000	10.0.3.153	10.0.3.152	TCP	66	39004 > 50010 [ACK] Seq=76 Ack=3 Win=29696 Len=0 TSval=22247
11112	419.167264000	10.0.3.152	10.0.3.153	TCP	67	50010 > 39004 [PSH, ACK] Seq=3 Ack=76 Win=28960 Len=1 TSval=22247
11113	419.167287000	10.0.3.153	10.0.3.152	TCP	66	39004 > 50010 [ACK] Seq=76 Ack=4 Win=29696 Len=0 TSval=22247
11114	419.180185000	10.0.3.153	10.0.3.152	TCP	13098	39004 > 50010 [ACK] Seq=76 Ack=4 Win=29696 Len=13032 TSval=22247
11115	419.180290000	10.0.3.152	10.0.3.153	TCP	66	50010 > 39004 [ACK] Seq=4 Ack=13108 Win=55024 Len=0 TSval=22247
11116	419.180308000	10.0.3.153	10.0.3.152	TCP	1221	39004 > 50010 [PSH, ACK] Seq=13108 Ack=4 Win=29696 Len=1155 TSval=22247
11117	419.186144000	10.0.3.152	10.0.3.153	TCP	74	50010 > 39004 [PSH, ACK] Seq=4 Ack=14263 Win=57920 Len=8 TSval=22247
11118	419.186528000	10.0.3.152	10.0.3.153	TCP	67	50010 > 39004 [PSH, ACK] Seq=12 Ack=14263 Win=57920 Len=1 TSval=22247
11119	419.186637000	10.0.3.153	10.0.3.152	TCP	66	39004 > 50010 [ACK] Seq=14263 Ack=13 Win=29696 Len=0 TSval=22247
11120	419.186831000	10.0.3.152	10.0.3.153	TCP	67	50010 > 39004 [PSH, ACK] Seq=13 Ack=14263 Win=57920 Len=1 TSval=22247
11121	419.187053000	10.0.3.152	10.0.3.153	TCP	67	50010 > 39004 [PSH, ACK] Seq=14 Ack=14263 Win=57920 Len=1 TSval=22247
11122	419.187163000	10.0.3.153	10.0.3.152	TCP	66	39004 > 50010 [ACK] Seq=14263 Ack=15 Win=29696 Len=0 TSval=22247
11123	419.187342000	10.0.3.152	10.0.3.153	TCP	67	50010 > 39004 [PSH, ACK] Seq=15 Ack=14263 Win=57920 Len=1 TSval=22247
11124	419.198386000	10.0.3.153	10.0.3.152	TCP	70	39004 > 50010 [PSH, ACK] Seq=14263 Ack=16 Win=29696 Len=4 TSval=22247
11125	419.204095000	10.0.3.153	10.0.3.152	TCP	66	39004 > 50010 [FIN, ACK] Seq=14267 Ack=16 Win=29696 Len=0 TSval=22247
11126	419.205357000	10.0.3.152	10.0.3.153	TCP	66	50010 > 39004 [FIN, ACK] Seq=16 Ack=14268 Win=57920 Len=0 TSval=22247
11127	419.205402000	10.0.3.153	10.0.3.152	TCP	66	39004 > 50010 [ACK] Seq=14268 Ack=17 Win=29696 Len=0 TSval=22247

▶ Frame 11114: 13098 bytes on wire (104784 bits), 13098 bytes captured (104784 bits) on interface 3

▶ Ethernet II, Src: 00:00:00_00:00:03 (00:00:00:00:00:03), Dst: 00:00:00_00:00:02 (00:00:00:00:00:02)

▶ Internet Protocol Version 4, Src: 10.0.3.153 (10.0.3.153), Dst: 10.0.3.152 (10.0.3.152)

▶ Transmission Control Protocol, Src Port: 39004 (39004), Dst Port: 50010 (50010), Seq: 76, Ack: 4, Len: 13032

▶ Data (13032 bytes)

```
0000 00 00 00 00 00 02 00 00 00 00 00 03 08 00 45 00  ....E.
0010 33 1c ed 1b 40 00 40 06 ff 8f 0a 00 03 99 0a 00  3...@.@. ....
0020 03 98 98 5c c3 5a 1b e3 88 5d 7e 18 aa 2e 80 10  ...\.Z... ]~....
0030 00 3a 4e 3f 00 00 01 01 08 0a 00 03 65 0b 00 03  ..N?....e....
```

File: /tmp/wireshark_18_interfaces_... Profile: Default

Tecnologias Utilizadas e Próximos Passos

● Tecnologia Utilizada

- Mininet com controlador POX
- Hadoop
- Containers

● Próximos Passos

- Implementar algoritmo de roteamento (Dijkstra, etc?)
- Rotas de Contingência

Referências



- [1] Guohui Wang, T. S. Eugene Ngy, Anees Shaikh: Programming Your Network at Run-time for Big Data Applications. In ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN'12)
- [2] mininet.org/overview
- [3] http://docs.hortonworks.com/HDPDocuments/HDP1/HDP-1.2.0/bk_reference/content/reference_chap2_1.html
- [4] <http://containerops.org/2013/11/19/lxc-networking>
- [5] <http://ofirm.wordpress.com/2014/01/05/creating-a-virtualized-fully-distributed-hadoop-cluster-using-linux-containers/>
- [6] <http://hadoop.apache.org/>