Análise de tráfego Hadoop em rede OpenFlow

Fabio Luiz de Paula

Rafael de Almeida

UFSCAR - Universidade Federal de São Carlos
Pós-graduação em Ciência da Computação

TÓPICOS

- Problema
- Tecnologias utilizadas
- Arquitetura
- Solução desenvolvida
- Resultados e próximos passos

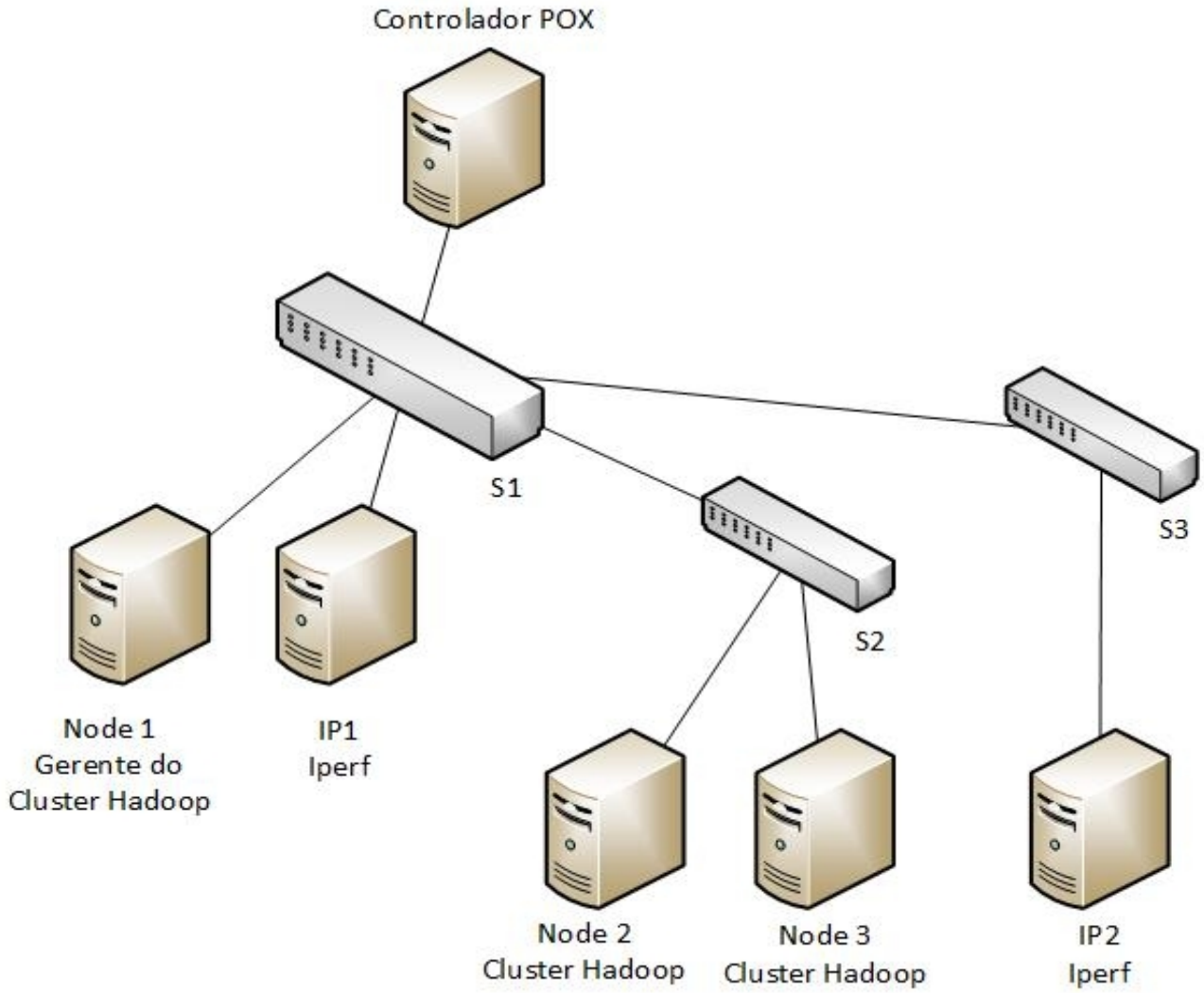
Descrição do Problema

- Como identificar um tráfego de dados Hadoop/MapReduce numa rede OpenFlow?
  - Analise do fluxo elefante ou pela porta?
- Como tratar este tráfego?

Tecnologias Utilizadas

- Mininet;
- Oracle VirtualBox;
- Hadoop CDH 4.7;
- Controlador Pox;
- Iperf.

Arquitetura



Solução desenvolvida

- Obstáculos no cenário;
- Script para criar switches no Mininet e links virtuais para conexão às VMs;
- Identificação do tráfego através das **portas** usadas pelo Hadoop;
- Uso da estrutura ***match*** no código para comparação do tráfego;
- Envio do tráfego MapReduce para uma porta especifica de um determinado switch;
- O script conta ainda com o fluxo regular, atuando como *L2-learning*.

Solução desenvolvida

```
net = Mininet(controller=RemoteController, switch=OVSKernelSwitch)
c1 = net.addController('c1', controller=RemoteController, ip="127.0.0.1", port=6633)

print "*** Creating switches"
s1 = net.addSwitch( 's1', dpid="0000000000000001")
s2 = net.addSwitch( 's2', dpid="0000000000000002")
s3 = net.addSwitch( 's3', dpid="0000000000000003")

info( '*** Add links\n')
n1 = Node( 'n1' )
n2 = Node( 'n2' )
n3 = Node( 'n3' )
ip1 = Node( 'ip1' )
ip2 = Node( 'ip2' )
net.addLink(n1, s1)
net.addLink(s2, s1)
net.addLink(ip1, s1)
net.addLink(s3, s1)
net.addLink(n2, s2)
net.addLink(n3, s2)
net.addLink(ip2, s3)
```

Cria os Switches

Cria os Links

Solução desenvolvida

```
#Identifica se o fluxo Map Reduce
if mr.tp_dst == 8020 or mr.tp_dst == 8021 or mr.tp_dst == 50010 or mr.tp_src == 8020
or mr.tp_src == 8021 or mr.tp_src == 50010:
  #Rotas  envio
  if event.dpid == 0000000000000001 and mr.nw_dst == ("10.0.0.2"):
    log.debug("flow mapred N1 -> N2/S2 %s.%i -> %s.%i" %
              (packet.src, event.port, packet.dst, port))
    msg = of.ofp_flow_mod()
    msg.match = of.ofp_match.from_packet(packet, event.port)
    msg.actions.append(of.ofp_action_output(port = 3))
    self.connection.send(msg)
    return
  if event.dpid == 0000000000000002 and mr.nw_dst == ("10.0.0.2"):
    log.debug("flow mapred N1/S2 -> N2 %s.%i -> %s.%i" %
              (packet.src, event.port, packet.dst, port))
    msg = of.ofp_flow_mod()
    msg.match = of.ofp_match.from_packet(packet, event.port)
    msg.actions.append(of.ofp_action_output(port = 1))
    self.connection.send(msg)
    return
```

Verifica se o fluxo é MapReduce

Identifica em que switch está e define a porta de saída do fluxo

Resultados e próximos passos

•Análise e identificação do tráfego MapReduce – objetivos atingidos através da solução proposta;

• Encaminhamento do tráfego – possível através da solução proposta, porém de forma estática;

• Arquitetura limitada;

• Implementar melhorias no código, otimização para cenários.