

---

---

# OrchFlow

Uma Ferramenta para Orquestração de  
Múltiplos Controladores OpenFlow

---

---

**OrchFlow**  
Manual de instalação



LERIS - Laboratory of Studies in Networks,  
Innovation and Software  
UFSCar - Sorocaba  
<http://leris.sor.ufscar.br/>

**Título:**

OrchFlow - Uma Ferramenta para Orquestração de Múltiplos Controladores OpenFlow

**Evento:**

Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)

**Fórum:**

Salão de Ferramentas do SBRC - 2016

**Grupo de trabalho:**

Universidade Federal de São Carlos (UFSCar) - Sorocaba

Instituto Federal de São Paulo (IFSP) - Boituva

**Participante(s):**

Marcelo Frate

Marcelo K. M. Marczuk

**Orientador(s):**

Fábio L. Verdi

**Cópias:** 1

**Numero de páginas:** 7

**Data de Publicação:**

04 de Abril de 2016

**Resumo:**

O principal objetivo das redes definidas por software é a centralização da lógica de controle, porém é possível dividir esta lógica entre dois ou mais controladores com o intuito de garantir a escalabilidade. O protocolo OpenFlow define a comunicação entre switches e controladores, entretanto não prevê a comunicação entre controladores, necessária para qualquer tipo de distribuição no plano de controle. Faz-se necessário, portanto, o desenvolvimento de soluções independentes do protocolo, capazes de distribuir essa lógica dentro de um mesmo domínio administrativo. Neste cenário, o OrchFlow surge como uma ferramenta capaz de orquestrar uma rede definida por software, com dois ou mais controladores OpenFlow, permitindo a gerência e o monitoramento da topologia em tempo real.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Requisitos . . . . .	1
<b>2</b>	<b>Instalação do OrchFlow</b>	<b>2</b>
<b>3</b>	<b>Implantando a rede de testes</b>	<b>4</b>
<b>A</b>	<b>Lista de arquivos</b>	<b>7</b>

# Capítulo 1

## Introdução

O OrchFlow atua como um orquestrador, um agente integrador entre as diversas aplicações disponíveis na rede e os diferentes controladores OpenFlow, sob um mesmo controle administrativo, definido aqui como domínio administrativo. O OrchFlow possibilita a comunicação entre as diferentes aplicações através de uma interface Norte, capaz de receber solicitações e invocar serviços pré-determinados. Essas interfaces fazem parte de um único sistema WEB, utilizando *Representational State Transfer* (REST), um protocolo que torna possível a troca de informações entre aplicativos e serviços web, pela qual são solicitados todos os recursos necessários para o estabelecimento de serviços fim a fim. Ao receber tais solicitações, o OrchFlow as processa e de forma orquestrada atua sobre cada um dos controladores conforme o subdomínio a ser alcançado.

### 1.1 Requisitos

- Apache Tomcat8;
- Banco de Dados Neo4j V2.3;
- Controlador Floodlight;
- javac 1.8.0\_72;
- Mininet V 2.2.1;
- Módulo ARPReply;
- Módulo Reactive;
- OpenFlow V 1.3;
- Oracle VirtualBox
- ovs-vsctl (Open vSwitch) 2.0.2;
- Sistema Operacional Linux;

## Capítulo 2

# Instalação do OrchFlow

Utiliza-se aqui uma máquina virtual (VM):

Para baixar e instalar o Oracle VirtualBox acesse:

<https://www.virtualbox.org/>

Crie uma VM e instale o Ubuntu Server 14.04.4 LTS.

Para baixar e instalar o Ubuntu acesse:

<http://www.ubuntu.com/download/server>

Crie o usuário "orchflow" e senha "OrchFlow2016"

Configure a placa de rede eth0 com IP: 192.168.56.1/24

Instale o banco de dados Neo4J

Para baixar e instalar o Neo4J acesse:

<http://neo4j.com/download/>

Instale o Apache Tomcat 8.0.33 Released

Para baixar e instalar o Tomcat8 acesse:

<http://tomcat.apache.org/>

Após a instalação, o usuário precisará configurar o Neo4j e o servidor de aplicação Apache Tomcat para o correto funcionamento do OrchFlow. Para tal, siga os seguintes passos:

Após a instalação do Neo4j você precisou configurar a senha de acesso ao banco de dados, porém para evitar maiores dificuldades de configurações, vamos desabilitar a necessidade de uso de senhas para acesso ao Neo4j e modificar alguns limites.

Editar o arquivo: `/var/lib/neo4j/conf/neo4j-server.properties`  
e altere a seguinte linha:

```
dbms.security.auth_enabled=true
```

para

```
dbms.security.auth_enabled=false
```

Editar o arquivo: `/etc/security/limits.conf`

e incluir as seguintes linhas:

```
neo4j soft nofile 40000
```

```
neo4j hard nofile 40000
```

Editar o arquivo: /etc/pam.d/su

e descomentar, retirar o caractere "#" da seguinte linha:

```
#session required pam_limits.so
```

Executar o seguinte comando:

```
#ulimit -n 40000
```

O arquivo de instalação do OrchFlow é disponibilizado em formato .war (Web application ARchive) que pode ser facilmente instalado. O procedimento para implantar a aplicação no servidor Apache Tomcat pode ser visto na Figura 2.1:

Acesse a página do servidor Tomcat através de seu navegador.

<http://192.168.56.1:8080/manager/html>

Manager App > Deploy > WAR File to deploy

Selecione o arquivo orchflow.war e clique em deploy.

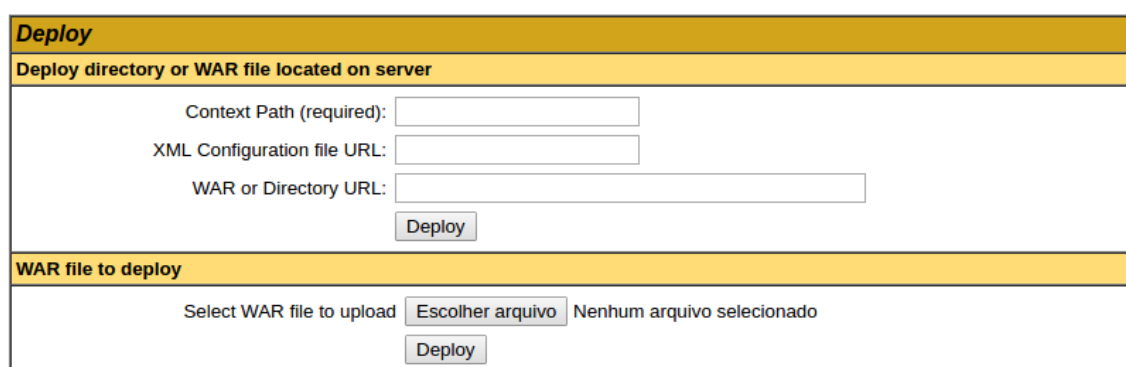


Figura 2.1: Tomcat-deploy.

Após o deploy, será possível administrar a aplicação OrchFlow no servidor Tomcat conforme a Figura 2.2.

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/OrchFlow	None specified	OrchFlow	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	None specified	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	None specified	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Figura 2.2: Tomcat-applications.

## Capítulo 3

# Implantando a rede de testes

**VM1:** Nesta VM, será instalado o Mininet, com o qual o usuário deverá emular a sua rede. Será possível também executar uma rede emulada através de um script preparado com três subdomínios interligados em forma de anel, com 7 switches cada em forma de árvore e 4 hosts, conforme a Figura 3.1, que ilustra a topologia a ser utilizada nos testes do OrchFlow.

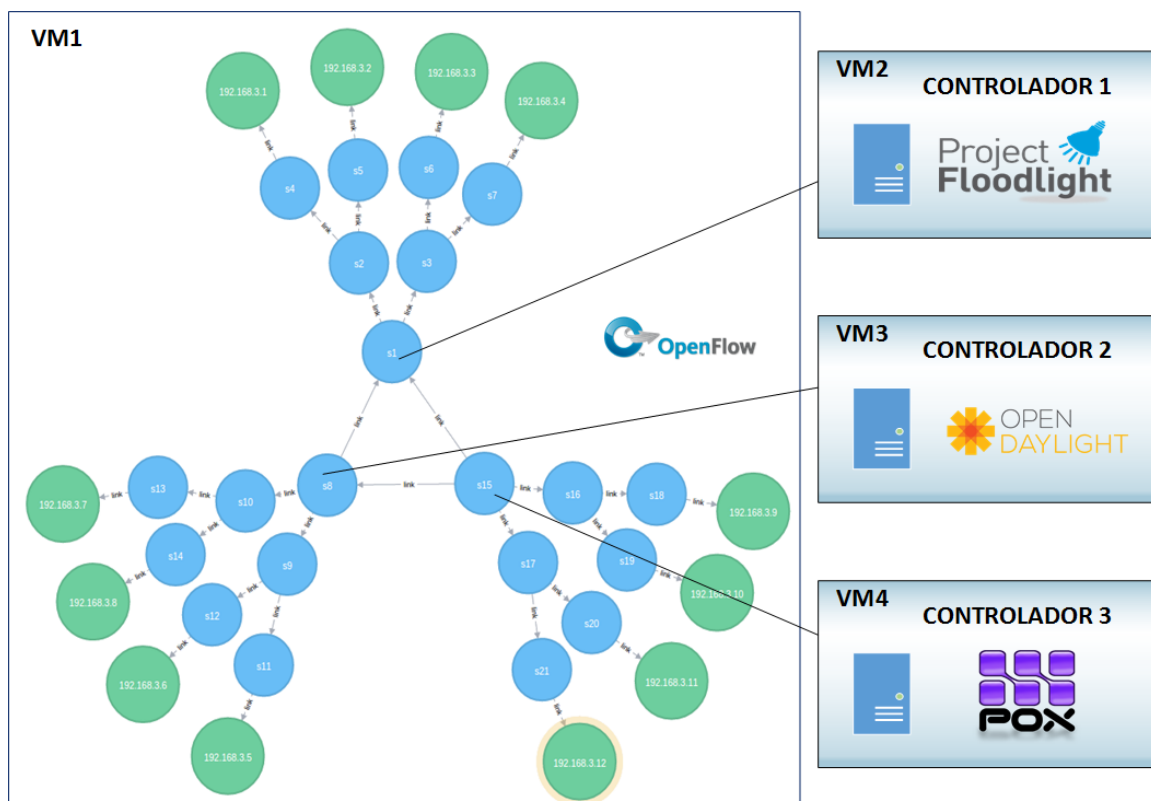


Figura 3.1: Topologia de rede.

Crie uma VM e instale o Ubuntu Server 14.04.4 LTS.

Para baixar e instalar o Ubuntu acesse:  
<http://www.ubuntu.com/download/server>

Crie o usuário "mininet" e senha "mininet"

Configure a placa de rede eth0 com IP: 192.168.56.100/24

Configure a placa de rede eth1 com IP: 10.0.0.100/24

verifique o acesso ao servidor OrchFlow:

```
mininet@mininet:~$ ping 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.
64 bytes from 192.168.56.1: icmp_seq=1 ttl=64 time=0.314 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=64 time=0.248 ms
```

Baixe o arquivo topologia.py e cole na pasta:

```
\home\mininet\mininet\custom
```

Verifique a permissão do arquivo para que possa ser executado:

```
chmod 777 \home\mininet\mininet\custom\topologia.py
```

**VM2, VM3 e VM4:** Em cada VM, um controlador Floodlight será instalado e configurado para receber as chamadas dos switches das redes instaladas na VM1.

Crie três VMs e instale o Ubuntu Server 14.04.4 LTS.

Para baixar e instalar o Ubuntu acesse:

<http://www.ubuntu.com/download/server>

Crie o usuário "mininet" e senha "mininet"

Para o controlador 1:

Configure a placa de rede eth0 com IP: 192.168.56.101/24

Configure a placa de rede eth1 com IP: 10.0.0.101/24

Para o controlador 2:

Configure a placa de rede eth0 com IP: 192.168.56.102/24

Configure a placa de rede eth1 com IP: 10.0.0.102/24

Para o controlador 3:

Configure a placa de rede eth0 com IP: 192.168.56.103/24

Configure a placa de rede eth1 com IP: 10.0.0.103/24

verifique o acesso ao servidor OrchFlow em cada controlador:

```
mininet@mininet:~$ ping 192.168.56.1
PING 192.168.56.1 (192.168.56.1) 56(84) bytes of data.
64 bytes from 192.168.56.1: icmp_seq=1 ttl=64 time=0.314 ms
64 bytes from 192.168.56.1: icmp_seq=2 ttl=64 time=0.248 ms
```

verifique o acesso ao mininet em cada controlador:

```
mininet@mininet:~$ ping 192.168.56.100
PING 192.168.56.100 (192.168.56.100) 56(84) bytes of data.
64 bytes from 192.168.56.100: icmp_seq=1 ttl=64 time=0.314 ms
64 bytes from 192.168.56.100: icmp_seq=2 ttl=64 time=0.248 ms
```



A rede 192.168.56.0 será utilizada para o acesso aos serviços de gerenciamento e uso do OrchFlow, por onde toda a configuração dos controladores serão realizadas.

A rede 10.0.0.0 será utilizada pelo protocolo OpenFlow, é a camada de controle da rede.

Para fins de demonstração neste Salão, estamos utilizando apenas controladores Floodlight em cada uma das três VMs.

Para baixar e instalar o Floodlight acesse:

<https://floodlight.atlassian.net/wiki/display/floodlightcontroller/Installation+Guide>\*

Antes de executar o controlador é preciso instalar e configurar os módulos ARPReply e Reactive.

Para descobrir como criar um módulo para o Floodlight acesse:

<https://floodlight.atlassian.net/wiki/display/floodlightcontroller/How+to+Write+a+Module>\*

Baixe os arquivos:

ARPReply.zip

Reactive.zip

Descompacte e cole na pasta:

/home/mininet/floodlight/src/main/java/net/floodlightcontroller

Você verá as duas pastas arpreply e reactive.

Em seguida, baixe o arquivo:

net.floodlightcontroller.core.module.IFloodlightModule

E cole na pasta:

/home/mininet/floodlight/src/main/resources/META-INF

Depois, baixe o arquivo:

floodlightdefault.properties

E cole na pasta:

/home/mininet/floodlight/src/main/resources

E recompile o controlador:

```
root@controlador1:/home/mininet/floodlight# ant;
```

Agora é só rodar o controlador:

```
root@controlador1:/home/orchflow# java -jar /home/orchflow/floodlight/target/floodlight.jar  
-cf /home/orchflow/floodlight/src/main/resources/floodlightdefault.properties
```

# Apêndice A

## Lista de arquivos

O manual do usuário assim como o código fonte dos módulos necessários para Floodlight, os arquivos de configuração e o script da topologia utilizada estão disponíveis em:

<https://github.com/marcelofrate/OrchFlow/>

- ARPReply.zip
- floodlightdefault.properties
- net.floodlightcontroller.core.module.IFloodlightModule
- Reactive.zip
- topologia.py