Innovative R&D by NTT

Lagopus handson

NTT Network Innovation lab

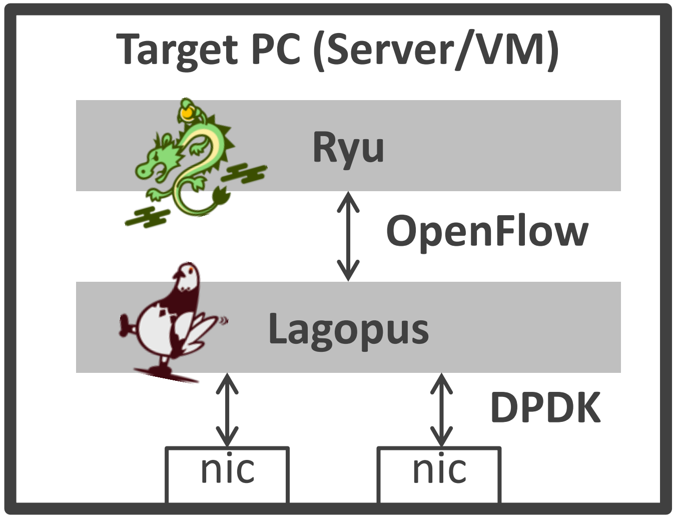
version 0.4

Last Update : 12/14/2015

Today's Goal

Step1

- Install and set up OpenFlow switch (Lagopus) and OpenFlow controller (Ryu) in your virtual machine
- Run sample application



Step2

- Run lagopus on mininet

#mininet: SDN-based network emulation tool

Preparation for Installing Lagopus and Ryu

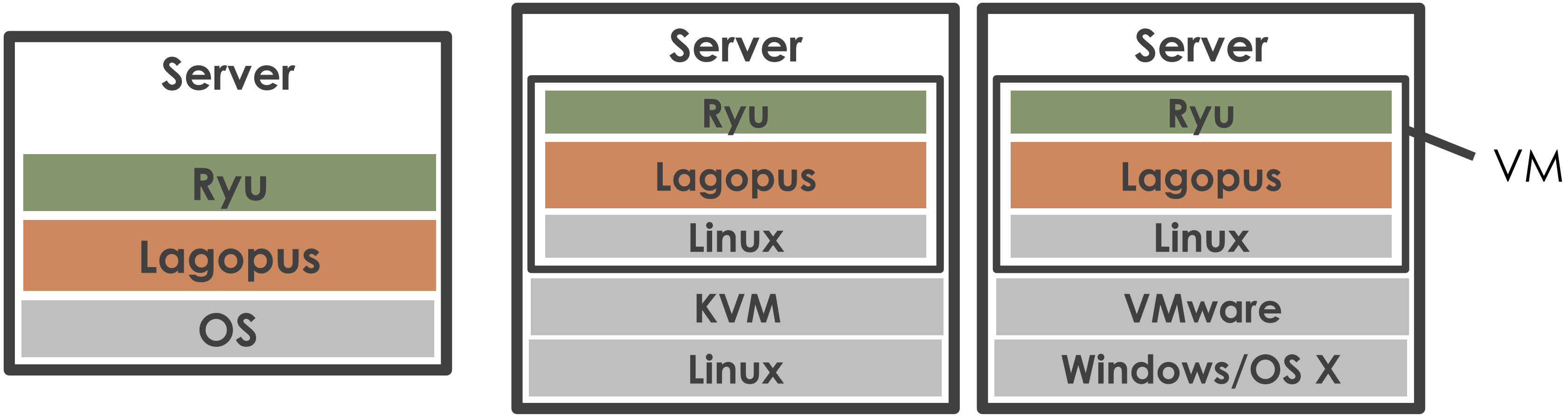
Which do you choice as a target system ?

1. **Bare metal system (PC or server)**

   You must choice the DPDK available hardware

2. **Virtual machine**

   You must choice the DPDK available virtual hardware



1. Bare metal

2. Virtual machine

1. Bare metal system requirements

CPU

- **Type**

  - It is better that 64-bit instruction is available.

  - The latest Intel CPUs are better

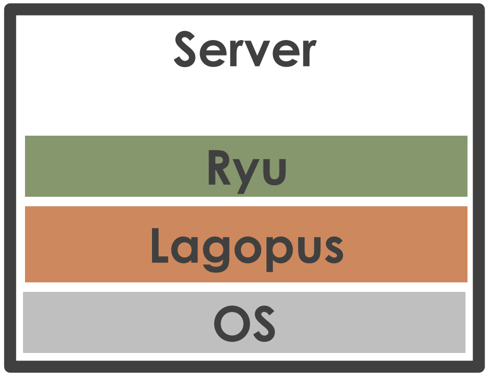
    #A little old cpu may be supported because DPDK is available from 2009 year(#1).

  - Code Name

    Broadwell/Haswell/Ivy Bridge/Sandy Bridge/Nehalem

- **Cores**

  >= 2 CPU cores



Memory

- **Size**

  >= 1Gbytes

#1: http://www.intel.com/content/dam/www/public/us/en/documents/presentation/dpdk-packet-processing-ia-overview-presentation.pdf

Notice: We can not guarantee the complete combination of hardwares.

1. Bare metal system requirements

Network Interface Card (NIC)

- **Supported network drivers in DPDK**
  - **e1000**

    82540, 82545, 82546
  - **e1000e**

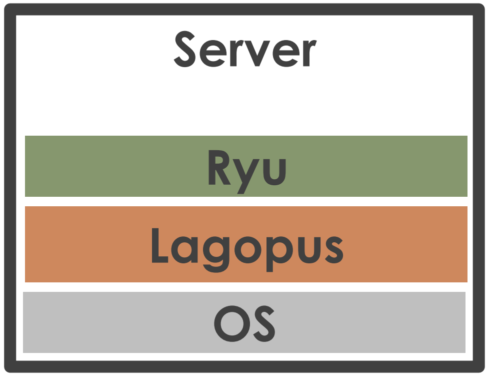
    82571-82574, 82583, ICH8-ICH10, PCH-PCH2
  - **Igb**

    82575..82576, 82580, I210, I211, I350, I354, DH89xx
  - **Ixgbe**

    82598..82599, X540, X550
  - **I40e**

    X710, XL710
  - **Fm10k**
- **The number of NICs**

  >= 3 (2 NICs for Lagopus, 1 NIC for management)



Server

Ryu

Lagopus

OS

Notice: We can not guarantee the complete combination of hardwares.

1. Bare metal system requirements

Operating system

- **Supported OS in DPDK**

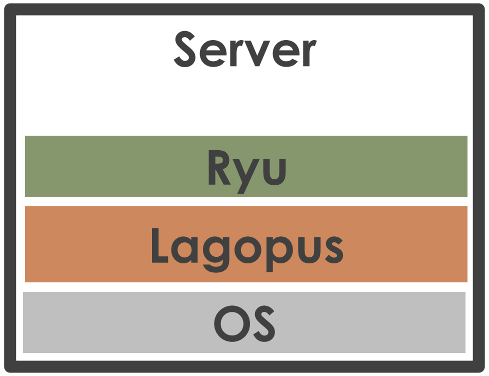
  - **Linux**

    #Operation checked OS

    - **Ubuntu 14.04 LTS/Ubuntu 12/04 LTS**
    - **Cent OS 6.5**

  - **Free BSD**

    >= lagopus version 0.2



Notice: We can not guarantee the complete combination of hardwares.

2. Virtual machine requirements

vCPU

- **Type**

  Some CPU flags may be set on booting VM.
  (The next page shows the instruction to specify such CPU flags)

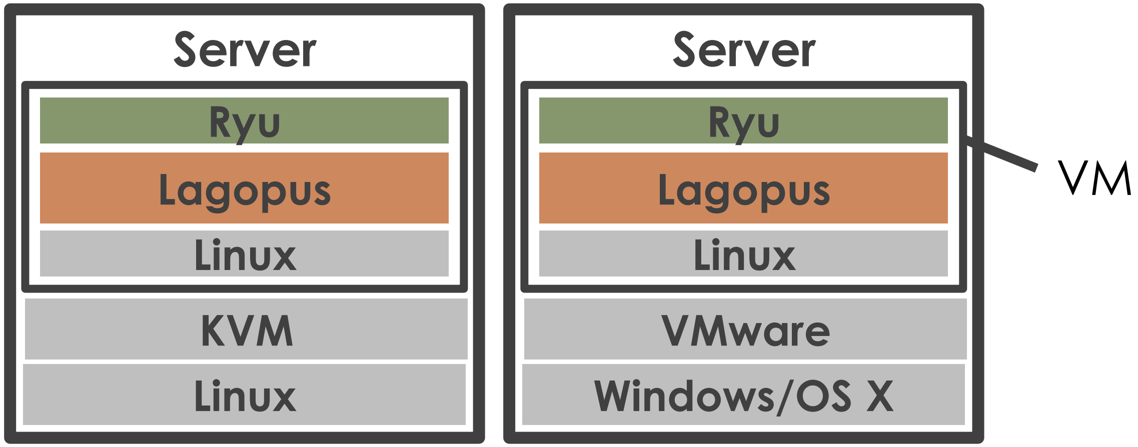
- **Cores**

  >= 2 cores

vMemory

- **Size**

  >= 1Gbytes



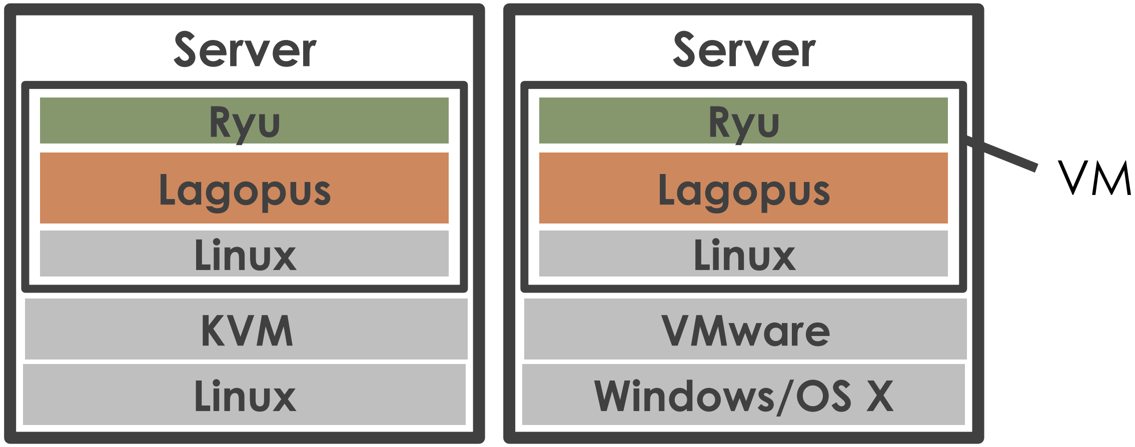
Notice: We can not guarantee the complete combination of hardwares.

2. Virtual machine requirements

vNIC

e1000 device

#KVM/Vmware supports e1000 virtual device



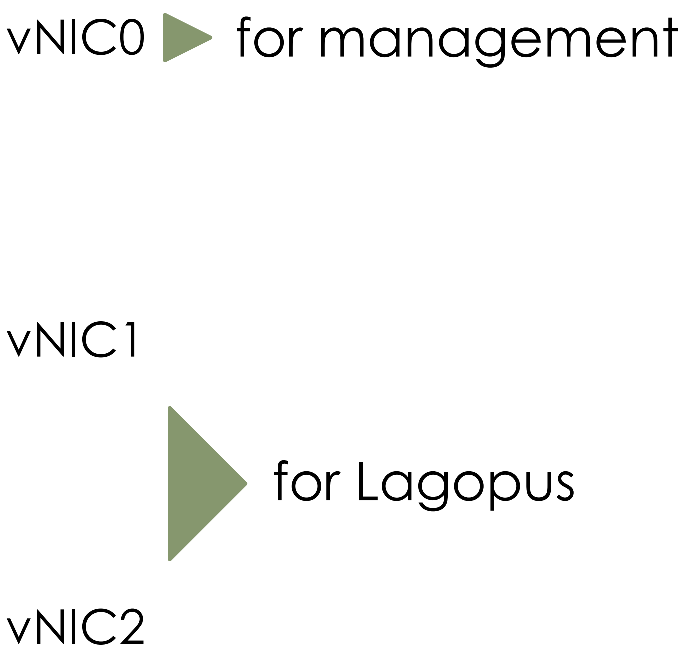
2. Virtual machine requirements

vNIC

Sample Vmware Fusion configuration on OS X.

virtualDev=“e1000”

```
ethernet0.present = "TRUE"
ethernet0.connectionType = "bridged"
ethernet0.virtualDev = "e1000"
ethernet0.wakeOnPcktRcv = "FALSE"
ethernet0.addressType = "generated"
ethernet0.linkStatePropagation.enable = "TRUE"
ethernet0.generatedAddress = "00:0c:29:67:06:91"
ethernet0.pciSlotNumber = "33"
ethernet0.generatedAddressOffset = "0"
ethernet1.present = "TRUE"
ethernet1.connectionType = "hostonly"
ethernet1.virtualDev = "e1000"
ethernet1.wakeOnPcktRcv = "FALSE"
ethernet1.addressType = "generated"
ethernet1.linkStatePropagation.enable = "TRUE"
ethernet1.generatedAddress = "00:0c:29:67:06:9b"
ethernet1.generatedAddressOffset = "10"
ethernet1.pciSlotNumber = "37"
ethernet2.present = "TRUE"
ethernet2.connectionType = "hostonly"
ethernet2.virtualDev = "e1000"
ethernet2.wakeOnPcktRcv = "FALSE"
ethernet2.addressType = "generated"
ethernet2.linkStatePropagation.enable = "TRUE"
ethernet2.generatedAddress = "00:0c:29:67:06:a5"
ethernet2.generatedAddressOffset = "20"
ethernet2.pciSlotNumber = "38”
```



vNIC0 for management

vNIC1

for Lagopus

vNIC2

2. Virtual machine requirements

OS

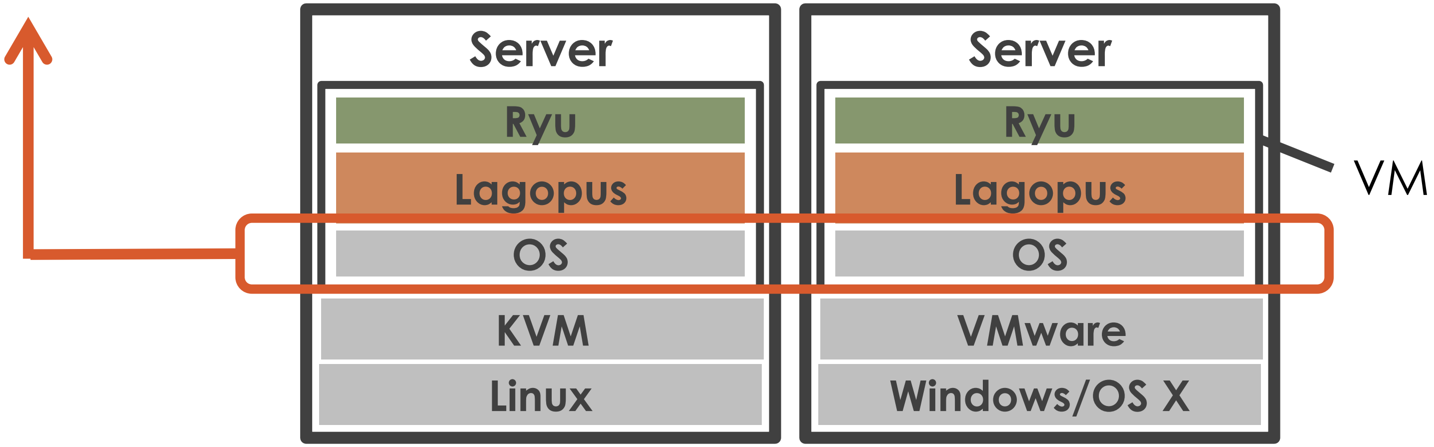
- **Supported OS in DPDK**

  - Linux

  #Operation checked OS

  - Ubuntu 14.04 LTS/Ubuntu 12/04 LTS

  - Cent OS 6.5



Handson Start!

1: Install Lagopus

2: Install Ryu

3: Run Lagopus with Ryu

Handson Start!

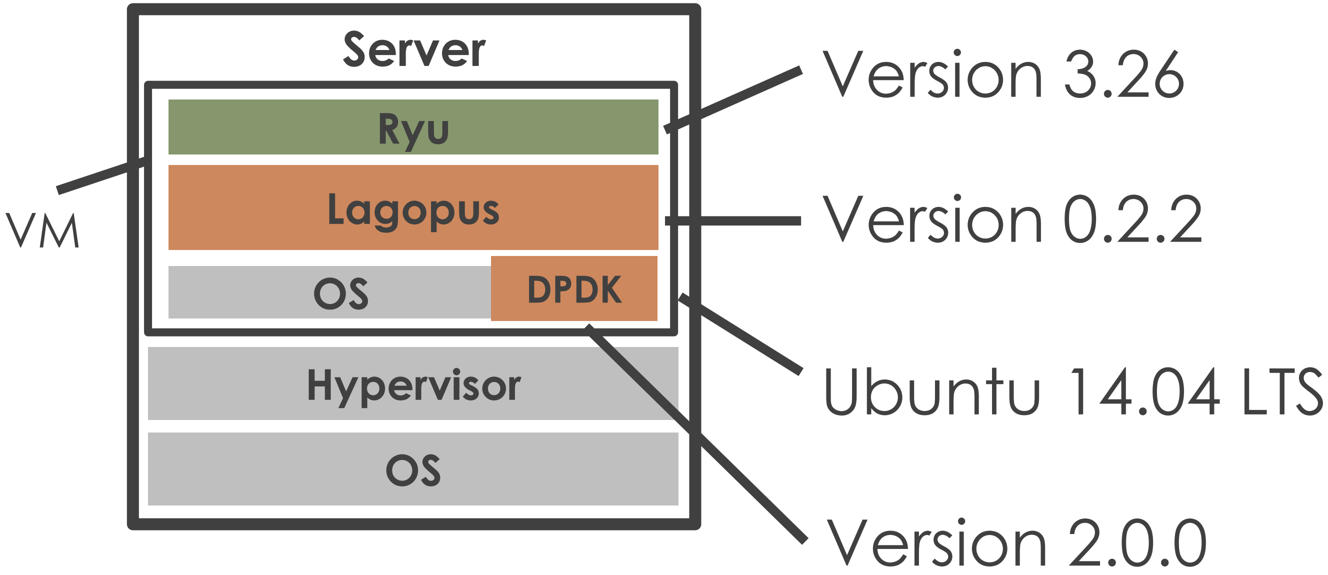
1: Install Lagopus

2: Install Ryu

3: Run Lagopus with Ryu

1: Install Lagopus

Environment



1: Install Lagopus

Download lagopus and Intel DPDK

- **Download “Lagopus”**

```
$sudo apt-get git

$git clone https://github.com/lagopus/lagopus
```

You must move to your home directory

- **Download “Intel DPDK”**

```
$wget http://dpdk.org/browse/dpdk/snapshot/dpdk-
2.0.0.zip

$unzip dpdk-2.0.0.zip
```

Need unzip(sudo apt-get install unzip)

1: Install Lagopus

Setup essential software packages

```
$ sudo apt-get update

$ sudo apt-get install make coreutils gcc Binutils
build-essential libgmp-dev libssl-dev libpcap-dev
byacc Flex python-dev python-setuptools python-pip
ethtool
```

1: Install Lagopus

Setup essential software packages

1. make
2. coreutils
3. gcc
4. Binutils
5. build-essential
6. libgmp-dev
7. libssl-dev
8. libpcap-dev
9. byacc
10. Flex
11. python-dev
12. python-setuptools
13. python-pip
14. ethtool

For DPDK version

1: Install Lagopus

Setup intel DPDK

- **Compile DPDK libraries**

```
$ cd ~/dpdk-2.0.0
$ cd tools
$ ./setup.sh
```

```
[9] x86_64-native-linuxapp-gcc
..............................................................................................
            :
[30] Exit Script
Option:9
```

1: Install Lagopus

For DPDK version

Setup intel DPDK

●Loading kernel module

```
[12] Insert IGB UIO module
..........................................................................
           :
[30] Exit Script
Option:12
```

```
[14] Insert KNI module
..........................................................................
           :
[30] Exit Script
Option:14
```

1: Install Lagopus

For DPDK version

Setup intel DPDK

- **Setup hugepages**

```
[15] Setup hugepage mappings for non-NUMA systems
..............................................................................
            :
[30] Exit Script
Option:15
```

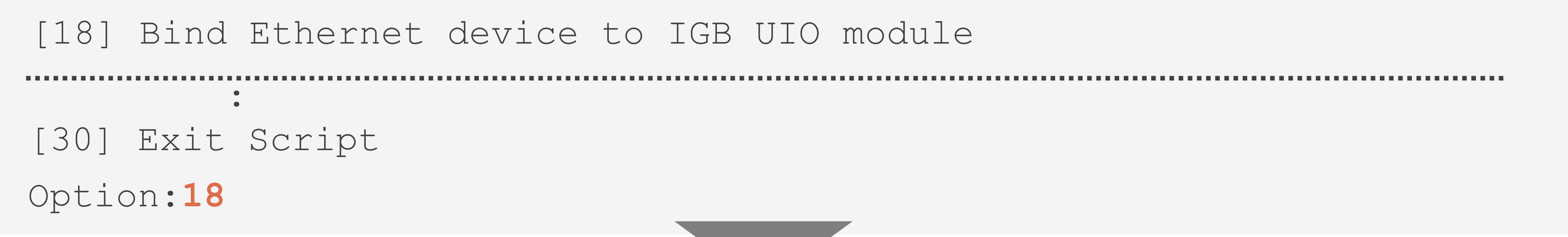
```
Input the number of 2MB pages
 Example: to have 128MB of hugepages available, enter '64' to
 reserve 64 * 2MB pages

Number of pages:256
```

For DPDK version

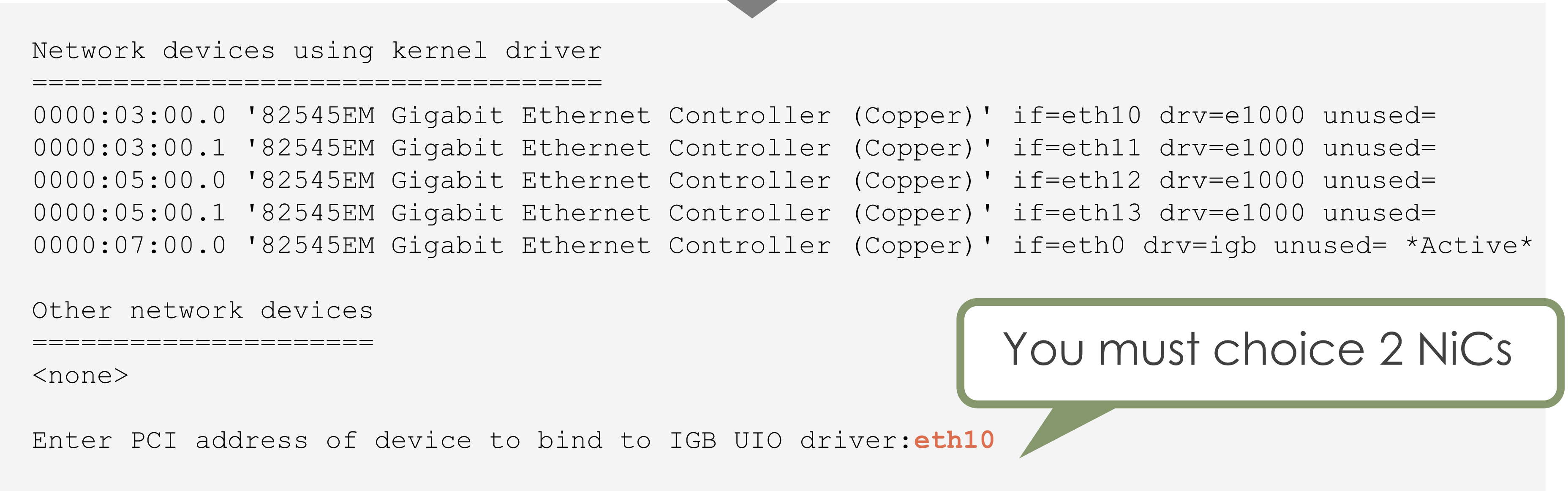
1: Install Lagopus

Setup intel DPDK

- **Bind NiCs to DPDK**



```
[18] Bind Ethernet device to IGB UIO module
.................................................................................
            :
[30] Exit Script
Option:18
```



```
Network devices using kernel driver
===================================
0000:03:00.0 '82545EM Gigabit Ethernet Controller (Copper)' if=eth10 drv=e1000 unused=
0000:03:00.1 '82545EM Gigabit Ethernet Controller (Copper)' if=eth11 drv=e1000 unused=
0000:05:00.0 '82545EM Gigabit Ethernet Controller (Copper)' if=eth12 drv=e1000 unused=
0000:05:00.1 '82545EM Gigabit Ethernet Controller (Copper)' if=eth13 drv=e1000 unused=
0000:07:00.0 '82545EM Gigabit Ethernet Controller (Copper)' if=eth0 drv=igb unused= *Active*

Other network devices
=====================
<none>

Enter PCI address of device to bind to IGB UIO driver:eth10
```

You must choice 2 NiCs

1: Install Lagopus

For DPDK version

Compile

```
$ cd lagopus
$ ./configure --with-dpdk-dir=${HOME}dpdk-2.0.0
$ make
```

Install

```
$ sudo make install
```

1: Install Lagopus

Compile

```
$ cd lagopus
$ ./configure
$ make
```

Install

```
$ sudo make install
```

Handson Start!

1: Install Lagopus

2: Install Ryu

3: Run Lagopus with Ryu

2: Install Ryu

```
$ sudo pip install ryu

$ sudo pip install six --upgrade
```

Handson Start!

1: Install Lagopus

2: Install Ryu

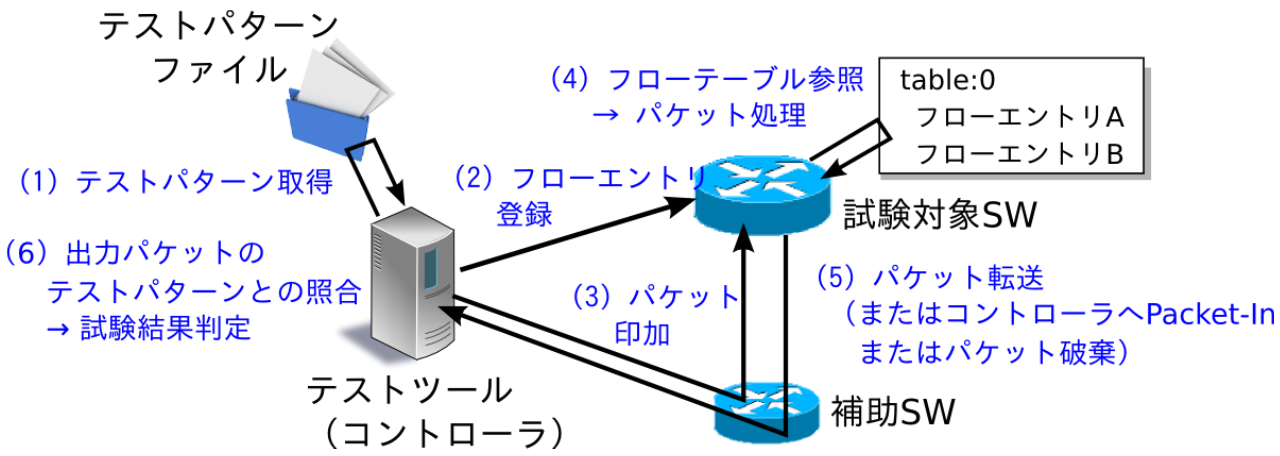
3: Run Lagopus with Ryu

3: Run lagopus with Ryu

```
$cd /usr/local/lib/python2.7/dist-
packages/ryu/tests/switch
$ ryu-manager --test-switch-dir of13 tester.py
```

- **Ryu Certification**

  OpenFlowスイッチがどのくらいOpenFlowの仕様に準拠しているかテストするためのツール

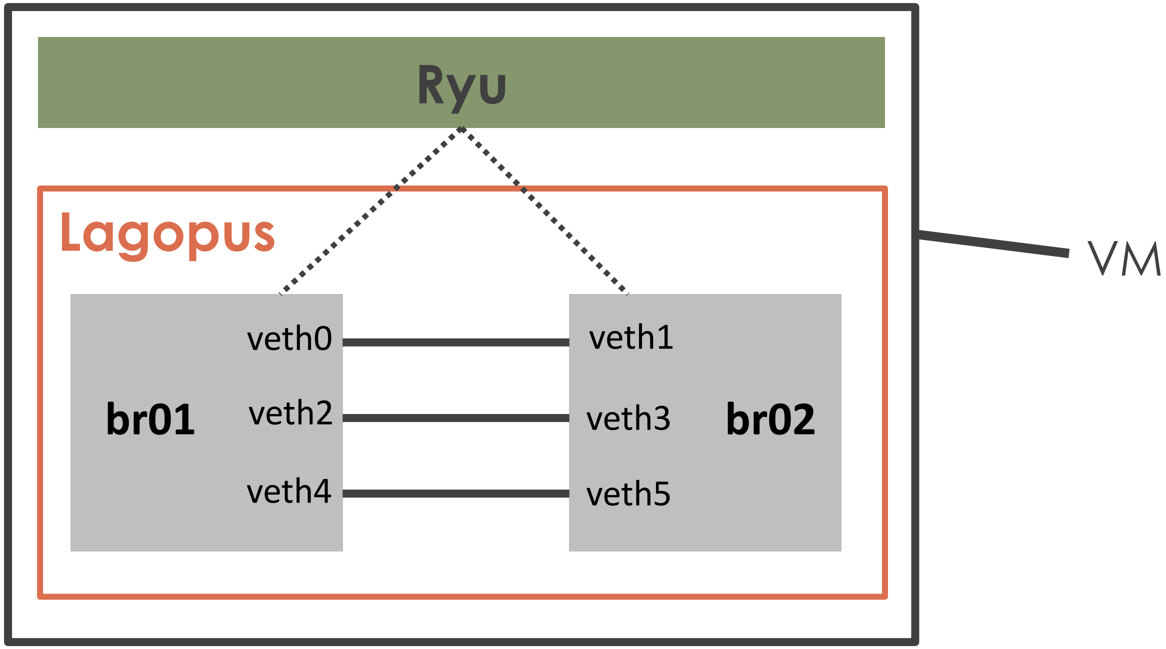


http://osrg.github.io/ryu-book/ja/html/

3: Run lagopus with Ryu

Setup Lagopus configuration file

- Created topology



3: Run lagopus with Ryu

Setup network namespace

```
$ git clone https://github.com/lagopus/handson
$ cd handson
$ ./setup_rawsocket.sh
```

```
$sudo ip link add veth0 type veth peer name veth1
$sudo ip link add veth2 type veth peer name veth3
$sudo ip link add veth4 type veth peer name veth5
$sudo ip link set veth0 up
$sudo ip link set veth1 up
$sudo ip link set veth2 up
$sudo ip link set veth3 up
$sudo ip link set veth4 up
$sudo ip link set veth5 up
```

3: Run lagopus with Ryu

Setup Lagopus configuration file

```
$ cd /usr/local/etc/
$ mkdir lagopus
$ sudo cp ~lagopus/misc/examples/lagopus.dsl /usr/
  local/etc/lagopus/
$ sudo vi/usr/local/etc/lagopus/lagopus.dsl
```

```
$ mkdir /usr/local/etc/lagopus
$ git clone https://github.com/lagopus/handson
$ cd handson
$ cp lagopus.dsl /usr/local/etc/lagopus/
```

For DPDK version

3: Run lagopus with Ryu

Setup Lagopus configuration file

- **Lagopus.dsl**

Network namespaceを活用

```
channel channel01 create -dst-addr 127.0.0.1 -protocol tcp
channel channel02 create -dst-addr 127.0.0.1 -protocol tcp

controller controller01 create -channel channel01 -role equal -connection-type main
controller controller02 create -channel channel02 -role equal -connection-type main

interface interface01 create -type ethernet-dpdk-phy -port-number0
interface interface02 create -type ethernet-dpdk-phy -port-number1
interface interface03 create -type ethernet-dpdk-phy -port-number2
interface interface04 create -type ethernet-dpdk-phy -port-number3
interface interface05 create -type ethernet-dpdk-phy -port-number4
interface interface06 create -type ethernet-dpdk-phy -port-number5

port port01 create -interface interface01
port port02 create -interface interface02
port port03 create -interface interface03
port port04 create -interface interface04
port port05 create -interface interface05
port port06 create -interface interface06

bridge bridge01 create -controller controller01 -port port01 1 -port port03 2 -port port05 3 -dpid

0x1
bridge bridge02 create -controller controller02 -port port02 1 -port port04 2 -port port06 3 -dpid
0x2

bridge bridge01 enable
bridge bridge02 enable
```

3: Run lagopus with Ryu

Setup Lagopus configuration file

- **Lagopus.dsl**

Network namespaceを活用

```
channel channel01 create -dst-addr 127.0.0.1 -protocol tcp
channel channel02 create -dst-addr 127.0.0.1 -protocol tcp
controller controller01 create -channel channel01 -role equal -connection-type main
controller controller02 create -channel channel02 -role equal -connection-type main
interface interface01 create -type ethernet-rawsock -device veth0
interface interface02 create -type ethernet-rawsock -device veth1
interface interface03 create -type ethernet-rawsock -device veth2
interface interface04 create -type ethernet-rawsock -device veth3
interface interface05 create -type ethernet-rawsock -device veth4
interface interface06 create -type ethernet-rawsock -device veth5
port port01 create -interface interface01
port port02 create -interface interface02
port port03 create -interface interface03
port port04 create -interface interface04
port port05 create -interface interface05
port port06 create -interface interface06
bridge bridge01 create -controller controller01 -port port01 1 -port port03 2 -port port05 3 -dpid
0x1
bridge bridge02 create -controller controller02 -port port02 1 -port port04 2 -port port06 3 -dpid
0x2
bridge bridge01 enable
bridge bridge02 enable
```

3: Run lagopus with Ryu

For DPDK version

Run Lagopus

```
$ sudo lagopus -d -- -c3 -n1 --vdev eth_pipe0 --vdev
eth_pipe1 --vdev eth_pipe2 -- -p3f --core-assign balance
```

- **Options**

-d:	Debug mode (foreground)
-c *bitmask*:	Which CPU cores to use
-n *channels*:	Memory channels
-p *bitmask*:	Number of port
-l filename:	Specify a log file path (default:syslog)

3: Run lagopus with Ryu

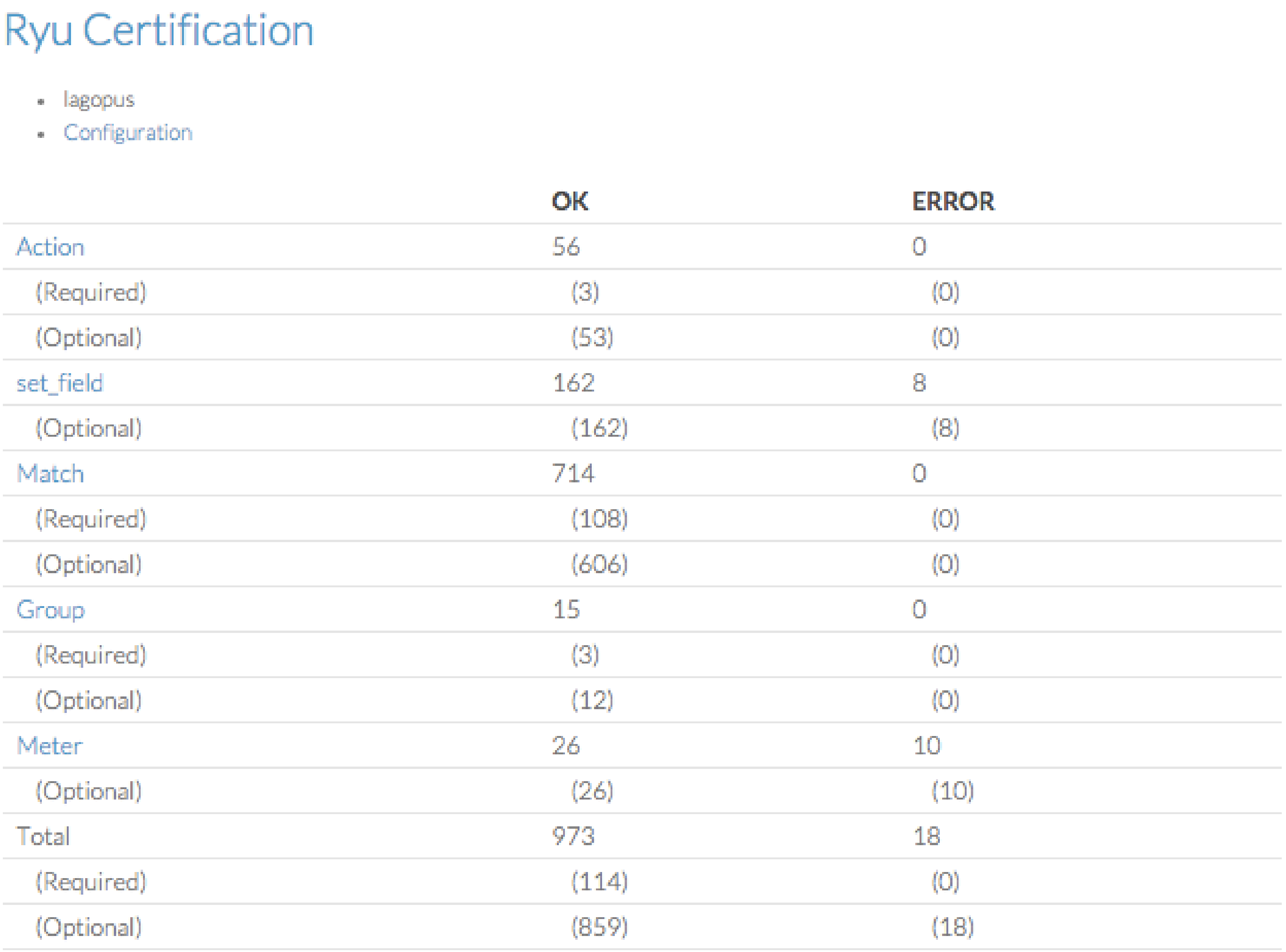
Run Lagopus

```
$ sudo lagopus -d
```

- **Options**

  -d   : Debug mode (foreground)

  -l filename: Specify a log file path (default:syslog)



Ryu Certification

- lagopus
- Configuration

	OK	ERROR
Action	56	0
(Required)	(3)	(0)
(Optional)	(53)	(0)
set_field	162	8
(Optional)	(162)	(8)
Match	714	0
(Required)	(108)	(0)
(Optional)	(606)	(0)
Group	15	0
(Required)	(3)	(0)
(Optional)	(12)	(0)
Meter	26	10
(Optional)	(26)	(10)
Total	973	18
(Required)	(114)	(0)
(Optional)	(859)	(18)

https://osrg.github.io/ryu-certification/switch/lagopus

Further information

- より高速にお使いいただくために

  gcc4.X, ixgbeを利用する場合

```
$ sudo vi dpdk-2.0.0/config/common_linuxapp
```

```
CONFIG_RTE_LIBRTE_IXGBE_PF_DISABLE_STRIP_CRC=n
CONFIG_RTE_LIBRTE_IXGBE_RX_ALLOW_BULK_ALLOC=y
CONFIG_RTE_IXGBE_INC_VECTOR=n
CONFIG_RTE_IXGBE_RX_OLFLAGS_ENABLE=y
```

Further information

- Lagopusの状態を確認する

```
$ lagosh
```

- 各種情報の表示

```
$lagosh> Show flow         #フロー/フローの統計情報の表示
$lagosh> Show bridge       #ブリッジの情報/ブリッジの統計情報の表示
$lagosh> Show controller                   ⋮
$lagosh> Show meter
$lagosh> Show group
```

- その他

```
$ lagosh> Configure
Configure# edit            #.dslの内容を書き換えたり..
```

```
$ lagosh> stop             #Lagopusを停止します
```

Appendix: Install by Ansible

- Setup

```
$ sudo apt-get install git ansible
$ git clone https://github.com/lagopus/lagopus-tools
```

```
$ cd lagopus-tools/ansible
$ echo “[target]” > inventories/hosts_setup
$ echo “127.0.0.1” >> inventories/hosts_setup
$ echo “work_dir: /home/lagopus” > playbooks/vars.yml
$ echo “user: lagopus” >> playbooks/vars.yml
$ ansible-playbook -i inventories/hosts_setup
playbooks/setup.yml -K --connection local
```

Appendix: mininet

- ## Setup

```
$ git clone https://github.com/lagopus/mininet
$ cd mininet
$ git checkout lagopus
$ ./util/install.sh -n
```

- ## Sample

```
$ ryu-manager /usr/local/lib/python2.7/dist-packages/ryu/app/simple_switch_13.py
```

```
$ sudo ~/mininet/examples/simplelagopus.py
```

Thank you for your attention

This research is a part of the project for “Research and Development of Network Virtualization Technology” supported by the Ministry of Internal Affairs and Communications.