# Report for week2

1. **Original general idea of the Architecture**

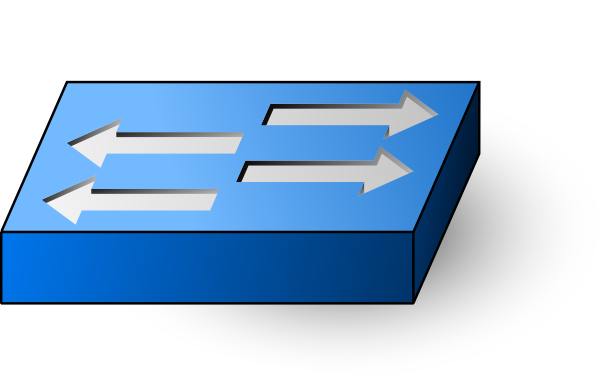
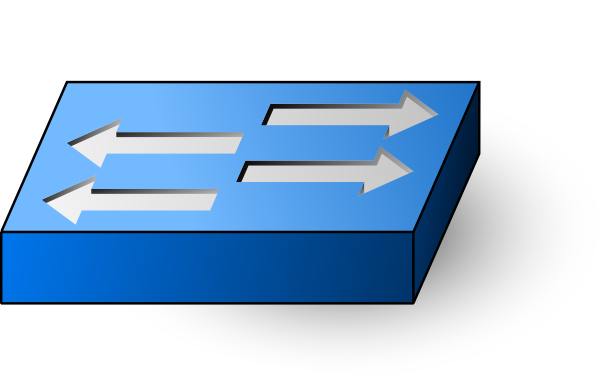
Hadoop spark

REST

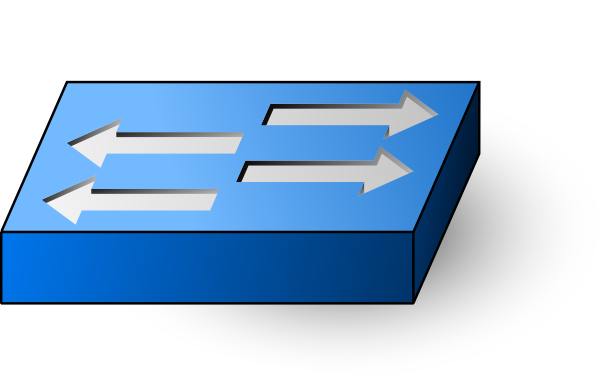
tT

**Migration Brain**

**SDN Flooglight controller**

**Mininet or Physical device**



**Migration Brain need to answer 3 questions:**

1. where to migrate?
2. Which LVM should use migration?
3. When should we migrate?

In fact, it is the application which use SDN northbound API to collecting the statistics of the network, then based on the statistics make some decision.

**Floodlight controller**

SDN as the network background, and take the advantage of SDN,capture some useful information,then use the statistic to do the migration decision.

**Switch device**

Can use mininet deploy the virtual switch or use the physical switch

**Host**

The host connected the switch will run docker container.

**Migration**

It could be node migration,link migration,still not decided

1. **[floodlight] how to process the packet-in msg floodlight**

1)**init** initialize the data structure user defined,and load the dependencies

2)**getModuleDependencies** wire it up to module loading system

3)**startup** implement the basic listener,capture packet-in messages

4) **getName**  put an ID for OFMessage listener

5)**receive** behavior for packet\_in message

1. **how to create a module**

1)getServices

2)getDependencies

3)init

4)startup

C

1. **openflow**

S1

H2

H1

10.0.0.1 10.0.0.2

OpenFlow message exchange between controller and switch,there are two message types,message in and message out.

H1 ping h2

10.0.0.1 -> 10.0.0.2 packet\_in ICMP message from h1 to h2,switch send packet\_in msg to controller

127.0.0.1->127.0.0.1 packet-out controller send package out to switch to tell which is the path

10.0.0.2 -> 10.0.0.1 packet\_in host2 reply,switch ask controller for the path to go to host1

127.0.0.1 ->127.0.0.1 packet\_add controller add the flow in the flow table

1. **statistics I got for the floodlight**

http://127.0.0.1:8081/wm/core/switch/all/aggregate/json

{

"00:00:00:00:00:00:00:02":{

"aggregate":{

"version":"OF\_13",

"flow\_count":"1",

"packet\_count":"3740",

"byte\_count":"636136",

"flags":[

]

}

},

"00:00:00:00:00:00:00:03":{

"aggregate":{

"version":"OF\_13",

"flow\_count":"1",

"packet\_count":"3360",

"byte\_count":"610786",

"flags":[

]

}

},

"00:00:00:00:00:00:00:01":{

"aggregate":{

"version":"OF\_13",

"flow\_count":"1",

"packet\_count":"3361",

"byte\_count":"607897",

"flags":[

]

}

}

}

http://127.0.0.1:8081/wm/statistics/bandwidth/00:00:00:00:00:00:00:01/1/json

[

{

"dpid":"00:00:00:00:00:00:00:01",

"port":"1",

"updated":"Mon Sep 11 21:39:33 CEST 2017",

"link-speed-bits-per-second":"10000000",

"bits-per-second-rx":"0",

"bits-per-second-tx":"235"

}

]

http://127.0.0.1:8081/wm/routing/paths/00:00:00:00:00:00:00:01/00:00:00:00:00:00:00:03/2/json

{

"results":[

{

"src\_dpid":"00:00:00:00:00:00:00:01",

"dst\_dpid":"00:00:00:00:00:00:00:03",

"hop\_count":"2",

"latency":"24",

"path\_index":"0",

"path":[

{

"dpid":"00:00:00:00:00:00:00:01",

"port":"2"

},

{

"dpid":"00:00:00:00:00:00:00:02",

"port":"2"

},

{

"dpid":"00:00:00:00:00:00:00:02",

"port":"3"

},

{

"dpid":"00:00:00:00:00:00:00:03",

"port":"2"

}

]

}

]

}

**6.Paper reading**:

* **Live Virtual Machine Migration Techniques:Survey and Research Challenges**

SDN whch allows migrating entire virtual networks from one data center to another or dynamically rerouting external traffic after VM has been migrated.

1App structure

front-end

Back end

business

1. what consider when doing the migration?

* Total migration time
* The pre-copy time
* Down time
* Network storage memory

1. what is the pro-copy mechanism

-> Iterative push pase

-> stop and copy

->resume phase

1. parameters for migration:

* Maximum transfer rate
* Maximum allowed downtime
* Migration timeout

Downtime and total migration time tend to have opposite behaviors

## Optimizing Live Migration of Multiple Virtual Machines

(1) The downtime can be reduced up to two order of magnitudes while increasing the number of transferring rounds.

(2) The total migration time improvement diminishes as we increase the number of transferring

Rounds.

(3) Few transferring rounds are enough to minimize the total migration time.

(4) Near optimal values of bit rate are reached after only few iterations of the primal-dual interior point algorithm.

(5) As the page dirtying rate increases, it is convenient to increase also the number of transferring rounds.

(6)Live migration lowers the total migration time.Increasing the number of transferring rounds does not help anymore after the first few rounds.

(7)The optimization gain is maximized for a small number of transferring rounds.