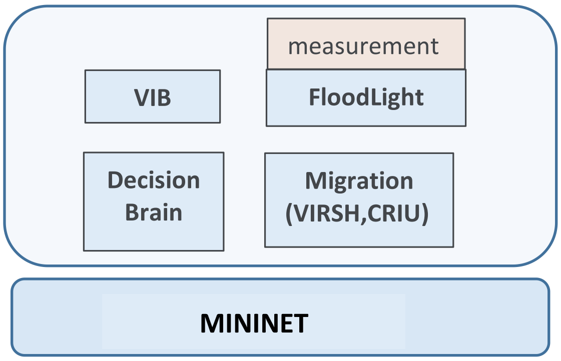
Report for week3

Architecture:(logic of prototype of the thesis)



VIB:

The statistics got from the floodlight,such as bandwidth,latency,aggregate transmission packets number.

Floodlight:

The SDN controller which can manege the network,and provide the northbound API to get the statistics of the network.

Decision Brain:

The logic of the architecture which can use the statistics to make brain decision.

Option1:

Leave it for users to decide.

Option2:

Do research or finding a good method based on measurements collected.

Migration:

The tools can be used to do the migration,VIRSH for migrating the VMs, CRIU for migrating LVMS.

Different migration can be done according to the configuration file,migrate VMs or LVMs.

What can I do based on the Architecture?

1) Create a system that can use SDN to perform monitoring operations on my mininet network Such as software defined traffic measurement(openscretch).
2) Use the measurements info to manege the network better,such as routing,migration,repair. Build an application that can use VIB data structure queues as input of network management,make migration or routing decisions.
3) As use case we will show better to use my Software Defined system to perform migration
   - ➢ Node migration
   - ➢ Link migration (routing)
   - ➢ VM migration
   - ➢ LVM migration

Paper readings:

- **Software Defined Traffic Measurement with OpensSketch**

why?

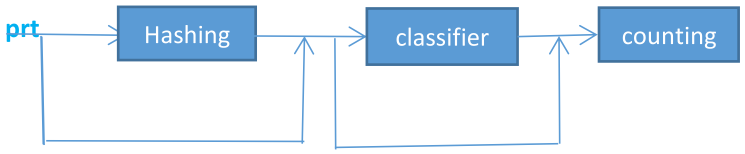
- ➢ Flexible measurement data plane
- ➢ Picking the packets to measure
  - Classify flows(different resource/accuracy)
  - Hashes to represent a compact set of flows

- storing and exporting the data,diverse mappings between counters and flows

Insights:

Select count(*) from * where dstip=10.10.20.3 group by srcip
Groupby: can be accomplished by a hash
Where :can be accomplished by a classifier
Count:by a count primitive

A three-stage pipeline



Eg.
#bytes from 23.43.12.1

Hash1	3	0	**5**	1	9
Hash2	0	1	9	**3**	0
Hash3	1	2	0	3	4

- ✓ Automatic configure with a task
- ✓ Resource collection cross tasks

Conclusion:

- **Sketches accuracy memory trade-off**
- **Making sketches easy to implement and use**
  -> different measurement task
  -> easy to implement with commodity switch hardware
  -> modularized library for easy programming

MOZART:Temporal Coordination of Measurement

MOZART: MOnitor flowZ At the Right Time
Temporal coordination of measurement

Why?

It is useful for one device to tell another one which flows to monitor at which time,rather than monitoring all the flows all the time.

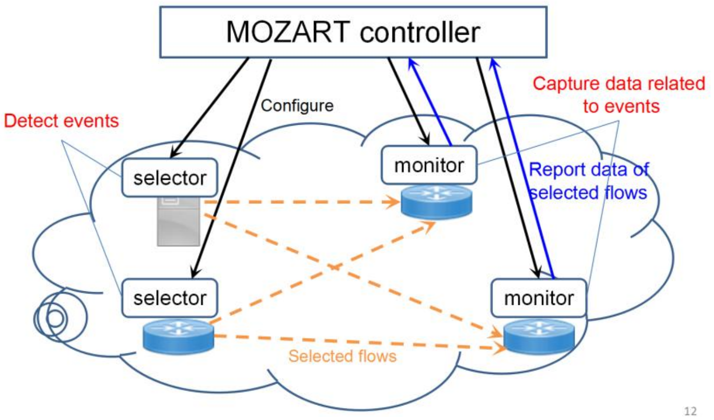
How?

Two key components:

- Selectors which capture networks events and select related flows
- Monitors which collect flow-level statistics of the selected flows

Architecture:

MOZART framework



Conclusion:

- ❖ **Temporal coordination is important**
  --collect data related to events
  --Different views/abilities of devices
- ❖ **MOZART design highlights**
  --Coordination algorithms
  --Placement algorithm for maximizing tasks to run
- ❖ **Benefits**
  --High measurement accuracy
  --Support more tasks
  --Meet memory constraints in devices

● Re-evaluating Measurement Algorithms in Software

Why?

Measurement is Critical for NFVs, make decisions based on measurement input

Firewall,loading balancing,intrusion detection systems(IDS)

Measurement for managing NFVs

Profiling NFV usage,resource scheduling..

Nowadays,the software switch is popular,so the concern is not as the same as traditional switch,network device,before memory is a critical part.now there are plenty of memory for measurement.

Conclusion:

- ❖ NFV is the new trend in data-center and ISPs
- ❖ Measurement is a key component for NFV
- ❖ Simple hash table works best
  - For many tasks,the working set fits in the cache
    - Especially when the traffic is skewed