OpenFlow Data Center A Case Study

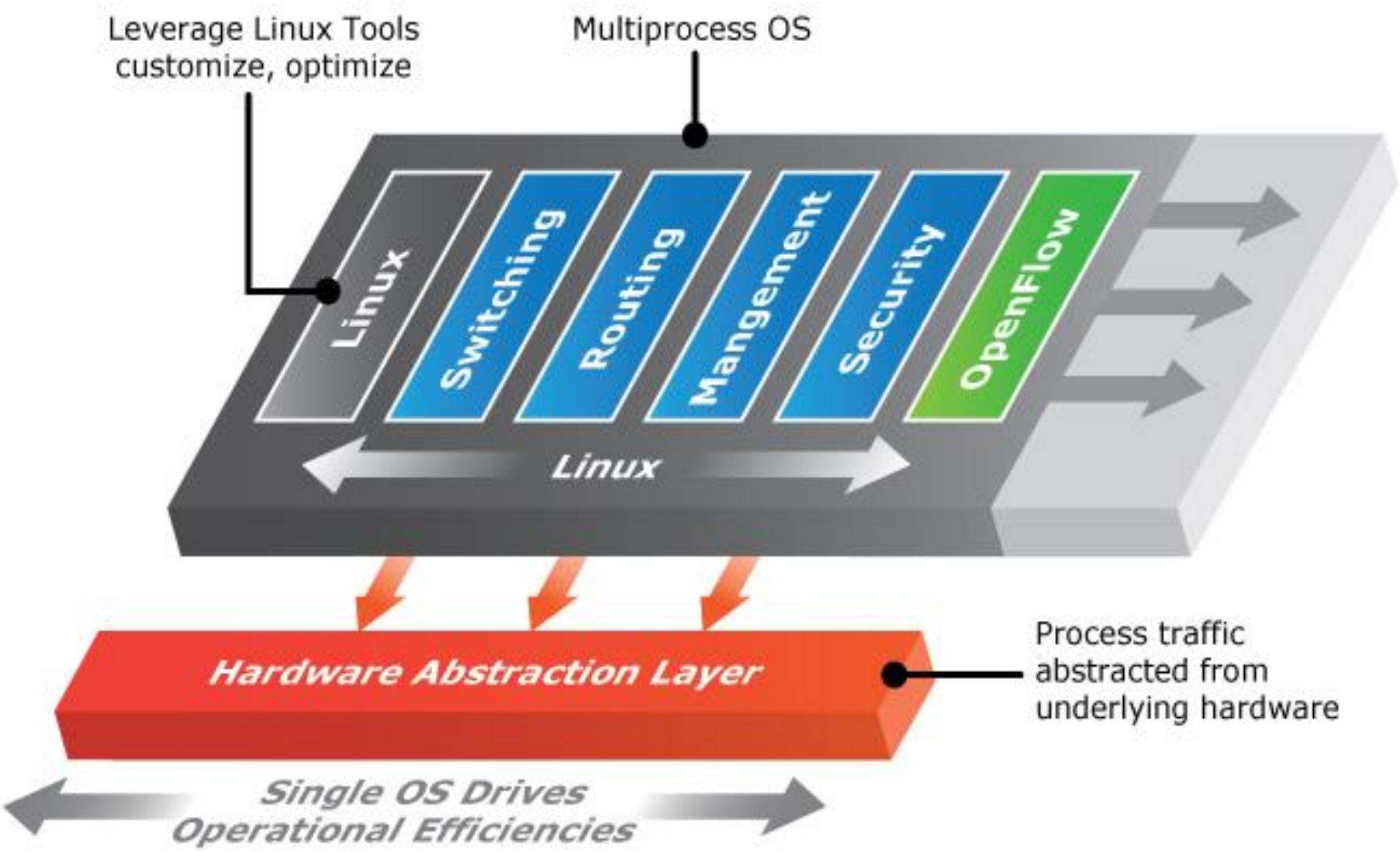
David.Liu@Pica8.com

2/27/2014

Topics

- Pica8 Introduction
- Network design goals
- Constraints
- How to scale
- Key OpenFlow features
- Controller
- Network management
- Current status

Pica8 Introduction

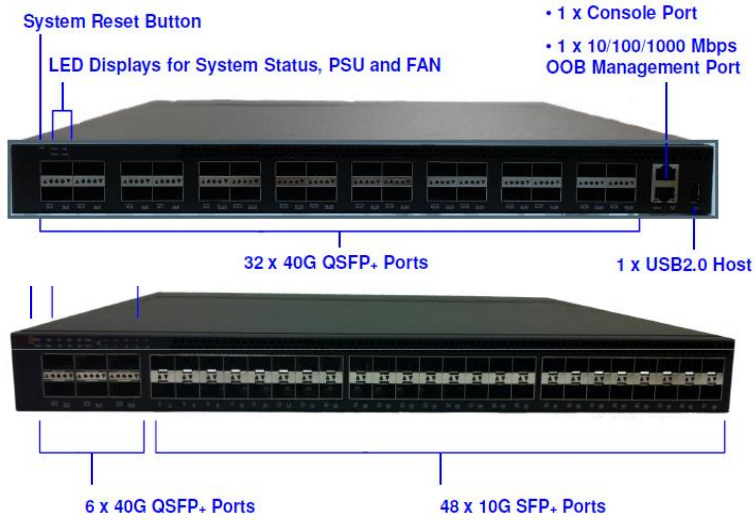


White Box Platforms

- Broadcom Fire Bolt & Triumph2
  - 48 x 1G + 4 x 10G

- Broadcom Trident +
  - 48 x 10G + 4 x 40G

- Broadcom Trident 2
  - 32 x 40G
  - 48 x 10G + 6 x 40G

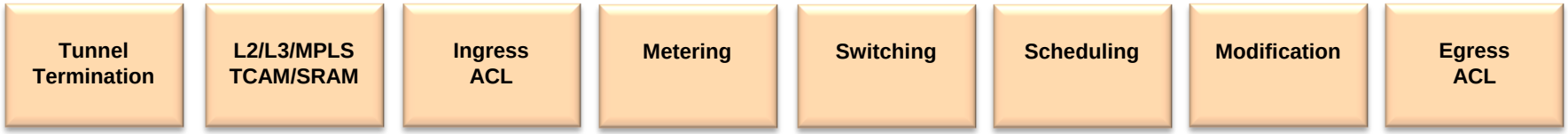


Network Design Goals

- Green fields white box data center
- No legacy inside data center
- Floodless network
- Complete control over
  - IP addresses
  - Forwarding decisions, while and black lists
  - Topology
  - Controller architecture
- Maximum visibility
- 100% automation

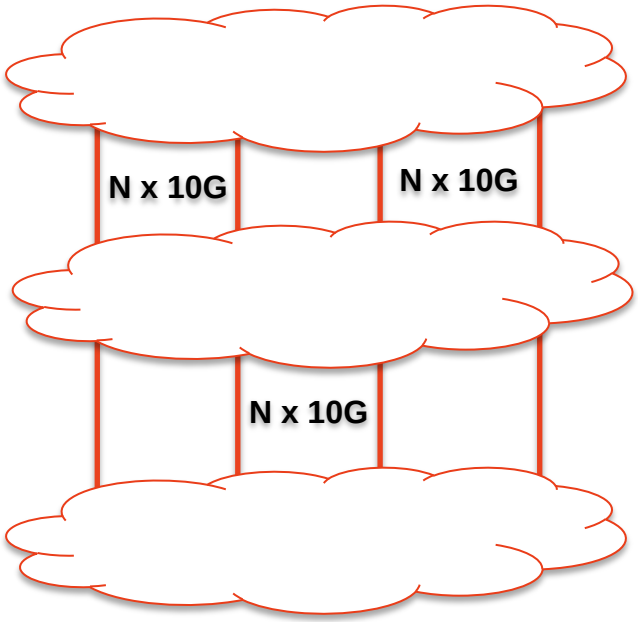
Starting Constraints (4-2013)

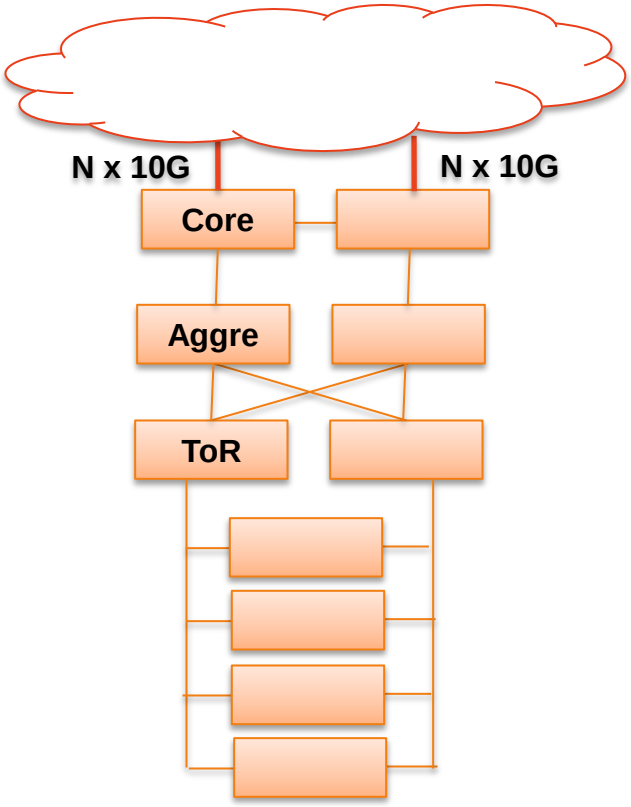
- TCAM size, 1K – 2K entries based on model
- Fixed ASIC pipeline, not all OpenFlow features can fit



- No multiple table support
  - How to forward based on dl_src then set_queue without multiplying the number of rules
- No positive ack after flow_mod
- OpenFlow v 1.2
- Group table/Select, ECMP & fast failover
- Zero touch provisioning
- Metering

Scale under Constraints





- **Scaling unit = 2 Core + 2 Aggregation + 12 Racks (2 ToR + 40 servers)**

ToR Rules

- **Drop**

Priority	eth_type	actions
50000	ipv6	actions=drop
50000	rarp	actions=drop
50000	dl_type=0x8809	actions=drop
50000	dl_type=0x8100	actions=drop

- **Learning**

Priority	eth_type	match		actions
40000	ip	in_port=29	nw_src=100.0.0.29	actions=CONTROLLER:1024
40000	arp	in_port=23	arp_spa=100.0.4	actions=CONTROLLER:1024
40000	ip	in_port=36	nw_src=100.0.0.36	actions=CONTROLLER:1024
40000	ip	in_port=48	nw_src=100.0.0.154	actions=CONTROLLER:1024

- **Forwarding**

Priority	eth_type	match			Actions
20000	ip	in_port=48	nw_src=100.0.0.128/25	nw_dst=100.0.0.5	actions=set_field:00:11:90:9b:c8:38->eth_dst, output:5
20000	ip		nw_src=100.0.0.0/25	nw_dst=100.0.0.5	actions=set_field:00:11:90:9b:c8:38->eth_dst, output:5
10000	ip	in_port=49	nw_dst=100.0.0.5		actions=set_field:00:11:90:9b:c8:38->eth_dst, output:5
10000	ip	in_port=50	nw_dst=100.0.0.5		actions=set_field:00:11:90:9b:c8:38->eth_dst, output:5
10000	ip	in_port=51	nw_dst=100.0.0.5		actions=set_field:00:11:90:9b:c8:38->eth_dst, output:5
10000	ip	in_port=52	nw_dst=100.0.0.5		actions=set_field:00:11:90:9b:c8:38->eth_dst, output:5
20	ip		nw_src=100.0.0.5	nw_dst=100.0.0.0/8	actions=group:55
10	ip		nw_src=100.0.0.5		actions=set_field:00:11:90:c8:bc:3e->eth_dst,output:3

Aggregation Rules

- **Drop**

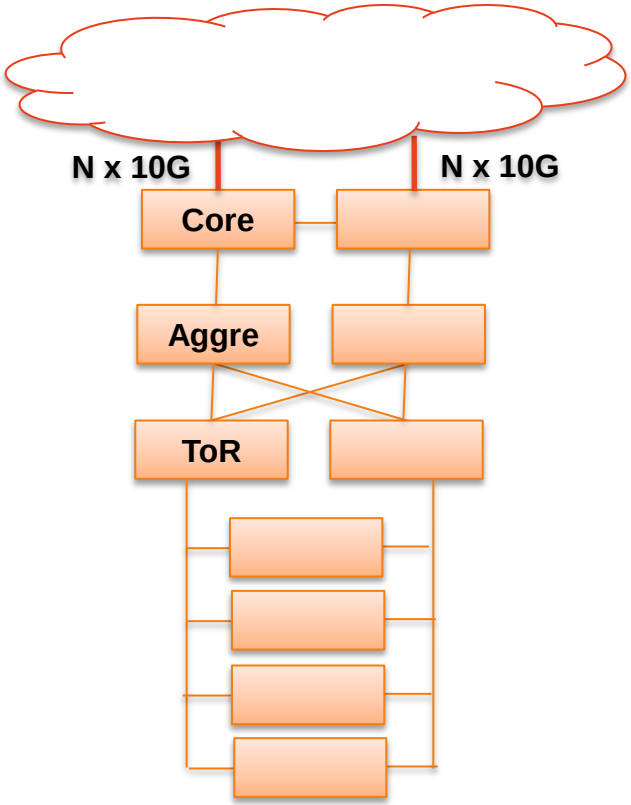
Priority	eth_type	actions
50000	ipv6	actions=drop
50000	rarp	actions=drop
50000	dl_type=0x8809	actions=drop
50000	dl_type=0x8100	actions=drop

- **Forwarding**

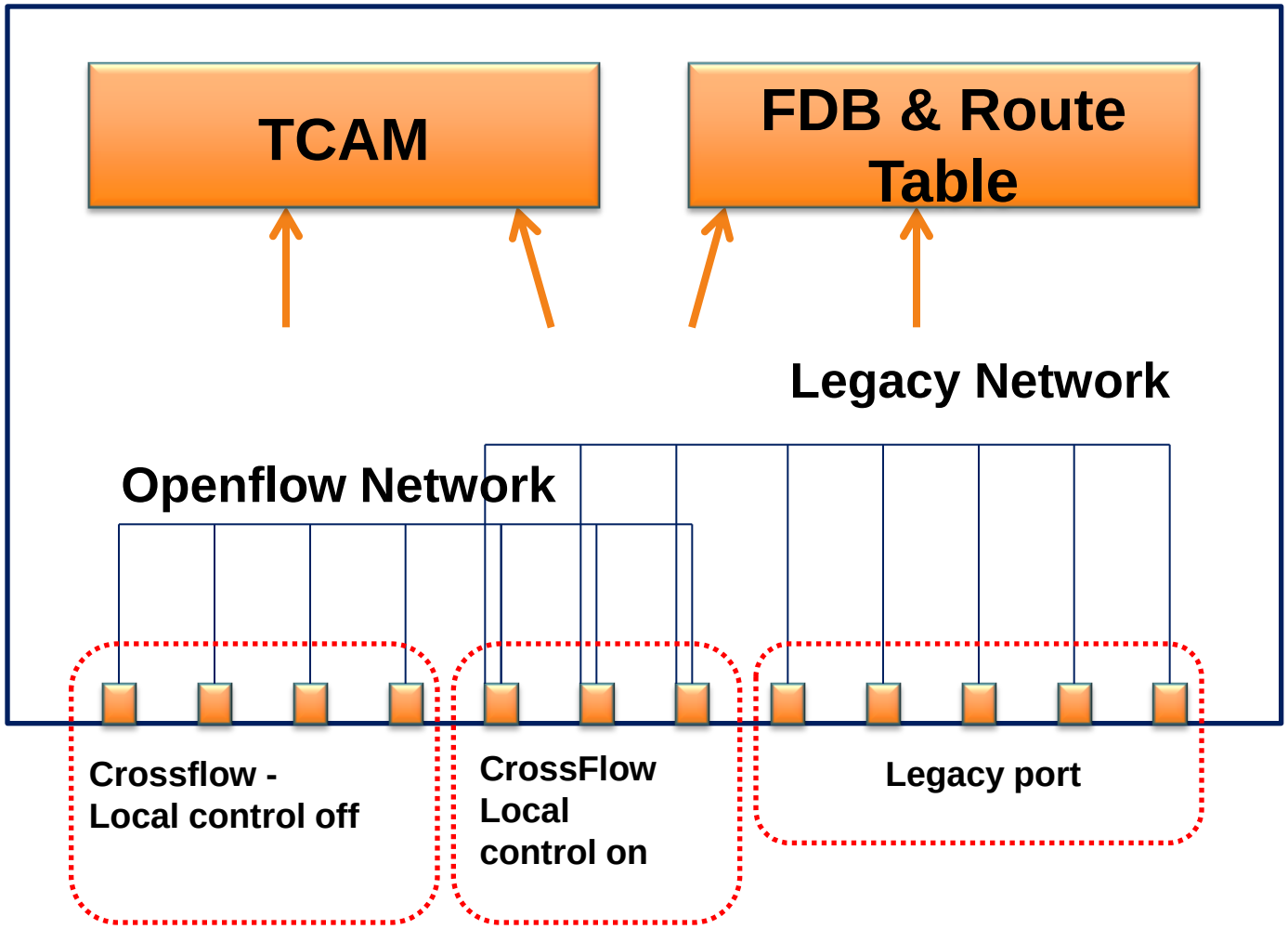
Priority	eth_type	match	actions
40000	ip	nw_dst=100.0.0.5	actions=group:53
40000	ip	nw_dst=100.0.0.0/22	actions=group:55
40000	ip	nw_dst=100.0.4.0/22	actions=group:58
30000	ip	nw_dst=100.0.16.135	actions=group:10

Key Features

- TCAM optimization (2K, 4K, & 8K entries)
- Group Select / ECMP
- OpenFlow v1.3
- Metering
- Pre-configured queuing
- Analytic DB
- Zero touch provisioning
- Configurable event reporting
- Switch inventory
- Cross flow mode



Cross Flow Mode



Controller

- In house controller development
- Use a stripped down open source controller parser
- Less than 1 year development & ready for production
- Centralized or distributed
- In progress
  - OVS DB interface integration
  - Inventory
  - Event reporting
- New ideas on how a controller framework could be

Network Management

- SNMP ?
- If it is needed, should it be on the switch ? Or ..
- Inventory
- Event reporting
- Controller management

Conclusion

- OpenFlow cannot solve all problems but fits well with some applications
- All OpenFlow application needs customization with domain knowledge, no off-the-shelves solution
- Scalability has to be designed
- Allow automation to save OPEX
- Controller is still the key to OpenFlow deployment