Revaliwhat?

Keeping kernel flows fresh

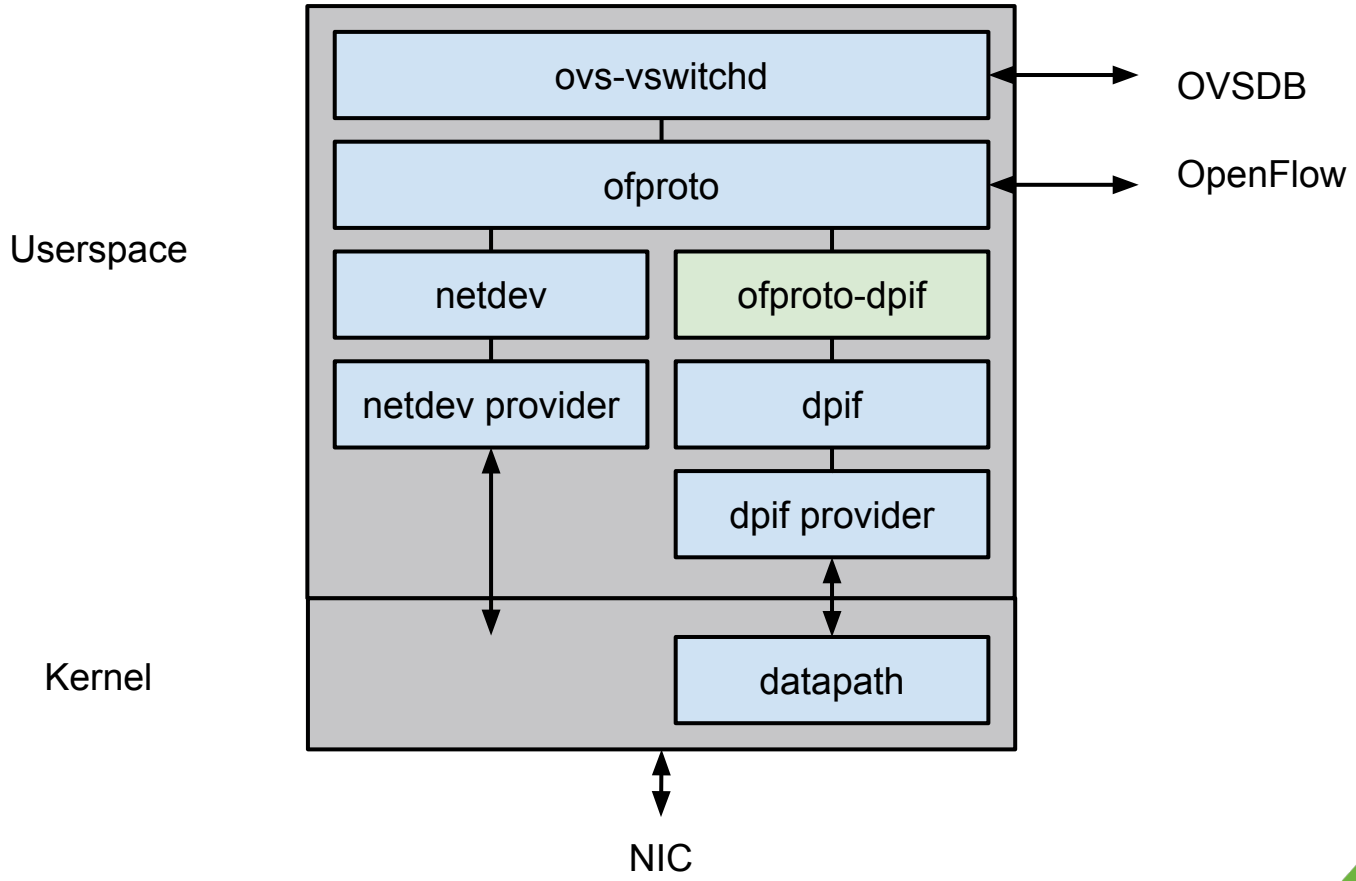
Joe Stringer, NSBU

vmware®

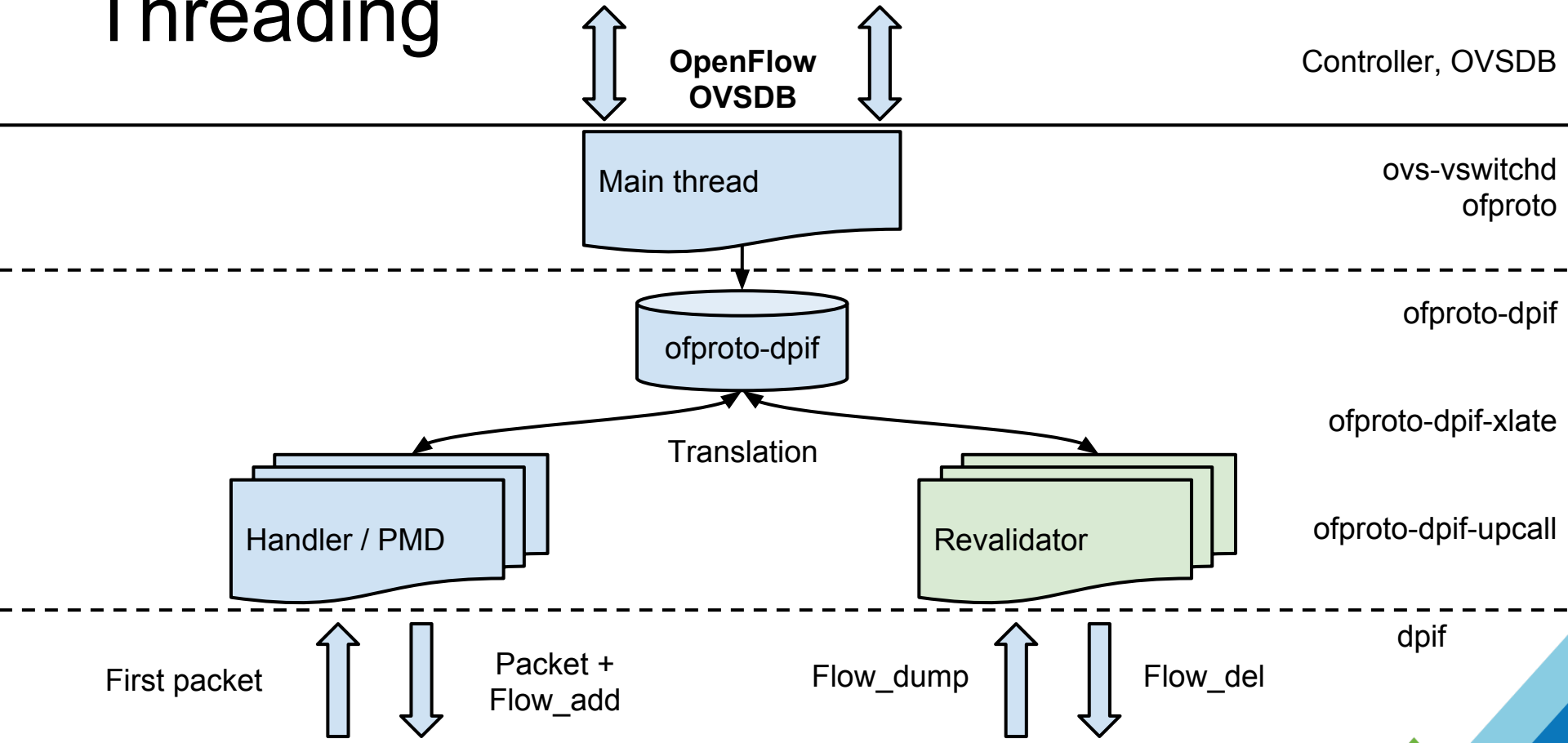
Overview

- This is a dev talk
- OVS Threading
- Lifecycle of a datapath flow
  - “OpenFlow rule” -> “rule”
  - “Datapath flow” -> “flow”
  - Userspace vs. Datapath (kernel/DPDK)
- Optimizations

Recap: OVS architecture

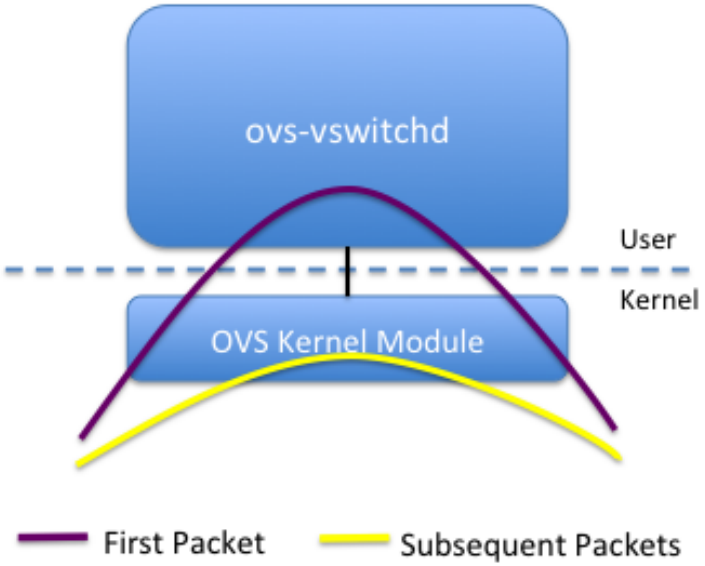


Threading



Miss Handling - “handler” threads

- Birth of a kernel flow
- Install and forget



http://networkheresy.com/2014/11/13/accelerating-open-vswitch-to-ludicrous-speed/

Cache validation (re-validation..)

- Flow was installed by handler.
- How do we delete the flow?
  - When do we delete it?
- Is it still valid? What about now?
  - OpenFlow/OVSDB changes
  - Rules that cause learning
- How do we keep statistics up to date?

Revalidator thread

- Fetch flows from datapath
- Translate from datapath format to userspace
- Run flow through ofproto-dpif classifier
  - Attribute statistics (rules, ifaces, netflow, etc)
  - Execute side-effects (L2 learning, rule learning)
- Check that the flow is correct
  - Delete idle/incorrect flow
  - Only needs to be done if something changed

Revalidator phases

- Dump flows - mark
  - Create lightweight "udpif_key" (ukey) cache.
    - One per flow; stores latest seen statistics,used
- Garbage collect - sweep
  - Iterate through ukey cache
  - Delete old ukey cache entries
  - If flow hasn’t been seen in a while, fetch/revalidate
- Synchronize all revalidator threads

Still with me?

- Fast enough with megaflows
  - Great! Can we get more performance?
- Insights and improvements
  - Testing
  - Flow limits
  - Flow deletion
  - Translation cache
  - Shorter identifiers

Testing revalidator performance

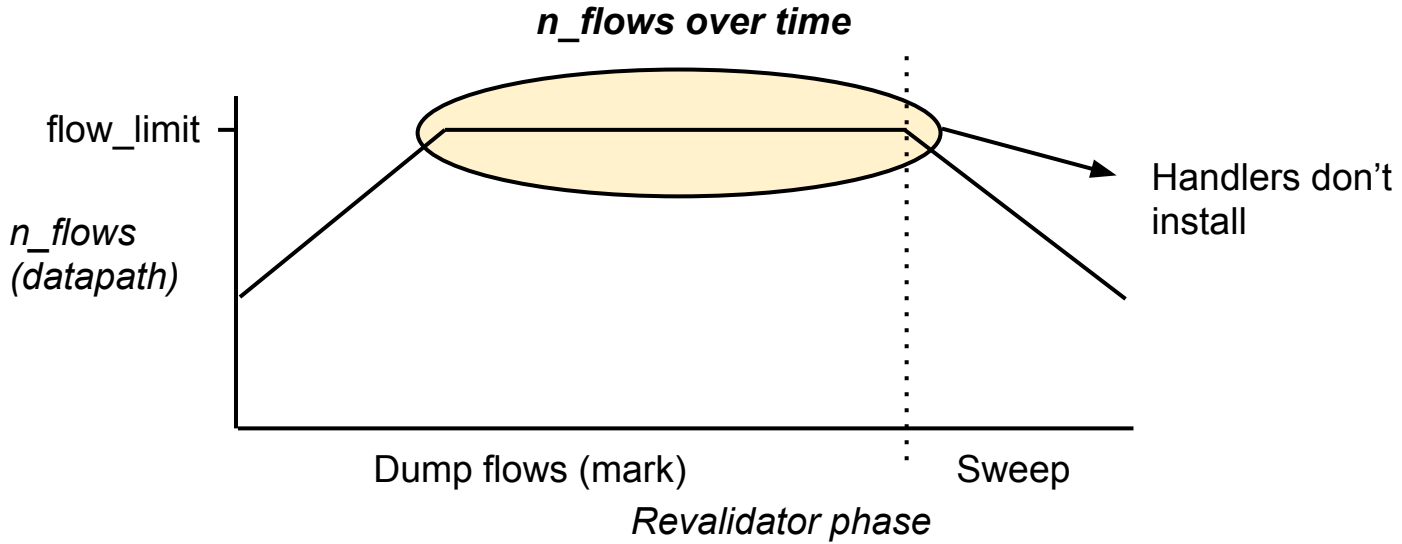
- This is not a sane way to run OVS
  - But useful to determine “worst case” revalidation
- ovs-appctl upcall/disable-megaflows
- ovs-appctl upcall/show
- netperf TCP_CRR
- nmap
- Change rule table several times a second
  - Raises “need_revalidate” flag

Flow limit

- How many flows can we support?
  - Keep revalidator cycle around 1 second
- Limit on # datapath flows
  - Handlers stop installing flows, only execute
- How do we determine the limit?
  - Revalidation cycle takes <1000ms? increase
  - >1200ms? decrease… >2000? decrease more
  - Linked to max idle time for flow

Mark and Sweep

- Flow limit for installation affects deletion



- Delete flows throughout dump phase

Flow Dump Accuracy

- Flow dumps not 100% accurate
  - Multiple threads inserting, deleting flows
  - Flow dump is just a pair of [bucket,index]
  - Dumping batches
  - Can cause duplicates or missed flows
- Keep ukey cache around until GC phase
  - Track whether the flow was previously deleted
  - Don’t handle duplicate flows

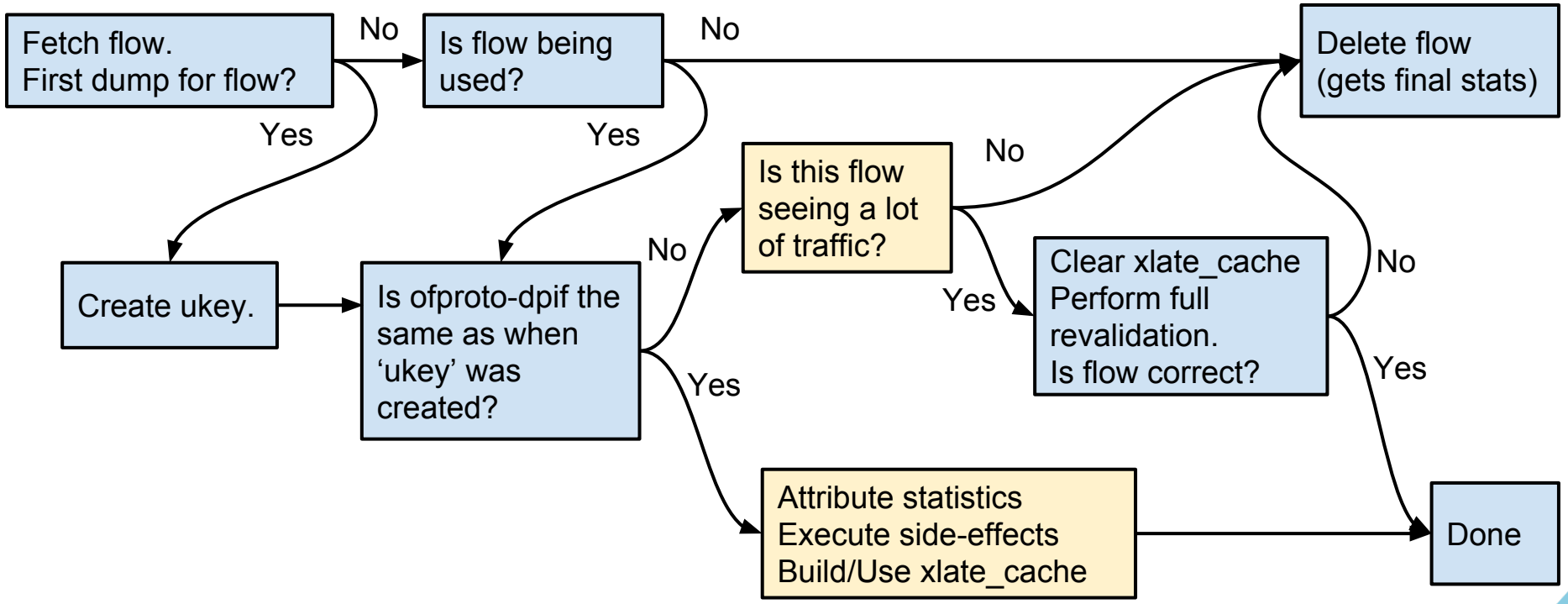
Datapath flow max_idle

- No use caching idle flows
  - If n_flows > flow_limit, we really need the space
- OVS 2.1-2.3: 1500ms idle
  - When n_flows > flow_limit, 100ms
  - When > 2*flow_limit, delete all
- Master: 10s
  - Remain cached if used less consistently
  - Small improvement in max flow_limit

Translation cache (“xlate_cache”)

- Translation/Classification is expensive
- Cache the results of translation/classification
  - Which statistics do we need to update
  - What side-effects do we need to execute
- Gets invalidated when ofproto changes
  - Simple case fast, “full revalidation” still slow
  - Delete low-throughput flows.
  - If full revalidation is too expensive, flows are deleted anyway.

Revalidator logic



50-80% revalidator performance improvement:
https://github.com/openvswitch/ovs/commit/b256dc525c8ef663daf2330463e67a26207cc5f1

Unique Flow Identifiers (“UFID”)

- Every flow operation requires full flow key
  - “in_port(2),eth(src=50:54:00:00:00:01,dst=50:54:00:00:00:03),eth_type(0x0800),ipv4(src=192.168.0.1,...)”
  - Flow dump = fetch 10^5 flows/sec
  - nla_format (flow serialization->netlink) is #1 in perf
- Replace with identifier (128-bit)
  - Handler creates “ukey” with with id,key,mask,acts
  - Revalidator only fetches id + stats
  - Up to 50% performance increase for revalidator

Thanks for listening!

Questions?

Potential future topics

- Improve flow dump correctness in datapath
  - Keep track of flow add/deletions
- Combine flow dump + flow delete
- Link rule changes to flows
  - Better xlate_cache invalidation
- DPDK revalidation
  - PMD threads could be more accurate