Optimizing OvS using DPDK Membership Library

Yipeng Wang & Sameh Gobriel

Intel Labs

Legal Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. **No computer system can be absolutely secure**. Check with your system manufacturer or retailer or learn more at intel.com.

Contributors

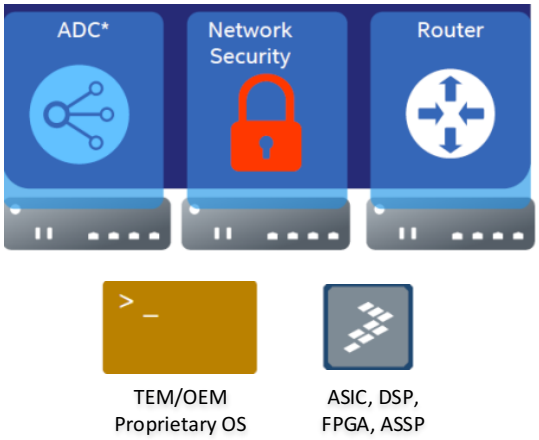
Charlie Tai Charlie.tai@intel.com

Ren Wang ren.wang@intel.com

Antonio Fischetti antonio.fischetti@intel.com

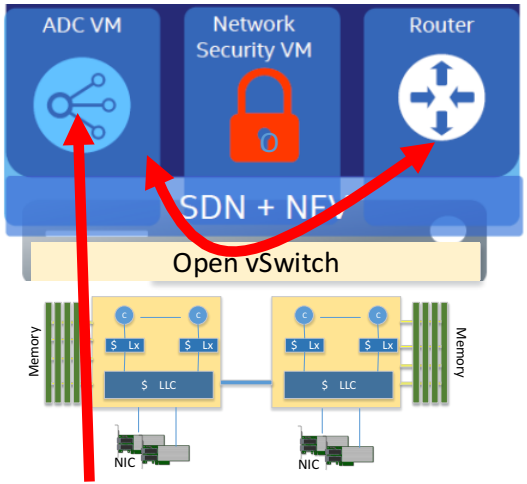
OvS De Facto Virtual Switch for NFV Environments

Monolithic Purpose-built Boxes



NFV

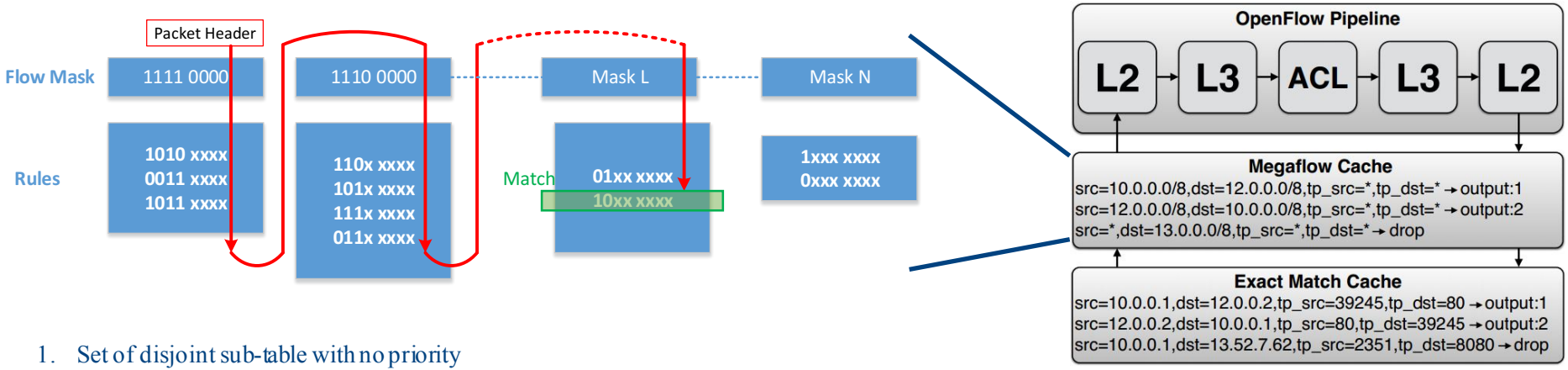
Networking VMs on Standard Servers



- Network appliances use purpose-built H/W & ASICs (e.g., TCAM) for flow classification
- Cost & power consumption are limiting factors to support large number of flows

- General purpose processors with Cache/memory hierarchy can support much larger flow tables.
- Multicores architecture provide a scalable competitive flow classification performance.

Open vSwitch Flow Lookup



1. Set of disjoint sub-table with no priority
2. Rule is only inserted into one sub-table (lookup terminates after first match)
3. Lookup is done by sequentially search each sub-table until a match is found

OvS Flow Classification is a bottleneck

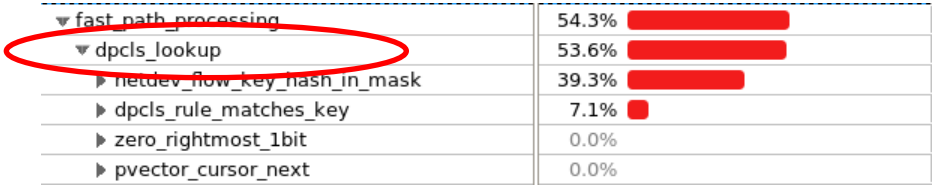
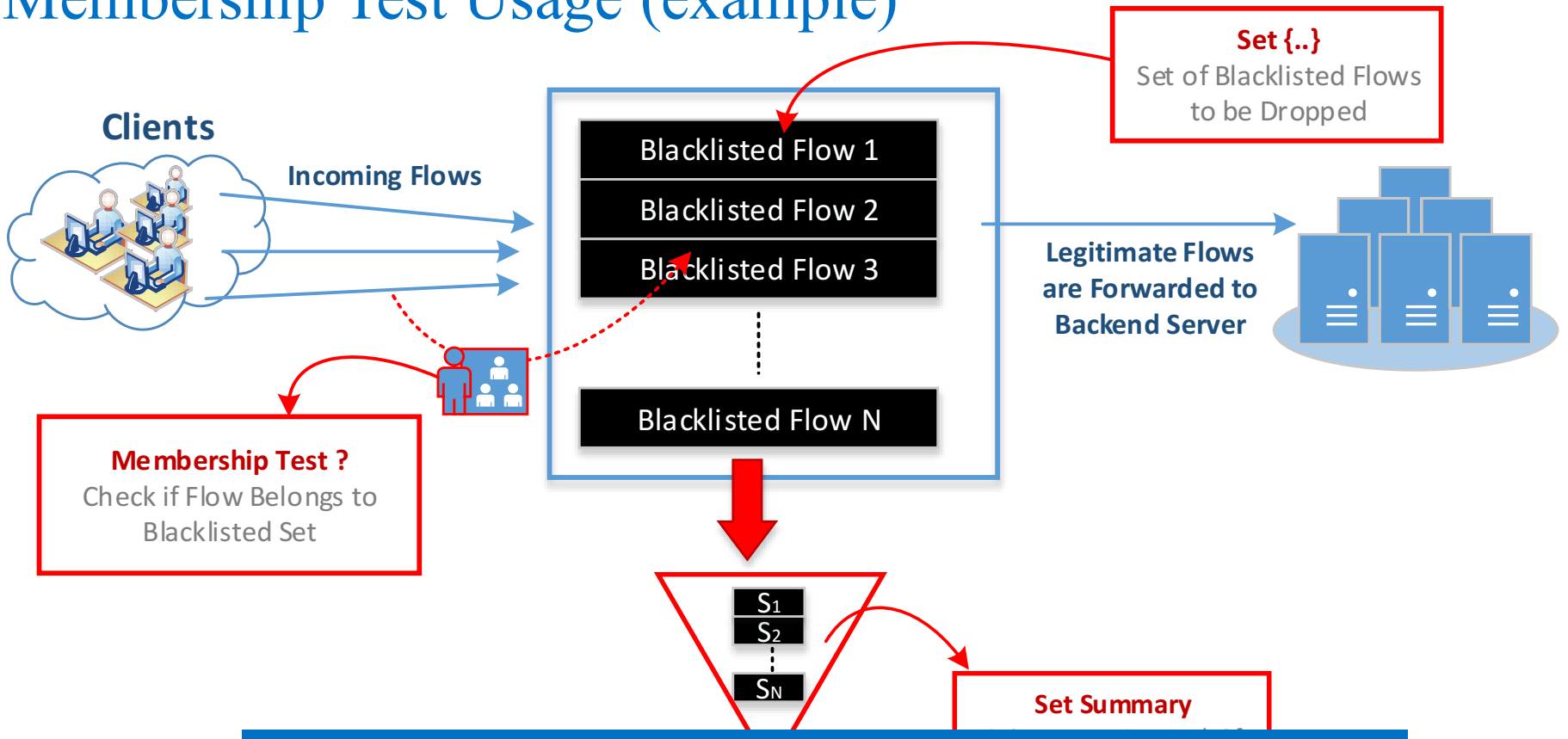


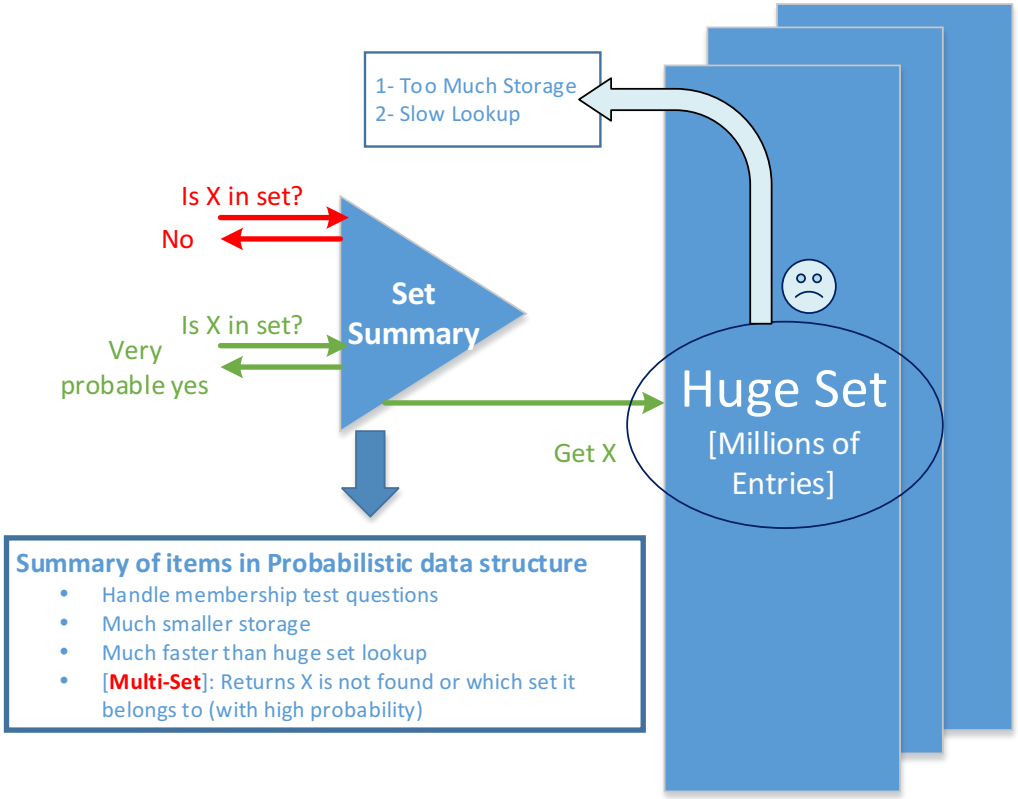
Fig. Vtunes OVS flow lookup process (bypass EMC). Test case: 20 sub-tables, each has 100 rules.

Membership Test Usage (example)



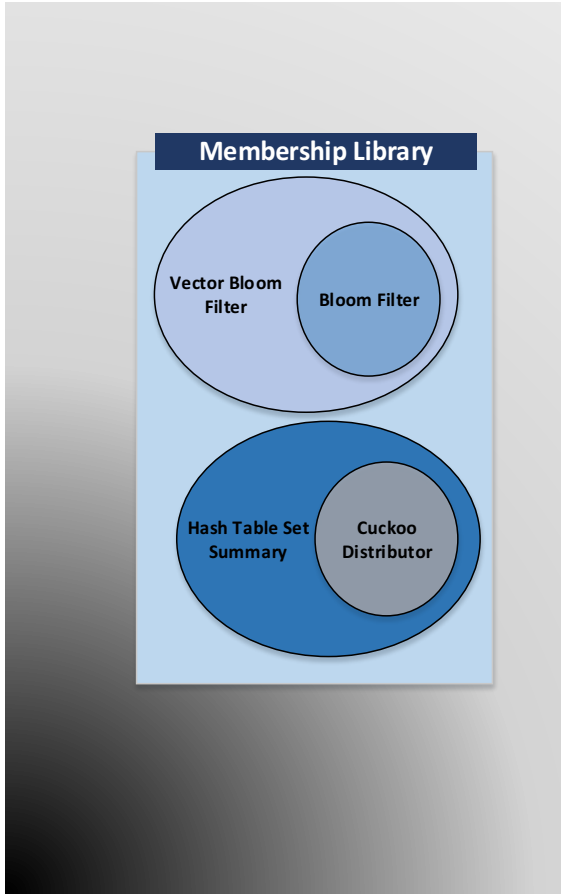
Membership Library is a DPDK Library to Provide Users the Functionality to Create Different Types of Set-Summaries

Overview of DPDK Membership Library in V17.11

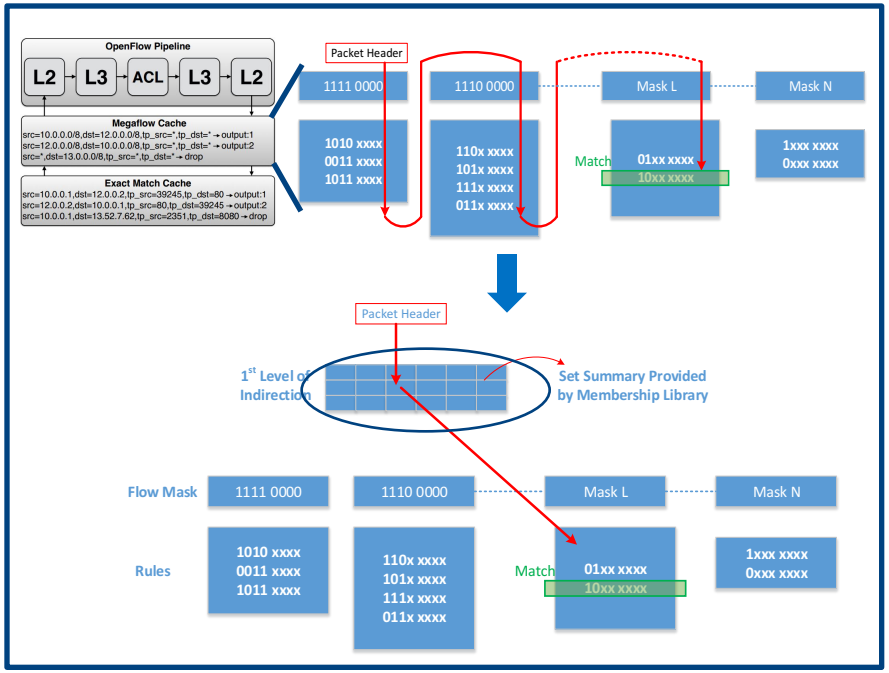


Summary of items in Probabilistic data structure

- Handle membership test questions
- Much smaller storage
- Much faster than huge set lookup
- [**Multi-Set**]: Returns X is not found or which set it belongs to (with high probability)

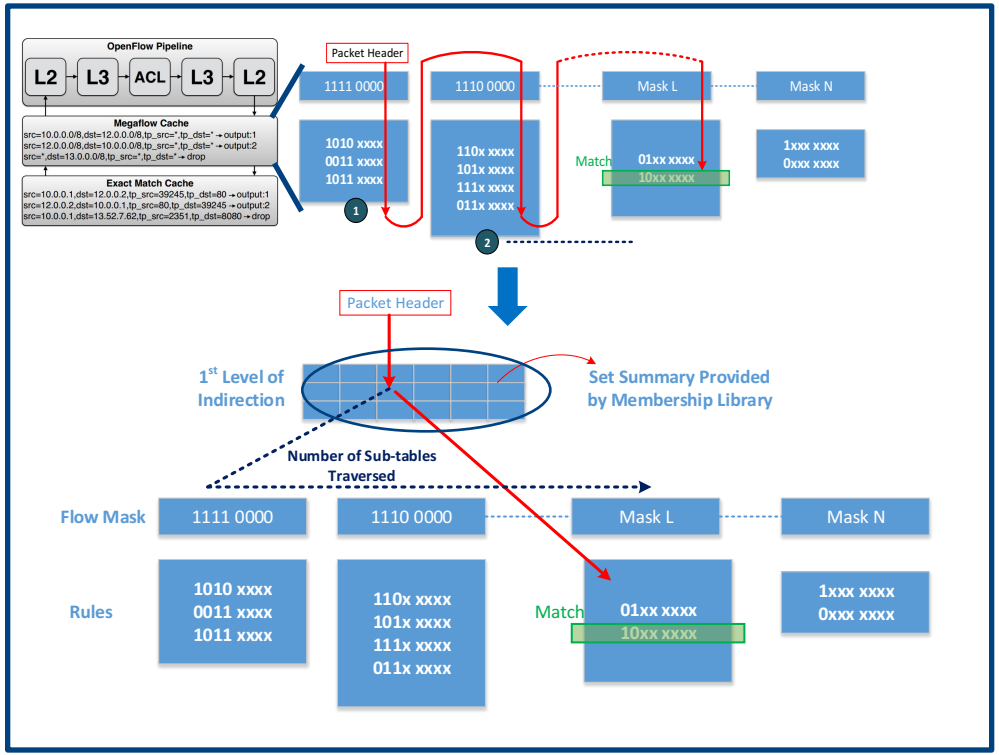


Two Level Lookup for MFC



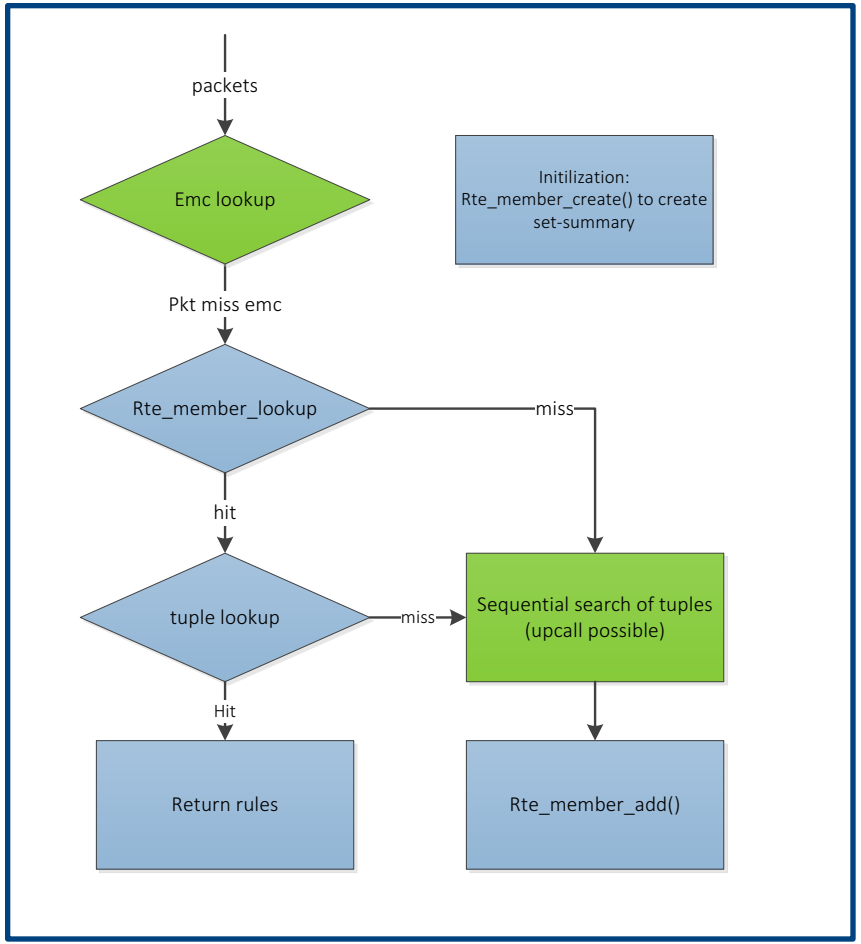
- Membership library used to create a 1st level set-summary indirection
- Flow Keys are looked up in set-summaries:
  - Hits: directs to the correct sub-table for searching (correct 97%)
  - Misses: “New” flow default sequential search & upcall if needed

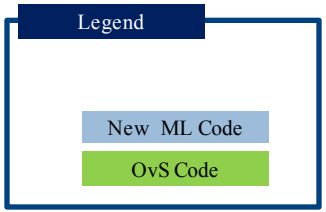
Dynamic Operation & Sub-Table Ranking



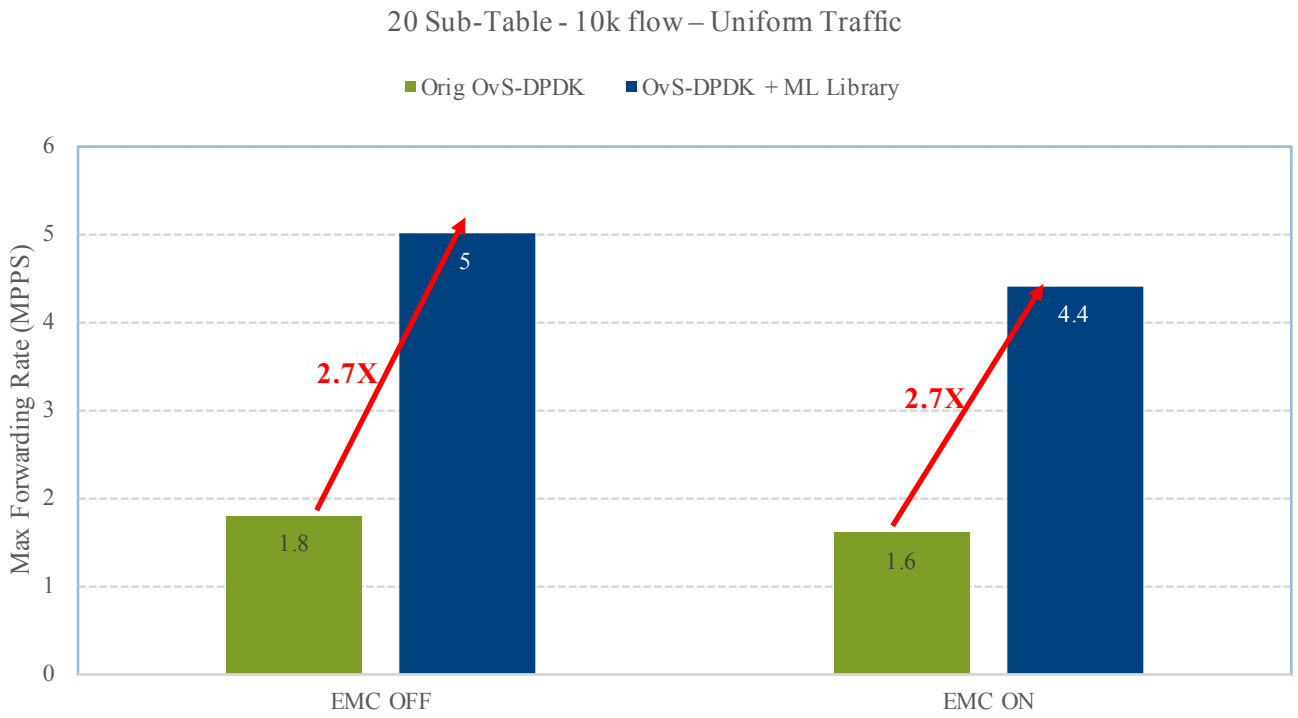
- Sub-table Ranking:
  - Based on number of hits per sub-table → optimize the order of sequential search.
- First level is switched ON/OFF
  - If average number of sub-tables (without first level) traversed is small → turn off

Implementation Overview





Performance Gain



2X-3X Throughput Improvement for OvS using DPDK Membership Library

Conclusion

- MegaFlow Lookup has scalability bottleneck, especially with uniform distribution traffic patterns.
- The membership structure optimizes flow lookup in OvS and avoids the sequential search of the sub-tables.
- Using DPDK Membership Library, first level of indirection is created to direct flow to the correct sub-table.
- Dynamic turning on/off to avoid overhead of first level when not needed.
- DPDK V17.11 released with Membership Library … Patch to be submitted to the mailing list, please review and test in your workload.

Questions?

sameh.gobriel@intel.com
charlie.tai@intel.com