OVN Controller
Incremental Processing

Han Zhou
OVN meetup 5/15/2018
Incremental Processing Engine

- DAG representing dependencies

- Each node contains
  - Data
  - Links to input nodes
  - Change handler for each input
  - Full recompute handler
  - Change

- Engine
  - DFS post-order traverse the DAG from the final output node
  - Fall back to recompute if for ANY of its inputs:
    - Change handler is not implemented for that input, or
    - Change handler cannot handle the particular change
Throughput Improvement

- Create and bind 10k ports on 1k HVs
  - Single thread
  - Batch size 100
  - Bind port one by one for each batch
  - Wait all ports up before next batch
Latency Improvement

- End to end latency when 10k port already there
  - Create one more logical port
  - Bind the port on HV
  - Wait until northd generating the lflows and enforcing on all other HVs
CPU Efficiency

![Chart showing CPU time comparisons between Branch-2.6, Branch-2.9, and Optimized versions.](image)
TODOs

- Code review (https://patchwork.ozlabs.org/project/openvswitch/list/?series=44369)
- Handle more input changes for incremental processing, e.g. address sets, port groups, mac-bindings, etc.
- Split runtime-data and flow-output nodes with more fine-grained dependency.
- Handle corner cases of port-binding changes for logical flow generation, e.g.
  - ACL is created before lsp is created
    - to-lport 1000 'outport="lsp_A" && inport="lsp_B_not_created_yet"' allow-related
- Improve change tracking
  - Keep data across main loop iterations
  - Provide old data when operation is update
- Further improvement of performance for ofctrl_put(), avoid full table scan and compare everytime.