

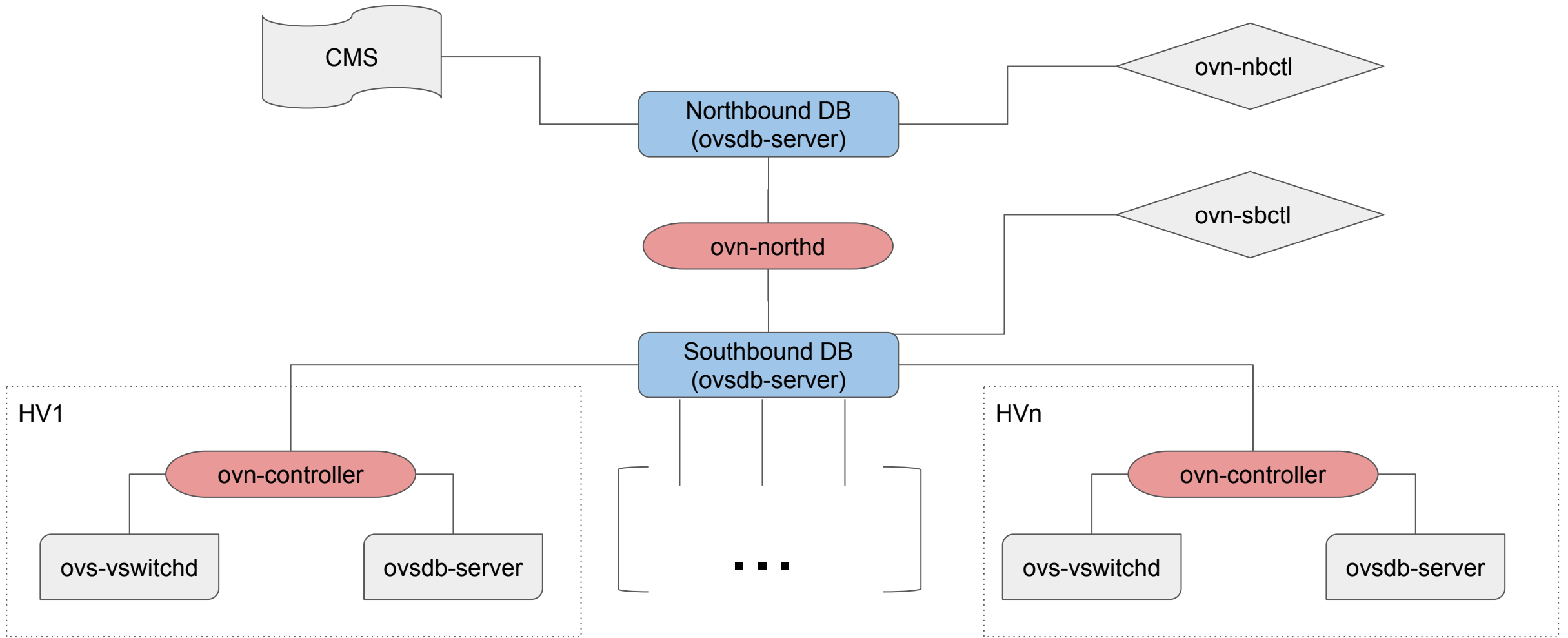
Debugging OVSDDB with stream record/replay

Open vSwitch and OVN 2020 Fall Conference

Ilya Maximets

Senior Software Engineer

OVN architecture



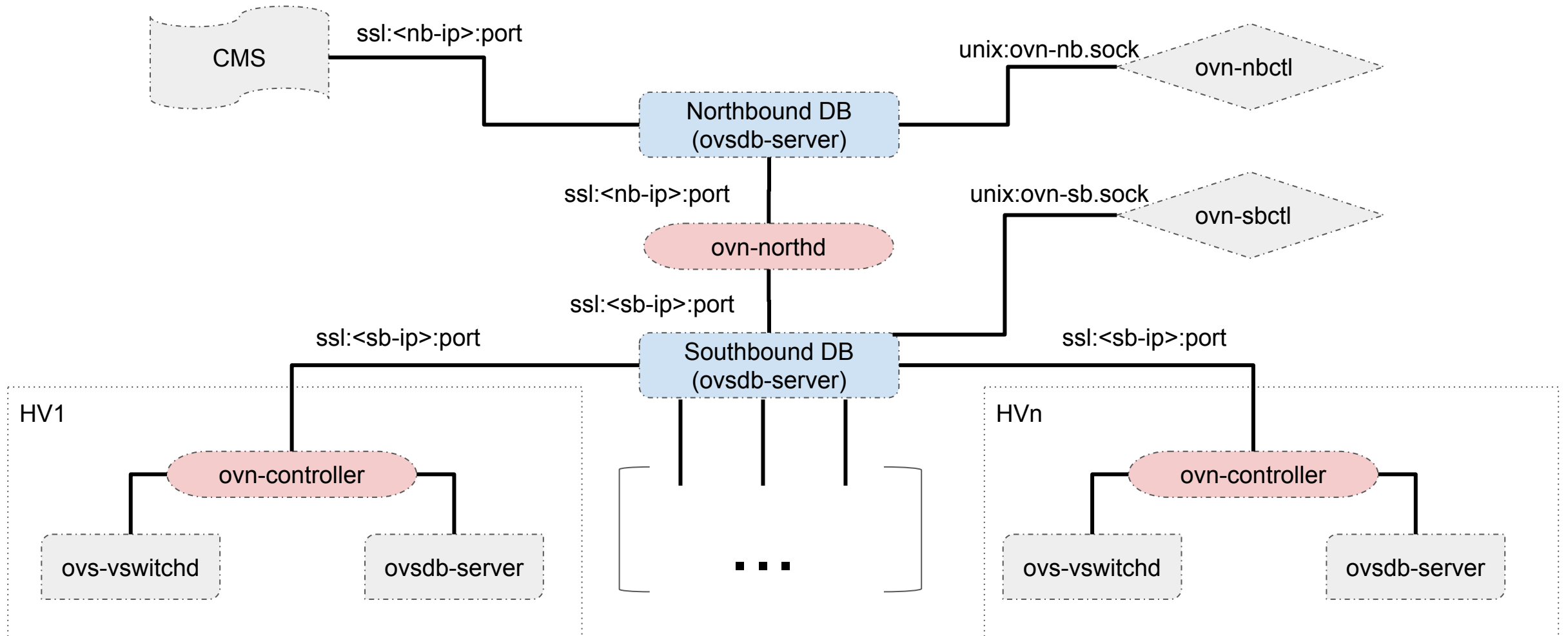
How to debug issues with OVN components?

- Live debug session on an affected setup.
 - Holding a faulty environment for a long time is usually not an option, especially if it's a large setup with hundreds of physical nodes.
- Copy database files from the affected setup and try to replicate in a sandbox (ovs-sandbox, ovn-fake-multinode).
 - Issue might require exact order of database operations or specific events like disconnection of one of the clients in order to reproduce. Might be not easy to replicate needed conditions in a sandbox environment.

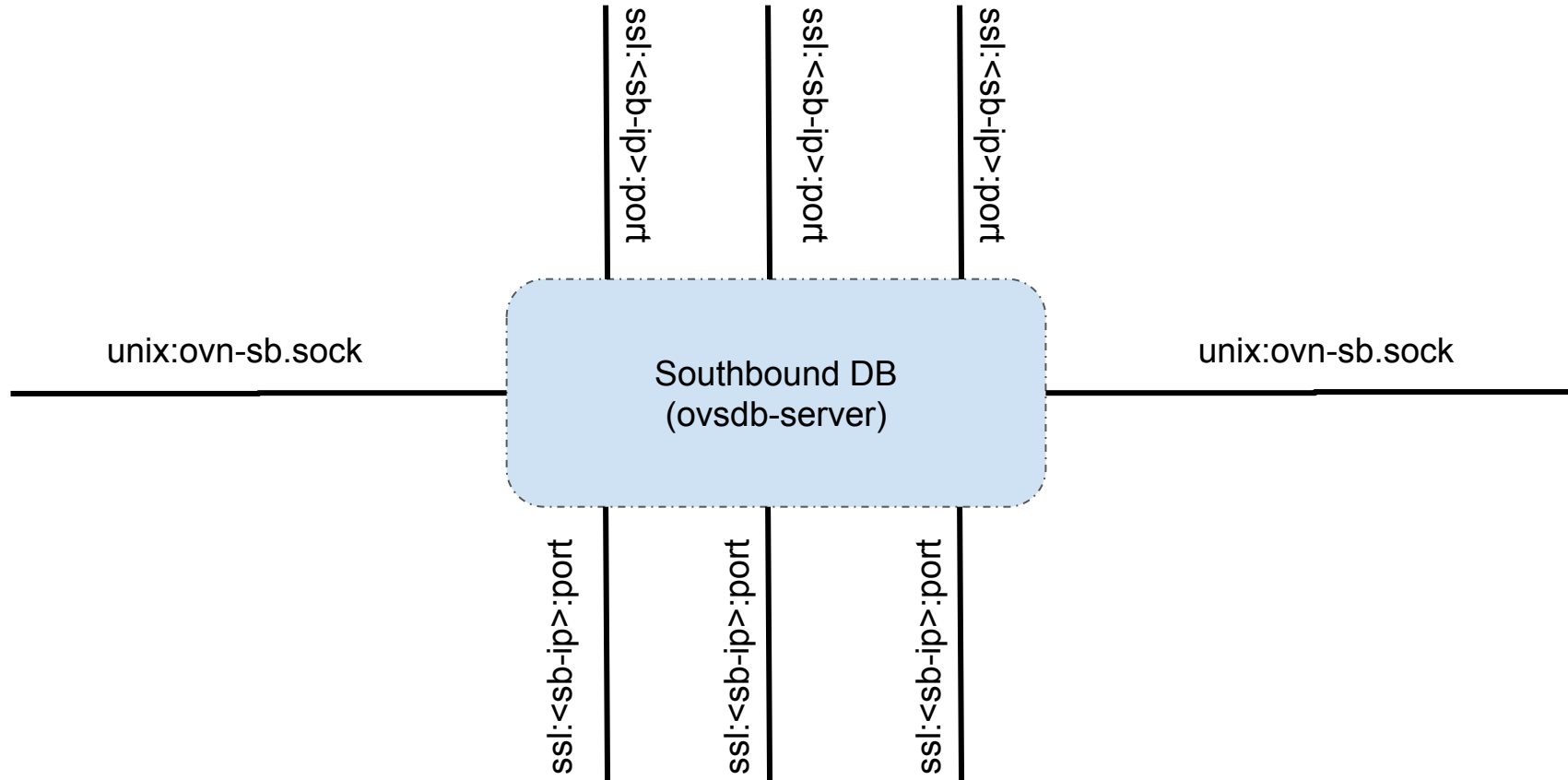
How to test performance optimizations?

- Typical solution is to run synthetic scalability tests, e.g. with ovn-scale-test (<https://github.com/ovn-org/ovn-scale-test>) or something similar.
 - Takes significant amount of time on high scale levels.
 - Might require decent amount of hardware to spin up fake clusters with big number of fake nodes.
 - Not a real-world workload.
- Real-world testing with actual workloads.
 - Not a convenient way to test optimizations during development.
 - Same issues with time and hardware.

OVN architecture (connections)



Southbound DB (connections)



Do we really need to have full OVN setup in order to test it?

- No need to have 100 nodes to test Southbound DB. What is needed is to have 100 incoming connections with particular data.
 - The same mostly applicable to all other OVN components.
- But how to do that?

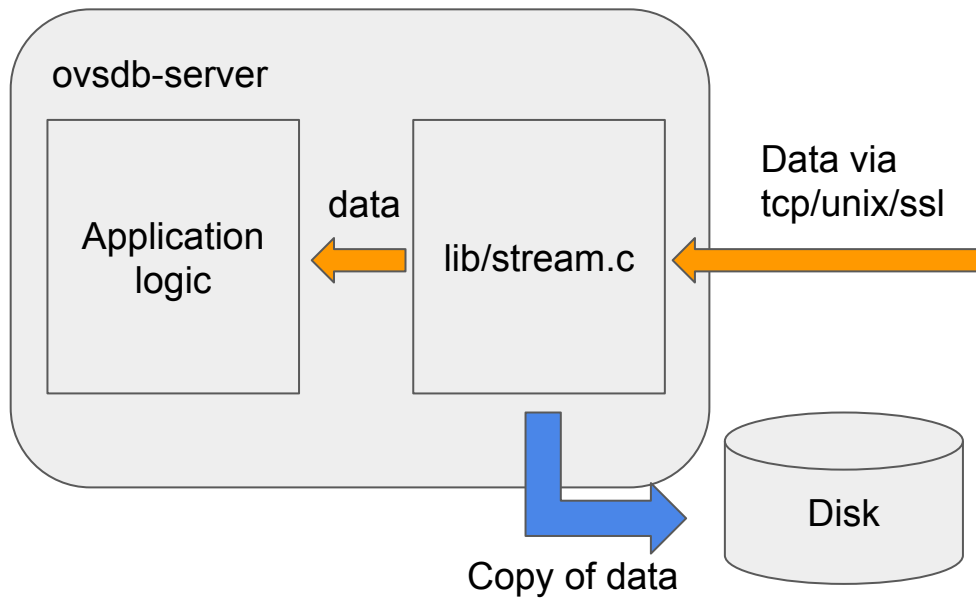
stream-record/replay

- All connections in OVS/OVN applications works via 'stream' library.
 - lib/stream.c
- All variants of stream connections (ssl,unix, tcp, ...) implements same stream-provider API, so they could be transparently replaced with special implementation.
 - lib/stream-provider.h

stream-record/replay

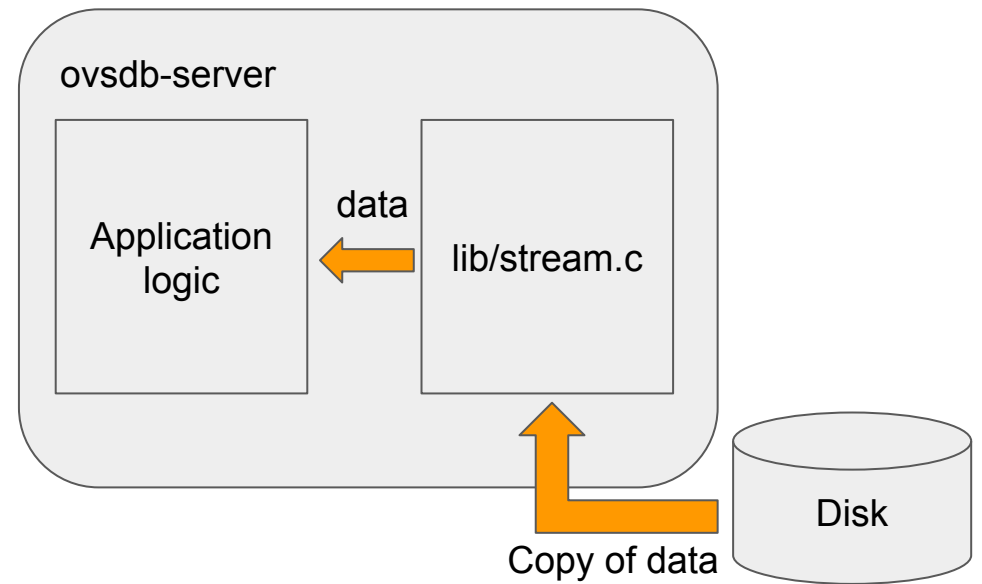
Recording

```
# ovsdb-server --stream-replay-record <...>
```



Replay

```
# ovsdb-server --stream-replay <...>
```



What is available now?

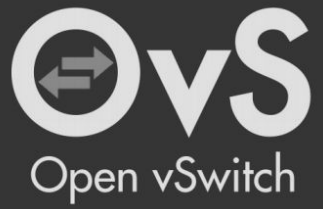
- Patch-set with the first implementation:
 - https://patchwork.ozlabs.org/project/openvswitch/list/?series=186549&state=*
 - Work on v2 is in progress.
- Current limitations:
 - For now it's required to use predictable uuid generation with '--predictable-uuids-with-seed=<seed>' option.
 - Currently doesn't work with internally generated time-based events.
 - Some functionality, like inactivity probes, disabled.
 - Works for ovssdb-server only in standalone mode (no RAFT).

Use cases

- Offline debugging: record on real setup - replay and debug locally
 - During replay it's possible to attach debugger, change log levels or even modify the binary adding some new logs or instrumentation.
- Standalone performance test.
 - If you're working on pure performance optimizations that doesn't affect in/out traffic from the process.
 - Record real world workload and replay without any latencies between messages.
- Offline debug of rarely reproducible issues:
 - Record once and have it 100% reproducible.

Future improvements

- Recording of generated UUIDs.
 - This way we could avoid enabling generation of predictable UUIDs.
- Correct work with internal time-based events.
 - There are few ideas on how to do that (record and replay accesses to time functions or time wrapping according to timestamps of stored data)
 - Will allow recording of RAFT clusters.



Thank you!

Ilya Maximets
Senior Software Engineer