Colocation in Stateless Network Functions

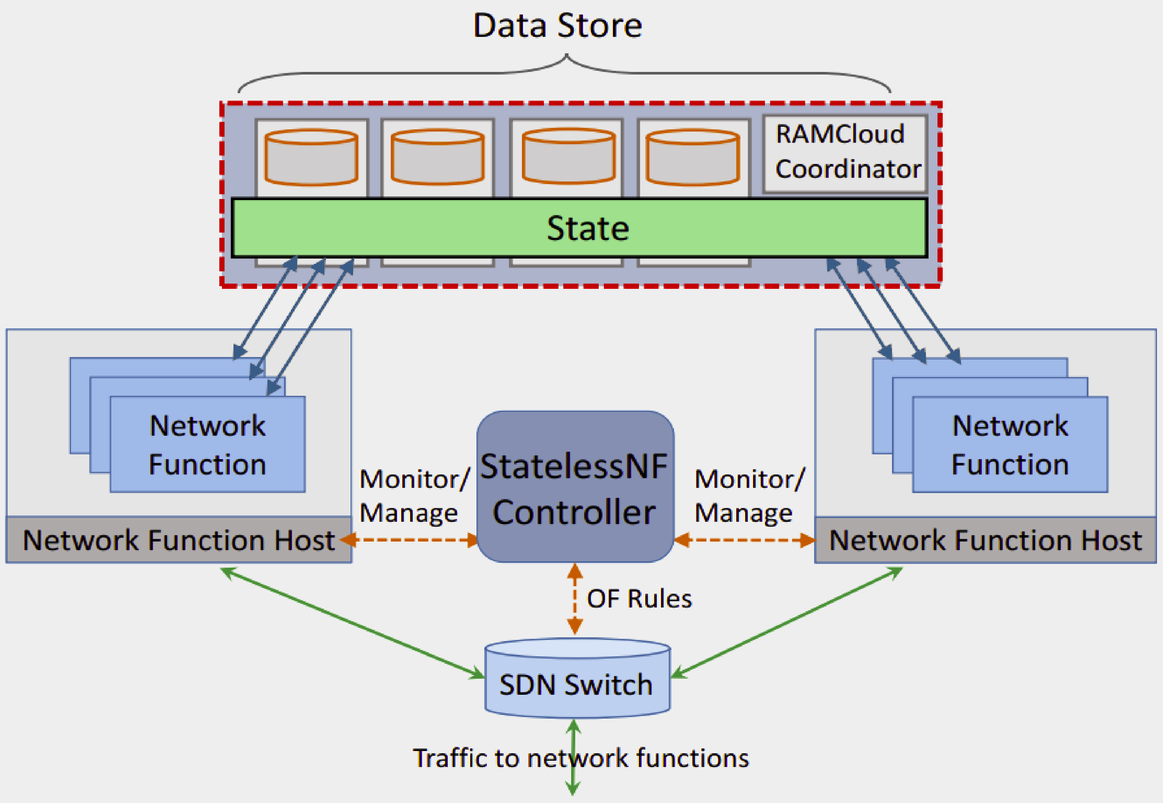
Anurag Dubey, Murad Kablan, Eric Keller

University of Colorado Boulder

Benefits of Decoupling State From Processing in Network Functions

- ▸ Dynamic network state persists across failures.
- ▸ State instantly available to Network Function (NF) instances upon scaling in or out.
- ▸ Built-in support for asymmetric and multi-path routing.
- ▸ Low packet processing latency.

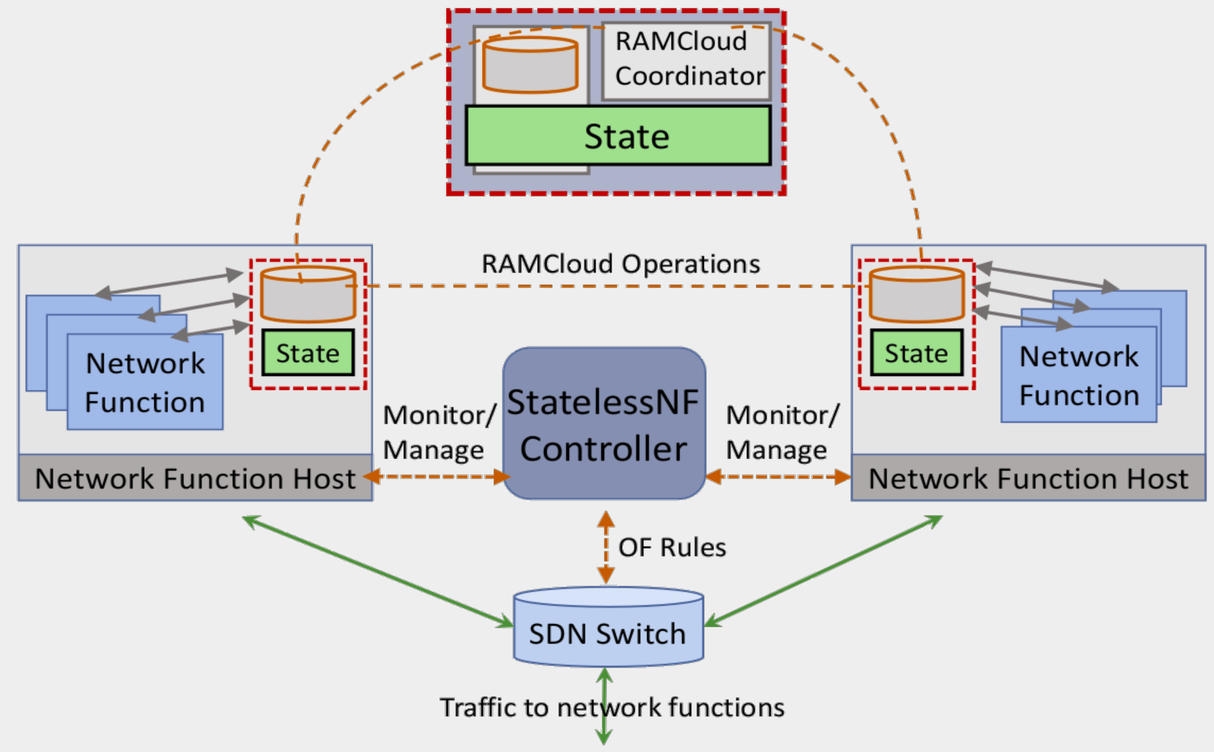
Current Stateless Network Functions Architecture



- ▸ Data store provides low latency access to dynamic state.
- ▸ Network functions run in containers or commodity servers.
- ▸ Controller acts as an an orchestration component.
- ▸ Challenges in this StatelessNF design:
  - Added Latency
  - Affect throughput with frequent read/writes

Kablan, et. al., “Stateless Network Functions: Breaking the Tight Coupling of State and Processing”, NSDI 2017

Goal: Optmimized Data Store Node Placement



- ▸ Keep frequently accessed data closer to NFs.
- ▸ Maintain logical disaggregation of data.
- ▸ Do not rely on caching (avoid coherency problems).
- ▸ Data access through nodes actually storing that data.
- ▸ Coherent data replication among nodes.

Challenge: Data Placement

Problem:

Need a way to store dynamic state so that every node has fast access any object while still benefiting from data colocation.

Present Approach:

- ▸ Use packet header as key and store state in Value.
- ▸ Hash based sharding of data.
- ▸ Compute key hash to discover slot and thus the node a specific key belongs to.
- ▸ Scaling involves movement of hash slots among nodes.
- ▸ Very efficient for random data placement but does not allow optimization of data locations.

Proposed Approach : Directory Data Store

NF nodes create and store state on the local instance of data store.

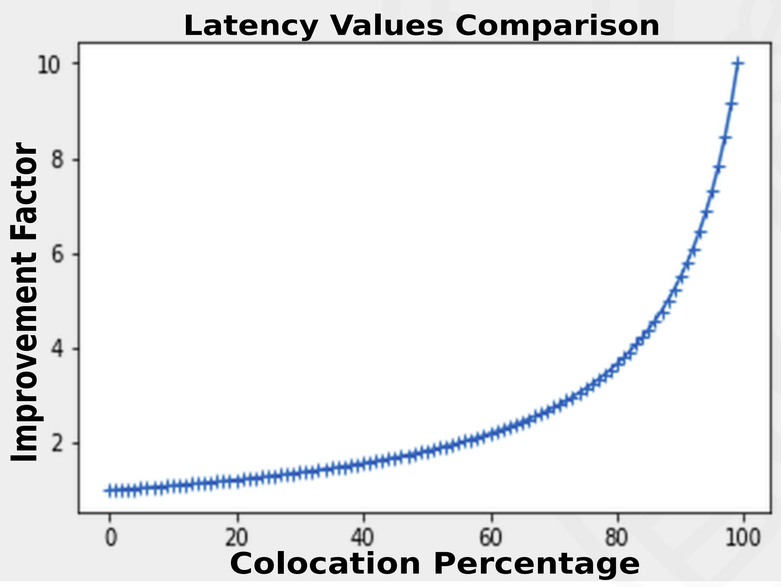
Data is replicated for high availability.

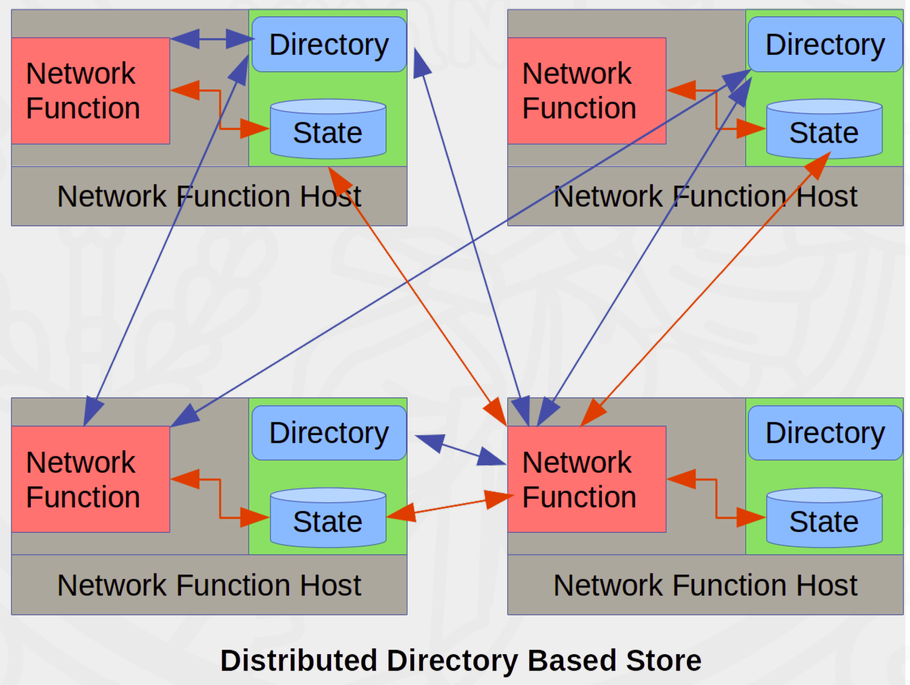
Hash Slots act like directories and store pointers to actual data node(s).

Access remote data though pointers.

Cache directory entries.

Latency Improvement Factor =
(Remote access latency)/
(Directory based access latency)





Distributed Directory Based Store