Clustere Based (Without stateless or shared memory approach)

Each NF in a cluster shares states among clusters after each batch ??

meaning ??

Scaling

(i) যদি প্রথমে NF Cluster-এর বাকি NF-এর সাথে consistency maintain করে after processing a batch then There will be heige latency related problem I guess.

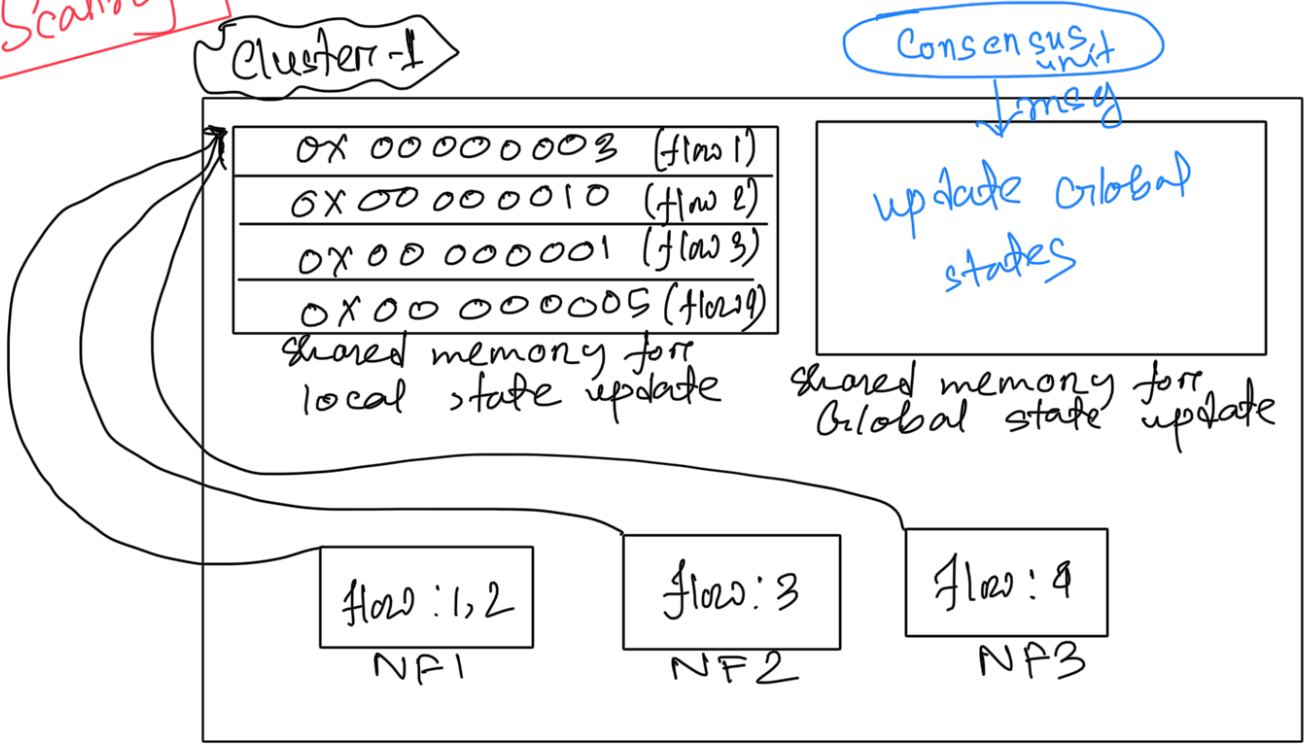
# of NF → x

#of 2PC → $x P_2$ after processing x batch

(ii) আবার প্রত্যেক batch-এর পর পর সব NF-এর state-এর image টি identical. তাহলে scaling করলে state migration করা লাগবেনা বা explicitly. → that's own good but identical বানাতে আবারও অনেক latency create হচ্ছে।

Cluster based (shared memory / stateless inside a cluster approach)

Scaling



(i) সব NF এর জন্য shared memory-এর starting address থাকবে,

(ii) So, কোনো particular flow-এর state update এর জন্য getAddress (flowID) দিলে address বের করবে.

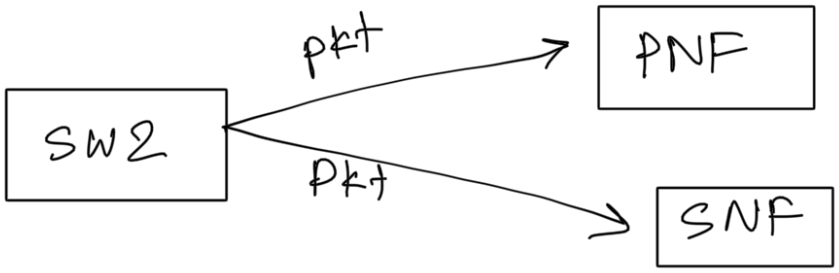
(iii) So scaling এর জন্য আমরা

(i) SDN controller-এর rule change করব.

(ii) আমরা কিছু time-এর জন্য buffered pkt আগে NF এ পাঠিয়ে দিব,

Failover

# of NF in a cluster even



(i) after processing each batch

(ii) SNF → (i) Next expected ID এর আগের pkt সবগুলো discard করে দিবে।

(ii) If FDV detects failure of NF, then SNF becomes new PNF and new instance of NF is created (SNF)