Guided by NFOS profiler/recipes, developers can productively improve NF throughput by 2 – 91x.

Developing scalable NFs is hard and error-prone

- Writing concurrent code is error-prone
- Finding the root causes of scalability bottlenecks is hard
- Fixing bottlenecks requires rewriting concurrent code

We found that 28 concurrency bugs existed in Cisco VPP NFs.
Semantic gaps exist between low-level profiling events and NF application-level behaviors.
This may introduce new concurrency bugs or scalability issues.

Insight:
Writing single-threaded code is much less error-prone.
A smart runtime can scale single-threaded code as needed.

NFOS: Transparent scaling of single-threaded NF to multicore

- Write single-threaded NF code
- Identify root causes of scalability bottlenecks in NF application logic
- Remove scalability bottlenecks by changing the single-threaded code

Programming model:
- Specify “packet set” and NF state local to it.
- Access “global state” shared by packet sets through the NFOS built-in data structures.

Scalability profiler:
- Report conflict causes, which encode the reasons why transactions abort in the built-in data structures.

Scalability-enhancing recipes
- Recipes show how to reduce the number of transaction aborts from the identified conflict causes.

Process packets in transactions to exploit fine-grained parallelism.
Encapsulate common concurrency wisdom in the built-in data structures.

Conflict causes bridge the semantic gaps between the low-level aborts and high-level NF behaviors.

NFOS-based NFs (NAT, Bridge, Load balancer) achieve 0.75–2.5x throughput of the hand-parallelized Cisco VPP NFs.

Insight:
NFs are essential building blocks of today’s Internet.

NAT
Firewall
Load balancer

Diyu Zhou
Sanidhya Kashyap
George Candea
Yueyang Pan
Lei Yan