WARF Accelerator Program Proposal - 09/01/2015 – 08/31/2015

Improving Performance, Availability and Usability in OpenNF

PI: Srinivasa Aditya Akella

WARF IP & Status

Filed. WARF Case number P150073. Status: pending

Introduction

Give a brief introduction of your project, summarizing the key elements of the research, the field in which it is relevant, and why this is an important area for WARF to consider funding.

Network functions (NFs), also called middleboxes, are in-network devices that perform custom packet processing actions. Examples include firewalls, application accelerators, gateways, intrusion detections systems, load balancers, etc. In recent years, two trends have significantly reshaped how networks use NFs. First, Network Functions Virtualization (NFV), where NFs are deployed within VMs or as software processes (as opposed to the traditional hardware-based appliances), has made it remarkably simple to instantiate or tear down NFs within networks. In addition, software defined networking (SDN) is being employed to flexibly steer traffic through different NF instances to realize a variety of interesting objectives.

The confluence of NFV and SDN opens up the door to two exciting sets of scenarios, namely, distributed NF processing and NF service chaining. Unfortunately, key attributes of NFs---namely their stateful nature, and the fact that they mangle traffic passing through them---impede our ability to realize the full potential of distributed processing and service chaining. Our work to date has been on developing novel abstractions centered around the “OpenNF” system to enable such scenarios. To date, we have focused on ensuring that dynamic reallocation of processing across distributed NF instances can happen in a safe, fast and cost-effective manner. We have show how to elastically scale NFs, or balance load across them in a rapid fashion, while ensuring that traffic allocation actions don’t result in the NFs taking incorrect actions.

In the future, we hope to address three key outstanding problems pertaining to OpenNF:

- Ensure that OpenNF’s correctness guarantees extend to dynamic reallocation across different kinds of NFs that are “chained” together
- Ensuring that OpenNF’s overall performance can match the high line rates that are beginning to emerge in modern networks today (e.g., 10 and 40Gbps).
- Ensuring the OpenNF’s control plane is highly available, and that availability does not come at a significant performance cost
- Building an interface to enable operates to easily implement and realize orchestration actions

As the industry is increasingly espousing NFV, they are starting to encounter correctness problems when performing state/traffic reallocation. The “solution” for this today is to only reallocate new flows and let existing traffic flows stay put. Unfortunately, this is quite inflexible as it can lead to either resources being wasted, or reallocation not having the desired effect and resulting in poor overall performance. As traffic volumes grow, as performance becomes central, as reliance on strong SLAs for outsourced services grows, and as the need to keep infrastructure costs low becomes central, we strongly believe that systems that provide high performance, correct, and cost-efficient behavior for a range of NF deployments are going to be necessary if not fundamental. We believe that such systems will be adopted in the next 4-5 year time horizon.

Project Background

Provide an expanded view of the project. Describe in concise terms what research has been conducted to date, the results produced, the future goals of the research project, and what end result is expected to be delivered from the requested research funding. This section should also describe a clear delineation between hardware and software (if relevant) that needs to be developed as part of the project and clearly reflect the total scope of the project.

Network functions virtualization (NFV) together with software defined networking (SDN) has the potential to help operators satisfy tight service level agreements, accurately monitor and manipulate network traffic, and minimize operating expenses. However, in scenarios that require packet processing to be redistributed across a collection of network function (NF) instances, simultaneously achieving all three goals requires a framework that provides efficient, coordinated control of both internal NF state (which is complex and dynamically updated) and network forwarding state. To achieve such control and enable operators, we design a control plane called OpenNF. OpenNF employs carefully designed APIs and a clever combination of events and forwarding updates to address race conditions, bound overhead, and accommodate a variety of NFs.

Our work to date has been on developing the algorithms for loss-free and order-preserving moves of state across network function instances, and developing a proof-of-concept system that implements these algorithms and allows a small number of control applications to be written (examples include elastic scaling, load balancing and high availability). Our work shows that loss free and order preserving state transfers can be completed in roughly 200-400ms for flows correspond to 500-1000 traffic flows for the Bro Intrusion detection system (IDS). We also observed similar results for other NFs, including load balancers and NAT devices. We found that scale up and scale down of the Bro IDS can happen within 200ms. The workloads we employed had a traffic load of 10K packets/s.

The above work essentially shows that our proposed architecture for dynamic redistribution across instances of an NF offers the promised guarantees. However, it suffers from several drawbacks:

- Our current work reallocates flows across multiple instances of a single NF. However, NFV, especially in conjunctions with emerging solutions for traffic steering (including the idea proposed in our Stratos work), enables chaining of different kinds of NF. In such situations, reallocation of traffic and state at one of the NFs (and its replicas) in a chain in isolation may lead to crucial issues. To be more precise, chain-wide reallocation of processing across different NF instances for different NF types causes an NF deployment to transition between “states” where each state reflects how/what traffic is processed by a certain instance of an NF. However, transitioning between chain states is not instantaneous, due to delays in how traffic/state is reallocated across different NFs. Thus, there could be intermediate time instances where NF instances can be overloaded, or the intermittent network links can be congested. Further, traffic that is redirect to aid in the state move operation can add additional load to both network links and compute resource. If not orchestrated carefully such time periods of contention can last for several minutes! Our first goal is to design and implement mechanisms for chain-wide congestion free “state transitions”.
- Our second key goal is to improve the performance of both the existing OpenNF architecture and the extensions proposed above. In recent months, we developed a mechanism that enables peer-to-peer transfers of state across NF instances. Further, we developed the notion of packet

reprocessing. Together, these enhancements ensure that the OpenNF controller is not bottlenecked due to excessive involvement in state transfers, buffer overflow issues at the controller are minimized, packet processing latency is eliminated as packets are immediately processed, and it is possible to recover from buffer overflow issues at the NF instances. However, these solutions, like the original OpenNF system were designed for a chain with multiple instances of a single type of NF. Our next goal therefore is to extend these solutions with the aforementioned ideas for cross-chain safety and further to implement and test those solutions.

- Our third goal is to improve the availability of the OpenNF controller. In our current implementation, the controller is a single point of failure. It is possible to use standard mechanisms to replicate the controller and improve its availability. However, naively adopting existing replication mechanisms can impact the systems overall performance because state transfer may be impacted by the overhead of keeping controller replicas in synch with reach other cause state transfers to take an undue amount of time. Thus, our third goal is to implement techniques that re-engineer existing availability and replication mechanisms to be more aligned with the OpenNF architecture and its other goals in particular those relating to high performance.
- Our final goal is to build a simple high-level interface that allows operators to specify network NF deployment chains, and annotate individual NFs and the chain as a whole with safety properties and possibly specific locations to deploy them at. It would further allow operators to specify the kind of performance bounds the operator wishes to provide for the chain. The underlying backend translates these high level specifications into low level commands that: (a) deploy NF VMs, possibly at specific network locations, (b) chain then as specific by the operator, (c) monitor load and redistribute processing to meet the operators stated performance goals.

Commercialization opportunity

Describe your view of the market opportunity for the technology.

As mentioned earlier, we believe that this problem will become very real in the next few years. We already have a head start with OpenNF, and further improving our systems along the lines of the ideas outlined above, would give us a strong head-start over the best state-of-the-art in the industry. We have received initial interest from BigSwitch. They indicated to us that this is a problem that several vendors they partner with had indicated the correctness problems in reallocation and chaining, but that developing solutions---while it is destined to happen---is still 1-2 years out. In all, we believe that developing OpenNF along the list listed above over the next year or so would make the technology very compelling to be licensed by a company such as BigSwitch. That said, commercializing OpenNF independently is also a possibility and we believe it could lead to much more exciting outcomes given the significant amount of buzz and exciting surround NFV (which far exceeds the buzz surrounding SDN, even!)

Please don't hesitate to contact us should you need more information.

Milestones and financial summary

Describe the key milestones in the research project with corresponding dates and the level of funding and manpower needed to achieve those milestones.

The four goals listed above outline a significant amount of work that we ***ideally*** hope to achieve during the course of 12 months. We realize that due various issues such as being able to recruit personnel in a timely fashion, and bugs in software, our timeline may be dilated by a few months. We anticipate this dilation to be no more than 3 months.

Given our aggressive timeline, we hope to recruit three full time (50%) graduate students to realize the above goals.

All our testing and deployment will happen on CloudLab.

We have four key milestones corresponding to the four tasks listed above. We anticipate the following timeline:

Task 1 □ Months 1—4
Task 2 □ Months 4—7
Task 3 □ Months 7—10
Task 4 □ Months 10—12

By the end of month 12, we hope that we will have a fully implemented/tested prototype ***AND*** a polished demo to present to interested parties.

With half the funding, we can finish the project in 21 months, instead of 12 months. Furthermore, we anticipate that the quality of the final prototype/demo will be inferior. [This is because creating effective demos and user interfaces requires reasonably sized teams where team members focused on implementation work hand in hand with those focused on user interface/demo. My experience has been that two students cannot pull off a polished demo ***AND*** a full implementation on their own even given additional time. Two students dedicated to implementation and one to demo is required for that.]

Summary

Include a closing statement about the relevance of you research for UW. This is your opportunity to sell your idea.

Network Functions Virtualization is rapidly transforming both the cloud and service provider industries. It promises to present vast opportunities to offer interesting new services and vastly more flexible implementations of existing services. However to realize this promise effectively, we need systems that can offer provable correctness and performance properties, while ensuring low infrastructure costs. The proposed project leverages the unique research and infrastructure expertise at UW-Madison on NFV, SDN, and cloud technologies to develop a system that addresses this challenge comprehensively. Thus, the project, if successful, will put UW-Madison, its faculty and students, on the world’s radar as being the leaders on NFV research and development.