Overview

OpenNetVM (ONVM) is a highly efficient network packet processing framework that provides simple abstraction for developing and running network functions. The ONVM platform provides load balancing, flexible packet flow management, individualized service abstractions, and basic software defined networking (SDN) capabilities. The platform already offers best-in-class performance: it has the ability to maintain 40 Gbps speeds while routing packets through dynamically created chains of network functions (NFs). OpenNetVM also lowers the barriers to deploying production network functions in software because it can run on inexpensive commodity hardware. However, this idea can be improved upon: we aim to make it possibly for multiple instances of ONVM to process packets in tandem, with seamless inter-routing, making it possible to run software-based network services at production scale.

Subtopic: Network Function Virtualization
Key Words: networking, software defined networking, packet processing, the cloud

Intellectual Merit

This Small Business Innovation Research Phase I project will push the limits of the performance possible by commodity servers. OpenNetVM is already a highly optimized platform, and any extensions will need to maintain existing performance standards. One technical challenge will be implementing an intelligent algorithm to discover the optimal node to route a packet that can not be processed locally. The system will need to scale to an arbitrary number of instances of ONVM that can collaborate to process packets in the most efficient way possible, especially if there are multiple possible destinations for a packet. The system needs to keep management overhead minimal and ensure the NF critical path is as fast as possible, because excessive network traffic and computation overhead will degrade the capabilities of the “production” network.

In order to reach these goals, the existing platform needs to be modified slightly to support distributed capabilities. A new abstraction layer needs to be added, so the system can process packets going to and from either local or remote instances of ONVM. This abstraction layer should be generic enough that the two critical paths share as much code as possible and remain highly optimized. Then, a routing protocol that meets the above considerations needs to be designed and implemented. This will necessitate the inclusion of a distributed datastore so that the nodes can share data; the cost of additional overhead will be a major performance consideration, since the platform can not afford constantly doing slow data lookups.

Broader/Commercial Impact

This research will drive costs for enterprises building production-scale network services. OpenNetVM can run on cheap commodity hardware, reducing the need to order specialized appliances. For example, using a specially crafted NF running in OpenNetVM, a company can analyze all packets flowing into and out of their network and transparently filter and quarantine malicious traffic without performance degradation. Since the NF is running software, the

company does not need to purchase dedicated Intrusion Detection System (IDS) hardware, and malware signatures can be rapidly iterated. Additionally, running all NFs on standard hardware allows users to more efficiently allocate their hardware capacity, all while spending less capital on servers. This project can revolutionize the NFV space, bringing high performance scalable networking applications to enterprise consumers at a significantly lower cost.

Elevator Pitch

OpenNetVM is is intended to benefit enterprise customers. These companies run large-scale, high-performance networking applications. These network applications tend to require expensive specialized hardware that can only be purchased from one of few suppliers. For example, a large enterprise typically has an Intrusion Detection System (IDS) protecting the entry point to their network. A popular website or an entry point to the corporate VPN may saturate a 10 Gigabit connection with incoming requests. All of this traffic needs to be screened for malicious traffic. Furthermore, the enterprise may also require a firewall, DDOS detection system, and dynamic routing (like Software Defined Networking); all of these features require different expensive and specialized hardware. So, when building an enterprise-quality network, the infrastructure costs will add up and can quickly become burdensome.

OpenNetVM will allow large enterprises to run their high-performance network applications at a fraction of the cost. OpenNetVM can run on commodity hardware which vastly reduces the amount of capital required to spin up a corporate network. This platform will not sacrifice performance either; packets can be processed at 10 Gigabit speeds, the same speeds as the more expensive dedicated hardware. By utilizing the flexibility of software, OpenNetVM also integrates dynamic network topology. Packets can be routed via a “service chain,” which means a packet flow can be processed by multiple network functions (NFs). This functionality allows the functionality of what would previously require multiple dedicated appliances to be consolidated into a smaller number of commodity servers. Furthermore, as system requirements change over time, the topology of the network function interconnections can be dynamically modified. For example, if the company’s website is experiencing additional load, more instances of a load balancer can be started to accommodate the additional traffic and spread the incoming requests appropriately.

OpenNetVM is able to provide these revolutionary features by implementing Network Function Virtualization (NFV). For every “appliance” that an enterprise IT department would want on their network, they can abstract the functionality into software and run it on the OpenNetVM platform. OpenNetVM uses the high-performance DPDK networking library to interface directly with the hardware. Packets can be shared between NFs and processed without copying overhead – the platform works entirely with shared memory. Furthermore, the OpenNetVM platform will be able

to function in a distributed manner: packet flows can be seamlessly routed among multiple hosts to be processed in an optimal way. This means the system can scale seamlessly and automatically recover from failed hosts without loss in service. Automatic failover and horizontal scaling provide features that enterprise customers can rely on in production operations. Traditionally, these features were very expensive to implement, as they require specialized networking hardware. OpenNetVM can provide these advantages for free on inexpensive servers by harnessing the power and flexibility of software.

Commercial Opportunity

OpenNetVM is targeted towards enterprise consumers for use in their data centers. These companies often serve public facing services, internal applications, and other networking infrastructure from their own data centers. These services traditionally need specialized hardware to run and additional hardware to properly secure. Networking hardware is expensive to purchase and difficult to configure and maintain. OpenNetVM aims to alleviate these pain points by utilizing the flexibility of software to dynamically create performant and secure networks for enterprise applications.

OpenNetVM is an open platform - the software is open source and can run on commodity hardware. This is critical for growing a user base as there are few barriers for a potential customer to try the system. The first step in growing the platform is to get users, and an open platform that can be cheap, fast, and secure is a great value proposition. However, enterprise customers demand stability and reliability: we need to offer support, and this is our route to monetization. We will model ourselves after RedHat, a company that has had great success monetizing free software by providing support to enterprise customers.

Enterprise support contracts involve large dollar amounts, or enough to sustain our business. Our goal is for OpenNetVM to be easily evaluated (since it is a free and open platform) and then the customer is impressed at the savings due to the reduced hardware requirements, so they purchase an annual support plan to keep the system maintained. This approach will still end up being cheaper up front to the customer because networking hardware is incredibly expensive. Furthermore, OpenNetVM will provide immense future flexibility: the company easily add services to their network and dynamically scale as usage grows, all for only the cost of commodity hardware.

It is hard to overstate the value of quality support to an enterprise customer. Companies will pay for the privilege of reliability, especially when their revenue-drivers are on the line. We will provide support for outages and help troubleshoot failures and bring the system back online. We will also provide R&D support for a company evaluating our project and assist them while they

are exploring OpenNetVM. If they have any questions during this process, we'd be able to give them priority support in answering their questions.

Another means of revenue for OpenNetVM is prioritized feature development. If someone is interested in seeing a specific feature get developed, they can either pay to prioritize it or wait for it to get accomplished from the open-source roadmap. If someone is prioritizing a feature, they will pay an hourly "consulting" rate for development efforts related to implementing that feature. This is valuable for companies that have a slightly specialized use case for OpenNetVM. If they need a certain change made and can't wait (since it would affect their business operations), they can pay to have it written faster, generating supplemental revenue for the project.

There are few competitors in the NFV (Network Function Virtualization) space. The closest competitor is ClickOS (http://www.read.cs.ucla.edu/click/). ClickOS is a "modular router" that can dynamically route packets. Both products are already currently in the market in beta form as research projects, but ClickOS does not appear to support a "clustering" mode, limiting its scalability potential. OpenNetVM's clustering capabilities allow it to grow to infinite scale and process arbitrarily large packet streams, making it a better candidate for running enterprise services (which need to run at incredible scale).

Societal Impact

OpenNetVM is a platform for other people to build products, therefore, there is not a lot of direct societal impact. However, it is a platform other companies build upon to make the world a better place. The cloud has played a major part in making technology more accessible to the world, and wide availability of the OpenNetVM platform will continue this trend. Recently, the barrier to entry in the tech space has been drastically reduced by the commoditization of cloud service infrastructure (it is easier than ever to create an app or website and bring your idea to the world). OpenNetVM takes that concept and brings it to networking infrastructure. Ideally, other companies can build amazing services on the OpenNetVM platform and positively impact society.

There are not any real regulatory issues surrounding this project. It is an open platform, meaning that users can do as they wish. OpenNetVM is released under a BSD License, which allows the source code to be redistributed without repercussion (provided the license is maintained) and is provided "as is" without any explicit or implicit warranties. Therefore, users can do whatever they like with OpenNetVM, and that is perfectly fine. Ideally, the project would only be used for "good" purposes, but that can not be controlled under the terms of the project

license. We do not anticipate any malicious or unethical uses of the project, but seeing as it is an open platform, there is a very little amount of control we can exert on it.

Any group of people can be impacted by this project. A new startup would be wise to use this platform to host their services. They can leverage OpenNetVM's scaling feature and not have to worry about infrastructure (and instead focus on building the best application possible). The startup can also use security-focused NFs, like an IDS (Intrusion Detection System) or a Firewall to ensure the integrity of their application. And when this startup needs to change their configuration, they don't have to wait on any lengthy hardware orders, since they can simply reconfigure their network on the fly due to the flexibility of software-based networking. Finally, as the startup grows, they can leverage OpenNetVM's native load balancing capabilities to make sure they don't exhaust the capabilities of their hardware and continue providing valuable services to their customers, even in the face of heavy loads. All of this can be accomplished using only commodity hardware in the data center. Not being reliant on hardware orders allows the company to not have to worry about their networking infrastructure and focus on the parts that matter: building great services to make the world a better place.