Platform for Evaluating Real-Time Resource Management Algorithms for Network Function Virtualization

Dept. of CIS - Senior Design 2014-2015[*†]

Alex Brashear
brashear@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Alex Brashear
brashear@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Alex B
brashear@s
Univ. of P
Philade

ABSTRACT

Network Functions Virtualization (NFV) is an approach to implementing network infrastructure that emerged in 2012. NFV aims to address the inflexibility of classical network hardware by leveraging virtualization technology to consolidate network equipment onto industry standard servers. To date, several reports have demonstrated the feasibility of virtualizing network functions. In order for NFV to succeed in practice, virtualized network systems must account for the unique real-time requirements of network functions when approaching such problems as resource management. To that end, Professor Linh Phan and her research group at Penn have developed algorithms to manage resources in the context of NFV. One use case of interest is to provide network services that meet functional requirements, which are typically end-to-end latency requirements. In this project, we propose to design, implement, and evaluate a testing platform that can simulate real-time scheduling of virtual machines as well as functional and nonfunctional services in order to evaluate the effectiveness of the algorithm. We will collect metrics to understand, validate, and later improve the real-time scheduling algorithms. On the hardware side, we will deploy the platform on four 32-core computers with network switches connecting them. The testing platform will consist of an orchestration layer that uses their algorithms to manage resources and a cluster of computers that provide computing and networking resources. This system will enable the evaluation of the algorithms and aid in the development of future algorithms for efficiently implementing network functions.

1. INTRODUCTION

A vast number of hardware equipments (known as middleboxes) provide important services for the network such as mediating data packets being sent to and from one network node to another. These critical services filter or manipulate packets in order to improve efficiency and maintain security. For instance, a firewall is one such example of a middlebox that filters out malicious or irrelevant traffic. However, network hardware poses a growing problem because it is difficult to install and maintain. Furthermore, middleboxes burden enterprises with financial and administrative cost [white paper]. In the same way that cloud computing freed service providers from the management of physical hardware, Network Functions Virtualization (NFV) has been proposed as a solution to move network hardware functions to the cloud, where hardware services will be managed by software. Through NFV, it is much easier to install, manage, and upgrade hardware services [white paper].

Now, suppose a service provider wants to virtualize hardware network functions in order to take advantages of NFV such as economic savings, automation, and scalability. To satisfy its customer, the service provider has to finish processing requests with a reasonable latency performance. If a customer imposes a certain latency requirement for their service request, how can the service provider schedule services in the cloud in order to meet this criteria? Although research has shown that it can be beneficial to move network hardware to the cloud, few studies have been conducted to suggest ways to optimally manage resources for network functions in the cloud in real-time systems.

Recently, Dr. Phan at the University of Pennsylvania has developed an algorithm that can schedule and manage NFV services in the cloud in real-time, meaning services will be effectively scheduled on virtual machines per each new customer requirement. The algorithmâĂŹs main goal is to create effective configuration of virtual machines in the cloud and schedule services in a way that minimizes latency in order the meet the latency requirement. At a high level, the algorithm uses a combination of linear programming and real time analysis to determine the assignment of services to virtual machines in the cloud. In middleboxes, a data packet coming from a customer will be processed by the first service, and then the output of that will be sent to the second service to be processed, then the third, and so on. When scheduling services in the cloud, it is optimal to place services within close proximity to one another in order to minimize the latency that will be incurred with the distance that packets have to travel from service to another, after being processed. Another important consideration that the algorithm looks at is effective ways to schedule CPU and network resources on the cloud. Both allocations of CPU and network resources occur in real time based on the latency requirement imposed by customers. The algorithm tries to ensure that the end-to-end latency, or the total latency acquired from processing a data packet through the chain of services from the virtual machines, is minimized.

Currently, no analysis platform exists to evaluate effec-

[*]Advisor: Insup Lee (lee@cis.upenn.edu).

[†]Do not list your advisors amongst the authors as that may cause Google Scholar to add this work to their list of publications. Your advisor must also sign a hard-copy of your proposal.

tive resource management algorithms for virtualized network functions on the cloud. As our research project, we propose a testing platform that can be used to analyze the ability of network service scheduling algorithms to meet customer latency requirements. Specifically, we propose to create software that simulate a service provider in the cloud handling customer requests. Through careful design of this software, it can be used to test and prove the correctness of real-time scheduling algorithms. On a broader standpoint, this software will ultimately help cloud service provider make real-time latency guarantees to customers. The specifics of this system is described below.

2. RELATED WORK

Perhaps the most important section of your proposal is *related work*. Here you demonstrate that you have read and understand what others in the field have done. This ensures you (1) know the state-of-the-art, (2) are not re-doing others work, and (3) you know the performance levels you must achieve to make a contribution. As you discuss each related work, make note of how each has advanced the field. Keep in mind that this section should not read like a regular research paper you write for other classes. In other words, you should not just discuss related work for the sake of having a related work section; rather, tell a story about the state-of-the-art of the field and where your work fits in.

This section should have in-line citations to your bibliography (really all sections should have citations, but we expect them to be most dense in this section). We are going to require that your proposal has at least 6 references. Fortunately, LaTeX makes citations easy. Your TA has had no difficulty, as the work of Ivanov *et al.* [**?**] demonstrates. Need help with LaTeX? Be sure to check out [**?**] and [**?**], two helpful on-line resources.

What defines a good resource? Wikipedia is **NOT** a good resource. We would like to see references from academic journals/conferences (ACM, IEEE, etc.). We realize not everyone is doing pure research and for students with 'implementation' projects such sources may be rare. No matter the case, your sources need to be reputable.

Let us return to your factorization proposal. You should put out the earliest related work; naïve methods like trial divison and the Sieve of Eratosthenes, but state they are of no modern relevance. Then discuss modern methods like the Quadratic Sieve and General Number Field Sieve. Note the humongous time and memory bounds of these algorithms. But wait! You are going to propose a better way . . .

3. PROJECT PROPOSAL

Now is the time to introduce your proposed project in all of its glory. Admittedly, this is not the easiest since you probably have not done much actual research yet. Even so, setting and realizing realistic research goals is an important skill. Begin by summarizing what you are going to do and the expected benefit it will bring.

3.1 Anticipated Approach

Having summarized *what* you are going to do, its time to describe *how* you plan to do it. Our factorization example does not work so well here (it is likely impossible to realize) – so let us suppose you are going to create a service that takes a cell-phone picture of a building and returns via text-message, the name of that building[1].

In this case you might want to talk about establishing a server to receive pictures via MMS. Once the picture is received, you will run an edge extraction algorithm over it. Then, similarity between the submitted picture and those stored (and tagged) in a MySQL database will be computing using algorithm XYZ. Finally, the tag of the most similar image will be returned to the user. Do not bore the reader with trivial details, but give them an overview; a block-flow diagram would prove helpful (and is required).

3.2 Technical Challenges

In this subsection note where you anticipate having <u>novel</u> difficulty. Maybe you have never setup a MySQL database or even used SQL before at all – yes, that is a challenge – but not one readers care about. More novel would be the fact that many buildings on Penn's campus look similar and your classifier may be inaccurate in such instances. The purpose of this section is two-fold: 1) you will think about which parts of your project would require the most time and effort and 2) you will convince the reader that this is a project worth undertaking.

3.3 Evaluation Criteria

Suppose you have implemented your approach and it is functioning. Now how are you going to convince readers your approach is better than what exists? In the factorization example, you could just compare run-times between algorithms run on the same input. The image recognition example might use a percentage of accurate classifications. Other fields may have established testing benchmarks.

No matter the case, you need to prove you have contributed to the field. This will be easier for some than others. In particular, those with 'sensory' projects involving visual or sonic elements need to think this point through – objective measures are always better than subjective ones.

4. RESEARCH TIMELINE

Finally, we would like you to speculate about the pace of your research progress. This section need not be lengthy, we would just like you to specify some milestones so we can gauge your progress during our intermediate interviews. Let us follow through with our image recognition example:

- ALREADY COMPLETED: Preliminary reading. Began implementation of image-recognition algorithm.
- PRIOR-TO THANKSGIVING : Photograph buildings for DB. Make algorithm more efficient, tune parameters.
- PRIOR-TO CHRISTMAS : Create server-MMS interface. Expand tagged DB collection.
- COMPLETION TASKS : Verify implementation is bug-free. Conduct accuracy testing. Complete write-up.
- IF THERE'S TIME : Investigate image pre-processing techniques to improve accuracy.

[1]Do not use this idea – someone did it in a previous year.

APPENDIX

A. OTHER SPECIFICS

Your proposal need not have appendices like this section and the next but we still have info to share:

1. PROPOSAL LENGTH: We require that your proposal be 4–5 pages in length, bibliography included. Be careful, LaTeX and our style-file in particular are *extremely* space efficient. An 9-page MS-Word document could easily become a 5-page LaTeX one.
2. PLAGARISM: **DO NOT** plagarize. If you are caught, you will fail the class (*i.e.*, not graduate), or worse.

B. LaTeX EXAMPLES

At this point, the proposal specification is complete. From here on out, we are just going to show off some commonly used LaTeX technique. Be sure to look at the 'code behind' and see Tab. 1, Eqn. 1 and Fig. 1 for the output! Keep in mind that the appendix is usually not a good place for your figures. Place them where you need them and remember to refer to them in the body of your text; otherwise, the reader will keep reading and will miss them!

$$M(p) = \int_0^\infty (1 + \alpha x)^{-\gamma} x^{p-1} dx \tag{1}$$

User Type	Cleanup%	Honesty%
Good	90-100%	100%
Purely Malicous	0-10%	0%
Malicious Provider	0-10%	100%
Feedback Malicous	90-100%	0%
Disguised Malicous	50-100%	50-100%
Sybil Attacker	0-10%	Irrelevant

Table 1: Example Table

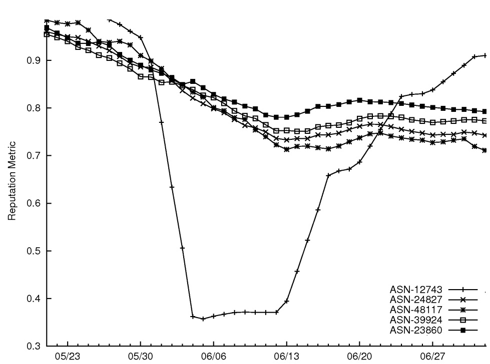

Figure 1: Example Figure/Graph