Towards a Scalability and Energy Efficiency Benchmark for VNF

Norbert Schmitt[1], Jóakim v. Kistowski[1], and Samuel Kounev[1]

University of Würzburg, 97074 Würzburg, Germany
{norbert.schmitt, joakim.kistowski, samuel.kounev}@uni-wuerzburg.de

Abstract. Network Function Virtualization (NFV) is the transfer of network functions from dedicated devices to high-volume commodity servers. It opens opportunities for flexibility and energy savings. Concrete insights on the flexibility of specific NFV environments require measurement methodologies and benchmarks. However, current benchmarks are not measuring the ability of a virtual network function (VNF) to scale either horizontally or vertically. We therefore envision a new benchmark that measures a VNF's ability to scale while evaluating its energy efficiency at the same time. Such a benchmark would enable the selection of a suitable VNF for changing demands, deployed at an existing or new resource landscape, while minimizing energy costs.

1 Introduction

Data centers in the United States consumed an estimate of 61 billion kWh annually in 2006, according to a Berkeley National Laboratory reported to congress [1]. By 2013, the energy consumption has risen to an estimated 93 billion kWh. National Resource Defense Council (NRDC) [2] projected the power consumption to climb to 140 billion kWh by 2020. Roughly 5-10% [1] of this consumed energy is used by networking equipment with its power demand expected to increase proportionally with the increasing server power demand.

The rise of cloud computing, enabling new products such as Software as a Service (SaaS), calls for increased flexibility in terms of service locality and network configuration abilities. The introduction of software defined networking (SDN) allows for greater flexibility in the network configuration. Yet the network infrastructure is mostly relying on dedicated appliances with limited flexibility in locality and scalability.

With growing data centers, the demand for performance in network equipment increases as well. Yet typical service demands are not constant over time but highly variable [3] and large amounts of resources remain unused when the system is not under peak load. Virtualization allows the on demand allocation of required resources to a certain task without a decrease in Quality of Service (QoS) or Quality of Experience (QoE). With the introduction of Network Function Virtualization (NFV) by the European Telecommunications Standards Institute (ETSI) [4], this trend towards virtualization is applied in the network domain by replacing dedicated appliances with high-volume commodity servers. NFVs based on commodity servers might not be more energy efficient than dedicated hardware devices when under peak load due to the optimized hardware within the dedicated network appliances. However, peak load only accounts for a fraction of the total time the service is available. Combined with the ability to scale both horizontally and vertically, NFV opens up opportunities for energy saving and reduced operational costs.

In this paper we describe our vision for scalability and energy efficiency benchmarking for virtual network functions (VNFs). Different techniques for auto-scaling in a cloud environment exist [5] today and research is still ongoing. The introduction of NFV also introduced the ability to scale network functions horizontally and vertically. This enables network functions to be used in auto-scaling scenarios in cloud environments. Yet, the differences in scalability of different or competing VNFs in an NFV environment remains unknown. Different implementations of an otherwise identical network function could behave differently when scaled. We therefore envision a new benchmark suite that rates a VNF's ability to scale horizontally and vertically.

While performance is a key characteristic, energy efficiency gains importance with the rising demand in flexible networking equipment. An energy efficiency aware benchmark could show opportunities for energy saving and subsequently reductions in operational costs. Our main goal for a new VNF benchmark is the rating of scalability, performance and energy efficiency of VNF implementations to select and deploy energy efficient VNFs without a decrease in QoS or QoE. Thus, we not only rate the performance of a VNF when scaled, but combine it with its energy efficiency for the performance demand.

The remainder of this paper is structured as follows: At first, we give an outline of the current state of the art. In Sec. 3, we formulate the problem statement of our envisioned scalability and energy efficiency benchmark. Section 4 describes our vision for a new benchmark followed by an approach to realize such a benchmark in Sec. 5. This includes preliminary methodology and setup of the benchmark. Finally, Sec. 6 provides a conclusion and an outlook for the next tasks towards our vision.

2 State of the Art

Huppler motivates the importance of efficiency benchmarks in his work [9] with many examples of benchmarks, including the Green500 ranking for supercomputers and TPC-Energy [26]. The latter is also the focus of [10], which introduces the new metric of energy proportionality. Energy proportionality is designed to represent a system's ability to adapt to changes in demand. This underlines the need for different load levels in energy efficiency benchmarking, also described in the SPEC Power Methodology [8] that is used for the Standard Performance Evaluation Corporation (SPEC) Server Efficiency Rating Tool (SERT) [28], ChauffeurWDK [29] and SPECpower_ssj2008 [27].

There is also a variety of existing virtualization benchmarks like the Standard Performance Evaluation Corporation (SPEC) VIRT_SC 2013 [23], TPC-VMS [24] and TPCx-V [25]. However, these benchmarks are measuring the performance of a workload together with the virtualization technique and software stack. In contrast, we intend for our benchmark to be independent of the virtualization technique, software stack and hardware, to increase its range of possible applications and making different VNF implementations directly comparable.

Lange et al. [14] also states VNFs are more regularly used in higher abstraction levels, especially when used inside cloud environments. In addition, complexity and concurrency increase as well, due to the abstraction and interactions with other network functions. Subsequently the complexity of performance benchmarks for VNFs will also rise in complexity. Yet, our focus is not on the sole performance of VNF. A methodology for performance benchmarking network devices was already published as RFC 2544 [16] in 1999 and extended by RFC 6201 [17] and RFC 6815 [18]. In [15], the authors analyzed the performance of a single VNF (virtual router) and identified four performance bottlenecks. The relevance of these bottlenecks for other VNF types is questionable as only a single type was evaluated.

The expired RFC draft [20] made an early effort towards a VNF performance benchmarking methodology. It is listing required documentation and reporting, such as CPUs, caches, storage system, hypervisor and others. It also categorizes benchmarks in a $3x3$ matrix for deployment, operation and de-activation of VNFs. A second, also expired, RFC draft [19] provides a testbed setup for VNF benchmarking. Yet, it also focuses only on performance.

In [6], Herbst et al. describe elasticity as the autonomic provisioning and deprovisioning of resources, such that the provided resources always match the demand as closely as possible. For a system to be elastic, it must be either horizontally or vertically scalable. A horizontally scalable system provisions and deprovisions more virtual or physical machines to a task to accommodate changes in resource demand. A vertically scalable system must be able to allocate more computing resources (i.e, CPU cores, memory size and network I/O) to an existing machine. An elastically managed system can be in three states, shown in Fig. 1. If the resource demand (red) is higher than the resources currently supplied (blue), the system is in an underprovisioned state U_n for the duration A_n. In case the resource supply is higher than the demand, the system is overprovisioned O_n for time B_n and has more resources than needed. If the system is neither overprovisioned nor underprovisioned, it is in an optimal state for a given demand.

3 Challenges

For a new scalability and energy efficiency benchmark, specifically built for VNF benchmarking, we identify four main challenges based on Sec. 2. While the performance of a VNF can also be dependend on its location, our benchmark should measure the performance of a VNF itself. Its score should not reflect the solution

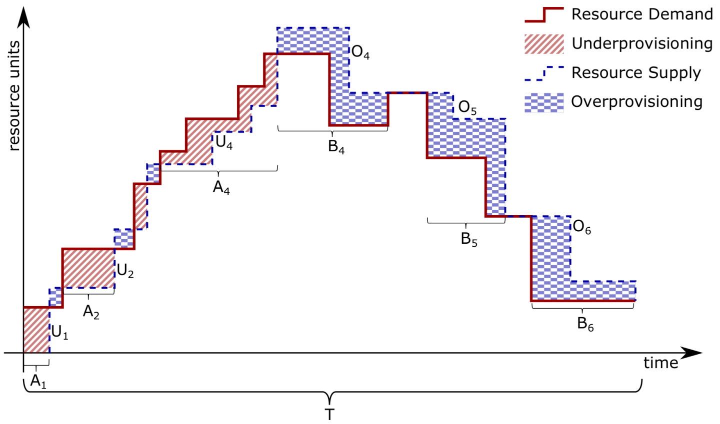

Fig. 1. Resource over- and underprovisioning [6]

to placement problems, such as the ones shown in [21] and [22]. The challenges we want to address stem mainly from the variety of application domains a VNF can be deployed in and from the ever increasing abstraction and complexity:

1. As mentioned, VNF implementations should be directly comparable. Therefore the performance of the VNF must be isolated from the underlying software stack, virtualization and hardware. This includes research on metrics that can represent a VNFs performance independently from these factors. Yet, it should be taken into consideration that full isolation might not be possible. In this case, a fixed reference virtualization technique could be selected to keep the benchmark's relevance and fairness.
2. We intend to empirically show the correctness of our benchmark. Therefore a selection of VNFs must be made that not only shows that the benchmark works but is also representative to a wide variety of possible VNFs under test. As VNFs can differ significantly depending on the domain, different evaluation groups could be formed. For example a Carrier Grade Network address translator (CGN) might be relevant to an Internet Service Provider (ISP) but less relevant to a video streaming service provider. Finding suitable VNFs is therefore necessary for empirical evaluation and can also aid in showing the benchmark's generality or limitations.
3. VNFs come in many types, all needing a special setup and configuration for traffic generation and validation. VNFs can have different numbers of sources (s) and receivers (r), as shown in Fig. 2. These range from simple configurations with a single source and receiver to VNFs that require multiple sources or receivers. A $n : m$ relationship between sources and receivers is also possible as a combination of the two rightmost examples in Fig. 2. VNFs could also alter the traffic flow in form of a firewall, which blocks packets, or a Network Address Translator (NAT), changing the packet header informa-

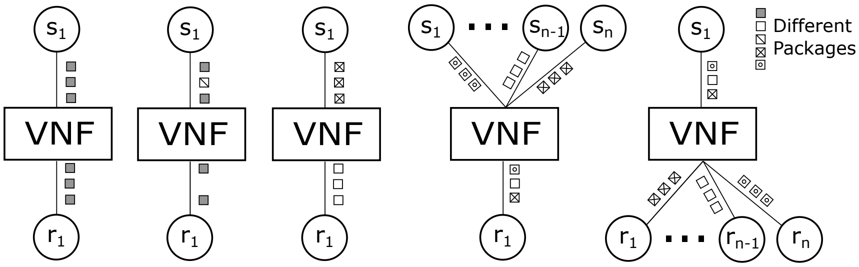

Fig. 2. Different VNFs with different needs for configuration and validation

tion. The list is not exhaustive nor final and a VNF might not exclusively belong to a single category. Yet, it shows the complexity in potential setups that must be handled by the benchmark.

4. Regarding vertical scaling, it is not known in advance which step size for resources should be used when scaling. For horizontal scaling, it is a question of how many resources should be allocated to each instance. This could be left to the benchmark user or defined by the benchmark methodology. If left to the user, the step size and allocation size might get optimized to his or her use case and might reduce comparability of results. If it is defined by the methodology, it is questionable if all possible VNFs can be represented and would further increase the challenge of finding relevant VNFs or VNF groups with predefined step sizes, able to indicate a VNF's behavior when scaled.

4 The Vision of a Scalability and Energy Efficiency Benchmark

Our envisioned benchmark includes two main goals. The first is measuring the vertical and horizontal *scalability* of a VNF. The second goal is measuring a VNF's *energy efficiency* when scaled. To make the benchmark versatile for a wide audience, it should be able to handle a large variety of VNFs with different needs on traffic generation and validation. We envision a benchmark that is agnostic to the VNF that is under test and can be freely configured. For our benchmark we focus on the following resource types as they can largely influence scalability, performance and power consumption and therefore energy efficiency: *i)* number of CPU cores, *ii)* size of main memory, *iii)* filesystem I/O bandwidth and *iv)* network bandwidth. Other metrics that are not mentioned can also be taken into consideration, depending on future research.

Measuring a VNF's ability to scale allows selecting the best performing VNF for an existing infrastructure by matching it to the available resource landscape. If for example only a single but powerful system is available, a vertically scalable VNF implementation could be deployed, while an environment with many but less powerful machines might be better suited for a horizontally scalable VNF. The user can therefore deploy a VNF implementation that suits its available resource landscape and achieve optimal performance when the VNF needs to be scaled. Measuring the scalability and the corresponding performance is also the first step to our second goal.

Adding energy efficiency to the benchmark allows the user to select a VNF that is most efficient for the given task. This allows energy savings and in turn reduces operational costs. Measuring not only the scalability but also the energy efficiency widens the audience for the benchmark. It can be applied by SaaS providers to select the most energy efficient VNF for their offerings but can also allow Infrastructure as a Service (IaaS) users to select a VNF implementation that is scalable within the provided resource landscape. An example use case is shown in Fig. 3. Three different VNF implementations with identical functions are measured and the results are stored in a database. The scalability demand by the customer and the efficiency demand are shown as the horizontal line. While VNF 1 satisfies the customers demand, it is not suited for the service provider. VNF 3 on the other hand is the best solution for the provider but not for the customer. Yet, service provider and customer can both agree on the second implementation as a compromise. This shows, that results from such a benchmark can also be used to reach agreements between providers and consumers for a VNF that suits both needs, the customer's need for satisfactory performance and the provider's for minimizing cost.

5 Planned Approach

We have planned the following approach to achieve our vision of a scalability and energy efficiency benchmark for VNF. It consists of a benchmark methodology based on existing power and energy efficiency benchmarks and tools. We also show a preliminary benchmark setup for our approach. The term configuration used in the following section describes the amount of instances of a VNF for horizontal scaling and the number of different setups in the form of allocated network bandwidth, CPU count, main memory, filesystem I/O and network bandwidth for vertical scaling.

5.1 Methodology

Our methodology consists of two distinct parts and aims to comply with key characteristics for benchmarks, described in [7]. First, we describe the energy efficiency methodology, followed by the scalability methodology.

Energy Efficiency For our work, we base our energy efficiency methodology on the SPEC Power Methodology [8]. Existing work has shown that this methodology ensures high accuracy for power measurements [12] and supports the characterization of system power over multiple load levels [13]. The goal is the power measurement in a steady-state at multiple load levels. DC Power is measured

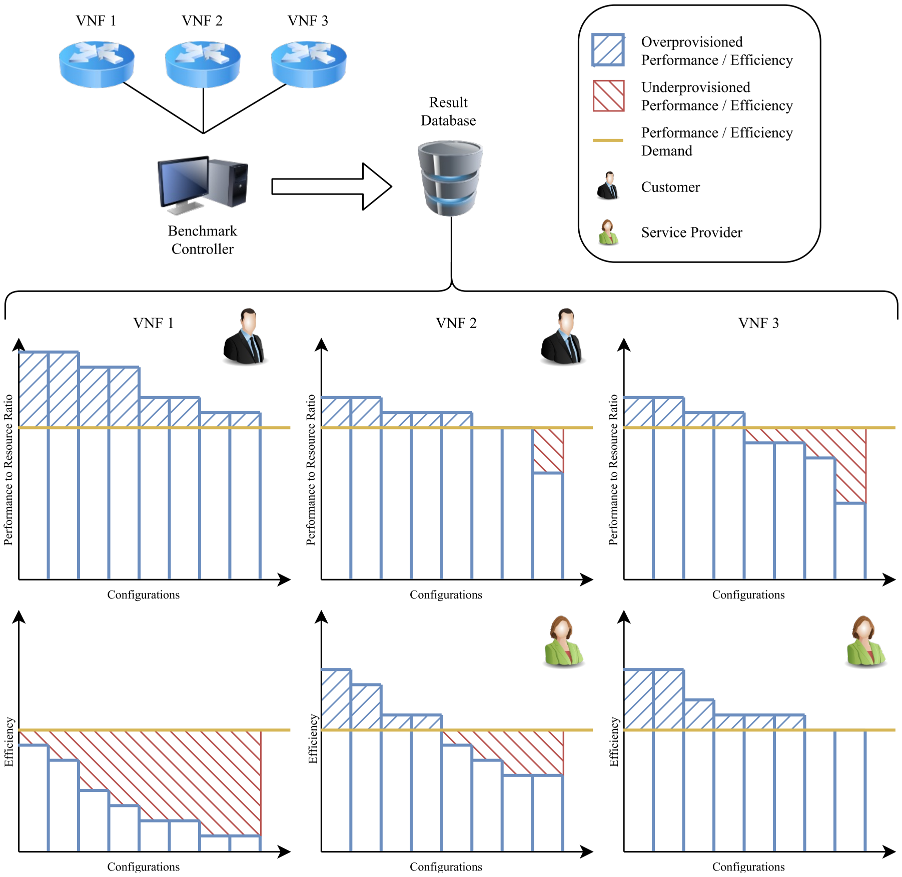

Fig. 3. Scalability and Energy Efficiency Example Use Case

at the Server's power inlet. Loads used by the benchmark must have the characteristics of transactional benchmarking loads. In the context of performance benchmarking, we consider any load that consists of work packets with a clearly defined and measurable start and end time to be transactional. Transactionality of loads enables multiple benchmark features, such as throughput measurements, load calibration, and more. The System Under Test (SUT) is first calibrated to determine its maximum throughput. To achieve calibration, load is generated at a level that is guaranteed to exceed the SUT's capacity and the SUT's throughput is measured. The recorded throughput, averaged over multiple calibration intervals, is assumed to be the maximum (100 %) load level achievable by the SUT. Lower load levels below 100 % are reached by adding random exponentially distributed waiting times between transactions. The mean delay is chosen so that the target transaction rate corresponds to the load level. Figure 4 shows an example of an efficiency measurement according to [8] with the calibration

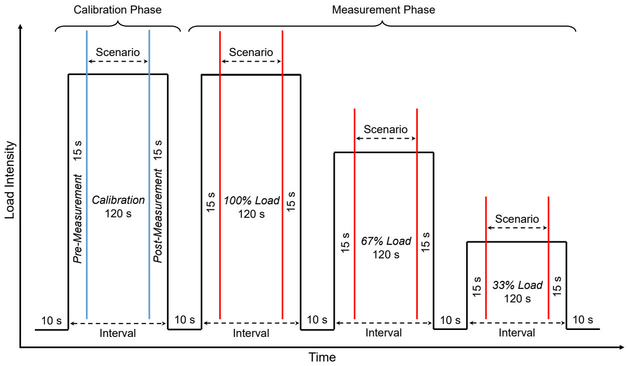

Fig. 4. Energy efficiency benchmark phases [11]

Fig. 5. Example result from an energy efficiency measurement with four load levels

and three load levels. All measurement intervals as well as the calibration have a pre- and post-measurement phase to allow the SUT to reach steady-state before measurements begin.

Figure 5 shows an example result from an energy efficiency measurement of a control plane VNF (SDN controller) for four different load levels, demonstrating the practical applicability of the envisioned approach. Results from measurements with our methodology should also include reporting requirements on which configuration was tested under what load. The figure shows that the VNF under test consumes different amounts of power for each of the load levels. As performance of the load levels varies by design, the resulting energy efficiency (the ratio of performance over power) differs as well.

The environmental conditions of the benchmark can have a significant impact on the power consumption of electrical devices. E.g., a hot environment would need more powerful cooling systems drawing more power. As a result, it is necessary not only to measure the power consumption but also the environmental temperature. The temperature for air cooled server systems is measured not more than 50 mm from the air inlet as described in [8].

Scalability We currently plan to express a VNF's scalability as the ratio between its maximum performance and used resources. For horizontal scaling, *used resources* is the number of instances deployed and running. Vertical scaling uses

CPU count, main memory, I/O bandwidth and network bandwidth as the scaled resource. In addition, a combination of resources might be possible, such that main memory and CPU count are scaled at the same time. Yet, benchmarking all possible resource combinations could increase the number of configurations and subsequently the benchmarks runtime significantly. Hence, we see the need to let the benchmark user select the resources that are scaled under well defined rules to achieve optimal results for the VNF under test. The resources selected must then be documented and combined with the benchmark results to keep them comparable. The benchmark should also include the scalability results for the four resources mentioned in Sec. 4 as a baseline to all benchmark runs.

As energy efficiency is measured simultaneously with scalability, the correct distribution of load levels across the different configurations becomes an important factor. We identified three different possibilities to distribute the load levels over all configurations as shown in Fig. 6:

A) The first option is to distribute all measured load levels equally across all possible configurations and calibrate only once at the configuration with the highest performance. In the example in Fig. 6, each possible configuration has an equal number of load levels that are determined by the single calibration C_0. This keeps the benchmark runtime low, as only a single calibration has to be performed. Yet, this option has some drawbacks. First, if the number of configurations is not fixed in the benchmark, the number of load levels varies as well. This reduces the comparability between different VNF implementations that are measured with a different number of configurations. Even if the number of configurations is fixed, it cannot be guaranteed by the benchmark that the load levels for a specific configuration are representative. Second, in the example, the peak load for the second configuration is not measured. Neither by L_2 nor L_3. Also L_4 overloads the third configuration and possibly invalidating the measurement for this load level.

B) The second option is to have a fixed number of load levels per configuration and calibrate each configuration. The load levels are distributed according to the SPEC measurement methodology [8], based on the calibration for the specific configuration. While this option avoids the second problem of not measuring peak load or overload of a configuration, this option also has drawbacks. In the example shown in Fig. 6, each configuration has three load levels, full load, 66 % and 33 %. For the first configuration, L_1 is below the maximum performance of the second configuration and L_2 is near the full load of configuration three. These load levels are not necessary for elasticity (see Sec. 2) as the system should adapt to the performance demand and will not operate in an overprovisioned state, elongating the benchmark's runtime unnecessarily.

C) The third option is to distribute a fixed number of load levels between the calibrated full load of the current configuration and the next smaller (less peak performance) configuration. In our example (Fig. 6), the load levels for the first configuration are distributed between the calibration C_0 and C_1. For the second configuration the load levels are between C_1 and C_2. The configuration with the lowest maximum performance will distribute its load levels between its calibration (C_2 in the example) and the idle state (L_8). This removes unnecessary load levels and increases the relevance with more load levels. Yet, it introduces an idle measurement at the lowest configuration.

Of the three introduced options for load level distribution among different system configurations. We discourage the first option *A)* with the most drawbacks in favor of either option *B)* or *C)*. Between *B)* and *C)*, we see the latter as the most promising option with the least drawbacks that does not include possibly unnecessary measurements that, in return, might not be relevant to the benchmark user.

5.2 Setup

The envisioned setup of our benchmark consists of at least the following components: *i)* experiment controller, *ii)* load generator, *iii)* traffic receiver and validator, *iv)* power analyzer, *v)* temperature analyzer and *vi)* SUT. Optionally, a meter controller can also be used for managing the dedicated power and temperature measurement devices. Each component can be seen in Fig. 7.

The experiment controller starts and stops the measurements. It collects all data from the involved components and compiles the final report. Communication takes place in a dedicated control network so measurements are not disturbed.

The load generator produces traffic that matches the load level that should be measured and network related configuration, such as packet size, packet content and protocol that must be used to stress the VNF under test. It also distributes the traffic to all instances of VNFs deployed for measurement. Even though a dedicated load balancer is possible, we discourage using a load balancer not delivered together with the benchmark to keep the benchmark reproducible. An optimized load balancer could skew the results in favor of a specific VNF implementation, especially if the load balancer and its configuration is not publicly available for other users to reproduce and verify the results.

After the traffic has passed through the SUT, it must be validated to check if the VNF performed the operation according to specification. This is done at the traffic receiver and validator component. Both, the load balancer and validator, must be configurable to match the VNF that is under test appropriately.

Power and temperature measurements are performed with dedicated measurement devices deployed at the test site. They are either connected directly to the experiment controller or an optional meter controller if required.

The setup of the SUT varies. It depends on how many instances are deployed and on how many hypervisors or physical machines they are distributed. This changes over the course of a benchmark run as the system is scaled. Therefore, we do not specify any general setup restrictions.

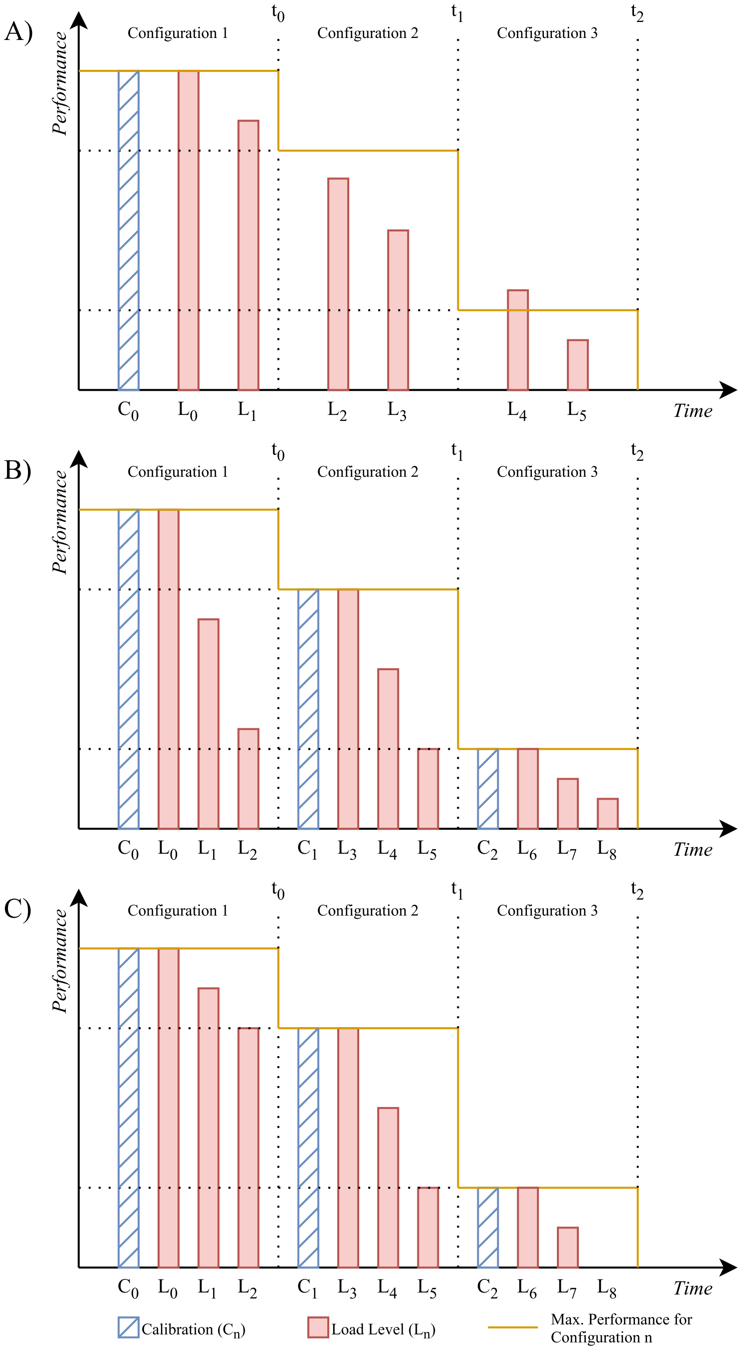

Fig. 6. Options for scalabilty and energy efficiency load level distribution across multiple configurations

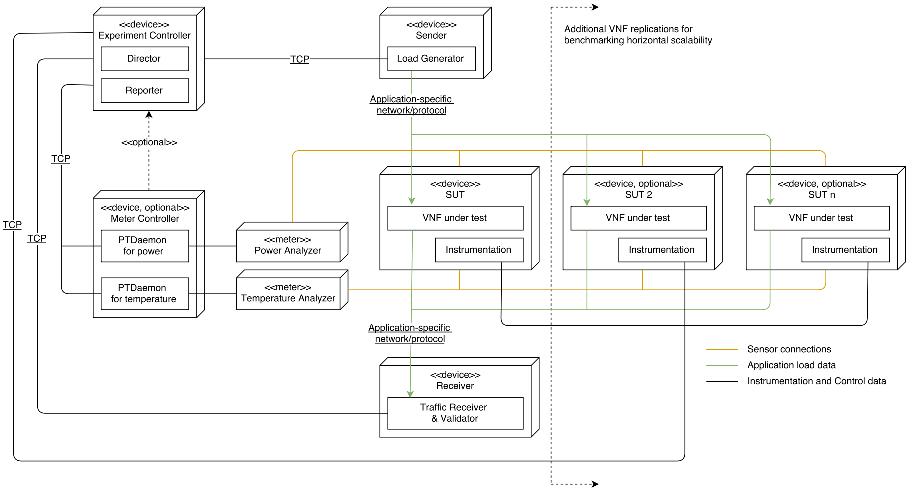

Fig. 7. Envisioned benchmark setup

6 Conclusion and Next Steps

In this work, we present a new benchmark for scalability and energy efficiency specifically for VNFs. Such a benchmark can give customers and service providers a rating to select a VNF implementation that fits their need. We presented current work on VNF benchmarking and its methodology as well as virtualization and energy efficiency benchmarks. Based on the current work, we identify four key issues due to rising complexity and abstraction in a cloud environment, but also the flexibility of VNFs and the domain they are used in. We present our vision and the approach to measure and quantify energy efficiency and scalability in a single benchmark together with the proposed setup.

To proceed towards our vision, we first must resolve the identified issues in Sec. 3. The first step is to find an abstraction of the software stack, virtualization and hardware to make VNF implementations comparable with each other. If this is not possible, all supporting systems must be accounted for in the benchmark's methodology. From this basis we can further proceed to build our methodology and resolve the remaining problems and technical issues on the way towards our vision.

References

1. Brown, R., Masanet, E., Nordman, B., Tschudi, B., Shehabi, A., Stanley, J., Koomey, J., Sartor, D., Chan, P., Loper, J., Capana, S., Hedman, B., Duff, R., Haines, E., Sass, D., Fanara, A.: Report to Congress on Server and Data Center Energy Efficiency: Public Law 109–431. (2007)
2. Whitney, J., Delforge, P.: Scaling Up Energy Efficiency Across the Data Center Industry: Evaluating Key Drivers and Barriers. Data Center Efficiency Assessment, (2014)
3. v. Kistowski, J., Herbst, N., Kounev, S.: Modeling Variations in Load Intensity over Time. LT'14 Proceedings of the third international workshop on Large scale testing, 1–4 (2014)
4. Chiosi, M., Clarke, D., Willis, P., Reid, A., Feger, J., Bugenhagen, M., Khan, W., Fargano, M., Cui, C., Deng, H., Benitez, J., Michel, U., Damker, H., Ogaki, K., Matsuzaki, T., Fukui, M., Shimano, K., Delisle, D., Loudier, Q., Kolias, C., Guardini, I., Demaria, E., Miverva, R., Manzalini, A., López, D., Salguero, F., Ruhl, F., Sen, P.: Network Functions Virtualisation - Introductory White Paper. SDN and OpenFlow World Congress, (2012)
5. Lorido-Botran, T., Miguel-Alonso, J., A. Lozano, J.: A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments. Journal of Grid Computing, 559–592 (2014)
6. Herbst, N., Kounev, S., Reussner, R.: Elasticity in Cloud Computing: What It Is, and What It Is Not. In: Proceedings of the 10th International Conference on Autonomic Computing ICAC, San Jose, California, USA (2013)
7. v. Kistowski, J., A. Arnold, J., Huppler, K., Lange, K.-D., L. Henning, J., Cao, P.: How to Build a Benchmark. In: Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering ICPE, Austin, Texas, USA (2015)
8. Power and Performance Benchmark Methodology V2.1. Standard Performance Evaluation Corporation (SPEC), Gainesville, USA (2012)
9. Huppler, K.: Performance Per Watt - Benchmarking Ways to Get More for Less. In: Nambiar R., Poess M. (eds) Selected Topics in Performance Evaluation and Benchmarking. TPCTC 2012. Lecture Notes in Computer Science, vol 7755. Springer, Berlin, Heidelberg (2012)
10. Schall, D., Hoefner, V., Kern, M.: Towards an Enhanced Benchmark Advocating Energy-Efficient Systems. In: Nambiar R., Poess M. (eds) Topics in Performance Evaluation, Measurement and Characterization. TPCTC 2011. Lecture Notes in Computer Science, vol 7144. Springer, Berlin, Heidelberg (2012)
11. v. Kistowski, J., Becket, J., Lange, K.-D., Block, H., A. Arnold, J., Kounev, S.: Energy Efficiency of Hierarchical Server Load Distributions Strategies. 2015 IEEE 23rd International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS), Atlanta, USA (2015)
12. v. Kistowski, J., Block, H., Beckett, J., Spradling, C., Lange, K.-D., Kounev, S.: Variations in CPU Power Consumption. ICPE'16 Proceedings of the 7th ACM/SPEC International Conference on Performance Engineering, Delft, The Netherlands, 147–158 (2016)
13. v. Kistowski, J., Block, H., Beckett, J., Lange, K.-D., A. Arnold, J., Kounev, S.: Analysis of the Influences on Server Power Consumption and Energy Efficiency for CPU-Intensive Workloads. ICPE'15 Proceedings of the 6th ACM/SPEC

International Conference on Performance Engineering, Austin, Texas, 223–234 (2015)

14. Lange, S., Nguyen-Ngoc, A., Gebert, S., Zinner, T., Jarschel, M., Köpsel, A., Sune, M., Raumer, D., Gallenmüller, S., Calre, G., Tran-Gia, P.: Performance Benchmarking of a Software-Based LTE SGW. 2015 11th International Conference on Network and Service Management (CNSM), Barcelona, Spain (2015)
15. Falkner, M., Leivadeas, A., Lambadaris, I., Kesidis, G.: Performance Analysis of Virtualized Network Functions on Virtualized Systems Architectures. 2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), Toronto, Canada (2016)
16. Bradner, S., McQuaid, J.: RFC 2544: Benchmarking Methodology for Network Interconnected Devices. IETF, (1999)
17. Bradner, S., McQuaid, J.: RFC 6201: Device Reset Characterization. IETF, (2011)
18. Bradner, S., Dubray, K., McQuaid, J., Morton, A.: RFC 6815: Applicability Statement for RFC 2544: Use on Production Networks Considered Harmful. IETF, (2012)
19. Rosa Ed., E., Szabo, R.: VNF Benchmarking Methodology (draft expired in Sept. 2016). IETF, (2016)
20. Morton, A.: Considerations for Benchmarking Virtual Network Functions and Their Infrastructure (draft expired in Aug. 2015). IETF, (2015)
21. Moens, H., De Turck, F.: VNF-P: A Model for Efficient Placement for Virtualized Network Functions. 2014 10th International Conference on Network and Service Management (CNSM), Rio de Janeiro, Brazil (2014)
22. Addis, B., Belabed, D., Bouet, M., Secci, S.: Virtual Network Functions Placement and Routing Optimization. 2015 IEEE 4th International Conference on Cloud Networking (CloudNet), Niagara Falls, Canada (2015)
23. SPEC VIRT_SC 2013. The Standard Performance Evaluation Corporation (SPEC), https://www.spec.org/virt_sc2013/, (2013)
24. D. Smith, W., Sebastian, S.: Virtualization Performance Insights from TPC-VMS. Transaction Processing Performance Council, (2013)
25. Bond, A., Johnson, D., Kopczynski, G., Taheri, H. R.: Profiling the performance of virtualized databases with the TPCx-V benchmark. In: Nambiar R., Poess M. (eds) Performance Evaluation and Benchmarking: Traditional to Big Data to Internet of Things, Springer, Berlin, Heidelberg (2015)
26. TPC-Energy Specification. Transaction Processing Performance Council, Version 1.5.0, http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-energy_v1.5.0.pdf, (2012)
27. SPECpower_ssj2008. The Standard Performance Evaluation Corporation (SPEC), https://www.spec.org/power_ssj2008/, (2007)
28. Server Efficiency Rating Tool 2. The Standard Performance Evaluation Corporation (SPEC), https://www.spec.org/sert2/, (2017)
29. Chauffeur$^{\mathrm{TM}}$WDK 2. The Standard Performance Evaluation Corporation (SPEC), https://www.spec.org/chauffeur-wdk/, (2017)