A Performance Evaluation Metric for NFV Elements on Multiple Timescales

David Stezenbach
University of Vienna, Austria
Future Communication Research Group
david.stezenbach@univie.ac.at

Kurt Tutschku, Markus Fiedler
Blekinge Institute of Technology (BTH)
School of Communication, S-371 79 Karlskrona, Sweden
{kurt.tutschku, markus.fiedler}@bth.se

***Abstract*—Todays virtualization technologies are manifold and a comparison is hardly achievable. Thus, a metric independent from the virtualization technology is required to compare different systems. The metric is measurable in a passive way, hence, no artificial traffic has to be generated, and the virtualization system needs not to be modified. It evaluates the throughput of events on different time slices. This methodology, in contrast to existing jitter evaluations, enables the identification of critical timescales in the virtualization system.**

In this demonstration a proof-of-concept for a performance metric for NFV elements on multiple timescales is presented. In a reduced environment consisting of a single virtual router host the influences of hardware resource sharing and other impact factors (e.g. cpu, memory or disc load) are made visible. The demonstration gives an example of a performance degrease on smaller timescales, which can not be identified by a common throughput measurement over time. Thus, the presented metric enables to identify critical system conditions and can be used to optimize the scheduling of NFV, to compare different virtualization technologies, or to grade the performance for specific applications.

I. Introduction

Network function virtualization could be one of the key functionalities of future network architectures. Still a central problem is the performance comparison of different virtualization technologies or the event processing in contrast to physical hardware. Common evaluation methodologies aim at the maximum total throughput or the adherence of timing deadlines, which are not feasible for the identification of scheduling problems in a virtualization environment, since they only provide absolute values. These values are highly dependent on the ingress event stream and do not consider performance influences resulting from the time and space division multiplexing used in virtualization environments.

The evaluation has to consider key requirements of virtual entities and grade them with a comparable numerical value. In this evaluation the two key requirements are considered:

- *The isolation between different virtualized network functions* is ideal if the execution of one function does not impact an other one sharing the same physical hardware
- *Full transparency of the virtual element* is achieved when the virtualized function can not identify that it is not directly running on physical hardware

The metric presented and evaluated in this demo grades the performance of a virtual element on different timescales. Therefore, the violation of both of the above requirements is described by the same observable measure, the inter-event time. If one assumes a constant processing time the inter-event time should be equal after the processing. This enables the comparison of the amount of incoming events and the output results on different timescales. Our suggested metric expresses both, the degree of *isolation* and *transparency* of a virtual system in a single comparable value. In contrast to common jitter analysis, which focus on absolute values and only represent maximum aberrations from a mean processing time, this metric helps to identify performance issues on multiple timescales in a clearly arranged and comparable manner.

Our measurement setup uses the comparative analysis [1] to analyze the performance of a simple XEN based virtual router. The input of our metric is the inter-packet time at the ingress compared to the egress. Our results show, that the time and space multiplexing in a virtual router affects the packet order between different virtual router instances. This results in an unfair sharing of resources on smaller timescales, although the total number of processed events, the throughput, is equal on the egress compared to the ingress.

This demo proposal is structured as follows: First, a theoretical illustration of the implemented metric is presented in Section II. The feasibility in an practical live environment is shown in Section III. In the final Section IV the demo setup and requirements are briefly described.

II. Theoretical Illustration

A. Measure Definition

In the following, basic measures are presented that show the feasibility of the inter-event time as an indicator for the performance degree in a virtualized environment. The inter-event times T between two subsequent events e and analogous for the replies$r$ are defined in Equations 1 & 2

$$T_i^{in} = t_{e_i}^{in} - t_{e_{i-1}}^{in} \quad (1)$$

$$T_i^{out} = t_{r_i}^{out} - t_{r_{i-1}}^{out} \quad (2)$$

In an theoretical ideal case, the processing time c_p equals zero and would lead to a fully *transparent* system. As the event processing is not possible in zero time, the assumption

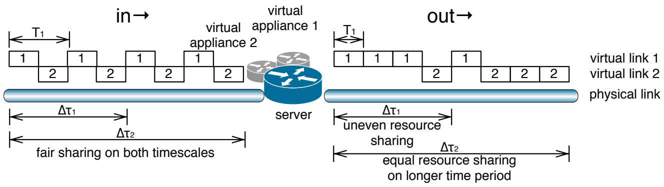

Fig. 1. Illustration of $T_i^{in/out}$, $\Delta\tau$ and isolated resource sharing

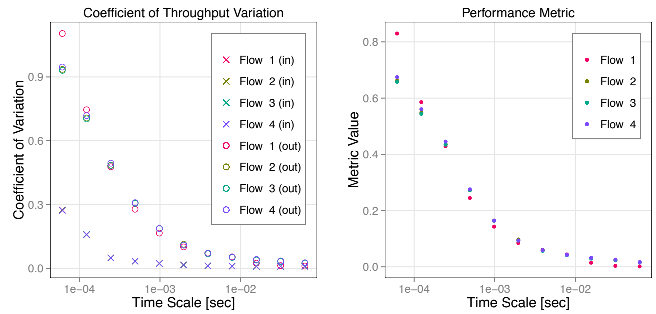

Fig. 2. Coefficient of Throughput Variation and the Resulting Performance Metric on Multiple Timescales

that the processing time is constant suits the requirements best. Furthermore, the processing of two events should not interfere each other. Thus, the processing time c_p is required to be constant in an *isolated* environment. Accepting this for each event the time relationship between the event and its corresponding reply follow Equation 3. This leads to the fact, that the inter-arrival time of events T_i^{in} should be equal to the inter-arrival time of replies T_i^{out} in an *isolated* and *transparent* environment.

$$t_{r_i}^{out} = t_{e_i}^{in} + c_p \tag{3}$$
$$T_i^{in} = T_i^{out}, \forall i \tag{4}$$

As the evaluation of the inter-arrival time of numerous event and reply pairs is not easy to interpret, a derived measure is needed. This measure should reduce the amount of data and group it on various timescales in order to enable a statistic evaluation based on this. With Equation 5 the total size ($x = 1 \Rightarrow$ event count) of processed events / replies is expressed until a certain moment in time. Thus, the event / reply processing in time intervals $\Delta\tau$ is represented by Equation 6. In the current measure subsequent time intervals are used, but a sliding window approach is planned for future work.

$$X^{in/out}(t) = \sum_{p:t_e^{in}/t_r^{out} \leq t} x_{e/r}^{in/out} \tag{5}$$
$$R_{j,\Delta\tau}^{in/out} = \frac{X^{in/out}(j\Delta\tau) - X^{in/out}((j-i)\Delta\tau)}{\Delta\tau} \tag{6}$$

In Figure 1 the relationship between $T_i^{in/out}$, $\Delta\tau$ and the isolated (fair sharing) use of resources is illustrated. The intervals at the egress are shifted by the processing time of the first event, thus in the case of an isolated event processing the replies fall in the corresponding time interval and R_j^{in} equals R_j^{out}. If the inter-arrival time T_i of events changes from the ingress to the output of replies, the replies grouped differently in the intervals $\Delta\tau$ and resource sharing is not fair.

In principle, the time scale $\Delta\tau$ can be chosen arbitrarily. In case of an continuous event stream with equal inter-arrival times, the smallest time scale should follow Equation 7 in order to expect one event per time interval. The scaling can be observed best of the different timescales are related. In this example power of two multiples of the smallest timescale are used.

$$\Delta\tau \in 2^n \cdot E\left[T_i^{in}\right] \Rightarrow E_n\left[R_{\Delta\tau}^{in}\right], n \in \mathbb{N}_0 \tag{7}$$

As a next step the characterization of a set $R_{i,\Delta\tau}^{in/out}$ is possible following Equation 8 using the coefficient of variation. This measure describes how the amount of events / replies per time interval varies on different timescales.

$$c_{\Delta\tau}^{in/out} = \sigma\left[\left\{R_{i,\Delta\tau}^{in/out}\right\}\right] / E\left[\left\{R_{i,\Delta\tau}^{in/out}\right\}\right] \tag{8}$$

Based on $c_{\Delta\tau}^{in/out}$ and the methodology of comparative analysis [1] the comparison of the ingress of events and the output of replies. The degree of influences cased by the virtualization is reflected by a change in the statistical behavior of $R_{i,\Delta\tau}^{in/out}$. In case of an perfectly isolated system, the statistical behavior of $R_{i,\Delta\tau}^{in/out}$ is equal. This allows the definition of the metric following Equation 9. The metric $v(\Delta\tau)$ is proposed to express the goodness of isolation and transparency by comparing the coefficient of variation of the output with the respective one ingress. In the ideal case $v(\Delta\tau) = 0$ the investigated system has no impact on the event / reply stream. Typically, the values of $v(\Delta\tau)$ are larger than zero, as the internal scheduling of the investigated system is suspected to process events for each network function in short bursts before switching to the next function. Results of a first off-line investigation are presented in Figure2 showing the coefficient of throughput variation and the resulting metric values. This example measurement was performed using a XEN based virtual router environment with four virtual routers and a round-robin ingress packet stream. The sharing of resources is nearly equal on larger timescales, while on small timescales the output has a significantly higher variation. This is a result of the internal scheduling between the different virtual routers and thus, a not isolated resource sharing.

$$v(\Delta\tau) = c_{\Delta\tau}^{out} - c_{\Delta\tau}^{in} \tag{9}$$

Thus, the performance metric grades the virtualization system under the current conditions and helps to identify performance issues. On its own, it does not explain why a

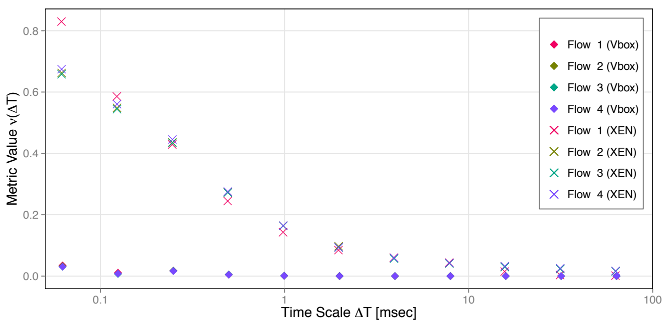

Fig. 3. Comparison of XEN and VirtualBox hypervisor for a Virtual Router Scenario with 100 Mbit/s, 4 Instances and 50 Bytes Packet Size

system performs differently compared to an other virtualization system. Hence, the results need to be interpreted for each specific system. Given the example in Figure 3 the XEN hypervisor seems to perform worse in this example compared to an identical system running a VirtualBox hypervisor. The current assumption is, that the use of the vSwitch in the XEN hypervisor without utilizing Intel DPDK prohibits fast packet processing.

III. Practical Investigations

In the demo the metric presented above is running in a live measurement environment. In order to have an independent capturing and evaluation unit, which does not interfere with the scheduling of the investigated virtual element, a setup with an external capturing unit as presented in Figure 4 is used. The capturing unit is equipped with a DAG card for high precision time stamping and to avoid problems with asynchronous clocks. For the plot shown in Figure 5 a capturing sample of 25 seconds is used, starting with a single flow passing a virtual router with line speed at 100 Mbit/s with a packet size of 1472 bytes. After 50 k packets, a second flow is started transmitted over a second virtual router. The third and fourth flow are started analogously. The coefficient of throughput variation and the performance metric is calculated using a sliding window approach with a window size of one second, this is why the lines do not start at zero seconds. During the phase with only one flow running the performance metric indicates an isolated and transparent resource sharing on both timescales (blue ⇒ 1 sec, red ⇒ 1 ms). When a second flow is started, the internal scheduling of the virtual routers decrease the performance and the performance metric shows an value above zero on smaller timescales. On the one second timescale this is not noticeable, thus, a plain throughput measurement would not identify this performance decrease. Furthermore, it is worth mentionable that the performance is not always getting worse, as seen on the second flow during the start of the fourth.

The live demo will consist of different scenarios and show the performance impact of competing flows over the same or different virtual routers. Furthermore, the performance decrease caused by cpu, memory or disc load can be visualized.

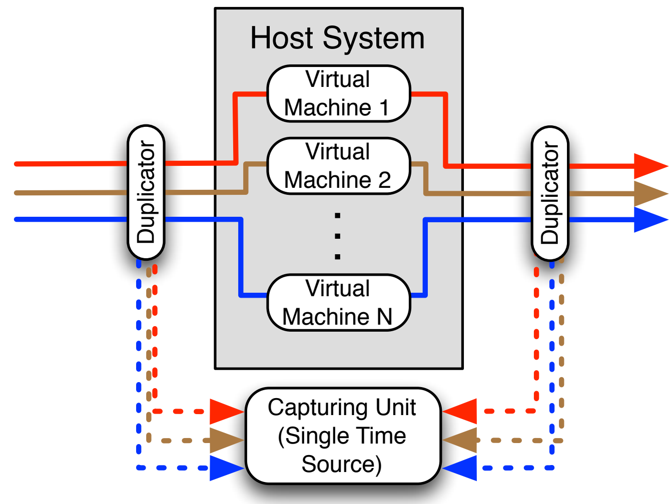

Fig. 4. The Passive Measurement Setup for the Comparative Analysis

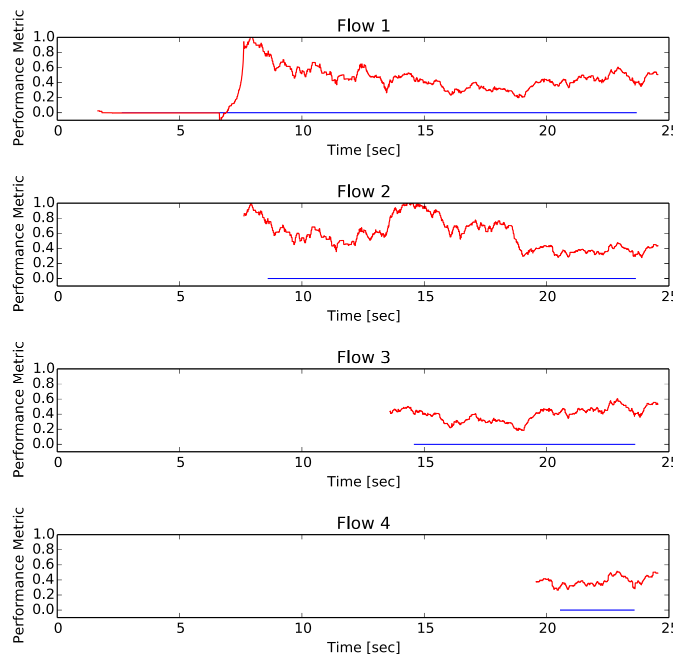

Fig. 5. Live Performance Measuring of a Four Router Example on Different Timescales (blue ⇒ 1 sec, red ⇒ 1 ms)

IV. Technical Setup and Requirements

The demo setup runs in a cloud environment, thus, no space is required. The visualization will be performed on a laptop. I room is available an external monitor could alleviate multiple people to watch the demo. Furthermore, there is only some minutes setup time required.

References

[1] M. Fiedler, K. Tutschku, P. Carlsson, and A. A. Nilsson, "Identification of performance degradation in ip networks using throughput statistics," in *Providing Quality of Service in Heterogeneous Environments. Proceedings of the 18th International Teletraffic Congress (ITC-18)*. Berlin, Germany, 2003, pp. 399–407.