

Performance Analysis of Virtualized Real-time Applications

Miroslav Voznak, Jiri Slachta and Jan Rozhon

Abstract— This article deals with the impact of virtualization techniques on interactive delay-sensitive applications running in real-time, particularly IP telephony. Many institutions, organizations and home users often adopt the virtualized solutions for their safety, ease of administration and backup. Virtualization, which was chiefly the prerogative of companies and the academic world in its early days, has gradually develop its platform to reach out to the ordinary users who can benefit from running virtual machines. The aim of this paper is to examine the impact of a virtual machine on real-time traffic, in our case IP telephony based on the SIP and the RTP, which are now the cornerstone of VoIP technology. This article also analyses the impact of memory size and the number of processor cores on the delay itself and its variance, thus allowing user to have full picture when deciding what virtualization tool to use and how to configure so it performs the best possible way.

Keywords— Asterisk, Benchmarking, Delay, Full virtualization, Jitter, KVM, Real-time applications, VirtualBox, VMware player.

I. INTRODUCTION

NOWADAYS virtualization presents a solution which can enhance security and efficiency and ease the infrastructure maintenance in the corporate and academic environment. Moreover, virtualization has now spread even to home PCs an environment which is rather different to what was common in the very beginning of this technology. The service providers' offers confirm this trend as they incorporate virtual machine hosting as one of the key services they provide [1], [2].

This way, the virtualization technology has found its way into many implementations in various areas including data centres, development studios and even households. There are three main advantages of virtualization. First, it allows us to separate critical applications which are considered problematic when running simultaneously. Second, the increased portability makes it possible to transfer the virtualized system

between physical devices equipped with different software and having different hardware architecture. Third, it eliminates the need to configure the operating system frequently as the work can be performed on the copy of the virtualized operating system leaving the original one intact for further use.

Besides the above mentioned advantages we also need to take into account the drawbacks of the technology, especially now that its use has become so prevalent. The main disadvantage is an overhead generated by the virtualization tool. The overhead causes virtual machines to be less efficient than the physical devices with similar attributes and reduces their performance. Further, the overhead can have a negative impact especially on real-time applications since it can cause long delays and increase the variance of delay between the individual packets [3], [4].

The aim of this paper is to find out what impact do different implementations of the virtualization technology have on the real-time traffic, represented here by IP telephony as it is one of the most widely spread real-time technologies. The influence of the number of processor cores and memory size is also to be analyzed.

II. TOOLS FOR VIRTUALIZATION

This part presents the three most common virtualization tools and outlines their basic parameters. These three virtualization platforms will be used to measure the delay and its variation and include VMware Player, Kernel-based Virtual Machine (KVM) and VirtualBox. All the mentioned tools are free of charge and can be used on multiple software platforms.

Since the KVM is fully focused on high performing Linux systems we can assume that its performance is going to be the highest. VirtualBox, on the other hand, will give the poorest performance, as it is the most universal platform from the point of view of the host system's architecture. The VMware player maintains the balance between performance and versatility so this can be considered as middle-class solution.

A. Kernel-based Virtual Machine

High performance requirements on the instruction translation in the binary form resulted into a combination of experience gained in the different virtualization models. When the hardware-assisted virtualization emerged, a new kernel-module-based hypervisor started to be developed for the GNU/Linux platform. This hypervisor combines both high

Manuscript received December 17, 2011. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 218086.

M. Voznak is with the VSB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava Poruba, Czech Republic (phone: +420 5969991699 ; e-mail: miroslav.voznak@vsb.cz).

J. Slachta is with the VSB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava Poruba, Czech Republic (phone: +420 5969991699 ; e-mail: jiri.slachta@gmail.com).

J. Rozhon is with the VSB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava Poruba, Czech Republic (phone: +420 5969991699 ; e-mail: jan.rozhon@vsb.cz).

performance and versatile usability. By extending the Linux kernel with the KVM hypervisor, the advantages of the model which allows for maintaining each single virtual machine as a standard Linux process [8] can be exploited.

B. VirtualBox

VirtualBox is a multiplatform virtualization tool designed to run under OS Windows, Mac OS X, GNU/Linux or Solaris on platforms using both most common architectures – IA-32 and x86-64. The tool allows performing full virtualization with a hosted hypervisor which means that an already installed operating system is required to run this tool. The vast versatility of this tool enables an easy transition between different hosts with different operating systems. The VirtualBox's modularity is the most interesting feature of this tool and offers the possibility of running the virtualized machine with defined parameters from the command line without displaying the screen output or setting the parameters in a user-friendly graphical interface and running the machine by a single click [9].

C. VMware Player

The most favored and well-known producer in the field of virtualization is indisputably VMware. Its products belong among the most used solutions and are mainly designated for the x86 architecture and its descendant x86-64. The company offers products that implement a so called "bare-metal" hypervisor or a hosted hypervisor allowing the company to cover a larger part of the market spanning from end-users with low requirements to servers and data centers in which high efficiency, performance and scalability is a must.

III. METHODOLOGY AND TESTING PLATFORM

The methodology used in this paper relies on and uses the free full virtualization tools. Their virtual machines will be run on high performing hardware with hardware assisted virtualization support.

A. Measuring Platform Preparation

As the KVM needs a hardware-assisted virtualization support, it is necessary to use a computer equipped with a processor supporting the Intel-VT_x or AMD-V technology, two incarnations of the mentioned hardware-assisted virtualization technology from both largest x86 processor producers.

Main hardware and software parameters of the used computer are summarized in further points:

- processor Intel(R) Core(TM) i7 CPU 870 @ 2.93GHz,
- 8 GiB RAM,
- two 1Gbps network interface controllers RTL8111/8168B PCI Express Gigabit Ethernet,
- 64-bit operating system Debian Squeeze.

The tested topology consists of one computer, one 1Gbps switch and traffic generator Optixia XM2 with suitable modules. The hosting computer on which the virtual machines will be run has two NICs. One is connected to the TUONET university network and the other is reserved as the network's test segment. The traffic generator is also connected to the university network which enables to communicate with IxLoad control software that not only performs the test itself but also the data analysis. The generator modules which will be used to generate traffic are connected to a switch from the tested network segment.

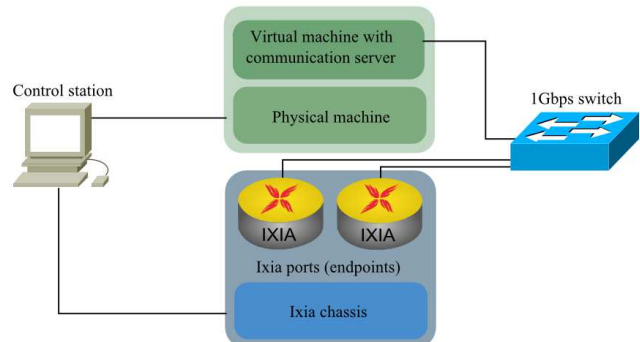


Fig. 1: Test topology

B. Measured Parameters

Real-time network applications including IP telephony depend on network parameters that influence the transmission quality. The concepts used to control the traffic quality in computer networks based on packet switching can be summarized with QoS (Quality of Service). The QoS is highly popular especially when transmitting voice and videos over IP networks where it is used to reserve a bandwidth or prioritizing one channel of traffic over others so that the critical parameters including delay, its variation and the packet loss are not exceeded.

Using the Optixia XM2 we are able to measure these parameters [5]:

Interarrival Jitter – This parameter is measured using the RTP timestamps and represents the mean difference of the arrival time between consecutive packets compared to the same difference on the sender's side.

Delay Variation Jitter – This parameter represents the time between the transmission and the reception of an RTP packet.

One Way Delay – This is the time the packet spends on the route from one end to the other. It cannot be measured when the device under test (DUT) changes the content of this packet.

Post Dial Delay – This is the delay between sending the INVITE message and the reception of the response from the other endpoint.

Media Delay – The time between the INVITE message dispatch and the reception of the first RTP packet.

Post Pickup Delay – This is the time measured on the receiver's end and is defined as the difference between the timestamp of the 200 OK dispatch and the reception of the first audio packet.

If we focus on the transmission chain as such, we will find out that besides the endpoints, many other elements, including switches and routers, can cause the delays and their variations.

Consequently, the delay is defined as the time period during which the packet passes the network between the sender and the receiver endpoint. ITU-T recommendation G.114 defines the following delay range classes:

- Delays of 150 ms are mostly acceptable.
- Delays between 150 and 400 ms deteriorate the quality of the communication significantly.
- Delays of more than 400 ms are entirely unacceptable [7].

We define a variation of delay as a jitter. It is the difference between the expected and the real time of the packet reception. This appears during the packet passage through the IP network when the time shift between packets occurs because of the queue ordering in routers [7], [10]. 30 ms are considered to be the top acceptable value. In this paper, the following parameters were chosen to carry out measurements on traffic generator and analyzer:

- Interarrival Jitter,
- Delay Variation Jitter,
- Post Dial Delay.

When the call is generated, only a low load is shown on virtual machines. More specifically, SIP traffic with voice session without transcoding took up only 30-45% CPU utilization and all the measured parameters were deeply under limits. Moreover we faced up to serious problem regarding a depletion of UDP sockets [6]. We analyzed the problem in SIP server under a high traffic load, SIP server did not open a new socket and the next call could not be proceeded. The behavior was accompanied by following warning message:

```
[Mar 22 15:36:10] WARNING[1020]: chan_sip.c:3587
__sip_xmit: sip_xmit of 0x7f746d358340 (len 480) to
172.16.100.153:5060 returned -1: No buffer space available
[Mar 22 15:36:10] WARNING[1020]: acl.c:495
ast_ouraddrfor: Cannot connect
```

If SIP server needs to proceed a socket then OS is called to open the socket within particular process. However a limitation of file descriptors at a process is always set in every OS in order to protect a machine from flooding (DoS attack). The extremely high load causes the depletion of file descriptors and the device under test refuses next attempts. In the first step, we decided to solve it by a modification of the mentioned limitation in file */etc/security/limits.conf*. Then we

modified value MAXFILES of init script in Asterisk as well, the vaule was set to 65535. The file */etc/security/limits.conf* contained following values:

<domain>	<type>	<item>	<value>
asterisk	soft	nofile	65535
asterisk	hard	nofile	65535

Though we modified the file descriptors, only 1023 active calls could be carried out for mentioned value 65535.

Due to the low virtual machine utilization and low number of the UDP sockets, finally, we decided to implement a codec translation which of course increases the utilization and after that we could observer a difference in performance of real-time applications on various virtualization pltaforms.. The translation was performed between G.711 and G.726 codecs.

Another expected parameter, One Way Delay, could have been excluded from the measurements due to the packet content change in Asterisk PBX. But it can be estimated using the Delay Variation Jitter. The last measured parameter, Post Dial Delay, has a very similar characteristic as the other SIP delay parameters (Media Delay and Post Pickup Delay).

For the sake of this paper and given the vast extent of this field, the Post Dial Delay, Delay Variation Jitter and Interarrival Jitter are going to be the only presented parameters.

C. Scenario Configuration

The testing configuration includes setting the network parameters and test control parameters in the IxLoad control software. The test scenario then describes the test flow as it is being executed by the Optixia XM2 generator.

The configuration can be split into three parts – the first part with global parameters, the second with network parameters, and the third describes the selected test activity. Therefore the test scenario consists of a fixed part, which is the same for all the tests, and the variable part which is determined for the selected activity. Activities can be combined, enabling measuring multiple parameters during single test iteration. The scenario creation is realized through the graphical interface similar to graphical programming languages.

The applied scenario is very simple]. The test starts with the MakeRegistration procedure. Once both sides of the communication are registered, the SIP MakeCallAuthentication and SIP ReceiveCallAuthentication procedures are executed. These are followed by the authentication RTP session. Once it is over, the call is ended.

Figure 2 provides an overview of the SIP traffic load used in the test.

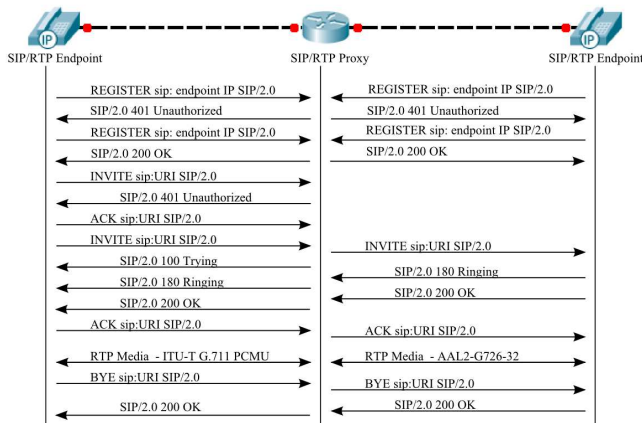


Fig. 2: SIP messages and RTP session used in test.

D. Test Methodology

The test scenario remains the same for all the tests though several parameters of the virtual machines, including RAM capacity, number of processor cores and used virtualization tool, changed.

Accepted values of the measured parameters were defined in section 3.2. Due to the above-mentioned limitations, the end-to-end delay variation can only be measured between the UA that generates the call and the communication server.

Under our scenario, the traffic has a linearly increasing trend but the utilization and delay increases are not linear at all. Asterisk PBX responds to an increasing load with a notch increase once a certain load threshold is exceeded. Once the hardware limit has been reached, Asterisk begins to refuse registrations and first unsuccessful calls appear.

The overall methodology consists of the following steps:

1. We set the parameters of virtual machines so that they can be identically configured as other virtual machines (each virtual machine uses a different hypervisor) under the test.
2. During the test we observe the delay parameters and try to find the upper load threshold at which delay characteristics do not exceed the defined values and no registration failures occur.

3. With the same preconfigured test scenario, we test all other virtual machines which were configured in the same way as the reference virtual machine. The results of the test are saved as CSV files.
4. The results of the test are saved in the CSV files stored in the output IxLoad directory. These files will be used to perform a statistical evaluation.
5. After testing one combination of parameters on all virtual machines, we repeated the measurement under a different setting.

The table 1 below shows the load as it was determined by the most performing virtualization tool which fulfilled the defined limitations.

TABLE 1: RELATION BETWEEN THE NUMBER OF VIRTUAL CPUs AND ACTIVE CALLS IN TEST

Number of processor Cores	Number of Active Calls
1	280
2	440
3	600
4	680

All the data obtained were analyzed using the Staghraphics Centurion XV statistical software.

IV. RESULTS

The data files were analyzed using the exploratory analysis applied to each individual parameter. The ANOVA test was applied to verify data independence and other required properties. Every result category consists of charts describing how the three most important parameters are influenced by the current environmental setting. These parameters are Post Dial Delay, Delay Variation Jitter (see tables 3-5) and Interarrival Jitter.

TABLE 3: RESULTS OF MEASUREMENTS DEALY VARIATION JITTER IN CASE OF VARIOUS VIRTUAL PLATFORMS

Dealy Variation Jitter on various Virtual Platforms									
	Count	Average	Median	Coeff. of variation	Minimum	Maximum	Range	Lower quartile	Upper quartile
KVM	4705	1109,35	982	53,9108%	96	7466	7370	621	1603
vbox	4705	12672,4	11369	76,8597%	603	38016	37413	2553	18604
vmware	4705	5505,47	2896	99,0376%	349	34799	34450	2290	5066
Total	14115	6429,07	2354	124,782%	96	38016	37920	1266	10565

TABLE 4: RESULTS OF MEASUREMENTS DEALY VARIATION JITTER IN CASE OF VARIOUS NUMBER OF PROCESSOR CORES

Dealy Variation Jitter on various number of cores									
	Count	Average	Median	Coeff. of variation	Minimum	Maximum	Range	Lower quartile	Upper quartile
core1	3528	12850,2	13311	83,7955%	96	38016	37920	798,5	19025,5
cores2	3528	6504,35	2568	113,216%	281	35454	35173	1460	11527,5
cores3	3528	4248,42	2706	110,38%	302	31319	31017	894	6298
cores4	3528	2117,12	1955,5	40,9808%	652	8223	7571	1600	2303,5

TABLE 5: RESULTS OF MEASUREMENTS DEALY VARIATION JITTER IN CASE OF VARIOUS MEMORY SIZES

Dealy Variation Jitter on various Memory Size									
	Count	Average	Median	Coeff. of variation	Minimum	Maximum	Range	Lower quartile	Upper quartile
GB05	3528	6075,16	2443,5	118,887%	96	35910	35814	1289,5	10553
GB1	3528	6662,6	2929	120,726%	127	37506	37379	1275,5	10386
GB2	3528	6494,56	2309	130,616%	147	38016	37869	1116	10301,5
GB4	3528	6487,78	2152	127,581%	119	37489	37370	1410	10875

A. Classification according to Virtualization Tool Performance

This classification describes data properties for three variables which represent individual virtualization tools.

1) Post Dial Delay

The first parameter we are going to analyze is Post Dial Delay. The Fig. 3 depicts three variables (KVM,VirtualBox, VMware) and their effect on the Post Dial Delay. The first variable, KVM, has very limited range of measured data especially when compared to the other variables (VirtualBox and VMware). However, due to this limited range it is impossible to determine how these values are distributed.

On the other hand this does not apply on the other variables. VirtualBox has its median value lower than the average meaning that the most values were observed mainly under the average, which is affected by several high values. This can be told about VMware as well, since the data distribution is similar to VirtualBox except the narrower data range. All these finding can be seen on the Fig.3.

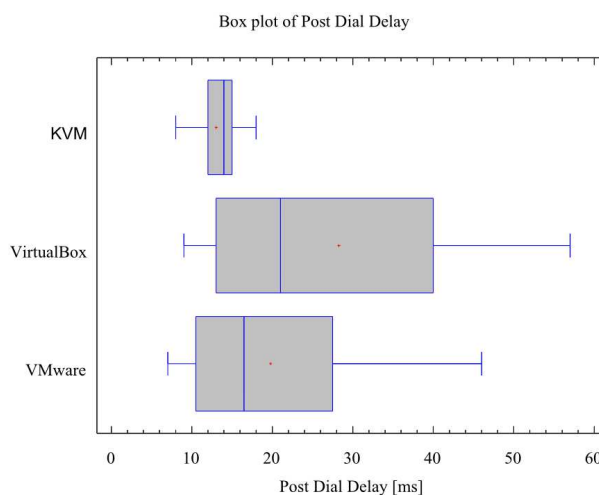


Fig. 3: Boxplot of Post Dial Delay for all three virtualization tools.

2) Delay Variation Jitter

Even in this parameter we can see (on the Fig. 4) the similar behavior of the virtualization tool. Again, the results measured with KVM are distributed in a very narrow area, while Virtualbox tends to spread the measured delays in the much wider range.

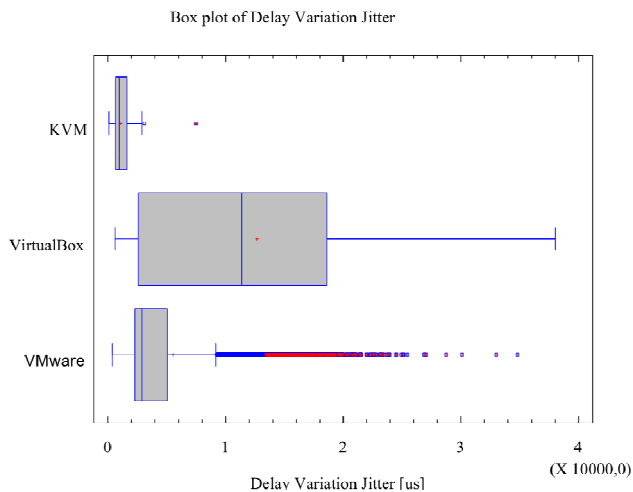


Fig. 4: Boxplot of Delay Variation Jitter for all three virtualization tools.

VMware as the only virtualization technology has results containing numerous outlying observations leading to conclusion that although the results are generally better the VirtualBox'es they suffer from many spikes. This behavior can be problematic in environments, where constant performance is a need.

3) Interarrival Jitter

Fig. 5 depicts the box charts for the Interarrival Jitter. As we can see there are no extreme outlying observations for the KVM and VirtualBox. For the KVM, the value range is very narrow, not enabling to perform the variance analysis on this data.

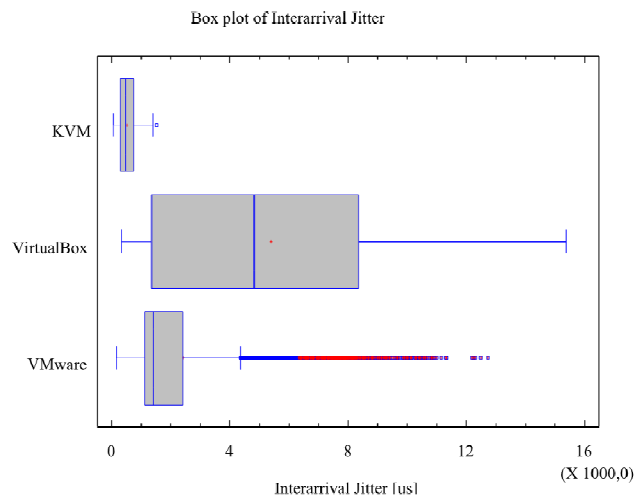


Fig. 5: Boxplot of Interarrival Jitter for all three virtualization tools.

On the other hand, VirtualBox has a very wide range of jitter values. VMware has a high number of outlying observations; one of them being the mean value. This means that in most cases, this virtualization tool functioned well. In some cases, however, the values got unexpectedly high which could have caused problems in voice quality in the real

environment.

B. Classification according to the Number of Processor Cores

This classification describes the properties of the data samples set up by the number of processor cores.

1) Post Dial Delay

On the Fig. 6 we can see the Post Dial Delay data distribution in relation to how many virtual cores were activated during the test run.

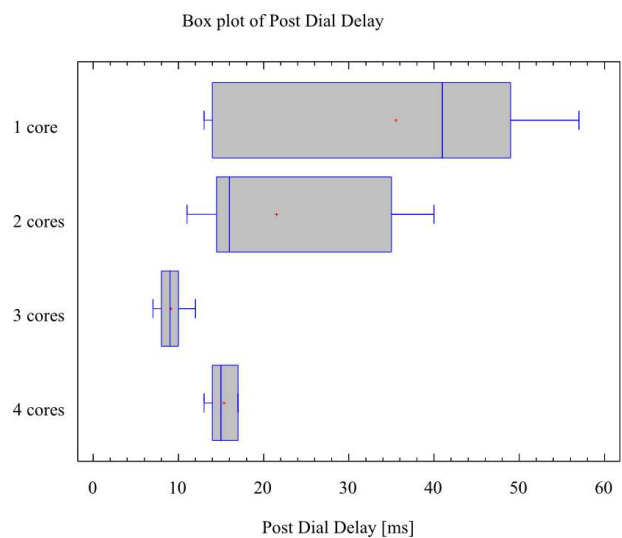


Fig. 6: Boxplot of Post Dial Delay for test runs with 1 4 virtual CPUs.

For all four cases no outlying observation has been identified, From the presented figure we can observe, that one core provides not enough computational power and the delays are therefore distributed over wide area. On the other hand, the three cores seem to provide best performance resulting in narrow data distribution area. Three and four cores have their median values and averages almost collocated indicating the uniform data distribution. According to PDD it is best to run a virtual machine with three or more virtual processor core to provide sufficient computational power allowing for other processes not to interfere with real-time applications.

2) Delay Variation Jitter

Compared to the previous results of Post Dial Delay we now can see on the Delay Variation Jitter results, which are depicted on the Fig. 7, that four two and more virtual CPUs the outlying observation can be identified. This fact indicates that this parameter tends to have spikes during the measurement leading to not uniform results as we have seen with Post Dial Delay.

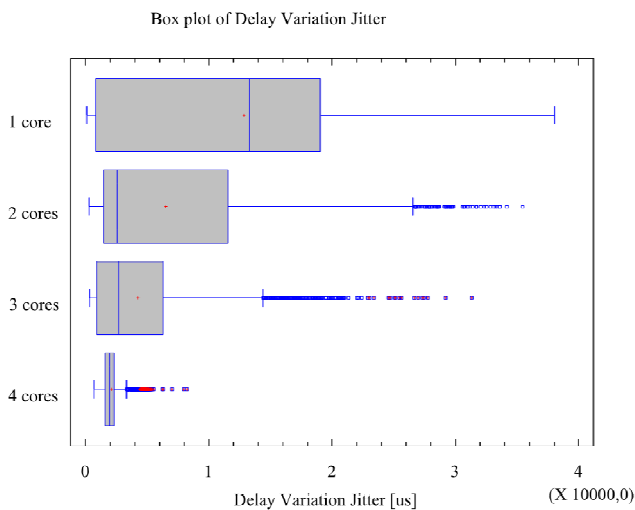


Fig. 7: Boxplot of Delay Variation Jitter for test runs with 1-4 virtual CPUs.

Again we can see the trend of result range narrowing with the increasing number of virtual CPUs leading to best results in case of 4 active virtual processor cores. This case however, has the results spread so close to each other that it is impossible to determine the actual data distribution in the data set.

3) Interarrival Jitter

The Interarrival Jitter has extreme outlying observations when measuring with two and three cores.

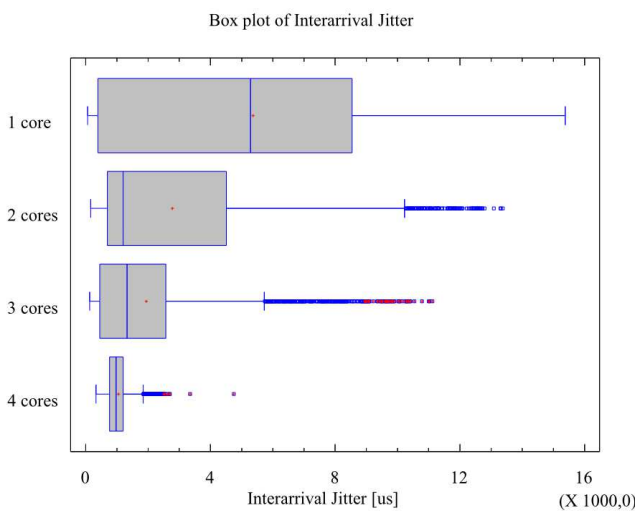


Fig. 8: Boxplot of Interarrival Jitter for test runs with 1-4 virtual CPUs.

When measuring with a single active processor core, the values range high and are evenly distributed as indicated by the position of the median and the mean value. The case with two active processor cores returns some outlying observations and its values are distributed mainly in the lower quartile. Three cores continue with this trend which means the values are distributed within a narrower area and the number of outlying observations is the highest. Four cores have the values

distributed within the narrowest area and with only several outlying observations.

C. Classification according to the Amount of RAM

The RAM capacity was set to 4 different values successively – 512 MB, 1GB, 2GB and 4GB – presenting an independent variable and the values of Interarrival jitter present the depending variable. Figure 9 illustrates the relation between these two sets of variables.

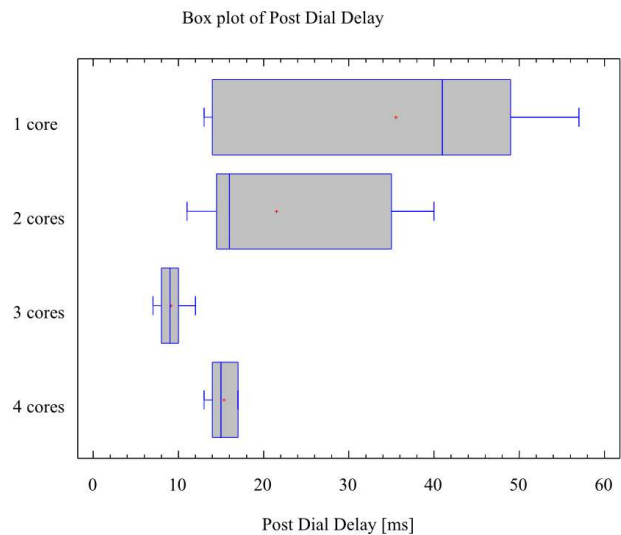


Fig. 9: Boxplot of Interarrival Jitter for different RAM capacity.

The average values and the values of the Interarrival Jitter distribution were almost the same in all four cases. All the cases suffered from a multitude of outlying observations as well. Accordingly, we can assume that virtual machines are memory independent when speaking about the reasonable amounts of memory.

D. Example of Variance Analysis

In this section we are going to explain the steps we have performed in order to confirm some basic assumptions for the statistical analysis. All tables that will be presented in this section contain values of Delay Variation Jitter in relation to the type of the used virtualization tool.

TABLE 6: CHI-SQUARED TEST OF DELAY VARIATION JITTER DATA SET NORMALITY

	Test	Chi-Squared
KVM	P-value	0.0
VirtualBox	P-value	0.0
VMware	P-value	0.0

First we need to find out whether the data set is compliant with the Normal distribution. To be able to do that, we are going to use Chi-squared test, the output of which is as follows

in the table 6. A chi-squared test or χ^2 test, is any statistical hypothesis test in which the sampling distribution of the test statistic is a chi-squared distribution

Since P-value for all the virtualization tools is equal to zero, we can state that data distribution in all cases is compliant with normal distribution.

Then we continue with the homoscedasticity test to find out if the variances of the data sets are equal or not. Two or more normal distributions, $N(\mu_i, \sum_i)$ are homoscedastic if they share a common covariance (or correlation) matrix $\sum_i = \sum_j, \forall i, j$. To confirm this, we use the Bartlett's test. Bartlett's test is used to test the null hypothesis, that all k population variances are equal against the alternative that at least two are different. Let $MS_e = \frac{1}{n-k} \sum_{i=1}^k (n_i - 1) s_i^2$, then MS_e is a residual variance and is applied in a variance analysis (1).

$$C = 1 - \frac{1}{3(k-1)} \left(\frac{1}{n-k} - \sum_{i=1}^k \frac{1}{n-1} \right) \quad (1)$$

If null hypothesis is valid then test statistic (2) returns approximately χ^2 distributions with $k-1$ degrees of freedom and for k samples with size n_i and sample variance s_i^2 Bartlett's test statistic is

$$B = \frac{1}{C} \left[(n-k) \ln MS_e - \sum_{i=1}^k (n_i - 1) \ln (s_i^2) \right] \quad (2)$$

TABLE 7: BARTLETT'S TEST OF DELAY VARIATION JITTER DATA SET HOMOSCEDASTICITY CONFIRMATION

	P-value
Bartlett's test	0.0

Since the null hypothesis assumes that the variances of individual data sets are equal, we can now state that according to the P-value obtained from the Bartlett's test this hypothesis can be rejected. This means for us that we can take the data sets as different and therefore continue with the Kruskal-Wallis test. For k independent observations ranked as $X_{11}, X_{12}, \dots, X_{1n_1}, \dots, X_{k1}, X_{k2}, \dots, X_{kn_k}$ we denote n as the total number of observations across all groups. Then we determine R_{ij} as the rank (among all observations) of observation j from group i and $T_i = \frac{1}{n_i} \sum_{j=1}^{n_i} R_{ij}$. The test statistic is given by (3).

$$Q = -3(n+1) + \frac{12}{n(n+1)} \sum_{i=1}^k \frac{T_i^2}{n_i} - 3(n+1) \quad (3)$$

Using this test can point us in direction of further data analysis and more importantly provide us the information whether median values of individual data sets are equal. Since the P-value obtained from this test is again 0.0 we can reject the null hypothesis and we now know that the median values differ, which can be confirmed from the presented boxplots. With Kruskal-Wallis test done, we can now proceed to post-hoc analysis of this test using the so called Dunn's test.

Dunn's method is used in cases of rejecting the zero hypothesis in the Kruskal-Wallis test. It is used for multiple median comparison and can say whether two chosen data sets differ greatly in their distribution, mainly median. The results of Dunn's method analysis are in the table 8.

TABLE 8: RESULTS OF DELAY VARIATION JITTER ANALYSIS USING THE DUNN'S METHOD.

	$t_i - t_j$	Critical Value
KVM-VirtualBox	4189.91	92.244
KVM-VMware	4130.29	92.244
VirtualBox-VMware	1812.44	92.244

Using the critical value from the table above we can learn that all three pairs differ significantly in their median values.

This way confirmed the properties of all the data sets in this paper, but for the sake of the reasonable size of the paper we do not publish them for all the measured parameters, since the same knowledge can be obtained by more intuitive way from the presented boxplots.

V. CONCLUSION

During the measurement we were careful not to exceed the limiting values of the delays and their variations as defined in the G.114 recommendation by ITU-T and based on practical experience. Although the properties of the test did not allow measuring data traffic between the sender's and receiver's user agents due to the codec translation, it was possible to compare the obtained values of communication between the user agent and the server providing the IP telephony services.

Looking at the results of the exploratory analysis, we can conclude that the pre-test assumptions regarding the virtualization tool performance were correct. The lowest range of values of the Post Dial Delay, Delay Variation Jitter and Interarrival jitter is achieved using the KVM. Although the VMware has a background of large and prosperous company, it did not perform well enough to beat its competitor KVM especially as regards the stability of results. From the real-time application point of view, VirtualBox can be considered as the least efficient and advantageous solution as the values of all

three measured parameters obtained while measuring with this virtualization tool were the worst in every aspect.

Looking at the results, we can also assume that the virtual machines are not memory dependent. Their dependence on the number of processor cores, on the other hand, is rather obvious.

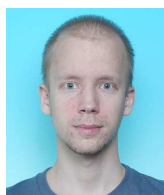
Using other statistical techniques we have confirmed, that the data for different categories (KVM, VMware, VirtualBox; CPU core categories) have different characteristics and their median values do not match. This and other possible interpretation of the results can be read from the presented boxplots.

REFERENCES

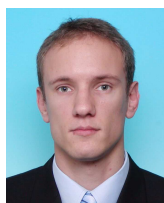
- [1] A. Elenkov, "Virtualization of virtual measurement machines as component of distributed artificial intelligence system," in *Proceedings of the 8th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases*, Cambridge, 2009, pp. 134-138.
- [2] L. Lazic, S. Popovic, N. Mastorakis, "A simultaneous application of combinatorial testing and virtualization as a method for software testing," *WSEAS Transactions on Information Science and Applications*, Volume 6 Issue 11, November 2009, pp. 1802-1813.
- [3] M. Voznak, "E-model modification for case of cascade codecs arrangement," *International Journal of Mathematical Models and Methods in Applied Sciences*, Volume 5, Issue 8, 2011, pp. 1439-1447.
- [4] M. Voznak, M. Halas, B. Borowik, Z. Kocur, "Delay model of RTP flows in accordance with M/D/1 and M/D/2 Kendall's notation," *International Journal of Mathematics and Computers in Simulation*, Volume 5, Issue 3, 2011, Pages 242-249.
- [5] Ixia, Hardware and Reference Manual Release, Part No. 913-1116 Rev. A, 2010. Available: <http://ixiacom.com> (URL)
- [6] J. Rozhon and M. Voznak, "Development of a speech quality monitoring tool based on ITU-T P.862," In *Proceedings 34th International Conference on Telecommunications and Signal Processing*, art. no. 6043771, Budapest, 2011, art. no. 6043771, pp. 62-66.
- [7] M. Voznak, M. Tomes, Z. Vaclavikova and M. Halas, "E-model Improvement for Speech Quality Evaluation Including Codecs Tandeming," In *Proceedings International Conference on Data Networks, Communications, Computers*, Faro, 2010, pp. 119-124.
- [8] Red Hat. *Red Hat Enterprise Linux 6 Virtualization Guide : Guide to Virtualization on Red Hat Enterprise Linux 6*. Raleigh(USA), 2010.
- [9] Oracle Corporation, Oracle VM Virtual Box, p.287, 2011. Available: <http://download.virtualbox.org> (URL)
- [10] L. Rucka, J. Hosek, "Study of VoIP traffic performance under congested MPLS network scenario," In *Conference Proceedings Knowledge in Telecommunication Technologies and Optics*, Szczyrk, Poland, 2011, pp. 41 – 45.



Miroslav Voznak holds position as an associate professor with Department of Telecommunications, Faculty of Electrical Engineering and Computer Science (FE ECS) VSB-Technical University of Ostrava, Czech Republic. He received his M.S. and Ph.D. degrees in telecommunications, dissertation thesis "Voice traffic optimization with regard to speech quality in network with VoIP technology" from the Technical University of Ostrava, in 1995 and 2002, respectively. The topics of his research include next generation networks, IP telephony, speech quality and network security. He is a member of the editorial boards of several journals and conference committees of international scientific conferences, a member of IEEE, the Czech Higher Education Development Fund Council for technical fields and the Scientific board of FE ECS in Ostrava.



Jiri Slachta is a M.S. student with Department of Telecommunications at Faculty of Electrical Engineering and Computer Science, VSB – Technical University of Ostrava. His professional activities are focused on Embedded systems, Networks and Application development for mobile systems.



Jan Rozhon received his M.S. degree in telecommunications from VSB – Technical University of Ostrava, Czech Republic, in 2010 and he continues in studying Ph.D. degree at the same university. His research is focused on performance testing of NGN and in this field he cooperates with CESNET association. He received rector's appreciation for scientific contribution of his diploma thesis in 2010.