COMMUNICATIONS

Miroslav Voznak – Jiri Slachta – Filip Rezac – Jan Rozhon *

EXPLORATORY ANALYSIS OF VIRTUALIZATION TOOLS IN DELAY-SENSITIVE ENVIRONMENT

This article deals with an exploratory analysis of virtualized real-time applications. We compare selected virtualizations tools and present the achieved results in boxplots based on the performed exploratory analysis. The paper analyzes the impact of memory size and the number of processor cores on the delay itself and its variance in selected set of virtualization tools.

Keywords: ANOVA, VMWare, KVM, VirtualBox, Real-time applications.

1. Introduction

Nowadays virtualization presents a solution which can enhance reliability, security, portability and ease the maintenance of computer clusters [1] and [2]. In order to answer the question, how the virtualization techniques influence overall delay of transmission; we chose several free full virtualization tools and compared each other. We found out that the selected virtualizations have different characteristics and their median values do not match. There are many advantages of virtualization but besides them we also need to take into account the drawbacks of the technology, especially now when 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] and [5]. The aim of this article is to find out what impact 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.

2. Virtualization tools

This part presents the three most common virtualization tools: VMware Player, Kernel-based Virtual Machine (KVM) and VirtualBox. Regarding KVM, 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 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 [6] can be exploited. The second VirtualBox is a multiplatform virtualization tool designated to run under OS Windows, Mac OS X, GNU/Linux or Solaris on platforms using most common HW architectures. 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 [7]. The most favoured 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 centres in which high efficiency, performance and scalability is a must.

3. Measuring platform and methodology

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 a hardware assisted virtualization support.

3.1. 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-VTx or AMD-V technology, two incarnations of the mentioned hardware-assisted virtualization technology from

* **Miroslav Voznak, Jiri Slachta, Filip Rezac, Jan Rozhon**
Department of Telecommunications, Faculty of Electrical Engineering and Computer Science, VSB-TU Ostrava, Czech Republic,
E-mail: voznak@ieee.org

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, 8 GB RAM, two 1Gbps NIC and 64-bit operating system Debian Squeeze. The tested topology consists of one computer with SW Asterisk PBX on virtualized platform, one 1Gbps switch and traffic generator Optixia XM2 with IxLoad control SW [8], the situation is depicted in Fig. 1.
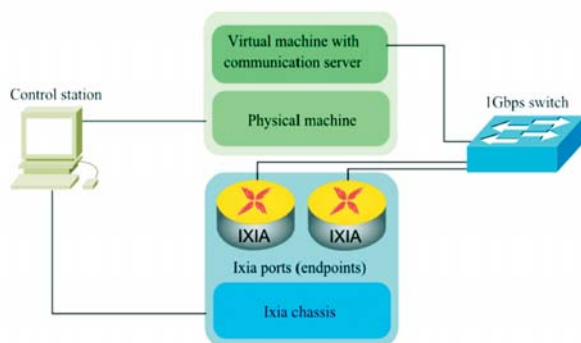

Fig. 1 Test topology

3.2. Measured parameters

Real-time network applications including IP telephony depend on network parameters that influence the transmission quality [9] and [10]. Using the Optixia XM2 we are able to measure Interarrival Jitter, Delay Variation Jitter, One Way Delay, Post Dial Delay, Media Delay and Post Pickup Delay [8]. We define a variation of delay as a jitter. It is the difference between the expected and real time of the packet reception. This appears during the packet transport through the IP network when the time shift between packets occurs because of the queue ordering in routers [10], [11] and [12]. In this article, the following parameters were chosen to carry out measurements on generator and analyzer: Interarrival Jitter, Delay Variation Jitter and Post Dial Delay. 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 observe a difference in performance of real-time applications on various virtualization platforms. 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 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.

3.3. 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. 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.

4. 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 and Interarrival Jitter.

4.1. Classification according to virtual tools performance

Fig. 2 depicts three variables (KVM, VirtualBox, VMware) and their effect on the Post Dial Delay and Delay Variation Jitter. The first variable, KVM, has a 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.

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 said about VMware as well, since the data distribution is similar to VirtualBox except the narrower data range. In the second parameter, Delay Variation Jitter, we can see the similar behaviour.

4.2 Classification according to memory size

The classification according to a memory size describes data properties for four variables which represent individual memory sizes. We had in use following four different values successively of the RAM size 512 MB, 1GB, 2GB and 4GB. Our exploratory analysis showed a low variety of results and a presence of outliers, as is depicted in Fig. 3. According to the results of measurements classified by a memory size, we can assume that virtual machines are memory independent when speaking about the reasonable amounts of memory.
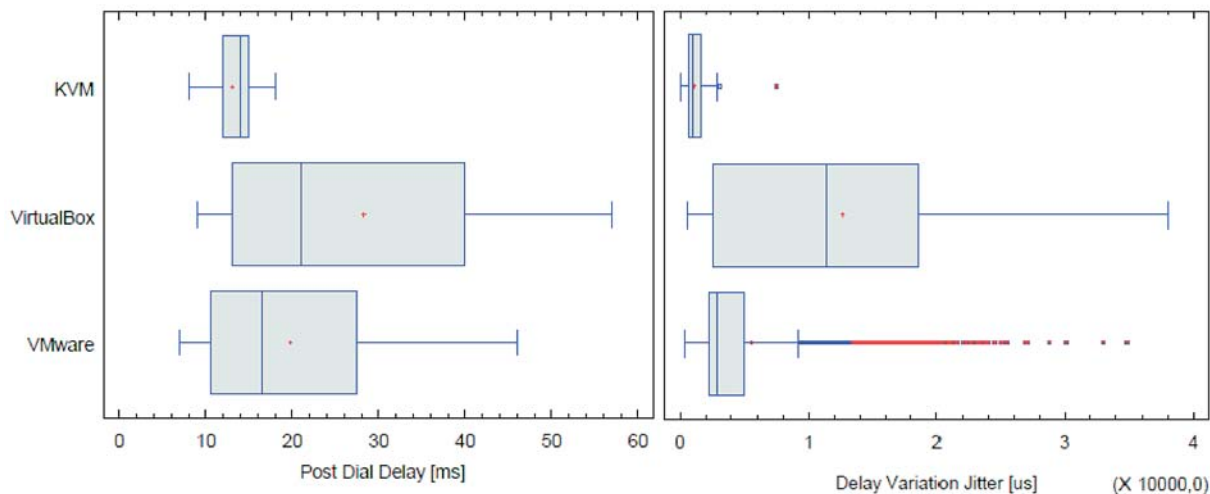
Fig. 2 Boxplot of Post Dial Delay and Delay Variation Jitter for all three virtualization tools
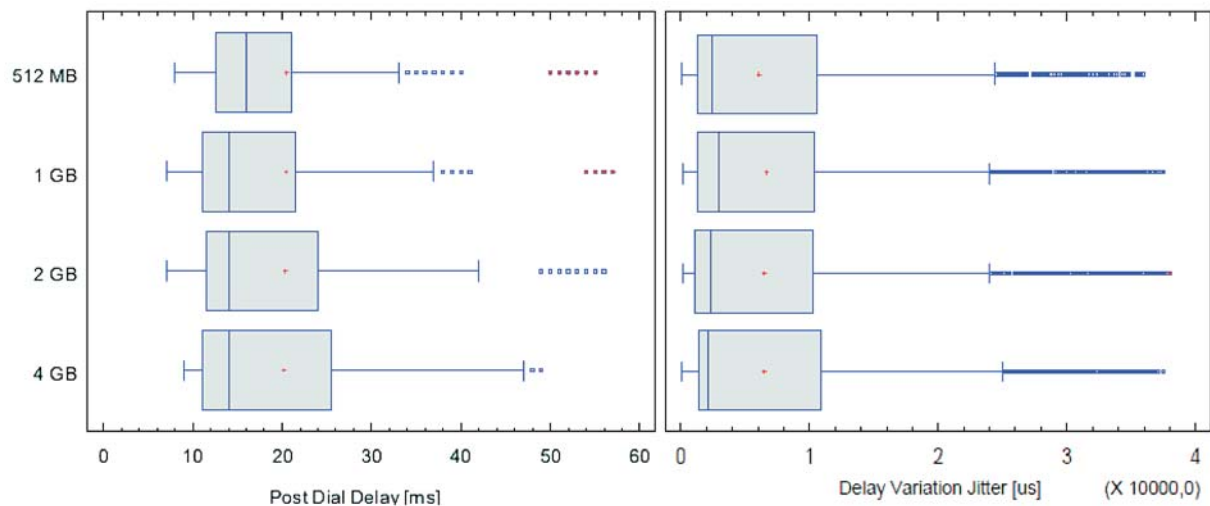

Fig. 3 Boxplot of Post Dial Delay and of Delay Variation Jitter for test runs with 512 MB to 4 GB RAM size

4.3 Classification according to processor cores

The classification according to a number of processor cores describes data properties for four variables which represent individual number of processor cores. We had the following sequence of processor cores: one core, two cores, three cores and four cores. Figure 4 shows the results for Delay Variation Jitter. The number of core processors significantly affects the evaluated results in all the measured parameters. Exploratory analysis was again performed from data set across all the tested platforms.

4.4 Variance analysis

All the tables that are presented in this section contain values of Delay Variation Jitter in relation to the type of the used virtu-
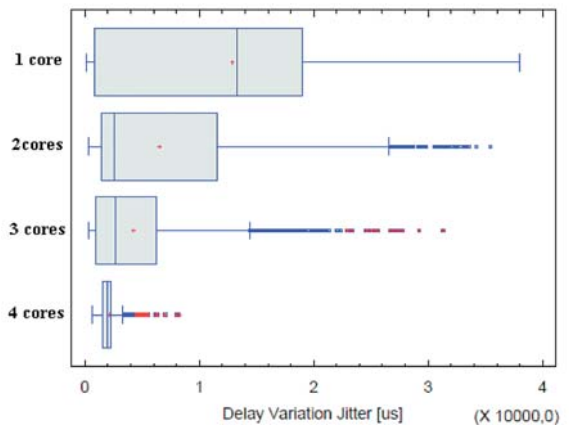

Fig. 4 Boxplot of Delay Variation Jitter for various numbers of cores

alization tool. First, we need to find out whether the data set is compliant with the Normal distribution; the results of chi-squared test are in Table 1.

Chi-Squared Test of Delay Variation Jitter   Table 1
Data Set Normality

	Test	Chi-Squared
KVM	P-value	< 0.01
VirtualBox	P-value	< 0.01
VMware	P-value	< 0.01

A chi-squared test or $\chi2$ 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 the normal distribution. Then we continue with the homoscedasticity test to find out if the variances of the data sets are equal or not. Data variables, $N(\mu_i, \sum_i)$ are homoscedastic if they share a common covariance (or correlation) matrix $\sum_i = \sum_j$, $\forall i,j$. Because all the data variables do not come from the normal distribution, to confirm their homoscedasticity we use the Levene's test instead of Bartlett's test. Levene'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 $Z_{ij} = \left| X_{ij} - \bar{X}_i \right|$, n be the total number of samples and n_i be the number of samples in the i-th group, then denote

$$\bar{Z}_i = \frac{1}{n_i}\sum_{j=1}^{n_i}Z_{ij} \, , \ \bar{Z} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n_i}Z_{ij} \tag{1}$$

$$SS_{ZB} = \sum_{i=1}^{k}n_i\left(\bar{Z}_i - Z\right)^2 , \ \ SS_z = \sum_{i=1}^{k}\sum_{j=1}^{n_i}\left(\bar{Z}_{ij} - Z\right)^2 \tag{2}$$

If the null hypothesis is valid then test statistic returns approximately Fisher-Snedecor distribution with $k-l$ degrees of freedom in the nominator and $n-k$ degrees of freedom in the denominator. Levene's test statistic is expressed in relation (3).

$$\frac{\dfrac{SS_{ZB}}{k-1}}{\dfrac{SS_z}{n-k}} \tag{3}$$

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 Levene's test in Table 2 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 $X_{11}, X_{12}, ..., X_{1n_1} ... X_{k1}$, $X_{k2}, ..., X_{kn_k}$ we denote n as the total number of observations across all the groups. Then we determine R_{ij} as the rank (among all the observations) of observation j from group i, thereby T_i is expressed as their mean value (4) and the test statistic is given by (5).

$$T_i = \frac{1}{n}\sum_{j=1}^{n_i}R_{ij} \tag{4}$$

$$Q = -3(n+1) + \frac{12}{n(n+1)}\sum_{i=1}^{k}\frac{T_i^2}{n_i} - 3(n+1) \tag{5}$$

Levene's Test of Delay Variation Jitter Data   Table 2
Set Homoscedasticity Confirmation

	P-value
Levene's test	< 0.01

Using this test can point us in direction of further data analysis and more importantly provide us with the information whether median values of individual data sets are equal. Since the P-value obtained from this test is 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 hypotheses in the Kruskal-Wallis test. It is used for multiple median comparisons and can say whether two chosen data sets differ greatly in their distribution, mainly median. The results of Dunn's method are presented in Table 3.

Results of Delay Variation Jitter Analysis   Table 3
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 article, but for the sake of the reasonable size of the paper we do not publish them for all the measured parameters.

5. Conclusion

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] BIRKOS, K., CHRYSIKOS, T., KOTSOPOULOS, S.: QoS Issues in Complex-based High Platform Communication Systems, *Communications - Scientific Letters of the University of Zilina*, vol. 9, No. 4, pp. 5–9, 2007.

[2] VOZNAK, M., SLACHTA, J., ROZHON, J.: Performance Analysis of Virtualized Real-Time Applications, *Intern. J. of Mathematical Models and Methods in Applied Sciences,* vol. 6, No. 2, pp. 305–313, 2012.

[3] KLIMO, M., KOVACIKOVA, M., SEGEC, P.: Selected Issues of IP telephony, *Communications - Scientific Letters of the University of Zilina,* vol. 6, No 4, pp. 63–70, 2004.

[4] VOZNAK, M., ROZHON, J.: Methodology for SIP Infrastructure Performance Testing, *WSEAS Transactions on Computers,* vol. 9, No. 9, pp. 1012–1021, 2010.

[5] MONFORT, J., ADLER, K., GIERLICH, H., POMY, J.: Multimedia Quality as Perceived by the User, *Communications - Scientific Letters of the University of Zilina,* vol. 10, No. 4, pp. 5–10, 2008.

[6] Red Hat. Red Hat Enterprise Linux 6 Virtualization Guide: Guide to Virtualization on Red Hat Enterprise Linux 6, 2010. URL https://access.redhat.com/knowledge/.

[7] Oracle Corporation, VM Virtual Box, p.287, 2011. URL http://download.virtualbox.org.

[8] Ixia, Hardware and Reference Manual Release, 2010. URL http://ixiacom.com.

[9] BLASKOVA, L., HOLUB, J.: How do Non-native Listeners Perceive Quality of Transmitted Voice?, *Communications - Scientific Letters of the University of Zilina,* vol. 10, No. 4, pp. 11–14, 2008.

[10] KOVAC, A., HALAS, M., ORGON, M., VOZNAK, M.: E-model MOS Estimate Improvement through Jitter Buffer Packet Loss Modelling, *Advances in Electrical and Electronic Engineering,* vol. 9, No. 5, pp. 233–242, 2011.

[11] VOZNAK, M., TOMES, M., VACLAVIKOVA, Z., HALAS, M.: E-model Improvement for Speech Quality Evaluation Including Codecs Tandeming, *Intern. Conference on Data Networks, Communications, Computers, Faro,* pp. 119–124, 2010.

[12] VOZNAK, M.: E-model Modification for Case of Cascade Codecs Arrangement, Intern. *J. of Mathematical Models and Methods in Applied Sciences,* vol. 5, No. 8, pp. 1439–1447, 2011.