Annales UMCS Informatica AI XIV, 3 (2014) 65–75
DOI: 10.2478/umcsinfo-2014-0015

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI

http://www.annales.umcs.lublin.pl/

Operating system efficiency evaluation on the base of measurements analysis with the use of non-extensive statistics elements

Paweł Dymora[1*], Mirosław Mazurek[1†], Kamil Żelazny[1‡]

[1] *Rzeszów University of Technology, Faculty of Electrical and Computer Engineering, Department of Distributed Systems, ul. Wincentego Pola 2, 35-959 Rzeszów, Poland*

Abstract – The major goal of this article was to evaluate the efficiency of Linux operating system using statistical self-similarity and multifractal analysis. In order to collect the necessary data, the tools available in Linux such as vmstat, top and iostat were used. The measurement data collected with those tools had to be converted into a format acceptable by applications which analyze statistical self-similarity and multifractal spectra. Measurements collected while using the MySQL database system in a host operating system were therefore analyzed with the use of statistical self-similarity and allowed to determine the occurrence of long-range dependencies. Those dependencies were analyzed with the use of adequately graduated diagrams. Multifractal analysis was conducted with the help of FracLab application. Two methods were applied to determine the multifractal spectra. The obtained spectra were analyzed in order to establish the multifractal dependencies.

1 Introduction

Generally, computer systems can be divided into two categories. The first category is defined as simple systems and the second – complex systems. As far as the definition of the complex systems is con-cerned many different explanations of the term are found in technical literature [**1, 2**]. In short, it can be stated that complex system is a system, which is composed of many diverse and autonomic, but mutually dependent and mutually connected components. However, it is important to notice that complex

*pawel.dymora@prz.edu.pl
†mirekmaz@prz.edu.pl
‡123138@stud.prz.edu.pl

systems cannot be described and characterized just by one simple definition. The main characteristics of complex systems are: common application structures and statistical self-similarity in time and space **[3, 4]**. Also, the efficiency planning is dynamic, controlling processes and managing systems resources are closely related and there is no guarantee of the service quality.

2 Statistical self-similarity

Stochastic processes are considered as sequences of variables, which can be characterized by using average value, variance, process probability distribution value and higher stochastic moments. A stationary process, on the other hand, is a stochastic process for which the process probability distribution value does not change. A characteristic element of a part of stochastic processes is the fact, that the values are mutually dependent in time. A value of a given process at a moment t is dependent on the value the process obtained at a moment preceding the moment t. However, such a process can be stationary because a process probability distribution value at a moment t is determined without any premises about the time preceding the process. A process with the long-range dependencies is a process for which the time is infinite. The autocorrelation function for such a process is a slowly-vanishing function. Another value necessary to determine the self-similar processes is a stochastic process X which is presented in the time range m. A stochastic process X, which is visible in a time range m, is a process determined by the following dependence (formula 1):

$$X^m(t) = \frac{1}{m} \int_{m[t]}^{m([t]+1)} X(s)ds. \tag{1}$$

The self-similar process is a static stochastic process for which the dependency, where X is a process in the time range and β is the constant is true (formula 2) **[5, 6]**

$$R_{xm}(k) = m^{-\beta} R_{x(1)}(k). \tag{2}$$

Unfortunately, processes determined by this dependency form a very narrow group and involve, among others, Fractional Brown Motions and some particular multifractal processes. That is why the definition of asymptotic statistical self-similarity proves much more useful.

This definition describes a process in which for every positive value there exist some boundary values k and m, and when the function exceeds them the autocorrelation function $X^{(m)}$ does not vary from the auto-correlation function $X^{(1)}$ for more than one selected positive value. Such process is called an asymptotic self-similar process. The result from the above definition is the fact that an essential value for statistical graduation of self-similar processes is the β exponent, which determines the rate of autocorrelation function vanishing when time range changes. It takes a value close to 1 for short-range correlation processes. The value of modulus fluctuates in a range (0;1). The modulus takes the value 0 for those processes in which autocorrelation functions in

higher time ranges are identical to the autocorrelation function in the basic time range. However, the value of 0 is taken by the modulus for the processes whose realization in individual time ranges is independent. The self-similarity of processes determined by the above definition is justified by their multifractal properties. The autocorrelation function characterizes the time dependencies in a described process. The β exponent is related with the Hurst exponent: $H = 1 - \beta/2$ [**7**]. There are many different ways of calculating it and most of those estimation methods do not need to use the autocorrelation function [**8**].

3 Multifractal spectra

Multifractal spectra can be determined in many different ways. The two basic methods are:

- Legendre transformation of the split function;
- determining spot metering histogram boundaries

The split function can be represented by the following equation, where A is overlay of plane of μ measure and δ is a diameter of the measurement plane (formula 3):

$$S_\delta(q) = \sum_{c \in A} \mu(C)^q. \tag{3}$$

Next, the Legendre transformation function $f : R \to R$ is represented as a transformation of the dependency (formula 4):

$$f^*(s) = \inf_x (sx - f(x)). \tag{4}$$

For the differential functions this transformation can be presented in the formula 5:

$$f^*(s(x)) = x \cdot f'(x) - f(x); \qquad s(x) = f'(x). \tag{5}$$

The multifractal spectrum based on the split function is represented as a Legendre transformation (6):

$$\tau(q) = \lim_{\delta \to \infty} \frac{\log S_\delta(q)}{\log \delta}. \tag{6}$$

As a result of this transformation we obtain a multifractal spectrum based on the split function 7:

$$f_L(\alpha) = \tau^*(x) = q\tau'(q) - \tau(q); \qquad \alpha(q) = \tau'(q). \tag{7}$$

While estimating multifractal spectrum based on the spot metering histogram, for simplification we take into consideration only the probability measurement μ defined in a [0,1] range and its sampling within a space of intervals [**9, 10**]:

$$I_k^n = [k \cdot 2^{-n}, (k+1) \cdot 2^{-n}]. \tag{8}$$

Assuming that $Y_n(\alpha) = \frac{-1}{n*\log 2}\|K_\alpha^n\|$, where $\{x : x = k * 2^{-n}$ i $\alpha(I_k^n) = \alpha$, $k \in \{0, 1, \ldots, 2^n - 1\}\}$, we can define the multifractal spectrum as a spot metering histogram boundary in the following way [**9**]:

$$f_G(\alpha) = \lim_{n\to\infty} Y_n(\alpha). \tag{9}$$

4 Operating system efficiency measurements and statistical analysis

An experiment, which allowed to collect the data necessary to analyze statistical self-similarity and mul-tifractal analysis, concerned MySQL service test hosted in the operating system environment. In order to carry out the test, a free benchmark, which mainly concentrates on the efficiency of converting data and com-mands based on the database system was used. The data was collected for six hours of testing the data-base. In order to measure the efficiency of MySQL, a test diagram with 6.000.000 lines was created. With the help of the built-in efficiency monitor vmstat in the Linux system, an appropriate amount of the data needed to carry out the statistical and multifractal analyses was collected. While the experiment was carried out, we wereable to notice the influence of MySQL load on the operating system and the various system components:

- *bi* – number of data blocks read from the disk;
- *system:in* – number of interruptions [s];
- *system:cs* – number of context switches [s];
- *cpu:us* – processor usage for user program execution [%];
- *cpu: sy* – processor usage for system tasks realization [%].

With the help of the SELFIS application written in Java, the collected data underwent the statistical self-similarity analysis [**11**]. The application has many different options, from the simplest which allows us to analyze the basic statistical values, to the more complex which can estimate the value of Hurst exponent of the given data, thanks to which it is possible to determine the degree of self-similarity of the analyzed data and establish if there are any long-range dependencies. In order to perform the statistical self-similarity analysis of the collected data, four methods were applied: Aggregate Variance, Periodogram, Variance of Residuals and Whittle Estimato.

The first of the analyzed parameters is a number of interruptions per second. To interpret the obtained values of the Hurst exponent we will use two dependencies:

1. If $0.5 < H < 1.0$, long-range dependencies occur, the process has a constant change direction (is persistent) and is stationary.
2. If $H < 0.5$, short-range dependencies occur, the process does not have a constant change direction and is not stationary [**9**].

For the parameter of the number of interruptions per second, each of the four applied methods determined the Hurst exponent as a value higher than 0.5. What stems from

Table 1. Hurst exponent for the system:in parameter (MySQL).

Method of determining H exponent	Value of H exponent
Aggregate Variance	0.552
Periodogram	0.803
Variance of Residuals	0.710
Whittle Estimator	0.725

those results is that there are long-range dependencies in the system and the system is stationary.

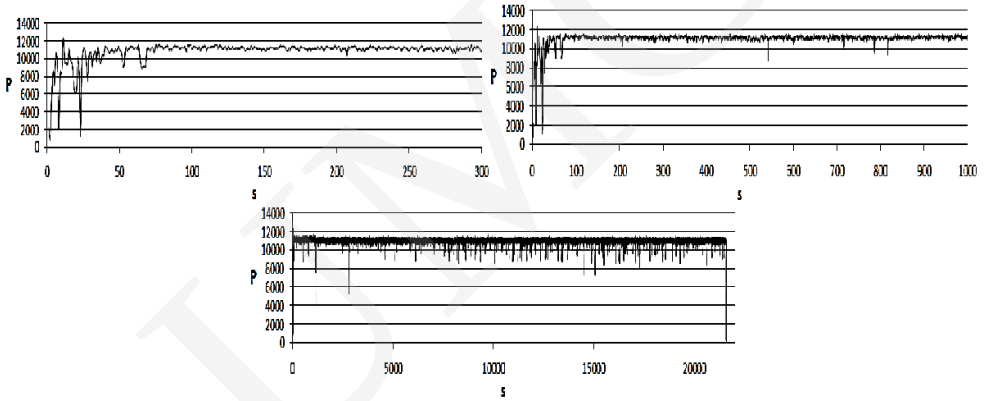

Fig. 1. The example of the process in different time ranges for the system:in parameter.

Fig. 1 shows an exemplary process in different time ranges – for the system:in parameter the time ranges were accordingly selected as: 300 s, 1000 s, 22000 s. The application in the extended time range shows very clearly that this process has a high auto-correlation value.

Fig. 2 presents the process in extended ranges for the following parameter: number of context changes in a second. It allows to observe that despite numerous, substantial changes in the time range, the characteristics of individual processes do not change. It stems from the fact that self-similarity can be noted only when a particular time range boundary is reached.

Each of the four methods applied to the number of context changes in the second parameter, measured the Hurst exponent as a value higher than 0.5. The results are presented in Table 2 and they prove that long-range dependencies occur in a process, and that the process is stationary and has a constant direction change.

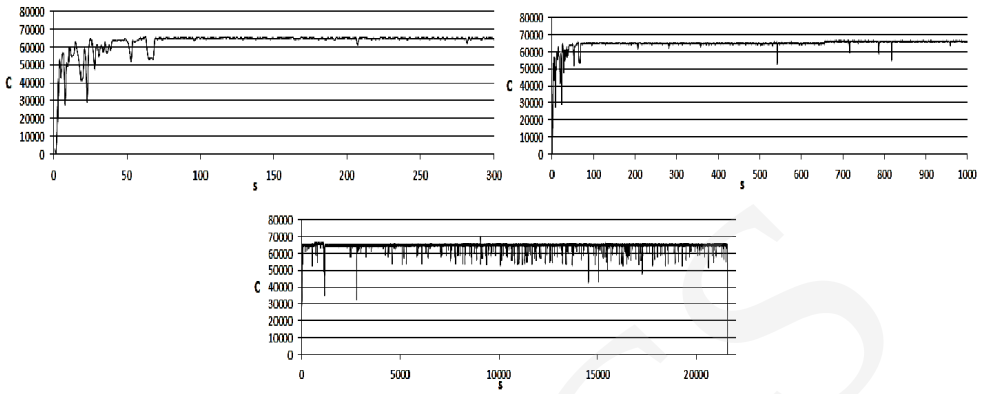

Fig. 2. The example of the process in different time ranges for the system:cs parameter.

Table 2. Hurst exponent for the system:cs parameter (MySQL).

Method of determining H exponent	Value of H exponent
Aggregate Variance	0.743
Periodogram	0.845
Variance of Residuals	0.905
Whittle Estimator	0.837

In the case of processor usage for a user program execution parameter, the extended time range shown in Fig. 3 allows to observe the power vanishing of the process variability measurements when the time range is extended and no changes in individual processes occur.

As far as the processor usage for user program execution is concerned each of the used methods calculated the Hurst exponent as a value higher than 0.5 (Table 3). Therefore it can be stated that the examined system is stationary and long-range dependencies occur.

Table 3. Hurst exponent for the cpu:us parameter (MySQL).

Method of determining H exponent	Value of H exponent
Aggregate Variance	0.538
Periodogram	0.549
Variance of Residuals	0.538
Whittle Estimator	0.538

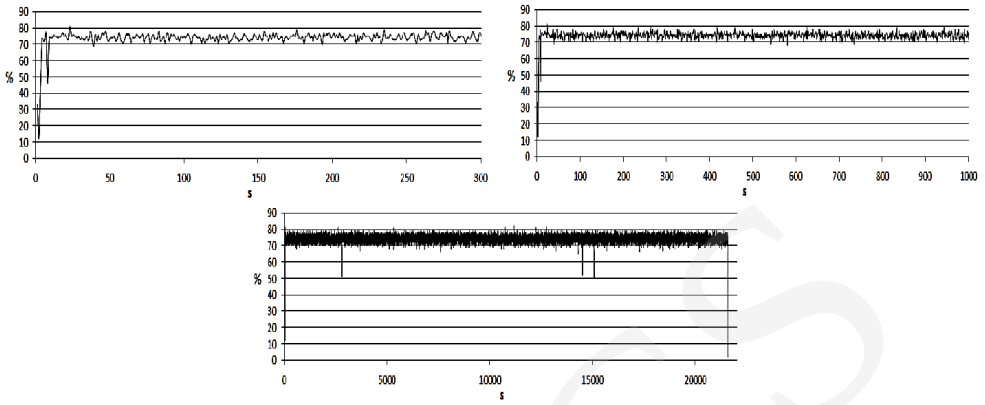

Fig. 3. The example of the process in different time ranges for the cpu:us parameter.

The last parameter for which the Hurst exponent was calculated is processor usage for system tasks re-alization. The value of the Hurst exponent for the predominant number of methods is higher than 0.5. Only the Periodogram method estimated the exponent as minimally lower and that is 0.499 which is shown in Table 4.

Table 4. Hurst exponent for the cpu:sy parameter (MySQL).

Method of determining H exponent	Value of H exponent
Aggregate Variance	0.538
Periodogram	0.499
Variance of Residuals	0.538
Whittle Estimator	0.538

Despite the fact that one of the applied methods determined the Hurst exponent as lower than 0.5, the examined system has shown that it is persistent, stationary and that long-range dependencies occur in it. The fact that there are no changes in the process can be also observed when it comes to the processor usage for the system tasks realization parameter which is shown in Fig. 4. It can be stated that the process is characterized by a high value of auto-correlation function.

5 Multifractal analysis

To conduct the multifractal analysis FracLab 2.1 application written in the Matlab language was used. The application offers two types of analyses. The first is a fractal analysis: a large number of procedures enable the user to calculate various amounts of

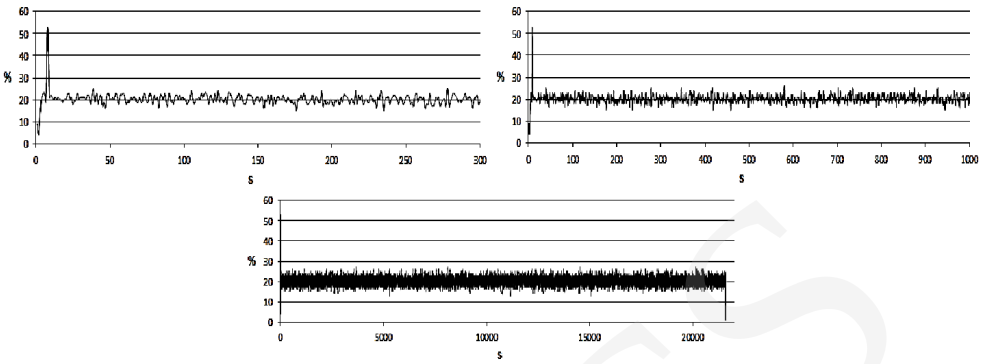

Fig. 4. The example of the process in different time ranges for the cpu:sy parameter.

fractals related to 1D or 2D signals such as: dimension, Holder exponent or multifractal spectra. The second analysis is a signal transformation which can be alternatively used to perform many basics tasks such as evaluation, detection and elimination of interference. For the multifractal analysis a procedure of determining multifractal spectrums was applied. This procedure was obtained by two methods: Legendre Spectrum with box counting algorithm (fL) and Large deviation Spectrum (fG) [**9, 10**]. The first method transforms the split function by applying the Legendre transform, which allows to obtain a multifractal spectrum based on the split function. The second method is based on calculating the boundary of an adequately graduated spot metering histogram. It was observed that the results obtained thanks to this method are inaccurate and difficult to interpret. The first of the analyzed parameters is a number of interruptions per second presented in Fig. 5.

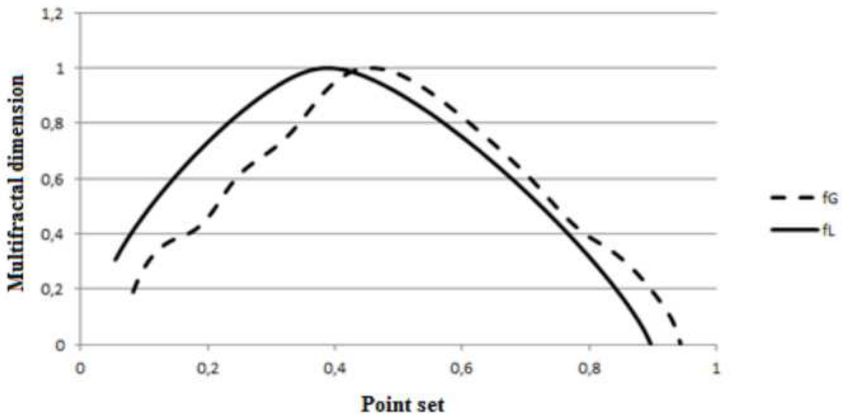

Fig. 5. Multifractal spectra fG and fL for the system:in parameter.

In the case of the analyzed system:in paramater, we can state that the boundary of the histogram shown in the diagram is similar to the presented fL spectrum. That is a very important dependency, as it allows us to reach a conclusion that a process which occurred when the study was conducted is characterized by multifractal properties. Another of the analyzed paramaters is a number of context changes in a second presented in Fig. 6.

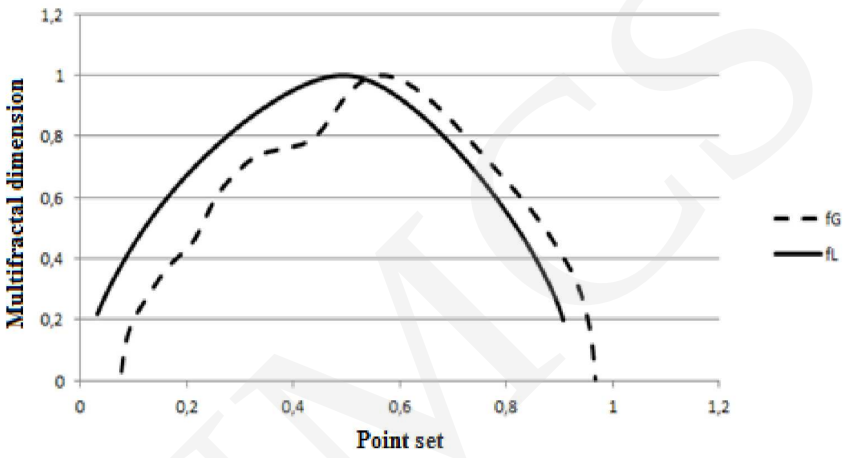

Fig. 6. Multifractal spectra fG and fL for the system:cs parameter.

In Fig. 6, we can see the multifractal spectra fG and fL for the system:cs property. In this case, it can be noted that the histogram boundary is similar to the presented fL spectra and that fG and fL spectra are convergent to each other. It can be also observed in this case that during the carried out study the process was characterized by multifractal properties.

Another two analyzed paramaters were as follows: the processor usage for user program execution and the processor usage for system tasks realization presented in Fig. 7.

While analyzing Fig. 7 it can be noticed that the multifractal spectrum fL does not coincide with the fG spectrum. This declination is difficult to interpret, but we can try to explain it by the fact that values diverging from 1 are extremely rare, which makes determining the value of fG function for such parameters very difficult.

6 Conclusions

This article was mainly concerned with presenting the basis of statistical self-similarity analysis and mul-tifractal analysis. Those analyses were applied in the study of the efficiency of Linux system. The amount of data which was collected allowed to apply the statistical self-similarity analysis and the multifractal analysis in order to determine the dependencies occurring in the system. It was also observed

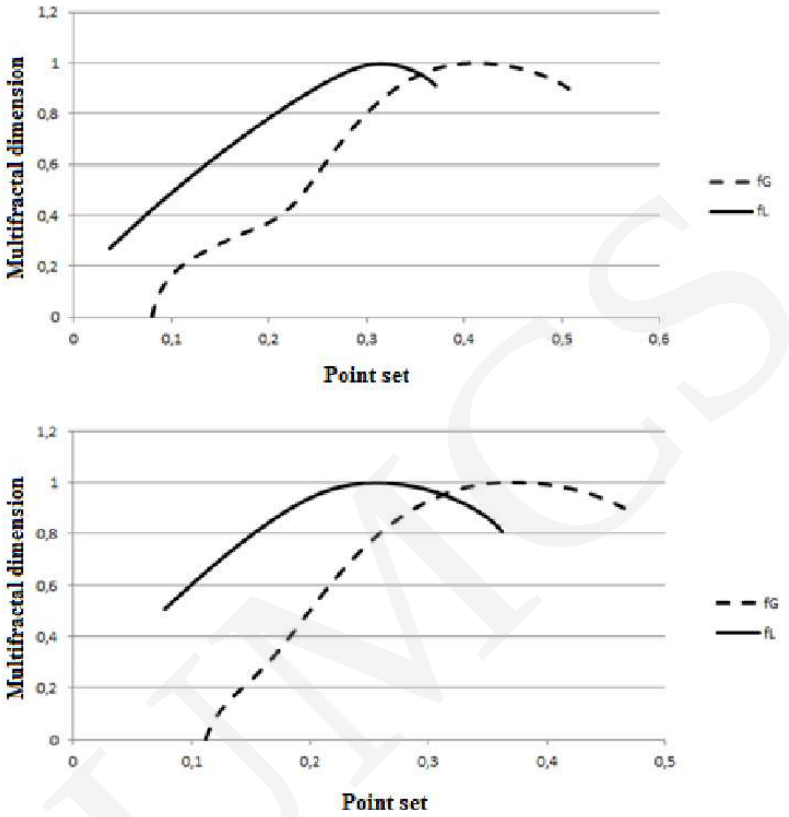

Fig. 7. Multifractal spectra fG and fL for the cpu:us and cpu:sy parameters.

that the processes which occurred while testing the Linux operating system efficiency were characterized by multifractal and long-range dependencies. The appropriate values of the Hurst exponent as well as adequately graded diagrams made it possible to conclude that the tested system is a complex system. The results of analytical considerations and experiment show that the self-similarity factor can be successfully used in the operating system efficiency analysis, and eventually leads to more efficient methods for both failure and degradation efficiency.

References

[1] Strzałka B., Mazurek M., Strzałka D., Queue Performance in Presence of Long-Range Dependencies – an Empirical Study, International Journal of Information Science 2(4) (2012): 47.

[2] Grabowski F., Strzałka D., Simple, complicated and complex systems – the brief introduction. in: 2008 Confer-ence On Human System Interactions, Vol. 1 and 2 (2008): 576.

[3] Field A. J., Harder U., Harrison P. G., Measurement and modeling of self-similar traffic in computer network, IEE Proc. Commun. 151(4) (2004).

[4] Martyn T., Fraktale i obiektowe algorytmy ich wizualizacji. Nakom (1996).
[5] Dymora P., Mazurek M., Strzałka D., Statistical mechanics of memory pages reads during man–computer system interaction, Metody Informatyki Stosowanej 1(26) (2011).
[6] Strzałka D., Non-extensive statistical mechanics – a possible basis for modeling processes in computer memory system, Acta Physica Polonica A 117(4) (2010): 652.
[7] Mazurek M., Dymora P., Network anomaly detection based on the statistical self-similarity factor for HTTP protocol, Przegląd elektrotechniczny, ISSN 0033-2097, R. 90 NR 1/2014 (2014): 127.
[8] Dymora P., Mazurek M., Strzałka D., Piękoś M., Influence of batch structure on cluster computing perfor-mance - complex systems approach, Annales UMCS Informatica XII (1) (2012).
[9] Jędruś S., Modelowanie multifraktalne natężenia ruchu sieciowego z uwzględnieniem samopodobieństwa statystycznego, Telekomunikacja cyfrowa – Technologie i Usługi 4 (2001).
[10] Cheng Q., Agterberg F. P., Multifractal Modeling and Spatial Statistics, Matematical Geology 28(1) (1996).
[11] Dymora P., Mazurek M., Strzałka D., Long-range dependencies in memory pages reads during man-compute system interaction, Annales UMCS Informatica XII (2) (2012).