HIGH-PERFORMANCE TWO-LEVEL PARALLEL COMPUTING SCHEME FOR NANOPARTICLES DETECTION IN SEM IMAGES

M. Yu. Kurbakov[a], V. V. Sulimova[a*]

[a] Laboratory of cognitive technologies and simulation systems,
Tula State University, Lenine Ave. 92, Tula, Russia, 300012 - muwsik@mail.ru, vsulimova@yandex.ru

Commission II, WG II/8

KEY WORDS: Image analysis, parallel computing, SEM-images, small bright objects detection, nanoparticles detection, MPI

ABSTRACT:

In our previous work, an exponential approximation (EA) method was proposed for the detection of nanoparticles in scanning electron microscope (SEM) images. It shows the best quality of nanoparticle detection compared to other methods. But its main drawback is that it takes a lot of time. In this paper, we propose a two-level parallel computing scheme and a corresponding high-performance python+MPI implementation of the EA method. Experiments have shown that the developed parallel implementation can significantly speed up the computational process.

1. INTRODUCTION

Electron microscopy is one of the methods for studying the microstructure of solids, their electric and magnetic fields, and local composition using a combination of electron probe methods. Today it is one of the most effective and advanced research methods, which is widely used at enterprises, in scientific and educational laboratories. The resolving power of scanning electron microscopes (SEM) makes it possible to obtain an increase of millions of times, which is ideal for studying microscopic objects (Klein et al., 2012).

Scanning electron microscopy can be used to solve many urgent practical problems. In particular, this paper deals with a problem of mapping ”hidden” defects on the surface of carbon materials that cannot be detected by other methods (Boiko, 2021). The solution of this problem is very important for many industrial processes (Mao, 2021), including processes using such an important catalyst as palladium on carbon (Liu, 2018; Felpin, 2006), since the presence of defects affects the properties of catalysts and their ability to accelerate chemical processes.

The mapping methodology consists in applying metal nanoparticles 1–5 nm sized as markers on the surface under study and then photographing it with an electron microscope. A direct relationship between the presence of defects and the ordering of nanoparticles was studied experimentally in (Boiko , 2021). Due to the predominant association of nanoparticles with defects, SEM-images obtained in this way correspond with sufficient accuracy to the distribution of defects (Fei, 2013, Pentsak, 2015).

The problem consists in that, chemists analyze SEM-images manually. At that, hundreds and even thousands of images are formed to study only one sample of the material. It makes manual analysis too time-consuming and requires automation of detection and analysis of the relative position of nanoparticles in images.

Visually, nanoparticles are small bright areas, usually rounded. The complexity of their automatic detection lies in the fact that the images, as a rule, are very noisy, and the nanoparticles themselves are very small in size, and, moreover, several nanoparticles can be located very close and even partially overlap each other.

In our previous work (Boiko, 2022) the Exponential approximation (EA) method was proposed to detect nanoparticles. According to experiments, it showed the best quality of nanoparticle detection compared to other methods. The main disadvantage of EA method consists in that it is time-consuming.

In this connection we propose here a two-level parallel computing scheme and a corresponding high-performance python+MPI implementation of the EA method. The choice of software implementation tools was significantly limited by the python programming language, in which the sequential version of the EA algorithm was previously implemented.

Experiments have shown that the developed parallel implementation can significantly speed up the computational process.

2. EXPONENTIAL APPROXIMATION (EA) METHOD FOR NANOPARTICLES DETECTION

The exponential approximation (EA) method is based on two assumptions. The first assumption is that the visual representation of each nanoparticle in the image can be modeled quite well using an exponential brightness function with a maximum at the center of the nanoparticle. The second assumption is that the parameters of the approximating functions found for small image fragments will differ for fragments containing and not containing a nanoparticle.

EA method for one image processing consists of five stages: preprocessing, extracting small image fragments, performing exponential approximation for each image fragment, detecting fragments containing a nanoparticle, and determining the radius of each particle.

2.1 Preprocessing

To exclude too bright (exposed) areas of the image, Top-Hat (Bright, 1987) is performed before the main processing.

[*] Corresponding author.

Publisher's note: Copernicus Publications has not received any payments from Russian or Belarusian institutions for this paper.

This contribution has been peer-reviewed.

2.2 Extracting small image fragments

According to EA method, at this stage, square image fragments of a small fixed size $fsize$ are selected, the center of which corresponds to the supposed centers of the nanoparticles (with the exception of fragments close to the image borders - they are truncated and have a displaced center). To reduce the number of calculations, a threshold is used to pre-filter fragments Th_{pref}. Fragments, the brightness at the central point of which is below this threshold, are excluded from consideration due to the low probability that the center of the nanoparticle has a low brightness value. The threshold Th_{pref} is a structural parameter of the method.

2.3 Performing exponential approximation for each image fragment

Let $f = [f(x_i, y_j), i, j = 1, ..., fsize]$ be a small image fragment having size $fsize$. For all fragments, we use the same element indexing $x_i, y_j \in \{-\lfloor fsize/2 \rfloor, \lfloor fsize/2 \rfloor\}$, constructed with respect to the center with coordinates
$x_{\lfloor fsize/2 \rfloor} = 0$ and $y_{\lfloor fsize/2 \rfloor} = 0$.

To approximate each fragment, an exponential function of the form

$$F(x, y, a, b) = ae^{-b(x^2+y^2)} \qquad (1)$$

is used.

The approximation is carried out in two steps. At the first step, for each value of the parameter b from some finite set $b \in \{b_1, .., b_n\}$ the optimal value of the parameter is determined as an argument that provides the minimum value of the root-mean-square approximation criterion

$$a(b_k) = \arg\min_{a} J(x, y, a, b_k),\ k = 1, .., n, \qquad (2)$$

where $J(x, y, a, b_k) = \sqrt{\sum_{i,j} (F(x_i, y_j, a, b_k) - f(x_i, y_j))^2}$.

In this case, according to the least squares method, the optimal value of the parameter a can be easily calculated by the formula:

$$a(b_k) = \frac{\sum_{i,j} f(x_i, y_j) e^{-b_k(x_i^2+y_j^2)}}{\sum_{i,j} \left(e^{-b_k(x_i^2+y_j^2)}\right)^2}. \qquad (3)$$

At the second step, those values of the parameters b_k and $a(b_k)$, which provide the minimum value of the criterion, are taken as the optimal values of the desired parameters

$$\begin{aligned} a^{opt} &= a(b^{opt}), \\ b^{opt} &= \arg\min\nolimits_{b_k} J(x, y, a(b_k), b_k),\ k = 1, ..., n \end{aligned} \qquad (4)$$

of approximating function 1.

2.4 Detecting fragments containing a nanoparticle

As a result of exponential approximation of image fragments, we get two matrices of coefficients $\mathbf{a}^{opt} = [a_{i,j}^{opt}]$ and $\mathbf{b}^{opt} = [b_{i,j}^{opt}]$, $i = 1, ..., sizeX, j = 1, ..., sizeY$, the size of which matches the size of the analyzed image.

To determine the fragments containing nanoparticles, we find all local maxima of the $\mathbf{a}^{opt}$ matrix and take only those of them whose value exceeds the value of the adaptive threshold

$$Th_{detect} = Cf_{detect} mean(LM(\mathbf{a}^{opt})), \qquad (5)$$

where $mean(LM(\mathbf{a}^{opt}))$ is the average value of all found local maxima and the coefficient $Cf_{detect} \in [0; 1]$, which is another structural parameter of the exponential approximation method.

2.5 Determining the radius of each particle

The radius of the nanoparticle centered at the point (i, j) is determined by the corresponding value $b_{i,j}^{opt}$ according to the formula:

$$r_{i,j} = \frac{1}{\sqrt{b_{i,j}^{opt}}}. \qquad (6)$$

3. TWO-LEVEL PARALLEL COMPUTING SCHEME

3.1 Level 1: Parallel detecting nanoparticles in a single image

It is easy to see that stages 2, 3 and 5 of the exponential approximation method that is described in the previous section are a set of independent calculations aimed at processing individual fragments of the analyzed image. However, the organization of parallel computing turns out to be appropriate only for the most time-consuming third stage - the stage of directly exponential approximation, while the computational complexity of the second and fifth stages is low, as a result of which the overhead costs associated with the organization of parallel computing at these stages turn out to be higher than the possible effect of parallelization. The corresponding scheme of parallel computing is shown in Figure 1.

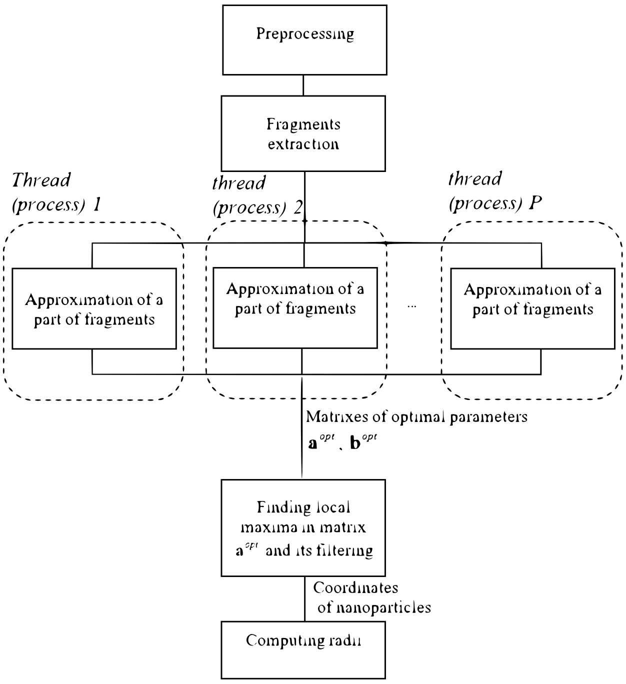
Preprocessing
Fragments extraction
Thread (process) 1
thread (process) 2
thread (process) P
Approximation of a part of fragments
Approximation of a part of fragments
...
Approximation of a part of fragments
Matrixes of optimal parameters $\mathbf{a}^{opt}$, $\mathbf{b}^{opt}$
Finding local maxima in matrix $\mathbf{a}^{opt}$ and its filtering
Coordinates of nanoparticles
Computing radii

Figure 1: Parallel computing scheme at the level of detecting nanoparticles

3.2 Level 2: Parallel SAM images processing

It should be noted that applied research may require the simultaneous processing of hundreds and even thousands of images, while saving the processing results in the form of text and graphic files. A large number of images requiring processing, together with the low disk performance while saving the results, makes it relevant to use additionally the second level of parallelism, which corresponds to parallel image processing.

3.3 Software implementation of a two-level scheme of parallel computing

Important condition for the development of a parallel implementation of the exponential approximation method, which impose significant restrictions on the structure of a parallel computing scheme and on the choice of implementation tools, was the using python programming language, which is the language for implementing the sequential version of the algorithm.

Since the distribution of intensive computations between python threads does not provide performance improvement because of the global interpreter lock (GIL), within the framework of this work, parallelization using processes is used at both of levels.

Namely, at the level 1 the python multiprocessing pool is used. This implementation of multiprocessing is convenient for use when there is a set of tasks, each of which can be solved independently of the others. In this case, such tasks are the tasks of exponential approximation for each of the selected small fragments of the analyzed image at the level 1.

To organize the parallel execution of exponential approximation of individual image fragments, a task pool is formed, in which each task is characterized by initial data (image fragment), as well as a set of parameters and the results of some auxiliary computations. Due to the overhead costs of organizing parallel computing, it turns out to be more expedient to use the number of processes, which is significantly less than the number of tasks, distributing the work among the processes as evenly as possible. After the completion of the calculations, the results of solving problems for each problem (in this case, the coordinates of the center of the processed fragment and the found optimal values of the approximation parameters) are stored in a common list in the space of the main process. After the list is fully formed, the information contained in it is used to fill in the matrices of optimal parameter values, on the basis of which, in the next steps, fragments containing nanoparticles are determined and their radii are calculated.

To implement parallel image processing at the level 2 Message Passing Interface (MPI) technology is used. It should be noted that the between-process information exchange in this case (in contrast to the level 1) is not required. And MPI technology actually is used to start independent tasks of image processing, ensuring the most uniform processors loading.

4. EXPERIMENTS

4.1 Characteristics of computing systems

An experimental study of the developed parallel software tool was carried out on two computing systems, the main characteristics of which are given in Table 1

Characteristic	System1	System2
CPU model	Intel Haswell-EP E5-2697v3	Intel Xeon Platinum 8255C
CPU frequency	2.6 GHz	2.5 GHz
Number of cores	14	24
RAM	64 Gb	62 Gb

Table 1: Main characteristics of computing systems

4.2 Performance study of parallel detection of nanoparticles on a single image

The operation time of EA method depends on the number of nanoparticles in the image. In this regard, three types of SEM-images, containing a significantly different amount of nanoparticles, were chosen for the experiments (Figure 2).

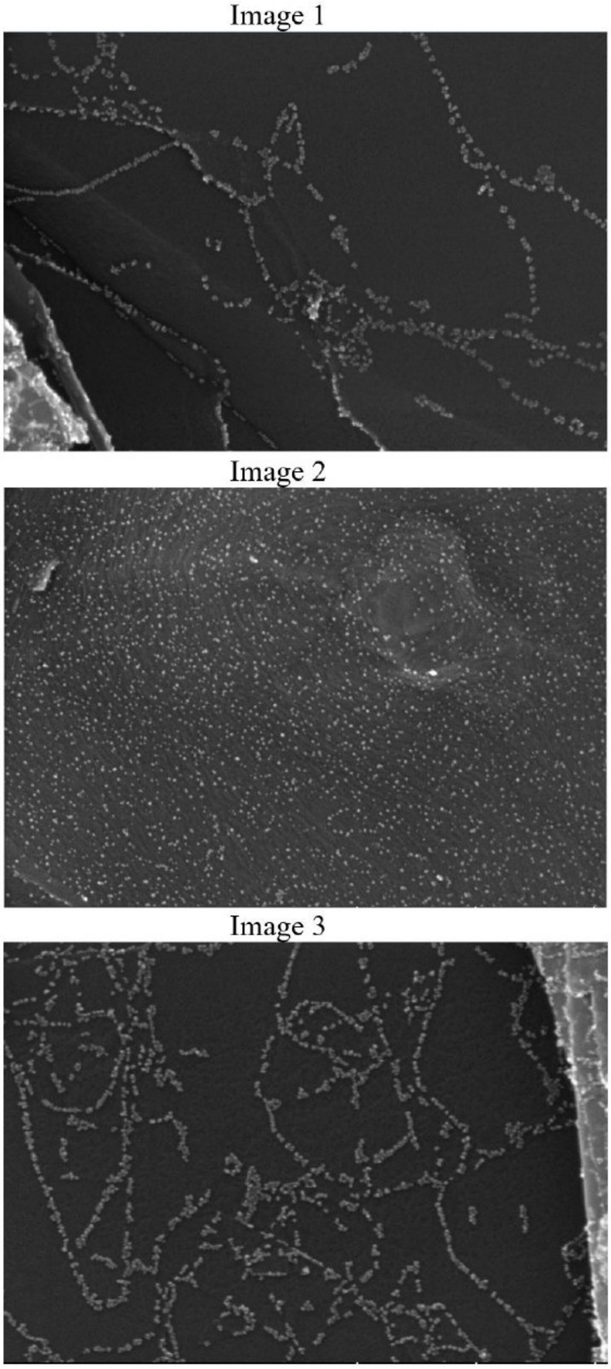

Figure 2: SEM-images to study performance of parallel detecting nanoparticles

In all experiments of this paper, the exponential approximation method was run with parameters that ensured the best quality of nano-particle detection according to preliminary studies (Boiko, 2022): fragment size for approximation $fsize = 7$; possible values of nanoparticle radii from 1 to 7 with a step of 0.1; threshold for pre-filtering fragments $Th_{pref} = 10$; adaptive coefficient for detecting fragments containing nanoparticles $Cf_{detect} = 0.4$. All experiments of this section were made on the System1 whose characteristics are presented in the Table 1.

For each image, EA method was run using a different number of processes $p \in \{1, 2, 4, 6, 8, 10, 12, 14, 16\}$. For each run, the time of the method was fixed. The quality of nanoparticle detection does not change with a change in the number of processes, therefore, in this case, it is of no interest.

Table 2 contains the average running times of the proposed algorithm over 5 runs, as well as obtained speedup S and efficiency E of the use of processor time.

The speedup shows the ratio of the execution time of the best sequential algorithm to the execution time of the parallel algorithm (for a fixed number of processes) and is calculated by the formula:

$$S = \frac{T_1}{T_p}, \tag{7}$$

where T_1 is the running time of the sequential algorithm, T_p is the running time of the parallel algorithm on p processes.

Another important characteristic of a parallel computing process is efficiency. It shows the ratio of speedup to the corresponding number of processes:

$$E = \frac{S}{p}, \tag{8}$$

where S is the algorithm speedup for p processes.

Efficiency can take values from 0 to 1. The ideal efficiency value is $E = 1$, which means that all processors are fully loaded throughout the entire computational process.

Image	p	time(sec)	Speedup	Efficiency
1	1	109.4	-	-
	2	60.15	1.82	0.91
	4	31.78	3.45	0.86
	6	21.85	5.01	0.84
	8	17.17	6.38	0.80
	10	14.49	7.56	0.76
	12	12.43	8.81	0.73
	14	12.57	8.71	0.62
2	1	349.36	-	-
	2	170.32	2.05	1.03
	4	97.48	3.58	0.90
	6	67.62	5.17	0.86
	8	52.03	6.71	0.84
	10	43.05	8.12	0.81
	12	35.09	9.96	0.83
	14	35.63	9.81	0.70
3	1	237.34	-	-
	2	115.38	2.06	1.03
	4	61.08	3.89	0.97
	6	43.33	5.48	0.91
	8	33.06	7.18	0.90
	10	26.80	8.86	0.89
	12	22.99	10.32	0.86
	14	23.84	9.96	0.71

Table 2: Performance evaluation results for parallel nanoparticale detection in a single image

The obtained dependencies are graphically presented in Figure 3.

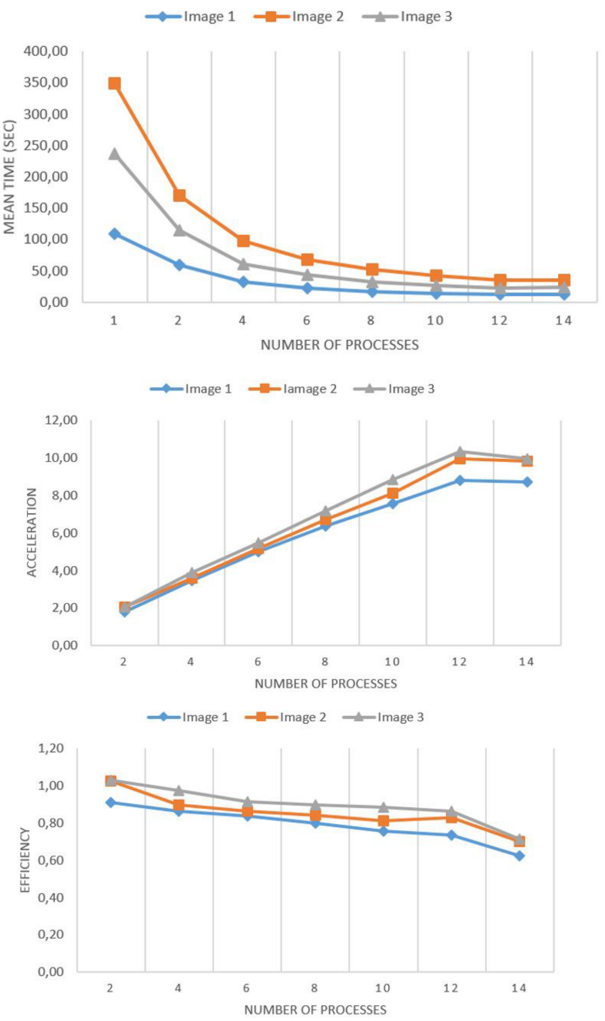

Figure 3: Dependencies of average time, acceleration (speedup) and efficiency on the number of processes for parallel nanoparticles detection

As we can see, in all cases the average running time decreases significantly with an increase in the number of processes used up to 12, providing a fairly high speedup (almost 10 times) and good efficiency in using processor time (the best possible value is 1). A further increase in the number of processes turns out to be inexpedient due to presence of sequential parts and an overhead of collecting results obtained by different processes.

4.3 Performance study of image pool processing

In this experiment we study performance of the proposed parallel algorithm with two levels of parallelism.

SEM images data set of 322 images of material S1 of data base (Boiko, 2020) was used.

For all parameters of the EA method, the same values were set as in the experiment from section 4.2.

For the full data set of 322 images the proposed method was run with different number of processes p_1 (for level 1) and p_2 (for level 2) on the System2, whose main characteristics are specified at the Table 1. The resulted times (in minutes) and the respective speedups that were computed relatively to the sequential processing time of 1239,70 minutes are presented at the Table 3.

$p_2 = 4$	p_1					
	1	2	4	6	8	12
Time (min)	292.8	151,7	104,1	74.3	65.0	59.2
Speedup	4.2	8,2	11,9	16.7	19.1	20.9
$p_2 = 6$	p_1					
	1	2	4	6	8	12
Time (min)	200.8	113.2	68.4	55.9	53.8	53.4
Speedup	6.2	11.0	18.1	22.2	23.0	23.2
$p_2 = 12$	p_1					
	1	2	4	6	8	12
Time (min)	107.3	58.2	52.4	51.4	51.2	51.2
Speedup	11.6	21.3	23.6	24.1	24.2	24.3
$p_2 = 24$	p_1					
	1	2	4	6	8	12
Time (min)	61.7	57.5	53.8	53.1	51,2	51,1
Speedup	20.1	21.6	23.0	23.3	24,2	24,3
$p_2 = 48$	p_1					
	1	2	4	6	8	12
Time (min)	70,3	58,6	54,5	53,8	53,1	52,7
Speedup	17,6	21,1	22,7	23,0	23,3	23,5

Table 3: Performance evaluation results for parallel nanoparticale detection in a single image

Figures 4 and 5 presents respectively dependencies of time (in minutes) and speedup on the number of processes p_2 at the level of image processing pool. Different charts at these figures corresponds to different number of processes p_1 at the level of detecting naoparticles in a single image. Figures 6 and 7 presents respectively dependencies of time (in minutes) and speedup on the number of processes p_2 at the level of image processing pool. Different charts at these figures corresponds to different number of processes p_1 at the level of nanoparticles detection in a single image.

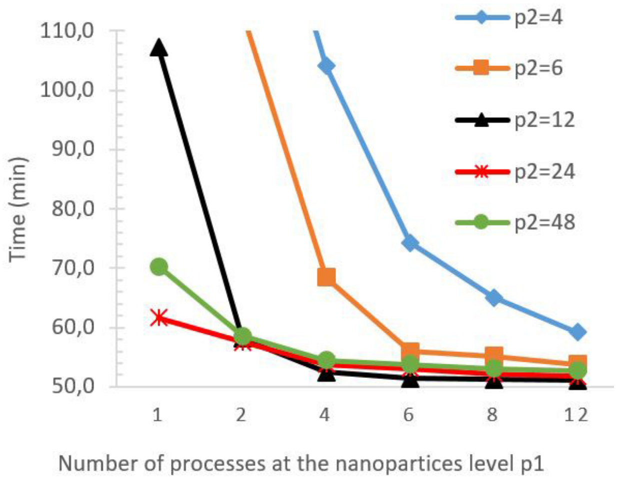

Figure 4: Dependencies of time on the number of processes at the level of detecting nanoparticles

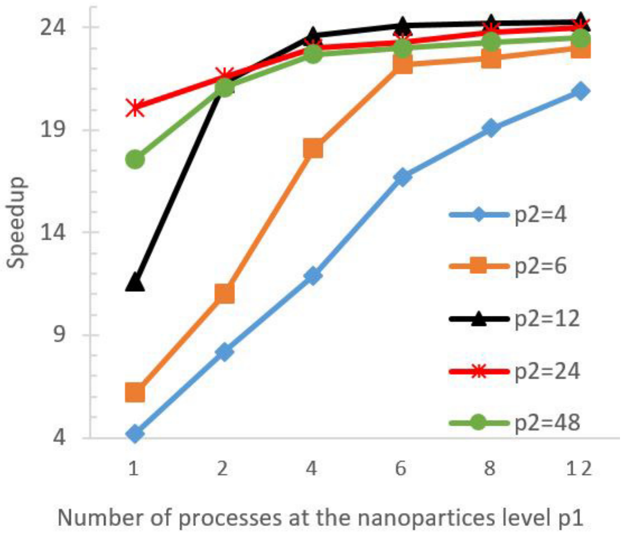

Figure 5: Dependencies of speedup on the number of processes at the level of detecting nanoparticles

As we can see from the Table 3 and figures 4 - 7, the organization of parallel computing both at the stage of detecting nanoparticles and at the stage of processing the pool of images makes it possible to increase the performance of the developed software and, accordingly, significantly reduce the time required for processing the pool of images. At the same time, it should be noted that the greatest speedup can be achieved due to the proposed and implemented in this paper the two-level parallel computing scheme. While the use of parallel processing only at one of the levels, although it increases performance, is still inferior to a two-level scheme. The highest acceleration is approximately equal to 24, which corresponds to the number of cores in the system and, accordingly, the maximum number of parallel processes. Thus, the proposed two-level system allows us to fully use all computing resources of the system and get the maximum possible acceleration.

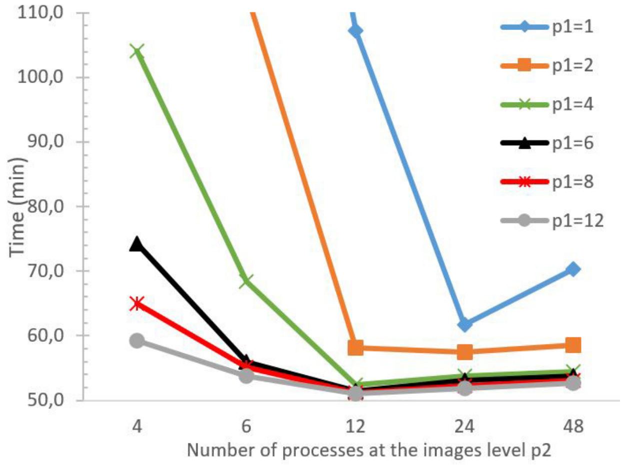

Figure 6: Dependencies of time on the number of processes at the level of image processing pool

Interestingly, for a fixed number of processes p2 at the image level, an increase in the number of processes p1 at the level of nanoparticles (even if the total number of processes exceeds the available number of processor cores) makes it possible to reduce the total operating time . However, the optimal number of processes at each level, according to this experimental study, is 12.

5. CONCLUSION

This paper proposes a two-level parallel computing scheme and a corresponding high-performance python+MPI implementation of the Exponential Approximation method for anoparticles detection in scanning electron microscope images.

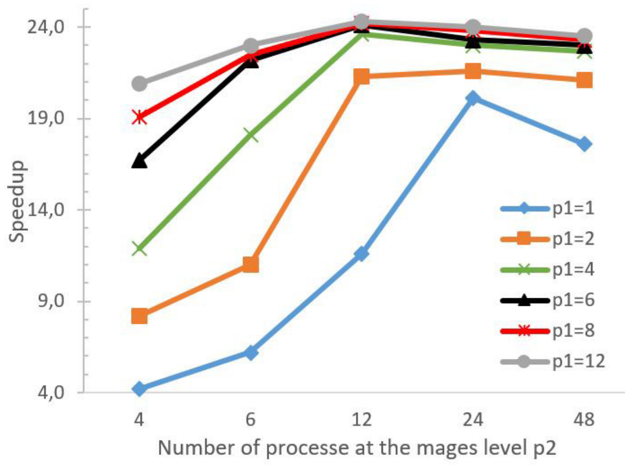

Figure 7: Dependencies of speedup on the number of processes at the level of image processing pool

Experiments on the real data set of 322 SEM images on the the high-performing server and the supercomputer system (Voevodin, 2019) show that the proposed computing scheme and its implementation allows essentially reduce the computing time.

At that the highest acceleration corresponds to the number of cores in the system and, accordingly, the maximum number of parallel processes. Thus, the proposed two-level computing scheme allows us to fully use all resources of the computing system and get the maximum possible acceleration.

ACKNOWLEDGEMENTS

The authors thank the Scientific School of Academic V.P. Ananikov for the research topic, useful discussions and provided experimental data.

This research is funded by the Ministry of Science and Higher Education of the Russian Federation within the framework of the state task FEWG-2021-0012.

Experiments were partially made using the equipment of the shared research facilities of HPC computing resources at the Lomonosov Moscow State University.

REFERENCES

Boiko, D.A., Pentsak, E.O., Cherepanova, V.A., Ananikov, V.P., 2020. Electron Microscopy Dataset for the Recognition of Nanoscale Ordering Effects and Location of Nanoparticles — Dataset 1 (ordered). figshare, https://doi.org/10.6084/m9.figshare.11783661

Boiko, D.A., Pentsak, E.O., Cherepanova, V.A., Gordeev, E.G., Ananikov, V.P., 2021. Deep neural network analysis of nanoparticle ordering to identify defects in layered carbon materials. *Chem. Sci.* 12, 7428–7441.

Boiko, D.A., Sulimova, V.V., Kurbakov, M.Y., Kopylov, A.V., Seredin, O.S., Cherepanova, V.A., Pentsak, E.O., Ananikov, V.P., 2022. Automated Recognition of Nanoparticles in Electron Microscopy Images of Nanoscale Palladium Catalysts. *Nanomaterials*. 12(21):3914. https://doi.org/10.3390/nano12213914.

Bright, D. S. and Steel, E. B., 1987. Two-dimensional top hat filter for extracting spots and spheres from digital images. *J. Microsc.* 146, 191–200.

Fei, Z. et al., 2013. Electronic and plasmonic phenomena at graphene grain boundaries. *Nat. Nanotechnol.* 8, 821–825.

Felpin, F., Ayad, T., Mitra, S., 2006. Pd/C: An Old Catalyst for New Applications – Its Use for the Suzuki–Miyaura Reaction. European J. Org. Chem. 2006, 2679–2690.

Klein, T., Buhr, E.; Frase, C. G., 2012. TSEM: A Review of Scanning Electron Microscopy in Transmission Mode and Its Applications. *Advances in Imaging and Electron Physics*. Vol. 171. pp. 297–356. doi:10.1016/B978-0-12-394297-5.00006-4. ISBN 9780123942975

Liu, X., Astruc, D., 2018. Development of the Applications of Palladium on Charcoal in Organic Synthesis. *Advanced Synthesis and Catalysis*, doi:10.1002/adsc.201800343.

Mao, Z., Gu, H., Lin, X., 2021. Recent Advances of Pd/C-Catalyzed Reactions. *Catalysts* 11, 1078.

Pentsak, E. O. et al., 2015. Spatial imaging of carbon reactivity centers in Pd/C catalytic systems. *Chem. Sci.* 6, 3302–3313.

Voevodin, Vl., Antonov, A., Nikitenko, D., Shvets, P., Sobolev, S., Sidorov, I., Stefanov, K., Voevodin, Vad., Zhumatiy, S. Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community,2019. *Supercomputing Frontiers and Innovations* Vol.6, No.2, 4–11. https://doi.org/10.14529/jsfi190201.