Usage of a Computer Cluster for Physics Simulations Using Bullet Engine and OpenCL

The study focuses on using a computer cluster for implementing real-time physical simulations, responding to a growing need for such use in various sectors, including medicine, video processing, automated transport management, robotics, and visualisation. The object of research is cluster and cloud technologies for conducting costly physical simulations for specific sectors, particularly high-budget and entertainment ones, such as cinematography and interactive entertainment.

Research methods include using a modified Bullet engine to carry out physical simulations, integrated with OpenCL to work with the cluster. The choice of these technologies was determined by their high performance and adaptability to cluster systems. The research was based on a typical Bullet framework’s benchmark falling tower scene with the primary goal of measuring computational performance in frames per second.

Results showed that the use of clusters is not advisable in environments with a low network throughput and the use of non-uniform computers. Under those conditions, simulations using a cluster become unstable with many objects and contacts between them and show a degradation in performance by an average of 50–60% (to values of 10–20 frames per second).

Despite the intermediate results of calculations on the cluster, the study met the expectations within the goals set and resources available. These results have significant implications for the further development of cluster and cloud technologies in physical simulations, providing valuable information about the limitations and capabilities of these systems.

Keywords: 3D space, continuous space, collision solution, collision detection, cloud technologies, distributed computing, high-performance computing.
existing approaches and tools for their practical applicability [3–6]. Adjacent fields and technologies that will only be presented in the future at the time of the study were also considered. This is mainly dedicated to the commercial analogues of that research, but other open research is also considered [7].

Several existing frameworks for conducting physical simulations (physics engines or APIs) were investigated during the work, including Nvidia PhysX, BulletsPhysics, Havok, and Box2d [8, 9].

To build the cluster, several personal computers of different configurations were used. The hardware characteristics significant for this research are presented in Tables 1, 2. As a result of API investigation, the BulletsPhysics API was selected as it meets the main requirements: open access to the code, ease of use, vast and complete documentation, and free licensing. OpenCL framework was selected as the tool for organizing computations on the cluster, as it is freely available and compatible with a wide range of hardware [10]. For the space partitioning structure in the scene, Dynamic Bounding Volume Tree was chosen [11].

The practical part of the research consisted of using the components selected during the theoretical stage, forming the cluster, and conducting physical simulations in different configurations: using the computer cluster and, alternatively, the client machine.

### Table 1

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel Celeron 14125</td>
</tr>
<tr>
<td>GPU</td>
<td>Integrate into the CPU</td>
</tr>
<tr>
<td>CPU core count</td>
<td>4</td>
</tr>
<tr>
<td>Hyper-Threading</td>
<td>Presented</td>
</tr>
<tr>
<td>GPU core count</td>
<td>12</td>
</tr>
<tr>
<td>CPU tick rate</td>
<td>2.0–2.7 GHz</td>
</tr>
<tr>
<td>GPU tick rate</td>
<td>250 MHz</td>
</tr>
<tr>
<td>RAM</td>
<td>4 Gb</td>
</tr>
<tr>
<td>VRAM</td>
<td>8 Gb</td>
</tr>
</tbody>
</table>

### Table 2

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel i7-10700K</td>
</tr>
<tr>
<td>GPU</td>
<td>Nvidia 2060 Super</td>
</tr>
<tr>
<td>CPU core count</td>
<td>8</td>
</tr>
<tr>
<td>Hyper-Threading</td>
<td>Presented</td>
</tr>
<tr>
<td>GPU core count</td>
<td>2682</td>
</tr>
<tr>
<td>CPU tick rate</td>
<td>3.8 GHz</td>
</tr>
<tr>
<td>GPU tick rate</td>
<td>1450 GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>32 Gb</td>
</tr>
<tr>
<td>VRAM</td>
<td>8 Gb</td>
</tr>
</tbody>
</table>

The final part of the research involves analysing the obtained results, explaining the reasons for these results, and establishing directions for further investigation and improvements of the received result.

### 3. Results and Discussion

In the first stage of the study, computers from Tables 1, 2 were used to conduct simulations, using these computers as client machines. The primary simulation selected for this study was the fall of a tower composed of 3000 cubes. The obtained results will serve as a benchmark for comparing the results of other experiments. Fig. 1 presents the fluctuations in frames per second (FPS) depending on the simulation time. The results for the computer from Table 1 are indicated in red, and those from Table 2 are in blue. Let’s observe that machine B, being more powerful, supports the simulation more stably (the deviation of the average number of frames per second is smaller) and overall processes more frames per unit of time.

Next, it is necessary to investigate the impact of the OpenCL API on the execution of the simulations. The results of the simulations for computers A and B are presented below in Fig. 2.

![Fig. 1. The efficiency of conducting the simulation on computers A and B](image-url)
As a result of using OpenCL, the efficiency indicators of the calculations have decreased. For computer A, the average FPS rate dropped by 14.7 % (a decrease from 41.72 to 35.70), and the difference between the minimum and maximum value increased by 7 % (an increase from 2.91 to 3.12). For computer B, these indicators are 0.7 % (a decrease from 59.44 to 59.06) and 143 % (a rise from 0.37 to 0.9).

Using computers A and B, let’s build a cluster using OpenCL. The behaviour when calculating on a cluster, presented in Fig. 3, differs significantly from the results obtained in previous experiments. It can be seen that the system does not have a stable FPS indicator for most of the simulation. Let’s observe a drop in performance indicators during the first 33 % of the simulation. The simulation’s heaviest moment occurs during the tower’s fall. The largest number of contacts is observed when the most significant number of intersections between the components occurs. In total, there are 3000 cubic bodies combined into a tower on the scene.

At the beginning of the simulation, only 650 contacts between bodies are active. At the moment of the 4000 milliseconds, 18500 contacts are already active, which leads to a sharp drop in performance. The system is at rest for some time, after which the performance recovery begins due to decreased contact between bodies.

At the same time, a separate question arises: why did the FPS indicators remain stable in previous experiments even at peak loads because all experiments’ simulations are identical. It can be assumed that such a problem arises due to the need to synchronize data between cluster nodes (issue of parallelism organization by the OpenCL framework), as well as the heterogeneity of cluster elements (machine B is much more powerful than machine A) and the cluster’s small size [2, 12].

**Fig. 2.** The efficiency of conducting the simulation using OpenCL

**Fig. 3.** Conducting the simulation using the cluster
The main limitations of this study are the small hardware base available at the time of this study. Limited access to powerful GPUs has forced to use urgent resources for personal access, limiting the number of computers in the cluster and the time allocated to conduct experiments. The limitations of this study can include the absence of investigation of the architecture of the software used to perform the experiments, as most of the available API is paid. Accordingly, no practical comparisons of the impact of different space partition structures and collision detection algorithms between various object forms were conducted [11, 13].

The martial law had a substantial impact on the conduct of this study because it was through it that there was a limitation of access to a large number of computing powers (part of the available computing resources was physically damaged or completely destroyed, another part of resource due to damage to communications during the study also was not available). One of the authors was in a non-controlled territory of the country when conducting the theoretical part of the study, which also led to further economic problems in performing the work.

Further research stems directly from existing constraints and results. This is, first of all, a study of the impact of the architecture of the simulation engine on the use of distributed computing, the impact of space partition structures and collision detection algorithms. It is also worth exploring the effect of data size transmitted between cluster nodes and the client itself. It is necessary to study the impact of the homogeneity of cluster nodes on the effectiveness of the calculation.

4. Conclusions
Considering the research and related studies, using a cluster to perform physical simulations is possible. However, at the moment, the efficiency and stability of these calculations could be more satisfactory. Only using OpenCL as a backend for calculations decreases average performance from 0.7 % to 14 % (according to results for machines A and B) and reduces the simulation’s stability. The use of a cluster as a source of computational power reduced the efficiency of calculations by 45 % in the conducted experiment (compared to machine B) and 6 % (compared to machine A), which in general can be summarized as the conclusion that performance, in this case, is heading to the results of the weakest machine in the cluster (although this statement is subject to further research).

It should be noted that these results were obtained using technologies initially not intended for cluster computing, and the test scene is quite complex, considering the simple technologies initially not intended for cluster computing, and reduces the simulation's stability. The use of a cluster as a backend for calculations decreases average performance from 0.7 % to 14 % (according to results for machines A and B) and reduces the simulation’s stability. The use of a cluster as a source of computational power reduced the efficiency of calculations by 45 % in the conducted experiment (compared to machine B) and 6 % (compared to machine A), which in general can be summarized as the conclusion that performance, in this case, is heading to the results of the weakest machine in the cluster (although this statement is subject to further research).

It should be noted that these results were obtained using technologies initially not intended for cluster computing, and the test scene is quite complex, considering the simple technologies initially not intended for cluster computing, and reduces the simulation's stability. The use of a cluster as a backend for calculations decreases average performance from 0.7 % to 14 % (according to results for machines A and B) and reduces the simulation’s stability. The use of a cluster as a source of computational power reduced the efficiency of calculations by 45 % in the conducted experiment (compared to machine B) and 6 % (compared to machine A), which in general can be summarized as the conclusion that performance, in this case, is heading to the results of the weakest machine in the cluster (although this statement is subject to further research).

Conflict of interest
The authors declare that they have no conflict of interest about this research, whether financial, personal, authorship or otherwise, that could affect the study and its results presented in this paper.

Financing
The research was performed without financial support.

Data Availability
The manuscript has no associated data.

References

Oleksandr Beznoyuk, PhD, Associate Professor, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0003-2775-6070

Oleksandr Syrotiuk, Postgraduate Student, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0002-4531-6290, e-mail: oleksandr.syrotiuk.dnev@gmail.com

Corresponding author