HP ProBook Hewlett Packard laptop with a Intel(R) Core(TM) i7-4900MQ processor (4 cores with 2 threads) with a configured core frequency of 3.20 GHz, 16 GB RAM, and 256 GB SSD.

At the same time, parallel execution of a small number of operations over sc-memory (for example, 100 or 10,000 operations) in some cases can be worse than their sequential execution (Table 2). This behavior is related to the peculiarities of the con-trol mechanisms of the processes in the shared semantic memory, the classes of operations to be performed, and their specified input values in the context of the problem to be solved. For example, all sc-construction search operations with the same sc-elements, executed in parallel, do not block each other. For example, the speed of parallel execution of operations on ostis-system files depends on the size of the buffer used when reading external information constructions and writing them to disk, as well as on the length of the information constructions themselves.

Table II

Efficiency of using 4 physical threads to perform 100 sc-memory operations compared to 1 physical thread

Number of physical threads	1 thread	4 threads	
	Response time, ms	Response time, ms	Speedup, times
Operations of addition (modification)			
Operation of sc-node creation	0.099	1.306	0.076
Operation of sc-connector creation	0.150	0.422	0.356
Operation of adding content to ostis-system file	9.521	4.128	2.307
Operations of search			
Operation of searching sc-connectors outgoing from a given sc-element	0.530	0.241	2.200
Operation of searching an ostis-system file by its contents	0.339	1.453	0.233
Operations of deletion			
Operation of deleting an sc-element	0.144	1.494	0.096
Operation of deleting sc-connectors outgoing from a given sc-element	0.182	0.938	0.194

The figure 1 shows Dependence of speedup coefficient from parallel execution of a group of operations of the same class on 4 processes on the number of operations in this group, and the figure 2 shows Dependence of the execution time of a group of operations on the The figure 1 shows Dependence of speedup coefficient from parallel execution of a group of operations of the same class on 4 processes on the number of operations number of processes used

C. Efficiency of network operations over sc-memory

Network access to sc-memory is provided by the server subsystem of the ostis-systems software platform, implemented on the basis of Websocket and JSON languages (protocols) and providing network operations (commands) over sc-memory [3]. In the process of testing the implementation, the throughput of its commands was calculated. During the load testing a test client system implemented in C++ was used. The same device was used as the device used for testing operations over sc-memory. As a result, it was found out that when sending 1000 different commands: *or creating sc-elements, commands for pro-cessing contents of ostis-system files* and *commands for deleting sc-elements* — the time spent on their processing did not exceed 0.2 seconds. At the same time, in some cases it took no more than 0.14 seconds to process 1000 *commands for creating sc-elements* , while for commands *commands for deleting sc-elements* it took no more than 0.12 seconds, commands for processing the contents of ostis-system files — no more than 0.10 seconds, commands to search for sc-constructions isomorphic to a given five-element graph-template — no more than 0.45 seconds.

D. Conclusions

From the test results, it is clear that the current implementation of the ostis-systems software platform is an effective means of processing distributed information using both the software interface and the network inter-face and communication protocols.

The current *Im lementation of sc-memory provides:*

- stability in single-threaded and multi-threaded modes;
- ast speed of work in single-threaded and multi-threaded modes;
- owledge and data storage and pro-cessing in single- and multi-threaded modes.

The proposed shared semantic memory model enables efficient tracking and synchronization of parallel data accesses. The implementation of this model demonstrates a significant increase (by 2-3 orders of magnitude) in the throughput of parallel task execution compared to previous versions of the platform. However, to ensure (causal, sequential) consistency of processes and their operations, besides the data level, it is necessary to manage the knowledge level [37] .

V. Conclusion

In this paper, a model and implementation of the shared semantic memory has been proposed and discussed in detail, including (!):

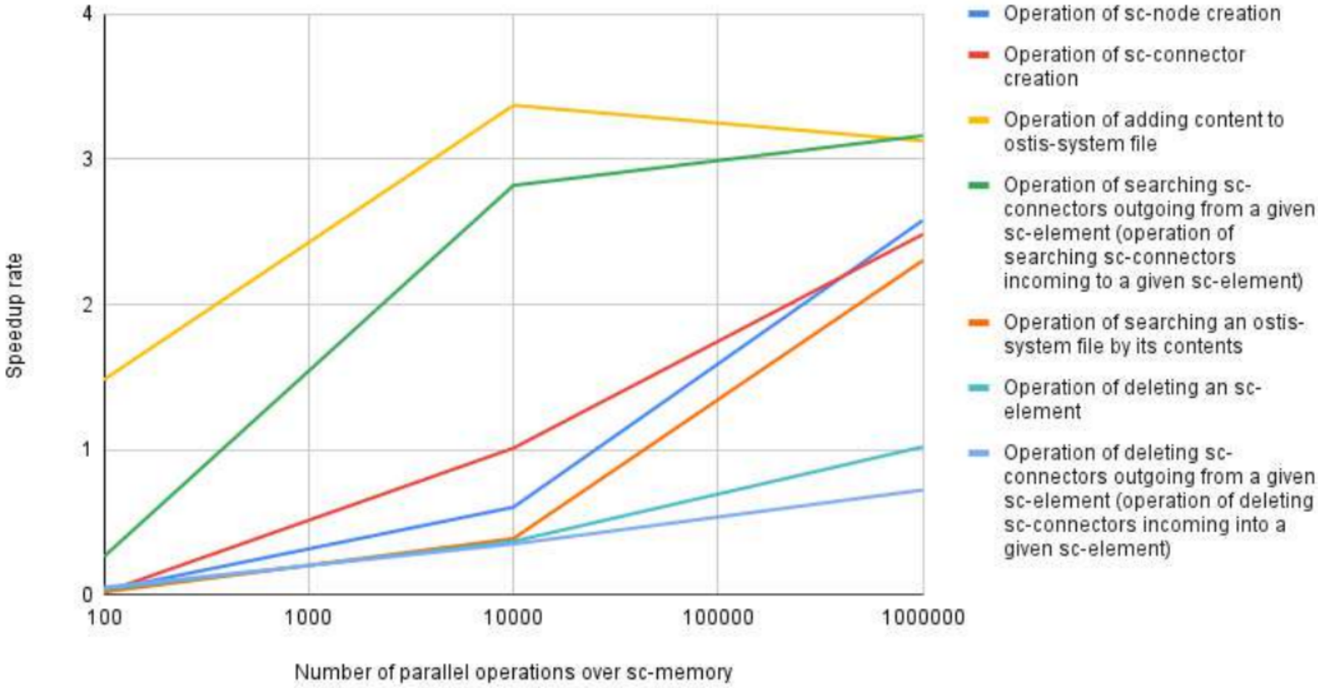

Figure 1: Dependence of speedup coefficient from parallel execution of a group of operations of the same class on 4 processes on the number of operations in this group

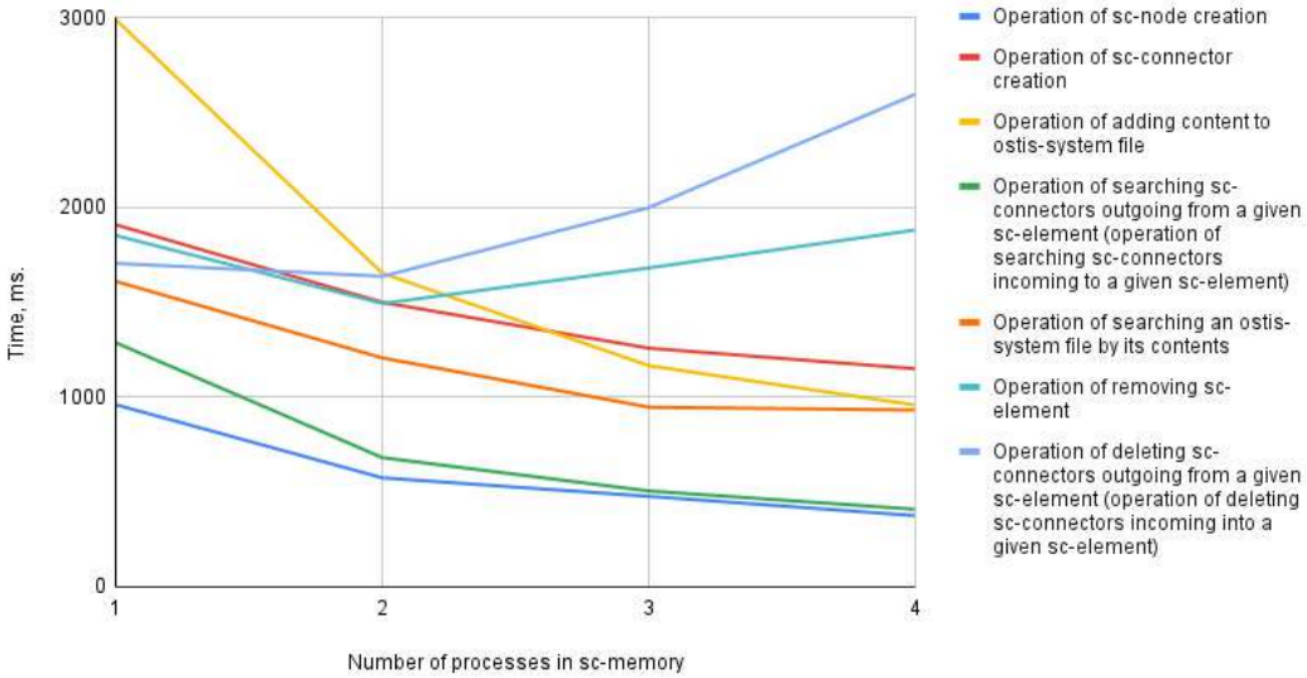

Figure 2: Dependence of the execution time of a group of operations on the number of processes used

- a storage for unified representation and processing of graph constructions;for unified representation and processing of graph constructions;
- storage for unified representation of string constructions used as file contents in graph constructions;
- storage for managing events in this memory;
- a storage for managing processes running in this memory;
- a set of operations for working with this memory. The proposed model of the shared semantic memory includes:
- models and algorithms for allocating and releasing cells in this memory, providing:
  – reusability of the released memory segments;
  – ability to utilize new vacant memory segments;
- Models and algorithms to efficiently allocate processes in this memory;
  – rapid parallel creation of elements in the memory by allocating processes over the segments of the memory;
  – fast unblockable parallel search of constructions, provided there are no other operations on these constructions.
- Models and algorithms for managing subscriptions to events in this memory;
- odels and algorithms for synchronizing the execution of processes in the shared memory sections, providing:
  – parallel access to sc-memory, i.e. possibility of parallel execution of actions in sc-memory with-out violating correctness of the data structures in it;
  – absence of deadlocks, races and hungry processes operating in sc-memory.

  Promising directions to further this line of work are:
  The formally described model of semantic memory
- evelopment of a model for distributedof information in the unified semantic memory unified representation and processing of information in the unified semantic memory;
- evelopment of a model for representation and stor-age of platform-dependent agent programs;a model for representation and stor-age of platform-dependent agent programs agent programs;a model for representation,
- evelopment of a consistency model to ensure correctness of agents' operation on constructions in the memory;
- development of a model of memory configuration from the memory itself.

  In addition, other equally important areas of work are:
- improving the documentation of the current Imple-mentation of sc-memory and the current Software implementation of ostis-platform;
- improvement of methodologies and tools for devel-oping documentation of software systems;sc-memory and the current Software implementation of ostis-platform;
- improvement and mass distribution of the Software implementation of ostis-platform and intelligent sys-tems developed on its basis.improvement and mass distribution of the Software implementation.

The formally described model of semantic memory is consistent with the previously described ontological model of this memory [3]. The author of this paper believes that the used approach to modeling of complex objects will help to simplify the understanding of the operation of intelligent systems developed according to the principles of the OSTIS Technology

Acknowledgment

The author would like to thank the scientific teams of the departments of Intelligent Information Technologies of Belarusian State University of Informatics and Radio-electronics and Brest State Technical University for their help in the work and valuable comments.

References

[1] N. Zotov, "Design principles, structure, and development prospects of the software platform of ostis-systems," *Otkrytye se-manticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, pp. 67—76, 2023. [2] ——, "Implementation of information retrieval subsystem in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open se-mantic technologies for intelligent systems]*, pp. 77–94, 2023. [3] ——, "Software platform for next-generation intelligent computer systems," in *Otkrytye semanticheskie tekhnologii [Open semantic technologies for intelli-gent systems]* BSUIR, Minsk, 2022, pp. 297—326. [4] V. Golenkov, N. Guliakina, and D. Shunkevich, *Otkrytaja tehnologija ontologicheskogo proektirovanija, proizvodstva i jekspluatacii semanticheski sovmestimyh gibridnyh intellektual sistem [Open technology of ontological design, production and operation of semantically compatible hybrid intelligent computer systems]*, V. Golenkov, Ed. Minsk: Bestprint [Bestprint], 2021. [5] V. Golenkov et al., "Associative semantic computers for intelligent computer systems of a new generation," in *Otkrytyetekhnologii proektirovaniya intellektual'nykh system [Open Semantic Technologies for Intelligent Systems (OSTIS)]*, vol. 7. BSUIR, Minsk, 2023, pp. 39–60.

[6] V. Golenkov, "Ontology-based design of intelligent systems," *rytye semanticheskie tekhnologii proektirovaniya intellek-tual'nykh system [Open semantic technologies for intelligent systems]* , pp. 37–56, 2017.

[7] D. Shunkevich, "Ontology-based design of hybrid problem solvers," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelli-gent systems]*. Cham: Springer International Publishing, 2022, pp. 101–131.

[8] S. Auer, V. Kovtun, M. Prinz, A. Kasprzik, M. Stocker, and M. E. Vidal, "Towards a knowledge graph for science," in *Proceedings of the 8th international conference on web intelligence, mining and semantics,* 2018, pp. 1–6.

[9] V. Golenkov, Ed., *Tehnologija kompleksnoj podderzhki zhiznennogo cikla semanticheski sovmestimyh intellektual'nyh komp'juternyh sistem novogo pokolenija [Technology of complex life cycle support of semantically compatible intelligent computer systems of new generation]* Bestprint, 2023.

[10] A. Palagin and N. Petrenko, "To the question of system-ontological integration of subject area knowledge," *Mathematical Machines and Systems* , vol. 1, no. 3-4, pp. 63–75, 2007